

Radiant Service Desk

Руководство Администратора

Дата создания:

2022-09-08

RSD – Руководство Администратора

Copyright © 2010-2023 Радиант Систем

Эта работа защищена авторским правом ООО «Радиант Систем».

Это руководство может быть скопировано целиком или частично, при условии, что это уведомление об авторских правах будет включено в каждую копию.

Все торговые названия используются без гарантии их свободного использования и, возможно, являются зарегистрированными торговыми марками. Все продукты, упомянутые в данном руководстве, могут быть товарными знаками соответствующего производителя.

Оглавление

Предисловие.....	5
1. Введение.....	6
1. Назначение системы.....	6
2. Требования к аппаратному и программному обеспечению.....	6
3. Сообщество.....	7
4. Профессиональные услуги для RSD.....	7
2. Установка.....	8
1. Подготовка операционной системы.....	8
2. Установка RSD.....	11
3. Первоначальные настройки RSD.....	14
4. Дополнительные приложения.....	23
3. Первые шаги.....	24
1. Веб-интерфейс агента.....	24
2. Веб-интерфейс клиента.....	25
3. Общедоступный веб-интерфейс.....	25
4. Первая авторизация.....	26
5. Веб-интерфейс - обзор.....	27
6. Панель Управления/Обзор.....	28
7. Что такое очередь?.....	34
8. Что такое просмотр очередей?.....	34
9. Пользовательские настройки.....	35
4. Администрирование.....	39
1. Панель администрирования RSD.....	39
1.1. Основы.....	39
1.2. Агенты, Группы и Роли.....	39
1.3. Клиенты и Группы Клиентов.....	48
1.4. Очереди.....	55
1.5. Приветствия, Подписи, Вложения и Шаблоны.....	57
1.6. Автоответы.....	63
1.7. Системные адреса электронной почты.....	66
1.8. Уведомления по заявкам.....	67
1.9. S/MIME.....	70
1.10. PGP.....	70
1.11. Состояния.....	70

1.12. System Configuration.....	71
1.13. Использование учетных записей почты	72
1.14. Фильтрация входящих почтовых сообщений.....	72
1.15. Выполнение автоматизированных заданий с помощью Планировщика задач	78
1.16. Административные сообщения.....	79
1.17. Управление сеансами.....	80
1.18. Обслуживание системы.....	81
1.19. Системный журнал.....	82
1.20. Журнал сеансов связи.....	82
1.21. SQL-запросы.....	86
1.22. Менеджер пакетов	86
1.23. Веб-сервисы	89
1.24. Динамические поля.....	90
1.25. Календари.....	90
2. Настройка Системы	90
2.1. Конфигурационные файлы RSD	90
2.2. Настройка системы через веб-интерфейс.....	91
3. Резервное копирование системы.....	93
3.1. Резервное копирование.....	93
3.2. Восстановление	94
4. Настройки электронной почты.....	94
4.1. Отправка/получение электронных писем.....	94
4.2. Защищенная электронная почта с помощью PGP	104
4.3. Защищенная электронная почта с помощью S/MIME.....	109
5. Использование внешних бэкендов.....	115
5.1. Данные клиентов.....	115
5.2. Серверная часть пользователя клиента.....	116
5.3. Хранилища данных (бэкенды) для аутентификации агентов и клиентов.....	131
5.4. Настройка саморегистрации клиента	139
6. Настройки заявок.....	143
6.1. Состояния заявок.....	143
6.2. Приоритеты заявок	148
6.3. Ответственность за Заявку & Наблюдение за Заявкой	148
7. Функции, связанные с датой и временем	151
7.1. Настройка рабочих часов, праздников и часовых поясов	151
7.2. Автоматическая разблокировка.....	152

8. Настройка вывода PDF	153
9. Отчеты	153
9.1. Конфигурация и использование статистики	153
9.2. Администрирование системы статистики	159
10. Динамические Поля.....	160
10.1. Введение.....	160
10.2. Настройка	161
11. Универсальный интерфейс (Generic Interface).....	179
11.1. Уровни универсального интерфейса.....	179
11.2. Информационный поток универсального интерфейса	182
11.3. Веб-сервисы	184
11.4. Графический интерфейс Веб-сервиса	184
11.5. Интерфейс командной строки Веб- Сервиса	205
11.6. Конфигурация Веб-сервиса.....	205
11.7. Коннекторы (Connectors)	212
12. Демон RSD (The RSD Daemon)	255
12.1. Графический интерфейс демона RSD	257
12.2. Интерфейс командной строки демона RSD.....	258
5. Настройка.....	259
1. Списки Управления Доступом(ACLs)	259
2. Управление Процессами.....	278
6. Настройка производительности	354
1. RSD	354
2. База данных	358
3. Веб-сервер	359

Предисловие

Эта инструкция предназначена для использования администраторами RSD. Она также послужит хорошим справочником для новичков RSD.

В следующих главах описывается установка, настройка и администрирование программы RSD. В первой трети текста описываются ключевые функциональные возможности программного обеспечения, в то время как остальной текст является ссылкой на полный набор настраиваемых параметров.

1. Введение

1. Назначение системы

Программное обеспечение Radiant Service Desk (RSD) обеспечивает поддержку пользователей и автоматизацию бизнес-процессов поддержки для внутренних и внешних команд любого масштаба в ней нуждаетесь, а также, каким образом можно связаться с сообществом.

Radiant Service Desk обеспечивает автоматизацию следующих процессов:

Управление Инцидентами

- Регистрация и управление
- Приоритезация и категоризация
- Маршрутизация и совместная работа

Управление Запросами

- Регистрация и категоризация
- Маршрутизация на основе Сервисов
- Многопоточные работы и согласования

Управление каталогом услуг и SLA

- Поддержка многоуровневой структуры
- Автоматизация обработки заявок
- Бизнес-часы и календари

База знаний

- Интеграция с Клиентским порталом
- Оценка качества статей
- Тэги и вложения

Управление Проблемами

- Диагностика и решение
- Интеграция с базой знаний
- Связи с Инцидентами, изменениями

2. Требования к аппаратному и программному обеспечению

Требования к программному обеспечению

Поддержка Perl

- Perl 5.16 или выше

Поддержка Веб-сервера

- Apache2 + mod_perl2 или выше (рекомендуется)
- Веб-сервер с поддержкой CGI-интерфейса (CGI-интерфейс не рекомендуется)

Поддержка базы данных

- MySQL 5.0 или выше (MySQL 8+ не поддерживается)
- MariaDB
- PostgreSQL 9.2 или выше

Для установки RSD используются бинарные пакеты, собранные специально под вашу операционную систему (rpm), который включает в себя все необходимые модули языка Perl, или пакеты, содержащие все необходимые Perl-модули. Менеджер пакетов вашей системы должен позаботиться о зависимостях необходимых модулей Perl.

Поддержка веб-браузера

При использовании RSD необходимо использовать последние версии браузеров с включенной поддержкой JavaScript. Поддерживаемые браузеры:

- Microsoft Edge
- Mozilla firefox 100.x;
- Opera 90.x;
- Google chrome 105.x;
- Yandex browser 23.1.x.
- Internet Explorer не ниже версии 11.0
- Apple Safari (в том числе мобильная версия)

Мы рекомендуем использовать самые последние версии браузеров, т.к. они всегда обладают наилучшей производительностью JavaScript и рендеринга. Критическая разница в производительности может возникать при больших объемах данных при использовании старых версий. Мы всегда рады проконсультировать вас по этому вопросу.

3. Сообщество

RSD имеет большое сообщество пользователей. Пользователи и разработчики обсуждают RSD и обмениваются информацией по связанным вопросам через [форум](#) и ТГ-каналы для [русскоязычных](#) и [англоязычных](#) пользователей. Домашняя страница RSD-сообщества: <https://radiantsystem.ru/blog>.

4. Профессиональные услуги для RSD

[Radiant System](#) предлагает вам профессиональную поддержку от нашей команды, надежные и регулярные бесплатные обновления безопасности, а также специальный набор дополнительных бизнес - функций, которые вы можете легко подключать или отключать, в зависимости от выбранного сценария внедрения.

[Radiant System](#) также предлагает специальные [программы обучения](#). Вы можете воспользоваться внутренним обучением, которое охватывает все конкретные потребности вашей компании.

2. Установка

В этой главе описывается установка и базовая настройка центральной структуры RSD. Здесь вы найдете информацию об установке RSD на операционную систему Centos 7.

В этой главе рассматриваются такие вопросы как: конфигурация веб-сервера и сервера базы данных, интерфейс между RSD и базой данных, установка дополнительных Perl-модулей, установка соответствующих прав доступа для RSD, настройка планировщика задач cron jobs для RSD и основных параметров в конфигурационных файлах.

Следуйте пошаговой инструкции для установки RSD на вашем сервере. После успешной установки системы вы сможете использовать ее веб-интерфейс для дальнейшего администрирования.

1. Подготовка операционной системы

1. Добавление репозитория EPEL

```
yum install epel-release -y
```

2. Установка tmux или screen

Установка tmux или screen не является обязательным, но мы рекомендуем использовать эти инструменты для работы. Если вы не пользуетесь ни тем, ни другим, уверены в канале связи, то можете устанавливать RSD без них.

```
yum install tmux -y
```

После установки надо перейти в консоль tmux:

```
tmux
```

3. Смена имени хоста

```
tmuxhostnamectl set-hostname host_name  
  
systemctl restart systemd-hostnamed  
  
hostnamectl status
```

4. Установка часового пояса

```
timedatectl set-timezone Europe/Moscow или Europe/Berlin или другой
```

5. Обновление yum

```
yum update -y
```

6. Установка дополнительных инструментов

Не все дополнительные инструменты требуются для установки RSD, часть мы рекомендуем для удобства работы.

```
yum install net-tools vim wget man vim-enhanced screen mc telnet wget crontabs  
openssh-clients mailx sudo make gcc htop nmon sysstat sysbench hdparm lsof -y
```

7. Проверка и настройка межсетевого экрана (firewall)

На некоторых системах межсетевого экрана нет вообще или он отключен. Если межсетевой экран установлен, мы увидим список открытых протоколов. В случае, если список будет пустой:

```
firewall-cmd --list-all
```

нам необходимо установить межсетевой экран.

```
yum install firewalld -y  
  
systemctl enable firewalld  
  
systemctl start firewalld
```

После установки межсетевого экрана убираем "dhcpv6" и добавляем "http(https)":

```
firewall-cmd --permanent --zone=public --remove-service=dhcpv6-client  
  
firewall-cmd --permanent --zone=public --add-service=http  
  
firewall-cmd --permanent --zone=public --add-service=https
```

8. Изменение номера порта для ssh

Обычно для доступа по ssh по умолчанию используется 22 порт, что является небезопасным. Мы рекомендуем изменять порт для доступа к серверу по ssh:

```
vim /etc/ssh/sshd_config  
  
## Port 22 -> Port 22222
```

После изменения номера порта необходимо внести изменения в сервис межсетевого экрана

```
vim /usr/lib/firewalld/services/ssh.xml  
  
<port protocol="tcp" port="22222"/>
```

и применить настройки.

```
systemctl restart sshd && systemctl restart firewalld
```

9. Настройка дополнительной защиты сервера

Настройка защиты от брутфорса (brute-force):

```
yum install fail2ban -y  
  
systemctl enable fail2ban  
  
systemctl start fail2ban  
  
systemctl status fail2ban  
  
echo '  
  
[DEFAULT]  
  
# Ban hosts for one hour:  
  
bantime = 3600  
  
# Override /etc/fail2ban/jail.d/00-firewalld.conf:  
  
banaction = iptables-multiport  
  
[sshd]  
  
enabled = true' > /etc/fail2ban/jail.local  
  
systemctl restart fail2ban
```

Отключение SELinux:

```
sed -i 's/^SELINUX=.*SELINUX=disabled/g' /etc/selinux/config  
sed -i 's/^SELINUX=.*SELINUX=disabled/g' /etc/sysconfig/selinux  
setenforce 0
```

Важно

После выполнения вышеуказанных настроек желательно выполнить перезапуск системы (reboot) и проверить, что все было сделано и сохранено.

2. Установка RSD

1. Установка веб-сервера

```
yum install httpd -y  
systemctl enable httpd  
systemctl start httpd
```

2. Установка СУБД

PostgreSQL

Рассмотрим установку на примере версии 12.

```
yum install postgresql-server postgresql-contrib  
/usr/bin/postgresql-setup initdb  
systemctl enable postgresql  
systemctl start postgresql
```

Добавим доступ на локальном хосте по паролю:

```
vim /var/lib/pgsql/data/pg_hba.conf  
Меняем #host all all 127.0.0.1/32 ident  
На host all all 127.0.0.1/32 password
```

Установим пароль для пользователя postgresql:

```
sudo -u postgres psql  
  
alter role postgres with password 'свой_пароль';  
  
\q  
  
systemctl restart postgresql
```

Примечание

Установка согласно данной инструкции подходит для большинства систем, но, в случае с высоконагруженной системой, рекомендуем провести дополнительные настройки базы данных.

3. Установка и актуализация Perl модулей

Установка основных модулей:

```
yum install fetchmail perl-URI mod_perl perl-libwww-perl perl-Net-DNS perl-IO-Socket-SSL perl-XML-Parser perl-DBI procmail perl-DateTime-Format-DateParse perl-core perl-Crypt-SSLeay perl-JSON-XS perl-PDF-API2 perl-Text-CSV_XS perl-Digest-SHA perl-LDAP perl-Encode-HanExtra perl-Text-CSV perl-YAML-LibYAML fcgi perl-Archive-Zip perl-Template-Toolkit perl-XML-LibXSLT bash-completion perl-DBD-Pg perl perl-File-MimeInfo perl-DBD-MySQL -y
```

Обновление perl модулей:

```
#Ставим cranm  
  
yum install perl-App-cranminus -y  
  
#Обновляем cranm  
  
cranm App::cranminus  
  
#Удаляем cranm установленный из репо  
  
yum erase perl-App-cranminus  
  
#Создаем линк для новой версии  
  
ln -s /usr/local/bin/cranm /bin/cranm  
  
#Ставим cranoutdated  
  
cranm App::cranoutdated  
  
#Обновляем все пакеты, которые считаются устаревшими
```

```
/usr/local/bin/cpan-outdated -p | cpanm
```

```
#Поскольку outdated обновляет не все пакеты, то дальше руками
```

```
cpanm      Apache::DBI      Apache2::Reload      Archive::Tar      Archive::Zip  
Crypt::Eksblowfish::Bcrypt Crypt::SSLeay Date::Format DateTime DBI DBD::Pg  
Digest::SHA Encode::HanExtra JSON::XS List::Util::XS LWP::UserAgent  
Mail::IMAPClient IO::Socket::SSL Authen::SASL Authen::NTLM ModPerl::Util Net::DNS  
Net::LDAP Net::SMTP Template Template::Stash::XS Text::CSV_XS Time::HiRes  
XML::LibXML XML::LibXSLT XML::Parser YAML::XS
```

```
cpanm Archive::Zip XML::LibXML DBD::File Moo Date::Calc
```

Важно

Такой способ обновления можно выполнять только на тестовых системах или при первоначальной установке. Если требуется выполнить обновление на продуктовой системе, обязательно сделайте снапшот или резервное копирование.

4. Установка RSD

Установка последней версии из репозитория:

```
useradd radiant -c 'Radiant user' -d /opt/radiant/  
  
usermod -aG apache radiant  
  
cd /opt && wget https://download.radiantsd.ru/latest/radiant.tar.gz  
  
tar xvf radiant.tar.gz  
  
cp /opt/radiant/Kernel/Config.pm.dist /opt/radiant/Kernel/Config.pm  
  
sudo /opt/radiant/bin/SetPermissions.pl --radiant-user=radiant --web-group=apache  
/opt/radiant  
  
sudo ln -s /opt/radiant/scripts/apache2-httpd.include.conf \  
/etc/httpd/conf.d/zzz_radiant.conf
```

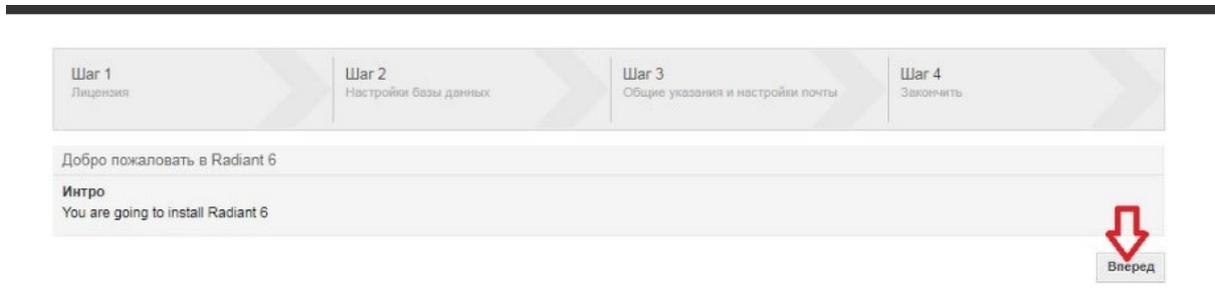
Перезапускаем апач и идём на веб интерфейс по ссылке в конце установки пакета. Вы можете написать его сами по аналогии с примером, если хостнейм и fqdn не совпадают, например, “http://youserver/radiant/installer.pl”.

```
systemctl restart httpd
```

3. Первоначальные настройки RSD

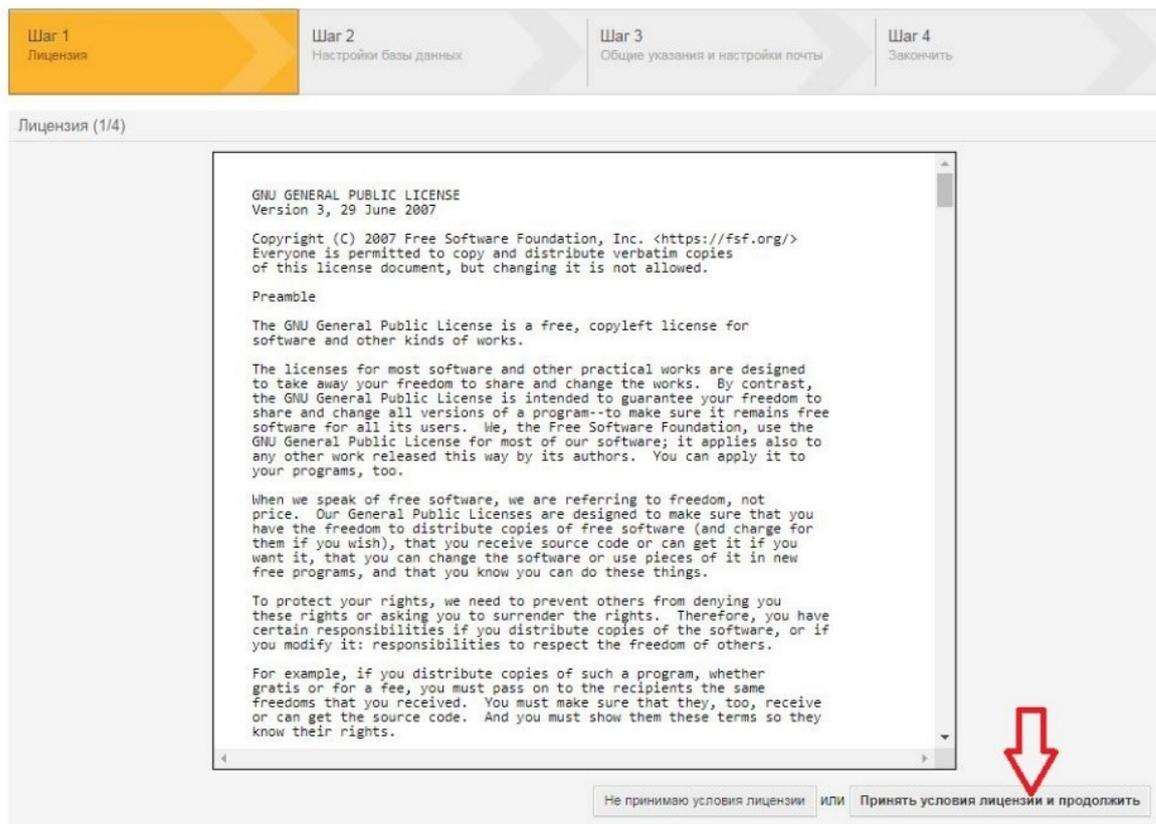
1. Веб конфигурация - настройка базы данных

Окно приветствия. Нажимаем «Вперед»



Шаг 1 – Лицензия

Принимаем условия лицензии и продолжаем настройку.



Шаг 2 – Настройки базы данных

Выбираем свою СУБД, в нашем случае это PostgreSQL.

Нажимаем «Вперед».

The screenshot shows a four-step progress bar at the top: Step 1 (Лицензия), Step 2 (Настройки базы данных), Step 3 (Общие указания и настройки почты), and Step 4 (Закончить). The main content area is titled "Выбор базы данных (2/4)". It features a dropdown menu for "Тип:" with options MySQL, Oracle, and PostgreSQL. A red arrow points to the PostgreSQL option. Below the dropdown, there are two radio buttons under "Тип установки:". The first is selected and labeled "Создать новую базу данных для Radiant", with a red arrow pointing to it. The second is "Использовать существующую базу данных Radiant". A red arrow points to the "Вперед" button in the bottom right corner.

Используется в Radiant System™

Вводим учетные данные, в нашем случае «пароль». Затем нажимаем «Проверить настройки базы данных».

The screenshot shows the same four-step progress bar. The main content area is titled "База данных (2/4)". It contains three input fields: "Пользователь:" with the value "postgres", "Пароль:" with masked characters "*****", and "Сервер:" with the value "127.0.0.1". Red arrows point to each of these fields. Below the fields is a button labeled "Проверить настройки БД", with a red arrow pointing to it. In the bottom right corner, there are two buttons: "Назад" and "Вперед".

Используется в Radiant System™

После успешной проверки СУБД нажимаем «Вперед».

Шаг 1 Лицензия Шаг 2 Настройки базы данных Шаг 3 Общие указания и настройки почты Шаг 4 Закончить

База данных (2/4)

Пользователь: postgres
 Пароль:
 Сервер: 127.0.0.1

Результат проверки базы данных
 ✓ База данных проверена успешно.

Пользователь базы данных (Новое)
 Пользователь: radiant
 Для этой системы Radiant будет создан новый пользователь базы данных с ограниченными правами.
 Пароль:
 Повторите пароль:
 Сгенерированный пароль: 1qYCiXuYZsnTuF

База данных
 Имя базы данных: radiant

Назад Вперед

Используется в Radiant System™

Система автоматически выполнит необходимые действия в СУБД и выведем отчет о выполненной работе.

Нажимаем «Вперед».

Шаг 1 Лицензия Шаг 2 Настройки базы данных Шаг 3 Общие указания и настройки почты Шаг 4 Закончить

Создать базу данных (2/4)

- ✓ CREATE ROLE "radiant"
- ✓ CREATE DATABASE "radiant"
- ✓ Processing radiant-schema
- ✓ Processing radiant-initial_insert
- ✓ Processing post statements

База данных настроена успешно!

Вперед

Используется в Radiant System™

Шаг 3 – Общие указания и настройки почты

Заполняем название компании, если необходимо, почту администратора и убираем проверку MX.

Нажимаем «Вперед».

Шаг 1 Лицензия	Шаг 2 Настройки базы данных	Шаг 3 Общие указания и настройки почты	Шаг 4 Закончить
-------------------	--------------------------------	---	--------------------

Системные параметры (3/4)

SystemID:

Идентификатор системы. Каждый номер заявки и каждый идентификатор HTTP-сессии - содержат этот номер.

Системное FQDN:

Полное доменное имя вашей системы.

Адрес администратора:

Адрес электронной почты системного администратора.

Организация:

Журнал

Модуль журнала:

Какой бэкенд использовать для собственно записи журнала.

Файл журнала:

Веб-интерфейс

Язык по умолчанию:

Язык по умолчанию.

Проверять записи MX:

Вручную вводимые адреса электронной почты будут проверяться на MX-записи в DNS. Не используйте эту опцию, если ваш DNS-сервер медленный или не разрешает публичные адреса.

Используется в Radiant System™

Настройки почты мы рекомендуем пропустить, лучше производить настройки почты из веб-интерфейса.

Нажимаем «Пропустить этот шаг».

Шаг 1 Лицензия	Шаг 2 Настройки базы данных	Шаг 3 Общие указания и настройки почты	Шаг 4 Закончить
-------------------	--------------------------------	---	--------------------

Конфигурация почты (3/4)

Конфигурация исходящей почты

Тип исходящей почты:

Выберите протокол/способ для отправки исходящей почты.

Порт исходящей почты:

Выберите порт исходящей почты.

Конфигурация входящей почты

Тип входящей почты:

Выберите протокол/способ для получения входящей почты.

Почтовый сервер для входящей почты:

Почтовый сервер для входящей почты.

Имя пользователя для входящей почты:

Имя пользователя для входящей почты.

Пароль для входящей почты:

Пароль для входящей почты.

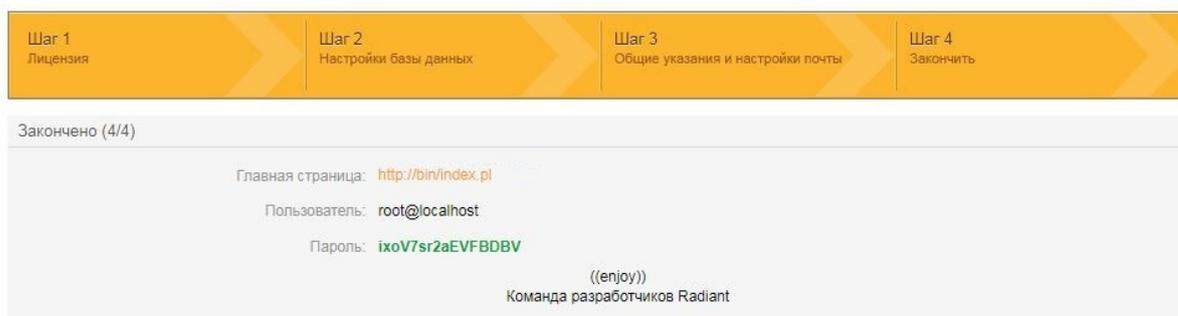
или

Используется в Radiant System™

Шаг 4 – Завершение установки

После выполнения всех шагов, система отобразит данные для входа в веб интерфейс:

- Главная страница
- Пользователь
- Пароль



Используется в Radiant System™

Все, основная часть установки успешно завершена.

2. Запуск Daemon и Cronjobs

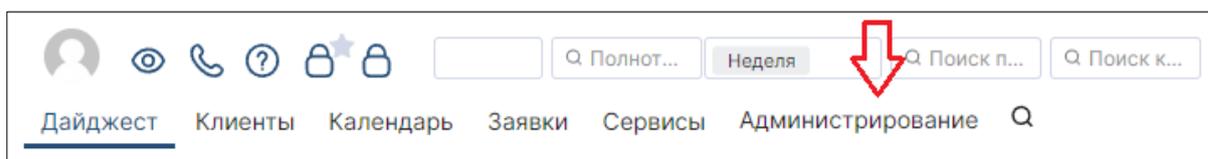
Запускаем службу Cron.

```
/opt/radiant/bin/Cron.sh start radiant
```

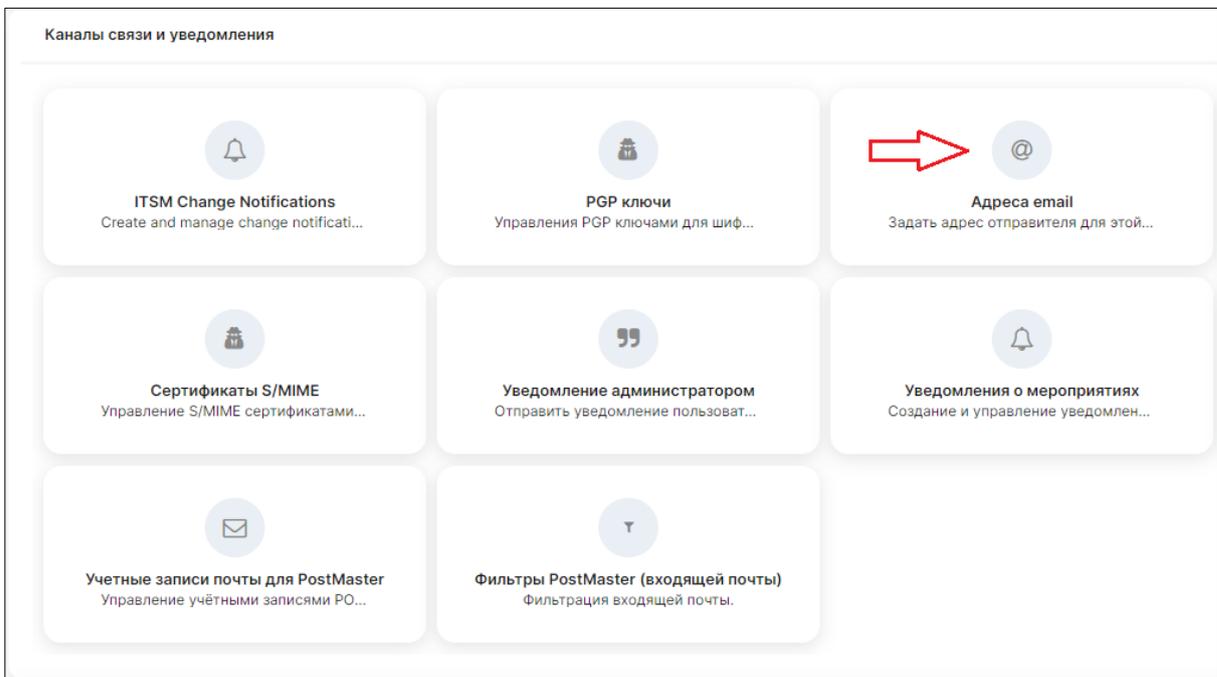
После этого выполняем проверку, что задание (Cronjob) появилось. В нашем случае задание одно, это запуск Daemon, который в свою очередь управляет остальными заданиями.

```
crontab -u radiant -l
```

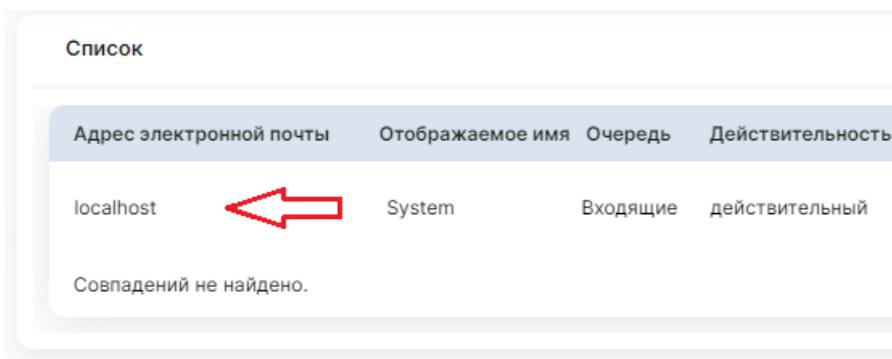
3. Настройка электронной почты



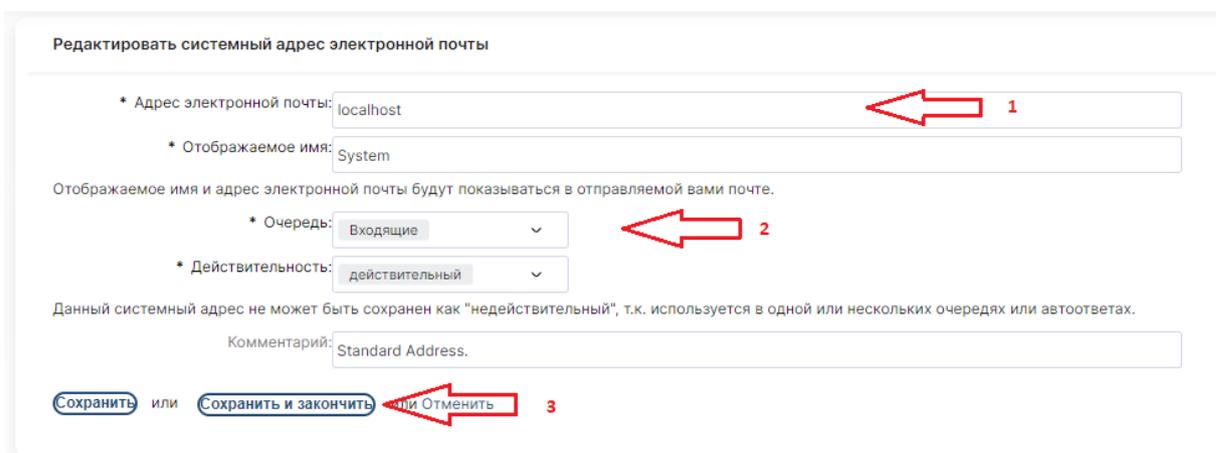
Заходим в агентский интерфейс с выданными нам учётными данными. Переходим в раздел «Администрирование».



Почта настраивается в 3-х (трёх) местах. Сначала настраиваем адрес, от имени которого система будет отправлять письма из очереди. Переходим с адреса email.



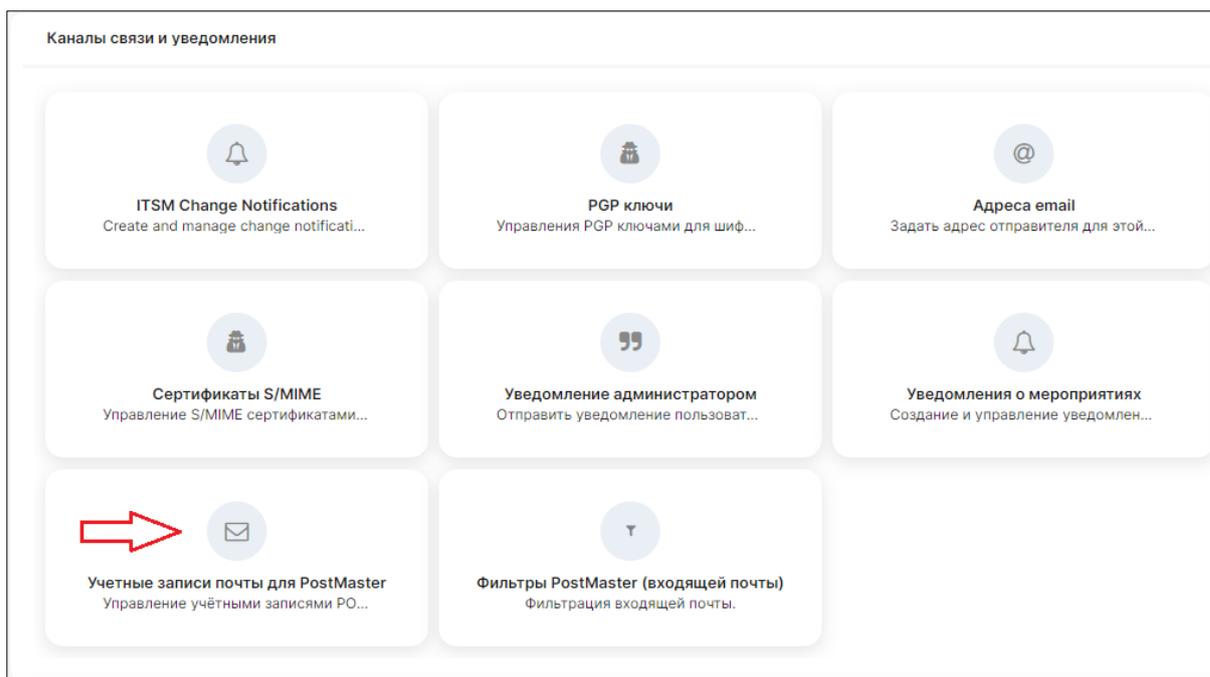
Выбираем localhost. Будем менять его на реальный адрес.



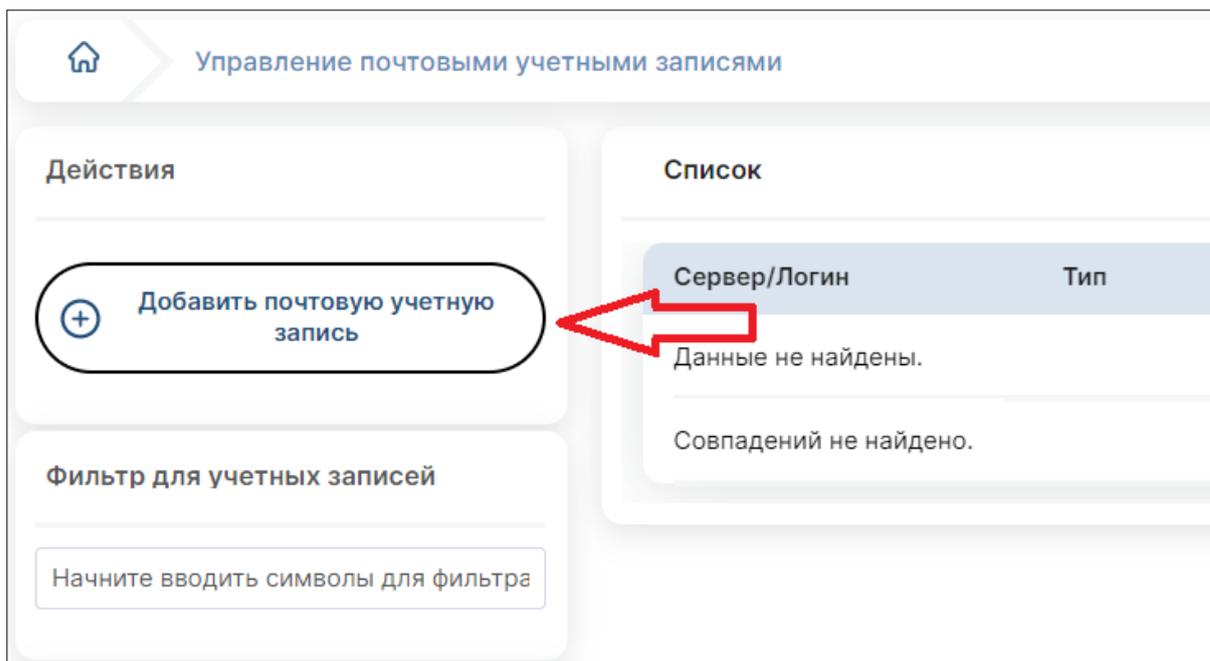
Вписываем реальный адрес (на картинке 1), от которого “Radiant ServiceDesk” будет отправлять письма. Выбираем (на картинке 2), из какой очереди будут уходить уведомления и ответы под этим именем, в данном случае - “Входящие”. Если запись одна, то из всех очередей отправка будет от этой записи. Если создать несколько, то

от каждой указанной очереди письма будут отправляться от имени того, кого вы укажете. Сохраняем и выходим (на картинке 3).

Настраиваем входящую почту:



Переходим в «Администрирование» – «Учетные записи почты для Postmaster».



Нажимаем «Добавить почтовую учетную запись».

Добавить почтовую учетную запись

1 * Тип: IMAP

2 * Логин:

3 * Пароль:

4 * Сервер:

Пример: mail.example.com

Папка IMAP: INBOX

Измените только в том случае, если нужно забирать почту из папки, отличной от INBOX.

* Доверенная: Нет

5 * Перенаправление: Перенаправление по выбранной очереди

6 * Очередь: 1-я линия поддержки

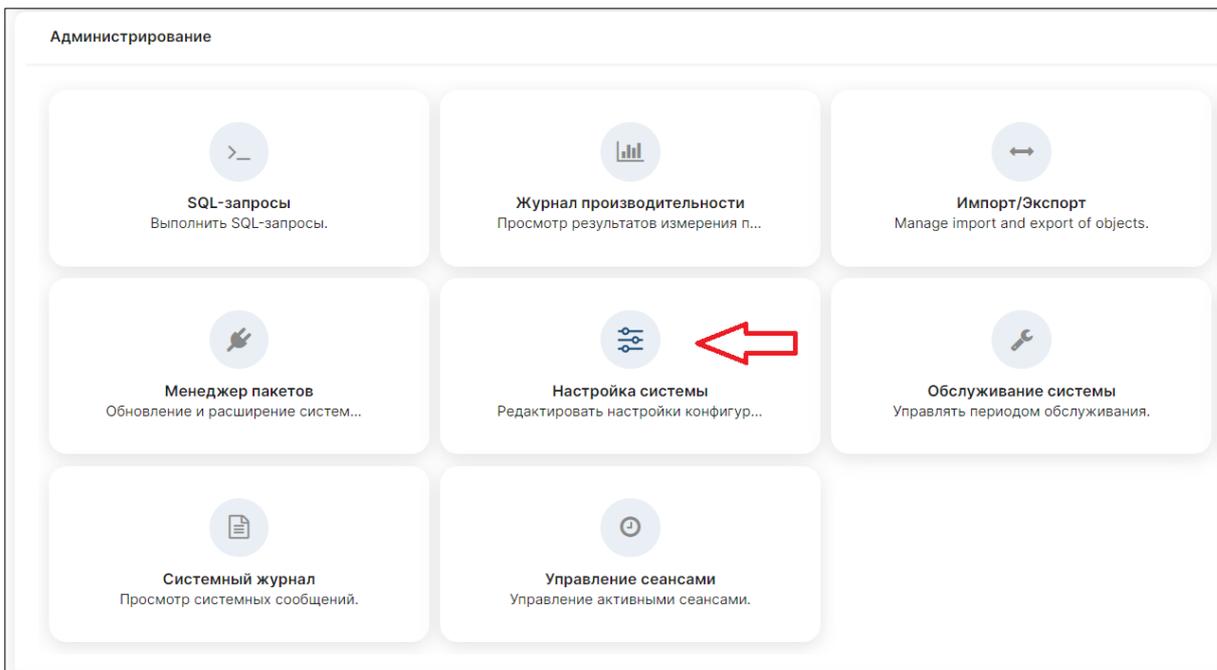
* Действительность: действительный

Комментарий:

7 [Сохранить](#) или Отменить

- (1) Выберите протокол. Это может быть IMAP, POP3 или его разновидность.
- (2) Заполните логин.
- (3) Введите пароль.
- (4) Ставим хост. Если порт отличается от порта по умолчанию, вы можете указать mail.yourmailserver.com:port.
- (5) Отправка. Выберите «Отправка по выбранной очереди».
- (6) Выберите очередь для получения писем и создания тикетов с этой учетной записи электронной почты.
- (7) Сохранить.

Дальше настраиваем исходящую почту:



«Администрирование» -> «Настройка Системы». Далее переходим в раздел «Навигация» -> «Core» -> «Email».

SendmailModule	← 1	Kernel::System::Email::SMTPS	Defines the module to send emails. "DoNotSendEmail" doesn't send emails at all. Any of the "SMTP" mechanisms use a specified (external) mailserver. "Sendmail" directly uses the sendmail binary of your operating system. "Test" doesn't send emails, but writes them to \$OTRS_HOME/var/tmp/CacheFileStorable/EmailTest/ for testing purposes.
SendmailModule::AuthPassword	← 2	Если любой из "SMTP" механизмов был выбран для SendmailModule, и аутентификация на почтовом сервере необходима, пароль должен быть задан.
SendmailModule::AuthUser	← 3	MailservLogin	Если любой из "SMTP" механизмов был выбран для SendmailModule, и аутентификация на почтовом сервере необходима, username должно быть задано.
SendmailModule::CMD		/usr/sbin/sendmail -i -f	Если "Sendmail" было выбрано для SendmailModule, расположение модулей sendmail и требуемых параметров должно быть задано.
SendmailModule::Host	← 4	mail.example.com	Если любой из "SMTP" механизмов был выбран для SendmailModule, mailhost, который отправляет почту, должен быть задан.
SendmailModule::Port	← 5	25	Если любой из "SMTP" механизмов был выбран для SendmailModule, порт, на котором, почтовый сервер проверяет входящие соединения, должен быть задан.

- (1) Прописываем протокол SMTP, SMTPS, SMTPTLS. Или MultiSMTP, если он установлен. Тогда настройки исходящей почты делаются в соответствующем пункте «Администрирование» – «SMTPнастройки».
- (2) Вписываем пароль.
- (3) Вписываем логин.
- (4) Вписываем имя почтового сервера для исходящей почты.
- (5) Вписываем порт, если он отличен от дефолтного.
- (6) Подтверждаем изменение (данное действие необходимо выполнить в каждом пункте).

Выполним настройки хранения статей, чтобы не перегружалась база данных и логи. Находим настройку в разделе «Администрирование» - «Настройки системы»: Ticket::Article::Backend::MIMEBase::ArticleStorage и изменяем на значение «ArticleStorageFS».

Ticket::Article::Backend::MIMEBase::ArticleStorage

Kernel::System::Ticket::Article::Backend::MIMEBase::ArticleStorageDB

Kernel::System::Ticket::Article::Backend::MIMEBase::ArticleStorageFS



Нажимаем применить изменения, проверяем, что все корректно и сохраняем.

У вас есть непримененные настройки, желаете применить? →



Установка RSD на этом закончена.

4. Дополнительные приложения

Вы можете установить дополнительные пакеты приложений для расширения функциональности платформы RSD. Это можно сделать с помощью менеджера пакетов на странице администратора, который загружает приложения из онлайн-репозитория и управляет зависимостями пакетов. Также возможна установка пакетов из локальных файлов.

1. FAQ (Часто задаваемые вопросы)

FAQ является компонентом Базы Знаний. Он поддерживает редактирование и просмотр статей FAQ. Статьи могут быть просмотрены и ограничены агентами, пользователями-клиентами или анонимными пользователями. Их также можно структурировать по группам и читать на разных языках.

3. Первые шаги

Цель этой главы — дать краткий обзор на RSD и структуры его веб-интерфейса. Мы рассмотрим термины «агенты», «клиенты» и «администраторы», а также войдем в систему как администратор RSD и более подробно рассмотрим пользовательские настройки, доступные для каждой учетной записи.

1. Веб-интерфейс агента

Примечание

Прежде чем впервые войти в систему, рассмотрите возможность активации HTTPS на своем веб-сервере, чтобы RSD служил безопасным приложением по протоколу SSL/TLS. Подробные инструкции о том, как это сделать, см. в документации вашего веб-сервера (например, Apache2).

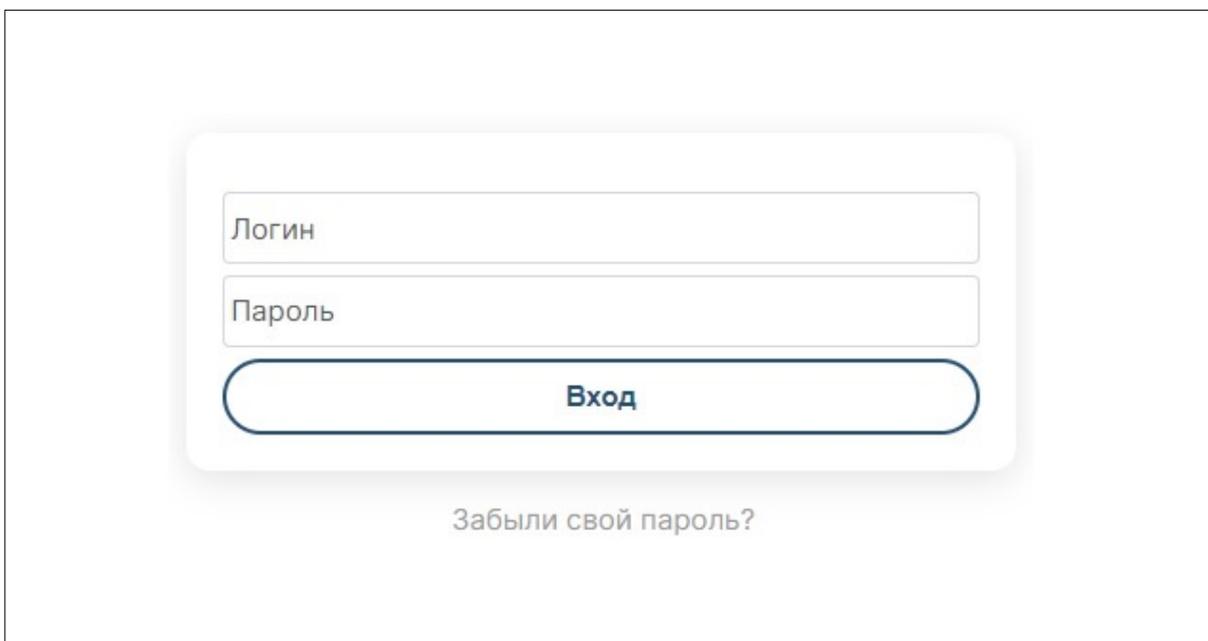
После активации HTTPS установите для параметра конфигурации `HttpType` значение `https` в `SysConfig`. Это гарантирует, что все внутренние ссылки в RSD действительно используют HTTPS.

Хорошей практикой является также перенаправление всего HTTP-трафика на HTTPS на случай, если кто-то попытается получить доступ к RSD по небезопасной ссылке. Это должно быть сделано на уровне конфигурации веб-сервера для максимального эффекта. Однако, если это невозможно, вы можете установить для `HTTPSForceRedirect` значение «Да» в `SysConfig`, чтобы сделать это на уровне приложения.

Веб-интерфейс агента позволяет агентам отвечать на запросы клиентов, создавать новые заявки для клиентов или других агентов, писать заявки о телефонных звонках с клиентами, писать ответы на часто задаваемые вопросы, редактировать данные клиентов и т. д.

Предположим, что ваш хост RSD доступен по URL-адресу <http://www.example.com>, тогда на экран входа в RSD можно попасть, используя адрес <http://www.example.com/rsd/index.pl> в веб-браузере. (см. рисунок ниже).

Рисунок 3.1. Экран входа в интерфейс агента

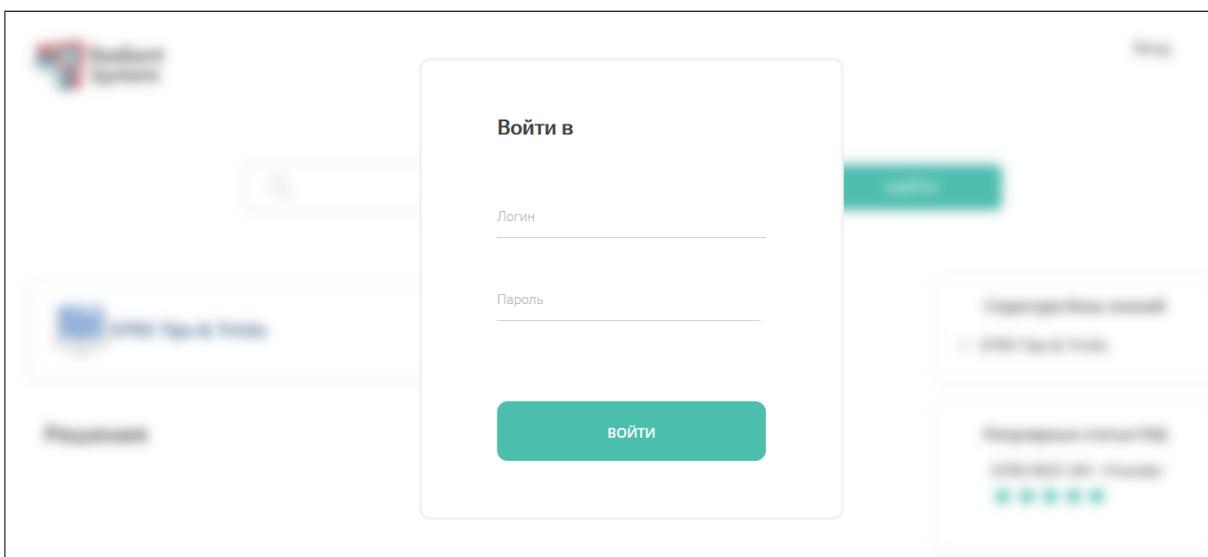


2. Веб-интерфейс клиента

У клиентов есть отдельный веб-интерфейс в RSD, через который они могут создавать новые учетные записи, изменять настройки своей учетной записи, создавать и редактировать заявки, получать обзор созданных ими заявок и т. д.

Продолжая приведенный выше пример, к экрану входа клиента можно перейти, используя URL-адрес <http://www.example.com/rsd/customer.pl> в веб-браузере (см. рисунок ниже).

Рисунок 3.2. Экран входа в клиентский интерфейс



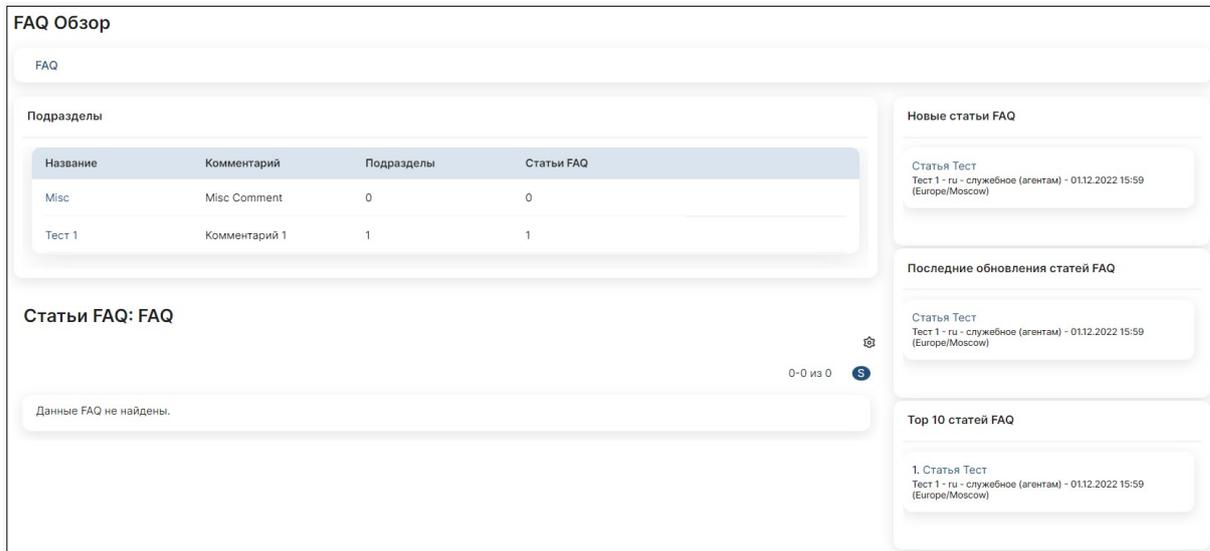
3. Общедоступный веб-интерфейс

В дополнение к веб-интерфейсам для агентов и клиентов, RSD также имеет общедоступный веб-интерфейс, доступный через модуль часто задаваемых вопросов. Этот модуль необходимо установить отдельно. Он обеспечивает публичный доступ к

системе часто задаваемых вопросов и позволяет посетителям выполнять поиск по записям часто задаваемых вопросов без каких-либо специальных разрешений.

В нашем примере доступ к общедоступному веб-интерфейсу можно получить по любому из следующих URL-адресов:
<http://www.example.com/rsd/index.pl?Action=AgentFAQExplorer>.

Рисунок 3.3. Общедоступный веб-интерфейс



4. Первая авторизация

Откройте экран входа в систему, как описано в разделе Веб-интерфейс Агента. Введите имя пользователя и пароль. Поскольку система только что была установлена, а пользователи еще не созданы, сначала войдите в систему как администратор RSD, используя «root@localhost» в качестве имени пользователя и сгенерированный пароль на последнем этапе установки.

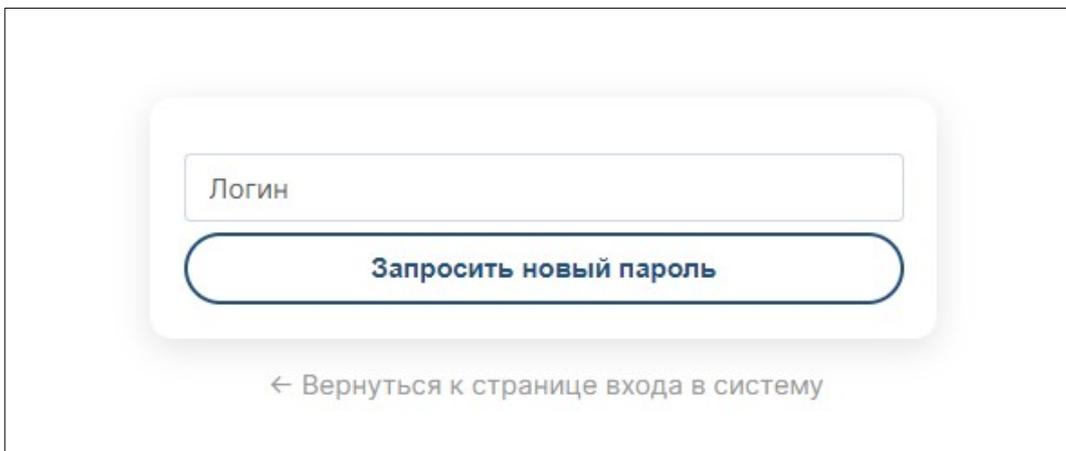
Предупреждение

Эти данные учетной записи действительны для каждой вновь установленной системы RSD. Вам следует как можно скорее сменить пароль администратора RSD! Это можно сделать на экране настроек учетной записи администратора RSD.

Если вы не хотите входить в систему как администратор RSD, просто введите имя пользователя и пароль для своей обычной учетной записи агента.

Если вы забыли свой пароль, вы можете запросить у системы новый пароль. Просто нажмите на ссылку под кнопкой «Войти», введите имя пользователя, зарегистрированное для вашей учетной записи RSD, в поле ввода и нажмите кнопку «Запросить новый пароль» (см. рисунок).

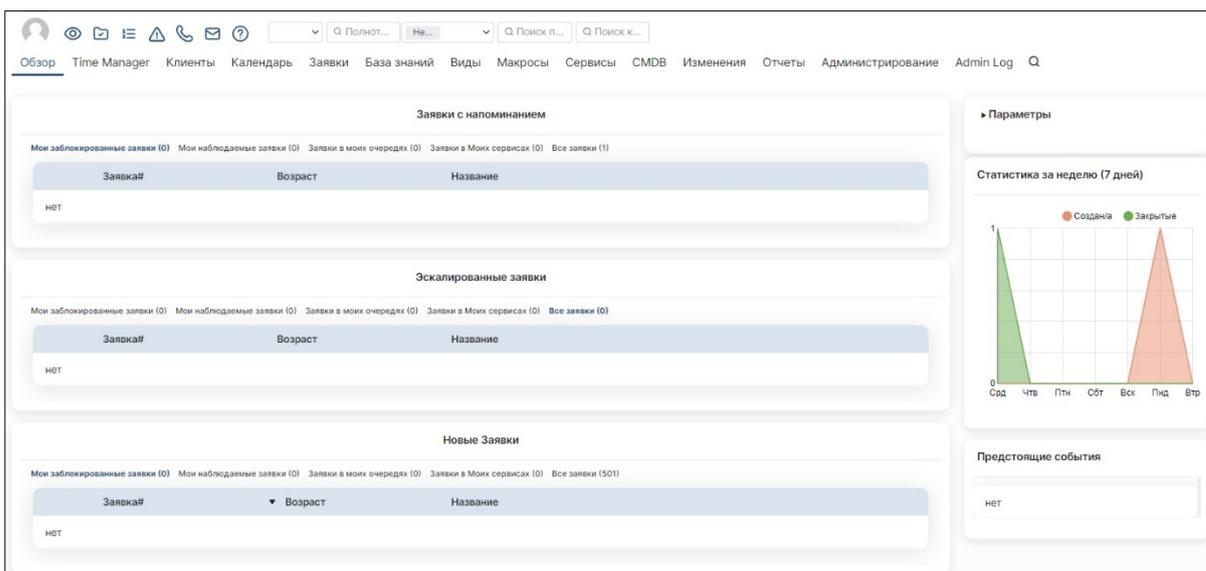
Рисунок 3.4. Запрос нового пароля



5. Веб-интерфейс - обзор

После успешного входа в систему вам будет представлена страница Обзор (см. рисунок ниже). Он показывает ваши заблокированные заявки, обеспечивает прямой доступ через меню к просмотру очереди, статуса и эскалации, а также содержит параметры для создания новых заявок по телефону и электронной почте. Он также представляет краткую сводку заявок с использованием различных критериев.

Рисунок 3.5. Обзор в интерфейсе агента



Для большей наглядности общий веб-интерфейс разделен на разные области. В верхнем ряду каждой страницы отображается некоторая общая информация, такая как кнопка выхода, значки с указанием количества заблокированных заявок и прямым доступом к ним, ссылки для создания новой заявки по телефону/электронной почте и т. д. Также есть значки для перехода к просмотру очереди, статуса и эскалации.

Под строкой значков находится панель навигации. Он показывает меню, которое позволяет вам переходить к различным областям или модулям системы, позволяя вам выполнять некоторые глобальные действия. Нажав кнопку «Обзор», вы попадете на панель инструментов. Если вы нажмете кнопку «Заявки», вы получите подменю с опциями для изменения вида заявки, создания новой заявки (телефон/электронная почта) или поиска конкретной заявки. Кнопка «Отчеты» представляет собой меню,

которое позволяет вам выбрать обзор зарегистрированной статистики, создать новую или импортировать существующую. Кнопка «Клиенты» ведет к экрану «Управление клиентами». Нажав кнопку «Администрирование», вы можете получить доступ ко всем модулям администратора, что позволяет создавать новых агентов, очереди и т. д. Также имеется кнопка «Поиск» для поиска заявок.

Если также установлены какие-либо дополнительные приложения, например, FAQ или Опросы, кнопки для доступа к этим приложениям также отображаются.

В области под панелью навигации могут отображаться различные системные сообщения. Если вы вошли в систему как пользователь-администратор RSD по умолчанию, вы получите красное сообщение, предупреждающее вас о том, что вы не должны работать с этой системной учетной записью.

Под заголовком раздела, в котором вы сейчас находитесь, есть несколько подразделов, содержащих соответствующую информацию об экране, над которым вы работаете, каждый из которых находится в отдельном поле.

Эти поля содержат основную часть каждого экрана, обычно они отображаются в одном или нескольких столбцах, каждое поле может хранить соответствующую информацию о текущем экране, например, инструкции, советы, обзоры и т. д. Также отображается необходимая форма или инструмент для выполнения определенного действия, например, добавления, обновления или удаления записей, проверки журнала системы, изменения параметров конфигурации и т. д.

Наконец, внизу страницы отображается нижний колонтитул сайта (см. рисунок ниже). Он содержит ссылку для прямого доступа к официальному веб-сайту RSD или для переключения в мобильный режим.

Обычно ряд значков, меню и нижний колонтитул одинаковы на всех экранах веб-интерфейса.

Рисунок 3.6. Нижний колонтитул



6. Панель Управления/Обзор

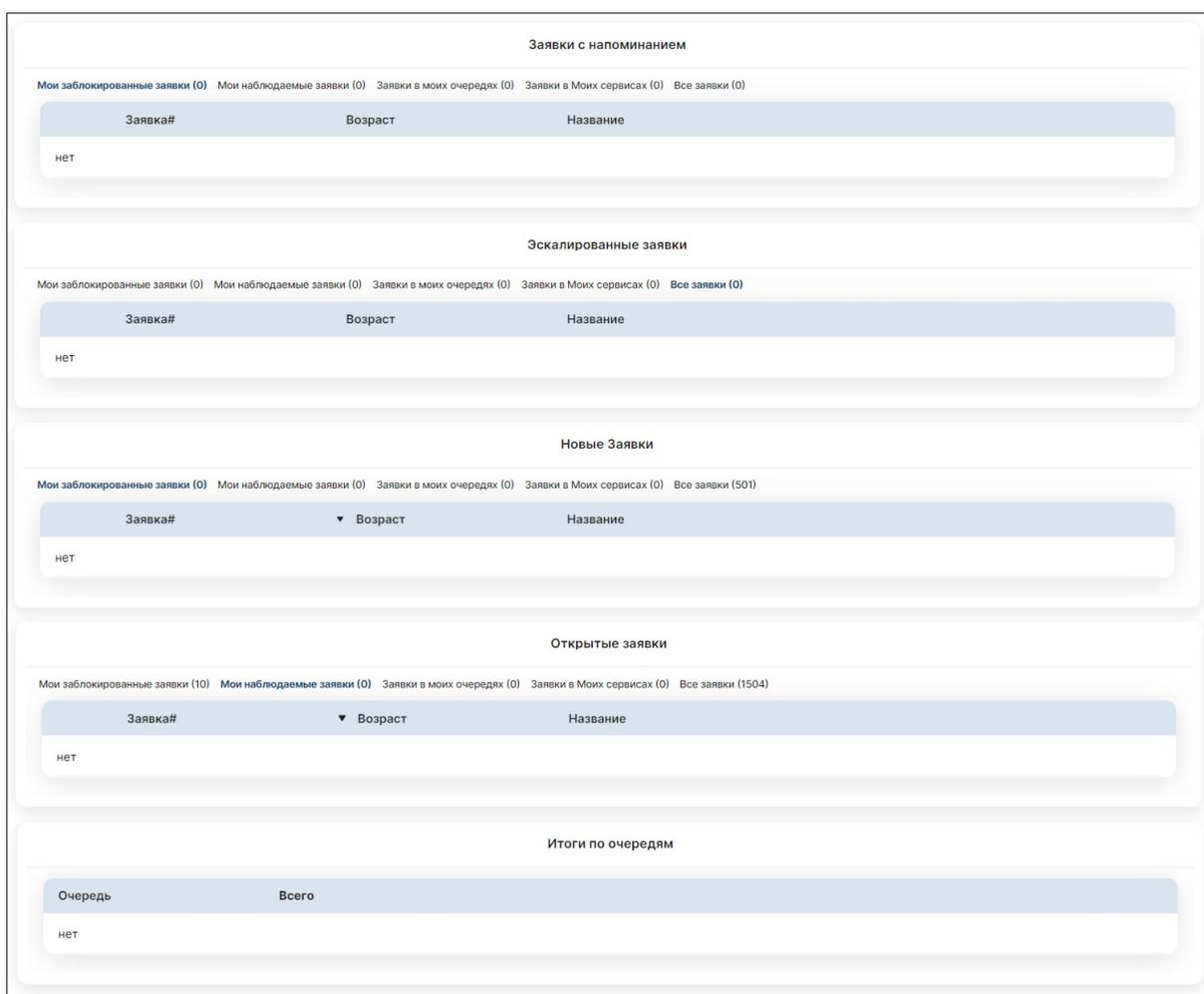
Экран Обзора— это главная страница системы, здесь вы можете получить обзор заявок и другие сведения, связанные с активностью заявок. Предполагается, что это отправная точка для повседневной работы агента, по умолчанию он представляет краткую сводку заявок, которые отложены, эскалированы, новые и открытые, и другую, полезную информация.

Одной из наиболее важных особенностей «Обзора» является то, что он полностью настраивается. Это означает, что вы можете настроить каждую часть по своему усмотрению, отображая или скрывая элементы. Можно даже переместить эти

элементы в пределах одного столбца, щелкнув и перетащив заголовок элемента/таблицы в другое место. Каждый элемент называется «Виджет», в системе есть несколько виджетов, готовых к использованию «из коробки», но модульная конструкция экрана приборной панели позволяет легко интегрировать пользовательские виджеты.

Содержимое панели управления расположено в двух основных столбцах, в левом столбце обычно отображается информация о заявках, классифицированных по их состояниям, например: напоминание, эскалация, новая и открытая. В каждом из этих виджетов вы можете отфильтровать результаты, чтобы увидеть все заявки, к которым вам разрешен доступ, заявки, которые вы заблокировали, те, которые расположены в очередях, определенных агентом, среди других фильтров. В этой колонке есть и другие виды виджетов, и все они описаны ниже.

Рисунок 3.7. Виджеты Панели Управления



Виджеты левой стороны Обзора.

- Виджеты списков заявок

Виджеты в этой категории имеют одинаковое общее поведение, внешний вид. Этот виджет показывает список заявок в определенном состоянии. Количество заявок, отображаемых на каждой странице списка, можно настроить в настройках виджета (они

появляются при наведении указателя мыши на верхнюю правую часть виджета). Эти виджеты поддерживают следующие фильтры:

- Мои заблокированные заявки

Список заблокированных заявок агента.

- Мои наблюдаемые заявки

Заявки агента, которые находятся в списке наблюдаемых. Параметр Ticket::Watcher должен быть включен для их отображения.

- Заявки, где я ответственный

Заявки, для которых агент является ответственным. Параметр Ticket::Responsible должен быть включен, чтобы этот фильтр работал.

- Заявки в моих очередях

Заявки, находящиеся в очередях, которые агент определяет как «Мои очереди».

- Заявки в моих сервисах

Заявки, назначенные сервисами, которые агент определяет как «Мои сервисы» и которые находятся в очередях с разрешениями как минимум на чтение.

- Все заявки

Все заявки, к которым у агента есть доступ. Это виджеты:

- Заявки с напоминанием

Отложенные заявки, у которых назначенная дата напоминания наступила.

- Эскалированные заявки

Заявки, которые эскалированы.

- Новые Заявки

Заявки, имеющие состояние "Новая/New".

- Открытые Заявки / Ожидают ответа

Заявки в состоянии "Открыта" и готовые для работы с ними.

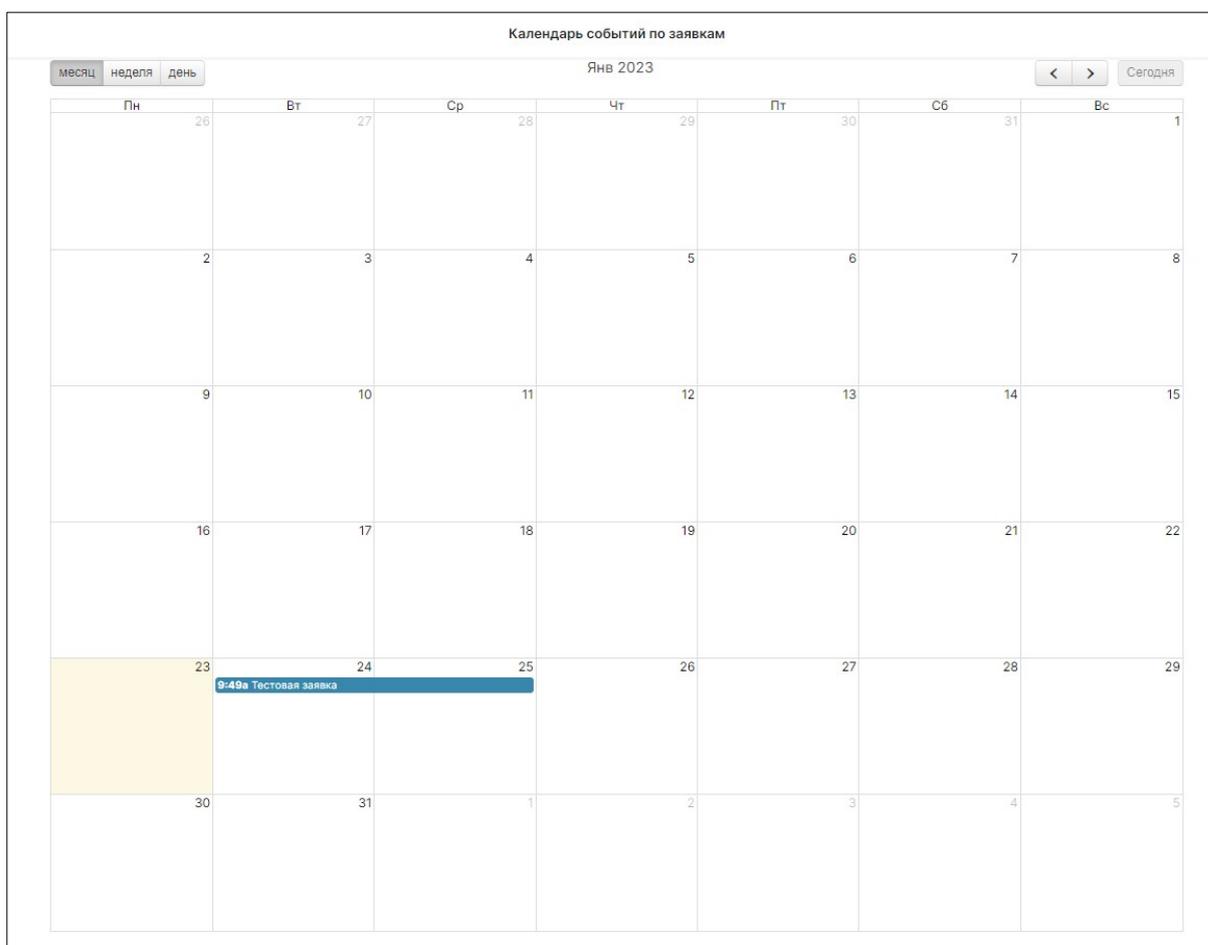
- Календарь событий по заявкам

Событие календаря (для этого виджета) определяется при создании новой заявки, функция Календарь событий для заявок должна быть включена и требуется наличие двух новых полей на экране создания заявки, одно для времени начала события, а другое для времени окончания. Эти значения определяют длительность события.

Этот виджет включает в себя следующие представления: месяц, неделя и день. Агенты могут прокручивать страницы, используя стрелки вправо и влево.

Как упоминалось ранее, просто включить виджет недостаточно, необходимо добавить пару динамических полей типа «Дата/Время» для заявок в систему (через ссылку «Динамические поля» в панели «Администрирование») и настроить их в SysConfig для этого виджета. Оба динамических поля должны быть настроены для отображения на экранах создания заявки, они должны быть заполнены во время создания заявки или на любом другом экране действия заявки (например, свободные поля) для описания временных рамок события календаря (время начала и окончания). Можно задать отображение этих полей на экране просмотра заявки/TicketZoom для контроля, если сочтете это необходимым.

Рисунок 3.8. Виджет "Календарь событий по заявкам"



Дальнейшие настройки этого виджета можно найти в подгруппе Frontend::Agent::Dashboard::EventsTicketCalendar в SysConfig:

- Ширина календаря

Определяет ширину календаря в процентах. По умолчанию 95%.

- DynamicFieldStartTime

Определяет имя динамического поля для времени начала.

- `DynamicFieldEndTime`

Определяет имя динамического поля для времени окончания.

- Очереди

Только заявки в очередях, указанных в этой настройке, будут учитываться в календаре.

- `DynamicFieldForEvents`

Задаёт динамические поля, отображаемые во всплывающем окне календаря при наведении мыши на заявку.

- `TicketFieldsForEvents`

Задаёт атрибуты заявки, отображаемые во всплывающем окне календаря при наведении мыши на заявку.

- Итоги по очередям

Этот виджет отображает матрицу подсчета заявок, где строки представляют очереди, а столбцы — состояния заявок, а затем в каждой ячейке отображается количество заявок в определенном состоянии, принадлежащем конкретной очереди.

Виджет также отображает строку «Итоги» и столбец «Итоги». В строке «Итоги» показана сумма заявок для каждого состояния во всех представленных очередях, а в столбце «Итоги» представлена сумма заявок для каждой очереди во всех представленных состояниях.

Представленные очереди и состояния можно изменить с помощью Sysconfig.

При клике по любому номеру счетчика заявок откроется страница результатов поиска заявок, на которой агенты смогут получить более подробный обзор.

В правой колонке расположен специальный виджет настроек, который позволяет вам управлять виджетами, которые вы хотите показать или скрыть. Нажмите на его заголовок, чтобы развернуть раздел и увидеть все доступные виджеты, как показано на рисунке. Каждое имя виджета имеет флажок. Используйте эти флажки, чтобы определить видимость виджетов на панели инструментов (неотмеченные виджеты не будут отображаться) после того, как вы определите параметры видимости и нажмете «Сохранить настройки» для сохранения изменений. Этот раздел фиксируется на экране, это означает, что вы не можете перетащить его или закрыть.

Рисунок 3.9. Параметры Панели Управления

▼ **Параметры**

- Новости о продукте
- Заявки с напоминанием
- Эскалированные заявки
- Новые Заявки
- Открытые заявки
- Запущенные Процессные заявки
- Статистика за неделю (7 дней)
- Предстоящие события
- Итоги по очередям
- Календарь событий по заявкам
- Вне офиса
- Последние обновления статей FAQ
- Новые статьи FAQ
- Logged-In Users
- Новости
- Мероприятия

Сохранить настройки

Виджеты панели инструментов в правой колонке.

- Статистика за неделю (7 дней)

На нем показан график активности заявок за последние 7 дней, включающий 2 линии. Один, обычно синего цвета, представляет количество созданных заявок в день, а второй, обычно оранжевый, представляет количество закрытых заявок в день.

- **Предстоящие События**

Здесь перечисляются заявки подлежащие эскалации или уже эскалированные. Информация этого виджета очень полезна, чтобы решить, какие заявки требуют вашего внимания, установить приоритеты или просто проверить, что происходит.

- **Новости RSD**

Полный список действий RSD и важная информация о новых выпусках продуктов или исправлениях.

- **Онлайн**

Здесь показана сводка о текущих агентах, вошедших в систему, она также включает раздел для онлайн-клиентов. Пожалуйста, обратите внимание, что этот виджет обычно скрыт, его можно отобразить с помощью виджета «Настройки», описанного ранее.

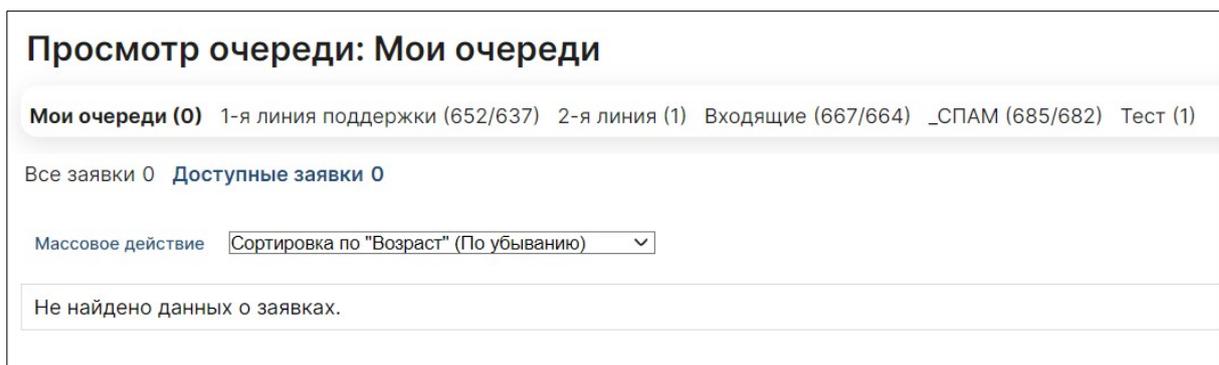
7. Что такое очередь?

Очередь - объект системы, предназначенный для разграничения доступа к заявкам. Обычно очередь соответствует функциональной группе исполнителей. Это может быть, например, 1-я линия поддержки, группа поддержки серверов или программного обеспечения. Если планируется использовать RSD для глобальной поддержки пользователей по всем вопросам, то возможно создавать Очереди для отделов и служб АХО, бухгалтерии, кадровых специалистов, отдела продаж.

8. Что такое просмотр очередей?

Просмотр очередей дает возможность видеть все очереди, в которых есть заявки, и в которых агент имеет права RW.

Рисунок 3.10. Обзор очередей (по умолчанию) для агентов



Просмотр очереди предлагает различные варианты ежедневной работы с RSD. Первый из них — «Мои очереди». В настройках агента или при администрировании агентов можно определить набор очередей, для работы с которыми назначен агент. Все заявки будут отображаться в этом разделе по умолчанию при переходе через меню «Заявки» -> «Просмотр очереди».

Второй вариант, предлагаемый просмотром очереди, — это переход к отдельным очередям и дочерям, содержащим заявки для обработки.

В обоих типах просмотра у пользователя также есть дополнительная возможность просмотреть либо все разблокированные заявки (это фильтр по умолчанию), либо пользователь может выбрать просмотр всех доступных заявок. Заявки должны находиться в одном из доступных для просмотра типов состояния, чтобы они отображались в просмотре очереди. По умолчанию это открытые, новые, ожидающие напоминания, ожидающие автозакрытия.

Для помощи пользователю имеются визуальные оповещения.

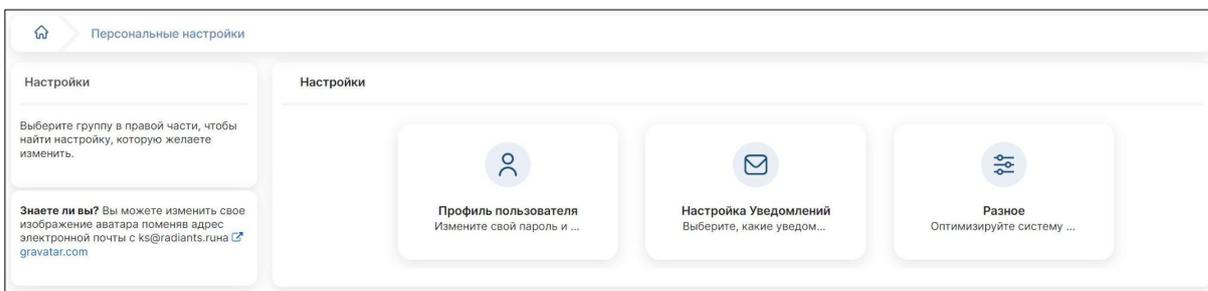
Рисунок 3.11. Просмотр визуальных оповещений в очереди агентов

- Highlight Age 1: устанавливает возраст в минутах (первый уровень) для выделения очередей, содержащих нетронутые заявки. Видно, выше в очереди "Raw".
- Highlight Age 2: устанавливает возраст в минутах (второй уровень) для выделения очередей, содержащих нетронутые заявки. Видно, выше в очереди "Postmaster".
- Blink: активирует механизм мигания очереди, содержащей самую старую заявку. Поддерживается не во всех браузерах. В этом случае он будет красным, как показано выше в очереди «Нежелательная почта».
- Жирный: текущая очередь будет выделена полужирным шрифтом, как показано выше в разделе «Мои очереди».

9. Пользовательские настройки

Пользователи RSD, такие как клиенты, агенты и администратор RSD, могут настраивать параметры своей учетной записи в соответствии со своими потребностями. Агент может получить доступ к экрану конфигурации, щелкнув значок шестеренки в верхней части веб-интерфейса (см. рисунок ниже), а клиенты должны щелкнуть ссылку «Настройки» (см. рисунок ниже).

Рисунок 3.12. Персональные настройки агента



Агент может настроить 3 различные категории предпочтений: профиль пользователя, настройки уведомлений и другие настройки. Возможные варианты по умолчанию:

Профиль пользователя

- Изменение текущего пароля.
- Изменение изображения своего аватара.
- Выбор языка интерфейса.
- Выбор личного часового пояса.
- Изменение темы веб-интерфейса.
- Активация и настройка времени отсутствия на рабочем месте.

Настройка Уведомлений

- Выбор очереди, которые вы хотите отслеживать, в меню «Мои очереди».
- Выбор сервисов, которые вы хотите отслеживать, в разделе «Мои сервисы».
- Настройка уведомлений о заявках, которые вы хотите получать (для каждого метода транспортировки).
- Выбор типа изменений мероприятий, по которому вы желаете получать уведомления.

Разное

- Переключение между темами интерфейса.
- Выбор периода обновления для обзоров (панель инструментов, представление очереди и т. д.).
- Выбор экрана, который будет отображаться после создания заявки.
- Выбор экрана, который будет отображаться после создания задачи.
- Выбор лимитов на страницу для различных обзорных экранов (заявка, динамические поля и журнал связи).

Рисунок 3.13. Персональные настройки клиента

Параметры



Язык интерфейса

Time zone

Интервал обновления

Заявок на страницу

Выбранная цветовая схема

Голубой Старый кирпич
 Терракот Нефрит Гранатовый

Название

Имя

Фамилия

Логин

Пароль

Подтвердите пароль:

Email

ID компании

Загрузить фото Файл не выбран

Настройка отображения заявок

<input type="checkbox"/> Кем изменен	<input type="checkbox"/> Число
<input type="checkbox"/> FieldData	<input type="checkbox"/> Тест
<input type="checkbox"/> Эскалация - время первого ответа	<input type="checkbox"/> Название
<input type="checkbox"/> Эскалация - время обновления	<input type="checkbox"/> ITSMDecisionDate
<input type="checkbox"/> Блокировать	<input type="checkbox"/> Возраст
<input type="checkbox"/> Владелец	<input type="checkbox"/> Изменен
<input type="checkbox"/> Приоритет	<input type="checkbox"/> Дата создания
<input type="checkbox"/> Очередь	<input type="checkbox"/> Клиент
<input type="checkbox"/> Ответственный	<input type="checkbox"/> Эскалация - время решения
<input type="checkbox"/> SLA	<input type="checkbox"/>
<input type="checkbox"/> Сервис	
<input type="checkbox"/> Состояние	
<input type="checkbox"/> ID заявки	
<input type="checkbox"/> Тип	

Клиент может выбрать язык веб-интерфейса, установить интервал обновления для обзора заявок и выбрать максимальное количество показанных заявок. Также можно установить новый пароль.

4. Администрирование

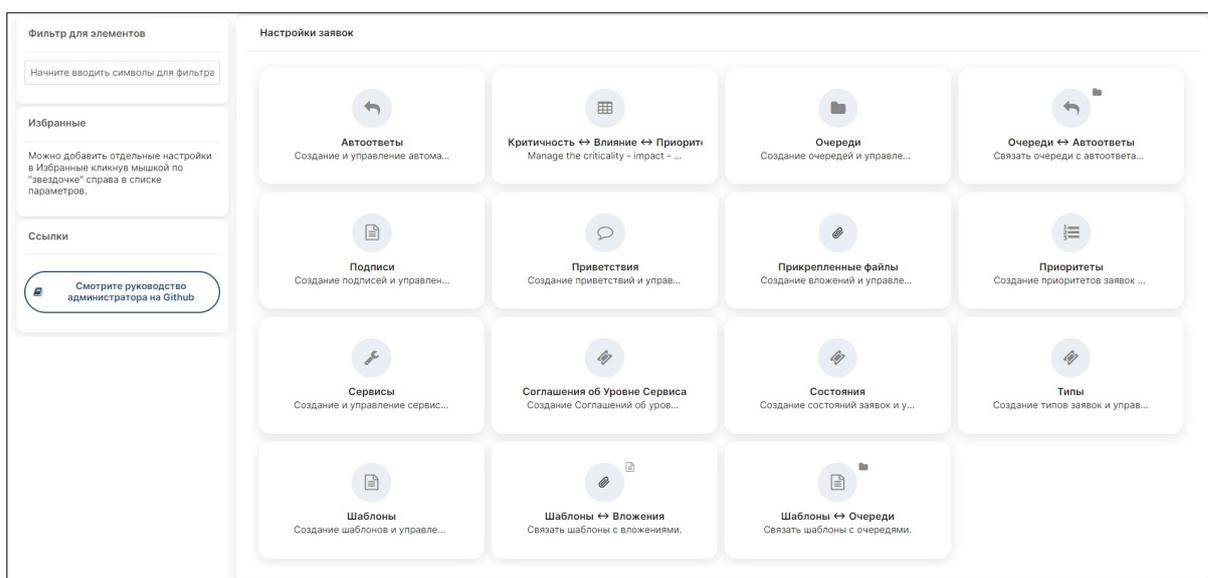
1. Панель администрирования RSD

1.1. Основы

Администраторы используют страницу администрирования веб-интерфейса RSD для настройки системы - добавления агентов, клиентов и очередей, заявок, настройки почты, установки дополнительных пакетов, таких как FAQ и ITSM и многое другое.

Агенты, входящие в группу *admin* могут получить доступ к панели администрирования, нажав на кнопку *Администрирование/Админ* на панели навигации (см. рисунок ниже). Остальные агенты не будут видеть эту ссылку.

Рисунок 4.1. Экран панели администрирования RSD



1.2. Агенты, Группы и Роли

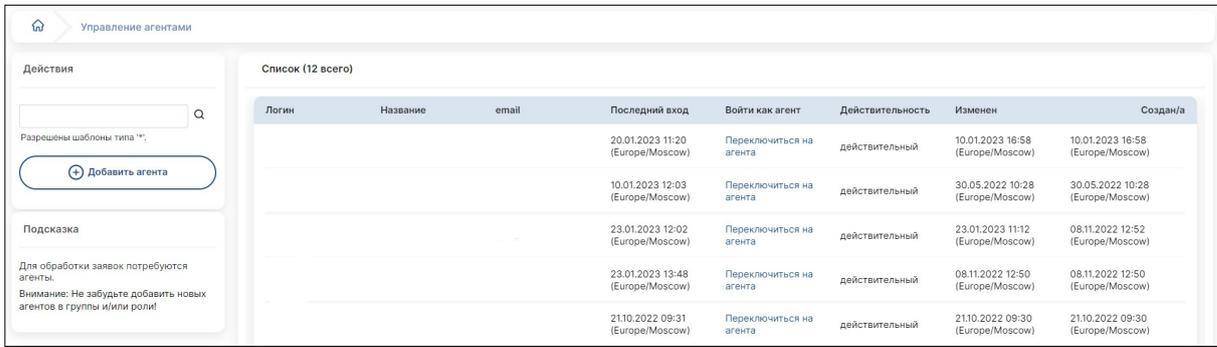
Агенты

Нажав на ссылку «*Агенты*», вы получите доступ к странице управления агентами RSD (см. рисунок ниже). Администраторы могут добавлять, изменять или отключать аккаунты агентов. Администраторы также могут управлять личными настройками агентов, например, изменять язык их веб-интерфейса, настройки уведомлений.

Примечание

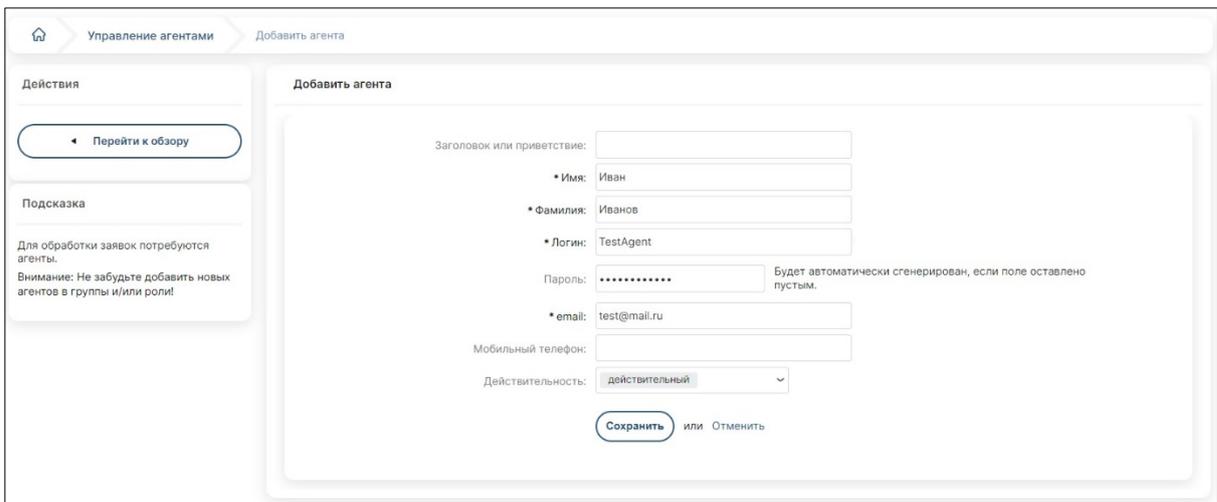
Учетная запись (аккаунт) агента может быть отключена, но не удалена. Отключение осуществляется путем установки значения *недействительный* или *временно недействительный* в параметре «*Действительность*».

Рисунок 4.2. Управление агентами



Чтобы зарегистрировать агента, нажмите кнопку «Добавить агента», введите необходимые данные и нажмите кнопку «Сохранить» в нижней части экрана, как показано на рисунке.

Рисунок 4.3. Добавление нового агента



После создания новой учетной записи агента вы должны сделать агента членом одной или нескольких групп, или ролей. Информация о группах и ролях доступна в разделах Группы и Роли этой главы.

Группы

Каждая учетная запись агента должна принадлежать по крайней мере к одной группе или роли. После установки системы есть три заранее заданные группы, которые приведены в Таблице 4-1.

Таблица 4.1. Предопределенные группы, доступны для новой (только что) установленной системы RSD

Группа	Описание
admin	Допускается для выполнения административных задач в системе.

stats	Квалифицирован для доступа к модулю отчетов RSD и создания отчетов.
users	Агенты должны входить в эту группу с правами для чтения и записи. Тогда они смогут получить доступ ко всем функциям системы обработки заявок.

Примечание

В совершенно новой установке RSD агент «root@localhost» по умолчанию принадлежит к группам admin, stats, users.

Чтобы перейти на страницу управления группами, нажмите в панели администрирования ссылку *Группы*.

Рисунок 4.4. Управление Группами

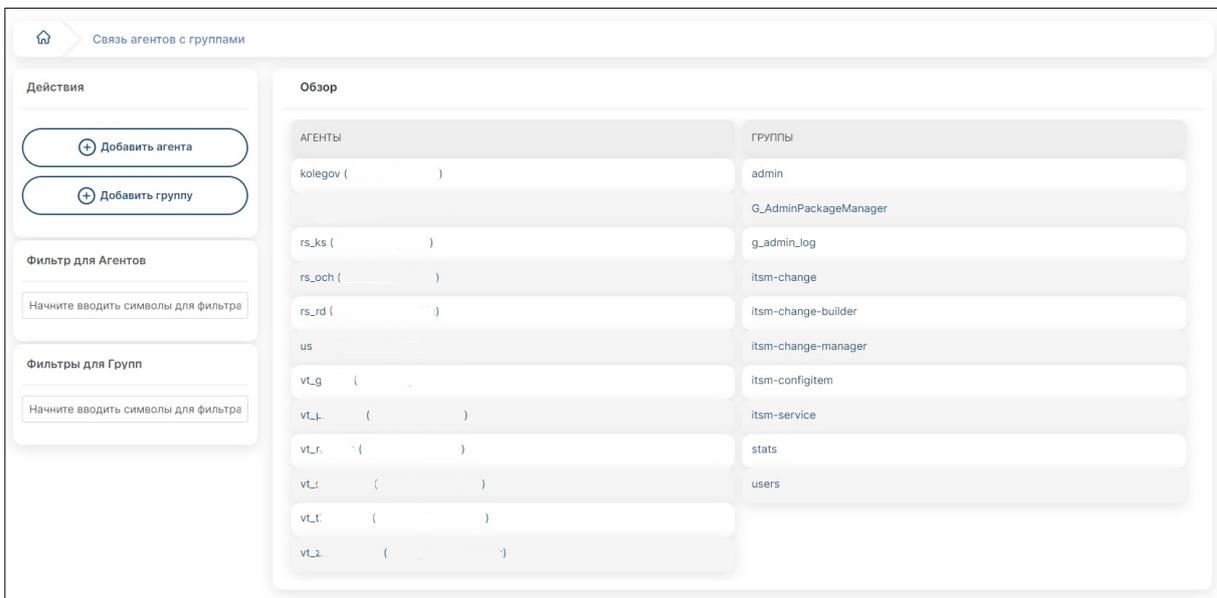
Название	Комментарий	Действительность	Изменен	Создан/а
G_AdminPackageManager		действительный	12.01.2023 13:16 (Europe/Moscow)	12.01.2023 13:16 (Europe/Moscow)
admin	Group of all administrators.	действительный	30.05.2022 10:28 (Europe/Moscow)	30.05.2022 10:28 (Europe/Moscow)
g_admin_jog	Group for viewing AdminUserLog	действительный	30.05.2022 10:37 (Europe/Moscow)	30.05.2022 10:37 (Europe/Moscow)
itsm-change	Group for ITSM Change mask access in the agent interface.	действительный	30.05.2022 10:30 (Europe/Moscow)	30.05.2022 10:30 (Europe/Moscow)
itsm-change-builder	Group for ITSM Change Builders.	действительный	30.05.2022 10:30 (Europe/Moscow)	30.05.2022 10:30 (Europe/Moscow)
itsm-change-manager	Group for ITSM Change Managers.	действительный	30.05.2022 10:30 (Europe/Moscow)	30.05.2022 10:30 (Europe/Moscow)
itsm-configitem	Group for ITSM Configitem mask access in the agent interface.	действительный	30.05.2022 10:30 (Europe/Moscow)	30.05.2022 10:30 (Europe/Moscow)
itsm-service	Group for ITSM Service mask access in the agent interface.	действительный	30.05.2022 10:30 (Europe/Moscow)	30.05.2022 10:30 (Europe/Moscow)
stats	Group for statistics access.	действительный	30.05.2022 10:28 (Europe/Moscow)	30.05.2022 10:28 (Europe/Moscow)
users	Group for default access.	действительный	30.05.2022 10:28 (Europe/Moscow)	30.05.2022 10:28 (Europe/Moscow)

Примечание

Как в случае с агентами, группа в RSD может быть только отключена, но не удалена. Отключение осуществляется путем установки флага Действительный в значение *недействительный* или *временно недействительный*.

Чтобы добавить агента в группу, или изменить принадлежность агентов к группе, используйте ссылку *Агенты ↔ Группы* на странице Администрирования (см. рисунок ниже).

Рисунок 4.5. Агенты ↔ Группы



На этой странице отображается обзор всех групп и агентов в системе. Вы также можете использовать доступные фильтры для поиска определенного объекта. Если вы хотите изменить группы, в которые входит агент, просто нажмите на имя агента (см. рисунок ниже). Чтобы изменить агентов, связанных с группой, просто нажмите на группу, которую вы хотите изменить (см. рисунок ниже).

Рисунок 4.6. Изменение группы, к которой принадлежит агент.

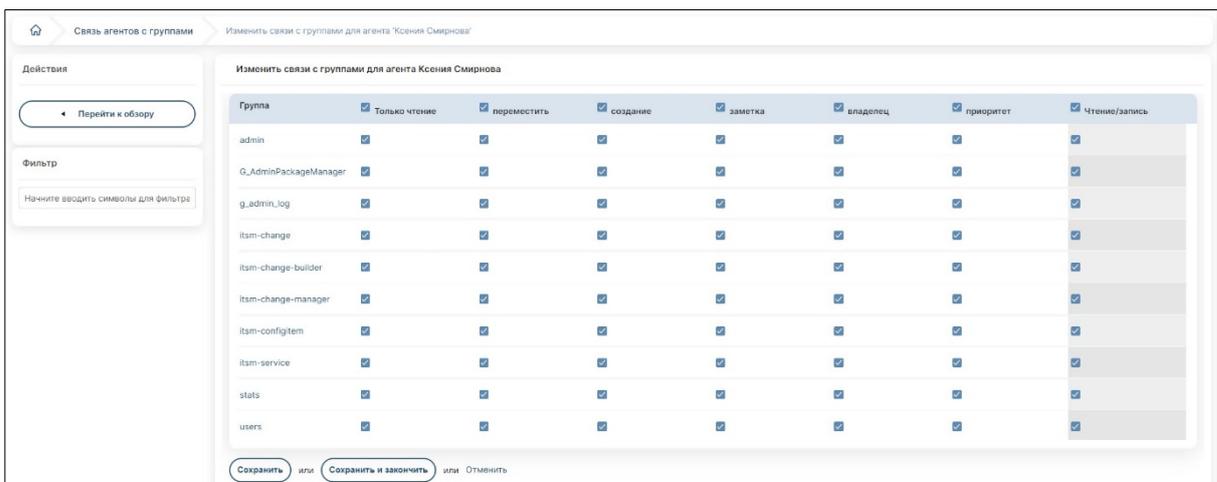
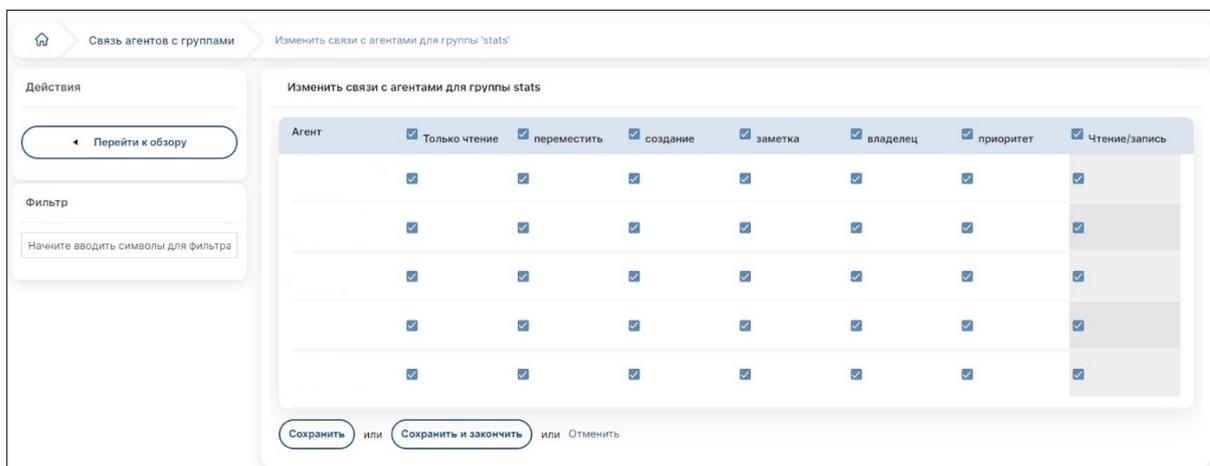


Рисунок 4.7. Изменение агентов, входящих в определенную группу.



Каждая группа имеет набор связанных с ней прав, и каждый член группы (агент) может иметь некоторую комбинацию этих прав для себя. Список разрешений/прав показан в таблице 4-2.

Таблица 4.2. Права, связанные с группами RSD

Права	Описание
ro	Права «только для чтения» заявок, записей и очередей этой группы
move into	Права перемещать заявки или записи между очередями или областями, принадлежащими этой группе.
create	Права для создания заявок или записей в очередях или области группы.
note	Права добавлять заметки к заявкам или записям в очередях или области группы.
owner	Права для изменения владельца заявок или записей в очередях, принадлежащих этой группе.
priority	Права для изменения приоритета заявок или записей в очереди или области, принадлежащей этой группе.

rw	Полный доступ к заявкам или записям в очереди или области, принадлежащей к этой группе.
----	---

Примечание

По умолчанию, Обзор Очередей/ОчередьView отображает только заявки в очередях, в которых агент имеет права *rw*, т.е доступ к заявкам, с которыми он должен работать. Если вы хотите изменить этот порядок, вы можете установить параметр `Ticket::Frontend::AgentTicketQueue###ViewAllPossibleTickets` в значение *Да*.

Не все доступные права отображаются по умолчанию. Эти дополнительные права могут быть добавлены.

Таблица 4.3. Дополнительные группы прав

Права	Описание
stats	Предоставляет доступ к странице отчетов.
bounce	Право перенаправить/bounce почтовое сообщение (кнопка перенаправить/ bounce в TicketZoom).
compose	Право составления ответа на заявку.
customer	Право изменить клиента для этой заявки
forward	Право пересылать сообщения (с помощью кнопки Переслать).
pending	Право перевести заявку в ожидание.
phone	Право добавить содержание звонка клиента к заявке. В виде сообщения/ заметки.

responsible	Право изменить ответственного за заявку.
-------------	--

Примечание

Эти права могут быть добавлены изменением параметра System::Permission

Роли

Роли — это мощная функция, позволяющая очень просто и быстро управлять правами доступа многих агентов. Они особенно полезны для больших и сложных систем поддержки, с большим количеством агентов, групп и очередей. В приведенном ниже примере объясняется, как и когда они могут использоваться.

Давайте представим, что у нас есть система, в которой зарегистрировано 100 агентов, 90 из них имеют доступ только к очереди "Поддержка", в которой хранятся все запросы о поддержке. Очередь "Поддержка" содержит несколько дочерей. Остальные 10 агентов имеют права доступа ко всем очередям, существующим в системе. Эти 10 агентов работают с заявками, просматривают очередь "Raw" и перемещают спам-сообщения в очередь "Junk".

В один день компания открывает новый отдел для продажи товаров. Запрос и принятие заказа, подтверждение, счета и т.д. должны быть обработаны, и некоторые из агентов компании должны делать это используя RSD. Разным агентам нужно предоставить доступ к очередям, которые планируется создать.

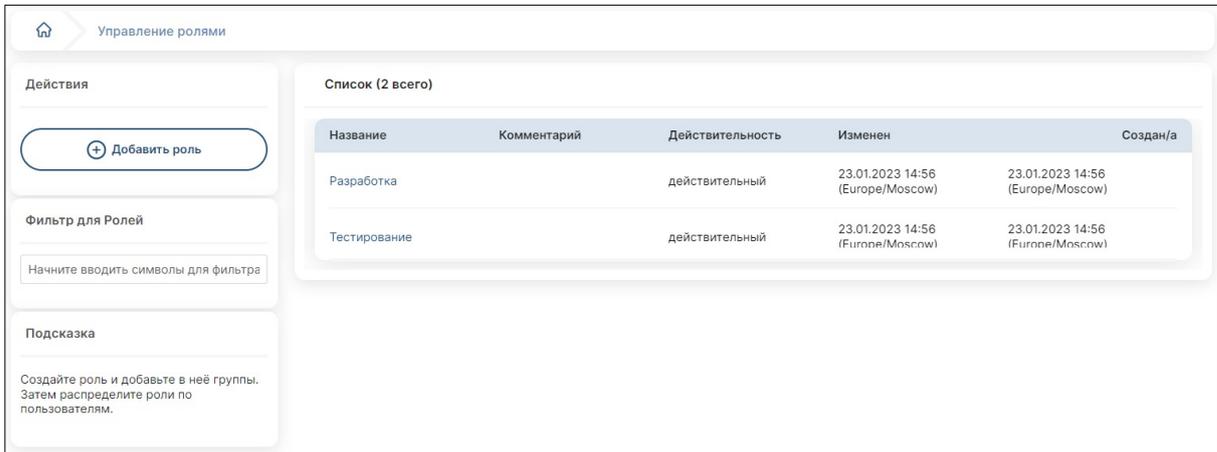
Поскольку изменения вручную прав доступа для разных агентов займет много времени, то для определения различных уровней доступа можно создать роли. Затем, агенты могут быть добавлены к одной или нескольким ролям, таким образом их права будут изменены автоматически. Если будет создана новая учетная запись агента, ее также можно с легкостью добавить к одной или нескольким ролям.

Примечание

Роли действительно полезны при работе со сложными организациями и при обслуживании больших установок RSD. Тем не менее следует быть аккуратным с этим механизмом. Смешивание назначения групп для агента с назначением ролей для агента может создать сложную схему управления доступом, которую трудно будет понять и поддерживать. Если вы хотите использовать только роли и отключить опцию Агенты ↔ Группы в области администрирования, вы можете сделать это, изменив Frontend::Module###AdminUserGroup в SysConfig. Имейте в виду, что это не приведет к удалению уже существующих назначений агентов в группу!

Вы можете получить доступ к разделу управления ролями (см. рисунок ниже), нажав на ссылку «Роли» на странице «Администрирование».

Рисунок 4.8. Управление Ролями

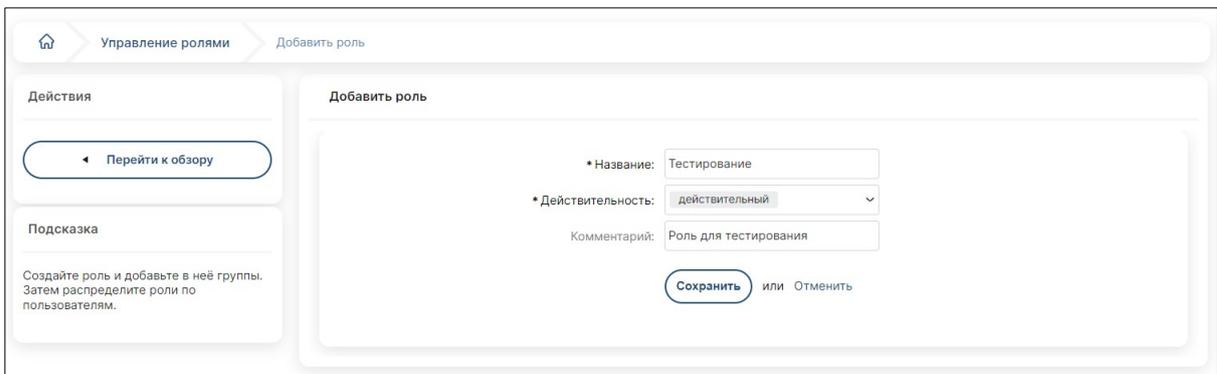


Примечание

Как в случае с агентом и группами, однажды созданные роли могут быть деактивированы, но не удалены. Чтобы отключить, установите опцию Действительность в значение *недействительный* или временно *недействительный*.

Отображается обзор всех ролей в системе. Чтобы изменить настройки роли, щелкните имя роли. В новой установке RSD роли по умолчанию не определены. Чтобы зарегистрировать ее, нажмите кнопку «Добавить роль», укажите необходимые данные и сохраните их (см. рисунок ниже).

Рисунок 4.9. Добавление новой роли



Чтобы получить список всех ролей и агентов в системе, перейдите по ссылке [Агенты ↔ Роли](#) в Панели Администрирования. Также, для поиска конкретного элемента можно использовать фильтры. Если необходимо изменить роли, связанные с определенным агентом, просто кликните на имя агента (см. нижеприведенный Рисунок). Чтобы изменить агентов, связанных с ролью, нажмите на роль, которую нужно отредактировать (см. рисунок ниже).

Рисунок 4.10. Изменение Ролей, связанных с Агентом.

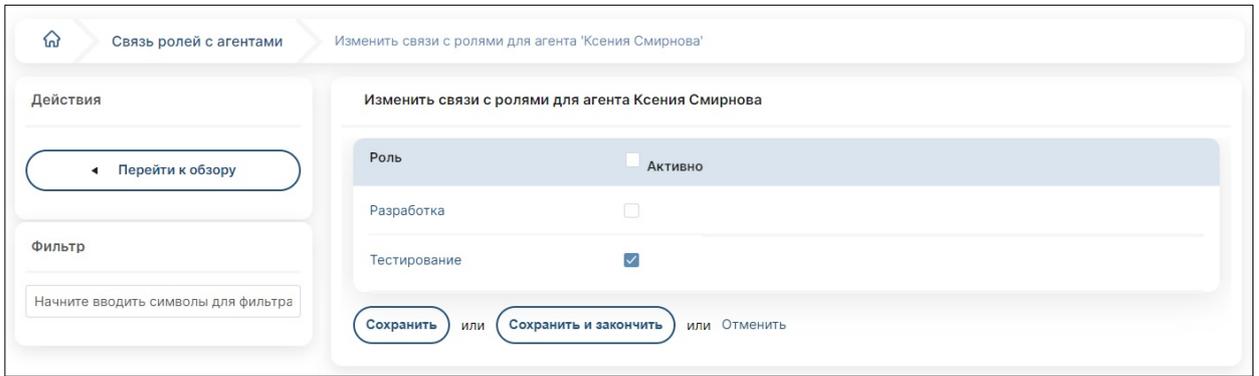
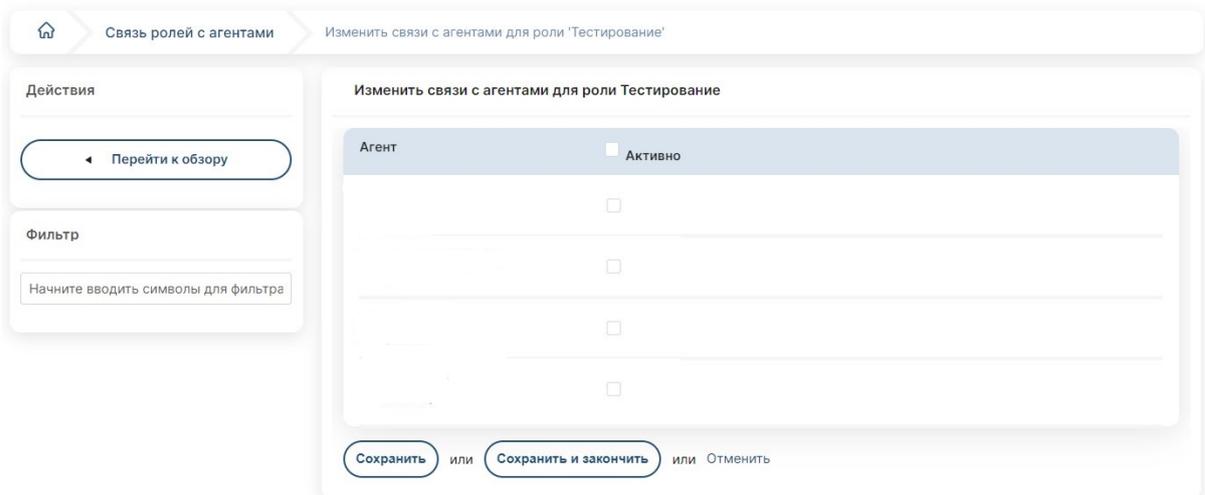
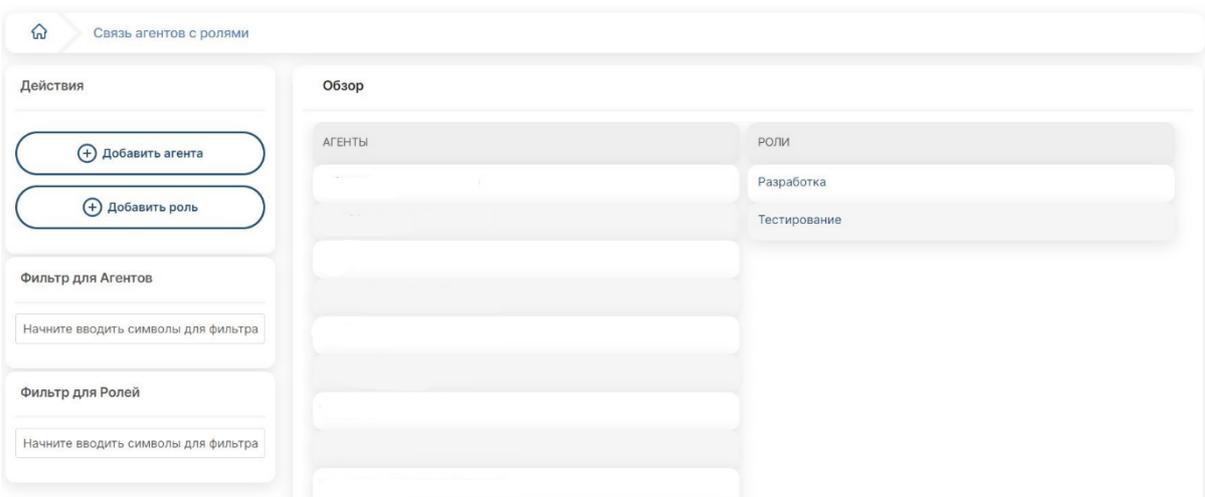


Рисунок 4.11. Изменение Агентов, связанных с конкретной Ролью.



Чтобы получить список всех ролей и групп системы, перейдите по ссылке Роли ↔ Группы на странице Администрирования. На экране появится страница, приведенная ниже на рисунке. Для поиска конкретного элемента также можно использовать фильтры.

Рисунок 4.12. Управление связями Роли - Группы



Чтобы определить различные права доступа для роли, нажмите на имя роли или группы (см. ниже Рисунок 4.13 и 4.14, соответственно).

Рисунок 4.13. Изменить связи группы для роли

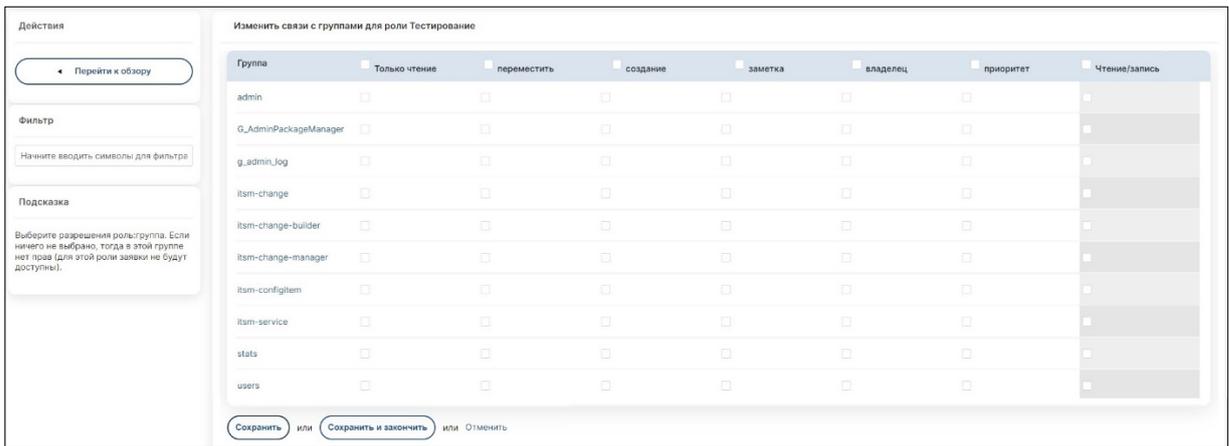
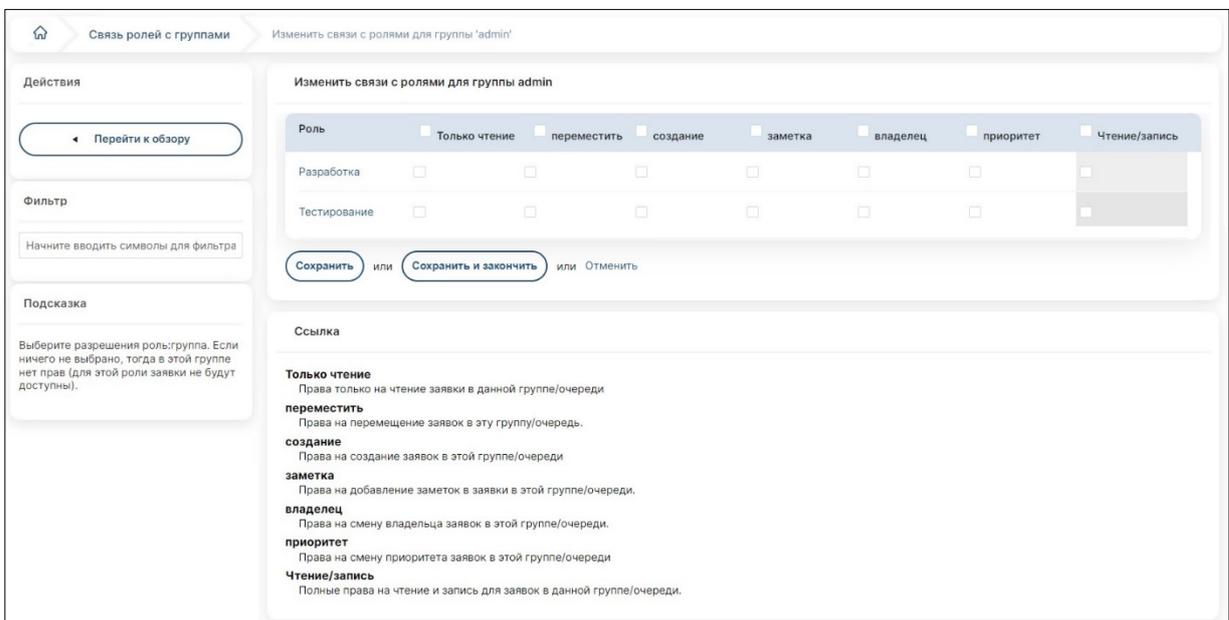


Рисунок 4.14. Изменить связи группы для роли



1.3. Клиенты и Группы Клиентов

1. Клиенты

RSD поддерживает разные типы пользователей. С помощью ссылки «Клиенты» (через панель навигации или страницу администратора) вы можете управлять учетными записями ваших клиентов (см. рис. ниже), которые могут войти в систему через клиентский интерфейс (customer.pl). С помощью этого интерфейса ваши клиенты могут не только создавать заявки, но и просматривать свои прошлые заявки на наличие новых обновлений. Важно знать, что клиент нужен для истории заявок в системе.

Рисунок 4.15. Управления Клиентами

Управление учётными записями клиентов

Действия

Разрешены шаблоны типа **

Добавить учётную запись клиента

Бэкэнд Базы данных

Подсказка

Необходимо для наличия данных о клиенте и подключения к системе через интерфейс клиента.

Список (5 всего)

Логин	Название	email	ID клиента	Последний вход	Действительность
customer_002	Никита Кожемяка	2customer_email@radiants.ru	ООО Ланс Трейд		действительный
customer_010	Добрыня Никитич	10customer_email@radiants.ru	ООО Пота и Копыта	17.01.2023 16:21 (Europe/Moscow)	действительный
customer_011	Алёша Попович	11customer_email@radiants.ru	ООО ТоргПрофи		действительный
customer_012	Илья Муромец	12customer_email@radiants.ru	ООО Модуль Плюс		действительный
customer_013	Иван Сушанин	13customer_email@radiants.ru	ООО ТоргПрофи		действительный

Вы можете найти зарегистрированного пользователя-клиента или изменить его настройки, нажав на его имя. У вас также есть возможность изменить серверную часть клиента, для получения дополнительной информации см. главу о внешних серверных частях.

Чтобы создать новую учетную запись пользователя-клиента, нажмите кнопку «Добавить учетную запись клиента» (см. рис. ниже). Некоторые поля являются обязательными, т. е. они должны содержать значения, поэтому, если вы оставите одно из них пустым, оно будет выделено красным.

Рисунок 4.16. Добавление клиента

Управление учётными записями клиентов **Добавить учётную запись клиента**

Действия

← Перейти к обзору

← Назад к результатам поиска

Подсказка

Необходимо для наличия данных о клиенте и подключения к системе через интерфейс клиента.

Добавить учётную запись клиента

Заголовок или приветствие:

* Имя:

* Фамилия:

* Логин:

Пароль:

* email:

* ID компании:

телефон:

Факс:

Мобильный телефон:

Улица:

Индекс:

Город:

Страна:

Комментарий:

* Действительность:

Аутентификатор Google:

Enter your shared secret to enable two factor authentication.
WARNING: Make sure that you add the shared secret to your generator application and the application works well. Otherwise you will be not able to login anymore without the two factor token.Общий секрет

Язык интерфейса: Выберите главный язык интерфейса.Язык

Часовой пояс: Выберите свой часовой пояс: Все даты и время будут отображаться относительно этого часового пояса.Часовой пояс

Обзор заявок: Выберите период, после которого просмотр списка заявок будет автоматически обновлен.Интервал обновления

Количество отображаемых заявок: Выберите как много заявок будет показываться в обзоре по умолчанию.Заявок на страницу

PGP ключ: Фа...ран Загрузите ваш PGP-ключPGP ключ

Сертификат S/MIME: Фа...ран Загрузите Ваш S/MIME сертификат.Сертификат S/MIME

или

Пользователи-клиенты могут получить доступ к системе, указав свое имя пользователя и пароль. CustomerID необходим системе для идентификации пользователя и

связанных с ним заявок. Поскольку адрес электронной почты является уникальным значением, его можно использовать в качестве идентификатора.

С помощью параметра конфигурации AdminCustomerUser::UseAutoComplete в SysConfig можно активировать поле автозаполнения для выбора идентификатора клиента на экране добавления или редактирования пользователя клиента.

Примечание

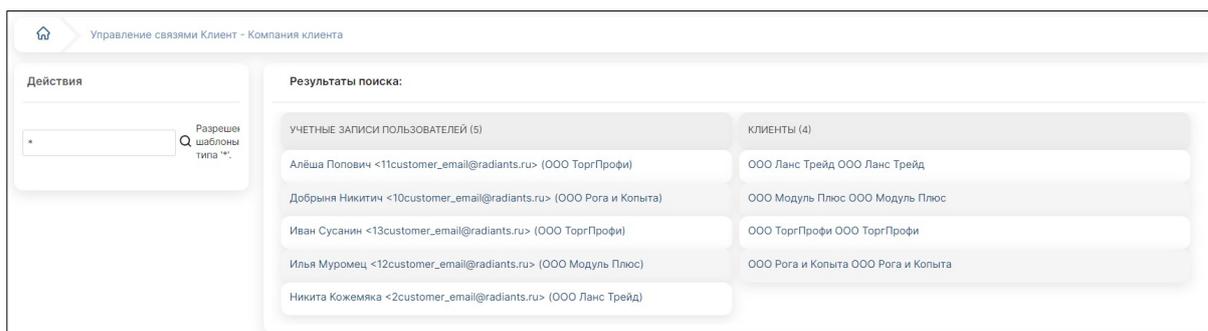
Как и в случае с агентами, группами и ролями, система не позволяет удалять клиентов, а только деактивировать их, установив опцию Действительность в значение *недействительный* или *временно недействительный*.

2. Клиенты - Компании

Пользователи-клиенты могут быть назначены нескольким компаниям (и наоборот), используя ссылку «Клиенты» ↔ «Компании» на странице администрирования. В дополнение к своей собственной (основной) компании они будут связаны с любой другой компанией, для которой существует определенное отношение.

Это дает им доступ к заявкам пользователей клиентов от этих компаний.

Рисунок 4.17. Управление связями Клиенты - Компании



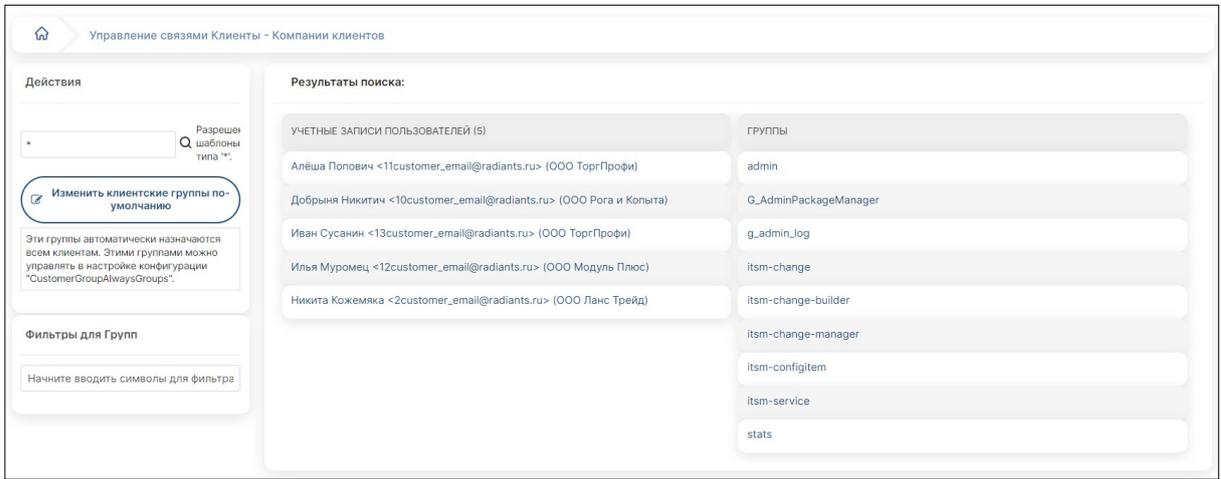
3. Клиенты ↔ Группы

Пользователи-клиенты также могут быть добавлены в группу, что может быть полезно, если вы хотите добавить пользователей-клиентов той же компании с доступом к одной или нескольким очередям. Сначала создайте группу, к которой будут принадлежать пользователи ваших клиентов, с помощью модуля управления группами. Затем добавьте очереди и выберите новую группу для очередей.

Следующим шагом является активация поддержки группы клиентов. Это можно сделать с помощью параметра конфигурации CustomerGroupSupport в SysConfig. Используя параметр CustomerGroupAlwaysGroups, вы можете указать группы по умолчанию для вновь добавленного пользователя-клиента, чтобы каждая новая учетная запись автоматически добавлялась в эти группы.

По ссылке Клиенты ↔ Группы вы можете управлять тем, какие клиенты должны принадлежать к разным группам (см. рисунок ниже).

Рисунок 4.18. Управление связями Клиент - Группа



Чтобы определить различные группы, в которые должен входить клиент, и наоборот, щелкните соответствующее имя пользователя или группу клиента (см. рисунки 4.19 и 4.20 ниже соответственно).

Рисунок 4.19. Изменение связи между Группами и Пользователем

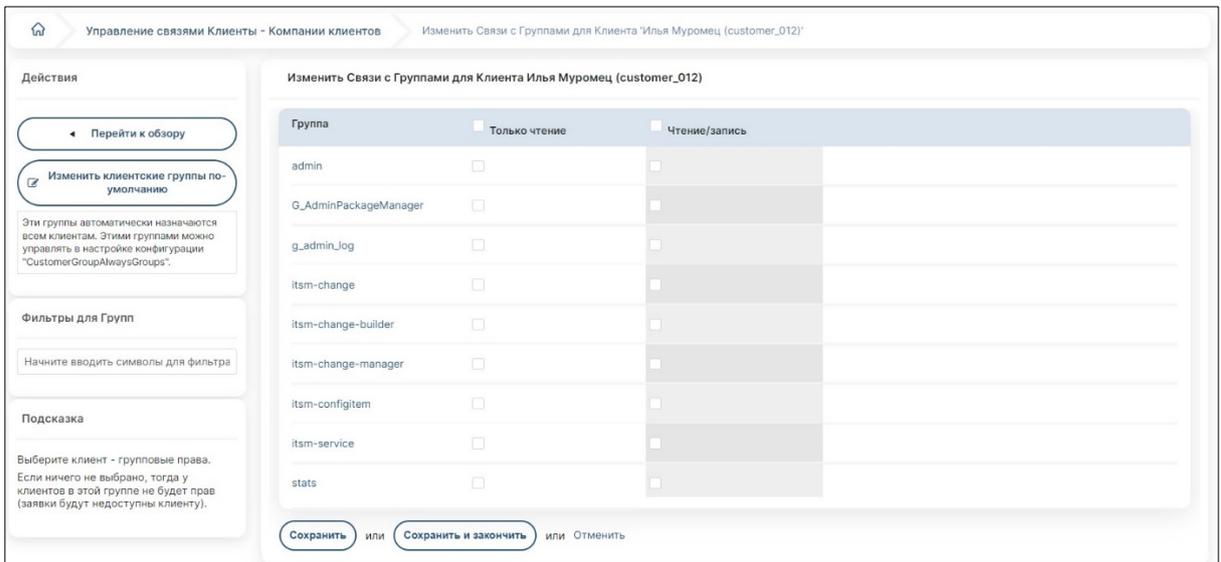
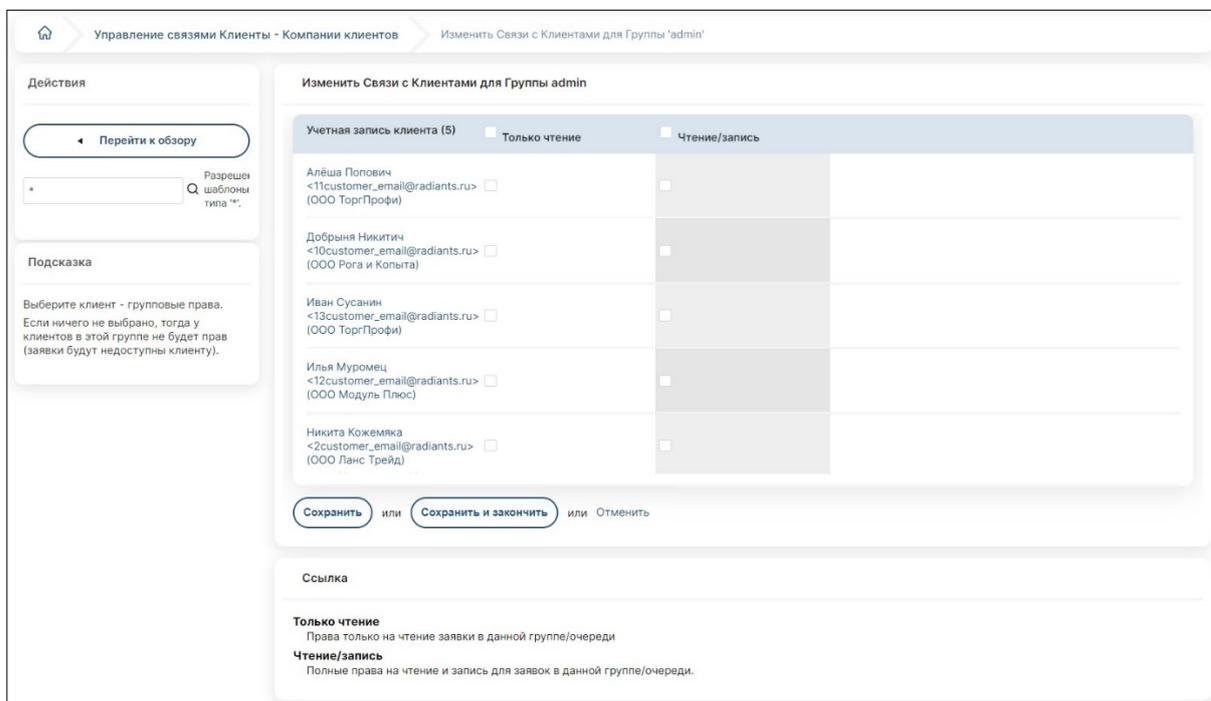


Рисунок 4.20. Изменение связи между Клиентами и Группой



4. Компании ↔ Группы

Компании также могут быть добавлены в группу, что может быть полезно, если вы не хотите добавлять групповые отношения к каждому отдельному пользователю клиента. Сначала создайте группу, к которой будут принадлежать ваши компании, с помощью модуля управления группами. Затем добавьте очереди и выберите новую группу для очередей.

Следующим шагом является активация поддержки группы клиентов. Это можно сделать с помощью параметра конфигурации CustomerGroupSupport в SysConfig. Используя параметр CustomerGroupAlwaysGroups, вы можете указать группы по умолчанию для новых добавленных компаний, чтобы каждая новая компания автоматически добавлялась в эти группы. Доступные уровни прав можно изменить с помощью конфигурации.

Назначенные права будут общими для всех пользователей-клиентов этой компании, поэтому разрешения не нужно давать индивидуально каждому пользователю-клиенту (хотя это все еще возможно, если требуется).

Доступны различные контексты назначения прав, которые будут определять, как наследуются права пользователями клиентов.

Доступны следующие контексты:

- Такой же клиент

Групповые права будут унаследованы всеми пользователями этой компании. Это предоставляет им доступ к заявкам всех пользователей клиента из этой компании, если они находятся в очереди, назначенной этой группе.

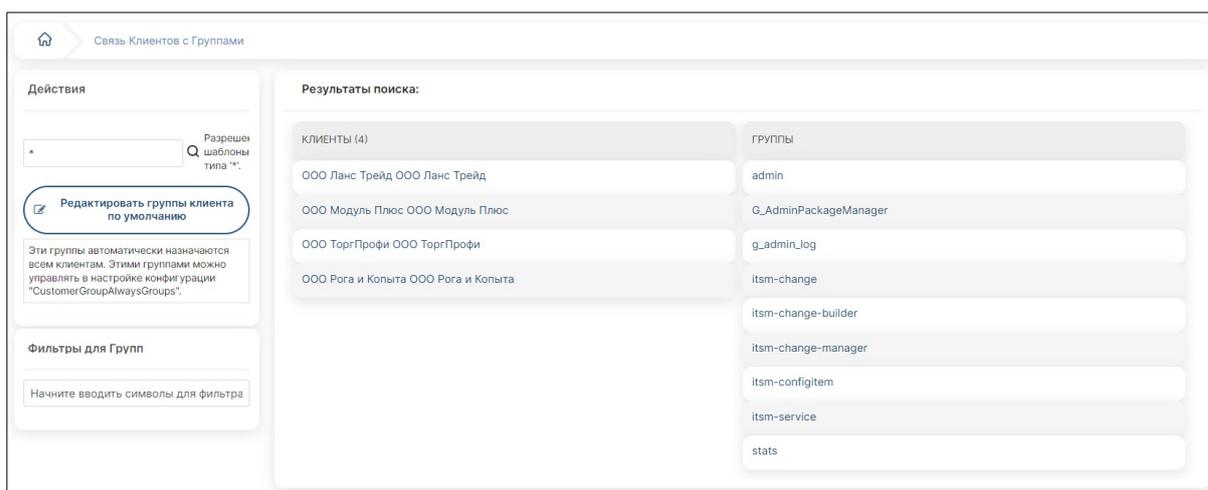
- Другой клиент

Групповые права будут унаследованы всеми пользователями этой компании. Это обеспечивает им доступ к заявкам всех пользователей клиентов от других компаний, если они находятся в очереди, назначенной этой группе, и, если другая компания также имеет права доступа к этой группе.

Доступ ограничен уровнем пользователя-клиента заявки.

По ссылке *Компании* ↔ *Группы* вы можете управлять тем, какие компании должны принадлежать к разным группам (см. рисунок ниже).

Рисунок 4.21. Управление связями Компании ↔ Группы



Чтобы определить различные группы, в которые должен входить компания, и наоборот, щелкните соответствующее наименование компании или группы клиента (см. рисунки 4.22 и 4.23 ниже соответственно).

Рисунок 4.22. Изменение связи между Группами и Компанией

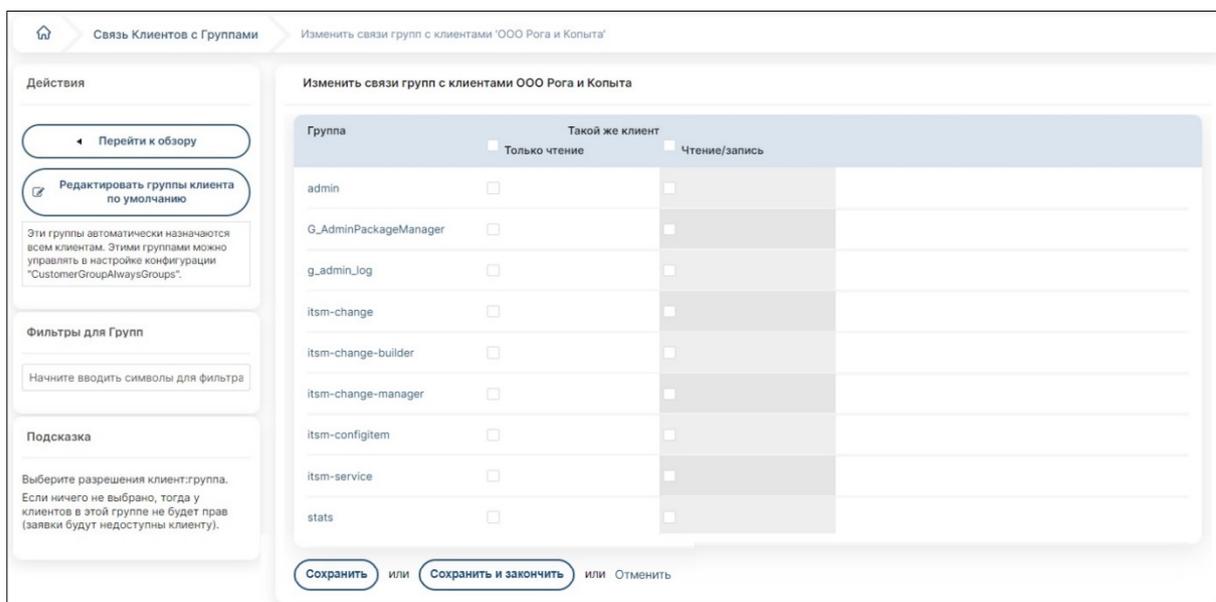
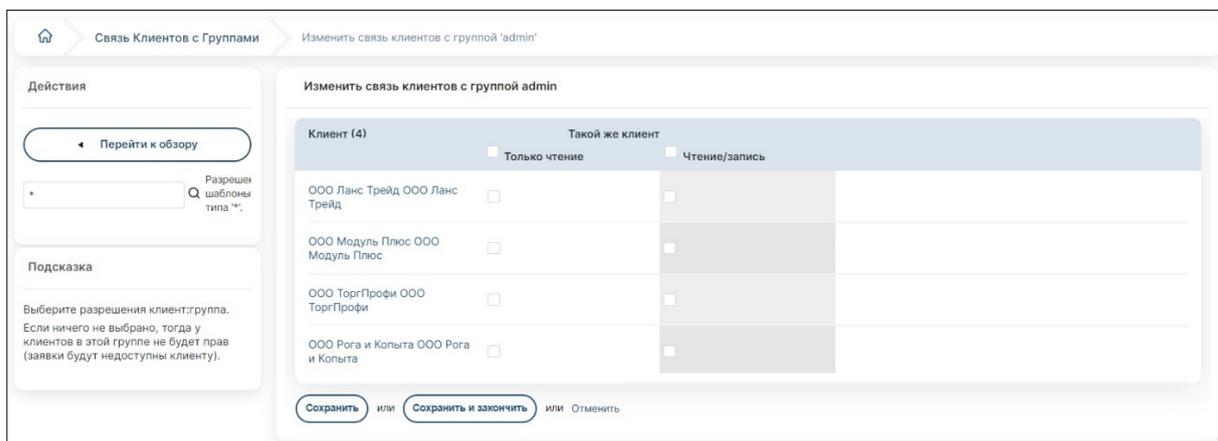


Рисунок 4.23. Изменение связи между Компаниями и Группой



5. Объяснение функций прав клиента (компании)

Доступ к заявкам в клиентском интерфейсе с включенной поддержкой групп в основном оценивается комбинацией групповых и индивидуальных (компания/клиент-пользователь) прав. Только при соблюдении обоих критериев предоставляется определенный доступ.

Если результирующий доступ rw, пользователь-клиент может просматривать и изменять заявку, если доступ ro возможен только просмотр.

Для создания заявок используются только групповые права, и пользователь-клиент может создавать заявки для всех очередей с разрешениями rw (если нет иных ограничений, например, ACL или CustomerPanelOwnSelection).

Групповые права являются аддитивными (это означает, что только один метод должен предоставить разрешения), и учитываются следующие возможности:

- Группы пользователей по умолчанию через настройку SysConfig.
- Группы, назначенные пользователю-клиенту через Пользователи-клиенты ↔ Компании на странице Администрирования
- Группы клиентов по умолчанию через настройку SysConfig.
- Группы, назначенные клиенту через Компании ↔ Группы на странице Администрирования

Для описанных выше методов используются все компании, связанные с пользователем клиента. Сюда входят «основной» клиент (выбранный в интерфейсе администрирования пользователей-клиентов), дополнительные клиенты (добавленные в интерфейсе администратора «Клиенты» ↔ «Компании») и другие клиенты, которые могут существовать в бэкенде.

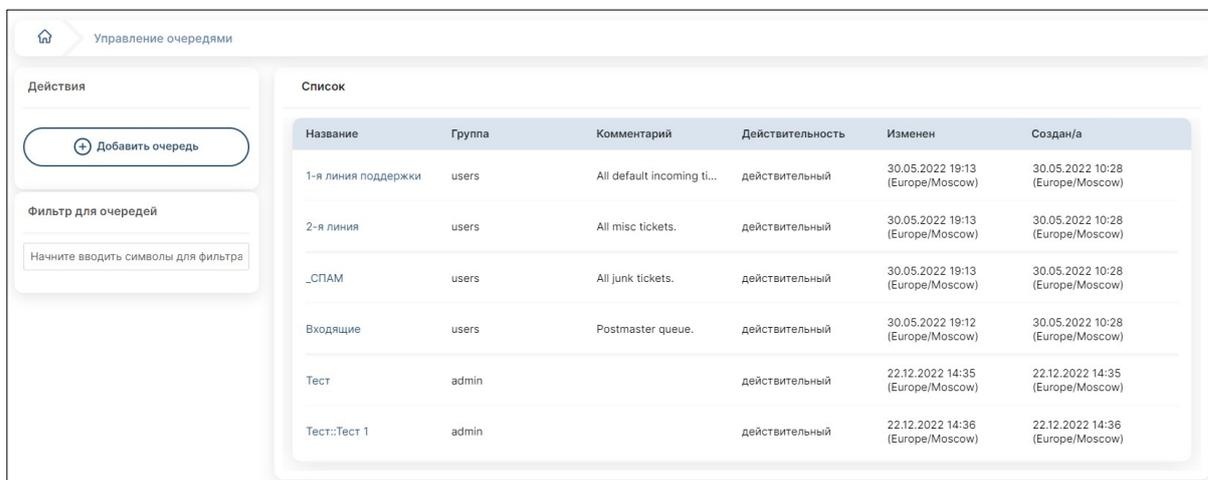
Индивидуальные проверки прав требуют выполнения одного из следующих условий:

- Заявка назначается пользователю - клиенту
- Заявка назначается компании, с которым связан пользователь-клиент (как описано выше).
- Заявка назначается клиенту с групповыми правами для очереди заявок, в то время как компания, связанная с пользователем-клиента, имеет разрешение «Другие клиенты» для той же группы.

1.4. Очереди

Нажав на ссылку «Очереди» на странице администрирования, вы можете управлять очередями вашей системы (см. рисунок ниже). В новой установке RSD есть 4 очереди по умолчанию: Raw, Junk, Misc и Postmaster. Все входящие сообщения будут храниться в очереди "Raw", если не заданы правила фильтрации. Очередь «Junk» может использоваться для хранения спам-сообщений.

Рисунок 4.24. Управления очередями



Название	Группа	Комментарий	Действительность	Изменен	Создан/а
1-я линия поддержки	users	All default incoming ti...	действительный	30.05.2022 19:13 (Europe/Moscow)	30.05.2022 10:28 (Europe/Moscow)
2-я линия	users	All misc tickets.	действительный	30.05.2022 19:13 (Europe/Moscow)	30.05.2022 10:28 (Europe/Moscow)
_СПАМ	users	All junk tickets.	действительный	30.05.2022 19:13 (Europe/Moscow)	30.05.2022 10:28 (Europe/Moscow)
Входящие	users	Postmaster queue.	действительный	30.05.2022 19:12 (Europe/Moscow)	30.05.2022 10:28 (Europe/Moscow)
Тест	admin		действительный	22.12.2022 14:35 (Europe/Moscow)	22.12.2022 14:35 (Europe/Moscow)
Тест::Тест 1	admin		действительный	22.12.2022 14:36 (Europe/Moscow)	22.12.2022 14:36 (Europe/Moscow)

Здесь вы можете добавлять очереди (см. рисунок ниже) и изменять их. Вы можете указать группу, которая должна использовать очередь. Вы также можете установить очередь как подочередь существующей очереди.

Рисунок 4.25. Добавление новой очереди

Управление очередями Добавить очередь

Действия

Добавить очередь

★ Название:

Подочередь для:

★ Группа:

Срок блокировки минут:

0 = не разблокировать - 24 часов = 1440 минут - С учетом только рабочего времени.
 Если агент блокирует заявку и не закрывает её, когда подошел таймаут разблокировки, заявка будет разблокирована и станет доступна другим агентам.

Эскалация - время первого ответа (минут): (Уведомление от)

0 — без эскалации - 24 часов = 1440 минут - С учетом только рабочего времени.
 Если к новой заявке не добавляются контакты клиента, либо телефонные, либо внешний email, до истечения указанного здесь времени, заявка эскалируется.

Эскалация - время обновления (минут): (Уведомление от)

0 — без эскалации - 24 часов = 1440 минут - С учетом только рабочего времени.
 Когда к заявке добавляется сообщение, через клиентский портал или электронной почтой, счетчик времени эскалации по обновлению сбрасывается и начинает отсчитываться заново. Если к заявке не добавляются контакты клиента, либо телефонные, либо внешний email, до истечения указанного здесь времени, заявка эскалируется.

Эскалация - время решения (минут): (Уведомление от)

0 — без эскалации - 24 часов = 1440 минут - С учетом только рабочего времени.
 Если заявка не закрыта до истечения указанного здесь времени разрешения, она эскалируется.

★ Последующие обращения:

Что делать с последующим обращением по уже закрытой заявке: переткрывать заявку, отклонять обращение или создать новую заявку.

★ Блокировка заявки после получения повторного ответа

клиента: Если заявка закрыта, а клиент снова посылает ответ, заявка будет заблокирована на старого владельца.

★ Адрес системы:

Установка адреса отправителя для ответов в этой очереди.

Ключ подписи по умолчанию:

To use a sign key, PGP keys or S/MIME certificates need to be added with identifiers for selected queue system address.

★ Приветствие:

Приветствие для ответов в электронных письмах.

★ Подпись:

Подпись для писем

Календарь:

★ Действительность:

Комментарий:

или

Вы можете определить тайм-аут разблокировки для очереди — если агент блокирует заявку и не закрывает ее до истечения тайм-аута разблокировки, заявка будет автоматически разблокирована и доступна для работы с другими агентами.

Существует три параметра времени эскалации, которые могут быть связаны на уровне очереди:

Эскалация — время первого ответа

- После создания заявки, если время, указанное здесь, истекает без какой-либо связи с клиентом по электронной почте или телефону, заявка передается на более высокий уровень.

Эскалация — время обновления

- Если есть последующее обращение клиента по электронной почте или через клиентский портал, что записано в заявке, время обновления эскалации сбрасывается. Если до истечения указанного здесь времени контакт с клиентом отсутствует, заявка эскалируется.

Эскалация — время решения

- Если заявка не закрыта до истечения указанного здесь времени, заявка эскалируется.

С помощью поля «Блокировка заявки после получения повторного ответа клиента» вы можете определить, следует ли установить для заявки значения «заблокировано» для старого владельца, если заявка, которая была закрыта, а клиент снова посылает ответ

и заявка открывается повторно. Это гарантирует, что заявка будет обработана агентом, который ранее обрабатывал эту заявку.

Параметр для системного адреса указывает адрес электронной почты, который будет использоваться для исходящих заявок из этой очереди. Существует также возможность связать очередь с приветствием и подписью для ответов по электронной почте. Для получения более подробной информации, пожалуйста, обратитесь к разделам адреса электронной почты, приветствия и подписи.

Примечание

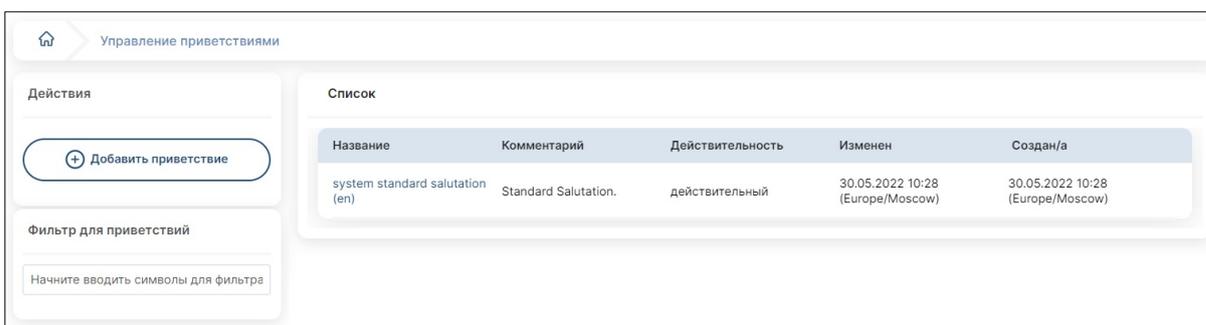
Как в случае с агентами, группами и клиентами, очереди не могут быть удалены, а только отключены, путем установки опции "Действительная" в значение *недействительная* или *временно недействительная*.

1.5. Приветствия, Подписи, Вложения и Шаблоны

1. Приветствия

Приветствие — это текстовый модуль для шаблона. Приветствия могут быть связаны с одной или несколькими очередями, как описано в разделе об очередях. Приветствие используется только в том случае, если на заявку из очереди, с которой связано приветствие, ответили. Для управления различными приветствиями вашей системы используйте раздел «Приветствия» на странице Администрирования (см. рисунок ниже).

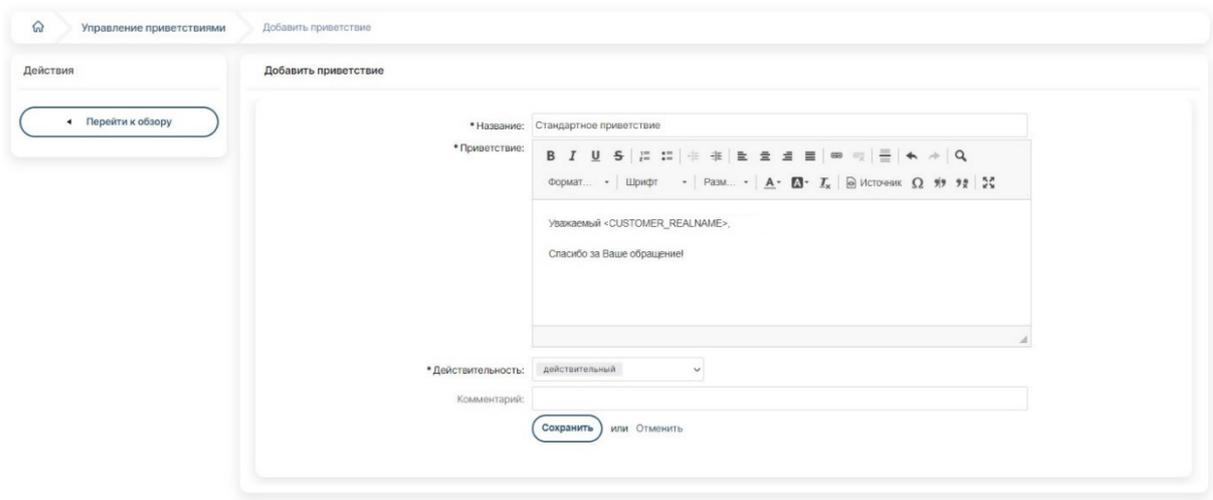
Рисунок 4.26. Управления приветствиями



После установки в системе есть одно предопределенное приветствие: "system standard salutation (en)".

Чтобы создать новое приветствие, нажмите на кнопку "Добавить приветствие", введите необходимые данные и нажмите кнопку "Сохранить" (см. Рисунок ниже)

Рисунок 4.27. Добавление нового приветствия



В приветствиях можно использовать тэги. Когда вы отвечаете на заявку, имена тэгов будут заменены на их значения.

Примечание

Как и в случае с другими сущностями системы RSD, система не позволяет удалять приветствия, только деактивировать их, установив параметр Действительность в значение *недействительный* или временно *недействительный*.

2. Подписи

Еще одним текстовым модулем для шаблона является подпись. Подписи могут быть связаны с очередью, как описано в разделе об очередях. Обратите внимание, что подпись будет добавлена к тексту шаблона только в том случае, если он ранее был связан с очередью. Вы можете управлять подписями в своей системе, перейдя по ссылке «Подписи» на странице Администрирования (см. рисунок ниже).

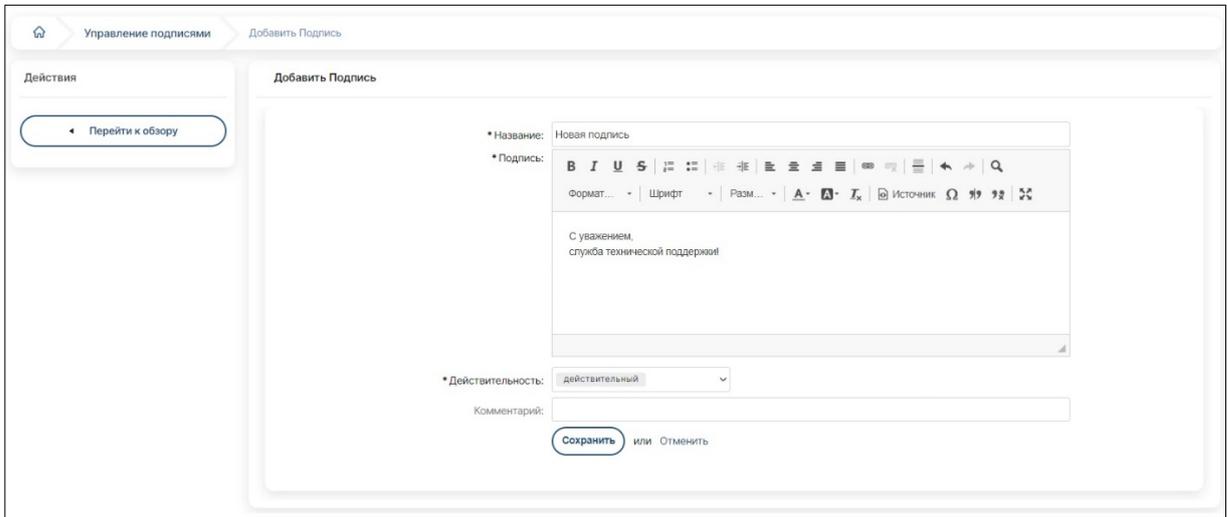
Рисунок 4.28. Управление подписями



После новой установки RSD в вашей системе хранится одна предопределенная подпись, «системная стандартная подпись (en)».

Чтобы создать новую подпись, нажмите кнопку «Добавить подпись», укажите необходимые данные и сохраните их (см. рис. ниже).

Рисунок 4.29. Добавление новой подписи



Как и приветствия, подписи также могут содержать динамическое содержимое, например, имя и фамилию агента, отвечающего на заявку. Здесь также можно использовать теги для замены содержимого текста подписи для каждой заявки. Теги, которые можно использовать, см. в нижней части экрана подписей. Например, если вы включите переменную `<RADIANT_LAST_NAME>` в подпись, фамилия агента, отвечающего на заявку, заменит эту переменную.

Примечание

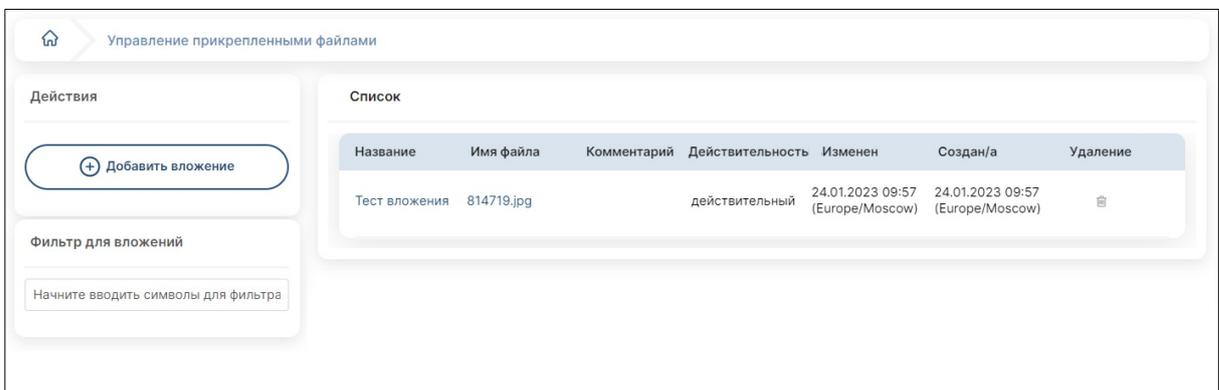
Как и в случае с приветствиями, подписи не могут быть удалены, только деактивированы путем установки параметра Действительность в значение *недействительный* или *временно недействительный*.

3. Вложения

Вы также можете дополнительно добавить одно или несколько вложений в шаблон. Если шаблон выбран, вложения будут прикреплены к окну составления сообщения. При необходимости агент может удалить вложение из отдельного шаблона перед отправкой его заказчику.

Через ссылку «Вложения» на странице администрирования вы можете загрузить вложения в базу данных системы (см. рис. ниже).

Рисунок 4.30. Управление вложениями



Чтобы создать новое вложение, нажмите кнопку «Добавить вложение», укажите необходимые данные и сохраните их (см. рис. ниже).

Рисунок 4.31. Добавление нового вложения

Управление прикрепленными файлами > Добавить вложение

Действия

← Перейти к обзору

Добавить вложение

* Название: Тест вложения

* Вложение: Выберите файл 814719.jpg

* Действительность: действительный

Комментарий:

Сохранить или Отменить

Если вложение сохранено, оно может быть связано с одним или несколькими шаблонами. Для этого перейдите в раздел «Шаблоны ↔ Вложения» на странице администрирования (см. рис. ниже).

Рисунок 4.32. Связать вложения с шаблонами

Управление связями шаблон-вложение

Фильтр для Шаблонов

Начните вводить символы для фильтра

Фильтр для вложений

Начните вводить символы для фильтра

Обзор

ШАБЛОНЫ	ПРИКРЕПЛЕННЫЕ ФАЙЛЫ
Ответ - empty answer	Тест вложения (814719.jpg)
Ответ - test answer	

Чтобы связать разные вложения с определенным шаблоном и наоборот, нажмите на название соответствующего шаблона или вложения (см. ниже рисунки 4.33 и 4.34 соответственно).

Рисунок 4.33. Изменить связь Вложения с Шаблоном

Управление связями шаблон-вложение > Изменить связь Вложения с Шаблоном 'Answer - empty answer'

Действия

← Перейти к обзору

Фильтр

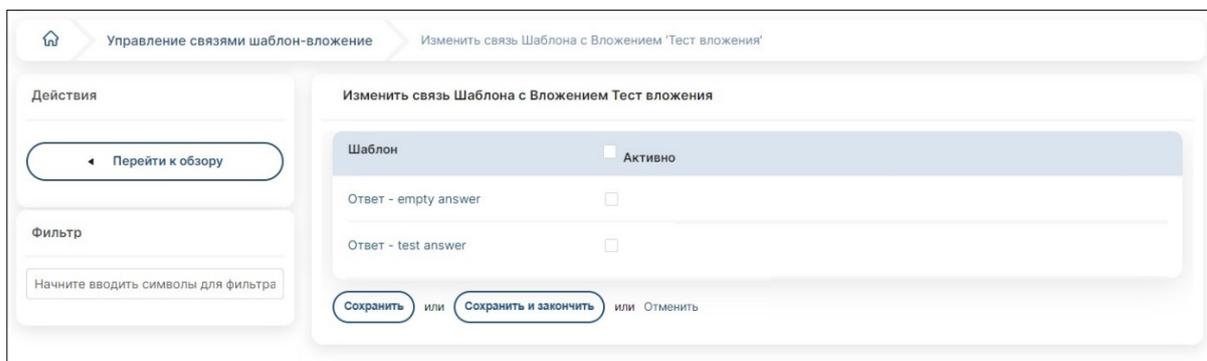
Начните вводить символы для фильтра

Изменить связь Вложения с Шаблоном Answer - empty answer

Вложение	Активно
Тест вложения (814719.jpg)	<input type="checkbox"/>

Сохранить или Сохранить и закончить или Отменить

Рисунок 4.34. Изменить связь Шаблона с Вложением



4. Шаблоны

Чтобы ускорить обработку заявок и стандартизировать внешний вид ответов, вы можете задать шаблоны в RSD. Шаблон может быть связан с одной или несколькими очередями и наоборот.

Существует несколько различных типов шаблонов, которые могут быть использованы для различных целей в различных местах в RSD. Ниже приведен список доступных типов шаблонов:

- Ответ: будет использоваться в качестве ответа на заявку
- Создать: будет использоваться в качестве Телефонной Заявки или Email заявки
- Переслать: используется для пересылки сообщения кому-то другому
- Телефонный звонок: для регистрации входящих и исходящих звонков агентом

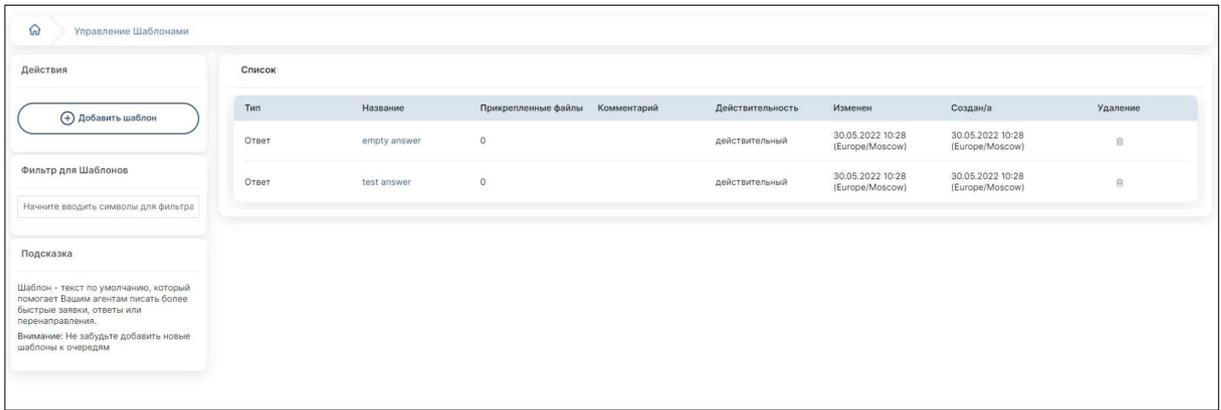
Использовать шаблоны ответов можно двумя способами: при подробном просмотре заявки в меню сообщений/заметок, или более быстрым - с экрана Обзора статусов или заявок в формате предпросмотра/Large. Во вновь установленной RSD, шаблон (ответа) "пустой ответ" задан для всех очередей.

Как только шаблоны «Переслать» будут добавлены и назначены очередям, кнопка «Переслать» в масштабе заявки (которая обычно приводит к пустому экрану переадресации текста) изменится на выпадающий список, список будет заполнен добавленными шаблонами «Переслать», выбрав один из шаблонов, экран переадресации будет отображаться предварительно заполненным текстом шаблона и вложениями (аналогично окну выбора ответа с шаблонами ответов).

Создание шаблонов типа «Создать» и «Телефонный звонок» сделает видимым окно выбора «Текстовый шаблон» на соответствующих экранах, при выборе шаблона для списка будут заполнены поля «Текст» и «Вложение» (если они доступны в шаблоне). Обратите внимание, что любое предыдущее изменение текста или вложений будет перезаписано выбором шаблона.

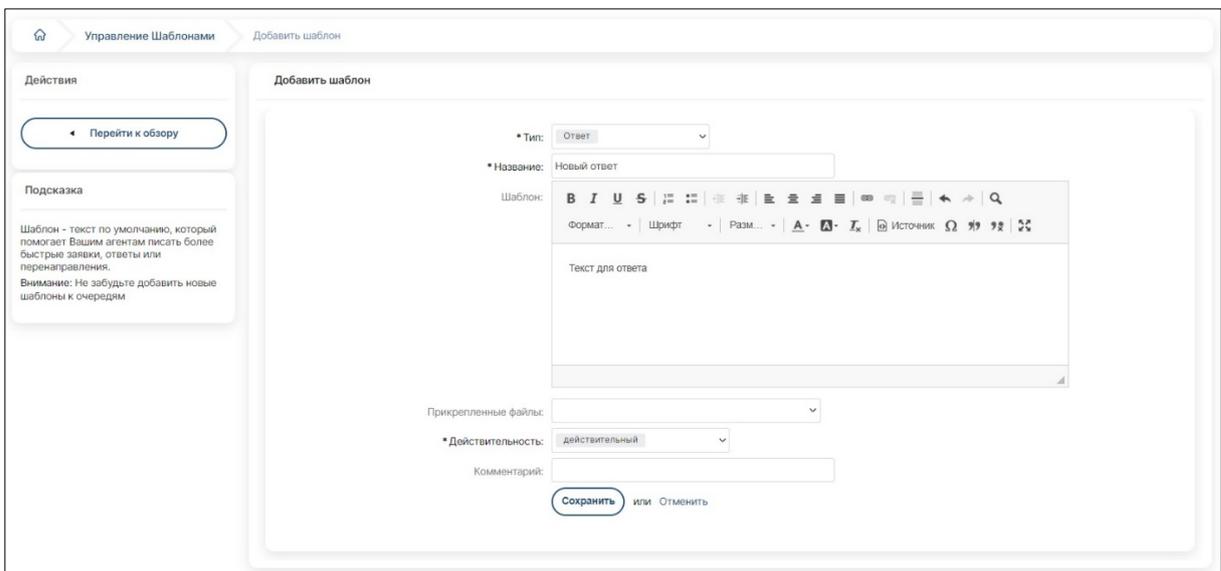
Нажав на ссылку «Шаблоны» на странице администрирования, вы попадете на экран управления шаблонами (см. рисунок ниже).

Рисунок 4.35. Управление Шаблонами



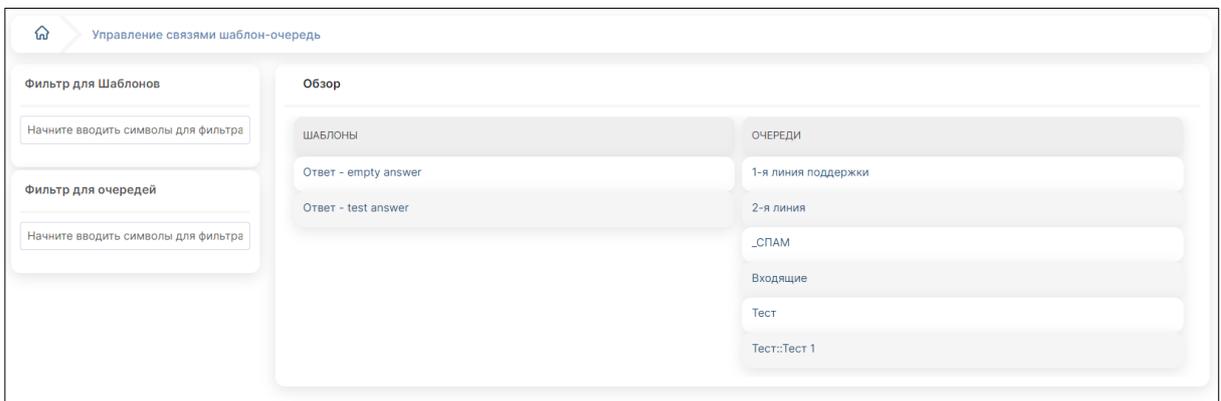
Чтобы создать новый шаблон, нажмите кнопку «Добавить шаблон», укажите необходимые данные (не забудьте выбрать соответствующий тип шаблона) и сохраните его (см. рисунок ниже).

Рисунок 4.36. Добавление шаблона



Чтобы добавить/удалить шаблоны в одну или несколько очередей, нажмите на ссылку «Шаблоны ↔ Очереди» на странице администрирования (см. рис. ниже). Вы также можете использовать фильтры для получения информации о конкретном объекте.

Рисунок 4.37. Настройка Шаблонов для Очередей



Чтобы определить разные шаблоны, которые будут доступны для очереди и наоборот, щелкните соответствующий шаблон или очередь (см. ниже рисунки 4.38 и 4.39 соответственно).

Рисунок 4.38. Изменить связь Очереди с Шаблоном

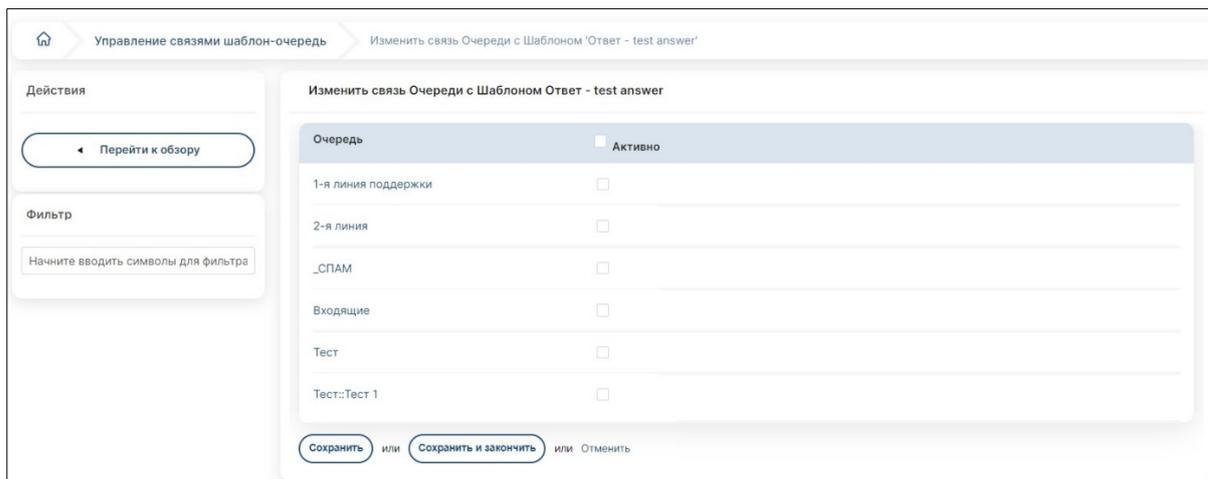
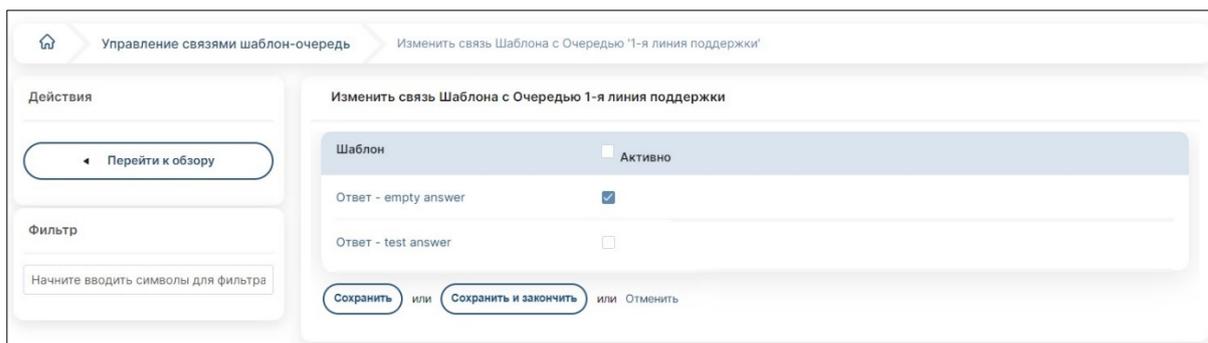


Рисунок 4.39. Изменить связь Шаблона с Очередью



При выборе шаблона в текст шаблона может быть добавлена дополнительная информация, это зависит от типа шаблона:

Шаблоны «PhoneCall/Телефонный звонок» и «Create/Создать» не добавляют никакого содержимого в текст шаблона, однако экран «New Email Ticket/Электронная почта» добавляет назначенную очередь подпись к полученному телу электронной почты (на этом экране есть отдельное поле для визуализации подписи).

Текст шаблонов при выборе типа «Ответ» также включает приветствие, связанное с очередью, затем текст шаблона, затем текст цитируемой заявки и, наконец, подпись, связанную с очередью.

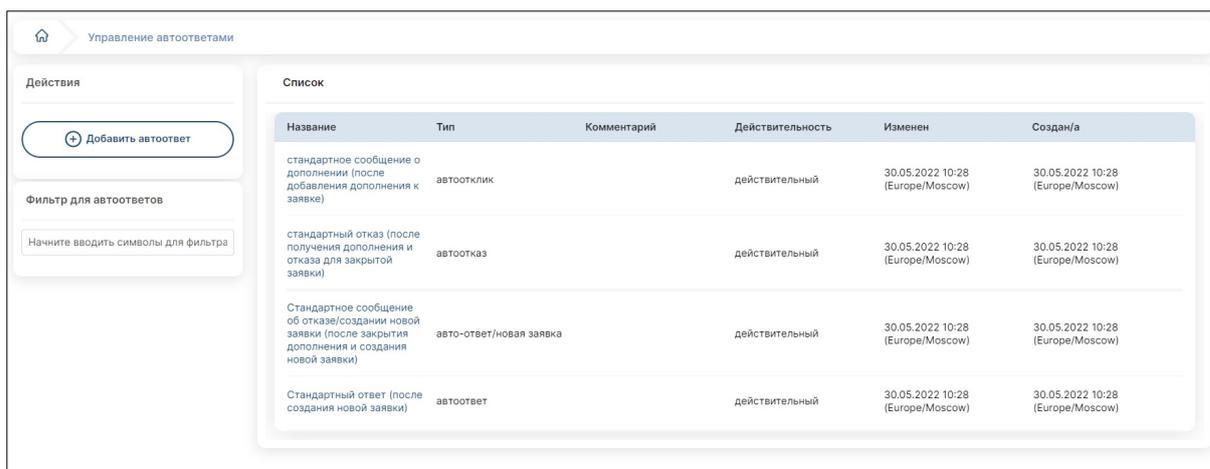
Шаблоны «Переслать» аналогичны шаблонам ответов, но не включают часть приветствия.

1.6. Автоответы

RSD позволяет отправлять автоматические ответы клиентам на основе возникновения определенных событий, таких как создание заявки в определенной очереди, получение последующего сообщения в отношении заявки, закрытие или отклонение заявки, и т.

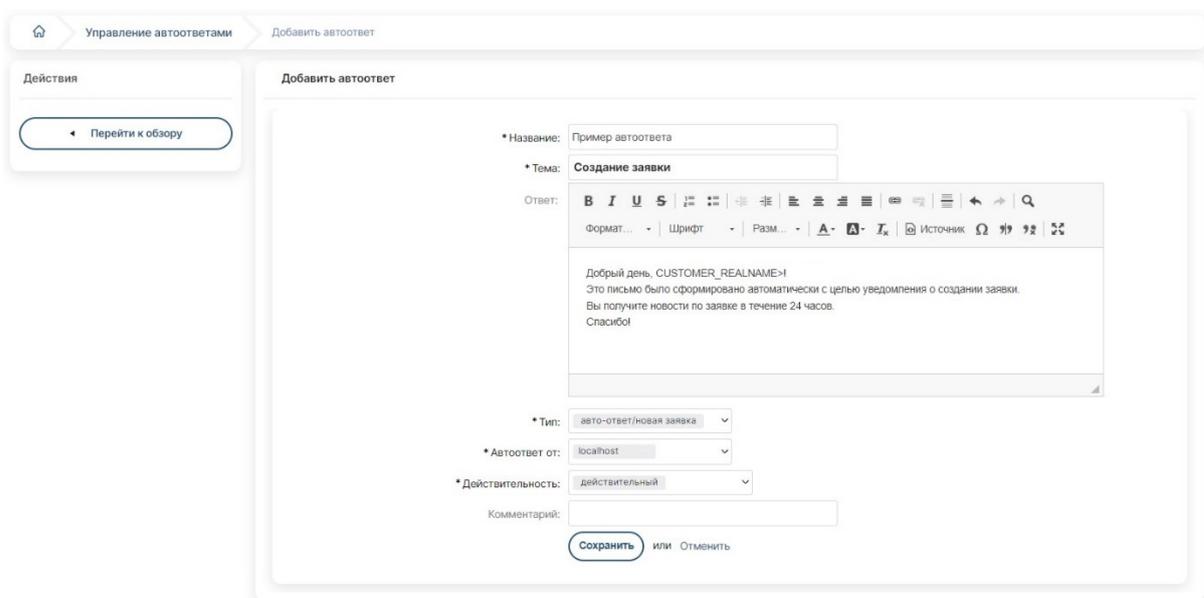
д. Для управления такими ответами нажмите ссылку «Автоответы» на странице администрирования (см. рис. ниже).

Рисунок 4.40. Управление автоответами



Чтобы создать автоответ, нажмите кнопку «Добавить автоответ», укажите необходимые данные и сохраните их (см. рис. ниже).

Рисунок 4.41. Добавление автоответа



Тема и текст автоответа могут быть сгенерированы тегами, как в подписях и приветствиях. Например, если вы вставите тег `<RADIANT_CUSTOMER_EMAIL[5]>` в тело автоответа, первые 5 строк текста письма клиента будут вставлены в автоответ. Вы найдете более подробную информацию о допустимых тегах, которые можно использовать, в нижней части экрана, показанного на рисунке.

Для каждого автоматического ответа вы можете указать событие, которое должно его вызвать. Системные события, доступные после установки по умолчанию, описаны в Таблице 4-4.

Таблица 4.4. События для автоответов

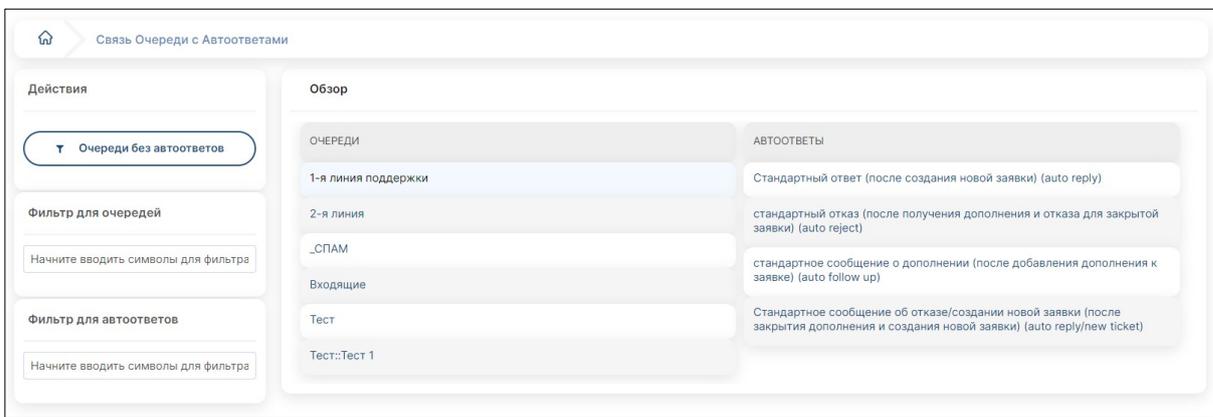
Название	Описание
автоответ	Создание заявки в определенной очереди
автоответ/новая заявка	Повторное открытие уже закрытой заявки, например, если клиент отвечает на такую заявку.
автоотклик	Прием входящих заявок
автоотказ	Автоматический отказ от заявки, созданной системой.
автоудаление	Удаление заявки, созданной системой.

Примечание

Как в случае с другими сущностями RSD, автоответы также не могут быть удалены, только деактивированы путем установки опции Действительность в значение *недействительный* или *временно недействительный*.

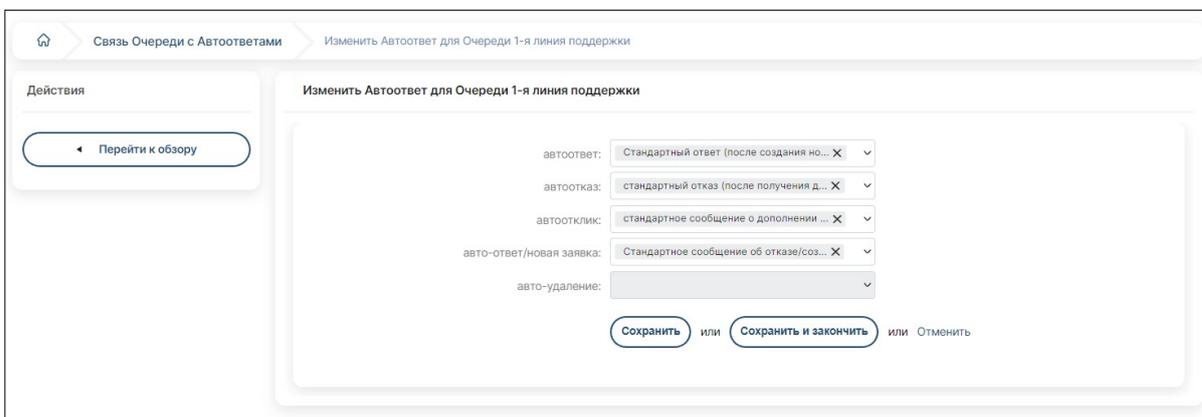
Чтобы добавить автоответ в очередь, используйте раздел «Автоответ ↔ Очереди» на странице администрирования (см. рис. ниже). Все системные события перечислены для каждой очереди, а автоматический ответ с тем же событием можно выбрать или удалить с помощью списка.

Рисунок 4.42. Настройка автоответов для очередей



Чтобы определить различные автоматические ответы, которые будут доступны для очереди, нажмите на имя соответствующей очереди (см. рисунок ниже). Также возможно отредактировать существующий автоответ — для этого нажмите на ответ и отредактируйте так же, как и при редактировании нового автоответа.

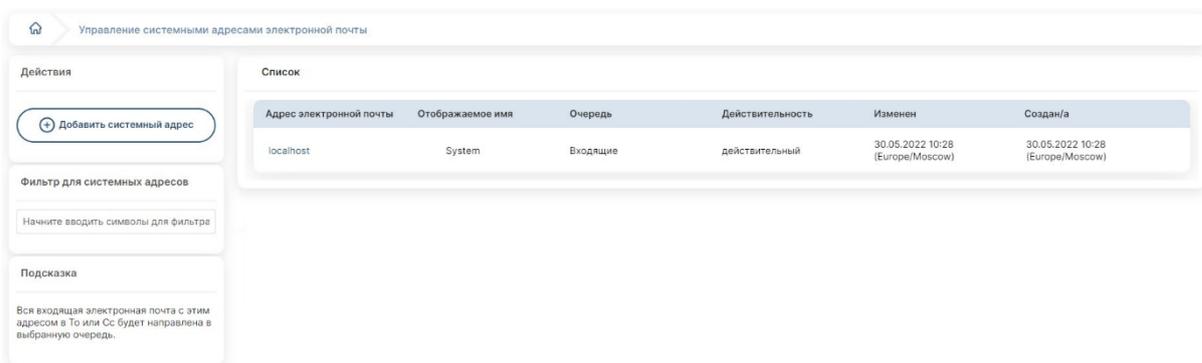
Рисунок 4.43. Изменение автоответов для очереди



1.7. Системные адреса электронной почты

Чтобы разрешить RSD отправлять электронные письма, вам нужен действующий адрес электронной почты, который будет использоваться системой. RSD может работать с несколькими адресами электронной почты, поскольку во многих установках поддержки необходимо использовать более одного. Очередь может быть связана со многими адресами электронной почты и наоборот. Адрес, используемый для исходящих сообщений из очереди, можно задать при создании очереди. Используйте раздел «Адреса электронной почты» на странице администрирования для управления всеми адресами электронной почты системы (см. рисунок ниже).

Рисунок 4.44. Управление системными адресами электронной почты



Если вы создаете новый почтовый адрес (см. рисунок ниже), вы можете выбрать очередь или подочередь, которые будут связаны с ним. Эта позволяет системе сортировать входящие сообщения по адресу, указанному в почте в поле «Кому:», в нужную очередь.

Рисунок 4.45. Добавление системного адреса исходящей почты

Управление системными адресами электронной почты > Добавить системный адрес электронной почты

Действия

← Перейти к обзору

Подсказка

Вся входящая электронная почта с этим адресом в To или Cc будет направлена в выбранную очередь.

Добавить системный адрес электронной почты

* Адрес электронной почты: support@mycompany.com

* Отображаемое имя: Поддержка

Отображаемое имя и адрес электронной почты будут показываться в отправляемой вами почте.

* Очередь: 1-я линия поддержки

* Действительность: действительный

Комментарий:

Сохранить или Отменить

Примечание

Как и другие сущности RSD, email адреса не могут быть удалены, только деактивированы, установив поле Действительность в значение *не действительный* или *временно недействительный*.

1.8. Уведомления по заявкам

RSD позволяет отправлять уведомления о заявках агентам и клиентам в зависимости от наступления определенных событий. Агенты могут настроить параметры уведомлений о заявках по ссылке настроек.

С помощью раздела «Уведомления по заявкам» на странице администрирования вы можете управлять уведомлениями о заявках вашей системы (см. рисунок ниже). RSD поставляется с набором предопределенных уведомлений, которые охватывают широкий спектр вариантов использования.

Рисунок 4.46. Настройка уведомлений по заявкам

Управление уведомлениями

Действия

Добавить уведомление

Экспортировать уведомления

Фильтр для уведомлений

Начните вводить символы для фильтра

Импорт конфигурации

Здесь вы можете загрузить конфигурационный файл для импорта Ticket Notifications в вашу систему. Файл должен быть в формате .xml, экспортированный из модуля Ticket Notification.

Выберите файл | Файл не выбран

Перезаписать существующие уведомления?

Импортировать настройки уведомлений

Список

Название	Комментарий	Действительность	Изменен	Создан/а	Экспортировать	Копировать	Удаление
Уведомление о создании заявки		действительный	30.05.2022 10:28 (Europe/Moscow)	30.05.2022 10:28 (Europe/Moscow)	↓	☰	🗑️
Уведомление об отказе доставки электронной почты заявки		действительный	30.05.2022 10:28 (Europe/Moscow)	30.05.2022 10:28 (Europe/Moscow)	↓	☰	🗑️
Уведомление об эскалации заявки		действительный	30.05.2022 10:28 (Europe/Moscow)	30.05.2022 10:28 (Europe/Moscow)	↓	☰	🗑️
Уведомление о предупреждении по эскалации заявки		действительный	30.05.2022 10:28 (Europe/Moscow)	30.05.2022 10:28 (Europe/Moscow)	↓	☰	🗑️
Уведомление об ответе по заявке (для заблокированных)		действительный	30.05.2022 10:28 (Europe/Moscow)	30.05.2022 10:28 (Europe/Moscow)	↓	☰	🗑️
Уведомление об ответе по заявке (для разблокированных)		действительный	30.05.2022 10:28 (Europe/Moscow)	30.05.2022 10:28 (Europe/Moscow)	↓	☰	🗑️
Уведомление об истечении срока блокировки заявки системой		действительный	30.05.2022 10:28 (Europe/Moscow)	30.05.2022 10:28 (Europe/Moscow)	↓	☰	🗑️
Уведомление о новой заметке в заявке		действительный	30.05.2022 10:28 (Europe/Moscow)	30.05.2022 10:28 (Europe/Moscow)	↓	☰	🗑️

Вы можете настроить многие аспекты уведомлений. Нажмите на уведомление, которое хотите изменить, и его содержимое будет загружено для редактирования (см. рис. ниже).

Рисунок 4.47. Настройка уведомления

▼ Получатели

Кому отправить:

Отправить этим агентам:

Отправить всем членам группы (только агенты):

Отправить всем членам роли:

Отправить для отсутствующих на месте: Также отправить, если пользователь в настоящее время отсутствует на месте.

Один раз в день: Уведомить пользователя только раз в день для каждой отдельной заявки, используя указанный способ доставки.

Кроме того, вы можете указать, должно ли уведомление отправляться агентам, находящимся вне офиса, и ограничить отправку до одного раза в день.

Уведомления можно отправлять разными способами. В RSD доступен метод уведомления «Электронная почта». С RSD вы также получаете возможность хранить и просматривать уведомления в базе данных (так что для использования RSD не требуется почтовый клиент), а также отправлять их по SMS (например, для очень важных уведомлений).

Рисунок 4.49. Настройка способов уведомлений

▼ Способы уведомлений

Существует несколько способов для отправки этого уведомления каждому из получателей. Выберите хотя бы один из приведенных ниже.

email

Активировать этот способ уведомлений:

Дополнительный адрес получателя:

Use comma or semicolon to separate email addresses.
Вы можете использовать тэги типа <TICKET_DynamicField_...> для вставки значений из текущей заявки.

Сообщение видно клиенту:

Сообщение/Заметка будет создана, если уведомление отправляется клиенту или на дополнительный адрес почты.

Шаблон письма:

Используйте этот шаблон для создания полного почтового сообщения (только для сообщений в HTML формате).

Включить защиту почты:

PGP and SMIME not enabled.

Уровень защиты почты:

Если ключ/сертификат подписи отсутствует:

Если ключ/сертификат шифрования отсутствует:

Веб просмотр

Эта функция в данный момент не доступна

СМС (от англ. Short Message Service — «служба коротких сообщений»)

Эта функция в данный момент не доступна

Поле «Дополнительный адрес получателя» предоставляет возможность использовать RADIANT-теги типа <RADIANT_TICKET_DynamicField_...> для вставки значений из текущей заявки. Соответствующее поле SMS (доступно только в RSD) также имеет эту функцию.

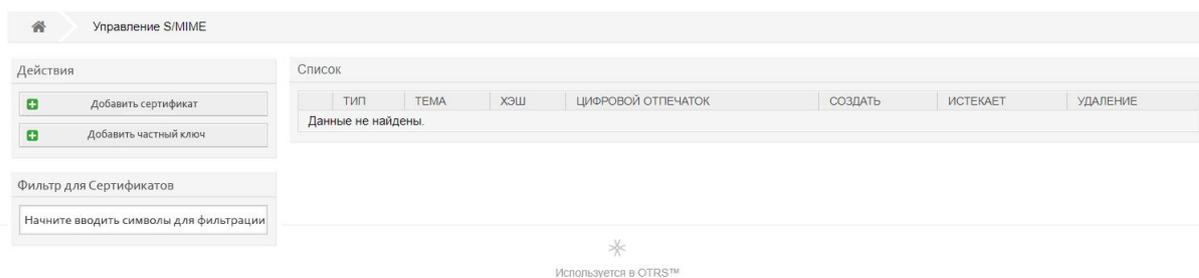
Поскольку передача электронной почты RSD 5s содержит параметры безопасности для каждого уведомления, это включает в себя возможности подписи и шифрования с помощью PGP и S/MIME, а также возможность решить, что делать в случае отсутствия ключа или сертификата.

1.9. S/MIME

RSD может обрабатывать зашифрованные сообщения S/MIME и подписывать исходящие письма. Для использования этой возможности необходимо сначала активировать ее и изменить ряд параметров в SysConfig/Конфигурации системы.

Ссылка «Сертификаты S/MIME» на странице администратора позволяет вам управлять своими сертификатами S/MIME (см. рисунок ниже). Вы можете добавлять или удалять сертификаты, а также выполнять поиск по данным SMIME.

Рисунок 4.50. Настройка S/MIME

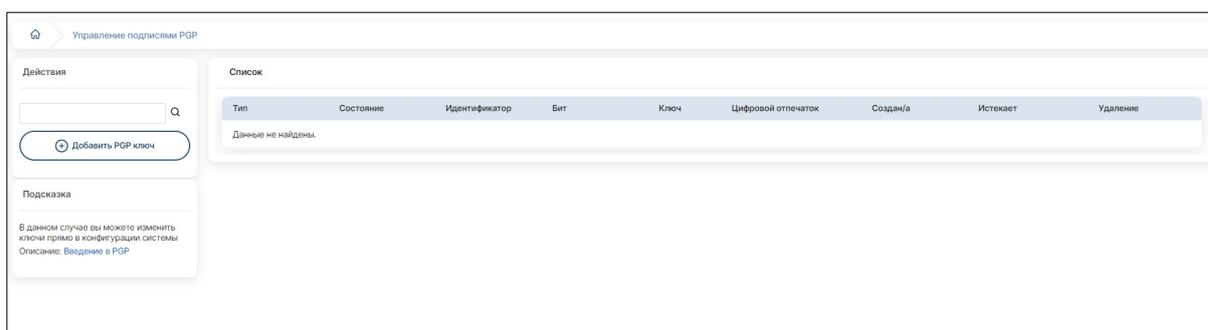


1.10. PGP

RSD обрабатывает ключи PGP, что позволяет вам шифровать/дешифровать сообщения и подписывать исходящие сообщения. Прежде чем эту функцию можно будет использовать, ее необходимо активировать и изменить некоторые параметры конфигурации в файле SysConfig.

С помощью ссылки «PGP ключи» на странице администрирования можно управлять списком ключей пользователя, который будет использоваться для PGP с RSD (см. рисунок ниже), например, локальным пользователем RSD или пользователем веб-сервера. Можно добавлять и удалять ключи и подписи, а также выполнять поиск по всем данным в списке ключей.

Рисунок 4.51. Настройка PGP



1.11. Состояния

С помощью раздела «Состояния» на странице администрирования вы можете управлять различными состояниями заявок, которые вы хотите использовать в системе (см. рисунок ниже).

Рисунок 4.52. Настройка статусов

Управление состояниями

Действия

Добавить состояние

Фильтр для состояний

Начните вводить символы для фильтра

Подсказка

Внимание: Пожалуйста, обновите также состояние и в Конфигурации Системы (там, где необходимо). См. также: <https://doc.otrs.com/doc>

Список

Название	Тип	Комментарий	Действительность	Изменен	Создан/а
	закрота	Ticket is closed ...	действительный	24.01.2023 12:37 (Europe/Moscow)	30.05.2022 10:30 (Europe/Moscow)
	ожидает автозакрота	Ticket is pending...	действительный	24.01.2023 12:37 (Europe/Moscow)	30.05.2022 10:28 (Europe/Moscow)
removed	удалена	Customer removed ...	действительный	30.05.2022 10:28 (Europe/Moscow)	30.05.2022 10:28 (Europe/Moscow)
в работе	открыта	Open tickets.	действительный	30.05.2022 19:01 (Europe/Moscow)	30.05.2022 10:28 (Europe/Moscow)
закрота	закрота	Ticket is closed ...	действительный	30.05.2022 19:01 (Europe/Moscow)	30.05.2022 10:28 (Europe/Moscow)
новое	новая	New ticket create...	действительный	30.05.2022 19:01 (Europe/Moscow)	30.05.2022 10:28 (Europe/Moscow)
объединены	объединена	State for merged ...	действительный	30.05.2022 19:05 (Europe/Moscow)	30.05.2022 10:28 (Europe/Moscow)
ожидание	ожидает напоминания	Ticket is pending...	действительный	30.05.2022 19:04 (Europe/Moscow)	30.05.2022 10:28 (Europe/Moscow)
ожидание клиента	ожидает напоминания		действительный	30.05.2022 19:05 (Europe/Moscow)	30.05.2022 19:05 (Europe/Moscow)
отклонено	закрота	Ticket is closed ...	действительный	30.05.2022 19:04 (Europe/Moscow)	30.05.2022 10:28 (Europe/Moscow)
решено	ожидает автозакрота	Ticket is pending...	действительный	30.05.2022 19:04 (Europe/Moscow)	30.05.2022 10:28 (Europe/Moscow)

После установки в системе будут следующие предустановленные состояния:

- закрыта успешно
- закрыта неуспешно
- объединенные
- новая
- открытая
- ожидание авто закрытие+
- ожидание авто закрытие-
- ожидание напоминание
- удаленная

Каждое состояние связано с типом, который необходимо указать, если создается новое состояние. По умолчанию установлены следующие типы состояния:

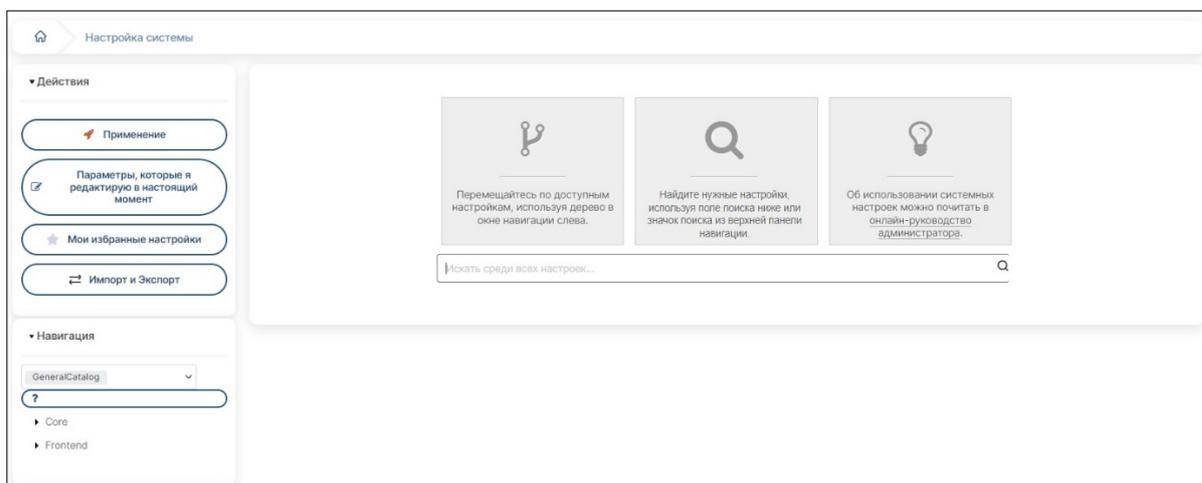
- закрытая
- объединенные
- новая
- открытая
- ожидание автозакрота
- ожидание напоминание
- удаленная

1.12. System Configuration

Ссылка «Конфигурация системы» ведет к разделу, в котором поддерживаются многие параметры конфигурации RSD.

Ссылка для настройки системы на странице администрирования загружает графический интерфейс пользователя для настройки системы (см. рисунок ниже). Вы можете загрузить свои собственные файлы конфигурации для системы, а также сделать резервную копию всех текущих настроек в файл. Почти все параметры конфигурации платформы RSD и любые установленные расширения можно просматривать и изменять с помощью этого интерфейса. Все параметры конфигурации являются частью определенной группы, что позволяет быстро перемещаться по огромному количеству существующих параметров. Также можно выполнить полнотекстовый поиск по всем параметрам.

Рисунок 4.53. Графический интерфейс для настройки системы

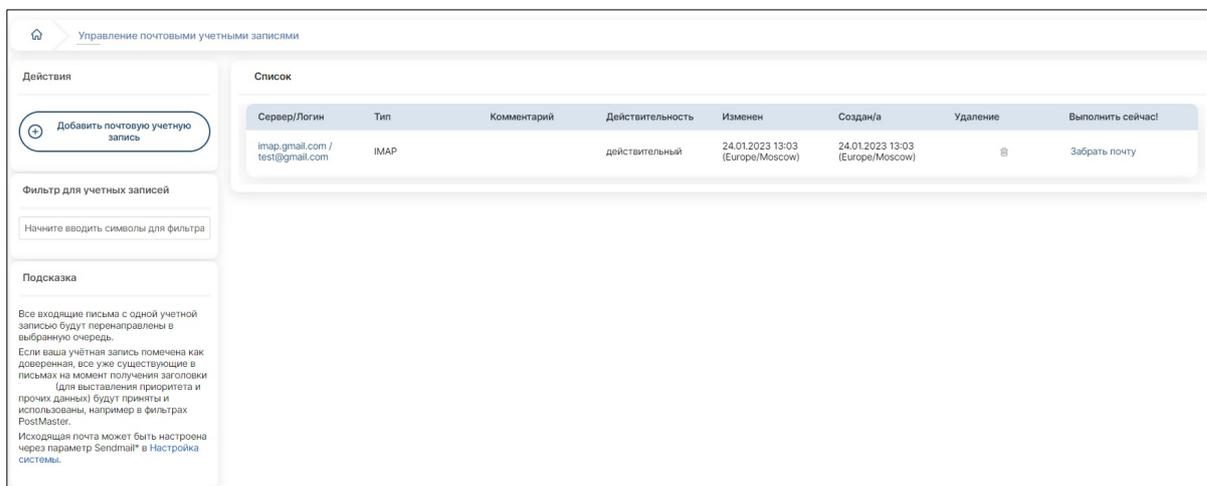


Графический пользовательский интерфейс настройки системы подробно описан в главе «Настройка системы через веб-интерфейс».

1.13. Использование учетных записей почты

Для переноса новых электронных писем в RSD используются почтовые учетные записи, администрирование которых осуществляется через веб-интерфейс. Ссылка "Почтовые учетные записи PostMaster" на странице администратора загружает консоль управления для почтовых учетных записей (см. рисунок ниже). RSD поддерживает почтовые протоколы: POP3, POP3S, IMAP и IMAPS.

Рисунок 4.54. Управление почтовыми ящиками



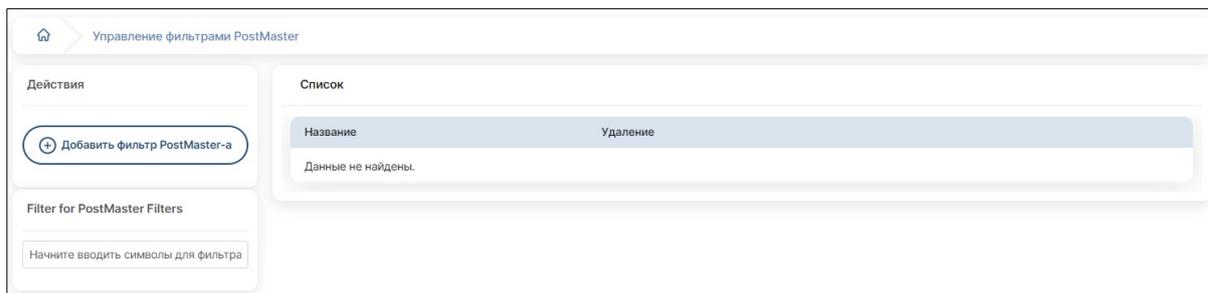
За более подробной информацией обращайтесь к разделу Почтовые аккаунты PostMaster

1.14. Фильтрация входящих почтовых сообщений

RSD имеет возможность фильтровать входящие сообщения электронной почты. Например, можно автоматически помещать определенные электронные письма в указанные очереди или устанавливать определенное состояние или тип заявки для некоторых писем. Фильтры применяются ко всем входящим сообщениям. Для каждого

почтового заголовка можно определить несколько условий фильтрации. Вы можете управлять своими фильтрами через ссылку "Фильтр PostMaster" на странице администрирования (см. рисунок ниже).

Рисунок 4.55. Управление фильтрами Postmaster



Фильтр состоит из одного или нескольких критериев, которые должны быть соблюдены для выполнения определенных действий в электронной почте. Критерии фильтрации могут быть определены для заголовков или тела электронного письма, например, поиск определенных записей заголовка, таких как адрес отправителя, или строк в теле письма. Для расширенного сопоставления с образцом можно использовать даже регулярные выражения. Если ваш фильтр соответствует, вы можете установить новые поля, используя заголовки X-RADIANT в графическом интерфейсе. Эти значения будут применяться при создании заявки или последующего сообщения в RSD. В таблице ниже перечислены различные заголовки X-RADIANT и их значение.

Примечание

Вы также можете использовать заголовки X-RADIANT-FollowUp-* для установки значений для последующих электронных писем.

Таблица 4.5. Функции различных X-RADIANT-заголовков

Название	Возможные значения:	Описание
X-RADIANT-Priority:	<ul style="list-style-type: none"> ● 1 очень низкий ● 2 низкий ● 3 нормальный ● 4 высокий ● 5 очень высокий 	Устанавливает приоритет заявки.

X-RADIANT-Queue:	Название очереди в системе.	Устанавливает очередь, из которой должна быть распределена заявка. Если с этим заголовком установлена очередь, этот параметр имеет приоритет над всеми другими правилами фильтрации, относящимися к очередям. Если вы используете подочередь, укажите ее как Parent::Sub.
X-RADIANT-Lock:	<ul style="list-style-type: none"> • блокировать • разблокировать 	Устанавливает состояние блокировки заявки.
X-RADIANT-Ignore:	Yes или True	Если для этого заголовка X-RADIANT установлено значение «Yes», входящее сообщение будет полностью проигнорировано и никогда не будет доставлено в систему.
X-RADIANT-State:	<ul style="list-style-type: none"> • новая • открыта • закрыта успешно • закрыта неуспешно • и т.д. 	Устанавливает следующее состояние заявки.
X-RADIANT-State-PendingTime:	<p>например, 2010-11-20</p> <p>00:00:00</p>	<p>Задаёт время ожидания для заявки (вы также должны установить состояние ожидания, используя X-RADIANT-State). Можно задать абсолютное значение типа "2010-11-20</p>

		<p>00:00:00" или</p> <p>относительную дату, основанную на времени поступления письма. Используйте формат "+\$Number \$Unit", где \$Unit должно быть 's' (секунды), 'm' (минуты), 'h' (часы) или 'd' (дни). Только одна единица измерения может быть задана. Примеры правильного ввода: "+50s" (отложить на 50 секунд), "+30m" (30 минут), "+12d" (12 дней). Значение типа "+1d 12h" недопустимо. Для такого случая правильным будет "+36h".</p>
X-RADIANT-Type:	по умолчанию (зависит от ваших настроек)	Устанавливает тип заявки (если активировано Ticket::Type).
X-RADIANT-Service:	(зависит от ваших настроек)	Указывает сервис для заявки (если Ticket::Service включен). Если вы желаете задать подсервис сервиса, он указывается в формате "Parent::Sub".
X-RADIANT-SLA:	(зависит от ваших настроек)	Устанавливает SLA для заявки (если активирована поддержка Ticket::Service).
X-RADIANT-CustomerUser:	Пользователи - клиенты	Устанавливает пользователя - клиента для заявки.
X-RADIANT-CustomerNo:	CustomerNo	Устанавливает ID пользователя для заявки.
X-RADIANT-SenderType:	<ul style="list-style-type: none"> ● агент ● система ● клиент 	Устанавливает тип отправителя заявки.

X-RADIANT- IsVisibleForCustomer:	<ul style="list-style-type: none"> • 0 • 1 	Определяет, будет ли отображаться заметка в клиентском интерфейсе.
X-RADIANT- DynamicField-<Dynamic- FieldName>:	<p>В зависимости от конфигурации Динамических Полей (Текст: Notebook,</p> <p>Дата: 2010-11-20 00:00:00,</p> <p>Количество: 1)</p>	Сохраняет дополнительную информацию для заявки в динамическом поле <DynamicFieldName>.
X-RADIANT-Loop:	True	Если этот заголовок X-RADIANT установлен, автоматический ответ отправителю сообщения не доставляется (защита от почтового цикла).
X-RADIANT-FollowUp- State- Keep:	<ul style="list-style-type: none"> • 0 • 1 	Если этот заголовок X-RADIANT установлен, входящее последующее сообщение не изменит состояние заявки. Для этого заголовок можно настроить в SysConfig с помощью настройки KeepStateHeader.
X-RADIANT- AttachmentExists:	<ul style="list-style-type: none"> • no • yes 	В зависимости от того, включены ли вложения в электронное письмо, этот заголовок X-RADIANT имеет значение "да" или имеет значение "нет", если вложения не включены.
X-RADIANT- AttachmentCount:	(количество вложений)	Этот заголовок X-RADIANT содержит в качестве значения количество вложений, содержащихся в электронном письме (т. е. «0» для писем без вложений).

Вы должны указать имя для каждого правила фильтрации. Критерии фильтрации можно указать в разделе "Условие фильтрации". Выберите в полях списка "Заголовок 1", "Заголовок 2" и так далее для частей сообщений, в которых вы хотели бы выполнить поиск, и укажите справа значения, по которым вы хотите выполнить фильтрацию. В

разделе "Настройка заголовков электронной почты" вы можете выбрать действия, которые будут выполняться, если правила фильтрации совпадают. Вы можете выбрать "Заголовок 1", "Заголовок 2" и так далее, чтобы выбрать заголовок X-RADIANT и установить соответствующие значения (см. рисунок ниже).

Правила фильтрации оцениваются в алфавитном порядке, и все они выполняются, за исключением случаев, когда для параметра "Остановка после совпадения" в одном из правил установлено значение "Да" (в этом случае оценка остальных фильтров отменяется).

Рисунок 4.56. Добавление фильтра Postmaster

The screenshot shows the 'Добавить фильтр PostMaster-a' configuration page. It includes a sidebar with 'Действия' (Actions) and 'Подсказка' (Hint) sections. The main form area contains the following fields and sections:

- Название:** ClosingMarkPerfect
- Прекратить проверку после совпадения:** Нет
- Условие фильтра (Условие "И" (AND)):** A list of 10 filter rules, each with a 'Найти поле заголовка:' dropdown, a 'для значения:' input field, and an 'Отрицание ("НЕ"):' checkbox.
- Выставить заголовки письма:** Two entries, each with a 'Выставить заголовок письма:' dropdown and a 'Со значением:' input field.

Пример 4.1. Сортировка нежелательной почты в определенную очередь

Полезным правилом фильтрации было бы позволить RSD автоматически перемещать письма, помеченные как спам, с помощью инструмента обнаружения спама, такого как SpamAssassin, в очередь "Junk". SpamAssassin добавляет заголовок "X-Spam-Flag" к каждому проверенному письму. Когда почта помечена как спам, для заголовка устанавливается значение "Да". Таким образом, критерием фильтрации будет "X-Spam-Flag: Yes". Чтобы создать правило фильтрации с этим критерием, вы можете вставить имя, например, "спам-письма". В разделе "Условие фильтрации" выберите "X-Spam-Flag:" для "Заголовка 1" из списка. Вставьте "Да" в качестве значения для этого заголовка. Теперь заданы критерии фильтрации. Чтобы убедиться, что все спам-сообщения помещаются в очередь "Junk", выберите в разделе "Установить заголовки электронной почты" запись "X-RADIANT-Queue:" для "Заголовка 1". Укажите "Junk" в качестве значения для этого заголовка. Наконец, добавьте новое правило фильтрации, чтобы активировать его для новых сообщений в системе.

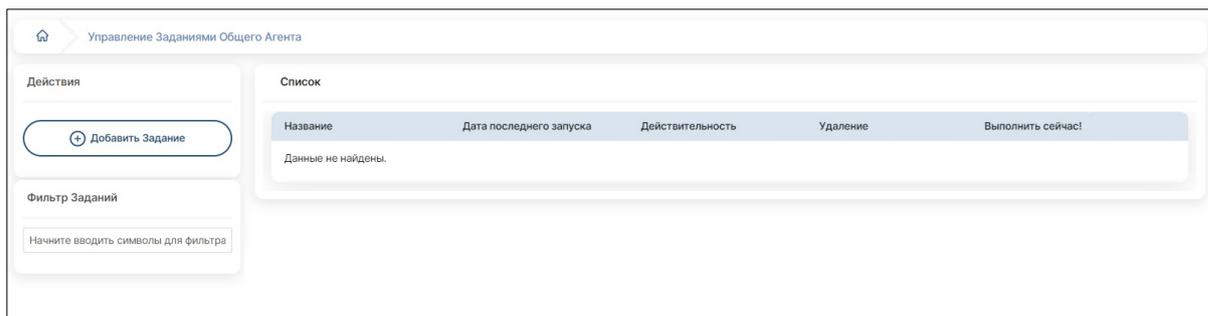
Существуют дополнительные модули, которые можно использовать для более специфической фильтрации входящих сообщений. Эти модули могут быть полезны при работе с более крупными и сложными системами.

1.15. Выполнение автоматизированных заданий с помощью Планировщика задач

Планировщик задач — это инструмент для автоматического выполнения задач. Например, планировщик задач может закрывать или перемещать заявки, отправлять уведомления об эскалированных заявках и т. д.

Щелкните ссылку «Планировщик задач» на странице администрирования (см. рисунок ниже). Отображается таблица со всеми автоматизированными заданиями в системе. Затем эти задания можно редактировать, запускать вручную или полностью удалять

Рисунок 4.57. Список заданий для планировщика задач



Нажмите «Добавить задание», чтобы создать новую задачу. Сначала вам нужно указать имя. Нажмите кнопку «Добавить вакансию», чтобы создать новую вакансию. Затем вы можете указать, как задание будет выполняться: автоматически в фиксированное время (подобно cronjob, этот режим будет работать со всеми заявками, найденными фильтром заявок) или на основе событий заявки (сразу после изменения отдельной заявки, если она соответствует фильтр заявки). Обратите внимание, что, если вы вручную запускаете задания, основанные на событиях, с экрана обзора, они будут работать со всеми заявками, прошедшими фильтр заявок.

Рисунок 4.58. Создание заданий для планировщика задач

Управление Заданиями Общего Агента > Добавить Задание

Действия

Настройки задания

* Имя задания: Пример задания

Действительность: Да

▶ Автоматическое выполнение (несколько заявок)

▶ Управляемое событием выполнение (одна заявка)

▶ Выбрать заявки

▶ Обновить/добавить атрибуты заявки

▶ Добавить заметку/сообщение

▶ Выполнить действия над заявкой

▶ Запустить пользовательский модуль

Сохранить изменения

Сохранить или Отменить

Для каждого задания вы можете указать фильтр заявок, например, для работы только с заявками в определенной очереди. Для выполнения задания по заявке должны быть выполнены все критерии фильтрации.

Наконец, заявку можно изменить, установив новые значения для различных полей заявки, такие как новая очередь или состояние. К заявке (заявкам) можно прикрепить заметку или запустить настроенный модуль. У вас также есть возможность удалить заявку (заявки) из базы данных. Это может быть полезно для удаления устаревших или недействительных данных из системы.

Предупреждение

Если вы воспользуетесь функцией удаления заявки, все затронутые заявки и их вложения будут удалены из базы данных без возможности восстановления!

После редактирования задания, RSD вернет вас на экран обзора заданий. Здесь вы можете запустить любое задание вручную. Если вы решите запустить задание, вы сначала увидите все заявки, которые будут обработаны заданием после старта. Этот список поможет вам убедиться, что фильтры были заданы правильно. В этот момент никаких действий с заявками еще не выполнялось. Задание будет выполнено только после подтверждения вами продолжения работы.

1.16. Административные сообщения

Администраторы RSD могут отправлять сообщения определенным пользователям или группам. Ссылка «Уведомление администратором» на странице администрирования открывает экран, на котором можно выбрать агентов и группы, которые должны быть уведомлены (см. рисунок ниже).

Рисунок 4.59. Экран уведомлений администратора

Сообщение администратора

Подсказка

С помощью этого модуля администраторы могут отправлять сообщения агентам, являющимся членами группы или роли.

Создать сообщение администратора

* Отправитель: support@yourhost.example.com

Отправить сообщение пользователям: Ксения Смирнова X

Отправить сообщение членам группы:

Члены группы должны иметь разрешение: Только чтение Чтение/запись

Отправить сообщение членам роли: Тестирование X

Также отправить клиентам в группах:

* Тема: Тема сообщения администратора

* Тело письма:

Сообщение администратора

Отправить

Можно указать отправителя, тему и основной текст уведомления. Вы также можете выбрать агентов, группы и роли, которые должны получать сообщение.

1.17. Управление сеансами

Вы можете увидеть всех вошедших в систему пользователей и информацию об их сеансах, щелкнув ссылку «Управление сеансами» в области администрирования (см. рисунок ниже).

Рисунок 4.60. Управление сеансами

Управление сеансами

Действия

Все сеансы 4

Сеансы агента 4

Сеансы клиента 0

Уникальные агенты 3

Уникальные клиенты 0

Завершить все сеансы

Фильтр для сеансов

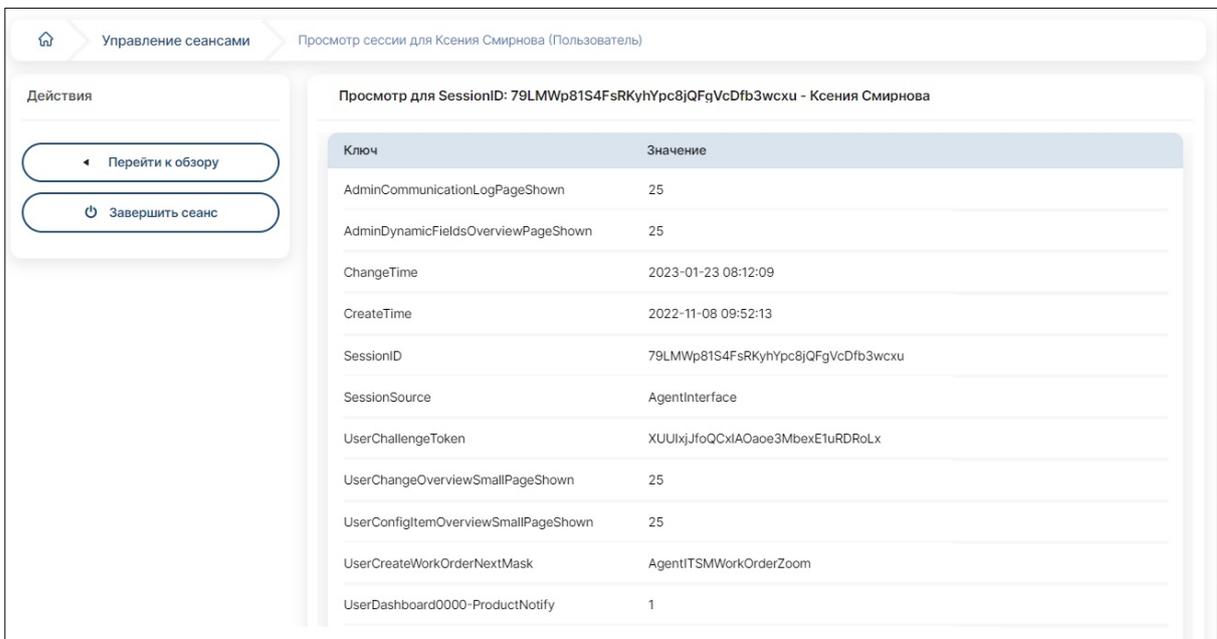
Ничего вводить символы для фильтра

Список

Сеанс	Тип	Пользователь	Завершить
79ISCGANtzGxxvscRP1hoWkuS0ck8ZFIMw	Agent	Олеся Чухлебова	Завершить сеанс
79ILMhp81S4fsRkuYmpc8JQfgydDfo3wscia	Agent	Ксения Смирнова	Завершить сеанс
79cVN67YQp4dCIVhCVGFprxcAeuUOKednS	Agent	Даниил Прусак	Завершить сеанс
79rk8dsXplqRQwKRcSJThHzwuz6tkuwdr	Agent	Олеся Чухлебова	Завершить сеанс

Отображается некоторая статистика обо всех активных сеансах, например, сколько агентов и пользователей-клиентов вошли в систему, а также количество активных сеансов. Любой отдельный сеанс можно удалить, щелкнув ссылку «Завершить сеанс» в правой части списка. У вас также есть возможность «Завершить все сеансы», что может быть полезно, если вы хотите перевести систему в автономный режим. Также доступна подробная информация по каждому сеансу (см. рисунок ниже).

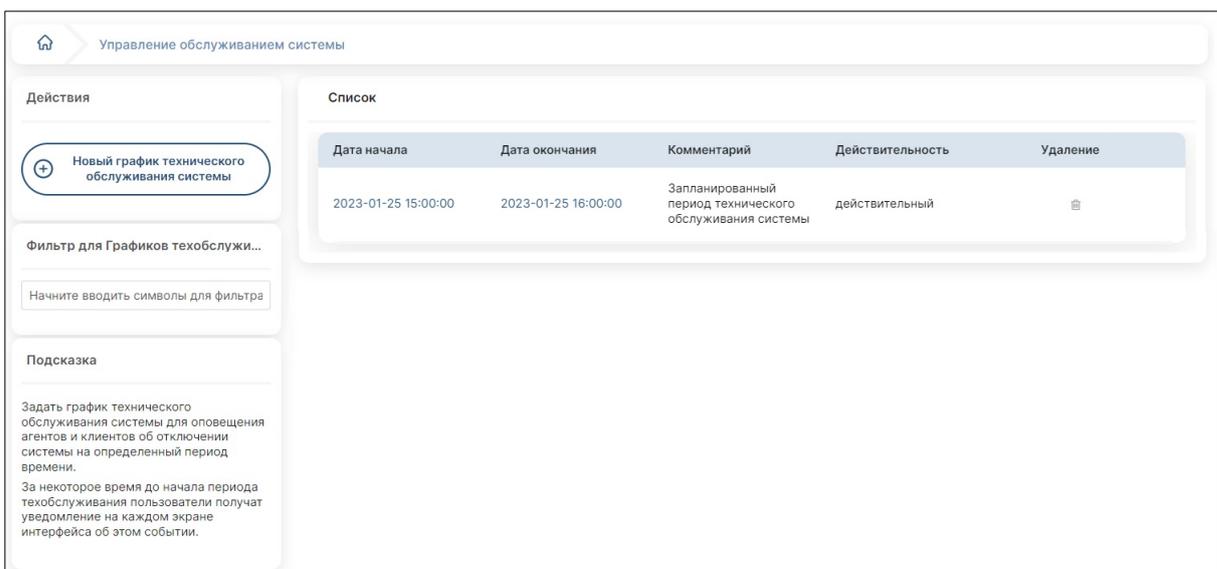
Рисунок 4.61. Подробности сеанса/сессии



1.18. Обслуживание системы

Раздел конфигурации "Обслуживание системы" дает возможность управлять одним или более периодами технического обслуживания системы. В течение этого периода, агенты и клиенты не могут подключиться к системе (за исключением агентов, включенных в группу "admin"). Уже подключенные пользователи получают уведомление о техобслуживании (до и в течение периода техобслуживания). Администраторы имеют возможность закрыть сеансы подключенных агентов и клиентов, это нужно в рамках подготовки, чтобы иметь возможность вносить изменения в систему (например, обновление системы) в "безопасной" среде.

Рисунок 4.62. Обзорный экран обслуживания системы с некоторыми запланированными периодами



Поля «Дата начала» и «Дата окончания» являются обязательными, и единственным правилом для этой комбинации является то, что «Дата начала» не может быть датой, следующей за «Датой окончания».

Рисунок 4.63. Экран редактирования графика обслуживания системы

Управление обслуживанием системы > Новый график технического обслуживания системы

Действия

← Перейти к обзору

Новый график технического обслуживания системы

* Дата начала: 25 . 01 . 2023 - 14 : 46

* Дата окончания: 25 . 01 . 2023 - 14 : 46

* Комментарий: **Запланированный период технического обслуживания системы**

Сообщение при входе: Система временно недоступна.

Показать сообщение при входе:

Сообщение оповещения: Оповещение о запланированный периоде технического обслуживания системы.

* Действительность: действительный

Сохранить или Сохранить и закончить или Отменить

После того, как новый период обслуживания определен, отображается обзор и подробная информация о текущих активных сеансах, откуда администраторы могут закрыть эти сеансы один за другим или все (кроме текущего), если это необходимо.

1.19. Системный журнал

Ссылка «Системный журнал» на странице администрирования показывает записи журнала системы, отсортированные в обратном хронологическом порядке, начиная с самых последних (см. рисунок ниже).

Рисунок 4.64. Системный журнал

Системный журнал

Фильтр для записей журнала

Начните вводить символы для фильтра

Подсказка

Здесь вы найдете логи с информацией о вашей системе.

Скрыть это сообщение

Свежие записи в логе

Время	Приоритет	Объект	Сообщение
Tue Jan 24 17:08:58 2023 (Europe/Moscow)	error	CGI-79	No 'StartActivity' for Process 'Process-554e2b805a3e7730440a811f20e680bc' found!
Tue Jan 24 17:09:57 2023 (Europe/Moscow)	notice	CGI-79	User 'rs_k8' authentication ok. (Method: sha256_REMOTE_AOCS: 102.168.3.1)
Tue Jan 24 17:02:24 2023 (Europe/Moscow)	error	.Console.pl-Maint-PostMaster-MailAccountFetch-79	CommunicationLog(ID:54,AccountType-,AccountID-,Direction:incoming,Transport:Email,ObjectLog => IMAP: Can't connect to imap.gmail.com
Tue Jan 24 16:52:23 2023 (Europe/Moscow)	error	.Console.pl-Maint-PostMaster-MailAccountFetch-79	CommunicationLog(ID:53,AccountType-,AccountID-,Direction:incoming,Transport:Email,ObjectLog => IMAP: Can't connect to imap.gmail.com
Tue Jan 24 16:42:22 2023 (Europe/Moscow)	error	.Console.pl-Maint-PostMaster-MailAccountFetch-79	CommunicationLog(ID:52,AccountType-,AccountID-,Direction:incoming,Transport:Email,ObjectLog => IMAP: Can't connect to imap.gmail.com
Tue Jan 24 16:37:55 2023 (Europe/Moscow)	error	CGI-79	No 'StartActivity' for Process 'Process-554e2b805a3e7730440a811f20e680bc' found!
Tue Jan 24 16:35:51 2023 (Europe/Moscow)	error	CGI-79	No State for pending reminder found!
Tue Jan 24 16:35:51 2023 (Europe/Moscow)	error	CGI-79	No State for open found!

Каждая строка в журнале содержит отметку времени, приоритет журнала, системный компонент и саму запись в журнале.

Примечание

Системные журналы доступны через веб-интерфейс только в системах Linux/Unix.

1.20. Журнал сеансов связи

Интерфейс журнала сеансов связи в области администрирования позволяет вам просматривать внутренние журналы обработки сообщений в RSD, т.е. отправлять /получать электронные письма.

Рисунок 4.65. Обзор журнал сеансов связи

Интервал времени

Последние 24 часа

Показать записи журнала связи созданные в определенном промежутке времени.

Фильтр по сеансам связи

Начните вводить символы для фильтрации

Подсказка

На этом экране вы можете видеть обзор входящих и исходящих сеансах связи. Чтобы изменить сортировку или порядок столбцов, щёлкните мышью по заголовку столбцов. Кликая по различным записям, вы будете перенаправляться на экран подробного просмотра о сообщении.

Состояние: Последние 24 часа

Нет проблем с учетными записями

Нет проблем с сеансами связи

Нет активных сеансов связи

Среднее время сеанса
Меньше секунды

▼ Список сеансов связи (145)

Вс (145) Успешно (145) Выполнение (0) Не удалось выполнить (0)

СОСТОЯНИЕ	ПЕРЕДАЧА	АККАУНТ	▼ НАЧАЛО	ОКОНЧАНИЕ	ПРОДОЛЖИТЕЛЬНОСТЬ
✓	Email	mail.your-server.de / ks@otrsdemo.com (IMAP)	19.09.2022 13:30:12	19.09.2022 13:30:12	0 s
✓	Email	mail.your-server.de / ks@otrsdemo.com (IMAP)	19.09.2022 13:20:11	19.09.2022 13:20:11	0 s
✓	Email	mail.your-server.de / ks@otrsdemo.com (IMAP)	19.09.2022 13:10:09	19.09.2022 13:10:09	0 s

1-25 из 145 - Страница: 1 2 3 4 5 >> >|

Обзорная страница — это экран, похожий на информационную панель, с несколькими показателями, указывающими на общее состояние системы в зависимости от отфильтрованных сообщений.

- Статус учетной записи будет сигнализировать, если у вас есть какие-либо проблемы с настроенными учетными записями, используемыми для получения или отправки сообщений.
- Статус связи сообщит вам, если возникнут какие-либо ошибки либо с подключением к учетной записи, либо с обработкой сообщений.
- Виджет состояния связи будет отображаться, если в настоящее время в системе есть какие-либо активные соединения.
- Среднее время обработки — это кумулятивная статистика времени, необходимая для завершения связи.

Вы можете выбрать временной диапазон на левой боковой панели, чтобы фильтровать сообщения в зависимости от времени их создания. В дополнение к этому вы также можете динамически фильтровать любые ключевые слова, состояние связи и сортировать обзорную таблицу по всем столбцам.

Рисунок 4.66. Выбор временного диапазона журнала сеансов связи

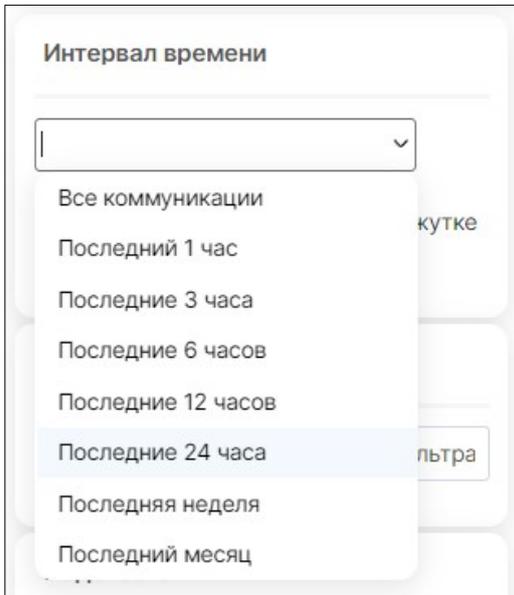


Рисунок 4.67. Статус учетной записи журнала сеансов связи

Журнал сеансов связи > Статус учетной записи

Действия

Назад в обзорный список

Интервал времени

Последние 24 часа

Показать записи журнала связи созданные в определенном промежутке времени.

Фильтр по учетным записям

Начните вводить символы для фильтрации

Фильтр по сеансам связи

Начните вводить символы для фильтрации

Подсказка

На этом экране вы можете видеть обзор входящих и исходящих сеансов связи. Вы можете поменять сортировку и порядок колонок, кликнув по заголовку. Кликая по различным записям, вы будете перенаправляться на экран подробного просмотра о сообщении.

Статус учетной записи для: Последние 24 часа

▲ СОСТОЯНИЕ	АККАУНТ	СРЕДНЕЕ ВРЕМЯ СЕАНСА	РЕДАКТИРОВАТЬ
✓	mail.your-server.de / ks@otrsdemo.com (IMAP)	Меньше секунды	
✓	SMTP	1 минута	

▼ Подробный просмотр записи журнала сеансов связи для: (25) 1-25 из 25

СОСТОЯНИЕ	ПЕРЕДАЧА	⇄	НАЧАЛО	ОКОНЧАНИЕ	ПРОДОЛЖИТЕЛЬНОСТЬ
✓	Email	←	19.09.2022 13:50:15	19.09.2022 13:50:15	0 s
✓	Email	←	19.09.2022 13:40:13	19.09.2022 13:40:13	0 s
✓	Email	←	19.09.2022 13:30:12	19.09.2022 13:30:12	0 s
✓	Email	←	19.09.2022 13:20:11	19.09.2022 13:20:11	0 s
✓	Email	←	19.09.2022 13:10:09	19.09.2022 13:10:09	0 s
✓	Email	←	19.09.2022 13:00:08	19.09.2022 13:00:08	0 s
✓	Email	←	19.09.2022 12:50:06	19.09.2022 12:50:06	0 s
✓	Email	←	19.09.2022 12:40:05	19.09.2022 12:40:05	0 s
✓	Email	←	19.09.2022 12:30:03	19.09.2022 12:30:04	1 s
✓	Email	←	19.09.2022 12:20:02	19.09.2022 12:20:03	1 s
✓	Email	←	19.09.2022 12:10:01	19.09.2022 12:10:01	0 s
✓	Email	←	19.09.2022 12:00:19	19.09.2022 12:00:19	0 s
✓	Email	←	19.09.2022 11:50:18	19.09.2022 11:50:18	0 s
✓	Email	←	19.09.2022 11:40:16	19.09.2022 11:40:16	0 s
✓	Email	←	19.09.2022 11:30:15	19.09.2022 11:30:15	0 s
✓	Email	←	19.09.2022 11:20:14	19.09.2022 11:20:14	0 s
✓	Email	←	19.09.2022 11:10:12	19.09.2022 11:10:12	0 s
✓	Email	←	19.09.2022 11:00:11	19.09.2022 11:00:11	0 s

Нажатие на виджет учетной записи приведет вас к экрану состояния учетной записи, где вы можете дополнительно отфильтровать сообщения, связанные с конкретными учетными записями.

Если вы нажмете на строку сообщения в любой таблице, вам будет представлен экран подробного просмотра.

Рисунок 4.68. Подробный просмотр журнала сеансов связи

Действия

Назад в обзорный список

Фильтр для записей журнала

Начните вводить символы для фильтра

Показывать записи только с указанным приоритетом и выше:

Трассировка

Подсказка

На этом экране вы можете видеть обзор входящих и исходящих сеансов связи. Вы можете поменять сортировку и порядок колонок, кликнув по заголовку. Кликнув по различному записям, вы будете перенаправлены на экран подробного просмотра о сообщении.

Подобный просмотр для outgoing сеанс начал в 26.01.2023 16:00:49

Просмотр журнала сеансов связи (2)

Состояние	Тип	Аккаунт	Время начала	Время окончания	Продолжительность
✓ Connection	→ SMTPTLS		26.01.2023 16:01:12	26.01.2023 16:01:12	0 s
✓ Message	→ SMTPTLS		26.01.2023 16:00:49	26.01.2023 16:01:12	23 s

Подробный просмотр записи журнала сеансов связи для: (3)

№	Приоритет	Модуль	Сведения	Создан(а)
1	🚩	Kernel-System:Email-SMTP	Testing connection to SMTP service (3 attempts max).	26.01.2023 16:01:12
2	🚩	Kernel-System:Email-SMTP	1: Trying to connect to 'mail.your-server.de:587' on visual.otrademo.com with SMTP type 'SMTPTLS'.	26.01.2023 16:01:12
3	🚩	Kernel-System:Email-SMTP	Using SMTP authentication with user 'ks@otrademo.com' and (hidden) password.	26.01.2023 16:01:12

Каждое сообщение может содержать один или несколько журналов, которые могут иметь тип соединения или сообщения.

- Журналы подключений будут содержать все сообщения журнала, поступающие от модулей, отвечающих за подключение к вашим учетным записям и получение/получение сообщений.
- Журналы сообщений будут содержать любые сообщения журнала, связанные с обработкой конкретных сообщений. Любой модуль, работающий с сообщением, может регистрировать свои действия в этом журнале, что дает вам четкое представление о том, что происходит.

Вы можете фильтровать записи журнала в зависимости от их приоритета, выбрав нужный приоритет на левой боковой панели.

Рисунок 4.69. Фильтр приоритета журнала сеансов связи

Действия

Назад в обзорный список

Трассировка

Отладка

Информация

Замечание

Предупреждение

Ошибка

Фильтра

ИНЫМ

Применяются правила уровня журнала: выбрав определенный приоритет, вы получите записи журнала с этим установленным приоритетом и выше, причем Ошибка будет самым высоким.

1.21. SQL-запросы

Ссылка «SQL-запросы» на странице администрирования открывает экран, позволяющий запрашивать содержимое таблиц в базе данных RSD (см. рисунок ниже). Невозможно изменить содержимое таблиц, разрешены только запросы «select».

Рисунок 4.70. SQL-запросы

The screenshot shows a web interface for SQL queries. On the left, there is a sidebar with a home icon and the title 'SQL-запросы'. Below the title is a 'Подсказка' (Hint) section with text: 'Здесь вы можете ввести SQL-запрос и напрямую отправить его в базу данных приложения. Невозможно изменение данных в таблицах, только выборка.' The main area is titled 'Настройки' (Settings) and contains a form with the following fields: '* SQL:' with a text input containing 'SELECT * FROM'; 'Ограничение:' (Limit) with a numeric input containing '40'; 'Формат вывода:' (Output format) with a dropdown menu set to 'HTML'; and a 'Выполнить запрос' (Execute query) button.

1.22. Менеджер пакетов

С помощью ссылки «Менеджер пакетов» на странице «Администрирование» вы можете устанавливать и управлять пакетами, расширяющими функциональность RSD (см. рисунок ниже). Обсуждение расширений, доступных из репозитория RSD, см. в разделе Дополнительные приложения.

Рисунок 4.71. Менеджер пакетов

The screenshot shows the 'Менеджер пакетов' (Package Manager) interface. On the left, there is a sidebar with 'Действия' (Actions) and buttons: 'Выбор файла | файл не выбран', 'Установить пакет', 'Bundle Manager', 'Обновить информацию о репозитории', and 'Обновить все установленные пакеты'. The main area is divided into two sections: 'Онлайн репозиторий' (Online repository) and 'Локальный репозиторий' (Local repository). Both sections show a table of packages with columns: 'Название' (Name), 'Версия' (Version), 'Издатель' (Publisher), 'Описание' (Description), 'Состояние' (Status), and 'Действие' (Action).

Название	Версия	Издатель	Описание	Состояние	Действие
ModernSkin	6.0.5	Radiant System	A modern and flat agent skin by Efflux	установлено	Деинсталлировать/Переустановить
RS_AdminUserLog	6.2.2	Radiant System	Admin User Log and View	установлено	Деинсталлировать
RS_AdvancedTicketSearch	6.2.1	Radiant System	Расширенный поиск заявок	установлено	Деинсталлировать/Переустановить
RS_AdvancedTicketStates	6.1.32	Radiant System	Расширение состояний заявок	установлено	Деинсталлировать
RS_Macros	6.4.10	Radiant System	Макросы	установлено	Деинсталлировать/Переустановить
RS_NotificationEventMod	6.2.17	Radiant System	NotificationEventMod	установлено	Деинсталлировать
RS_RadiantSystemSkin	6.0.2	Radiant System	Radiant System Skin	установлено	Деинсталлировать/Переустановить
RS_TicketViews	6.3.3	Radiant System	Виды	установлено	Деинсталлировать/Переустановить
RS_TimeManager	6.1.4	Radiant System	Time Manager	установлено	Деинсталлировать/Переустановить
RS_XLS_Bundle	6.0.29	Radiant System	Сохранение списка тикетов в XLS + Добавляет XLS-форм...	установлено	Деинсталлировать/Переустановить

Менеджер пакетов показывает пакеты дополнений RSD, которые вы в настоящее время установили на свой сервер, вместе с номерами их версий.

Вы можете установить пакеты с удаленного хоста, выбрав репозиторий в разделе «Онлайн-репозиторий» и нажав кнопку «Обновить информацию о репозитории». Доступные пакеты отобразятся в соответствующей таблице. В правой части экрана показаны доступные пакеты. Чтобы установить пакет, нажмите «Установить». После установки пакет отображается в разделе «Локальный репозиторий».

Чтобы обновить установленный пакет, в списке доступных пакетов в онлайн-репозитории будет указано «Обновить» в столбце «Действие» для любого пакета,

версия которого выше версии, установленной локально. Просто нажмите «Обновить», и новая версия пакета будет установлена в вашей системе.

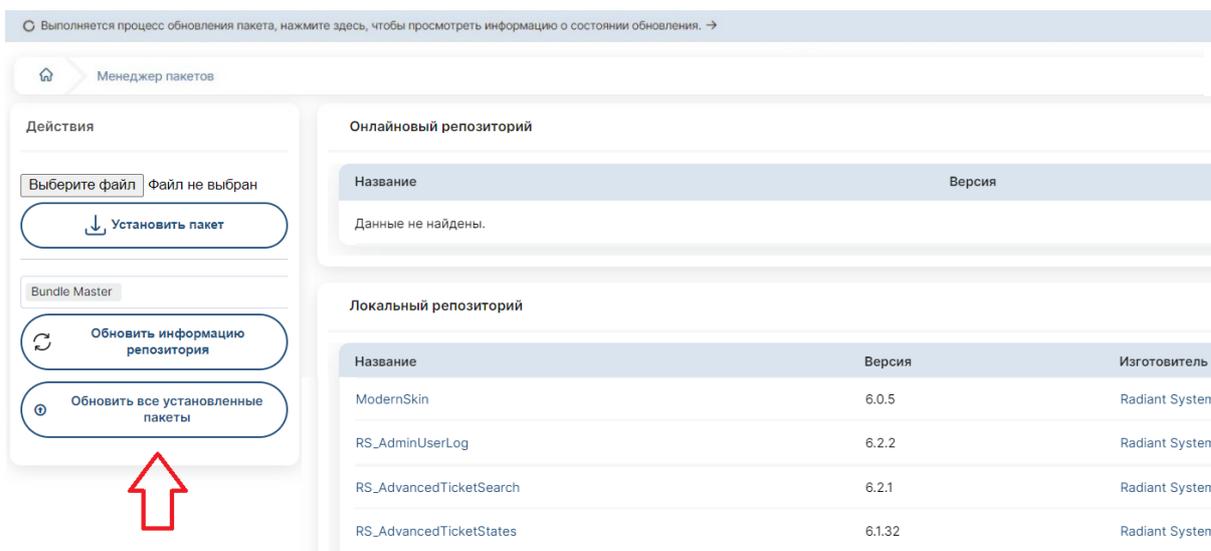
В некоторых случаях, например, когда ваша система RSD не подключена к Интернету, вы также можете установить те пакеты, которые вы загрузили на локальный диск. Нажмите кнопку «Обзор» на боковой панели «Действия» и выберите файл .orm пакета на своем диске. Нажмите «Открыть», а затем «Установить пакет». После завершения установки пакет отображается в разделе «Локальный репозиторий». Вы можете использовать те же шаги для обновления уже установленного пакета.

В особых случаях может потребоваться настроить Менеджер пакетов, например, для использования прокси-сервера или локального репозитория. Просто взгляните на доступные параметры в SysConfig в разделе Core::Package.

Обновление всех установленных пакетов

Начиная с RSD 6, на боковую панель экрана менеджера пакетов была добавлена новая кнопка с надписью «Обновить все установленные пакеты». После нажатия этой кнопки задача RSD Daemon будет запланирована для обновления всех пакетов. Кроме того, это действие также можно выполнить из интерфейса командной строки с помощью консольной команды Admin::Package::UpgradeAll.

Рисунок 4.72. Кнопка «Обновить все установленные пакеты» в менеджере пакетов



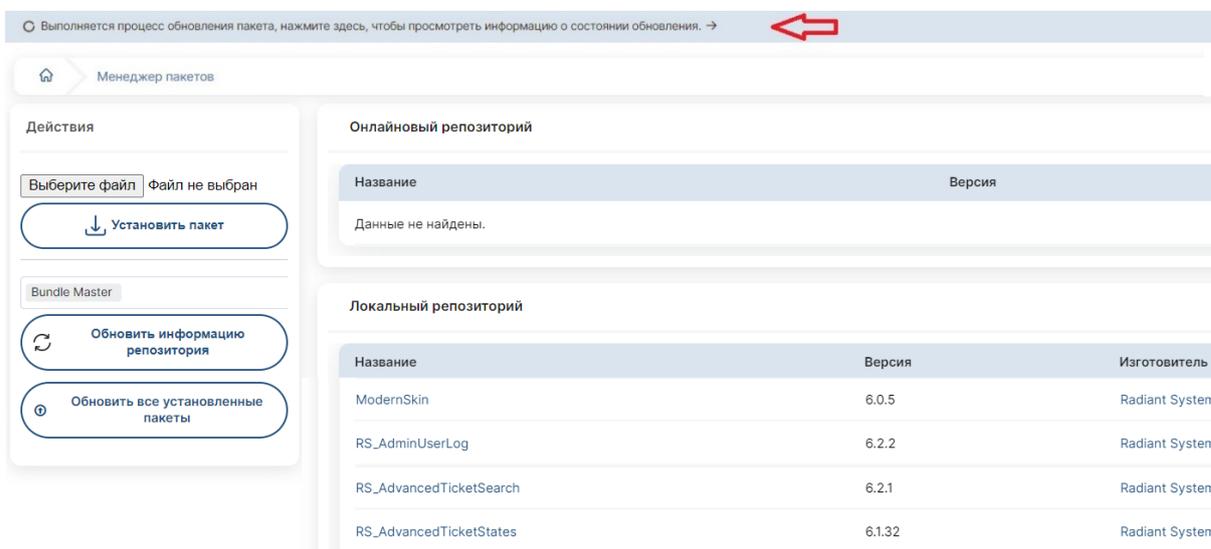
Эта функция считывает информацию обо всех определенных репозиториях пакетов и определяет, есть ли новая версия для каждого установленного пакета в системе, и вычисляет правильный порядок обновления пакетов с учетом всех других зависимостей пакетов, даже если для новых версий существующих пакетов требуются новые пакеты, еще не установленные в системе.

Примечание

Если установлены пакеты, для которых в системе не определен соответствующий репозиторий, они не могут быть обновлены этой функцией и будут помечены как неудавшиеся (из-за отсутствия онлайн-репозитория).

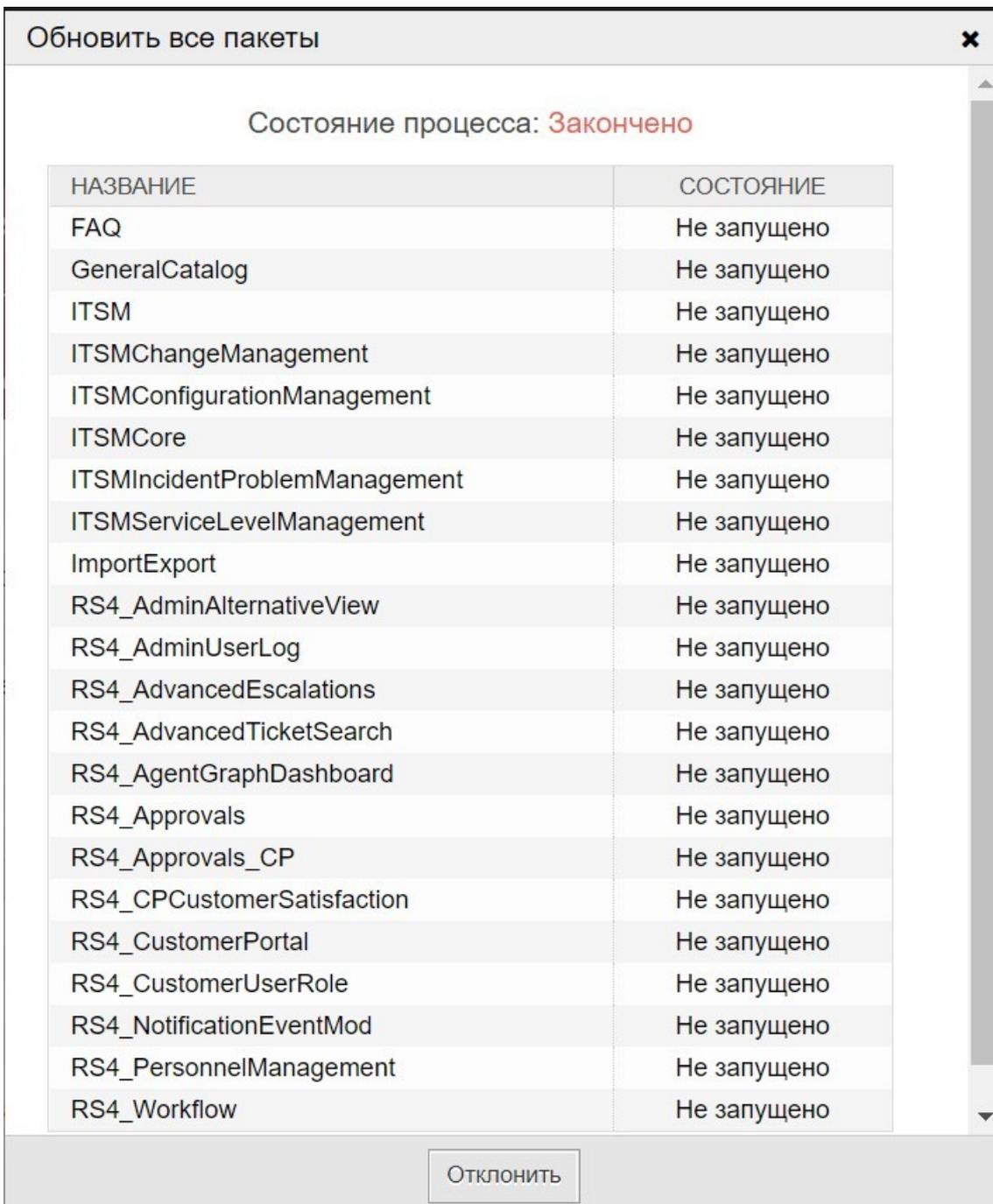
Когда процесс обновления всех пакетов запущен, на экран менеджера пакетов добавляется панель уведомлений для облегчения идентификации. При нажатии на ссылку уведомления отображается сводка хода выполнения с текущим ходом выполнения. Кроме того, когда процесс выполняет действия пакета, такие как «Установить», «Обновить», «Удалить» и «Переустановить», некоторые действия на экране могут быть временно отключены во избежание конфликтов.

Рисунок 4.73. Менеджер пакетов → Обновление всех установленных пакетов → Выполняется уведомление



Когда процесс завершится (только если он был запущен в графическом пользовательском интерфейсе), уведомление о ходе выполнения изменится, чтобы указать новое состояние: оно станет зеленым, если все пакеты были правильно обновлены, или красным, если были какие-либо сбои. Нажав на это уведомление, вы увидите окончательную сводку. В сводной таблице всплывающая подсказка для столбца состояния каждого пакета может указать дополнительные сведения о состоянии. После закрытия этого окна экран менеджера пакетов обновится, чтобы отобразить новую версию установленных пакетов.

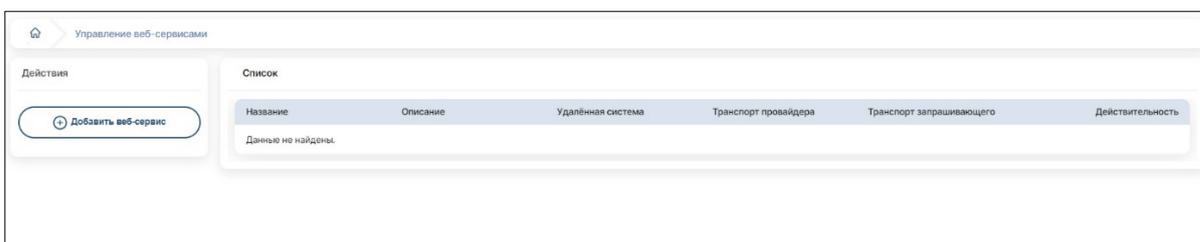
Рисунок 4.74. Менеджер пакетов → Обновить все установленные пакеты → Сводка



1.23. Веб-сервисы

Ссылка «Веб-сервисы» ведет к графическому интерфейсу, в котором создаются и поддерживаются веб-сервисы (для универсального интерфейса RSD) (см. рисунок ниже).

Рисунок 4.75. Графический интерфейс для веб-сервисов



Графический интерфейс для настройки веб-сервисов более подробно описан в разделе «Графический интерфейс веб-сервиса».

1.24. Динамические поля

Раздел «Динамические поля» позволяет создавать, настраивать и управлять пользовательскими полями для заявок и сообщений/заметок (см. рисунок ниже).

Рисунок 4.76. Обзорный экран динамических полей с некоторыми динамическими полями

Имя	Название	Порядок	Тип	Объект	Активность	Удаление
ProcessManagementProcessID	Процесс	1	ProcessID	Заявка	действительный	
ProcessManagementActivityID	Активность	2	ActivityID	Заявка	действительный	
ITSMCriticality	Критичность	3	Выпадающий список	Заявка	действительный	
ITSMImpact	Влияние	4	Выпадающий список	Заявка	действительный	
ITSMReviewRequired	Необходим просмотр	5	Выпадающий список	Заявка	действительный	
ITSMDecisionResult	Результат решения	6	Выпадающий список	Заявка	действительный	
ITSMRepairStartTime	Дата начала работ	7	Дата/Время	Заявка	действительный	
ITSMRecoveryStartTime	Дата восстановления сервиса	8	Дата/Время	Заявка	действительный	
ITSMDecisionDate	Дата решения	9	Дата/Время	Заявка	действительный	
ITSMDueDate	Дата исполнения	10	Дата/Время	Заявка	действительный	
CalendarStart	CalendarStart	11	Дата/Время	Заявка	действительный	
CalendarFinish	CalendarFinish	12	Дата/Время	Заявка	действительный	
ClosingMark	ClosingMark	13	Выпадающий список	Заявка	действительный	

Более подробное описание настройки динамических полей содержится в разделе "Настройка Динамических полей".

Каждый тип динамического поля имеет свои собственные настройки конфигурации и, следовательно, свой собственный экран конфигурации.

Примечание

Во фреймворке RSD динамические поля по умолчанию могут быть связаны только с заявками и заметками, но их можно распространить и на другие объекты.

1.25. Календари

Через интерфейс управления календарем можно добавлять и редактировать несколько календарей.

2. Настройка Системы

2.1. Конфигурационные файлы RSD

Все файлы конфигурации RSD хранятся в каталоге Kernel и его подкаталогах. Нет необходимости вручную изменять какой-либо другой файл, кроме Kernel/Config.pm, поскольку остальные файлы будут изменены при обновлении системы. Просто скопируйте параметры конфигурации из других файлов в Kernel/Config.pm и измените их в соответствии с вашими потребностями. Этот файл никогда не будет затронут в процессе обновления, поэтому ваши ручные изменения в безопасности.

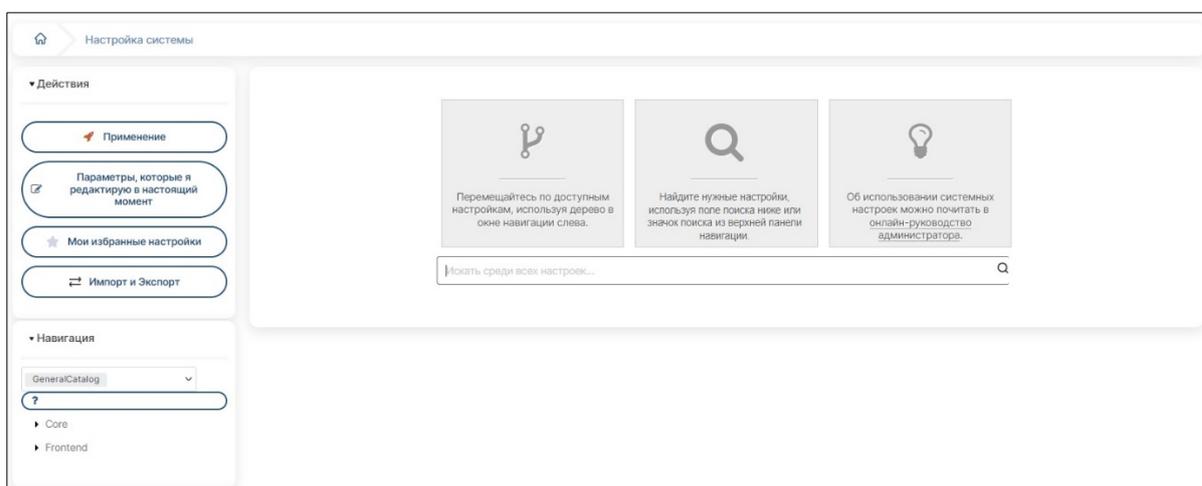
Конфигурация может быть перестроена либо вручную, выполнив команду `perl bin/radiant.Console.pl Maint::Config::Rebuild`, либо автоматически программой установки, операциями с пакетами (установка, обновление, переустановка и удаление) и т. д. Все `.xml`, расположенные в `Kernel/Config/Files/XML/`, анализируются, и их содержимое передается в базу данных RSD. Большинство файлов `.xml`, которые являются частью этого каталога, по умолчанию принадлежат RSD Free, но бесплатные функции RSD, такие как FAQ или Survey (или любое другое расширение RSD), также добавляют свои собственные файлы конфигурации в этот каталог.

Чтобы применить какие-либо изменения к конфигурации, необходимо их развернуть. Процесс развертывания извлечет все настройки и соответствующие значения из базы данных RSD, преобразует их в формат, понятный RSD, и поместит их в `Kernel/Config/Files/ZZZAAuto.pm`.

При обращении к веб-интерфейсу RSD настройки, содержащиеся в файлах `Kernel/Config/Files/ZZZAAuto.pm`, `Kernel/Config/Files/ZZZACL.pm` (если есть), `Kernel/Config/Files/ZZZProcessManagement.pm` (если есть) и любой другой пользовательский файл `.pm` будут оцениваться в алфавитном порядке. Эти файлы используются графическим веб-интерфейсом конфигурации системы и никогда не должны редактироваться вручную. Наконец, будет проанализирован файл `Kernel/Config.pm`, содержащий ваши индивидуальные настройки и измененные вручную параметры конфигурации. Чтение конфигурационных файлов в таком порядке гарантирует, что система правильно воспримет ваши конкретные настройки конфигурации.

2.2. Настройка системы через веб-интерфейс

Рисунок 4.78. Графический интерфейс для настройки системы



В настоящее время RSD Free содержит более 1700 параметров конфигурации, и существуют различные способы быстрого доступа к определенному параметру. При полнотекстовом поиске все параметры конфигурации можно просмотреть по одному или нескольким ключевым словам. Полнотекстовый поиск выполняет поиск не только по именам параметров конфигурации, но также по описаниям и значениям. Это позволяет легко найти элемент, даже если его имя неизвестно.

Кроме того, каждый параметр конфигурации классифицируется по категории и группе навигации. Группы навигации — это отдельные элементы в главном дереве навигации.

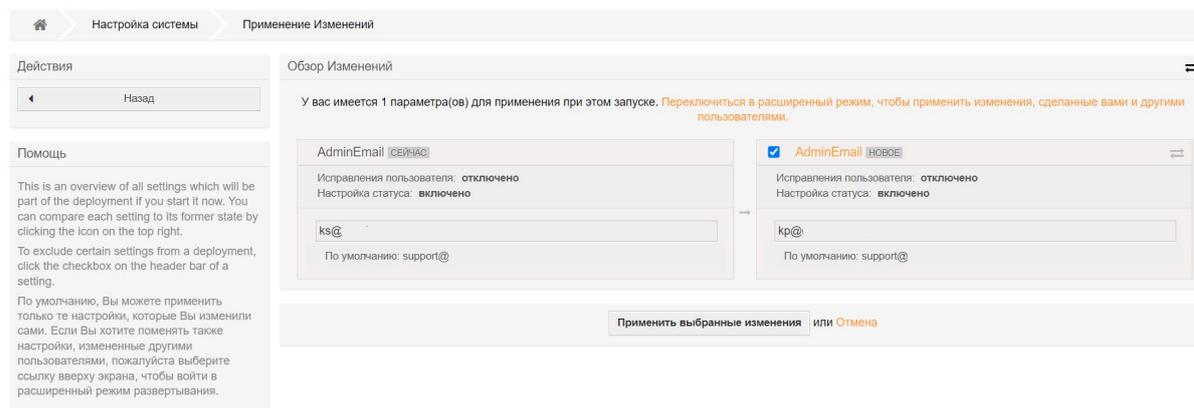
При выборе одного из этих элементов навигации будут показаны все настройки, назначенные выбранной группе. Пока расширения не установлены, выбор категории автоматически скрыт, но как только установлен пакет с собственными настройками конфигурации (например, «База знаний»), выбор категории будет показан. При выборе категории в главном дереве навигации отображаются только группы навигации, принадлежащие к выбранной категории.

По умолчанию все параметры конфигурации отображаются только для чтения. Чтобы изменить параметр, нажмите кнопку «Изменить этот параметр», которая появляется при наведении курсора мыши на параметр. После входа в режим редактирования изменения, внесенные в поля в настройках, можно сохранить, нажав кнопку «Сохранить» справа (или нажав клавишу «Ввод» на клавиатуре). Если параметр в настоящее время редактируется другим администратором, получить доступ к режиму редактирования для этого параметра невозможно, пока другой администратор не завершит свою работу. Редактирование можно отменить, нажав кнопку «Отмена» справа или нажав клавишу Escape на клавиатуре. При отмене редактирования все изменения, сделанные во время текущего сеанса редактирования, отменяются.

Щелкнув на имя настройки или значок меню в правом верхнем углу каждой настройки, вы откроете меню с некоторыми дополнительными параметрами (например, включение / выключение настройки, сброс значений настройки до значений по умолчанию, копирование прямой ссылки на настройку или добавление настройки в список избранных для быстрого доступа).

Чтобы изменения вступили в силу, необходимо запустить развертывание. На экране развертывания конфигурации системы отображается список параметров, которые должны быть развернуты. Если щелкнуть значок стрелки в левом верхнем углу каждого параметра или во внешнем окне Обзор изменений, отобразится сравнение «один к одному» между текущим состоянием настроек (которое будет развернуто) и предыдущим состоянием (или состоянием по умолчанию).

Рисунок 4.79. Развертывание конфигурации системы



Если вы хотите сохранить все изменения, внесенные вами в конфигурацию вашей системы (например, чтобы быстро настроить новую установку), вы можете использовать кнопку «Экспорт настроек», которая предоставит файл .yml для загрузки. Чтобы восстановить собственные настройки, используйте настройки импорта и выберите ранее созданный файл .yml.

Примечание

Из соображений безопасности параметры конфигурации для подключения к базе данных не могут быть изменены в графическом пользовательском интерфейсе конфигурации системы. Их необходимо установить вручную в Kernel/Config.pm.

3. Резервное копирование системы

В этой главе описывается резервное копирование и восстановление данных RSD.

3.1. Резервное копирование

Существует два типа данных для резервного копирования: файлы приложений (например, файлы в /opt/radiant) и данные, хранящиеся в базе данных.

Для упрощения создания резервных копий, сценарий scripts/backup.pl включен в каждую инсталляцию RSD. Он позволяет создавать резервную копию всех важных данных (см. Сценарий ниже).

```
linux:/opt/radiant# cd scripts/ linux:/opt/radiant/scripts# ./backup.pl --help backup.pl -
backup script

Copyright (C) 2001-2022 RSD, https://radiantsystem.ru/

usage: backup.pl -d /data_backup_dir/ [-c gzip|bzip2] [-r 30] [-t fullbackup|nofullbackup|
dbonly]

linux:/opt/radiant/scripts#
```

Сценарий: Получение справки о механизме резервного копирования RSD.

Выполните команду, указанную в приведенном ниже скрипте, чтобы создать резервную копию:

```
linux:/opt/radiant/scripts# ./backup.pl -d /backup/ Backup /backup//2010-09-07_14-
28/Config.tar.gz ... done

Backup /backup//2010-09-07_14-28/Application.tar.gz ... done Dump MySQL rdbms
... done

Compress SQL-file... done linux:/opt/radiant/scripts#
```

Сценарий: Создание резервной копии.

Все данные были сохранены в каталоге /backup/2010-09-07_14-28/ (см. Сценарий ниже). Дополнительно данные были сохранены в .tar.gz файл.

```
linux:/opt/radiant/scripts# ls /backup/2010-09-07_14-28/ Application.tar.gz
Config.tar.gz DatabaseBackup.sql.gz linux:/opt/radiant/scripts#
```

Сценарий: Проверка файлов резервных копий.

3.2. Восстановление

Для восстановления резервной копии, сохраненные данные приложения должны быть записаны обратно в каталог установки, например, /opt/radiant. База данных также должна быть восстановлена.

Сценарий scripts/restore.pl (см. Сценарий ниже), который упрощает процесс восстановления, поставляется с каждой установкой RSD. Он поддерживает MySQL и PostgreSQL.

```
linux:/opt/radiant/scripts# ./restore.pl --help restore.pl - restore script

Copyright (C) 2001-2022 RSD, https://radiant.com/ usage: restore.pl -b
/data_backup/<TIME>/ -d /opt/radiant/ linux:/opt/radiant/scripts#
```

Сценарий: Получение справки о механизме восстановления.

Данные, которые хранятся, например, в каталоге /backup/2010-09-07_14-28/, могут быть восстановлены с помощью команды, указанной в приведенном ниже скрипте, при условии, что RSD установлен в /opt/radiant.

```
linux:/opt/radiant/scripts# ./restore.pl -b /backup/2010-09-07_14-28 -d /opt/radiant/
Restore /backup/2010-09-07_14-28//Config.tar.gz ...

Restore /backup/2010-09-07_14-28//Application.tar.gz ... create MySQL
decompresses SQL-file ...

cat SQL-file into MySQL database compress SQL-file... linux:/opt/radiant/scripts#
```

4. Настройки электронной почты

4.1. Отправка/получение электронных писем

Отправка писем

1. Через Sendmail (по умолчанию)

RSD может отправлять электронные письма через [Sendmail](#), [Postfix](#), [Qmail](#) или [Exim](#). Конфигурация по умолчанию предполагает использование Sendmail и должна работать сразу после установки.

Вы можете настроить параметры sendmail через графический интерфейс конфигурации (Framework::Core::Sendmail).

2. Через SMTP-сервер или Smarthost

RSD может отправлять электронные письма через SMTP ([Simple Mail Transfer Protocol / RFC 821](#)) или безопасный SMTP.

Параметры SMTP-сервера можно настроить с помощью SysConfig (Framework::Core::Send-mail). Если вы не видите SMTPS в настройке, значит необходимые модули Perl отсутствуют. В этом случае обратитесь за инструкциями к разделу «Установка модулей Perl, необходимых для RSD».

3. Почтовая очередь

Перед отправкой RSD поставит все электронные письма в очередь и позволит задаче Демон асинхронно обработать их в нужное время. При обычной установке электронные письма будут отправляться демоном RSD каждую минуту.

Чтобы проверить статус отправки электронной почты, вы можете использовать консольную команду `bin/radiant.Console.pl Maint::E-mail::MailQueue` с ключом `--list`.

```
shell> cd /opt/radiant/

shell> bin/radiant.Console.pl Maint::Email::MailQueue --list

+---+-----+-----+-----+-----+-----+
+-----+-----+
| ID | ArticleID | Attempts | Sender Code | Last SMTP Message | Recipient | Due Time | Last SMTP
+---+-----+-----+-----+-----+-----+-----+
+-----+-----+
| 28 | 284 | 0 | radiant@localhost | marketing@aaa.ru | - | - | - |
| 29 | 285 | 0 | radiant@localhost | feedbac@aaa.ru | - | - | - |
+---+-----+-----+-----+-----+-----+
+-----+-----+

Mail queue contains 2 message(s)
```

Чтобы отфильтровать таблицу, вы можете использовать аргумент `--filter`, например, с идентификатором задания отправки.

```

shell> bin/radiant.Console.pl Maint::Email::MailQueue --list --filter ID::28

+---+-----+-----+-----+-----+-----+
+-----+-----+
| ID | ArticleID | Attempts | Sender Code | Last SMTP Message | Recipient | Due Time | Last SMTP
+---+-----+-----+-----+-----+-----+-----+
+-----+-----+
| 28 | 284 | 0 | radiant@localhost | marketing@aaaaa.ru | - | - | - |
+---+-----+-----+-----+-----+-----+
+-----+-----+

Mail queue contains 1 message(s) Done

```

Чтобы удалить электронные письма, находящиеся в очереди на отправку, укажите ключ `--delete-all` или `--delete` с активным фильтром.

```

shell> bin/radiant.Console.pl Maint::Email::MailQueue --delete --filter ID::28 Deleted
messages from mail queue.

Done.

```

Вы также можете вручную отправить электронную почту, не дожидаясь запуска задачи RSD Daemon. Просто укажите переключатель `--send`.

```

shell> bin/radiant.Console.pl Maint::Email::MailQueue --send

1 message(s) successfully sent!

Done.

```

Получение электронной почты

1. Почтовые учетные записи, настроенные через графический интерфейс RSD

RSD может получать электронные письма с почтовых учетных записей POP3, POP3S, IMAP и IMAPS. Настройте свои почтовые учетные записи с помощью ссылки «Учетные записи почты для PostMaster» на странице администрирования.

Если необходимо создать новую учетную запись электронной почты (см. рис. ниже), то необходимо указать ее имя почтового сервера, логин и пароль. Также вам необходимо выбрать тип почтового сервера, который может быть POP3, POP3S, IMAP или IMAPS.

Если вы не видите свой тип сервера в качестве опции, в вашей системе отсутствуют необходимые модули Perl. В этом случае обратитесь к разделу Установка модулей Perl, необходимых для RSD, за инструкциями.

Рисунок 4.80. Добавление учетной записи почты

Управление почтовыми учетными записями

Изменить почтовую учетную запись для хоста "imap.gmail.com" и учетную запись пользователя "test@gmail.com"

Действия

← Перейти к обзору

Подсказка

Все входящие письма с одной учетной записью будут перенаправлены в выбранную очередь.

Если ваша учетная запись помечена как доверенная, все уже существующие в письмах на момент получения заголовки для выставления приоритета и прочих (данных) будут приняты и использованы, например в фильтрах PostMaster.

Исходящая почта может быть настроена через параметр Sendmail* в Настройка системы.

Изменить почтовую учетную запись

* Тип: IMAP

* Логин: test@gmail.com

* Пароль:

* Сервер: imap.gmail.com

Пример: mail.example.com

Папка IMAP: INBOX

Измените только в том случае, если нужно забирать почту из папки, отличной от INBOX.

* Доверенная: Нет

* Перенаправление: Перенаправление по выбранной очереди

* Очередь: 1-я линия поддержки

* Действительность: действительный

Комментарий:

Сохранить или Отменить

Если вы выберете «Да» для значения параметра «Доверенная», все заголовки X-RADIANT, прикрепленные к входящему сообщению, оцениваются и выполняются. Так как заголовок X-RADIANT может выполнять некоторые действия в системе заявок, для известных отправителей вы должны установить для параметра «Доверенная» значение «Да». Заголовки X-RADIANT используются в RSD модулем фильтрации. Заголовки X-RADIANT описаны в этой таблице более подробно. Любые созданные правила фильтрации постмастера выполняются независимо от настройки параметра «Доверенная».

Распределением входящих сообщений можно управлять, если их нужно сортировать по очереди или по содержимому поля «Кому». Для поля «Перенаправление», если выбрано «Перенаправление по выбранной очереди», все входящие сообщения будут сортироваться в указанную очередь. Адрес, на который было отправлено письмо, в данном случае не учитывается. Если выбрано поле «Перенаправление по заголовку To: электронного письма», система проверяет, связана ли очередь с адресом в поле «Кому:» входящей почты. Вы можете связать адрес с очередью в разделе «Адреса email» на странице администрирования. Если адрес в поле «Кому:» связан с очередью, новое сообщение будет отсортировано в связанную очередь. Если связь между адресом в поле «Кому:» и какой-либо очередью не найдена, то сообщение будет сохраняться в системе в очереди "Raw», которая задана в PostmasterDefaultQueue по умолчанию после установки системы.

Все данные для почтовых аккаунтов сохраняются в базе данных RSD. Команда bin/radiant.Console.pl Maint::PostMaster::MailAccountFetch использует настройки в базе данных и извлекает почту. Вы можете выполнить ее вручную, чтобы проверить, правильно ли работают все настройки вашей почты.

При обычной установке почта будет получаться демоном RSD каждые 10 минут.

Примечание

При получении почты RSD удаляет почту с сервера POP или IMAP. Нет возможности также хранить копию на сервере. Если вы хотите сохранить копию на сервере, вам следует создать правила пересылки на вашем почтовом сервере. Подробности смотрите в документации вашего почтового сервера.

2. Через программу командной строки и Procmail (radiant.Console.pl Maint::PostMaster::Read)

Если вы не можете использовать почтовые учетные записи для отправки электронной почты в RSD, программа командной строки bin/radiant.Console.pl Maint::PostMaster::Read может решить эту проблему. Она принимает письма через STDIN и передает их напрямую в RSD. Это означает, что электронная почта будет доступна в вашей системе RSD, если MDA (агент доставки почты, например, procmail) выполнит эту программу.

Чтобы протестировать bin/radiant.Console.pl Maint::PostMaster::Read без MDA, выполните команду следующего скрипта.

```
linux:/opt/radiant# cd bin
linux:/opt/radiant/bin# cat ../doc/sample_mails/test-email-1.box | ./radiant.Console.pl
Maint::PostMaster::Read
linux:/opt/radiant/bin#
```

Сценарий: Тестирование PostMaster без MDA.

Если сообщения электронной почты отображаются в просмотре очереди, значит ваши настройки работают.

Пример 4.2. Маршрутизация через Procmail с помощью radiant.Console.pl

Для маршрутизации почты в определенной очереди с помощью radiant.Console.pl используйте следующий пример.

```
| $SYS_HOME/bin/radiant.Console.pl Maint::PostMaster::Read --target-queue=QUEUENAME
```

При сортировке во вложенную очередь вы должны разделить родительскую и дочернюю очереди символом «::».

```
| $SYS_HOME/bin/radiant.Console.pl Maint::PostMaster::Read --target-queue=QUEUENAME::SUBQUEUE
```

Procmail — очень распространенный фильтр электронной почты в среде Linux. Он установлен на большинстве систем. Если нет, взгляните на [домашнюю страницу procmail](#).

Чтобы настроить procmail для RSD (на основе MTA, сконфигурированного для procmail, такого как sendmail, postfix, exim или qmail), используйте файл `~radiant/.procmailrc.dist`, скопируйте его в `.procmailrc` и добавьте строки из приведенного ниже сценария.

```
SYS_HOME=$HOME

PATH=/bin:/usr/bin:/usr/local/bin # --

# Pipe all email into the PostMaster process. # --

:0 :

| $SYS_HOME/bin/radiant.Console.pl Maint::PostMaster::Read
```

Сценарий: Настройка procmail для RSD

Вся электронная почта, отправленная локальному пользователю RSD, будет направляться в `bin/radiant.Console.pl Main-t::PostMaster::Read`, а затем отображаться в просмотре очереди.

3. Получение электронной почты через POP3 или IMAP и fetchmail для `radiant.Console.pl Maint::PostMaster::Read`

Чтобы получать электронную почту с вашего почтового сервера через почтовый ящик POP3 или IMAP на машину RSD/локальную учетную запись RSD и на procmail, используйте fetchmail.

Примечание

Требуется работающая конфигурация SMTP на машине RSD.

Вы можете использовать `.fetchmailrc.dist` в домашнем каталоге RSD и скопировать его в `.fetch-mailrc`. Модифицируйте/измените его под свои нужды (см. пример ниже).

Пример 4.3. `.fetchmailrc`

```
#poll (mailserver) protocol POP3 user (user) password (password) is (localuser) poll
mail.example.com protocol POP3 user joe password mama is rsd
```

Не забудьте установить для `.fetchmailrc` значение 710 (**`chmod 710 .fetchmailrc`**)!

С `.fetchmailrc` из приведенного выше примера вся электронная почта будет пересылаться на локальную учетную запись RSD, если будет выполнена команда

fetchmail -a. Настройте cronjob с помощью этой команды, если вы хотите получать почту регулярно.

4. Фильтрация/Перенаправление модулями RSD/PostMaster (для более сложного перенаправления)

Если вы используете метод `bin/radiant.Console.pl Maint::PostMaster::Read` или `bin/radiant.Console.pl Maint::PostMaster::MailAccountFetch`, вы можете вставлять или изменять записи заголовка X- RSD с помощью модулей фильтрации PostMaster. С помощью заголовков X-RADIANT система заявок может выполнять некоторые действия с входящей почтой, сортировать их в определенную очередь, изменять приоритет или, например, идентификатор клиента. Более подробная информация о заголовках X-RADIANT доступна в разделе о добавлении почтовых учетных записей на странице администрирования RSD.

Есть некоторые предустановленные модули фильтрации:

Примечание

Имя задания (например, `$Self->{'PostMaster::PreFilterModule'}->{'JobName'}`) должно быть уникальным!

`Kernel::System::PostMaster::Filter::Match` — это модуль по умолчанию для сопоставления некоторых заголовков электронной почты (например, `From`, `To`, `Subject`, ...). Он может устанавливать новые заголовки электронной почты (например, `X-RADIANT-Ignore: yes` или `X-RADIANT-Queue: spam`), если правило фильтрации совпадает. Задания из приведенного ниже примера можно вставить в `Kernel/Config.pm`.

Пример 4.4. Примеры заданий для модуля фильтрации `Kernel::System::PostMaster::Filter::Match`

```
# Job Name: 1-Match

# (block/ignore all spam email with From: noreply@)

$Self->{'PostMaster::PreFilterModule'}->{'1-Match'} = { Module =>
'Kernel::System::PostMaster::Filter::Match', Match => {

From => 'noreply@',

},

Set => {

'X-RADIANT-Ignore' => 'yes',

},

};

# Job Name: 2-Match
```

```

# (sort emails with From: sales@example.com and Subject: **ORDER** # into queue
'Order')

$Self->{'PostMaster::PreFilterModule'}->{'2-Match'} = { Module =>
'Kernel::System::PostMaster::Filter::Match', Match => {

To => 'sales@example.com', Subject => '**ORDER**',

},

Set => {

'X-RADIANT-Queue' => 'Order',

},

};

```

Kernel::System::PostMaster::Filter::CMD - это модуль по умолчанию для передачи электронной почты во внешнюю команду. Выходные данные передаются в STDOUT, и если результат равен true, то устанавливается новый заголовок электронной почты (например, X-RADIANT-Ignore: yes или X-RADIANT-Queue: spam). Приведенный ниже пример можно использовать в Kernel/Config.pm

Пример 4.5. Пример задания для модуля фильтрации Kernel::System::PostMaster::Filter::CMD

```

# Job Name: 5-SpamAssassin

# (SpamAssassin example setup, ignore spam emails)

$Self->{'PostMaster::PreFilterModule'}->{'5-SpamAssassin'} = { Module =>
'Kernel::System::PostMaster::Filter::CMD',

CMD => '/usr/bin/spamassassin | grep -i "X-Spam-Status: yes"', Set => {

'X-RADIANT-Ignore' => 'yes',

},

};

```

Kernel::System::PostMaster::Filter::ExternalTicketNumberRecognition — это модуль по умолчанию, который добавляет возможность анализировать внешние идентификаторы в теме письма, в теле письма или в том и другом, используя регулярные выражения. Затем он сохраняет это значение в определенном динамическом поле. Когда приходит электронное письмо, RSD сначала ищет внешний идентификатор, а когда находит, запрашивает RSD в предварительно определенном динамическом поле. Если он найдет существующую заявку, он ее обновит, в противном случае он создаст новую заявку с внешним номером ссылки в отдельном поле.

RSD SysConfig уже предоставляет 4 различных параметра для установки разных номеров внешних заявок. Если нужны дополнительные настройки, их нужно добавить вручную. Следующий пример можно использовать в Kernel/Config.pm для расширения параметров SysConfig.

Пример 4.6. Пример задания для модуля фильтрации Kernel::System::PostMaster::Filter::ExternalTicketNumberRecognition

```
# Job Name: ExternalTicketNumberRecognition

# External Ticket Number Reconition, check for Incident-<number> in incoming mails
subject and

# body from the addeesses <sender>@externalticket.com, if number is found it will be
stored in

# the dynamic field 'ExternalNumber' (that need to be setup in the Admin Panel).

$Self->{'PostMaster::PreFilterModule'}->{'000-ExternalTicketNumberRecognition'} = {
'FromAddressRegExp' => '\s*@externalticket.com',

'NumberRegExp'      => 'Incident-(\d.*)', 'SearchInSubject'    => '1',

'SearchInBody'      => '1', 'TicketStateTypes'              => 'new;open'
'DynamicFieldName' => 'ExternalNumber', 'Module'              =>

'Kernel::System::PostMaster::Filter::ExternalTicketNumberRecognition', 'Name' =>
'Test External Ticket Number',

'SenderType'        => 'system',

};
```

Параметры конфигурации

- FromAddressRegExp

Это необязательная настройка. Только письма с заданным значением поля "From:" будут учитываться для этого фильтра. Вы можете настроить этот параметр на адрес отправителя, который ваша внешняя система использует для исходящих писем. В случае, если адреса различны, вы можете оставить этот параметр пустым. В этом случае RSD не будет проверять адрес отправителя.

- NumberRegExp

Это обязательная настройка. Этот параметр содержит регулярное выражение, которое RSD будет использовать для извлечения номера заявки из темы и/или тела заявки. Выражение по умолчанию будет проверять соответствие, например, 'Incident-12354' и

помещать часть значения между скобками в динамическое поле, в данном случае '12354'.

- SearchInSubject

Если для этого параметра установлено значение «1», в поле «Тема письма» будет осуществляться поиск номера заявки.

- SearchInBody

Если для этого параметра установлено значение «1», поиск номера заявки будет осуществляться в теле письма.

- TicketStateTypes

Это необязательный параметр. Если он задан, RSD будет искать открытые внешние заявки с заданными типами состояний. Типы состояний в списке разделяются точкой с запятой.

- DynamicField

Это обязательный параметр. Он определяет динамическое поле, которое используется для хранения внешнего номера заявки (имя поля должно существовать в системе и должно быть действительным).

- SenderType

Это определяет тип отправителя и используется при создании сообщений/ заметок в RSD.

Kernel::System::PostMaster::Filter::Decrypt - это модуль по умолчанию, который способен расшифровывать зашифрованное входящее сообщение электронной почты (S/MIME или PGP), помещая незашифрованное тело сообщения в заголовок электронной почты X-RADIANT-BodyDecrypted для последующей обработки. Кроме того, он также может обновить текст электронного письма до незашифрованной версии.

Чтобы расшифровать электронные письма, система должна быть правильно настроена для S/MIME и/или PGP и иметь необходимые закрытые ключи для расшифровки информации.

Этот модуль по умолчанию отключен, и его можно настроить непосредственно в конфигурации системы в панели администрирования.

Параметры конфигурации

- StoreDecryptedBody

Установите для этого параметра значение "1", чтобы обновить текст письма до незашифрованной версии, если расшифровка прошла успешно. Имейте в виду, что при этом электронные письма будут храниться в незашифрованном виде, и отменить это действие невозможно.

Конечно, вы также можете разработать свои собственные фильтрующие модули PostMaster.

5. Устранение неполадок фильтрации электронной почты

В этом разделе показаны некоторые распространенные проблемы и рекомендации, которые следует учитывать при устранении неполадок с фильтрами Postmaster.

- Фильтры работают в алфавитном порядке их названий. Последний фильтр выигрывает для установки определенного поля, когда критерии совпадают дважды.
- Stop After Match может предотвратить второе совпадение.
- Убедитесь, что регулярное выражение допустимо.
- Заголовки могут быть установлены для управления RSD, но в самой почте они не записываются.
- При сопоставлении одного From, CC, TO используйте EMAILADDRESS: <your@address>
- Почтовый ящик должен быть доверенным.
- Критериями соответствия являются условия И.
- Свойства заявки не могут быть сопоставлены с помощью фильтра postmaster.

Примечание

Подробная информация об обработке фильтра Postmaster также присутствует в журнале сеансов связи. Обязательно проверьте раздел «Журнал сеансов связи» на странице администрирования.

4.2. Защищенная электронная почта с помощью PGP

RSD имеет возможность подписывать или шифровать исходящие сообщения с помощью PGP. Кроме того, зашифрованные входящие сообщения могут быть расшифрованы. Шифрование и дешифрование выполняются с помощью инструмента GPL GnuPG. Чтобы настроить GnuPG для RSD, необходимо выполнить следующие шаги:

1. Установите GnuPG через менеджер пакетов вашей операционной системы.
2. Настройте GnuPG для использования с RSD. Необходимо создать необходимые каталоги для GnuPG и закрытый ключ. Команда, показанная в приведенном ниже сценарии, должна выполняться от имени пользователя rsd из оболочки.

```
linux:~# su radiant linux:/root$ cd linux:~$ pwd
/opt/radiant
linux:~$ gpg --gen-key
gpg (GnuPG) 1.4.2; Copyright (C) 2005 Free Software Foundation, Inc. This program
comes with ABSOLUTELY NO WARRANTY.
```

This is free software, and you are welcome to redistribute it under certain conditions. See the file COPYING for details.

gpg: directory `/opt/radiant/.gnupg' created

gpg: new configuration file `/opt/radiant/.gnupg/gpg.conf' created

gpg: WARNING: options in `/opt/radiant/.gnupg/gpg.conf' are not yet active during this run

gpg: keyring `/opt/radiant/.gnupg/secring.gpg' created gpg: keyring
`/opt/radiant/.gnupg/pubring.gpg' created Please select what kind of key you want:

- (1) DSA and Elgamal (default)
- (2) DSA (sign only)
- (5) RSA (sign only) Your selection? 1

DSA keypair will have 1024 bits.

ELG-E keys may be between 1024 and 4096 bits long. What keysize do you want?
(2048)

Requested keysize is 2048 bits

Please specify how long the key should be valid.

0 = key does not expire

<n> = key expires in n days

<n>w = key expires in n weeks

<n>m = key expires in n months

<n>y = key expires in n years Key is valid for? (0)

Key does not expire at all Is this correct? (y/N) y

You need a user ID to identify your key; the software constructs the user ID from the Real Name, Comment and Email Address in this form:

"Heinrich Heine (Der Dichter) <heinrichh@duesseldorf.de>"

Real name: Ticket System

Email address: support@example.com

Comment: Private PGP Key for the ticket system with address support@example.com
You selected this USER-ID:

"Ticket System (Private PGP Key for the ticket system with address support@example.com) <support@example.com>"

Change (N)ame, (C)omment, (E)mail or (O)kay/(Q)uit? O You need a Passphrase to protect your secret key.

Passphrase: secret

Repeat passphrase: secret

We need to generate a lot of random bytes. It is a good idea to perform some other action (type on the keyboard, move the mouse, utilize the disks) during the prime generation; this gives the random number generator a better chance to gain enough entropy.

+++++++.,+++++.....+++++.+++++...+++++
+++++++.

+++++++.,+++++.+++++.+++++>+
+++++++>+++++

.....>+++++<+++++.....+++++

Not enough random bytes available. Please do some other work to give the OS a chance to collect more entropy! (Need 280 more bytes)

+++++++.,+++++..+++++..+++++....+++++.,+++++
+++++.,+++++

+++++++.,+++++.,+++++.,+++++.,+++++.
+++++++

..+++++>.,+++++...>+++++.....

.....>+++++<+++++.....

```

.....++++^^^

gpg: /opt/radiant/.gnupg/trustdb.gpg: trustdb created gpg: key 7245A970 marked as
ultimately trusted public and secret key created and signed.

gpg: checking the trustdb

gpg: 3 marginal(s) needed, 1 complete(s) needed, PGP trust model

gpg: depth: 0 valid:      1 signed:      0 trust: 0-, 0q, 0n, 0m, 0f, 1u pub
1024D/7245A970 2006-02-03

Key fingerprint = 2ED5 BC36 D2B6 B055 7EE1 5833 1D7B F967 7245 A970

uid                Ticket System (Private pgp key for ticket system with addre
ss support@example.com) <support@example.com>

sub                2048g/52B97069 2006-02-03

linux:~$

```

Сценарий: Настройка GnuPG

Как показано в приведенном ниже сценарии, для большинства необходимых параметров могут быть применены настройки по умолчанию. Только значения для владельца ключа должны быть введены правильно, с правильным паролем, указанным для ключа.

3. На экране настроек PGP должен быть активирован PGP для RSD (первый вариант). Также следует задать и проверить путь к программе gpg.

Следующий параметр конфигурации (PGP::Options) также может потребовать изменения. С помощью этой настройки конфигурации можно указать параметры, используемые для каждого выполнения gpg пользователем 'rsd'. В частности, важен каталог конфигурационных файлов для GnuPG пользователя rsd. В примере используется /opt/radiant/.gnupg. Этот каталог был создан ранее во время настройки PGP.

С помощью следующей настройки конфигурации (PGP::Key::Password) можно указать пары идентификаторов ключей и их паролей для собственных закрытых ключей. Поскольку внешние партнеры по общению записывают в систему заявок свои сообщения, зашифрованные вашим открытым ключом, RSD может расшифровать эти сообщения с помощью идентификатора / паролей, указанных здесь.

Как получить идентификатор собственного закрытого ключа? Идентификатор вашего собственного закрытого ключа уже отображается во время генерации ключа (см. шаг 1

выше). Также возможно получить идентификатор, если команда, указанная в следующем скрипте, выполняется от имени пользователя rsd:

```
linux:~# su radiant linux:/root$ cd linux:~$ pwd
/opt/radiant
linux:~$ gpg --list-keys
/opt/radiant/.gnupg/pubring.gpg

pub       1024D/7245A970 2006-02-03

uid                               Ticket System (Private pgp key for ticket system with
address support@example.com) <support@example.com>

sub       2048g/52B97069 2006-02-03

linux:~$
```

Сценарий: Получение идентификатора вашего собственного закрытого ключа.

Идентификатор закрытого ключа можно найти в строке, начинающейся с "sub". Это шестнадцатеричная строка длиной восемь символов, в приведенном выше примере это "52B97069". Пароль, который вы должны указать для этого ключа в системе обработки заявок, тот же, который был указан при генерации ключа.

После того, как эти данные будут вставлены, можно использовать кнопку «Обновить» для сохранения настроек. Теперь RSD готов принимать и расшифровывать зашифрованные сообщения.

4. Наконец, импортируйте открытый ключ клиента. Это гарантирует, что зашифрованные сообщения могут быть отправлены этому клиенту. Существует два способа импортировать открытый ключ клиента.

Первый способ - указать открытый ключ клиента в интерфейсе управления клиентами.

Второй способ - указать ключ через настройки PGP, доступные со страницы администратора. В правой части этого экрана отображаются все уже импортированные открытые ключи клиентов. После того, как PGP был активирован и настроен для RSD, там также должен быть указан ваш собственный открытый ключ. В левой части экрана настроек PGP можно выполнить поиск ключей. Кроме того, новый открытый ключ может быть загружен в систему из файла.

Файлы с открытым ключом, которые необходимо импортировать в RSD, должны быть совместимы с GnuPGP. В большинстве случаев ключ, хранящийся в файле, является "бронированным ключом ASCII". RSD может работать с этим форматом.

4.3. Защищенная электронная почта с помощью S/MIME

На первый взгляд шифрование с помощью S/MIME кажется немного более сложным, чем с помощью PGP. Во-первых, вы должны создать Центр сертификации (CA) для системы RSD. Последующие шаги очень похожи на шаги, необходимые для PGP: настроить RSD, установить собственный сертификат, импортировать другие общедоступные сертификаты по мере необходимости и т. д.

Настройка S/MIME по большей части выполняется вне веб-интерфейса RSD и должна выполняться пользователем rsd в командной оболочке. Конфигурация MIME в Linux основана на SSL (OpenSSL). Поэтому прежде всего проверьте, установлен ли пакет OpenSSL в вашей системе. Пакет OpenSSL включает в себя скрипт, называемый CA.pl, с помощью которого могут быть выполнены наиболее важные этапы создания сертификата. Чтобы упростить процедуру, выясните, где в файловой системе хранится сценарий CA.pl, и временно введите его местоположение в переменную PATH оболочки (см. Сценарий ниже).

```
rsd@linux:~> rpm -ql openssl | grep CA
/usr/share/ssl/misc/CA.pl
rsd@linux:~> export PATH=$PATH:/usr/share/ssl/misc rsd@linux:~> which CA.pl
/usr/share/ssl/misc/CA.pl rsd@linux:~> mkdir tmp; cd tmp rsd@linux:~/tmp>
```

Сценарий: Настройка S/MIME.

Приведенный выше сценарий показывает, что создан новый временный каталог ~/tmp, в котором должен быть сгенерирован сертификат.

Чтобы создать сертификат, выполните следующие операции в командной строке (мы предполагаем, что администратор RSD должен создать SSL-сертификат для целей тестирования и обучения. Если у вас уже есть сертифицированный SSL-сертификат для шифрования, используйте его и пропустите эти шаги):

1. Создайте свой собственный центр сертификации для SSL. Он нужен вам для подтверждения запроса на ваш собственный SSL-сертификат (см. Сценарий ниже).

```
rsd@linux:~/tmp> CA.pl -newca
CA certificate filename (or enter to create)
Making CA certificate ...
Generating a 1024 bit RSA private key
...++++++
```

.....+++++

writing new private key to './demoCA/private/cakey.pem' Enter PEM pass phrase:

Verifying - Enter PEM pass phrase:

You are about to be asked to enter information that will be incorporated into your certificate request.

What you are about to enter is what is called a Distinguished Name or a DN. There are quite a few fields but you can leave some blank

For some fields there will be a default value, If you enter '.', the field will be left blank.

Country Name (2 letter code) [AU]:DE

State or Province Name (full name) [Some-State]:RSD-state Locality Name (eg, city) []:RSD-town

Organization Name (eg, company) [Internet Widgits Pty Ltd]:Your company
Organizational Unit Name (eg, section) []:

Common Name (eg, YOUR name) []:RSD Admin Email Address []:rsd@your-domain.tld
rsd@linux:~/tmp> ls -la demoCA/

total 8

-rw-r--r-- 1 rsd rsd 1330 2006-01-08 17:54 cacert.pem

drwxr-xr-x 2 rsd rsd 48 2006-01-08 17:53 certs

drwxr-xr-x 2 rsd rsd 48 2006-01-08 17:53 crl

-rw-r--r-- 1 rsd rsd 0 2006-01-08 17:53 index.txt

drwxr-xr-x 2 rsd rsd 48 2006-01-08 17:53 newcerts

drwxr-xr-x 2 rsd rsd 80 2006-01-08 17:54 private

-rw-r--r-- 1 rsd rsd 17 2006-01-08 17:54 serial

rsd@linux:~/tmp>

Сценарий: Создание центра сертификации для SSL.

2. Сгенерируйте запрос на сертификат (см. Сценарий ниже).

```
rsd@linux:~/tmp> CA.pl -newreq Generating a 1024 bit RSA private key
```

```
.....++++++
```

```
....++++++
```

```
writing new private key to 'newreq.pem' Enter PEM pass phrase:
```

```
Verifying - Enter PEM pass phrase:
```

You are about to be asked to enter information that will be incorporated into your certificate request.

What you are about to enter is what is called a Distinguished Name or a DN. There are quite a few fields but you can leave some blank

For some fields there will be a default value, If you enter '.', the field will be left blank.

```
Country Name (2 letter code) [AU]:DE\keyreturn
```

```
State or Province Name (full name) [Some-State]:RSD-state Locality Name (eg, city) []:RSD-town
```

```
Organization Name (eg, company) [Internet Widgits Pty Ltd]:Your company  
Organizational Unit Name (eg, section) []:
```

```
Common Name (eg, YOUR name) []:RSD admin Email Address []:rsd@your-domain.tld
```

Please enter the following 'extra' attributes to be sent with your certificate request

```
A challenge password []:
```

```
An optional company name []:
```

```
Request (and private key) is in newreq.pem rsd@linux:~/tmp> ls -la
```

```
total 4
```

```
drwxr-xr-x 6 rsd rsd 232 2006-01-08 17:54 demoCA
```

```
-rw-r--r-- 1 rsd rsd 1708 2006-01-08 18:04 newreq.pem rsd@linux:~/tmp>
```

Сценарий: Создание запроса на сертификат.

3. Подписание запроса на сертификат. Запрос на сертификат может быть либо подписан и, таким образом, сертифицирован вашим собственным центром сертификации, либо сделан более достоверным путем подписания другим внешним сертифицированным центром сертификации (см. Сценарий ниже).

```
rsd@linux:~/tmp> CA.pl -signreq
```

```
Using configuration from /etc/ssl/openssl.cnf Enter pass phrase for  
./demoCA/private/cakey.pem: Check that the request matches the signature Signature  
ok
```

Certificate Details:

Serial Number: fd:85:f6:9f:14:07:16:c8

Validity

Not Before: Jan 8 17:04:37 2006 GMT

Not After : Jan 8 17:04:37 2007 GMT

Subject:

countryName = DE

stateOrProvinceName = RSD-state localityName
= RSD-town

organizationName = Your Company

commonName = RSD administrator

emailAddress = rsd@your-domain.tld X509v3 extensions:

X509v3 Basic Constraints:

CA:FALSE

Netscape Comment:

OpenSSL Generated Certificate X509v3 Subject Key Identifier:

01:D9:1E:58:C0:6D:BF:27:ED:37:34:14:D6:04:AC:C4:64:98:7A:22

X509v3	Authority	Key	Identifier:
keyid:10:4D:8D:4C:93:FD:2C:AA:9A:B3:26:80:6B:F5:D5:31:E2:8E:DB:A8			

DirName:/C=DE/ST=RSD-state/L=RSD-town/O=Your Company/ CN=RSD
admin/emailAddress=rsd@your-domain.tld serial:FD:85:F6:9F:14:07:16:C7

```
Certificate is to be certified until Jan 8 17:04:37 2007 GMT (365 days) Sign the certificate? [y/n]:y
```

```
1 out of 1 certificate requests certified, commit? [y/n]y Write out database with 1 new entries
```

```
Data Base Updated
```

```
Signed certificate is in newcert.pem rsd@linux:~/tmp>
```

Сценарий: Подписание запроса сертификата.

4. Создайте свой собственный сертификат и все данные, связанные с ним, используя запрос подписанного сертификата (см. сценарий ниже).

```
rsd@linux:~/tmp> CA.pl -pkcs12 "RSD Certificate" Enter pass phrase for newreq.pem:
```

```
Enter Export Password:
```

```
Verifying - Enter Export Password: rsd@linux:~/tmp> ls -la
```

```
total 12
```

```
drwxr-xr-x 6 rsd rsd 328 2006-01-08 18:04 demoCA
```

```
-rw-r--r-- 1 rsd rsd 3090 2006-01-08 18:13 newcert.p12
```

```
-rw-r--r-- 1 rsd rsd 3791 2006-01-08 18:04 newcert.pem
```

```
-rw-r--r-- 1 rsd rsd 1708 2006-01-08 18:04 newreq.pem rsd@linux:~/tmp>
```

Сценарий: Генерация нового сертификата.

Теперь, когда эти операции выполнены, необходимо завершить настройку S/MIME в RSD.

Эта часть настройки выполняется на странице администрирования, выбрав ссылку "Сертификаты S/MIME". В случае, если общая поддержка S/MIME в RSD еще не включена, маска указывает на это администратору и предоставляет соответствующую ссылку для ее включения.

С помощью группы SysConfig "Crypt::SMIME" вы также можете включить и настроить общую поддержку S/MIME.

Здесь вы можете активировать поддержку S/MIME и определить пути для команды OpenSSL и каталог для сертификатов. Файл ключа, созданный выше, должен храниться в указанном здесь каталоге. В противном случае OpenSSL не сможет его использовать.

Следующий шаг выполняется в конфигурации S/MIME на странице администрирования RSD. Здесь вы можете импортировать закрытый ключ (и) системы RSD и открытые ключи других партнеров по коммуникации. Введите открытый ключ, который был создан в начале этого раздела и добавлен в RSD.

Очевидно, что все открытые S/MIME-ключи партнеров по коммуникации также можно импортировать с помощью инструмента администрирования пользователей клиента.

Извлекать сертификаты S/MIME из Клиентских пользовательских бэкендов

В качестве источника общедоступных сертификатов S/MIME можно использовать базу данных пользователя-клиента (например, LDAP). Эти сертификаты могут быть импортированы в систему и отображаться в конфигурации S/MIME на странице администратора RSD, и их можно использовать из RSD для отправки зашифрованных электронных писем клиентам.

Для того, чтобы включить эту функцию, необходимо:

1. Включите SMIME в SysConfig
2. Включите SMIME::FetchFromCustomer в SysConfig
3. Настройте серверную часть пользователя-клиента для предоставления атрибута UserSMIMECertificate с сертификатом пользователя клиента S/MIME (есть пример сопоставления пользователя клиента LDAP в \$RSD_HOME/Kernel/Config/Defaults.pm).

Эта функция может быть использована тремя различными способами:

1. Входящие электронные письма:

Специальный фильтр Postmaster (PostMaster::PreFilterModule###000-SMIMEFetchFromCustomer в SysConfig) извлечет адрес электронной почты из каждого входящего электронного письма и попытается найти адрес электронной почты в списке клиентов. Если он найден, он попытается получить сертификат S/MIME из пользовательских атрибутов клиента. Если сертификат найден, он попытается импортировать его (если только он уже не был импортирован).

2. Конкретный адрес электронной почты или всех клиентов:

Консольная команда Maint::SMIME::CustomerCertificate::Fetch может использоваться для импорта сертификата S/MIME одного адреса электронной почты клиента в виде:

```
shell> perl /opt/radiant/bin/radiant.Console.pl  
Maint::SMIME::CustomerCertificate::Fetch -- email customer@example.com
```

В этом случае консольная команда попытается сопоставить предоставленный адрес электронной почты с одним из пользователей клиента. Если он будет найден, он попытается добавить в систему сертификат S/MIME, найденный в свойствах пользователя - клиента (если сертификат еще не добавлен).

Та же консольная команда может использоваться для импорта сертификатов S/MIME всех пользователей клиента (ограничено свойством CustomerUserSearchListLimit из серверной части пользователя - клиента). Этот вариант не рекомендуется использовать, особенно для систем с большим количеством пользователей-клиентов, поскольку для его выполнения может потребоваться слишком много времени, и в зависимости от ограничения может оказаться, что не все сертификаты клиентов будут извлечены. Выполните консольную команду в этом режиме следующим образом:

```
shell> perl /opt/radiant/bin/radiant.Console.pl
Maint::SMIME::CustomerCertificate::Fetch --add- all
```

Для этого варианта консольная команда будет запрашивать серверные части пользователя-клиента, чтобы получить всех возможных клиентов, и для каждого она проверит наличие сертификата S/MIME. Если сертификат найден, она попытается добавить его в систему (если сертификат еще не добавлен).

3. Обновите существующие сертификаты:

Другая консольная команда `Maint::SMIME::CustomerCertificate::Renew` может использоваться для проверки всех существующих сертификатов в системе. Это подтверждает, что существующие сертификаты от пользователей-клиентов совпадают с теми, которые извлекаются свойствами пользователя-клиента. Любой новый сертификат в серверной части пользователя-клиента будет добавлен в систему (в этом процессе никакие сертификаты не удаляются).

Эта консольная команда выполняется один раз в день демоном RSD автоматически с задачей `Daemon::SchedulerCronTaskManager::Task####RenewCustomerSMIME-Certificates` (как показано в SysConfig), но она также может быть выполнена вручную по следующему запросу:

```
shell> perl /opt/radiant/bin/radiant.Console.pl
Maint::SMIME::CustomerCertificate::Renew
```

5. Использование внешних бэкендов

5.1. Данные клиентов

RSD работает со многими атрибутами данных о клиентах, такими как имя пользователя, адрес электронной почты, номер телефона и т. д. Эти атрибуты отображаются как в интерфейсе агента, так и в интерфейсе клиента, а также используются для аутентификации клиентов.

Данные клиентов, используемые или отображаемые в RSD, легко настраиваются. Однако для аутентификации клиента всегда необходима следующая информация:

- Логин пользователя
- Адрес электронной почты
- Пользовательский ИД

Используйте следующие параметры SysConfig, если вы хотите отобразить информацию о клиенте в интерфейсе вашего агента.

```
# Ticket::Frontend::CustomerInfo*

# (show customer info on Compose (Phone and Email), Zoom and # Queue view)

$Self->{'Ticket::Frontend::CustomerInfoCompose'} = 1;

$Self->{'Ticket::Frontend::CustomerInfoZoom'} = 1;
```

Сценарий: параметры конфигурации SysConfig.

5.2. Серверная часть пользователя клиента

Вы можете использовать два типа клиентских бэкендов: DB и LDAP. Если у вас уже есть другой клиентский сервер (например, SAP), разумеется, можно написать модуль, который его использует.

1. База данных (по умолчанию)

В приведенном ниже примере показана конфигурация клиентской базы данных БД, которая использует данные о клиентах, хранящиеся в базе данных RSD.

Пример 4.7. Настройка серверной части клиента БД

```
# CustomerUser (customer database backend and settings)

$Self->{CustomerUser} = {

  Name => 'Database Datasource',

  Module => 'Kernel::System::CustomerUser::DB', Params => {

    # if you want to use an external database, add the required settings # DSN =>
    'DBI:odbc:yourdsn',

    #           Type => 'mssql', # only for ODBC connections

    #           DSN => 'DBI:mysql:database=customerdb;host=customerdbhost',

    #           User => "",

    #           Password => "",

    Table => 'customer_user',
```

```

# CaseSensitive will control if the SQL statements need LOWER() #          function calls
to work case insensitively. Setting this to #          1 will improve performance dramatically on
large databases. CaseSensitive => 0,

},

# customer unique id

CustomerKey => 'login',

# customer #

CustomerID => 'customer_id',

CustomerValid => 'valid_id',

CustomerUserListFields => ['first_name', 'last_name', 'email'], CustomerUserSearchFields =>
['login', 'last_name', 'customer_id'], CustomerUserSearchPrefix => "",

CustomerUserSearchSuffix    =>    '*',    CustomerUserSearchListLimit    =>    250,
CustomerUserPostMasterSearchFields => ['email'],

CustomerUserNameFields => ['title', 'first_name', 'last_name'],

CustomerUserEmailUniqCheck => 1,

# # show not own tickets in customer panel, CompanyTickets #
CustomerUserExcludePrimaryCustomerID => 0,

# # generate auto logins # AutoLoginCreation => 0,

# AutoLoginCreationPrefix => 'auto',

# # admin can change customer preferences # AdminSetPreferences => 1,

# # cache time to live in sec. - cache any database queries # CacheTTL => 0,

# # just a read only source #

ReadOnly => 1,

Map => [

# note: Login, Email and CustomerID needed!

# var, frontend, storage, shown (1=always,2=lite), required, storage-type, http- link, readonly,
http-link-target, link class(es)

['UserTitle',          'Title',          'title',          1, 0, 'var', "", 0 ],

```

```

[ 'UserFirstname', 'Firstname', 'first_name', 1, 1, 'var', "", 0 ],
[ 'UserLastname',      'Lastname',      'last_name', 1, 1, 'var', "", 0 ],
[ 'UserLogin',         'Username',         'login',         1, 1, 'var', "", 0 ],
[ 'UserPassword',     'Password',         'pw',           0, 0, 'var', "", 0 ],
[ 'UserEmail',        'Email',            'email',        1, 1, 'var', "", 0 ],
# [ 'UserEmail',          Translatable('Email'),          'email',
  1,          1,          'var',          '[%          Env("CGIHandle")
%]?Action=AgentTicketCompose;ResponseID=1;TicketID=[% Data.TicketID | uri
%];ArticleID=[% Data.ArticleID | uri %]', 0, "", 'AsPopup RSDPopup_TicketAction' ], [
'UserCustomerID', 'CustomerID', 'customer_id', 0, 1, 'var', "", 0 ],

```

```

# [ 'UserCustomerIDs', 'Customer IDs', 'customer_ids', 1, 0, 'var', "", 0 ],
[ 'UserPhone',        'Phone',        'phone',        1, 0, 'var', "", 0 ],
[ 'UserFax',          'Fax',          'fax',          1, 0, 'var', "", 0 ],
[ 'UserMobile',       'Mobile',       'mobile',       1, 0, 'var', "", 0 ],
[ 'UserStreet',       'Street',       'street',       1, 0, 'var', "", 0 ],
[ 'UserZip',          'Zip',          'zip',          1, 0, 'var', "", 0 ],
[ 'UserCity',         'City',         'city',         1, 0, 'var', "", 0 ],
[ 'UserCountry',      'Country',      'country',      1, 0, 'var', "", 0 ],
[ 'UserComment',      'Comment',      'comments',     1, 0, 'var', "", 0 ],

```

```
[ 'ValidID',          'Valid',          'valid_id',      0,      1,      'int',      "      0      ],  
,
```

```
# Dynamic field example
```

```
# [ 'DynamicField_Name_X', undef, 'Name_X', 0, 0, 'dynamic_field', undef, 0, undef, undef, ],  
,
```

```
# default selections
```

```
Selections => {
```

```
UserTitle => {
```

```
'Mr.' => 'Mr.',
```

```
'Mrs.' => 'Mrs.',
```

```
},
```

```
},
```

```
};
```

Если вы хотите настроить данные о клиентах, измените заголовки столбцов или добавьте новые в таблицу `customer_user` в базе данных RSD. В качестве примера сценарий ниже показывает, как добавить новое поле для номера комнаты.

```
linux:~# mysql -p Enter password:
```

```
Welcome to the MySQL monitor. Commands end with ; or \g.
```

```
Your MySQL connection id is 116 to server version: 5.0.18-Debian_7-log Type 'help;' or '\h' for help. Type '\c' to clear the buffer.
```

```
mysql> use rsd;
```

```
Reading table information for completion of table and column names You can turn off this feature to get a quicker startup with -A
```

```
Database changed
```

```
mysql> ALTER TABLE customer_user ADD room VARCHAR (250); Query OK, 1 rows affected (0.01 sec)
```

```
Records: 1 Duplicates: 0 Warnings: 0
```

```
mysql> quit Bye linux:~#
```

Сценарий: Добавление поля room в таблицу customer_user.

Теперь добавьте новый столбец в массив MAP в Kernel/Config.pm, как показано в следующем сценарии.

```
# var, frontend, storage, shown (1=always,2=lite), required, storage-type, http-link,
readonly, http-link-target, link class(es)
```

```
[...]
```

```
[ 'UserRoom',          'Room',          'room',          0, 1, 'var', "", 0 ],
```

Сценарий: Добавление поля комнаты в файл Kernel/Config.pm.

Также возможно отредактировать всю эту информацию о клиенте с помощью ссылки «Клиенты» в интерфейсе агента.

Примечание

Пожалуйста, обратите внимание, что вы можете опустить ключи http-link-target и link class в элементах массива map, если они не будут использоваться. Эти ключи добавляют атрибуты target="" и class="" к элементу HTTP link соответственно. Они игнорируются, если http-ссылка не установлена.

Клиент с несколькими идентификаторами (заявки компании)

Клиенту можно присвоить более одного идентификатора клиента. Это может быть полезно, если клиент должен получить доступ к заявкам других клиентов, например, если супервайзер захочет посмотреть заявки своих помощников. Если клиент может получить доступ к заявкам другого клиента, то используется особенность RSD "заявки компании". Заявки компании могут быть доступны перейдя по ссылке "Заявки Компании" в клиентской панели управления.

Для использования заявок компании, новый столбец с идентификатором (IDs), который должен быть доступен для клиента, должен быть добавлен в таблицу customer_user базы данных RSD (см. Сценарий ниже).

```
linux:~# mysql -p Enter password:
```

```
Welcome to the MySQL monitor. Commands end with ; or \g.
```

```
Your MySQL connection id is 124 to server version: 5.0.18-Debian_7-log
```

```
Type 'help;' or '\h' for help. Type '\c' to clear the buffer.
```

```
mysql> use rsd;

Reading table information for completion of table and column names You can turn off
this feature to get a quicker startup with -A

Database changed

mysql> ALTER TABLE customer_user ADD customer_ids VARCHAR (250); Query
OK, 1 rows affected (0.02 sec)

Records: 1 Duplicates: 0 Warnings: 0

mysql> quit Bye linux:~#
```

Сценарий: Добавление поля `customer_ids` в таблицу `customer_user`.

Теперь новый столбец должен быть добавлен в MAP-массив в `Kernel/Config.pm`, как это показано в нижеприведенном Сценарии.

```
# var, frontend, storage, shown (1=always,2=lite), required, storage-type, http-link,
readonly, http-link-target, link class(es)

[...]

[ 'UserCustomerIDs', 'CustomerIDs', 'customer_ids', 1, 0, 'var', "", 0 ],
```

Сценарий: Добавление поля `UserCustomerIDs` в `Kernel/Config.pm`

Теперь, новый столбец для мульти-идентификатора (IDs) клиента можно редактировать с помощью веб-интерфейса Агента, в разделе управления клиентами.

Чтобы гарантировать, что один клиент может получить доступ к заявкам других клиентов, добавьте идентификаторы этих других пользователей в новое поле для нескольких идентификаторов клиентов. Каждый идентификатор должен быть разделен точкой с запятой (см. Пример ниже).

Пример 4.8. Использование заявок компании с серверной частью БД

В вашей системе существуют клиенты А, В и С. Клиент А хочет иметь доступ к заявкам В и С через панель клиентов. В и С не должны иметь доступа к заявкам других пользователей.

Для реализации этой структуры, измените таблицу `customer_user` и маппинг (преобразование) в `Kernel/Config.pm` как это показано выше. Затем загрузите настройки для клиента А по ссылке «Клиенты» в интерфейсе агента или через страницу

администратора. Если настройки отображаются, добавьте в поле для CustomerID значения «B;C;».

2. LDAP

Если у вас есть каталог LDAP с данными ваших клиентов, вы можете использовать его в качестве серверной части клиента с RSD, как показано в примере ниже.

Пример 4.9. Настройка LDAP в качестве клиентского бэкенда

```
UserPw => ",

# in case you want to add always one filter to each ldap query, use # this option. e. g.
AlwaysFilter => '(mail=*)' or AlwaysFilter =>

'(objectclass=user)'

AlwaysFilter => ",

# if the charset of your ldap server is iso-8859-1, use this: # SourceCharset => 'iso-
8859-1',

# Net::LDAP new params (if needed - for more info see perldoc Net::LDAP) Params
=> {

port => 389,

timeout => 120,

async => 0,

version => 3,

},

},

# customer unique id CustomerKey => 'uid', # customer # CustomerID => 'mail',

CustomerUserListFields => ['cn', 'mail'], CustomerUserSearchFields => ['uid', 'cn',
'mail'], CustomerUserSearchPrefix => "", CustomerUserSearchSuffix => "*",
CustomerUserSearchListLimit => 250, CustomerUserPostMasterSearchFields =>
['mail'], CustomerUserNameFields => ['givenname', 'sn'],

# show not own tickets in customer panel, CompanyTickets
CustomerUserExcludePrimaryCustomerID => 0,

# add an ldap filter for valid users (expert setting) # CustomerUserValidFilter =>
'!(description=locked)',
```

```

# administrator can't change customer preferences AdminSetPreferences => 0,

# # cache time to live in sec. - cache any database queries # CacheTTL => 0,

Map => [

# note: Login, Email and CustomerID are mandatory!

# var, frontend, storage, shown (1=always,2=lite), required, storage-type, http- link,
readonly, http-link-target, link class(es)

[ 'UserTitle',          'Title',          'title',          1,  0,
'var', ", 0 ],

[ 'UserFirstname', 'Firstname', 'givenname',          1,  1, 'var', ", 0
],

[ 'UserLastname',    'Lastname',      'sn',              1,  1,
'var', ", 0 ],

[ 'UserLogin',       'Username',       'uid',              1,  1,
'var', ", 0 ],

[ 'UserEmail',       'Email',          'mail',              1,  1,
'var', ", 0 ],

[ 'UserCustomerID', 'CustomerID', 'mail',              0,  1, 'var', ", 0
],

#          [ 'UserCustomerIDs', 'CustomerIDs', 'second_customer_ids', 1, 0, 'var', ",
0 ], [ 'UserPhone', 'Phone', 'telephonenumber', 1, 0, 'var', ", 0 ],

[ 'UserAddress',     'Address',        'postaladdress',    1,  0,
'var', ", 0 ],

[ 'UserComment',     'Comment',        'description',      1,  0,
'var', ", 0 ],

],

};

```

Если в вашем каталоге LDAP хранятся дополнительные атрибуты клиентов, такие как имя менеджера, номер мобильного телефона или отдел, и если вы хотите отобразить эту информацию в RSD, просто разверните массив MAP в Kernel/Config.pm. с записями для этих атрибутов, как показано в следующем скрипте.

```

# var, frontend, storage, shown (1=always,2=lite), required, storage-type, http-link,
readonly, http-link-target, link class(es)

```

```
[...]  
[ 'UserPhone',          'Phone',          'telephonenumber', 1, 0, 'var', "", 0 ],
```

Сценарий: Добавление нового поля в файл Kernel/Config.pm.

Клиент с мульти-идентификаторами (IDs) (заявки компании)

При использовании LDAP-бэкенда клиенту можно присвоить больше одного клиентского идентификатора (Customer ID). Для использования заявок компании, в LDAP- директорию нужно добавить новое поле, которое содержит доступные агенту идентификаторы (IDs).

Если в LDAP-каталоге было создано новое поле, его также нужно добавить в MAP-массив в Kernel/Config.pm, как это показано в нижеприведенном сценарии.

```
# var, frontend, storage, shown (1=always,2=lite), required, storage-type, http-link,  
readonly  
  
[...]  
  
[ 'UserCustomerIDs', 'CustomerIDs', 'customer_ids', 1, 0, 'var', "", 0 ],
```

Сценарий: Отображение новых полей в файл Kernel/Config.pm.

Клиентские идентификаторы (IDs) можно редактировать напрямую в LDAP-директории. RSD может только считать информацию из LDAP, но не записывать.

Чтобы обеспечить доступ клиента к заявкам других клиентов, добавьте идентификаторы клиентов, к заявкам которых следует получить доступ, в новое поле в вашем каталоге LDAP. Каждый идентификатор должен быть разделен точкой с запятой (см. Пример ниже).

Пример 4.10. Использование заявок Компании с LDAP- бэкендом

Клиенты А, В и С существуют в вашей системе, и А хочет иметь доступ к заявкам В и С через панель клиентов. В и С не должны иметь доступа к заявкам других пользователей.

Чтобы реализовать эту настройку, измените каталог LDAP и сопоставление в Kernel/Config.pm, как описано выше. Затем добавьте в поле для идентификаторов клиентов значения "В;С;" для клиента А в вашем каталоге LDAP.

3. Использование более чем одного Клиентского бэкенда с RSD

Если вы хотите использовать более одного источника клиентских данных, используемого с RSD (например, LDAP и серверную часть базы данных), параметр

конфигурации CustomerUser должен быть расширен числом, например. «CustomerUser1», «CustomerUser2» (см. пример ниже).

Пример 4.11. Использование более, чем одного клиентского бэкенда с RSD

Следующий пример конфигурации показывает использование как для базы данных клиентского бэкенда с RSD.

```
# 1. Customer user backend: DB

# (customer database backend and settings)

$Self->{CustomerUser1} = {

Name => 'Customer Database',

Module => 'Kernel::System::CustomerUser::DB', Params => {

# if you want to use an external database, add the # required settings

#       DSN => 'DBI:odbc:yourdsn',

#       Type => 'mssql', # only for ODBC connections

#       DSN => 'DBI:mysql:database=customerdb;host=customerdbhost', # User
=> "",

#       Password => "",

Table => 'customer_user',

},

# customer unique id

CustomerKey => 'login',

# customer #

CustomerID => 'customer_id',

CustomerValid => 'valid_id',

CustomerUserListFields => ['first_name', 'last_name', 'email'],

CustomerUserSearchFields => ['login', 'last_name', 'customer_id'],

CustomerUserSearchPrefix => "",

CustomerUserSearchSuffix => '*'
```

```

CustomerUserSearchListLimit => 250,

CustomerUserPostMasterSearchFields => ['email'],

CustomerUserNameFields => ['title','first_name','last_name'],

CustomerUserEmailUniqCheck => 1,

# # show not own tickets in customer panel, CompanyTickets #
CustomerUserExcludePrimaryCustomerID => 0,

# # generate auto logins # AutoLoginCreation => 0,

# AutoLoginCreationPrefix => 'auto',

# # admin can change customer preferences

# AdminSetPreferences => 1,

# # cache time to live in sec. - cache any database queries

# CacheTTL => 0,

# # just a read only source

# ReadOnly => 1,

Map => [

# note: Login, Email and CustomerID needed!

# var, frontend, storage, shown (1=always,2=lite), required, storage-type, http- link,
readonly, http-link-target

```

```

'UserTitle',      'Title',      'title',      1    0    'var'    "    ]
,                ,                ,                ,                ,                ,
'UserFirstnam    'Firstnam     'first_name'  1    1    'var'    "    ]
e',              e',           ,                ,                ,                ,
'UserLastnam     'Lastname     'last_name'   1    1    'var'    "    ]
e',              ',            ,                ,                ,                ,
'UserLogin',     'Usernam     'login',      1    1    'var'    "    ]
e',              e',           ,                ,                ,                ,

```



```

# default selections

Selections => {

UserTitle => {

'Mr.' => 'Mr.',

'Mrs.' => 'Mrs.',

},

},

};

# 2. Customer user backend: LDAP

# (customer ldap backend and settings)

$Self->{CustomerUser2} = {

Name => 'LDAP Datasource',

Module => 'Kernel::System::CustomerUser::LDAP',

Params => {

# ldap host

Host => 'bay.csuhayward.edu',

# ldap base dn

BaseDN => 'ou=seas,o=csuh',

# search scope (one|sub) SSCOPE => 'sub',

# The following is valid but would only be necessary if the

# anonymous user does NOT have permission to read from the LDAP tree UserDN =>

'',

UserPw => '',

# in case you want to add always one filter to each ldap query, use # this option. e. g.

AlwaysFilter => '(mail=*)' or AlwaysFilter =>

'(objectclass=user)'

AlwaysFilter => '',

```

```

# if the charset of your ldap server is iso-8859-1, use this: #
SourceCharset => 'iso-8859-1',

# Net::LDAP new params (if needed - for more info see perldoc Net::LDAP)

Params => {

port => 389,

timeout => 120,

async => 0,

version => 3,

},

},

# customer unique id CustomerKey => 'uid',

# customer # CustomerID => 'mail',

CustomerUserListFields => ['cn', 'mail'],

CustomerUserSearchFields => ['uid', 'cn', 'mail'],

CustomerUserSearchPrefix => "",

CustomerUserSearchSuffix => "*",

CustomerUserSearchListLimit => 250,

CustomerUserPostMasterSearchFields => ['mail'],

CustomerUserNameFields => ['givenname', 'sn'],

# show not own tickets in customer panel,

CompanyTickets CustomerUserExcludePrimaryCustomerID => 0,

# add a ldap filter for valid users (expert setting)

# CustomerUserValidFilter => '!(description=locked)',

# admin can't change customer preferences

AdminSetPreferences => 0,

Map => [

# note: Login, Email and CustomerID needed!

```

```

# var, frontend, storage, shown (1=always,2=lite), required, storage-type, http- link,
readonly

[ 'UserTitle',          'Title',          'title',          1, 0, 'var', "", 0
],

[ 'UserFirstname', 'Firstname', 'givenname',          1, 1, 'var', "", 0
],

[ 'UserLastname',    'Lastname',      'sn',             1, 1, 'var', "", 0
],

[ 'UserLogin',       'Username',      'uid',            1, 1, 'var', "", 0
],

[ 'UserEmail',       'Email',         'mail',           1, 1, 'var', "", 0
],

[ 'UserCustomerID', 'CustomerID', 'mail',           0, 1, 'var', "", 0
],

#          [ 'UserCustomerIDs', 'CustomerIDs', 'second_customer_ids', 1, 0, 'var', "",
0 ], [ 'UserPhone',          'Phone',          'telephonenumber', 1, 0, 'var',
"", 0 ],

[ 'UserAddress',     'Address',       'postaladdress',  1, 0, 'var', "", 0
],

[ 'UserComment',    'Comment',       'description',    1, 0, 'var', "", 0
],

],

};

```

It is possible to integrate up to 10 different customer backends. Use the customer management interface in RSD to view or edit (assuming write access is enabled) all customer data.

4. Хранение пользовательских данных в динамических полях

Иногда бывает полезным также сохранить данные клиента заявки непосредственно в динамических полях заявки, для создания некоторых отчетов по этим данным. Например, по заявкам компании, когда клиент в течение отчетного периода перешел в другую компанию того же провайдера услуг техподдержки.

Динамические поля заполняются при создании заявки или когда клиент заявки изменяется. Значения полей берутся из данных клиента. Это работает для всех бэкендов, но наиболее полезно для LDAP-бэкендов.

Для включения этой опции RSD, активируйте параметр "Ticket::EventModulePost###950-DynamicFieldFromCustomerUser" и "DynamicFieldFromCustomerUser::Mapping". Последний параметр содержит настройку, какие поля данных клиента должны сохраняться в каждом из динамических полей. Поля должны быть заранее созданы в системе и включены в AgentTicketFreeText, так чтобы они были доступны для ручного заполнения. Они должны быть недоступны для ввода в AgentTicketPhone, AgentTicketEmail и AgentTicketCustomer, во избежание перекрытия автоматически установленных значений.

5.3. Хранилища данных (бэкенды) для аутентификации агентов и клиентов

RSD предлагает опцию для проверки подлинности агентов и клиентов с использованием различных хранилищ данных (бэкендов).

1. Хранилища данных (бэкенды) для аутентификации агентов

База Данных (DB, по умолчанию)

Базой данных для аутентификации агентов, которая используется по умолчанию, является база данных RSD. Агентов можно добавлять и редактировать через интерфейс управления агентами на странице администрирования (см. пример ниже).

Пример 4.12. Проверка подлинности агентов путем использования Базы Данных (DB) в качестве хранилища информации.

```
$Self->{'AuthModule'} = 'Kernel::System::Auth::DB';
```

LDAP

Если в каталоге LDAP хранятся все данные вашего агента, вы можете использовать модуль LDAP для аутентификации ваших пользователей в RSD (см. пример ниже). Этот модуль имеет доступ только для чтения к дереву LDAP, что означает, что вы не можете редактировать свои пользовательские данные через интерфейс управления агентом.

Пример 4.13. Проверка подлинности агентов при использовании LDAP в качестве хранилища данных

```
# This is an example configuration for an LDAP auth. backend. # (Make sure
Net::LDAP is installed!)

$Self->{'AuthModule'} = 'Kernel::System::Auth::LDAP';

$Self->{'AuthModule::LDAP::Host'} = 'ldap.example.com';

$Self->{'AuthModule::LDAP::BaseDN'} = 'dc=example,dc=com';
```

```

$Self->{'AuthModule::LDAP::UID'} = 'uid';

# Check if the user is allowed to auth in a posixGroup # (e. g. user needs to be in a
group xyz to use rsd)

$Self->{'AuthModule::LDAP::GroupDN'} =
'cn=rsdallow,ou=posixGroups,dc=example,dc=com';

$Self->{'AuthModule::LDAP::AccessAttr'} = 'memberUid'; # for ldap posixGroups
objectclass (just uid)

# $Self->{'AuthModule::LDAP::UserAttr'} = 'UID';

# for non ldap posixGroups objectclass (with full user dn) # $Self-
->{'AuthModule::LDAP::UserAttr'} = 'DN';

# The following is valid but would only be necessary if the

# anonymous user do NOT have permission to read from the LDAP tree

$Self->{'AuthModule::LDAP::SearchUserDN'} = "";

$Self->{'AuthModule::LDAP::SearchUserPw'} = "";

# in case you want to add always one filter to each ldap query, use

# this option. e. g. AlwaysFilter => '(mail=*)' or AlwaysFilter => '(objectclass=user)'

$Self->{'AuthModule::LDAP::AlwaysFilter'} = "";

# in case you want to add a suffix to each login name, then

# you can use this option. e. g. user just want to use user but # in your ldap directory
exists user@domain.com

# $Self->{'AuthModule::LDAP::UserSuffix'} = '@domain.com';

# Net::LDAP new params (if needed - for more info see perldoc Net::LDAP)

$Self->{'AuthModule::LDAP::Params'} = { port => 389,

timeout => 120,

async => 0,

version => 3,

};

```

Параметры конфигурации, показанные в приведенном ниже сценарии, можно использовать для синхронизации пользовательских данных из вашего каталога LDAP

в вашу локальную базу данных RSD. Это сокращает количество запросов к вашему серверу LDAP и ускоряет аутентификацию с помощью RSD. Синхронизация данных выполняется при первой аутентификации агента. Несмотря на то, что данные могут быть синхронизированы в локальной базе данных RSD, LDAP-каталог является последней инстанцией для проверки подлинности пользователей, таким образом пользователи, неактивные для дерева каталогов LDAP не могут пройти аутентификацию в RSD, даже в том случае, когда данные о пользователе уже хранятся в базе данных RSD. Данные агентов в LDAP-директории не могут быть отредактированы с помощью веб-интерфейса RSD, так что управление данными должно осуществляться непосредственно в дереве каталогов LDAP.

```
# defines AuthSyncBackend (AuthSyncModule) for AuthModule # if this key exists and
is empty, there won't be a sync. # example values: AuthSyncBackend,
AuthSyncBackend2

$Self->{'AuthModule::UseSyncBackend'} = 'AuthSyncBackend';

# agent data sync against ldap

$Self->{'AuthSyncModule'} = 'Kernel::System::Auth::Sync::LDAP';

$Self->{'AuthSyncModule::LDAP::Host'} = 'ldap://ldap.example.com/';

$Self->{'AuthSyncModule::LDAP::BaseDN'} = 'dc=rsd, dc=org';

$Self->{'AuthSyncModule::LDAP::UID'} = 'uid';

$Self->{'AuthSyncModule::LDAP::SearchUserDN'} = 'uid=sys, ou=user, dc=rsd,
dc=org';

$Self->{'AuthSyncModule::LDAP::SearchUserPw'} = 'some_pass';

$Self->{'AuthSyncModule::LDAP::UserSyncMap'} = { # DB -> LDAP
UserFirstname => 'givenName', UserLastname => 'sn', UserEmail => 'mail',
}; [...]

# AuthSyncModule::LDAP::UserSyncInitialGroups

# (sync following group with rw permission after initial create of first agent # login)

$Self->{'AuthSyncModule::LDAP::UserSyncInitialGroups'} = [ 'users',
];
```

Сценарий: Синхронизация пользовательских данных из каталога LDAP в базу данных RSD.

В качестве альтернативы вы можете использовать группы LDAP для определения членства в группах или ролей в RSD. Для получения дополнительной информации и примеров см. Kernel/Config/Defaults.pm. Вот пример синхронизации из LDAP в группы RSD.

```
# Attributes needed for group syncs

# (attribute name for group value key)

$Self->{'AuthSyncModule::LDAP::AccessAttr'} = 'memberUid';

# (select the attribute for type of group content UID/DN for full ldap name) # $Self->{'AuthSyncModule::LDAP::UserAttr'} = 'UID';

# $Self->{'AuthSyncModule::LDAP::UserAttr'} = 'DN';

AuthSyncModule::LDAP::UserSyncGroupsDefinition

# (If "LDAP" was selected for AuthModule and you want to sync LDAP # groups to rsd groups, define the following.)

$Self->{'AuthSyncModule::LDAP::UserSyncGroupsDefinition'} = { # your ldap group

'cn=agent,o=rsd' => { # rsd group(s) 'admin' => {

# permission rw => 1,

ro => 1,

},

'faq' => {

rw => 0,

ro => 1,

},

},

'cn=agent2,o=rsd' => { 'users' => {

rw => 1,

ro => 1,

},

},

}
```

```
}  
};
```

3. HTTPBasicAuth-аутентификация для Агентов

Если вы хотите внедрить решение «единого входа» для всех своих агентов, вы можете использовать базовую аутентификацию HTTP (для всех ваших систем) и модуль HTTPBasicAuth для RSD (см. Пример ниже).

Пример 4.14. Аутентификация Агентов с помощью HTTPBasic

```
# This is an example configuration for an apache ($ENV{REMOTE_USER}) # auth.  
backend. Use it if you want to have a single login through  
  
# apache http-basic-auth  
  
$Self->{'AuthModule'} = 'Kernel::System::Auth::HTTPBasicAuth';  
  
# Note:  
  
#  
  
# If you use this module, you should use as fallback  
  
# the following configuration settings if the user is not authorized # apache  
($ENV{REMOTE_USER})  
  
$Self->{LoginURL} = 'http://host.example.com/not-authorized-for-rsd.html';  
  
$Self->{LogoutURL} = 'http://host.example.com/thanks-for-using-rsd.html';
```

4. Радиус

Параметры конфигурации, показанные в примере ниже, можно использовать для аутентификации агентов на сервере Radius.

Пример 4.15. Аутентификация (проверка подлинности) агентов с использованием Radius-сервера в качестве хранилища информации

```
# This is example configuration to auth. agents against a radius server  
  
$Self->{'AuthModule'} = 'Kernel::System::Auth::Radius';
```

```
$Self->{'AuthModule::Radius::Host'} = 'radiushost';  
$Self->{'AuthModule::Radius::Password'} = 'radiussecret';
```

2. Бэкенды аутентификации для клиентов

База данных (по умолчанию)

Серверной частью аутентификации пользователей по умолчанию для клиентов в RSD является база данных RSD. С помощью этого бэкенда все данные о клиентах можно редактировать через веб-интерфейс RSD (см. пример ниже).

Пример 4.16. Аутентификация Клиента в Базе Данных

```
# This is the auth. module against the rsd db  
  
$Self->{'Customer::AuthModule'} = 'Kernel::System::CustomerAuth::DB';  
  
$Self->{'Customer::AuthModule::DB::Table'} = 'customer_user';  
  
$Self->{'Customer::AuthModule::DB::CustomerKey'} = 'login';  
  
$Self->{'Customer::AuthModule::DB::CustomerPassword'} = 'pw';    ##$Self->  
>{'Customer::AuthModule::DB::DSN'} =  
  
"DBI:mysql:database=customerdb;host=customerdbhost";           ##$Self->  
>{'Customer::AuthModule::DB::User'} = "some_user";  
  
##$Self->{'Customer::AuthModule::DB::Password'} = "some_password";
```

LDAP

Если у вас есть каталог LDAP со всеми данными ваших клиентов, вы можете использовать модуль LDAP для аутентификации ваших клиентов в RSD (см. Пример ниже). Поскольку этот модуль имеет доступ только для чтения к серверной части LDAP, то редактировать данные клиента через веб-интерфейс RSD невозможно.

Пример 4.17. Аутентификация пользователей с помощью LDAP-бэкенда

```
# This is an example configuration for an LDAP auth. backend. # (make sure  
Net::LDAP is installed!)  
  
$Self->{'Customer::AuthModule'} = 'Kernel::System::CustomerAuth::LDAP';  
  
$Self->{'Customer::AuthModule::LDAP::Host'} = 'ldap.example.com';  
  
$Self->{'Customer::AuthModule::LDAP::BaseDN'} = 'dc=example,dc=com';
```

```

$Self->{'Customer::AuthModule::LDAP::UID'} = 'uid';

# Check if the user is allowed to auth in a posixGroup # (e. g. user needs to be in a
group xyz to use rsd)

$Self->{'Customer::AuthModule::LDAP::GroupDN'} =
'cn=rsdallow,ou=posixGroups,dc=example,dc=com';

$Self->{'Customer::AuthModule::LDAP::AccessAttr'} = 'memberUid'; # for ldap
posixGroups objectclass (just uid)

$Self->{'Customer::AuthModule::LDAP::UserAttr'} = 'UID'; # for non ldap posixGroups
objectclass (full user dn) # $Self->{'Customer::AuthModule::LDAP::UserAttr'} = 'DN';

# The following is valid but would only be necessary if the
# anonymous user does NOT have permission to read from the LDAP tree

$Self->{'Customer::AuthModule::LDAP::SearchUserDN'} = "";
$Self->{'Customer::AuthModule::LDAP::SearchUserPw'} = "";

# in case you want to add always one filter to each ldap query, use
# this option. e. g. AlwaysFilter => '(mail=*)' or AlwaysFilter => '(objectclass=user)'
$Self->{'Customer::AuthModule::LDAP::AlwaysFilter'} = "";

# in case you want to add a suffix to each customer login name, then # you can use
this option. e. g. user just want to use user but
# in your ldap directory exists user@domain.com
#$Self->{'Customer::AuthModule::LDAP::UserSuffix'} = '@domain.com';

# Net::LDAP new params (if needed - for more info see perldoc Net::LDAP)
$Self->{'Customer::AuthModule::LDAP::Params'} = { port => 389,
timeout => 120,
async => 0,

```

```
version => 3,  
};
```

3. HTTPBasicAuth-аутентификация для Клиентов

Если вы хотите внедрить решение «единого входа» для всех пользователей ваших клиентов, вы можете использовать аутентификацию HTTPBasic (для всех ваших систем) и использовать модуль HTTPBasicAuth с RSD (в RSD больше не требуется вход в систему). См. пример ниже.

Пример 4.18. Аутентификация клиентов с помощью HTTPBasic

```
# This is an example configuration for an apache ($ENV{REMOTE_USER}) # auth.  
backend. Use it if you want to have a single login through  
  
# apache http-basic-auth  
  
$Self->{'Customer::AuthModule'} = 'Kernel::System::CustomerAuth::HTTPBasicAuth';  
  
# Note:  
  
# If you use this module, you should use the following  
  
# config settings as fallback, if user isn't login through # apache  
($ENV{REMOTE_USER})  
  
$Self->{CustomerPanelLoginURL} = 'http://host.example.com/not-authorized-for-  
rsd.html';  
  
$Self->{CustomerPanelLogoutURL} = 'http://host.example.com/thanks-for-using-  
rsd.html';
```

4. Радиус

Настройки, показанные в примере ниже, можно использовать для аутентификации ваших клиентов на сервере Radius.

Пример 4.19. Аутентификация клиентов с использованием Radius

```
# This is an example configuration to auth. customer against a radius server  
  
$Self->{'Customer::AuthModule'} = 'Kernel::System::Auth::Radius';  
  
$Self->{'Customer::AuthModule::Radius::Host'} = 'radiushost';
```

```
$Self->{'Customer::AuthModule::Radius::Password'} = 'radiussecret';
```

5.4. Настройка саморегистрации клиента

Есть возможность настроить самостоятельную регистрацию новых клиентов, доступную через панель customer.pl. Можно добавить новые дополнительные или обязательные поля, такие как номер комнаты, адрес или состояние.

В следующем примере показано каким образом можно указать обязательные для заполнения поля в базе данных клиентов, в данном случае, для хранения номера клиента

1. Настройка веб-интерфейса

Для отображения нового поля для номера комнаты в веб-интерфейсе customer.pl необходимо изменить файл .tt, отвечающий за макет в этом интерфейсе. Отредактируйте файл Kernel/Output/HTML/Templates/Standard/CustomerLogin.tt, добавив новое поле вокруг строки 130 (см. сценарий ниже).

```
[...]  
  
<div class="NewLine">  
  
<label for="Room">[% Translate("Room{CustomerUser}") | html %]</label>  
  
<input title="[% Translate("Room Number") | html %]" name="Room" type="text"  
id="UserRoom" maxlength="50" />  
  
</div> [...]
```

Сценарий: Отображение нового поля в веб-интерфейсе.

2. Сопоставление клиентов

На следующем этапе, нужно внести изменения в маппинг (mapping), расширив его новым элементом - номером комнаты. Чтобы убедиться, что изменения не потеряются после обновления, установите настройки "CustomerUser" из файла Kernel/Config/Defaults.pm в файл Kernel/Config.pm. Теперь измените MAP-массив, добавив новое поле - номер комнаты, как это показано в нижеприведенном сценарии.

```
# CustomerUser  
  
# (customer database backend and settings)  
  
$Self->{CustomerUser} = {  
  
  Name => 'Database Backend',
```

```

Module => 'Kernel::System::CustomerUser::DB', Params => {

# if you want to use an external database, add the # required settings

#       DSN => 'DBI:odbc:yourdsn',

#       Type => 'mssql', # only for ODBC connections

#       DSN => 'DBI:mysql:database=customerdb;host=customerdbhost', # User
=> "",

#       Password => "",

        Table => 'customer_user',

    },

# customer unique id CustomerKey => 'login', # customer #

CustomerID => 'customer_id', CustomerValid => 'valid_id',

CustomerUserListFields => ['first_name', 'last_name', 'email'],

# CustomerUserListFields => ['login', 'first_name', 'last_name', 'customer_id',
'email'], CustomerUserSearchFields => ['login', 'last_name', 'customer_id'],
CustomerUserSearchPrefix => "",

CustomerUserSearchSuffix => '*', CustomerUserSearchListLimit => 250,
CustomerUserPostMasterSearchFields => ['email'],

CustomerUserNameFields => ['title', 'first_name', 'last_name'],
CustomerUserEmailUniqCheck => 1,

# # show not own tickets in customer panel, CompanyTickets #
CustomerUserExcludePrimaryCustomerID => 0,

# # generate auto logins # AutoLoginCreation => 0,

# AutoLoginCreationPrefix => 'auto',

# # admin can change customer preferences # AdminSetPreferences => 1,

# # cache time to live in sec. - cache database queries # CacheTTL => 0,

# # just a read only source # ReadOnly => 1,

Map => [

# note: Login, Email and CustomerID needed!

```

var, frontend, storage, shown (1=always,2=lite), required, storage-type, http-link, readonly, http-link-target

```
[ 'UserTitle', 'Title', 'title', 1, 0, 'var', "", 0 ],
[ 'UserFirstname', 'First name', 'first_name', 1, 1, 'var', "", 0 ],
[ 'UserLastname', 'Last name', 'last_name', 1, 1, 'var', "", 0 ],
[ 'UserLogin', 'User name', 'login', 1, 1, 'var', "", 0 ],
[ 'UserPassword', 'Password', 'pw', 0, 0, 'var', "", 0 ],
[ 'UserEmail', 'Email', 'email', 1, 1, 'var', "", 0 ],
[ 'UserCustomerID', 'CustomerID', 'customer_id', 0, 1, 'var', "", 0 ],
[ 'UserPhone', 'Phone', 'phone', 1, 0, 'var', "", 0 ],
[ 'UserFax', 'Fax', 'fax', 1, 0, 'var', "", 0 ],
[ 'UserMobile', 'Mobile', 'mobile', 1, 0, 'var', "", 0 ],
```

```

[ 'UserRoom', 'Room', 'room', 1, 0, 'var', '"', 0 ],
[ 'UserStreet', 'Street', 'street', 1, 0, 'var', '"', 0 ],
[ 'UserZip', 'Zip', 'zip', 1, 0, 'var', '"', 0 ],
[ 'UserCity', 'City', 'city', 1, 0, 'var', '"', 0 ],
[ 'UserCountry', 'Country', 'country', 1, 0, 'var', '"', 0 ],
[ 'UserComment', 'Comment', 'comments', 1, 0, 'var', '"', 0 ],
[ 'ValidID', 'Valid', 'valid_id', 0, 1, 'int', '"', 0 ],

```

```

],

```

```

# default selections Selections => {

```

```

  UserTitle => { 'Mr.' => 'Mr.',

```

```

    'Mrs.' => 'Mrs.',

```

```

  },

```

```

},

```

```

};

```

```

;

```

Сценарий: Внесение изменений в тар-массив

3. Настройка таблицы customer_user в базе данных RSD

Последним шагом является добавление нового столбца номера комнаты в таблицу customer_user в базе данных RSD (см. Сценарий ниже). В этой колонке будет храниться информация для номера комнаты.

```
linux:~# mysql -p Enter password:

Welcome to the MySQL monitor. Commands end with ; or \g.

Your MySQL connection id is 6 to server version: 5.0.18-Debian_7-log Type 'help;' or
'h' for help. Type 'c' to clear the buffer.

mysql> use rsd;

Reading table information for completion of table and column names You can turn off
this feature to get a quicker startup with -A

Database changed

mysql> ALTER TABLE customer_user ADD room VARCHAR (200); Query OK, 3 rows
affected (0.01 sec)

Records: 3 Duplicates: 0 Warnings: 0

mysql> quit Bye linux:~#
```

Сценарий: Добавление новой колонки в таблицу customer_user

Теперь новое поле для комнаты должно отображаться на Информационной панели клиента, если оно заполнено, и на экранах администрирования пользователя Клиента. Кроме того, новые клиенты должны указать номер своей комнаты, если они регистрируют новую учетную запись.

6. Настройки заявок

6.1. Состояния заявок

Предопределенные состояния

RSD позволяет изменять предопределенные состояния заявок и их типы, а также добавлять новые. Для состояния важны два атрибута: имя (state-name) и тип (state-type).

Предустановленные состояния в RSD: "закрыта успешно", "закрыта неуспешно", "объединенные", "новая", "открыта", "ожидает автозакрытия+", "ожидает автозакрытия-", "ожидает напоминания", "удаленная".

1. Новая

Заявки находятся в этом состоянии, когда они создаются на основе входящих сообщений электронной почты.

2. Открыта

Это состояние по умолчанию для заявок, которые присвоены очередям или агентам.

3. Ожидает напоминания

После того как время ожидания истекло, владелец заявки будет получать напоминание на электронную почту. Если заявка не закрыта, то напоминание о заявке будет отправлено всем агентам в очереди. Напоминание о заявках будет отправлено только в рабочее время и будет повторяться каждые 24-часа, пока агент не изменит состояние заявки. Время, которое заявка проведет с таким статусом будет добавлено к времени эскалации.

4. Ожидает автозакрытия -

Если время ожидания вышло, заявки с этим статусом будут установлены в "закрыта неуспешно". Время, проведенное заявкой в этом статусе будет добавлено к времени эскалации.

5. Ожидает автозакрытия +

Если вышло время ожидания, заявки с этим статусом будут установлены в "закрыта успешно". Время, проведенное заявкой в этом статусе будет добавлено к времени эскалации.

6. Объединенные

Это состояние для заявок, которые были объединены с другими заявками.

7. Удаленная

Это состояние для заявок, которые были удалены клиентом. Заявки на самом деле не будут удалены, они просто больше не будут отображаться как открытые. Чтобы включить это состояние в пользовательском интерфейсе, вам необходимо добавить тип состояния «удален» в настройку SysConfig «Ticket::Frontend::CustomerTicketZoom###StateType».

8. Закрыта Успешно

Это конечное состояние для заявок, которые были решены успешно. В зависимости от конфигурации, у вас будет или не будет возможности заново открыть ранее закрытые заявки.

9. Закрыта Неуспешно

Это конечное состояние для заявок, которые НЕ были успешно решены. В зависимости от конфигурации, у вас будет или не будет возможности заново открыть ранее закрытые заявки.

10. Настраиваемые состояния

Каждое состояние имеет имя (имя состояния) и тип (тип состояния). Щелкните ссылку «Состояния» на странице администрирования и нажмите кнопку «Добавить состояние», чтобы создать новое состояние. Вы можете свободно выбрать название нового состояния. Типы состояния не могут быть изменены через веб-интерфейс. База

данных должна быть изменена напрямую, если вы хотите добавить новые типы или изменить существующие имена. Типы состояния по умолчанию обычно не следует изменять, так как это может привести к непредсказуемым результатам. Например, расчеты эскалации и функция разблокировки основаны на определенных типах состояний.

Через этот экран можно изменить имя уже существующего состояния или добавить новые состояния. Если состояние «новая» было изменено через веб-интерфейс, это изменение также необходимо настроить через конфигурационный файл Kernel/Config.pm или через интерфейс SysConfig. Настройки, указанные в приведенном ниже сценарии, необходимо изменить, чтобы RSD работал с измененным состоянием для состояния «новая».

```
[...]  
  
# PostmasterDefaultState  
  
# (The default state of new tickets.) [default: new]  
  
$Self->{PostmasterDefaultState} = 'new';  
  
  
# CustomerDefaultState  
  
# (default state of new customer tickets)  
  
$Self->{CustomerDefaultState} = 'new'; [...]
```

Сценарий: Изменение параметров настройки в Kernel/Config.pm.

Если нужно добавить новый тип состояния, то это можно сделать с помощью клиентской программы управления базами данных, изменив таблицу ticket_state_type в базе данных RSD, как это показано в нижеприведенном сценарии

```
linux:~# mysql -p Enter password:  
  
Welcome to the MySQL monitor. Commands end with ; or \g.  
  
Your MySQL connection id is 23 to server version: 5.0.16-Debian_1-log Type 'help;' or  
'\h' for help. Type '\c' to clear the buffer.  
  
mysql> use rsd;  
  
Reading table information for completion of table and column names You can turn off  
this feature to get a quicker startup with -A
```

```
Database changed

mysql> insert into ticket_state_type (name,comments) values ('own','Own state type');

Query OK, 1 row affected (0.00 sec)

mysql> quit Bye linux:~#
```

Сценарий: Изменение базы данных RSD.

Теперь можно использовать новый тип состояния, который вы только что создали. После того, как состояние было связано с этим новым типом состояния, конфигурация RSD также должна быть изменена, чтобы гарантировать, что новое состояние можно использовать. Используя SysConfig внесите изменения в следующие опции:

Frontend::Agent::Ticket::ViewPhoneNew > AgentTicketPhone####StateDefault

- задает следующее состояние по умолчанию для новых телефонных заявок, созданных на основе телефонного звонка.

Frontend::Agent::Ticket::ViewPhoneNew > AgentTicketPhone####StateType - задает список последующих доступных состояний для новых заявок, созданных на основе телефонного звонка.

Frontend::Agent::Ticket::ViewEmailNew > AgentTicketEmail####StateDefault - задает следующее состояние по умолчанию для новых заявок, созданных на основе сообщений электронной почты.

Frontend::Agent::Ticket::ViewEmailNew > AgentTicketEmail####StateType - задает список последующих доступных состояний для новых заявок, созданных на базе сообщений электронной почты.

Frontend::Agent::Ticket::ViewPhoneOutbound > AgentTicketPhoneOutbound####State - задает следующее состояние по умолчанию для новых телефонных заметок.

Frontend::Agent::Ticket::ViewPhoneOutbound > AgentTicketPhoneOutbound####StateType - задает доступные следующие состояния для новых телефонных заметок.

Frontend::Agent::Ticket::ViewMove > AgentTicketMove####State - задает следующее состояние по умолчанию при перемещении заявки в другую очередь.

Frontend::Agent::Ticket::ViewMove > AgentTicketMove####StateType - задает список последующих доступных состояний при перемещении заявки в другую очередь.

Frontend::Agent::Ticket::ViewBounce > StateDefault - задает следующее состояние заявки по умолчанию после ее возврата.

Frontend::Agent::Ticket::ViewBounce > StateType - задает следующие доступные состояния заявки после ее возврата.

Frontend::Agent::Ticket::ViewBulk > StateDefault - задает следующее состояние по умолчанию при массовом действии.

Frontend::Agent::Ticket::ViewBulk > StateType - задает список последующих доступных состояний при массовом действии.

Frontend::Agent::Ticket::ViewClose > StateDefault - задает следующее состояние по умолчанию после закрытия заявки.

Frontend::Agent::Ticket::ViewClose > StateType - задает список последующих доступных состояний после закрытия заявки.

Frontend::Agent::Ticket::ViewCompose > StateDefault - задает следующее состояние по умолчанию на экране Создания (ответа).

Frontend::Agent::Ticket::ViewCompose > StateType - задает список последующих доступных состояний на экране Создания (ответа).

Frontend::Agent::Ticket::ViewForward > StateDefault - задает следующее состояние для экрана Переслать/Forward.

Frontend::Agent::Ticket::ViewForward > StateType - задает список последующих доступных состояний для экрана Переслать/Forward.

Frontend::Agent::Ticket::ViewForward > StateDefault - задает следующее состояние по умолчанию после редактирования свободных полей заявки.

Frontend::Agent::Ticket::ViewForward > StateType - задает список доступных состояний после редактирования свободных полей заявки

Core::PostMaster > PostmasterDefaultState - задает состояние заявки, созданной из электронного письма.

Core::PostMaster > PostmasterFollowUpState - задает состояние заявки после получения отклика клиента.

Core::PostMaster > PostmasterFollowUpStateClosed - задает состояние заявок после получения отклика клиента на уже закрытую заявку.

Core::Ticket > ViewableStateType - задает типы состояний заявок, которые отображаются в различных обзорах системы, например, в Обзоре очередей.

Core::Ticket > UnlockStateType - задает список типов состояний для разблокированных заявок.

Core::Ticket > PendingReminderStateType - задает тип состояния для заявок с напоминанием.

Core::Ticket > PendingAutoStateType - задает тип состояния для отложенных заявок.

Core::Ticket > StateAfterPending - задает тип состояния для заявок с ожиданием автозакрытия после истечения срока ожидания.

6.2. Приоритеты заявок

RSD поставляется с пятью предустановленными уровнями приоритетов, которые можно изменить перейдя по ссылке "Приоритеты" на Панели Администрирования. При создании настраиваемого списка приоритетов, пожалуйста помните, что они сортируются в алфавитном порядке. Также RSD сортирует заявки в QueueView по их внутреннему номеру (ID).

Примечание

Как и в случае с другими сущностями RSD, приоритеты не могут быть удалены, а только деактивированы путем установки параметра Действительный в значение *недействительный* или *временно недействительный*.

Важно

Если был добавлен новый приоритет или был изменен существующий, вы также можете изменить некоторые значения в SysConfig:

- Ticket:Core::Postmaster::PostmasterDefaultPriority - задает предустановленный приоритет для всех входящих сообщений электронной почты.
- Ticket:Frontend::Agent:Ticket::ViewPhoneNew:Priority - задает приоритет по умолчанию для новых Заявок созданных на основе Телефонных Звонков.
- Ticket:Frontend::Agent:Ticket::ViewEmailNew:Priority - задает приоритет по умолчанию на странице создания новой Email-Заявки в интерфейсе агента.
- Ticket:Frontend::Customer:Ticket::ViewNew:PriorityDefault - задает приоритет по умолчанию при создании заявки клиентом в интерфейсе клиента.

6.3. Ответственность за Заявку & Наблюдение за Заявкой

Начиная с RSD 2.1 и выше, в дополнение к владельцу заявки можно определить ответственного за нее агента. Кроме того, все мероприятия, связанные с заявкой могут просматриваться не только владельцем, но и другими людьми. Эти две возможности системы реализованы с помощью функций TicketResponsible и TicketWatcher и облегчают назначение задач и работу в рамках иерархических командных структур.

1. Ответственность за Заявку

Функция ответственности за заявку облегчает полную обработку заявки агентом, отличным от владельца заявки. Таким образом, агент, заблокировавший заявку, может передать ее другому агенту, не являющемуся владельцем заявки, чтобы второй ответил на запрос клиента. После того, как запрос будет обработан, первый агент может снять ответственность со второго агента.

С помощью параметров конфигурации Ticket::Responsible, можно активировать функцию ответственности за заявку. Это приведет к появлению в верхней части экрана 3-х дополнительных иконок/значков.

Ответственность за заявку можно назначить, вызвав содержимое заявки и нажав на ссылку «Ответственный» в меню действий заявки в интерфейсе агента (см. рис. ниже).

Рисунок 4.81. Изменение Ответственного заявки в меню действий заявки

Заявка#790002583 — Тестовая заявка

Назад Печать Приоритет Дополнительные ITSM поля Решение Агенты Взаимодействие Напоминание Наблюдать Закрыть Разное

Владелец
Ответственный
Клиент

Обзор заметок/сообщений - 2 Заметка(ок)

№	Отправитель	Через	Тема	Создан/а
2	Ксения Смирнова		Заметка	24.01.2023 22:40 (Европа/Москва)
1	Илья Муромец	телефон	Тестовая заявка	24.01.2023 22:36 (Европа/Москва)

#1 – Тестовая заявка – Илья Муромец – 24.01.2023 22:36 (Европа/Москва) через телефон от Ксения Смирнова

Для открытия ссылки в следующем сообщении/заметке необходимо нажать и удерживать клавишу Ctrl или Cmd или Shift и кликнуть по ссылке (зависит от вашего браузера и ОС).

Пометить Печать Разделить Переслать Ответить

Тест

Информация о заявке

Тип: Инцидент
Возраст: 4 мин
Создан/а: 24.01.2023 22:36 (Европа/Москва)
Создал: Ксения Смирнова
Состояние: в работе
Обработка: не обрабатывается
Очередь: 1-я линия поддержки
Приоритет: 3 средний
Дата: 27.01.2023 22:33 исполнения: (Европа/Москва)
ID клиента: ООО Модуль Плюс
Владелец: Ксения Смирнова
Ответственный: Admin

Время

Свое
оплачиваемое: 0 мин
неоплачиваемое: 0 мин

Связан
оплачиваемое: 0 мин
неоплачиваемое: 0 мин

Всего
оплачиваемое: 0 мин
неоплачиваемое: 0 мин

После нажатия на кнопку «Ответственный» откроется всплывающее диалоговое окно для изменения ответственного этой заявки (см. рисунок ниже). Это диалоговое окно также можно использовать для отправки сообщения новому ответственному агенту.

Рисунок 4.82. Всплывающее диалоговое окно для изменения ответственности заявки

Изменить ответственного для Заявка#790002583 — Тестовая заявка

Все поля отмеченные (*) являются обязательными

Отменить и закрыть

▼ Настройки заявок

* Название:

* Новый ответственный:

► Добавить заметку

или

Если активирована функция ответственности за заявку, то в шаблоне «Ответственные заявки» агентского веб-интерфейса RSD можно просмотреть список всех заявок, за которые агент несет ответственность.

2. Просмотр Заявок

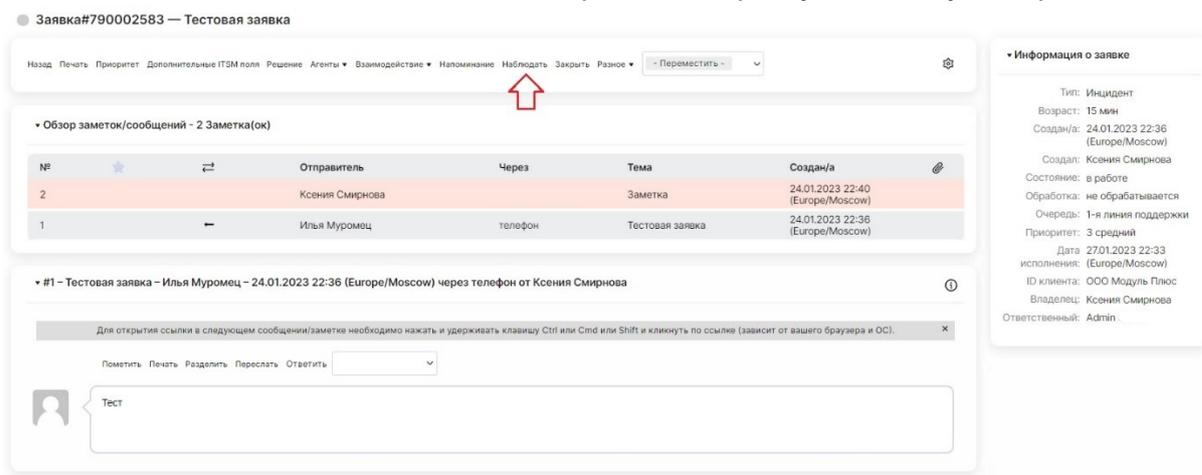
Начиная с RSD 2.1, некоторые агенты, такие как супервайзеры, могут просматривать определенные заявки в системе, не обрабатывая их, с помощью функции TicketWatcher.

Функцию TicketWatcher можно активировать с помощью параметра конфигурации Ticket::Watcher, который добавляет новые ссылки на панель инструментов. С помощью Ticket::WatcherGroup также можно определить одну или несколько групп пользователей с правами просмотра/наблюдения заявок.

Чтобы наблюдать за заявкой, перейдите в ее TicketZoom и нажмите на ссылку «Наблюдать» на панели инструментов (см. рис. ниже).

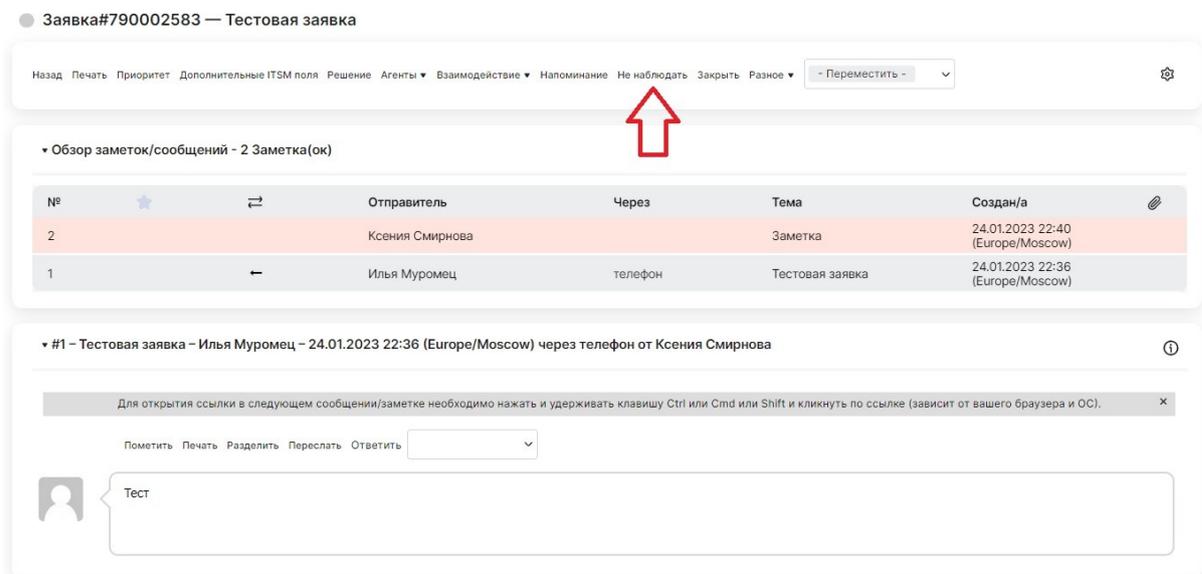
Рисунок 4.83. Подписка на наблюдение заявки в TicketZoom

Если вы больше не хотите смотреть конкретную заявку, перейдите в



ее TicketZoom и нажмите на ссылку «Не наблюдать» в меню действий по заявке (см. рис. ниже).

Рисунок 4.84. Отказ от наблюдения заявки в TicketZoom



Доступ к списку всех отслеживаемых заявок можно получить через просмотренное представление интерфейса агента RSD (см. рисунок ниже), как только активируется функция наблюдения за заявками.

Рисунок 4.85. Просмотр наблюдаемых заявок

Мои наблюдаемые заявки: Все

Все 1 Новое сообщение 0 Напоминание 0 Наступило время Напоминания 0

Массовое действие 1-1 из 1

	Заявка#	▲ Возраст	Отправитель	Название	Состояние T	Обработка T	Очередь T	Владелец T	ID клиента
<input type="checkbox"/>	790002583	27 мин	Илья Муромец	Тестовая заявка	в работе	не обрабатывается	1-я линия поддержки	Ксения Смирнова	ООО Модуль Плюс

7. Функции, связанные с датой и временем

7.1. Настройка рабочих часов, праздников и часовых поясов

Некоторые функции RSD, такие как эскалация и автоматическая разблокировка заявок, зависят от правильной настройки рабочего времени, часовых поясов и праздников. Вы можете определить их через интерфейс SysConfig в Core::Time. Вы также можете указать разные наборы рабочих часов, праздников и часовых поясов в виде отдельных «Календарей» в Core::Time::Calendar1 по Core::Time::Calendar9.

Календари могут быть определены параметрами очереди или уровнями SLA. Это означает, что, например, вы можете указать календарь с рабочими часами 5 x 8 для своего «стандартного» SLA, но создать отдельный календарь с поддержкой 7 x 24 для своего «золотого» SLA; а также установить календарь для вашей очереди «Support-USA» с другим временным окном, чем ваша очередь «Support-Japan». RSD может обрабатывать до 99 различных календарей.

1. Часовые пояса

RSD необходимо знать, какой часовой пояс следует использовать для хранения данных, связанных с датой и временем, в базе данных. Вы можете установить это в Core::Time:RSDTimeZone. По умолчанию используется UTC, и если вы настраиваете новый RSD, настоятельно рекомендуется оставить это значение. Если вы обновлялись с RSD более ранней версии, чем 6, вы должны убедиться, что для RSDTimeZone установлен часовой пояс, соответствующий вашей предыдущей настройке. В противном случае новые данные будут храниться с другим часовым поясом, чем ваши существующие данные. После того, как вы определились с часовым поясом и данные были сохранены (заявки и т. д.), вам больше не следует менять часовой пояс, потому что в противном случае вы получите данные, хранящиеся в разных часовых поясах.

Вы можете установить часовой пояс по умолчанию для новых агентов и пользователей клиентов через Core::Time:UserDefailTimeZone. Этот часовой пояс будет использоваться для всех пользователей, которые не выбрали часовой пояс в своих настройках.

2. Рабочие часы

Настройте рабочее время для вашей системы в SysConfig Core::Time:TimeWorkingHours или для вашего конкретного календаря в конфигурации календаря. RSD может работать с точностью до одного часа. Отметка в полях 8, 9, 10 ... 17 соответствует графику работы с 8:00 до 18:00.

Только в рабочее время заявки могут быть разблокированы, переданы на более высокий уровень, а также отправлены уведомления об эскалированных и ожидающих решениях.

3. Праздники с фиксированной датой

Праздники, приходящиеся на фиксированную дату каждый год, например, Новый год или Четвертое июля, можно указать в TimeVacationDays или в соответствующем разделе для календарей 1-9.

Заявки не будут эскалироваться и не будут разблокированы в даты, указанные в TimeVacationDays.

Примечание

По умолчанию RSD поставляется с установленными немецкими праздниками.

4. Плавающие праздники

Такие праздники, как Пасха, которые не имеют фиксированной даты в году, а меняются каждый год, можно указать в TimeVacationDaysOneTime.

Заявки не будут эскалироваться и не будут разблокированы в даты, указанные в TimeVacation-DaysOneTime.

Примечание

RSD не поставляется с предустановленными разовыми праздниками. Это означает, что вам нужно добавить праздники, такие как Пасха или День Благодарения, в систему при настройке RSD.

7.2. Автоматическая разблокировка

Заблокированные заявки могут быть автоматически разблокированы системой. Эта функция может быть полезна, если, например, агент заблокировал заявки, которые нужно обработать, но он не может работать с ними по какой-либо причине, например, потому что он находится вне офиса в связи с чрезвычайной ситуацией. Функция автоматической разблокировки разблокирует заявки через заданное время, чтобы гарантировать, что ни одна заблокированная заявка не будет забыта, тем самым позволяя другим агентам обрабатывать их.

Время, по истечении которого заявка будет разблокирована, можно указать в настройках каждой очереди. Команда `bin/radiant.Console.pl Maint::Ticket::Unlock`, которая периодически выполняется как задание cron, выполняет автоматическую разблокировку заявок.

Уведомления о разблокированных заявках рассылаются только тем агентам, у которых в «Моих очередях» установлена очередь с разблокированными заявками, и которые активировали уведомление о разблокированных заявках в личных настройках.

Заявки будут разблокированы при соблюдении всех следующих условий:

- Для очереди, в которой находится заявка, определен тайм-аут разблокировки.
- заявка заблокирована.

- Состояние заявки «открыта».

Таймер разблокировки будет сброшен, если агент добавит в заявку новую внешнюю заметку. Это может быть любой из следующих типов: электронная почта-внешняя, телефон, факс, смс или заметка-внешняя.

Кроме того, если последняя заметка в заявке создана агентом, а клиент добавит еще одну через веб-интерфейс или по электронной почте, таймер разблокировки будет сброшен.

Последним событием, которое сбрасывает таймер разблокировки, является назначение заявки другому агенту.

8. Настройка вывода PDF

В этом разделе рассматриваются настраиваемые параметры вывода PDF в RSD.

Если вы используете действие «Печать» из любого места в интерфейсе RSD, оно создаст отформатированный файл PDF.

Вы можете настроить внешний вид файлов, созданных RSD, создав собственный логотип и добавив его в PDF::LogoFile. Вы можете использовать PDF::PageSize, чтобы определить стандартный размер страницы сгенерированного файла PDF (DIN-A4 или Letter), а также PDF::MaxPage, чтобы указать максимальное количество страниц для файла PDF, что полезно, если пользователь по ошибке генерирует огромный выходной файл.

9. Отчеты

Модуль отчетов RSD содержит функции для отслеживания операционной статистики и создания настраиваемых отчетов, связанных с использованием RSD. Система RSD использует термин «отчет» в общем для обозначения одного отчета, представляющего различные показатели.

Правильная настройка модуля отчетов RSD связана с множеством требований и соображений. К ним относятся различные оцениваемые модули RSD, настройки разрешений пользователей, рассчитываемые индикаторы и уровни их сложности, простота настройки модуля отчетов, скорость и эффективность расчетов, а также поддержка богатого набора вариантов вывода.

Статистические элементы, т.е. файлы, которые дополняют функциональность модуля отчетов для конкретных требований, могут быть интегрированы для расчета сложной статистики.

9.1. Конфигурация и использование статистики

При входе в качестве агента модуль отчетов можно открыть, выбрав раздел «Отчеты» в главном меню.

1. Обзор

Выбор ссылки «Отчеты» на панели навигации, а затем ссылки подменю «Обзор» вызывает экран «Обзор». На экране Обзор представлен список всех предварительно настроенных отчетов, которые может использовать агент (см. рисунок ниже).

Рисунок 4.86. Обзор стандартных отчетов.

Stat#	Название	Объект	Экспортировать	Удаление	Выполнить
10001	Changes of status in a monthly overview	StateAction	↓	🗑️	🔄
10002	List of open tickets, sorted by time left until escalation deadline expires	TicketList	↓	🗑️	🔄
10003	Overview about all tickets in the system	TicketAccumulation	↓	🗑️	🔄
10004	List of open tickets, sorted by time left until response deadline expires	TicketList	↓	🗑️	🔄
10005	List of open tickets, sorted by time left until solution deadline expires	TicketList	↓	🗑️	🔄
10006	List of the most time-consuming tickets	TicketList	↓	🗑️	🔄
10007	List of tickets closed last month	TicketList	↓	🗑️	🔄
10008	List of tickets closed, sorted by response time.	TicketList	↓	🗑️	🔄
10009	List of tickets closed, sorted by solution time	TicketList	↓	🗑️	🔄
10010	List of tickets created last month	TicketList	↓	🗑️	🔄

Когда модуль отчетов установлен, в него предварительно загружено несколько примеров отчетов, импортированных в систему. Они отображаются в виде списка на странице обзора. Если обзорный список распространяется более чем на одну страницу, агент может просматривать различные страницы. Список отчетов можно отсортировать по желанию, щелкнув заголовок нужного столбца в списке. Чтобы сгенерировать конкретный отчет, щелкните ссылку отчета «Выполнить сейчас».

2. Генерация

Пользовательский интерфейс просмотра предоставляет настройки конфигурации отчетов (см. рисунок ниже).

Рисунок 4.87. Просмотр определенного отчета.

Управление статистикой > Изменить статистику

Действия

- Перейти к обзору
- Выполнить сейчас

Общая характеристика

- Название:** New Tickets
- Описание:** Total number of new tickets per day and queue which have been created during the last month.
- Права:** stats x
Вы можете выбрать одну или несколько групп для определения доступа для разных агентов.
- Форматы результата:** CSV x | Excel x | Диаграммы: Линейный график x | Диаграммы: левый/центр x | Диаграммы: области с наложением x | Печать x
- Часовой пояс:** Europe/Moscow
Выбранный период времени для отчета не зависит от временной зоны.
- Создать итоговую строку:** Да
Генерирует дополнительную строку, содержащую суммы для всех строк данных.
- Создать итоговую колонку:** Да
Генерирует дополнительную колонку, содержащую суммы по всем колонкам данных.
- Кэширование результатов:** Да
Сохраняет результирующие данные отчета в кэше для использования в последующих обзорах с теми же настройками (требуется наличие хотя бы одного поля типа дата/время).
- Показать как видит Дайджесты:** Нет
Представляет отчет как видит, который агент может активировать в своем Дайджесте. Обратите внимание, что включение виджета статистики в Дайджесте активирует кэширование для этого виджета.
- Действительность:** действительный
Если установлен недействительным, конечные пользователи не могут генерировать этот отчет.

Параметры конфигурации для конкретного отчета могут быть установлены в диапазоне параметров на экране просмотра. Настройки могут выполнять либо создатель отчета, либо любой другой пользователь с соответствующими правами.

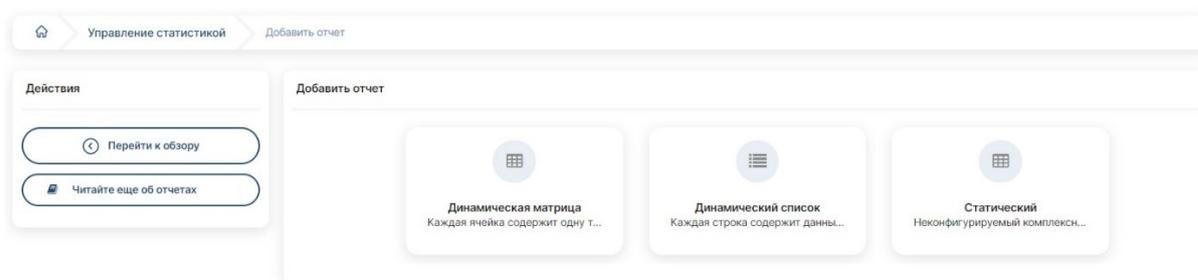
Нажатие кнопки «Выполнить сейчас» (внизу экрана) — последний шаг для генерации отчета.

3. Конфигурация

Агенты с правами записи могут редактировать существующую конфигурацию отчета, вызывая пользовательский интерфейс редактирования модуля отчетов. В качестве альтернативы они могут создать новый отчет.

Существует четыре возможных шага настройки отчетов: общие данные спецификации, настройка оси x, оси y и возможные фильтры данных для отчетных данных (или ограничений). В качестве примера создадим новую статистику, нажав кнопку «Добавить отчет» на обзорном экране. Наша цель будет заключаться в том, чтобы получить общее представление о том, сколько заявок с очень высоким приоритетом находится в каждой очереди (ось X) и состоянии (ось Y).

Рисунок 4.88. Добавление нового отчета, первый шаг.



В начале мы должны выбрать тип отчета, которую мы хотим добавить. Доступны три типа:

Динамическая матрица

Этот тип отчета сгенерирует матрицу вычисленных значений (например, кол-во новых заявок за день месяца и очереди). Все ячейки значений в матрице имеют один и тот же тип (число, среднее время и т.д.). Значения вычисляются из объектов в системе (например, заявок).

Некоторые матричные статистические данные поддерживают столбец и/или строку суммирования (полезно только для определенных данных).

Динамический список

Такого рода статистика сгенерирует таблицу, в которой каждая строка (не ячейка) представляет объект в системе (например, заявку). Столбцы в этой строке обычно настраиваются (ось x, см. Ниже) и содержат данные этого объекта (например, атрибуты заявки). Все ячейки значений в одном столбце имеют один и тот же тип.

Статический

Этот тип отчета не очень легко настраивается и обычно используется для очень специальных и / или сложных вычислений.

Итак, давайте выберем «Динамическую матрицу» для нашего примера. Затем под выбором типа статистики появится конфигурация «Общая характеристика».

Рисунок 4.89. Добавление нового отчета, второй шаг.

The screenshot shows a web form titled "Общая характеристика" (General Characteristics) for configuring a report. The form includes the following fields and options:

- * Название:** (Name) - empty text input field.
- * Описание:** (Description) - empty text area with a rich text editor icon.
- * Тип объекта:** (Object Type) - dropdown menu with "ITSMChangeManagement" selected.
- * Права:** (Rights) - dropdown menu with "stats x" selected. Below it, a note says: "Вы можете выбрать одну или несколько групп для определения доступа для разных агентов." (You can select one or more groups to define access for different agents.)
- * Форматы результата:** (Result Formats) - dropdown menu with "CSV X", "Excel X", and "и 4 более..." (and 4 more...) options.
- * Часовой пояс:** (Time Zone) - dropdown menu with "Europe/Moscow" selected. Below it, a note says: "Выбранный период времени для отчета не зависит от временной зоны." (The selected time period for the report does not depend on the time zone.)
- Создать итоговую строку:** (Create Summary Row) - dropdown menu with "Нет" (No) selected. Below it, a note says: "Генерирует дополнительную строку, содержащую суммы для всех строк данных." (Generates an additional row containing sums for all data rows.)
- Создать итоговую колонку:** (Create Summary Column) - dropdown menu with "Нет" (No) selected. Below it, a note says: "Генерирует дополнительную колонку, содержащую суммы по всем колонкам данных." (Generates an additional column containing sums for all data columns.)
- Действительность:** (Validity) - dropdown menu with "действительный" (valid) selected. Below it, a note says: "Если установлен недействительным, конечные пользователи не могут генерировать этот отчет." (If set to invalid, end users cannot generate this report.)

At the bottom of the form, there is a "Создать отчет" (Create Report) button and a "Сохранить" (Save) button with an "или Отменить" (or Cancel) link.

После предоставления названия и описания для нового отчета мы должны выбрать тип объекта, который мы хотим использовать. Это фактический внутренний модуль, который отвечает за сбор и анализ данных для нашего отчета. В нашем случае мы выберем «TicketAccumulation».

Настроив группы разрешений, мы можем облегчить ограничение групп (и, следовательно, агентов), которые впоследствии могут просматривать и генерировать предварительно настроенный отчет. Таким образом, различные статистические данные могут быть распределены между различными отделами и рабочими группами, которые в них нуждаются. Можно выделить один отчет для различных групп.

Пример 4.20. Группа разрешений статистики по умолчанию

Была выбрана группа «stats». Отчет доступен для просмотра всем пользователям, имеющим права хотя бы ro для группы "stats". Этот доступ доступен по умолчанию.

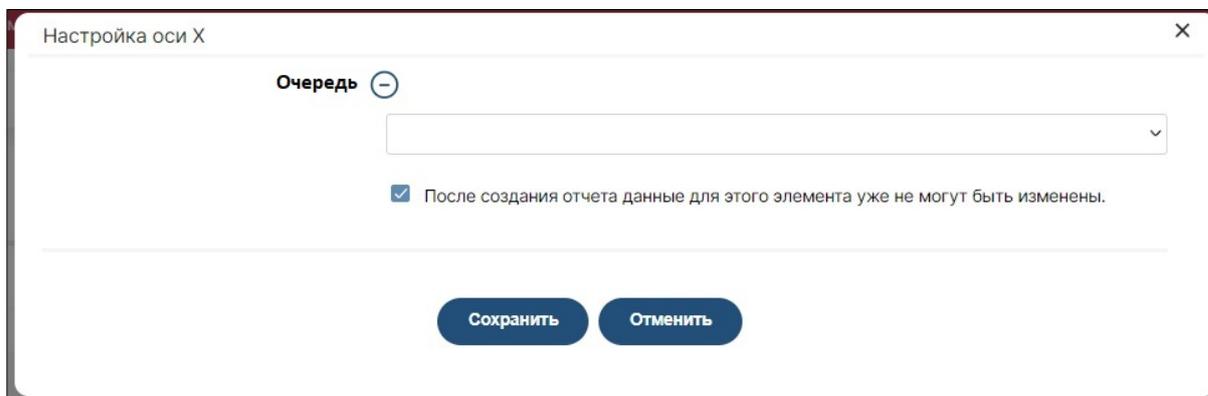
Пример 4.21. Настраиваемая группа разрешений статистики

Была выбрана группа под названием «Продажи». Все пользователи с правами ro для группы "продажи" могут видеть отчеты в режиме просмотра и генерировать их. Однако отчеты не будут доступны для просмотра другим пользователям.

Кроме того, можно выбрать возможные выходные форматы. Здесь мы можем просто сохранить все выходные форматы и выбрать тот, который будет использоваться при фактическом создании отчета. Давайте сохраним отчет сейчас.

Следующий экран покажет следующий шаг с выделенной кнопкой: мы должны настроить ось X. Нажав кнопку, появится диалоговое окно, в котором мы можем выбрать элемент, который будет использоваться для оси x. В нашем случае это будет очередь:

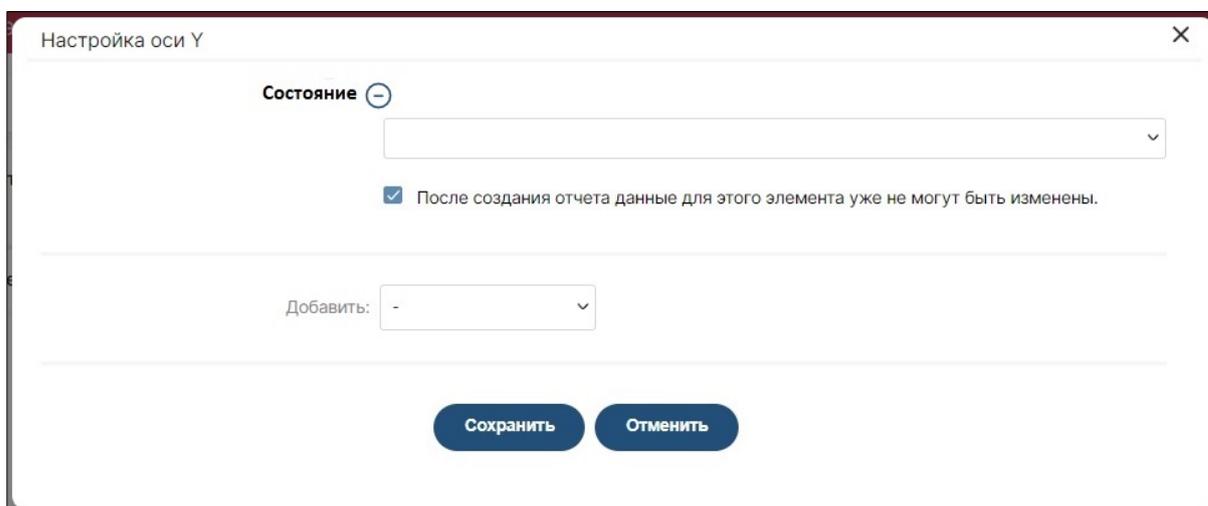
Рисунок 4.90. Настройка оси x отчета.



При желании мы можем ограничить отображаемые очереди, выбрав некоторые из них в поле очереди. С помощью флажка мы можем контролировать, может ли агент, который генерирует отчет, вносить изменения в выбор очереди. Мы сохраним значения по умолчанию и нажмем «Сохранить».

Теперь мы можем настроить ось Y таким же образом: выберем поле состояния.

Рисунок 4.91. Настройка оси Y отчета.

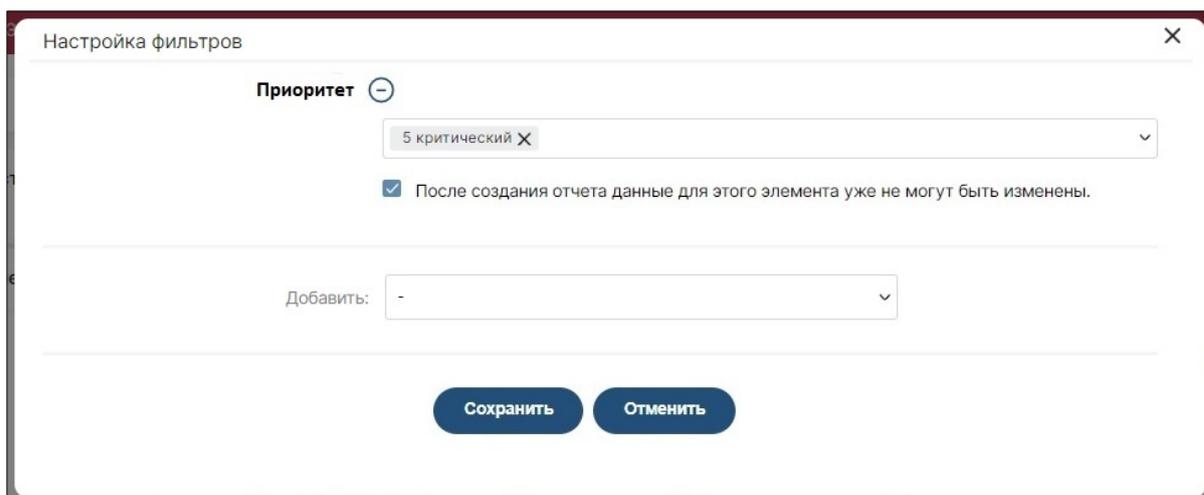


Здесь можно выбрать один элемент или два. В первом случае каждое значение элемента будет одним элементом на оси Y. Если выбраны два элемента, их перестановки будут элементами ряда значений. Например, вы можете выбрать «состояние» и «приоритет», и результирующие элементы будут «новая — 1 очень низкий», «новая — 2 низкий», ... «открыта — 1 очень низкий» и так далее. Давайте просто воспользуемся состоянием и нажмем «Сохранить».

Теперь, на последнем шаге, мы можем добавить фильтры данных, чтобы сообщать только о заявках, принадлежащих определенному клиенту, с определенными

приоритетами и так далее. Мы добавим фильтр для заявок с очень высоким приоритетом:

Рисунок 4.92. Настройка фильтра данных отчета.



Теперь снова нажмите «Сохранить». Конфигурация завершена.

Возможно, вы уже заметили, что в диалоговом окне конфигурации есть область предварительного просмотра, где мы можем проверить действие наших настроек конфигурации.

Рисунок 4.93. Настройка фильтра данных отчета.



Примечание

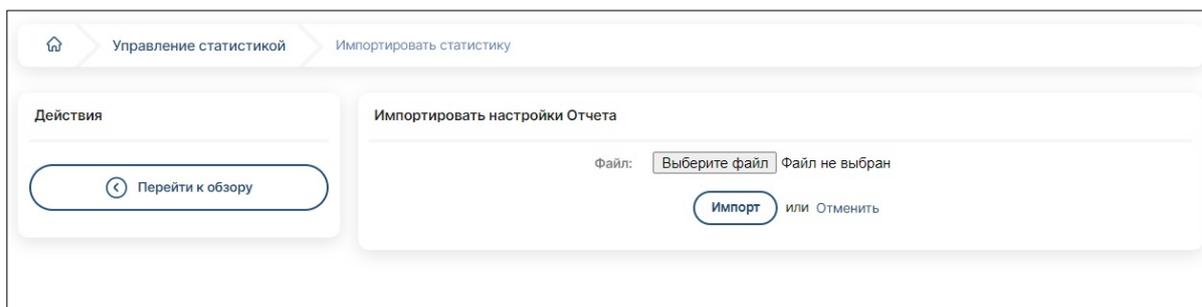
Обратите внимание, что предварительный просмотр использует случайные данные и не учитывает ограничения данных.

Отчет настроен. Нажав кнопку «Выполнить сейчас», мы можем перейти к экрану просмотра, где можно выбрать желаемый формат вывода и создать отчет в различных форматах.

4. Импорт

Доступ к пользовательскому интерфейсу импорта можно получить, нажав кнопку «Импортировать статистику» на экране «Обзор». Требуется права "rw" для модуля отчетов.

Рисунок 4.94. Импорт статистики



Облегчает импорт отчетов, а в сочетании с функцией экспорта модуля представляет собой очень удобную функцию. Статистику можно удобно создавать и тестировать на тестовых системах, а затем импортировать в производственную систему.

9.2. Администрирование системы статистики

В этом разделе представлена информация о задачах и обязанностях администратора RSD, работающего с модулем отчетов.

1. Настройки разрешений, группы и очереди

Конфигурация регистрации модуля по умолчанию предоставляет всем агентам с разрешениями группы «stats» доступ к модулю отчетов.

Доступ в соответствии с разрешенными настройками:

- rw. Позволяет настроить отчеты.
- ro. Позволяет генерировать предварительно настроенные отчеты.

Администратор RSD решает, будут ли агенты с разрешением генерировать предварительно настроенные отчеты наделены правами ro в группе «stats» или их соответствующие группы добавятся в модуль регистрации в SysConfig.

2. Настройки SysConfig

Группы SysConfig Core::Stats и Frontend::Agent::View::Stats содержат все параметры конфигурации для базовой настройки модуля отчетов. Кроме того, параметр конфигурации `$Self->{'Frontend::Module'}->{'AgentStats'}` управляет расположением и регистрацией модулей и значков в модуле отчетов.

3. Генерация статистики в командной строке

Отчет можно сгенерировать в командной строке с помощью команды `bin/radiant.Console.pl Maint::Stats::Generate`. В качестве примера см. вызов командной строки в следующем скрипте.

```
shell> bin/radiant.Console.pl Maint::Stats::Generate --number 10004 --target-directory /tmp
Generating statistic number 10004...

Writing file /tmp/List_of_the_most_time-consuming_tickets_Created_2015-09-08_14-51.csv.

Done.
```

Отчет из статистической конфигурации "Stat#10004" создается и сохраняется в виде CSV-файла в каталоге `/tmp`.

Сформированный отчет также можно отправить по электронной почте. Дополнительную информацию можно получить с помощью команды в приведенном ниже сценарии.

```
shell> bin/radiant.Console.pl Maint::Stats::Generate --help
```

Генерировать отчеты вручную через командную строку обычно не имеет смысла, так как модуль отчетов имеет удобный графический интерфейс. Однако создание отчетов вручную имеет смысл в сочетании с заданием `cron`.

Представьте себе следующий сценарий: в первый день каждого месяца начальники отделов хотят получить отчет за прошедший месяц. Комбинируя задание `cron` и вызов командной строки, отчеты могут автоматически отправляться им по электронной почте.

10. Динамические Поля

10.1. Введение

Динамическое поле — это поле особого типа в RSD, созданное для расширения информации, хранящейся в заявке или заметке. Эти поля не зафиксированы в системе, и они могут появляться только на определенных экранах, они могут быть обязательными или нет, и их появление на экранах зависит от типа поля, определенного во время их создания в соответствии с данными, которые должны содержаться в поле. Например, поля могут содержать текст, даты, набор из нескольких выбранных значений и т.д.

Динамические поля — это эволюция полей `TicketFreeText`, `TicketFreeKey`, `TicketFreeTime`, `ArticleFreeText` и `ArticleFreeKey`, которые широко использовались в RSD 3.0 и более ранних версиях. Ограничение этих «Свободных полей» заключалось в том, что они могут быть определены до 16 (текстовых или раскрывающихся) полей и 6 полей времени для заявки и только 3 (текстовых или раскрывающихся) полей для каждой заметки, не более.

Теперь, с появлением динамических полей, эти ограничения сняты, вы можете создавать любое количество динамических полей для заявок и сообщений. И, кроме того, интерфейс для управления ими, готов к расширению дальнейшего использования таких полей не только для заявок и сообщений, но и других объектов системы, например, для управления изменениями и др.

Этот новый интерфейс описания и управления динамическими полями, использует модульную структуру, в которой каждый тип динамических полей может рассматриваться как дополнительный модуль для интерфейса. Это означает, что все множество динамических полей может быть просто расширено для публичных модулей RSD, RSD Feature Addons, пользовательских разработок и пр.

В этом релизе доступны следующие типы динамических полей:

- Text (одна строка текста)
- Textarea (несколько строк текста)
- Checkbox
- Выпадающий список (единственный выбор, несколько значений)
- Multiselect (множественный выбор, несколько значений)
- Date
- Date / Time

10.2. Настройка

По умолчанию чистая установка RSD поставляется с двумя предустановленными внутренними динамическими полями, которые нельзя удалить. Если вы планируете использовать такие поля в заявках, заметках, пользователях-клиентах или компаниях-клиентах, вам необходимо создать новые настраиваемые динамические поля. Обратите внимание, что динамические поля для пользователей-клиентов и компаний должны быть соответствующим образом настроены в сопоставлении CustomerUser и CustomerCompany, обычно это можно найти в Kernel/Config.pm вашей установки RSD. Примеры их настройки можно найти в Kernel/Config/Defaults.pm.

Конфигурация динамического поля разделена на две части, чтобы добавить новое динамическое поле или управлять существующим, вам необходимо перейти в панель «Администрирование» по ссылке «Динамические поля». Для управления показом, обязательностью показа или скрытием динамических полей на одном экране, вам нужно изменить настройки RSD на экране «SysConfig».

1. Добавление Динамического Поля

Нажмите кнопку «Администрирование», расположенную на панели навигации, затем нажмите ссылку «Динамические поля» внутри раздела «Настройки заявки», расположенного в центре нижней части экрана. Обзор динамических полей будет выглядеть следующим образом:

Рисунок 4.95. Экран обзора динамических полей.

Управление динамическими полями

Действия

Заявка
Добавить новое поле для объекта: Заявка

Сообщение
Добавить новое поле для объекта: Сообщение

Клиент
Добавить новое поле для объекта: СвойствоСервера

Учетная запись клиента
Добавить новое поле для объекта: Клиент

База знаний
Добавить новое поле для объекта: База знаний

ITSMChange
Добавить новое поле для объекта: ITSMChange

ITSMWorkOrder
Добавить новое поле для объекта: ITSMWorkOrder

Список динамических полей

Название	Название	Порядок	Тип	Объект	Действительность	Удаление
ProcessManagementProcessID	Процесс	1	ProcessID	Заявка	действительный	
ProcessManagementActivityID	Активность	2	ActivityID	Заявка	действительный	
ITSMCriticality	Критичность	3	Выпадающий список	Заявка	действительный	
ITSMImpact	Влияние	4	Выпадающий список	Заявка	действительный	
ITSMReviewRequired	Необходим просмотр	5	Выпадающий список	Заявка	действительный	
ITSMDecisionResult	Результат решения	6	Выпадающий список	Заявка	действительный	
ITSMRepairStartTime	Дата начала работ	7	Дата/Время	Заявка	действительный	
ITSMRecoveryStartTime	Дата восстановления сервиса	8	Дата/Время	Заявка	действительный	
ITSMDecisionDate	Дата решения	9	Дата/Время	Заявка	действительный	
ITSMDueDate	Дата исполнения	10	Дата/Время	Заявка	действительный	
CalendarStart	CalendarStart	11	Дата/Время	Заявка	действительный	✕
CalendarFinish	CalendarFinish	12	Дата/Время	Заявка	действительный	✕
ClosingMark	ClosingMark	13	Выпадающий список	Заявка	действительный	✕

Обратите внимание, что этот экран будет меняться по мере добавления новых динамических полей. На этом экране уже могут быть некоторые поля, если установка была обновлена из более старой версии RSD.

Действия на боковой панели в левой части экрана описывают четыре возможности: Сообщение, Заявка, Клиент и Учетная запись клиента, каждая из которых имеет собственный раскрывающийся список динамических полей.

Примечание

Установка некоторых дополнительных пакетов RSD может привести к добавлению типов объектов в разделе "Действия".

Общая процедура создания динамического поля:

- Нажмите на нужный объект динамического поля в раскрывающемся списке на боковой панели «Действия».
- Щелкните тип динамического поля, который вы хотите добавить из списка.
- Заполните конфигурацию.
- Сохраните.

Диалоговые окна конфигурации для динамических полей разделены на две части, верхняя часть является общей для всех полей, а нижняя часть может отличаться от одного типа динамического поля к другому.

Общие настройки динамического поля:

- **Название:** Обязательное, уникальное, разрешены только буквы и цифры.

Это внутреннее имя поля, используемое, например, для отображения или скрытия поля на одном экране. Любое изменение имени поля (не рекомендуется) требует ручного обновления настроек «SysConfig», в которых есть ссылка на поле.

Метка (Название): обязательное.

Это имя поля, которое будет отображаться на экранах, оно поддерживает переводы.

Примечание

Переводы меток необходимо добавлять вручную в файлы языковых переводов.

- Порядок поля: обязательное.

Определяет относительный порядок, в котором поле будет отображаться на экране, по умолчанию каждое новое поле имеет последнюю позицию, изменение этого параметра повлияет на порядок других создаваемых динамических полей.

- Действительность: обязательное.

Недействительное динамическое поле не будет отображаться ни на одном экране, независимо от того, настроено ли оно для отображения.

- Тип поля: Обязательное, только для чтения.

Показывает текущий выбранный тип поля.

- Тип объекта: обязательное, только для чтения.

Показывает область поля.

Примечание

Чтобы проиллюстрировать настройки каждого конкретного типа поля, в нашем примере будет добавлено несколько полей. Эти новые поля будут упомянуты в последующих разделах.

Для следующих примеров все динамические поля будут созданы для объекта Заявка. Если вам нужно создать динамическое поле для объекта «Сообщение», просто выберите поле из выпадающего списка «Сообщение».

Таблица 4.6. В систему будут добавлены следующие поля:

Название	Метка	Тип
TestText	TestText	Текст
TestTextArea	TestTextArea	Textarea/Длинный текст
TestCheck	TestCheck	Чекбокс
TestDropdown	TestDropdown	Выпадающий список
TestMulti	TestMulti	Multiselect - множественный выбор

TestData	TestData	Дата
TestDataTime	TestDataTime	Дата/Время

2. Настройка Динамического Поля типа Текст

Текстовое динамическое поле используется для хранения одной простой строки. Настройки динамического поля типа Текст:

- Значение по умолчанию: необязательное.

Это значение, которое будет отображаться по умолчанию на экранах редактирования (например, New Phone Ticket/Новая телефонная заявка или Ticket Compose/Создать ответ).

- Показывать ссылку: необязательное.

Если установлено, то значение поля будет преобразовано в кликабельную ссылку, которая будет отображаться на экране.

Например, если "Показывать ссылку" установлено в <https://radiantsystem.ru/> нажатие на заполненное значение приведет к открытию веб-страницы RSD.

Примечание

Использование [% Data.NameX | uri %] в Set link value, где NameX — это имя поля, добавит значение поля как часть ссылки на ссылку.

Рисунок 4.96. Окно конфигурации Динамического поля типа Текст

The screenshot shows a configuration interface for a dynamic text field. The main title is 'Управление динамическими полями' (Manage dynamic fields) with a sub-title 'Заявка: Добавить Текст поле' (Request: Add Text field). On the left, there is a 'Действия' (Actions) section with a button 'Вернуться к обзору' (Return to overview). The main area is divided into two sections:

- Общие (General):**
 - Название (Name):** 'TestText'. A note states: 'Должно быть уникальным, и может состоять только из букв и цифр.' (Must be unique and can only consist of letters and numbers).
 - Действительность (Validity):** 'действительный' (valid).
 - Тип поля (Field type):** 'Текст' (Text).
 - Тип объекта (Object type):** 'Заявка' (Request).
 - Порядок поля (Field order):** '14'. A note states: 'Это порядок, в котором поле будет показываться среди других полей на тех экранах, где оно активно.' (This is the order in which the field will be displayed among other fields on the screens where it is active).
- Текст Настройки поля (Text Field Settings):**
 - Значение по умолчанию (Default value):** An empty text input field. A note below it says: 'Это значение по умолчанию для данного поля.' (This is the default value for this field).
 - Показывать ссылку (Show link):** An empty text input field.
 - Ссылка для предпросмотра (Preview link):** An empty text input field. A note below it says: 'Эта ссылка будет использоваться для предпросмотра, который будет отображаться по наведению на нее в экране заявки. Чтобы эта опция работала, поле со ссылкой выше также должно быть заполнено.' (This link will be used for preview, which will be displayed on hovering over it in the request screen. For this option to work, the link field above must also be filled).
 - Регулярное выражение для проверки (Regular expression for validation):** An empty text input field. A note below it says: 'Здесь можно задать регулярное выражение для проверки значения. Оно будет выполняться с расширением xms. Пример: "[0-9]5' (You can specify a regular expression for value validation here. It will be executed with the xms extension. Example: "[0-9]5').

At the bottom, there are buttons for 'Сохранить' (Save) and 'Отменить' (Cancel).

3. Настройка Динамического Поля типа Textarea/Длинный текст

Динамическое поле Textarea используется для хранения текста, состоящего из нескольких строк.

Настройки динамического поля Textarea:

- Число строк: Необязательное, Целое.

Используется для определения высоты поля на странице редактирования.

- Число столбцов: Необязательное, Целое.

Это значение используется для задания ширины поля при редактировании.

- Значение по умолчанию: Необязательное.

Это значение будет отображаться по умолчанию (может быть текстом, состоящим из нескольких строк).

Рисунок 4.97. Окно конфигурации Динамического поля типа Textarea

Управление динамическими полями | Заявка: Добавить Textarea/Длинный текст поле

Действия

Вернуться к обзору

Общие

- Название: TestTextArea
- Действительность: действительный
- Название: Должно быть уникальным, и может состоять только из букв и цифр. TestTextArea
- Тип поля: Textarea/Длинный текст
- Порядок поля: 14
- Тип объекта: Заявка

Textarea/Длинный текст Настройки поля

Число строк: [input type="text"]

Задает высоту (в строках) для этого поля в режиме редактирования.

Число столбцов: [input type="text"]

Задает ширину (в символах) для этого поля в режиме редактирования.

Значение по умолчанию: [input type="text"]

Это значение по умолчанию для данного поля.

Регулярное выражение для проверки: [input type="text"]

Здесь можно задать регулярное выражение для проверки значения. Оно будет выполняться с расширением xls.

Пример: *{0-9}\$

Добавить регулярное выражение: [input type="text"]

Сохранить или Отменить

4. Настройка Динамического Поля типа Чекбокс

Динамическое поле типа Чекбокс используется для хранения значений true или false, изображено как выбранный или не выбранный флажок.

Настройки динамического поля Checkbox:

- Значение по умолчанию: Обязательное.

Это значение, которое будет показано по умолчанию на страницах редактирования (например, заявка, созданная на основе телефонного звонка), значение по умолчанию для этого поля недоступно для выбора и может принимать значения Выбрано/Checked или Unchecked/Не выбрано.

Рисунок 4.98. Окно конфигурации Динамического поля типа Чекбокс

5. Настройка Динамического Поля типа Выпадающий список

Динамическое поле типа Выпадающий Список (Dropdown) используется для хранения одного значения, которое выбирается из закрытого списка.

Настройки динамического поля типа Выпадающий список (Dropdown):

- Возможные значения: Обязательное.

Список значений для выбора. При добавлении нового значения необходимо указать Ключ (внутреннее значение) и Значение (значение, которое будет отображено).

- Значение по умолчанию: Необязательное.

Это значение будет отображено по умолчанию на страницах редактирования (например, на форме создания новой телефонной заявки или на странице создания заметки), значение по умолчанию для этого поля - задается параметром "Значение по умолчанию" из заданного списка значений.

- Добавить пустое значение: Обязательное, (Да / Нет).

Если эта опция включена, задается возможность для отображения "-" в списке возможных значений, это специальное значение для указания пустого значения.

- Переводимые значения: Обязательное, (Да / Нет).

Этот параметр задает возможность перевода на язык системы значений этого поля. Переводятся только отображаемые значения, внутренние значения(ключи) не подлежат переводу. Перевод полей должен быть занесен вручную в соответствующий пользовательский файл локализации. Для русского языка - ru_custom.pm.

- Показывать ссылку: Необязательное.

Если установлено, то значение поля будет преобразовано в кликабельную ссылку, которая будет отображаться на экране.

Например, если в поле "Показывать ссылку" установлено <https://radiantsystem.ru/>, то нажатие на заполненное значение приведет к открытию веб-страницы RSD.

Примечание

Использование [% Data.NameX | uri %] в Set link value, где NameX — это имя поля, добавит значение поля как часть ссылки на ссылку.

Рисунок 4.99. Окно конфигурации Динамического поля типа, Выпадающий список

The screenshot shows a configuration window titled "Управление динамическими полями" (Management of dynamic fields) with a sub-header "Заявка: Добавить Выпадающий список поле" (Request: Add Dropdown list field). The window is divided into several sections:

- Общие (General):** Contains fields for "Название" (Name) set to "TestDropdown", "Действительность" (Validity) set to "действительный" (valid), "Тип поля" (Field type) set to "Выпадающий список" (Dropdown list), and "Тип объекта" (Object type) set to "Заявка" (Request). A "Порядок поля" (Field order) is set to "14".
- Выпадающий список Настройки поля (Dropdown list Field Settings):** This section includes:
 - Возможные значения (Possible values):** A table with three rows: "Ключ: 1" (Key: 1) with "Значение: Значение 1" (Value: Value 1), "Ключ: 2" (Key: 2) with "Значение: Значение 2" (Value: Value 2), and "Ключ: 3" (Key: 3) with "Значение: Значение 3" (Value: Value 3). There is a "Добавить значение" (Add value) button.
 - Значение по умолчанию (Default value):** A dropdown menu set to "Нет" (None).
 - Добавить пустое значение (Add empty value):** A dropdown menu set to "Нет" (None).
 - Иерархический вид (Hierarchical view):** A dropdown menu set to "Нет" (None).
 - Переводимые значения (Translatable values):** A dropdown menu set to "Нет" (None).
 - Показывать ссылку (Show link):** A checkbox that is currently unchecked.
 - Ссылка для предпросмотра (Preview link):** A text input field.

At the bottom of the window, there are buttons for "Сохранить" (Save) and "Отменить" (Cancel).

6. Настройка Динамического Поля типа Multiselect - множественный выбор

Динамическое поле Multiselect используется для хранения нескольких значений из заданного списка.

Настройка Динамического Поля Multiselect (множественный выбор):

- Возможные значения: Обязательное.

Список значений для выбора. При добавлении нового значения в список, необходимо указать Ключ (внутреннее значение) и Значение (значение, которое будет отображено).

- Значение по умолчанию: Необязательное.

Это значение будет отображено по умолчанию на страницах редактирования (например, на форме создания новой телефонной заявки или на странице создания заметки), значение по умолчанию для этого поля - задается параметром "Значение по умолчанию" из заданного списка значений.

- Добавить пустое значение: Обязательное, (Да / Нет).

Если эта опция включена, задается возможность для отображения "-" в списке возможных значений, это специальное значение для указания на пустое значение.

- Переводимые значения: Обязательное, (Да / Нет).

Этот параметр задает возможность перевода на язык системы значений этого поля. Переводятся только отображаемые значения, внутренние значения не подлежат переводу. Перевод полей должен быть занесен вручную в соответствующий пользовательский файл локализации. Для русского языка - ru_custom.pm.

Рисунок 4.100. Окно конфигурации Динамического поля типа Multiselect - множественный выбор

7. Настройка Динамического Поля типа Дата

Динамическое поле типа Дата используется для хранения значения даты (День, Месяц и Год).

Настройки динамического поля Date (Дата):

- Диапазон дат по умолчанию: Необязательное, Целое.

Количество секунд (положительное или отрицательное значение) между текущей датой и выбранной датой, которое будет отображаться на страницах редактирования по умолчанию (например, на странице «Новая Телефонная Заявка» или «Создать ответ»).

- Задать период лет, (Да / Нет).

Используется для установки определенного количества лет в прошлом и будущем на основе текущей даты, выбранной для этого поля. Если установлено значение «Да», доступны следующие параметры:

- Лет в прошлом: Необязательное, Положительное целое число.

Задаёт количество лет в прошлом от текущего дня для отображения в выборе года для этого динамического поля на экранах редактирования.

- Лет в будущем: Необязательное, Положительное целое число.

Задаёт количество лет в будущем от текущего дня для отображения в выборе года для этого динамического поля на экранах редактирования.

- Показывать ссылку: Необязательное.

Если установлено, то значение поля будет преобразовано в кликабельную ссылку, которая будет отображаться на экране.

Например, если в поле "Показывать ссылку" установлено <https://radiantsystem.ru/>, то нажатие на заполненное значение приведет к открытию веб-страницы RSD.

Примечание

Использование [% Data.NameX | uri %] в Set link value, где NameX — это имя поля, добавит значение поля как часть ссылки на ссылку.

Рисунок 4.101. Окно конфигурации Динамического поля типа Дата

8. Настройка Динамического Поля типа Дата/Время

Динамическое поле типа Дата/Время используется для хранения значения даты и времени (Минута, Час, День, Месяц и Год).

Настройки динамического поля Date / Time:

- Диапазон дат по умолчанию: Необязательное, Целое.

Количество секунд (положительное или отрицательное значение) между текущей датой и выбранной датой, которое будет отображаться на страницах редактирования по умолчанию (например, на странице «Новая Телефонная Заявка» или «Создать ответ»).

Задать период лет, (Да / Нет).

Используется для установки определенного количества лет в прошлом и будущем на основе текущей даты, выбранной для этого поля. Если установлено значение «Да», доступны следующие параметры:

- Лет в прошлом: Необязательное, Положительное целое число.

Задаёт количество лет в прошлом от текущего дня для отображения в выборе года для этого динамического поля на экранах редактирования.

- Лет в будущем: Необязательное, Положительное целое число.

Задаёт количество лет в будущем от текущего дня для отображения в выборе года для этого динамического поля на экранах редактирования.

- Показывать ссылку: Необязательное

Если установлено, то значение поля будет преобразовано в кликабельную ссылку, которая будет отображаться на экране.

Например, если в поле "Показывать ссылку" установлено <https://radiantsystem.ru/>, то нажатие на заполненное значение приведет к открытию веб-страницы RSD.

Примечание

Использование [% Data.NameX | uri %] в Set link value, где NameX — это имя поля, добавит значение поля как часть ссылки на ссылку.

Рисунок 4.102. Окно конфигурации Динамического поля типа Дата/Время

9. Редактирование Динамического Поля

Страница просмотра заполненных динамических полей (с предыдущими примерами) должна выглядеть так:

Рисунок 4.103. Страница просмотра заполненных динамических полей с тестовыми данными

Название	Название	Порядок	Тип	Объект	Действительность	Удаление
TestText	TestText	1	Текст	Заявка	действительный	🗑️
TestTextArea	TestTextArea	2	Текстarea/Длинный текст	Заявка	действительный	🗑️
TestCheck	TestCheck	3	Checkbox	Заявка	действительный	🗑️
TestDropdown	TestDropdown	4	Выпадающий список	Заявка	действительный	🗑️
TestMulti	TestMulti	5	Multiselect - множественный выбор	Заявка	действительный	🗑️
TestData	TestData	6	Дата	Заявка	действительный	🗑️
TestDataTime	TestDataTime	7	Дата/Время	Заявка	действительный	🗑️

Чтобы изменить или отредактировать динамическое поле, у вас должно быть определено хотя бы одно поле, выберите уже добавленное поле на экране обзора динамических полей и обновите его настройки.

Примечание

Не все настройки динамического поля можно изменить, Тип поля и Тип объекта фиксируются при выборе поля и не могут быть изменены.

Внутреннее имя поля менять не рекомендуется, но метку можно изменить в любой момент. Если внутреннее имя изменено, необходимо обновить все настройки «SysConfig», которые имеют ссылку на это конкретное поле, а также пользовательские настройки (если они определены).

10. Отображение Динамического Поля на Экране

Есть два обязательных условия, которые необходимо выполнить, чтобы отобразить динамическое поле на определенных страницах:

1. Динамическое поле должно быть действительным
2. Динамическое поле должно быть установлено в 1 или 2 на странице с настройками.

Следуйте этим шагам чтобы отобразить динамическое поле на экране

- Убедитесь, что динамическое поле установлено как действительное, вы можете увидеть действительность поля на экране обзора динамического поля. Установите действительность, отредактировав поле, если это необходимо.
- Откройте раздел "SysConfig"/Конфигурация системы выберите подраздел "Ticket» из выпадающего списка на боковой панели «Действия», расположенной в левой части экрана.

Примечание

Вы также можете выполнить поиск «DynamicField» в поле поиска выше или непосредственно по ключу «sysconfig», если вы его уже знаете.

- Найдите подгруппу параметров для экрана, который ищете и нажмите на его название. Например, "Frontend::Agent::Ticket::ViewPhoneNew".
- Найдите параметр, оканчивающийся на "####DynamicField". Например, «Ticket::Frontend::AgentTicketPhone####DynamicField».

- Если параметр пуст или не имеет требуемого имени динамического поля, нажмите кнопку «+», чтобы добавить новую запись. Например, Ключ: Поле1, Содержимое: 1.
- Если в параметре уже указано имя динамического поля, убедитесь, что установлено значение «1», чтобы отображать поле, или значение «2», чтобы отображать его как обязательное.
- Сохраните конфигурацию, нажав кнопку «Обновить» в нижней части экрана, и перейдите к экрану, на котором вы хотите отобразить поле.

1. Показ примеров

Ниже приведены примеры конфигураций «SysConfig» для отображения или скрытия динамических полей на разных экранах.

Пример 4.22. Активируйте TestText на странице Новая Телефонная Заявка.

- Группа: Ticket
- Подгруппа: Frontend::Agent::Ticket::ViewPhoneNew
- Настройка: Ticket::Frontend::AgentTicketPhone###DynamicField
- Значение:

Ключ	Значение
TestText	1

Рисунок 4.104. Динамическое поле TestText на экране «Новая телефонная заявка»

The screenshot shows a form for creating a new phone call request. The fields are as follows:

- Следующее состояние заявки:
- Дата ожидания: . . :
- Для всех состояний "ожидает ..."
- Влияние:
- Приоритет:
- TestText:
- Дата исполнения: . . :
- Затраченное время (единиц времени): Часы : Мин оплачиваемое

At the bottom of the form is a blue button labeled "Создать".

Пример 4.23. Активируйте динамическое поле TestText на экране «Новая телефонная заявка» как обязательное.

- Группа: Ticket
- Подгруппа: Frontend::Agent::Ticket::ViewPhoneNew
- Настройка: Ticket::Frontend::AgentTicketPhone###DynamicField
- Значение:

Ключ	Значение
TestText	2

Рисунок 4.105. Обязательное динамическое поле TestText на экране «Новая телефонная заявка»

Следующее состояние заявки:

Дата ожидания: . . - :

Для всех состояний "ожидает ..."

Влияние:

Приоритет:

* TestText:

Дата исполнения: . . - :

Затраченное время (единиц времени): :

Пример 4.24. Активируйте несколько динамических полей на экране «Новая телефонная заявка».

- Группа: Ticket
- Подгруппа: Frontend::Agent::Ticket::ViewPhoneNew
- Настройка: Ticket::Frontend::AgentTicketPhone###DynamicField
- Значение:

Ключ	Значение
TestText	1
TestTextArea	1
TestCheck	1
TestDropdown	1
TestMulti	1
TestData	1
TestDataTime	1

Рисунок 4.106. Несколько необязательных для заполнения динамических полей на экране «Новая телефонная заявка».

The screenshot shows a form with the following fields and controls:

- Приоритет: 3 средний (dropdown menu)
- TestText: empty text input field
- TestCheck: unchecked checkbox
- TestMulti: empty dropdown menu
- TestData: unchecked checkbox, followed by date pickers for 26, 01, and 2023.
- TestDataTime: unchecked checkbox, followed by date pickers for 26, 01, and 2023, and time pickers for 13 and 10.
- Дата исполнения: date pickers for 29, 01, and 2023, and time pickers for 13 and 10.
- Затраченное время (единиц времени): Часы, Мин, and оплачиваемое (checked checkbox).

At the bottom of the form is a blue button labeled "Создать".

Пример 4.25. Деактивируйте некоторые поля на экране «Новая телефонная заявка».

- Группа: Ticket
- Подгруппа: Frontend::Agent::Ticket::ViewPhoneNew
- Настройка: Ticket::Frontend::AgentTicketPhone###DynamicField
- Значение:

Ключ	Значение
TestText	1
TestTextArea	0
TestCheck	1
TestDropdown	0
TestMulti	1
TestData	0
TestDataTime	1

Рисунок 4.107. Некоторые деактивированные поля на экране «Новая телефонная заявка».

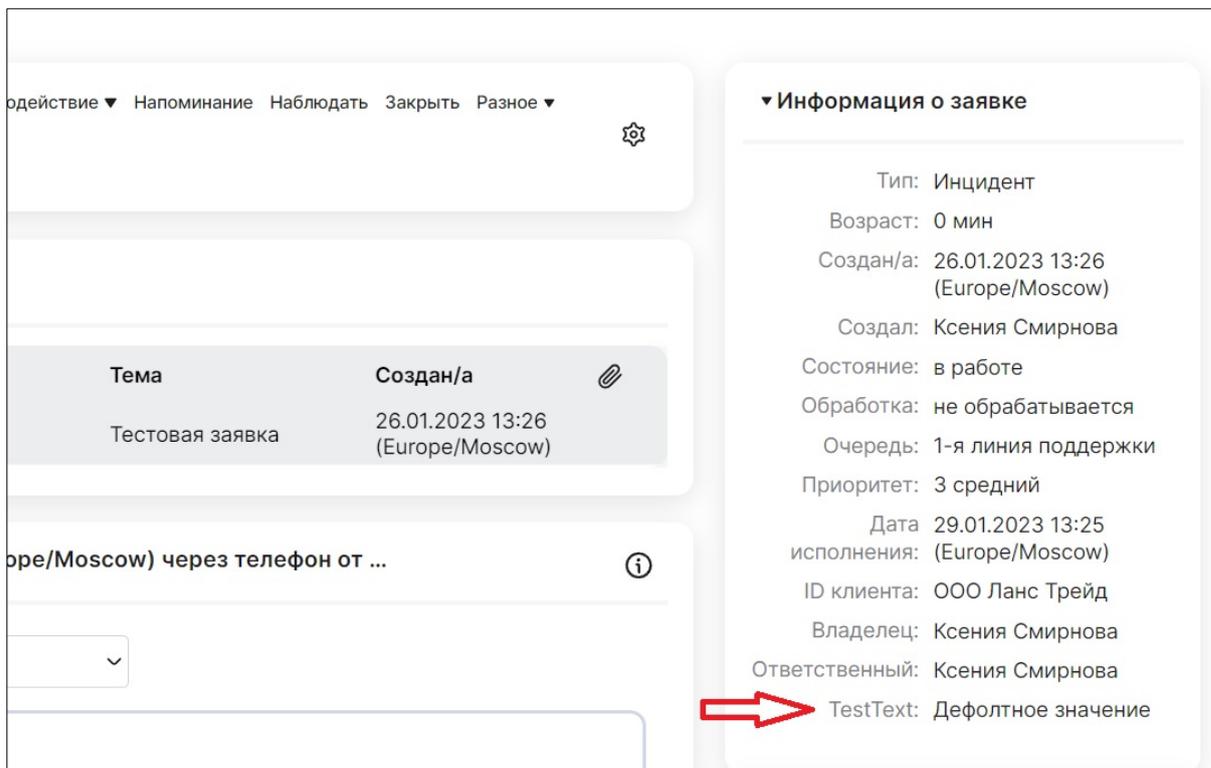
Пример 4.26. Активируйте динамическое поле TestText на странице просмотра подробной информации о заявке.

- Группа: Ticket
- Подгруппа: Frontend::Agent::Ticket::ViewZoom
- Настройка: Ticket::Frontend::AgentTicketZoom###DynamicField
- Значение:

Ключ	Значение
------	----------

TestText	1
----------	---

Рисунок 4.108. Динамическое поле TestText на странице просмотра подробной информации о заявке.



Пример 4.27. Активируйте динамическое поле TestText на странице просмотра обобщенной информации о заявке.

- Группа: Ticket
- Подгруппа: Frontend::Agent::TicketOverview
- Настройка: Ticket::Frontend::OverviewSmall###DynamicField
- Значение:

Ключ	Значение
TestText	1

Рисунок 4.109. Динамическое поле TestText на странице просмотра обобщенной информации о заявке.

Просмотр статуса: Открытые заявки							
Открытые заявки 2008		Закрытые заявки 577					
Массовое действие				1-25 из 2008 - Страница: 1 2 3 4 5 >> >			
<input type="checkbox"/>	Заявка#	▼ Возраст	TestText	Отправитель	Название	Состояние ↑	Обработка ↑
<input type="checkbox"/>	790002584	6 мин	Дефолтное значение	Никита Кожемяка	Тестовая заявка	в работе	не обрабатывается
<input type="checkbox"/>	790002583	1 дн 14 ч		Илья Муромец	Тестовая заявка	в работе	в обработке

Этот параметр влияет на: Escalation View, Locked View, Queue View, Responsible View, Status View, Service View и Watch View обзоры.

11. Установка Значения по Умолчанию с помощью модуля Ticket Event

Событие заявки (например, TicketCreate) может инициировать установку значения для определенного динамического поля, если это поле еще не имеет значения.

Примечание

При использовании этого метода, это значение по умолчанию не отображается на экране (например, «Новый телефонная заявка»), поскольку значение устанавливается после создания заявки.

Для активации этой функции необходимо включить следующую настройку: "Ticket::EventModulePost####TicketDynamicFieldDefault".

Пример 4.28. Активируйте динамическое поле TestText для события TicketCreate.

- Группа: Ticket
- Подгруппа: Core::TicketDynamicFieldDefault
- Настройка: Ticket::TicketDynamicFieldDefault####Element1

Примечание

Эту конфигурацию можно задать в любой из 16 настроек Ticket::TicketDynamicFieldDefail####Element.

Если необходимо настроить более 16 полей, пользовательский файл XML должен быть помещен в каталог \$RSD_HOME/Kernel/Config/files, чтобы расширить эту функцию.

- Значение:

Ключ	Значение
Event	TicketCreate
Name	TestText
Value	a new value

12. Установка Значений по Умолчанию с помощью Пользовательских Настроек

Значение динамического поля по умолчанию может быть перезаписано значением, определенным пользователем, которое хранится в пользовательских настройках.

При использовании этого способа, значение поля по умолчанию будет отображаться на всех экранах, для которых оно активировано (если поле уже не имеет другого значения).

Параметр «SysConfig» - «PreferencesGroups###DynamicField» расположен в подгруппе «Frontend::Agent::Preferences». Этот параметр является примером того, как создать запись на экране пользовательских настроек, чтобы задать значение по умолчанию исключительного динамического поля для выбранного пользователя. Ограничение этой настройки состоит в том, что она позволяет использовать только одно динамическое поле. Если два или более полей будут использовать эту функцию, необходимо создать настраиваемый файл конфигурации XML, чтобы добавить дополнительные настройки, подобные этой.

Примечание

Помните, что, если в новый XML добавляется больше новых параметров, имя каждого параметра должно быть уникальным в системе и отличаться от "Preferences-Group###DynamicField". Например: PreferencesGroups###101-Dynamic-Field-Field1, PreferencesGroups###102-DynamicField-Field2, PreferencesGroups###My-Field1, PreferencesGroups###My-Field2 и т. д.

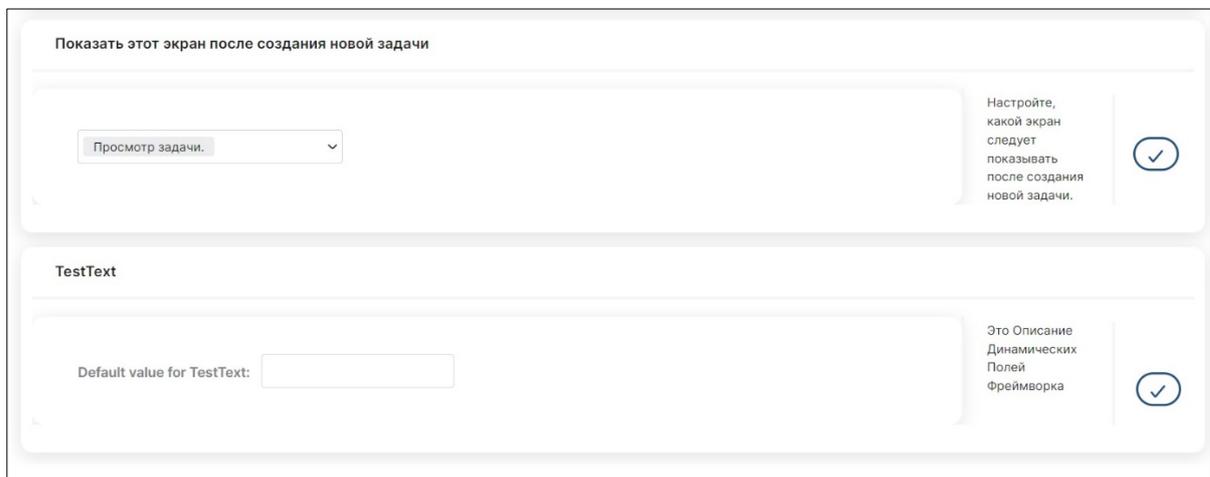
Пример 4.29. Активируйте динамическое поле TestText на странице Пользовательские Настройки.

- Группа: Ticket
- Подгруппа: Frontend::Agent::Preferences
- Настройка: PreferencesGroups###DynamicField
- Значение:

Ключ	Значение
Active	1
Block	Input
Column	Other Settings
Data	[% Env("UserDynamicField_TestText") %]
Key	Default value for TestText
Label	Test Text
Module	Kernel::Output::HTML::PreferencesGeneric

PrefKey	UserDynamicField_TestText
Prio	7000

Рисунок 4.110. Динамическое поле TestText на экране пользовательских настроек



11. Универсальный интерфейс (Generic Interface)

Универсальный интерфейс (Generic Interface) RSD состоит из многоуровневой структуры, которая позволяет RSD взаимодействовать с другими системами через веб-службу. Взаимодействие может быть двунаправленным:

- RSD как провайдер: RSD действует как сервер, прослушивающий запросы от внешней системы, обрабатывающий информацию, выполняющий запрошенное действие и отвечающий на запрос.
- RSD как запрашивающая сторона: RSD действует как клиент, собирающий информацию и отправляющий запрос в удаленную систему, ожидая при этом ответа.

11.1. Уровни универсального интерфейса

Универсальный интерфейс построен на основе многоуровневой модели, чтобы быть гибким и простым в настройке.

Уровень - это набор файлов, которые управляют тем, как универсальный интерфейс выполняет различные части веб-сервиса. Используя правильную конфигурацию,

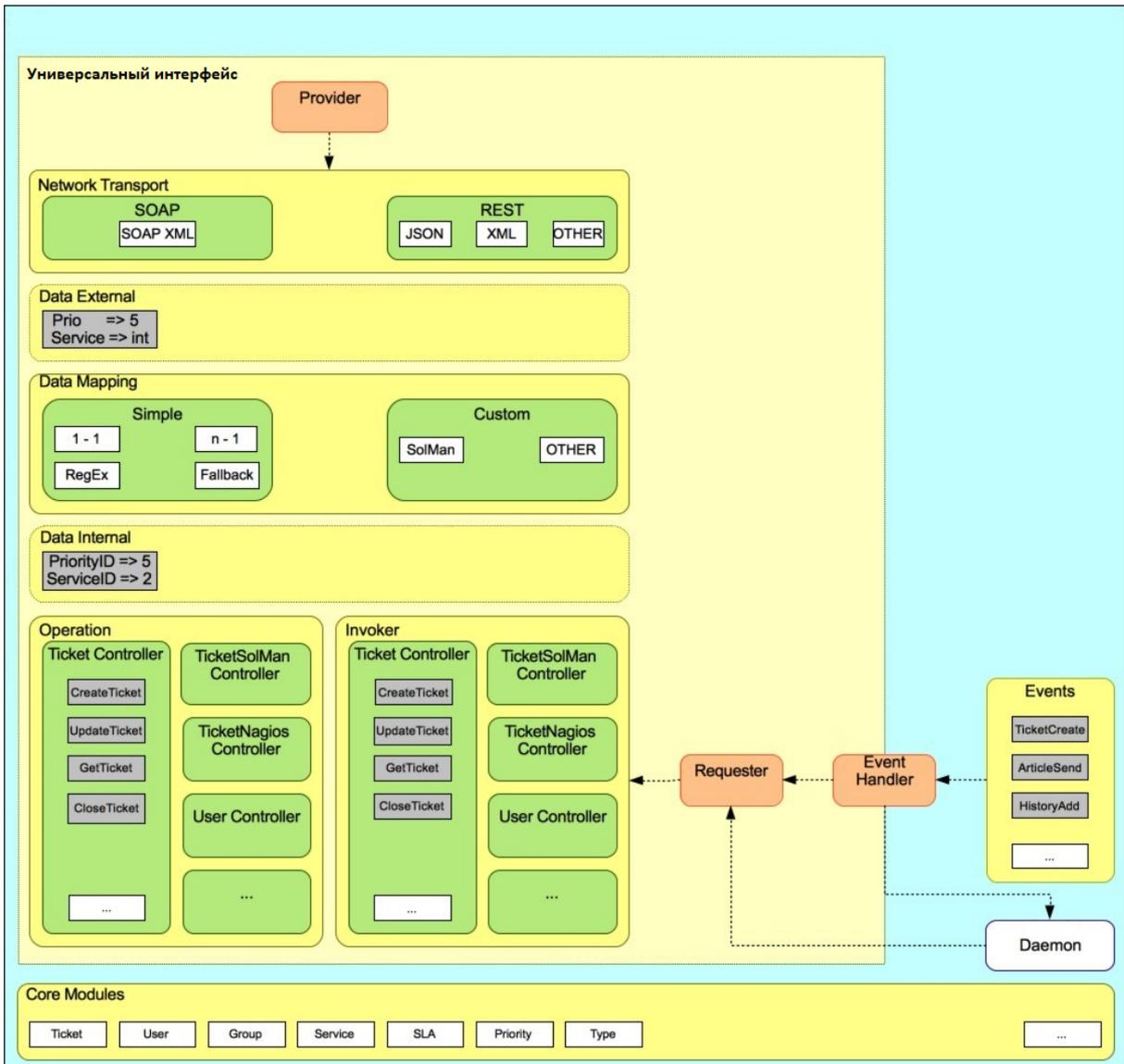
можно создавать разные веб-сервисы для разных внешних систем, без создания новых модулей.

Примечание

Если внешняя система не поддерживает текущие модули универсального интерфейса, необходимо разработать специальные модули для этой конкретной системы.

Список предоставляемых универсальных интерфейсных модулей, поставляемых с RSD, со временем будет обновляться и увеличиваться.

Рисунок 4.111. Графическое представление многослойной структуры



1. Сетевой транспорт

Этот уровень отвечает за правильную связь с удаленной системой. Он получает запросы и генерирует ответы, действуя в качестве поставщика. Действуя в качестве запрашивающего, он генерирует повторные запросы и получает ответы.

Связь с поставщиком обрабатывается дескриптором веб-сервера под названием "nph-genericinter-face.pl".

Связь с внешней системой может быть инициирована во время события, инициируемого универсальным модулем интерфейса или любым другим модулем RSD. Это событие перехватывается обработчиком событий, и, в зависимости от конфигурации, событие будет обработано непосредственно инициирующим модулем или делегировано "Планировщику событий" (отдельному "демону" (модуль работающий по расписанию), предназначенному для асинхронной обработки задач).

2. Отображение данных

Этот уровень отвечает за преобразование структур данных между RSD и удаленной системой (внутренний уровень данных и внешний уровень данных).

Удаленные системы могут иметь структуры данных, отличные от RSD (включая разные значения и имена для этих значений). Данный уровень преобразует полученную информацию в понятную для RSD и наоборот, при отправке преобразует информацию в понятную для удаленной системы.

Пример: "Приоритет" (или другие атрибуты) в удаленной системе может называться "Приоритет\Важность\Срочность", и может случиться так, что значение "Низкий" должно быть сопоставлено с "Очень низкий" в удаленной системе.

3. Контроллеры

Контроллеры - это наборы аналогичных операций или вызывающих устройств. Например, контроллер заявок может содержать несколько стандартных операций с заявками. Могут быть внедрены пользовательские контроллеры, например, контроллер "TicketExternalCompany", который может содержать функции, аналогичные стандартному контроллеру заявок, но с другим интерфейсом передачи данных или именами функций (для адаптации к именам функций удаленной системы) или дополнять другой код.

Одним из приложений для универсального интерфейса может быть синхронизация информации с удаленной системой, которая может взаимодействовать только с другой удаленной системой того же типа. В этом случае необходимо разработать новые контроллеры, а "Операции" (Operation) и "Вызывающие устройства" (Invoker) должны эмулировать поведение удаленной системы таким образом, чтобы интерфейс, предоставляемый RSD, был похож на интерфейс удаленной системы.

4. Операция (Operation) (RSD в качестве поставщика)

Операция - это одно действие, которое может быть выполнено в RSD. Все процедуры имеют один и тот же программный интерфейс, они получают данные в виде одного определенного параметра и возвращают структуру данных со статусом успешного выполнения, потенциальным сообщением об ошибке и возвращаемыми данными.

Обычно "операция" использует уже сопоставленные данные (внутренние) для вызова основных модулей и выполнения действий в RSD, таких как: Создание заявки, Обновление пользователя, Отключение очереди, Отправка уведомления и т.д. Операция имеет полный доступ к RSD API для выполнения действия.

5. Вызов (Invoker) (RSD в качестве запрашивающего)

Вызов - это действие, которое RSD выполняет в сторону удаленной системы. Вызовы используют основные модули RSD для обработки и сбора необходимой информации для создания запроса. Когда информация готова, она должна быть сопоставлена с форматом удаленной системы, для отправки в Удаленную систему. Удаленная система обработает информацию, выполнит действие и отправит ответ в RSD об успехе или ошибке.

11.2. Информационный поток универсального интерфейса

Универсальный интерфейс имеет определенный поток информации для выполнения действий в качестве поставщика и в качестве запрашивающего.

Эти потоки описаны ниже:

1. RSD как поставщик информации (Provider)

Удаленный запрос:

1. HTTP запрос
 - RSD получает HTTP-запрос и передает его через слои.
 - Данный модуль отвечает за выполнение и контроль этих действий.
2. Сетевой транспорт
 - Модуль сетевого транспорта декодирует полученные данные и отделяет название операции от остальных данных.
 - Название операции и данные операции возвращаются в модуль “поставщика”.
3. Внешние данные
 - Данные, отправленные из удаленной системы (это не модульный уровень).
4. Маппинг
 - Данные преобразуются из внешнего системного формата во внутренний формат RSD, как указано в конфигурации сопоставления для этой операции (Сопоставление для данных входящего запроса).
 - Уже преобразованные данные возвращаются в модуль “поставщика”.
5. Внутренние данные
 - Данные, преобразованные и подготовленные для передачи в операцию (это не модульный уровень).
6. Операция
 - Получает и проверяет данные.
 - Осуществляет контроль доступа пользователей.
 - Выполняет действие.

Ответ RSD:

1. Операция
 - Возвращает данные результата в модуль “поставщик”.
2. Внутренние данные
 - Данные, возвращенные после операции.
3. Маппинг
 - Данные преобразуются обратно в формат удаленной системы, как указано в конфигурации (сопоставление для данных исходящего ответа).
 - Уже преобразованные данные возвращаются в модуль “поставщик”.

4. Внешние данные
 - Данные преобразованы и подготовлены для передачи сетевому транспорту в качестве ответа.
5. Сетевой транспорт
 - Получает данные уже в формате удаленной системы.
 - Создает допустимый ответ для этого типа сетевого транспорта.

6. HTTP ответ

- Ответ отправляется обратно клиенту веб-сервиса.
- В случае ошибки в удаленную систему отправляется ответ об ошибке (например, ошибка SOAP, ошибка HTTP и т.д.).

2. RSD в качестве отправителя запроса (Requester)

RSD запрос:

1. Обработчик событий (триггер)

На основе конфигурации веб-служба определяет, будет ли запрос синхронным или асинхронным.

- Синхронный
 - Прямой вызов для создания нового запроса и передачи его в соответствующие уровни.
- Асинхронный
 - Создает новую задачу универсального интерфейса (Requester) как задача для демона RSD (переносом выполнение запроса на демон Scheduler, пользовательский интерфейс может быть значительно улучшен, в противном случае все время, необходимое для подготовки запроса и удаленного выполнения, будет добавлено к событиям RSD, которые запускают эти запросы).
 - В своем следующем цикле демонический процесс RSD считывает новую задачу и создает вызов Отправителю запроса, который создаст новый запрос, а затем передаст его в соответствующие уровни.

2. Вызов

- Получает данные от события.
- Проверяет полученные данные (при необходимости).
- Вызывает основные модули для дополнения данных (при необходимости).
- Возвращает структуру данных запроса или отправляет запрашиваемому сигнал остановки связи, чтобы отменить запрос.

3. Внутренние данные

- Данные, переданные от вызывающего устройства (это не модульный уровень).

4. Маппинг

- Данные преобразуются в формат удаленной системы, как указано в конфигурации (сопоставление данных для исходящего ответа).
- Уже преобразованные данные возвращаются отправителю запроса.

5. Внешние данные

- Данные, преобразованные и подготовленные для отправки в удаленную систему.

6. Сетевой транспорт

- Получает от отправителя запроса название удаленной операции и данные, уже преобразованные в формат удаленной системы.

- Создает действительный запрос для сетевого транспорта.
- Отправляет запрос в удаленную систему и ожидает ответа.

Ответ от удаленной системы:

1. Сетевой транспорт
 - Получает ответ и декодирует полезную нагрузку данных.
 - Возвращает данные отправителю запроса.
2. Внешние данные
 - Данные, полученные из удаленной системы.
3. Маппинг
 - Данные преобразуются из внешнего формата во внутренний формат RSD, как указано в конфигурации для этой операции (сопоставление входящих данных ответа).
 - Уже преобразованные данные возвращаются отправителю запроса.
4. Внутренние данные
 - Данные преобразованы и готовы к передаче обратно отправителю запроса.
5. Вызывающий
 - Получает возвращаемые данные.
 - Обрабатывает данные по мере необходимости конкретно для каждого вызывающего (включая обработку ошибок, если таковые имеются).
 - Возвращает результат вызывающему и отправителю запроса.
6. Обработчик событий или RSD демон (daemon)
 - Получает данные от отправителя запроса. В случае демона RSD эти данные могут содержать информацию для создания задачи в будущем.

11.3. Веб-сервисы

Веб-служба - это способ связи между двумя системами, в нашем случае RSD и удаленной системой.

Сердцем веб-сервиса является ее конфигурация, где определяется, какие действия веб-служба может выполнять внутри ("Операции") и какие действия в RSD может выполнять удаленная система ("Вызов"), как данные преобразуются из одной системы в другую ("Маппинг") и по какому протоколу будет осуществляться связь ("Сетевой транспорт").

Универсальный интерфейс - это фреймворк, который позволяет создавать веб-сервисы для RSD предопределенным способом, используя уже готовые строительные блоки, которые независимы друг от друга и взаимозаменяемы.

11.4. Графический интерфейс Веб-сервиса

Графический пользовательский интерфейс веб-сервиса (GUI) - это инструмент, который позволяет создавать сложные конфигурации веб-сервиса в удобном для пользователя интерфейсе. Основные функции:

- Создание и удаление веб-сервиса.
- Импорт и экспорт конфигураций (в формате файла YAML) для существующих веб-служб.
- Просмотр, Возврат и экспорт старых конфигураций для существующих веб-служб на экране "Истории" веб-служб.

- Отслеживание истории взаимодействия для каждого веб-сервиса на экране отладчика.

1. Обзор веб-сервиса

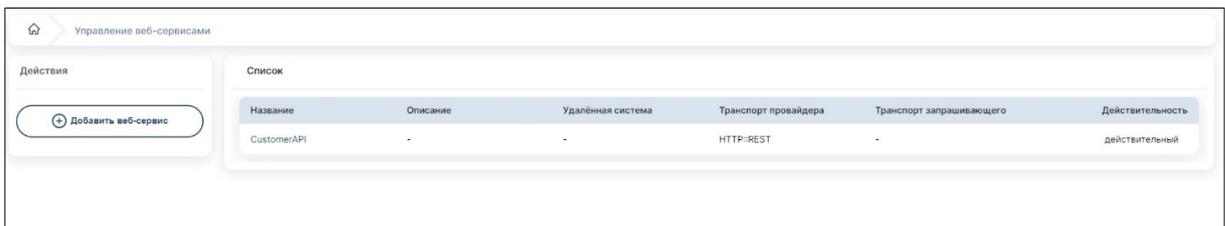
Ссылка на веб-сервисы на главном экране интерфейса администратора ведет на экран обзора веб-служб, где вы можете управлять конфигурациями своих веб-служб. На этом экране вы можете добавлять новые веб-сервисы или изменять конфигурацию существующих.

Каждый экран настройки веб-сервиса имеет в верхней части экрана навигационный путь в стиле хлебных крошек. Этот путь навигации полезен для того, чтобы точно знать, в какой части конфигурации веб-сервиса мы находимся, а также позволяет пользователю в любое время вернуться к любой части процесса настройки (без сохранения сделанных изменений).

Примечание

Чтобы создать новый веб-сервис, нажмите кнопку “Добавить веб-сервис” и заполните необходимую информацию.

Рисунок 4.112. Обзор веб-сервиса



2. Обзор веб-сервиса

Единственным обязательным полем в этой части является имя веб-сервиса, которое должно быть уникальным в системе и не может быть оставлено пустым. Другие поля также необходимы для конфигурации, такие как “порог отладки” и “действительность”, но эти поля уже заполнены значением по умолчанию.

Значение по умолчанию для порога отладки равно “отладка” (debug). При настройке таким образом все журналы связи регистрируются в базе данных. Каждое последующее пороговое значение отладки является более ограничительным и отбрасывает журналы связи более низкого порядка, чем тот, который установлен в системе.

Пороговые уровни отладки (от нижнего к верхнему)

- Отладка
- Информация
- Уведомление
- Ошибка

Также возможно определить сетевой транспортный протокол для RSD в качестве поставщика и RSD в качестве запрашивающего.

Нажмите на кнопку “Сохранить”, чтобы зарегистрировать новую веб-службу в базе данных, или нажмите кнопку “Отмена”, чтобы отменить операцию, после этого вы вернетесь к экрану обзора веб-сервиса.

Если у вас уже есть файл конфигурации веб-сервиса в формате YAML, вы можете нажать на кнопку «Импортировать веб-службу» в левой части экрана. Для получения дополнительной информации об импорте веб-служб, пожалуйста, ознакомьтесь со следующим разделом Изменение веб-сервиса.

Примечание

Чтобы изменить или добавить дополнительные сведения о веб-службе, щелкните по имени веб-сервиса на экране обзора веб-сервиса.

Рисунок 4.113. Добавление Веб-сервисов

Управление веб-сервисами | Добавить веб-сервис

Действия

- ← Вернуться к обзору
- ↓ Импортировать веб-сервис

Ready2Adopt веб-сервисы

Хотите ли получить выгоду от веб-сервисов, созданных экспертами? Обязательно для импорта нескольких сложных Ready2Adopt веб-сервисов.

Подсказка

После сохранения конфигурации вы будете перенаправлены обратно на экран редактирования. Чтобы вернуться к обзору, нажмите кнопку «Вернуться к обзору».

Общие

Имя: Порог оплада: Удаленная система: Действительность:

как провайдер

В режиме провайдера предоставляет удаленным системам веб-сервисы.

Параметры

Сетевой транспорт:

Модули обработки ошибок

Модули обработки ошибок используются для реагирования на ошибки во время взаимодействия. Эти модули запускаются в определенном порядке, который можно изменить путем их перетаскивания с помощью мыши.

#	Название	Описание	Бэкенд
1	Данные не найдены.		

Операции

Операции — это отдельные системные функции, которые могут вызываться удаленными системами.

Название	Описание	Контроллер	Входящий мэтлинг	Исходящий мэтлинг
Данные не найдены.				

как запрашивающий

В режиме запрашивающего использует веб-сервисы удаленных систем.

Параметры

Сетевой транспорт:

Модули обработки ошибок

Модули обработки ошибок используются для реагирования на ошибки во время взаимодействия. Эти модули запускаются в определенном порядке, который можно изменить путем их перетаскивания с помощью мыши.

#	Название	Описание	Бэкенд
1	Данные не найдены.		

Invoicers

Вызывающие модули подготавливают данные для запроса к удаленному веб-сервису и обрабатывают полученные от них ответы.

Название	Описание	Контроллер	Входящий мэтлинг	Исходящий мэтлинг
Данные не найдены.				

Сохранить

или

3. Изменение веб-сервиса

На этом экране у вас есть полный набор функций для управления каждой частью веб-сервиса. С левой стороны в колонке действий вы можете найти несколько кнопок, которые позволяют выполнять все возможные действия над веб-сервисом:

- Клонировать веб-сервис.
- Экспортировать веб-сервис.
- Импортировать веб-сервис.
- История конфигурации.
- Удалить веб-сервис.

- Отладчик.

Примечание

История конфигурации и отладчик приведут вас к разным экранам.

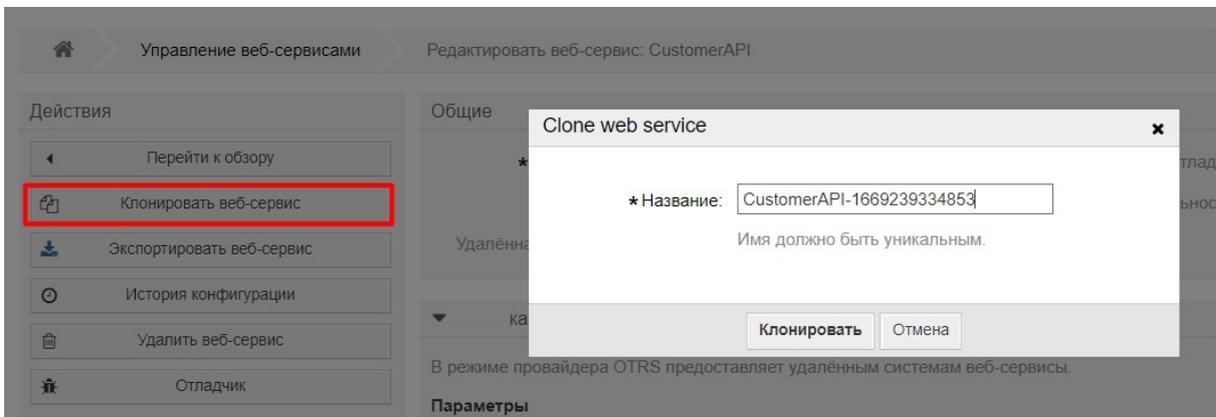
1. Клонирование веб-сервиса

Для клонирования веб-сервиса необходимо нажать кнопку «Клонировать веб- сервис», появится диалоговое окно, в котором можно использовать имя по умолчанию или же установить новое имя для (клонировемого) веб-сервиса.

Примечание

Помните, что имя веб-сервиса должно быть уникальным в системе. Нажмите кнопку «Клонировать», чтобы создать клон веб-сервиса, или «Отмена», чтобы закрыть диалоговое окно.

Рисунок 4.114. Клонирование веб-сервиса



2. Экспорт веб-сервиса

Кнопка "Экспортировать веб-сервис/Export web service" дает возможность выгрузить конфигурацию текущего веб-сервиса в файл формата YAML, для загрузки его и сохранения в файловой системе. Это может быть особенно полезно, если вы желаете перенести веб сервис с одного сервера на другой, например, из тестовой среды в рабочую.

Предупреждение

Все сохраненные пароли в настройках веб-сервиса будут экспортированы в виде обычного текста.

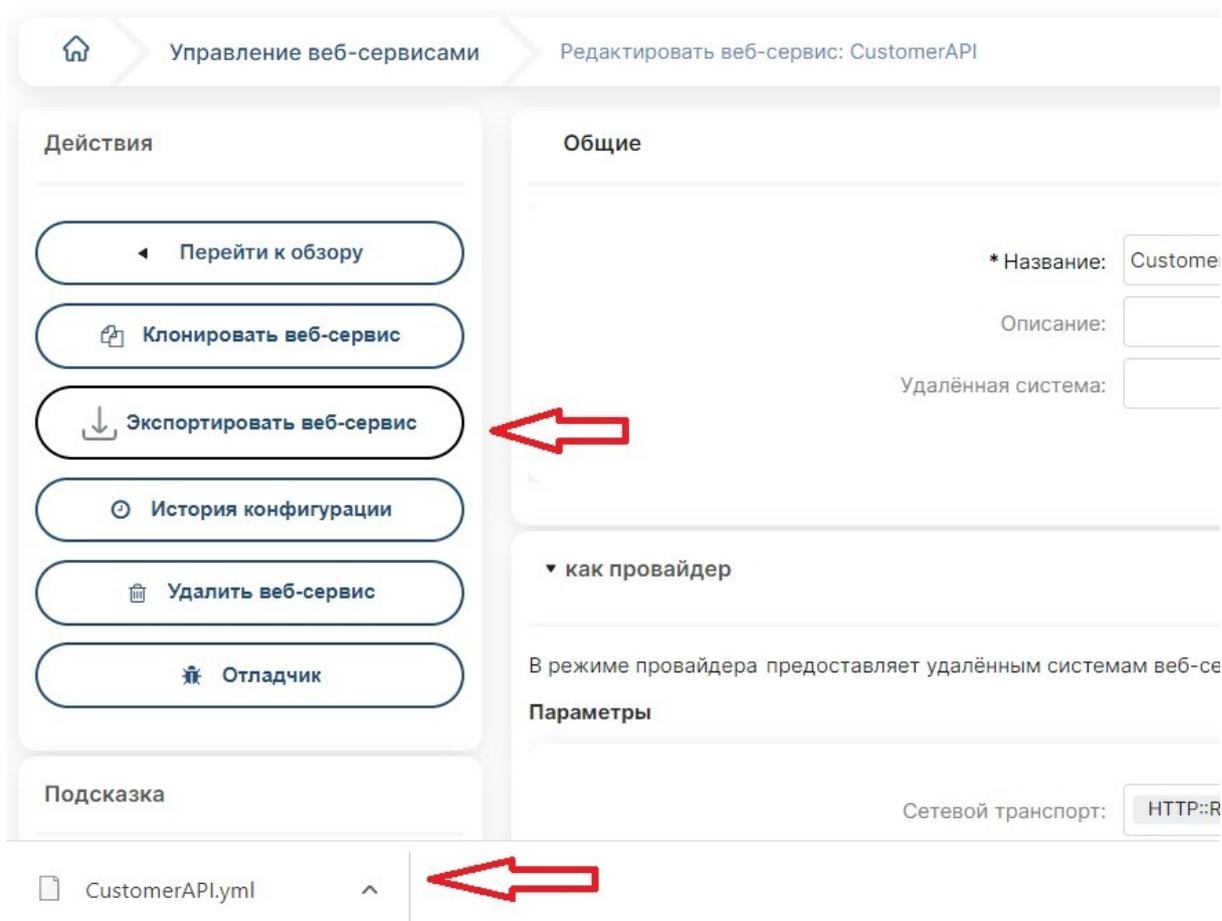
Сразу же после нажатия кнопки "Экспортировать веб-сервис" появится диалоговое окно сохранения, наподобие того, которое появляется, когда вы нажимаете в браузере ссылку сохранить.

Примечание

Каждый браузер в каждой операционной системе имеет собственный диалоговый экран и стиль сохранения. В зависимости от браузера и его конфигурации диалоговое окно может не отображаться, а файл будет сохранен в каталоге по умолчанию в

вашей файловой системе. Пожалуйста, обратитесь к документации вашего браузера для получения более конкретных инструкций, если это необходимо.

Рисунок 4.115. Экспортировать веб-сервис



3. Импорт веб-сервиса

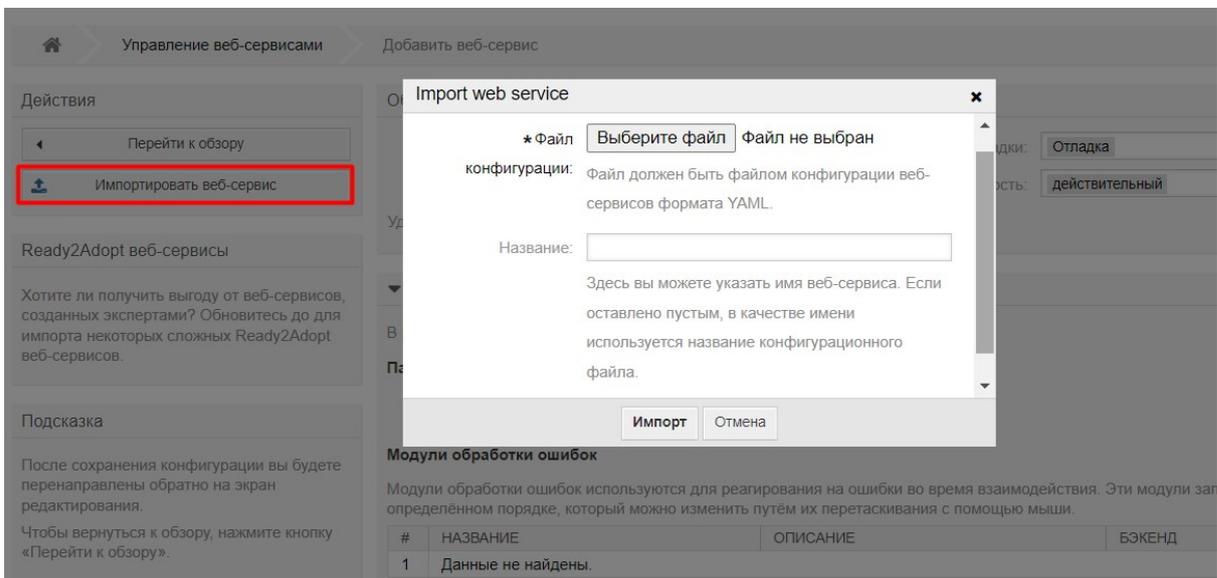
Для загрузки имеющегося YAML файла конфигурации веб-сервиса требуется использование функции импорта. Нажмите кнопку "Импорт веб-сервиса", найдите нужный файл или укажите полный путь к нему в окне ввода

Нажмите кнопку "Импорт" чтобы создать новый веб-сервис с файла или кнопку "Отменить" чтобы закрыть диалоговое окно.

Примечание

Имя веб-сервиса будет взято из имени конфигурационного файла (например, если имя файла MyWebservice.yml, то имя веб-сервиса будет - MyWebser- vice). Если в системе уже зарегистрирован веб-сервис с таким именем, система откроет экран изменения веб-сервиса, чтобы вы могли поменять имя импортируемого веб-сервиса.

Рисунок 4.116. Импортировать веб-сервис



4. История веб-сервиса

Каждое изменение конфигурации веб-сервиса создает новую запись в истории веб-сервиса (в виде журнала). На экране истории веб-сервиса отображается список всех версий веб-сервиса. Каждая строка (версия) в списке истории конфигурации представляет одну ревизию в истории веб-сервиса.

Нажмите на одну из строк, чтобы отобразить всю конфигурацию в том виде, в каком она была на конкретную дату/время. Конфигурация будет показана в разделе сведений об истории этого экрана. Здесь вы также можете экспортировать выбранную версию конфигурации веб-сервиса или восстановить эту версию в текущей конфигурации веб-сервиса.

Конфигурация «Экспорт веб-сервиса» ведет себя точно так же, как функция «Экспорт веб-сервиса» на экране изменения веб-сервиса. Для получения дополнительной информации обратитесь к этому разделу.

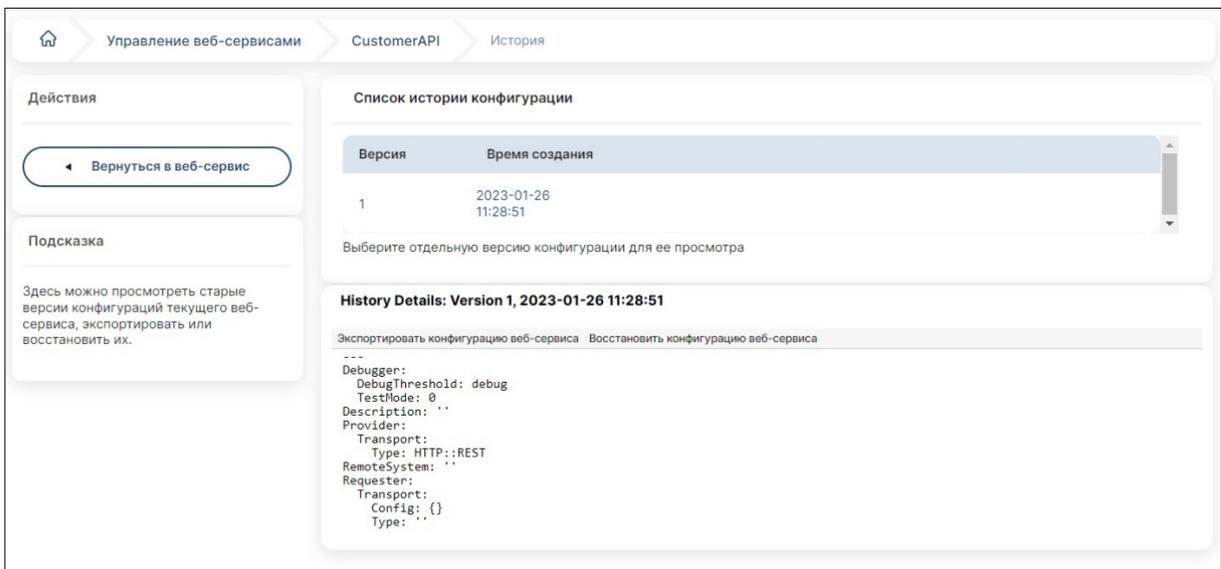
Если изменения в текущей конфигурации веб-сервиса не работают должным образом и отменить изменения вручную непросто, вы можете нажать кнопку Вернуть конфигурацию веб-сервиса. Откроется диалоговое окно с вопросом, уверены ли вы, что хотите отменить настройку веб-сервиса. Нажмите «Вернуть конфигурацию веб-сервиса» в этом диалоговом окне, чтобы заменить текущую конфигурацию выбранной версией, или нажмите «Отмена», чтобы закрыть диалоговое окно.

Предупреждение

Помните, что все пароли, сохраненные в конфигурации веб-сервиса, будут экспортироваться в виде обычного текста.

Будьте осторожны при восстановлении конфигурации, так как этот процесс необратим.

Рисунок 4.117. История веб-сервиса



5. Удаление веб-сервиса

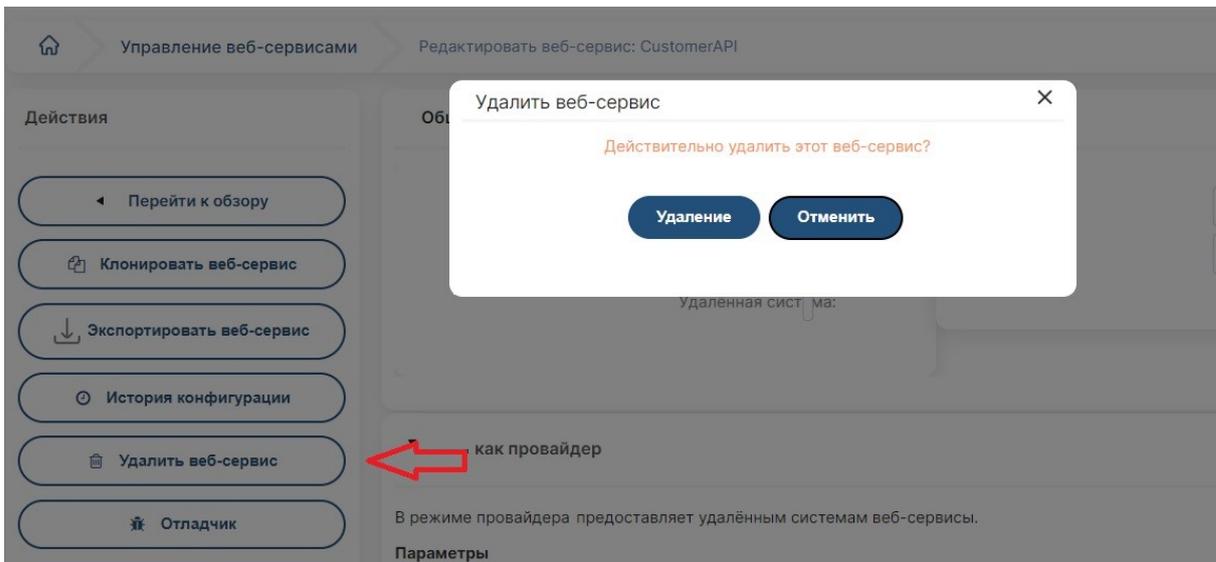
Иногда необходимо полностью удалить веб-сервис. Для этого вы можете нажать на кнопку «Удалить веб-сервис», и появится новое диалоговое окно с запросом на подтверждение.

Нажмите «Удалить», чтобы подтвердить удаление веб-сервиса, или «Отмена», чтобы закрыть диалоговое окно.

Предупреждение

Удаление веб-сервиса нельзя отменить, будьте осторожны при удалении веб-сервиса.

Рисунок 4.118. Удалить веб-сервис



6. Отладчик веб-сервиса

Отладчик хранит журнал веб-сервиса. На экране отладчика вы можете отслеживать все сообщения веб-сервиса для любого типа поставщика или запрашивающего.

При отображении этого экрана начинается загрузка списка запросов. После того, как список будет полностью заполнен, вы можете выбрать одну из строк (что означает последовательность связи), чтобы проверить ее детали. Эта информация появится в поле ниже.

Вы можете сузить список контактов, используя фильтр в правой части экрана. Вы можете фильтровать по:

- Типу связи (поставщик или запрашивающий)
- Дате: до и/или после определенной даты
- Удаленному IP-адрес
- Сочетанию всех

После установки параметров фильтра нажмите кнопку «Обновить», и появится новый список, соответствующий вашим критериям поиска.

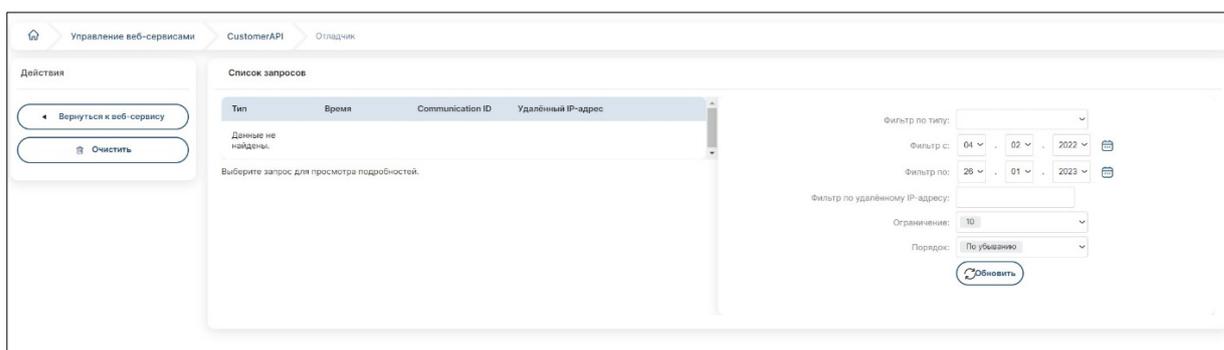
Примечание

В зависимости от критериев поиска для фильтров новый список может не дать результатов.

В левой части экрана под столбцом действий вы можете выбрать «Вернуться к веб-сервису» или очистить журнал отладчика, нажав кнопку «Очистить». Откроется диалоговое окно, в котором вас попросят подтвердить удаление журнала. Нажмите «Очистить» в диалоговом окне, чтобы выполнить действие, или нажмите «Отмена», чтобы закрыть это диалоговое окно.

В разделе Подробности запроса вы можете увидеть все детали для выбранного сообщения. Здесь вы можете отслеживать весь поток и проверять наличие возможных ошибок или подтверждать успешные ответы.

Рисунок 4.119. Отладчик веб-сервиса



7. Изменение конфигурации веб-сервиса

Возвращаясь к экрану изменения веб-сервиса, теперь мы рассмотрим его правую часть. Здесь у нас есть возможность изменить все общие данные для веб-сервиса, такие как имя, описание, порог отладки и т.д. Также ниже есть еще два раздела, которые позволяют нам изменять конкретные параметры для типов связи RSD как поставщик и RSD как запрашивающий.

Конфигурация веб-сервиса должна быть сохранена на каждом уровне. Это означает, что при изменении настройки ссылки на другие, более глубокие части конфигурации

будут отключены, что вынудит вас сохранить текущий уровень конфигурации. После сохранения отключенные ссылки будут снова включены, что позволит вам продолжить настройку.

В разделе RSD как поставщик можно установить или сконфигурировать сетевой транспортный протокол. В списке отображаются только зарегистрированные серверные части сетевого транспорта. К

настройте сетевой транспорт нажмите на кнопку Настроить. В этом поле также можно добавлять новые операции. Для этого выберите одну из доступных операций из списка «Добавить операцию». Это приведет вас к экрану настройки операции. После сохранения новой операции она будет указана в таблице выше.

RSD как запрашивающий очень похож на предыдущий, но вместо операций вы можете добавить сюда вызывающих.

Нажмите кнопку «Сохранить», чтобы сохранить и продолжить настройку веб-сервиса, «Сохранить и завершить», чтобы сохранить и вернуться к экрану обзора веб-сервиса, или «Отменить», чтобы отменить текущие изменения уровня конфигурации и вернуться к экрану обзора веб-сервиса.

Рисунок 4.120. Изменение веб-сервиса

Управление веб-сервисами Редактировать веб-сервис: Первый Веб-Сервис

Действия

- Перейти к обзору
- Клонировать веб-сервис
- Экспортировать веб-сервис
- История конфигурации
- Удалить веб-сервис
- Отладчик

Подсказка

После сохранения конфигурации вы будете перенаправлены обратно на экран редактирования. Чтобы вернуться к обзору, нажмите кнопку «Перейти к обзору».

Общие

Имя: Первый Веб-Сервис Порог отладки: Отладка

Описание: Действительность: действительная

Удаленная система:

▼ как провайдер

В режиме провайдера предоставляет удаленным системам веб-сервисы.

Параметры

Сетевой транспорт: HTTP-SOAP Ж Конфигурировать

Модули обработки ошибок

Модули обработки ошибок используются для реагирования на ошибки во время взаимодействия. Эти модули запускаются в определенном порядке, который можно изменить путем их перетаскивания с помощью мыши.

#	Название	Описание	Бэкенд
1	Данные не найдены.		

Добавить модуль обработки ошибок:

Операции

Операции — это отдельные системные функции, которые могут вызываться удаленными системами.

Название	Описание	Контроллер	Входящий мессинг	Исходящий мессинг
TicketCreate	-	Ticket:TicketCreate	-	-
TicketGet	-	Ticket:TicketGet	-	-

Добавить Operation:

Примечание

Как и другие экраны настройки общего интерфейса, такие как Сетевой транспорт, Операция, Вызывающий и отображение, экран начальной настройки (добавления) будет представлены два варианта: «Сохранить» и «Отменить». При повторном посещении конфигурации появится новая опция «Сохранить и завершить». Поведение этой функции определено ниже.

Сохранение сохранит текущий уровень конфигурации в базе данных, и он вернется к предыдущему экрану, чтобы просмотреть ваши изменения или настроить более глубокие настройки.

Сохранить и завершить сохранит текущий уровень конфигурации в базе данных, и он вернется к предыдущему экрану в иерархии конфигурации (к непосредственному верхнему уровню конфигурации).

«Отмена» отменит любое изменение конфигурации до текущего уровня конфигурации и вернет к предыдущему экрану в иерархии конфигурации.

7.1. Сетевой транспорт веб-сервиса

В будущем список доступных сетевых транспортных средств будет увеличен. В настоящее время только HTTP::SOAP и HTTP::REST транспорт доступен. Каждый транспорт имеет разные параметры конфигурации для настройки, и они могут использовать разные интерфейсные модули для их настройки.

Настроить довольно просто HTTP::SOAP протокол как поставщик. Есть только два параметра: пространство имен и максимальная длина сообщения. Эти поля обязательны для заполнения. Первый - это URI для предоставления методам SOAP контекста, уменьшая неоднозначности, а второй - это поле, в котором вы можете указать максимальный размер (в байтах) для сообщений SOAP, которые будут обрабатываться RSD.

Рисунок 4.121. Сетевой транспорт веб-сервиса(HTTP::SOAP)

The screenshot shows a web management interface for configuring a web service provider. The breadcrumb trail is: Управление веб-сервисами > Первый Веб-Сервис > Provider Transport HTTP::SOAP. The main content area is titled 'Транспортная Сеть' and 'Properties'. It contains several configuration fields:

- Тип:** HTTP::SOAP
- * Установить SOAPAction:** Да (dropdown menu)
- * SOAPAction схема:** <Namespace><Separator><Operation> (text input)
- * Разделитель SOAPAction.:** # (text input)
- * Пространство имен:** (text input)
- * Схема именования запросов:** <FunctionName>DATA<FunctionName> (text input)
- * Схема именования ответов:** <FunctionNameResponse>DATA<FunctionNameResponse> (text input)
- Дополнительные заголовки ответа:** + Добавить заголовок ответа (button)
- * Максимальная длина сообщения:** (text input)
- Опции сортировки:** Добавить новый элемент первого уровня: (text input) + Добавить (button)

At the bottom, there are three buttons: Сохранить, Сохранить и закончить, and Отмена.

При желании вы можете захотеть определить дополнительные заголовки ответов. Они могут быть использованы для добавления статических значений заголовка к каждому ответу. Просто нажмите на заголовок «Добавить ответ» и заполните поля ключ и значение. Количество дополнительных строк заголовка не ограничено.

Для HTTP::REST конфигурация может быть немного сложнее, поскольку она динамически увеличивается для каждой настроенной операции путем добавления: Отображение маршрута для операции '<Имя операции>': и допустимые методы запроса для операции '<Имя операции>': настройки к настройкам транспорта по умолчанию Максимальная длина сообщения: и Отправка Поддерживать жизнь:.

- Отображение маршрута для операции '<Название операции>':

В этом параметре задается путь к ресурсу. Этот путь должен быть определен в соответствии с потребностями веб-сервиса, учитывая, что путь в сочетании с методом HTTP-запроса определяет операцию универсального интерфейса, которая должна быть выполнена.

Путь может содержать переменные в форме '<имя_переменной>' каждая строка пути, которая соответствует позиции имени переменной, будет добавлена в полезную нагрузку запроса с использованием имени переменной, определенного в этом параметре.

Примеры:

Отображение маршрута: /Ресурс

- Действительные запросы:

<http://localhost/rsd/nph-genericinterface.pl/Webservice/Test/Resource/1>

<http://localhost/rsd/nph-genericinterface.pl/Webservice/Test/Resource/1?Param1=One>

- Недействительные запросы:

<http://localhost/rsd/nph-genericinterface.pl/Webservice/Test/Resource/>

<http://localhost/rsd/nph-genericinterface.pl/Webservice/Test/Resource/OtherResource>

<http://localhost/rsd/nph-genericinterface.pl/Webservice/Test/Resource/OtherResource?Param1=One>

Route mapping: /Resource/:ID

- Действительные запросы:

<http://localhost/rsd/nph-genericinterface.pl/Webservice/Test/Resource/1>

<http://localhost/rsd/nph-genericinterface.pl/Webservice/Test/Resource/1?Param1=One>

В обоих случаях идентификатор = 1 будет отправлен в операцию как часть полезной нагрузки. Во втором случае также будет добавлен Param1 = One, в зависимости от

метода HTTP-запроса будут добавлены другие параметры, если они представлены в виде строки JSON в заголовке запроса.

- Недействительные запросы:

`http://localhost/rsd/nph-genericinterface.pl/Webservice/Test/Resource`

`http://localhost/rsd/nph-genericinterface.pl/Webservice/Test/ Resource?Param1=One`

Route mapping: `/Resource/OtherResource/:ID/:Color`

- Действительные запросы:

`http://localhost/rsd/nph-genericinterface.pl/Webservice/Test/Resource/OtherResource/1/Red`

`http://localhost/rsd/nph-genericinterface.pl/Webservice/Test/Resource/OtherResource/123/Blue?Param1=One`

В первом примере ID = 1 и цвет = красный, в то время как во втором ID = 123 и цвет = Синий.

- Недействительные запросы:

`http://localhost/rsd/nph-genericinterface.pl/Webservice/Test/Resource/1`

`http://localhost/rsd/nph-genericinterface.pl/Webservice/Test/ Resource/OtherResource/1`

`http://localhost/rsd/nph-genericinterface.pl/Webservice/Test/Resource/OtherResource/1?Param1=One`

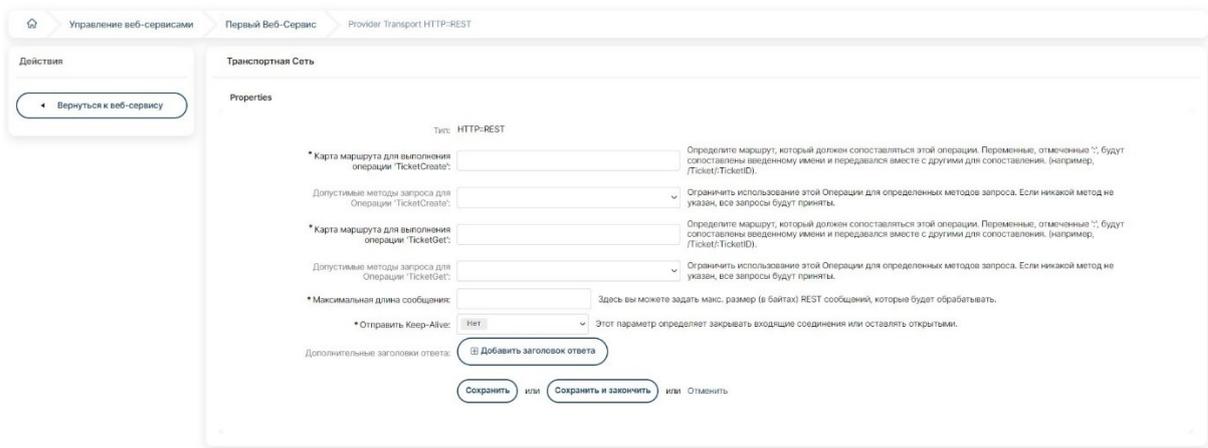
В первом примере отсутствует часть пути '/Other Resource', а также переменная: Color, во втором примере отсутствует только переменная: Color.

- Допустимые методы запроса для операции '<Имя операции>':

Методы HTTP-запроса для определения операции, которую следует использовать вместе с отображением маршрута, возможные варианты: CONNECT, DELETE, GET, HEAD, OPTIONS, PATCH, POST, PUT и TRACE.

Совершенно разные операции могут использовать один и тот же путь сопоставления, но метод запроса должен быть уникальным для каждой операции, чтобы правильно определить операцию, которую следует использовать для каждого запроса.

Рисунок 4.122. Сетевой транспорт поставщика веб-услуг (HTTP::REST)



7.2. Операции веб-сервиса

Действия, которые можно выполнить, когда вы используете RSD в качестве поставщика, называются операциями. Каждая операция принадлежит контроллеру. Контроллеры - это наборы операций или вызывающих устройств, обычно для операций с одного и того же контроллера требуются аналогичные настройки и используется один и тот же диалог настройки. Но при необходимости каждая операция может иметь независимые диалоговые окна настройки.

Имя, описание, серверная часть и сопоставления - это поля, которые обычно отображаются в каждой операции, другие специальные поля могут отображаться в диалоговых окнах конфигурации, отличных от стандартных, для удовлетворения конкретных потребностей операции.

Обычно для каждой операции существует два раздела конфигурации сопоставления: один для входящих данных, а другой для исходящих данных. Вы можете выбрать различные типы отображения (серверные части) для каждого направления отображения, поскольку их конфигурация независима друг от друга, а также независимы от операционной системы. Нормальная и наиболее распространенная практика заключается в том, что операция использует один и тот же тип отображения в обоих случаях (с инвертированной конфигурацией). Полная настройка отображения выполняется на отдельном экране, который зависит от типа отображения.

Серверная часть операции предварительно заполнена и недоступна для редактирования. Вы увидите этот параметр при выборе операции на экране редактирования веб-сервиса. Это поле носит исключительно информационный характер.

В левой части экрана в столбце действие у вас есть варианты: Вернуться к веб-сервису (отбросив все изменения с момента последнего сохранения) и «Удалить». Если вы нажмете на последнюю, откроется диалоговое окно, в котором вас спросят, хотите ли вы удалить операцию. Нажмите на кнопку «Удалить», чтобы подтвердить удаление операции и ее конфигурации, или «Отменить», чтобы закрыть диалоговое окно удаления.

Рисунок 4.123. Операции веб-сервиса

The screenshot shows a web interface for configuring a web service operation. The breadcrumb trail is: Управление веб-сервисами > Первый Веб-Сервис > Добавить операцию. The main area is titled 'Детали операции' and contains the following fields:

- * Название:** TicketCreate (Text input). Description: Имя обычно используется для вызова этой операции веб-сервиса из удаленной системы.
- Описание:** (Empty text input).
- Бэкенд для операций:** Ticket::TicketCreate (Dropdown menu). Description: Этот модуль будет вызываться для обработки запроса и подготавливать данные для ответа.
- Сопоставление для входящего запроса данных:** (Dropdown menu). Description: Запрашиваемые данные будут обработаны и преобразованы в формат поддерживаемый.
- Сопоставление данных для исходящего ответа:** (Dropdown menu). Description: Данные из ответа будут обработаны с помощью этого маппинга, чтобы преобразовать их к такому виду, который ожидает удалённая система.
- Вставить данные заявки:** Нет (Dropdown menu). Description: Включить данные заявки в ответе.

At the bottom of the form are two buttons: 'Сохранить' (Save) and 'Отменить' (Cancel).

7.3. Сетевой транспорт запрашивающего веб-сервиса

Конфигурация сетевого транспорта для отправителя запроса аналогична конфигурации для поставщика. Для запрашивающего HTTP::SOAP сетевой транспорт необходимо задать еще несколько полей.

Помимо конечной точки (URI интерфейса веб-сервиса удаленной системы для приема повторных запросов) и пространства имен, которые являются обязательными полями, вы также можете указать:

- Кодировка (например, utf-8, latin1, iso-8859-1, cp1250 и т.д.) для сообщения SOAP.
- Заголовок SOAPAction: вы можете использовать это для отправки пустого или заполненного заголовка SOAPAction. Установите значение No, и заголовок SOAPAction в сообщении SOAP будет пустой строкой, или установите значение Yes, чтобы отправить действие SOAP в формате Namespace#Action и определить разделитель (обычно "/" для .Сетевые веб-сервисы и "#" для ОСТАЛЬНЫХ).
- Аутентификация: чтобы настроить механизм аутентификации, установите значение "-", чтобы не использовать никакой аутентификации, или выберите один из списка, и появятся поля сведений.

Примечание

В настоящее время реализован только механизм аутентификации BasicAuth (HTTP). Вы можете решить, использовать его или нет, в зависимости от конфигурации удаленной системы. Если используется, вы должны указать имя пользователя и пароль для доступа к удаленной системе.

Предупреждение

Если вы введете пароль для аутентификации и после экспорта веб-сервиса в файл YAML, этот пароль будет обнаружен и записан в текстовую строку внутри файла YAML. Помните об этом и при необходимости примите меры предосторожности.

Рисунок 4.124. Сетевой транспорт запрашивающего веб-сервиса (HTTP::SOAP)

Управление веб-сервисами > Первый Веб-Сервис > Requester Transport HTTP::SOAP

Действия

← Вернуться к веб-сервису

Транспортная Сеть

Properties

Тип: HTTP::SOAP

* Конечная точка:

URI для указания точного расположения для доступа к веб-сервису, например https://local.otrs.com:8000/Webservice/Example

* Тайм-аут:

Значение Timeout для запросов.

* Установить SOAPAction:

Установить "Да" для отсылки заполненного SOAPAction заголовка. Установить "Нет" для отсылки пустого SOAPAction заголовка.

* SOAPAction схема:

Выбрать как SOAPAction должна быть построена. Некоторые веб-сервисы требуют специфичной конструкции.

* Разделитель SOAPAction.:

Символ, используемый в качестве разделителя между пространством имен и SOAP операцией. Обычно .Net веб-сервисы используют "/" в качестве разделителя.

* Пространство имен:

URI, предоставляющий SOAP методам контекст для уменьшения двусмысленности, например, url.otrs-com.soap.functions или http://www.otrs.com/GenencInterface/actions

* Схема именования запросов:

Выберите способ построения обертки функции SOAP запросов. 'FunctionName' используется в качестве примера для фактического имени Invoker или операции. 'FreeText' используется в качестве примера для фактического конфигурируемого значения.

* Схема именования ответов:

Выберите способ построения обертки функции SOAP ответов. 'FunctionName' используется в качестве примера для фактического имени Invoker или операции. 'FreeText' используется в качестве примера для фактического конфигурируемого значения.

Кодировка:

Кодировка символов для содержимого SOAP сообщений, например, utf-8, latin1, iso-8859-1, cp1250, и т.д.

Аутентификация:

Необязательный механизм аутентификации для доступа к удаленной системе.

* Использовать настройки Proxy :

Показать или скрыть параметры Proxy для подключения к удаленной системе.

* Использовать SSL параметры:

Показать или скрыть параметры SSL для подключения к удаленной системе.

Опции сортировки:

Добавить новый элемент первого уровня:

Outbound sort order for xml fields (structure starting below function name wrapper) - see documentation for SOAP transport.

или или

В случае HTTP::REST эта конфигурация также динамически расширяется в зависимости от настроенных инициаторов путем добавления сопоставления контроллера для Invoker '<InvokerName>': и действительной команды запроса для Invoker '<InvokerName>': для каждого вызова. Параметры аутентификации и SSL аналогичны параметрам HTTP::SOAP.

- Сервер

Имя хоста или IP-адрес и порт удаленной системы, если порт не указан, по умолчанию используется порт 80.

- Сопоставление контроллера для Invoker '<InvokerName>':

В этом параметре задается путь к ресурсу. Этот путь должен быть определен в соответствии с потребностями удаленного веб-сервиса и в соответствии с ее определением.

Путь может содержать переменные в форме '<VariableName>', поскольку имя каждой переменной, соответствующее текущим данным (для отправки), будет заменено соответствующим значением данных. Эти совпадающие имена и значения переменных

будут удалены из текущих данных. В зависимости от команды HTTP-запроса оставшиеся данные могут быть отправлены в виде строки JSON в теле запроса или в виде параметров запроса в URI.

Примеры:

Для данных: Var1 = One, Var2 = Two, Var3 = Three and Var4 = Four. Сопоставление контроллера: /Resource

- После замен:

/Resource

- Оставшиеся данные:

Var1 = One, Var2 = Two, Var3 = Three and Var4 = Four. Сопоставление контроллера: /Resource/:Var1

- После замен:

/Resource/One

- Оставшиеся данные:

Var2 = Two, Var3 = Three and Var4 = Four

Сопоставление контроллера: /Resource/:Var1?Param1=:Var2&Var3=:Var3

После замен:

/Resource/One?Param1=Two&Var3=Three

- Оставшиеся данные:

Var4 = Four

- Допустимая команда запроса для Вызова '<ИмяВызова>':

Это определяет используемый метод HTTP-запроса, возможные параметры: CONNECT, DELETE, GET, HEAD, OPTIONS, PATCH, POST, PUT и TRACE. Если команда не выбрана, используется команда по умолчанию.

- Команда по умолчанию

Используется как запасной вариант для всех Invoker без определенной команды запроса.

Рисунок 4.125. Сетевой транспорт запрашивающего веб-сервиса (HTTP::REST)

7.4. Вызов веб-сервиса:

Действия, которые можно выполнять, когда вы используете RSD в качестве реквестера, называются Invokers. Каждый вызывающий элемент принадлежит контроллеру (контроллеры — это наборы операций или вызывающих элементов). Обычно инициаторы из одного и того же контроллера нуждаются в одинаковых настройках и используют одни и те же диалоговые окна конфигурации. Каждый инициатор может иметь независимые диалоговые окна конфигурации, если это необходимо.

Name, Description, Backend и Mappings — это поля, которые обычно появляются в каждом вызывающем устройстве. Кроме того, список триггеров событий и другие специальные поля могут отображаться в диалоговых окнах конфигурации не по умолчанию для удовлетворения особых потребностей инициатора.

Обычно для каждого инициатора существует два раздела конфигурации сопоставления: один для входящих данных, а другой для исходящих данных. Вы можете выбрать разные типы сопоставления (бэкэнды) для каждого направления сопоставления, поскольку их конфигурация не зависит друг от друга, а также от бэкэнда вызывающего. Обычная и наиболее распространенная практика заключается в том, что инициатор использует один и тот же тип отображения в обоих случаях с инвертированной конфигурацией. Полная конфигурация сопоставления выполняется на отдельном экране, который зависит от типа сопоставления.

Серверная часть вызывающего устройства предварительно заполнена и не может быть изменена. Вы увидите этот параметр при выборе инициатора на экране редактирования веб-сервиса. Поле носит только информационный характер.

Триггеры событий — это события внутри RSD, такие как TicketCreate, ArticleSend и т. д. Они могут выступать в качестве триггеров для выполнения вызывающего объекта. Каждый вызывающий элемент должен иметь хотя бы один зарегистрированный триггер события, иначе вызывающий элемент будет бесполезен, поскольку он никогда не будет вызван. Кроме того, можно определить набор правил (условий) для каждого события, чтобы иметь больший контроль над иницированием событий. Эти правила зависят от данных объекта, связанного с событием. Асинхронное свойство триггеров

событий определяет, будет ли процесс RSD обрабатывать инициатора или он будет делегирован демону RSD.

Примечание

Демон RSD — это отдельный набор процессов, выполняющих задачи в фоновом режиме. Это не повлияет на сам процесс RSD, если удаленной системе потребуется много времени для ответа, если она недоступна или возникли проблемы с сетью. Если вы не используете демоны RSD, использующие веб-сервисы, это может замедлить работу RSD или сделать его не отвечающим. Поэтому настоятельно рекомендуется использовать асинхронные триггеры событий как можно чаще.

Чтобы добавить триггер события, сначала выберите семейство событий из первого списка, затем имя события из второго списка, затем установите свойство асинхронности (если флажок не установлен, это означает, что триггер события не будет асинхронным) и, наконец, нажмите кнопку «плюс». Будет создан новый триггер события, и он будет указан в списке триггеров событий вызывающей стороны.

В списке триггеров событий каждое событие показывает, содержит ли оно условия или нет. Кнопка редактирования рядом со свойством условия позволяет добавить или отредактировать текущие условия события.

Чтобы удалить триггер события, просто найдите триггер события, который нужно удалить, в списке триггеров событий и щелкните значок корзины в конце строки. Откроется диалоговое окно, в котором вас спросят, действительно ли вы хотите удалить триггер события. Нажмите «Удалить», чтобы удалить триггер события из списка, или «Отмена», чтобы закрыть диалоговое окно.

В левой части экрана в столбце действий у вас есть варианты: Вернуться к веб-сервису (отменив все изменения с момента последнего сохранения) и «Удалить». Если вы нажмете на последний, откроется диалоговое окно, в котором вас спросят, хотите ли вы удалить вызывающего. Нажмите кнопку «Удалить», чтобы подтвердить удаление инициатора и его конфигурации, или кнопку «Отмена», чтобы закрыть диалоговое окно удаления.

Рисунок 4.126. Вызов веб-сервиса

Управление веб-сервисами > Первый Веб-Сервис > Редактировать Invoker: Invoker one

Действия

← Вернуться к веб-сервису

🗑️ Удаление

Подробности для Invoker

*** Название:**
Имя обычно используется для вызова операции удаленного веб-сервиса.

Описание:

Invoker бэкэнд:
Этот модуль invoker будет вызываться для подготовки данных, отправляемых в удаленную систему и для обработки ответных данных.

Соответствие исходящему запросу данных:
Данные полученные из Invoker будут обработаны, для преобразования их в данные которые ожидает удаленная сторона.

Сопоставление данных для входящего запроса:
Запрашиваемые данные будут обработаны, преобразованы в формат поддерживаемый

Триггеры событий:

СОБЫТИЕ	АСИНХРОННЫЙ	УСЛОВИЕ	РЕДАКТИРОВАТЬ	УДАЛЕНИЕ
Данные не найдены.				

Этот invoker будет вызван при наступлении заданных событий.

Добавить триггер события:

Асинхронный

Чтобы добавить новое событие выберите объект и имя события и щелкните кнопку "+".
Асинхронные триггеры событий будут обрабатываться Планировщиком в фоновом режиме (рекомендуется).
Синхронные триггеры событий будут обрабатываться непосредственно при веб-запросе.

или или

7.5. Событие вызова веб-сервиса

Иногда определение события для запуска вызывающей стороны может привести к большому количеству ненужных или неправильных запросов к удаленному серверу. Условия события могут быть установлены для ограничения срабатывания инициатора в таких случаях.

Чтобы получить доступ к экрану настроек события, где можно определить условия, необходимо находиться на экране инициатора и оттуда щелкнуть значок редактирования рядом со статусом условия в событии, где это условие должно вступить в силу.

На экране настроек события на панели действий есть кнопка для возврата к экрану инициатора, а также кнопка для удаления всех условий события. По умолчанию экран предварительно заполнен первым условием. Обновите Тип связи между условиями, если планируется несколько условий, затем измените Тип связи с Условие 1, если планируется несколько полей. Оба связывающих поля принимают и, или или хог в качестве значений.

Заполните имя поля, установите тип соответствия (строка для точного соответствия, регулярное выражение для регулярного выражения или модуль проверки) и установите значение для соответствия (в случае модуля проверки полное имя класса, например: `Kernel::GenericInterface::Event:::Validation::ValidateDemo`).

Чтобы добавить дополнительные поля к условию, нажмите кнопку + в заголовке полей. Чтобы удалить поле, нажмите кнопку - в строке поля. Необходимо сохранить хотя бы одно поле для каждого условия.

Чтобы добавить дополнительные условия, нажмите кнопку под последним полем условия. Чтобы удалить условие, нажмите кнопку - в заголовке условия. Необходимо сохранить хотя бы одно условие в наборе. Чтобы удалить все условия, используйте кнопку на боковой панели.

Рисунок 4.127. Событие вызова веб-сервиса

Управление веб-сервисами > Первый Веб-Сервис > Invoker: Invoker one > Event: TicketCreate

Действия

- ← Вернуться к invoker
- 🗑️ Удалить все условия

Общие настройки

Событие: TicketCreate

Тип события: Заявки

Асинхронный:

▼ Заявки Событие Условия

Условия могут действовать только на заполненных полях.

Тип связи между Условиями: И

Условие 1

Тип связи: И

Поля

Название	Тип	Значение
	String	

Добавить новое Условие

Сохранить

Сохранить или Сохранить и закончить или Отмена

7.6. Отображение(Mapping) веб-сервисов

Бывают случаи, когда вам необходимо преобразовать данные из одного формата в другой (отобразить или изменить структуру данных), потому что обычно для взаимодействия с Удаленной Системой используется веб-сервис, то есть с высокой вероятностью не являющийся другой системой RSD и/или не мог понять структуры данных и значения RSD. В этих случаях необходимо изменить некоторые или все значения, а иногда даже имена значений (ключей) или даже всю структуру, чтобы они соответствовали ожидаемым данным на другом конце. Для выполнения этой задачи существует общий уровень отображения интерфейса.

Каждая удаленная система имеет свои собственные структуры данных, и для каждого случая можно создавать новые модули сопоставления (например, для SAP Solution Manager, поставляемого с RSD, имеется настраиваемый модуль сопоставления), но это не всегда необходимо. Модуль Mapping::Simple должен покрывать большую часть потребностей отображения.

Примечание

Когда Mapping::Simple не покрывает все потребности сопоставления для веб-сервиса, необходимо создать новый модуль сопоставления. Чтобы узнать больше о том, как создавать новые модули сопоставления, обратитесь к Руководству по разработке RSD.

Этот модуль дает вам возможность установить значения по умолчанию для сопоставления каждого ключа или значения для всех данных связи.

В начале экрана вы увидите общий раздел, в котором вы можете установить правила по умолчанию, которые будут применяться ко всем не сопоставленным ключам и значениям. Доступны три варианта, эти параметры перечислены ниже:

- Сохранить (оставить без изменений): никоим образом не затрагивает клавиши или значения.

- Игнорировать (удалить пару ключ/значение): когда это применяется к ключу, он удаляет ключ и значение, потому что, когда ключ удаляется, связанное с ним значение также удаляется. Когда это применяется к значению, удаляется только значение, сохраняя ключ, который теперь будет связан с пустым значением.
- MapTo (использовать предоставленный ключ или значение по умолчанию): все ключи и/или значения без определенного правила сопоставления будут использовать это по умолчанию, когда вы выберете эту опцию, появится новое текстовое поле для установки этого значения по умолчанию.

Нажав кнопку + для новой карты ключей, отобразится новое поле для конфигурации с одним сопоставлением. Вы можете добавить столько сопоставлений клавиш, сколько необходимо. Просто нажмите кнопку + еще раз, и новое поле сопоставления появится под существующим. Из этих полей сопоставления вы можете определить сопоставление для одного ключа со следующими параметрами:

- Точное значение(я): старая строка ключа будет заменена новой, если старый ключ точно совпадает.
- Регулярное выражение: ключевая строка будет заменена в соответствии с правилом регулярного выражения.

Нажатие кнопки + новая карта значений отобразит новую строку для карты значений. Здесь также можно определить правила для сопоставления каждого значения с теми же параметрами, что и для сопоставления ключей (точное значение и регулярное выражение). Вы можете добавить столько значений на карту, сколько необходимо, и, если вы хотите удалить одно из них, просто нажмите кнопку - для каждой строки значения сопоставления.

Удаление всего раздела сопоставления клавиш (поля) возможно, просто нажмите кнопку «-», расположенную в правом верхнем углу каждого поля, которое вы хотите удалить.

Если вам нужно удалить полную конфигурацию сопоставления: вернитесь к соответствующей операции или экрану вызывающего устройства, найдите направление сопоставления, которое вы выбрали ранее, и установите для него значение - и сохраните конфигурацию, чтобы применить изменения.

Рисунок 4.128. Отображение(Mapping) веб-сервисов

11.5. Интерфейс командной строки Веб- Сервиса

Команды `bin/radiant.Console.pl Admin::WebService::*` были разработаны для создания простых, но быстрых и мощных инструментов для работы с конфигурациями веб-сервисов. Они дают вам возможность выполнять следующие действия:

- **Add:** для создания веб-сервисов с использованием YAML-файлов в качестве источника конфигурации.
- **Update:** изменить существующий веб-сервис, конфигурация может быть изменена с использованием других или измененных YAML-файлов.
- **Dump:** сохранение текущей конфигурации веб-сервиса в файл.
- **List:** чтобы получить полный список всех веб-сервисов, зарегистрированных в системе.
- **Delete:** для удаления веб-сервиса из системы. Будьте осторожны при выполнении этой операции, ее невозможно отменить.

Пример: Создание новой конфигурации веб-сервиса:

```
shell> bin/radiant.Console.pl Admin::WebService::Add --name <webservice_name> -  
-source-path / path/to/yaml/file
```

11.6. Конфигурация Веб-сервиса

По своему дизайну веб-сервисы были задуманы так, чтобы их можно было переносить из одной системы RSD в другую, например, из среды тестирования или разработки в производственную систему. Поэтому был необходим простой способ извлечь конфигурацию веб-сервиса из базы данных и импортировать ее в другую. Для выполнения этой задачи универсальный интерфейс использует файлы YAML в качестве основы для настройки веб-сервисов.

Почему YAML? YAML — это язык разметки, разработанный для того, чтобы его было удобно читать и писать (его легче понять, чем JSON), он не имеет некоторых ограничений XML, таких как числовые теги, он открыт, стандартизирован и достаточно полон для хранения вся конфигурация веб-сервиса.

Примечание

Чтобы узнать больше о YAML пожалуйста посетите ссылку <http://www.yaml-l.org/>.

Ниже приводится конфигурация веб-сервера на основе файла конфигурации в формате YAML:

```
---  
  
Debugger:  
  
DebugThreshold: debug
```

Description: This an example of a web service configuration

Provider:

Operation:

CloseIncident:

Description: This is a test operation

MappingInbound: {}

MappingOutbound: {}

RemoteSystemGuid: "

Type: Test::Test

Test:

Description: This is a test operation

MappingInbound:

Config:

KeyMapDefault:

MapTo: "

MapType: Keep

KeyMapExact:

Prio: Priority

ValueMap:

Priority:

ValueMapExact:

Critical: 5 Very High

Information: 1 Very Low

Warning: 3 Normal

ValueMapDefault:

MapTo: 3 Normal

MapType: MapTo

Type: Simple

MappingOutbound:

Config:

KeyMapDefault:

MapTo: "

MapType: Ignore

KeyMapExact:

Priority: Prio

ValueMap:

Prio:

ValueMapExact:

1 Very Low: Information

3 Normal: Warning

5 Very High: Critical

ValueMapDefault:

MapTo: "

MapType: Ignore

Type: Simple

Type: Test::Test

Transport:

Config:

MaxLength: 10000000

Namespace: <http://www.example.com/actions>

Type: HTTP::SOAP

RemoteSystem: remote.system.description.example.com

Requester:

Invoker:

Test:

Description: This is a test invoker

Events:

- Asynchronous: 1

Condition:

Condition:

'1':

Fields:

Queue:

Match: Raw

Type: String

Type: and

ConditionLinking: and

Event: TicketCreate

- Asynchronous: 0

Event: ArticleUpdate

MappingInbound:

Type: Simple

MappingOutbound:

Type: Simple

Type: Test::Test

Transport:

Config:

Authentication:

Password: '*****'

Type: BasicAuth

User: rsd

Encoding: utf-8
Endpoint: http://www.example.com:8080/endpoint
NameSpace: <http://www.example.com/actions>
SOAPAction: Yes
SOAPActionSeparator: '#'
Type: HTTP::SOAP

Описание Конфигурации

1. Общий

- Description: краткий текст, описывающий веб-сервис.
- RemoteSystem: краткое описание Удаленной Системы.
- Debugger: хранилище для хранения настроек отладчика.
- Provider: хранилище для хранения настроек поставщика.
- Requester: хранилище для хранения настроек о запрашивающей стороне.

2. Отладчик

- DebugThreshold: уровень отладчика.

Возможные значения

- debug: все журналы хранятся в базе данных.
- info: информация, уведомления и журнал ошибок хранятся в базе данных.
- notice: ошибки уровня уведомление (notice) и ошибка (error) хранятся в базе данных.
- error: только ошибки уровня error хранятся в базе данных.

3. Поставщик

- Operation: хранилище для настроек каждой операции.
- Transport: хранилище для настроек поставщика транспортной сети.

Операция

- <OperationName>: Уникальное название операции, хранилище для своих собственных настроек (от 0..n, но без повторов).

1. <OperationName>

Этот раздел основан на операциях с типом "Test::Test". Другие операции могут содержать больше или другие настройки.

- Description: краткий текст, описывающий операцию.
- MappingInbound: хранилище для настроек отображаемых данных для данных из входящего запроса.

- MappingOutbound: хранилище для отображающихся настроек для исходящих данных.
- Type: бэкэнд операция в формате Controller::Operation.

1.1. MappingInbound

Этот раздел основан на отображаемых значениях с типом "Simple". Другие отображаемые значения могут содержать больше или другие настройки.

- Config: хранилище настроек для этих отображаемых значений.
- Type: бэкэнд сопоставления.

1.2. MappingOutbound

Тоже что и MappingInbound.

1.3. Передача (Transport)

Этот раздел основан на сетевом поставщике HTTP::SOAP, другие поставщики могут содержать больше или другие настройки.

- Config: хранилище для конкретных параметров конфигурации транспортной сети.
- Type: бэкэнд транспортной сети поставщика.

4. Запрашиваемая сторона

- Invoker: хранилище для настроек каждой метки.
- Transport: хранилище для настроек транспортной сети запрашивающей стороны.

Вызов (Invoker)

- <InvokerName>: Уникальное имя для вызывающей стороны, хранилище для своих собственных настроек (от 0..n, без повторов).

1. <InvokerName>

Этот раздел основан на типе "Test::Test", другие invoker-ры могут иметь больше или другие настройки.

- Description: короткий текст, описывающий инициатора.
- События: хранилище настроек не именованного списка действий.
- MappingInbound: хранилище для настроек отображаемых значений для данных поступающих от ответов.
- MappingOutbound: хранилище для настроек отображаемых значений для исходящих данных запроса.
- Type: серверная часть вызывающего устройства в формате Controller::Invoker.

1.1. События

- List Element: (от 0 .. n).
- Asynchronous: чтобы указать, будет ли выполнение инициатора делегировано демону RSD.

Возможные значения

- 0: не обрабатывается демоном RSD.

- 1: обрабатывается демоном RSD.
- Event: имя события триггера.

Возможные Значения (для событий заявки)

- TicketCreate
- TicketDelete
- TicketTitleUpdate
- TicketUnlockTimeoutUpdate
- TicketQueueUpdate
- TicketTypeUpdate
- TicketServiceUpdate
- TicketSLAUpdate
- TicketCustomerUpdate
- TicketPendingTimeUpdate
- TicketLockUpdate
- TicketArchiveFlagUpdate
- TicketStateUpdate
- TicketOwnerUpdate
- TicketResponsibleUpdate
- TicketPriorityUpdate
- HistoryAdd
- HistoryDelete
- TicketAccountTime
- TicketMerge
- TicketSubscribe
- TicketUnsubscribe
- TicketFlagSet
- TicketFlagDelete
- TicketSlaveLinkAdd
- TicketSlaveLinkDelete
- TicketMasterLinkDelete

Возможные Значения (для действий над заметками)

- ArticleCreate
- ArticleUpdate
- ArticleSend
- ArticleBounce
- ArticleAgentNotification
- ArticleCustomerNotification
- ArticleAutoResponse
- ArticleFlagSet
- ArticleFlagDelete
- and (и)
- or (или)
- xor

1.2. MappingInbound

Тоже что и Действие MappingInbound.

1.3. MappingOutbound

Тоже что и Действие MappingInbound.

1.4. Передача (Transport)

Этот раздел основан на транспортной сети запрашивающей стороны HTTP::SOAP, транспортные сети могут содержать больше или другие настройки.

- Config: хранилище для конкретных параметров конфигурации транспортной сети.
- Type: бэкэнд транспортной сети запрашивающей стороны.

11.7. Коннекторы (Connectors)

Коннектор, по сути, представляет собой набор действий, которые называются Операциями, если RSD выступает в качестве поставщика веб-сервиса, или Инвокерами, если RSD выступает в качестве запрашивающего веб-сервиса. Но он также может включать в себя специальные сопоставления или транспорты.

Один коннектор может иметь либо только операции, либо только инвокеры, либо и то, и другое. Соединитель может даже использовать части других соединителей, таких как сопоставления или транспорты, если они не являются специфическими для соединителя, который пытается их реализовать.

Другими словами, коннектор не ограничивается только уровнем контроллера, но при необходимости его можно расширить до уровней отображения данных или сетевого транспорта.

Из-за модульной конструкции универсального интерфейса соединитель можно рассматривать как подключаемый модуль; это означает, что путем добавления коннекторов возможности универсального интерфейса могут быть расширены с помощью: надстроек функций RSD, пользовательских модулей RSD, сторонних модулей и т. д.

Комплектные разъемы

В эту версию RSD включены следующие готовые к использованию коннекторы:

- Session (Сеанс)
- Ticket (Заявка)

1. Коннектор сеанса

Этот коннектор может создать действительный SessionID, который можно использовать в любой другой операции.

Обеспечивает:

- Operations:
 - SessionCreate
 - SessionGet

1.1. Операции

1.1.1. SessionCreate

Создает новый допустимый SessionID для использования в других операциях из других соединителей, таких как TicketCreate.

Примечание

Чтобы использовать SessionID в других операциях из других соединителей, необходимо, чтобы операция реализовывала аутентификацию по SessionID. Все остальные связанные операции могут принимать действительный SessionID в качестве метода аутентификации.

Возможные атрибуты:

```
<SessionCreate>
<!--You have a MANDATORY CHOICE of the next 2 items at this level-->
<!--Optional:-->
<UserLogin>?</UserLogin>
<!--Optional:-->
<CustomerUserLogin>?</CustomerUserLogin>
<!--Optional:-->
<Password>?</Password>
</SessionCreate>
```

1.1.2. SessionCreate

Собирает данные действительного сеанса, за исключением конфиденциальной информации, такой как пароль пользователя или токен вызова.

Примечание

Если какое-либо из его значений представляет собой сложную структуру, оно автоматически преобразуется в строку JSON.

Возможные атрибуты:

```
<SessionGet>
<SessionID>?</SessionID>
</SessionGet>
```

2. Коннектор Заявки

Этот коннектор предоставляет базовые функции для взаимодействия с заявками.

Обеспечивает:

- Операции:
 - TicketCreate
 - TicketUpdate
 - TicketGet
 - TicketSearch
 - TicketHistoryGet

2.1. Операции

2.1.1. TicketCreate

Предоставляет интерфейс для создания заявки в RSD. Заявка должна содержать заметку и может содержать несколько вложений, все определенные динамические поля также могут быть установлены в операции TicketCreate.

Возможные атрибуты:

```
<TicketCreate>

<!--You have a MANDATORY CHOICE of the next 3 items at this level-->

<!--Optional:-->

<UserLogin>?</UserLogin>

<!--Optional:-->

<CustomerUserLogin>?</CustomerUserLogin>

<!--Optional:-->

<SessionID>?</SessionID>

<!--Optional:-->

<Password>?</Password>

<Ticket>

<Title>?</Title>

<!--You have a MANDATORY CHOICE of the next 2 items at this level-->
```

<!--Optional:-->
<QueueID>?</QueueID>
<!--Optional:-->
<Queue>?</Queue>
<!--You have a CHOICE of the next 2 items at this level-->
<!--Optional:-->
<TypeID>?</TypeID>
<!--Optional:-->
<Type>?</Type>
<!--You have a CHOICE of the next 2 items at this level-->
<!--Optional:-->
<ServiceID>?</ServiceID>
<!--Optional:-->
<Service>?</Service>
<!--You have a CHOICE of the next 2 items at this level-->
<!--Optional:-->
<SLAID>?</SLAID>
<!--Optional:-->
<SLA>?</SLA>
<!--You have a MANDATORY CHOICE of the next 2 items at this level-->
<!--Optional:-->
<StateID>?</StateID>
<!--Optional:-->
<State>?</State>
<!--You have a MANDATORY CHOICE of the next 2 items at this level-->
<!--Optional:-->
<PriorityID>?</PriorityID>

```
<!--Optional:-->
<Priority?></Priority>
<!--You have a CHOICE of the next 2 items at this level-->
<!--Optional:-->
<OwnerID?></OwnerID>
<!--Optional:-->
<Owner?></Owner>
<!--You have a CHOICE of the next 2 items at this level-->
<!--Optional:-->
<ResponsibleID?></ResponsibleID>
<!--Optional:-->
<Responsible?></Responsible>
<CustomerUser?></CustomerUser>
<!--Optional:-->
<CustomerID?></CustomerID>
<!--Optional:-->
<PendingTime>
<!--You have a CHOICE of the next and the other 5 items at this level-->
<Diff?></Diff>
<Year?></Year>
<Month?></Month>
<Day?></Day>
<Hour?></Hour>
<Minute?></Minute>
</PendingTime>
</Ticket>
<Article>
```

```
<!--You have a CHOICE of the next 2 items at this level-->
<!--Optional:-->
<CommunicationChannelID?></CommunicationChannelID>
<!--Optional: Possible values Email, Internal or Phone-->
<CommunicationChannel?></CommunicationChannel>
<IsVisibleForCustomer?></IsVisibleForCustomer>
<!--You have a CHOICE of the next 2 items at this level-->
<!--Optional:-->
<SenderTypeID?></SenderTypeID>
<!--Optional:-->
<SenderType?></SenderType>
<!--Optional:-->
<From?></From>
<Subject?></Subject>
<Body?></Body>
<!--You have a CHOICE of the next 2 items at this level-->
<!--Optional:-->
<ContentType?></ContentType>
<Charset?></Charset>
<MimeType?></MimeType>
<!--Optional:-->
<HistoryType?></HistoryType>
<!--Optional:-->
<HistoryComment?></HistoryComment>
<!--Optional:-->
<AutoResponseType?></AutoResponseType>
<!--Optional:-->
```

```
<TimeUnit>?</TimeUnit>

<!--Optional:-->

<NoAgentNotify>?</NoAgentNotify>

<!--Zero or more repetitions:-->

<ForceNotificationToUserID>?</ForceNotificationToUserID>

<!--Zero or more repetitions:-->

<ExcludeNotificationToUserID>?</ExcludeNotificationToUserID>

<!--Zero or more repetitions:-->

<ExcludeMuteNotificationToUserID>?</ExcludeMuteNotificationToUserID>

</Article>

<!--Zero or more repetitions:-->

<DynamicField>

<Name>?</Name>

<!--1 or more repetitions:-->

<Value>?</Value>

</DynamicField>

<!--Zero or more repetitions:-->

<Attachment>

<Content>cid:61886944659</Content>

<ContentType>?</ContentType>

<Filename>?</Filename>

</Attachment>

</TicketCreate>
```

2.1.2. TicketUpdate

Операция TicketUpdate добавляет возможность изменять атрибуты существующей заявки или добавлять новую заметку, включая вложения и все определенные динамические поля для заявки и новой заметке.

Примечание

Нет необходимости создавать новую заметку для изменения атрибута заявки.

Возможные атрибуты:

```
<TicketUpdate>
<!--You have a MANDATORY CHOICE of the next 3 items at this level-->
<!--Optional:-->
<UserLogin>?</UserLogin>
<!--Optional:-->
<CustomerUserLogin>?</CustomerUserLogin>
<!--Optional:-->
<SessionID>?</SessionID>
<!--Optional:-->
<Password>?</Password>
<!--You have a CHOICE of the next 2 items at this level-->
<TicketID>?</TicketID>
<TicketNumber>?</TicketNumber>
<!--Optional:-->
<Ticket>
<!--Optional:-->
<Title>?</Title>
<!--You have a CHOICE of the next 2 items at this level-->
<!--Optional:-->
<QueueID>?</QueueID>
<!--Optional:-->
<Queue>?</Queue>
<!--You have a CHOICE of the next 2 items at this level-->
<!--Optional:-->
```

<TypeID>?</TypeID>
<!--Optional:-->
<Type>?</Type>
<!--You have a CHOICE of the next 2 items at this level-->
<!--Optional:-->
<ServiceID>?</ServiceID>
<!--Optional:-->
<Service>?</Service>
<!--You have a CHOICE of the next 2 items at this level-->
<!--Optional:-->
<SLAID>?</SLAID>
<!--Optional:-->
<SLA>?</SLA>
<!--You have a CHOICE of the next 2 items at this level-->
<!--Optional:-->
<StateID>?</StateID>
<!--Optional:-->
<State>?</State>
<!--You have a CHOICE of the next 2 items at this level-->
<!--Optional:-->
<PriorityID>?</PriorityID>
<!--Optional:-->
<Priority>?</Priority>
<!--You have a CHOICE of the next 2 items at this level-->
<!--Optional:-->
<OwnerID>?</OwnerID>
<!--Optional:-->

```
<Owner>?</Owner>

<!--You have a CHOICE of the next 2 items at this level-->

<!--Optional:-->

<ResponsibleID>?</ResponsibleID>

<!--Optional:-->

<Responsible>?</Responsible>

<!--Optional:-->

<CustomerUser>?</CustomerUser>

<!--Optional:-->

<CustomerID>?</CustomerID>

<!--Optional:-->

<PendingTime>

<!--You have a CHOICE of the next and the other 5 items at this level-->

<Diff>?</Diff>

<Year>?</Year>

<Month>?</Month>

<Day>?</Day>

<Hour>?</Hour>

<Minute>?</Minute>

</PendingTime>

</Ticket>

<!--Optional:-->

<Article>

<!--You have a CHOICE of the next 2 items at this level-->

<!--Optional:-->

<CommunicationChannelID>?</CommunicationChannelID>

<!--Optional: Possible values Email, Internal or Phone-->
```

```
<CommunicationChannel>?</CommunicationChannel>

<IsVisibleForCustomer>?</IsVisibleForCustomer>

<!--You have a CHOICE of the next 2 items at this level-->

<!--Optional:-->

<SenderId>?</SenderId>

<!--Optional:-->

<SenderType>?</SenderType>

<!--Optional:-->

<From>?</From>

<Subject>?</Subject>

<Body>?</Body>

<!--You have a CHOICE of the next 2 items at this level-->

<!--Optional:-->

<ContentType>?</ContentType>

<Charset>?</Charset>

<MimeType>?</MimeType>

<!--Optional:-->

<HistoryType>?</HistoryType>

<!--Optional:-->

<HistoryComment>?</HistoryComment>

<!--Optional:-->

<AutoResponseType>?</AutoResponseType>

<!--Optional:-->

<TimeUnit>?</TimeUnit>

<!--Optional:-->

<NoAgentNotify>?</NoAgentNotify>

<!--Zero or more repetitions:-->
```

```
<ForceNotificationToUserID?></ForceNotificationToUserID>

<!--Zero or more repetitions:-->

<ExcludeNotificationToUserID?></ExcludeNotificationToUserID>

<!--Zero or more repetitions:-->

<ExcludeMuteNotificationToUserID?></ExcludeMuteNotificationToUserID>

</Article>

<!--Zero or more repetitions:-->

<DynamicField>

<Name?></Name>

<!--1 or more repetitions:-->

<Value?></Value>

</DynamicField>

<!--Zero or more repetitions:-->

<Attachment>

<Content>cid:166861569966</Content>

<ContentType?></ContentType>

<Filename?></Filename>

</Attachment>

</TicketUpdate>
```

2.1.3. TicketGet

Эта операция используется для получения всех атрибутов заявки, включая динамические поля, все заметки и все вложения, принадлежащие заявке.

Возможные атрибуты:

```
<TicketGet>

<!--You have a MANDATORY CHOICE of the next 3 items at this level-->
```

```
<!--Optional:-->
<UserLogin>?</UserLogin>
<!--Optional:-->
<CustomerUserLogin>?</CustomerUserLogin>
<!--Optional:-->
<SessionID>?</SessionID>
<!--Optional:-->
<Password>?</Password>
<!--1 or more repetitions:-->
<TicketID>?</TicketID>
<!--Optional:-->
<DynamicFields>?</DynamicFields>
<!--Optional:-->
<Extended>?</Extended>
<!--Optional:-->
<AllArticles>?</AllArticles>
<!--Optional:-->
<ArticleSenderType>?</ArticleSenderType>
<!--Optional:-->
<ArticleOrder>?</ArticleOrder>
<!--Optional:-->
<ArticleLimit>?</ArticleLimit>
<!--Optional:-->
<Attachments>?</Attachments>
<!--Optional:-->
<GetAttachmentContents>?</GetAttachmentContents>
<!--Optional:-->
```

```
<HTMLBodyAsAttachment?></HTMLBodyAsAttachment>
```

```
</TicketGet>
```

2.1.4. TicketSearch

Операция TicketSearch возвращает список идентификаторов заявок, соответствующих определенным критериям.

Возможные атрибуты:

```
<TicketSearch>
```

```
<!--You have a MANDATORY CHOICE of the next 3 items at this level-->
```

```
<!--Optional:-->
```

```
<UserLogin?></UserLogin>
```

```
<!--Optional:-->
```

```
<CustomerUserLogin?></CustomerUserLogin>
```

```
<!--Optional:-->
```

```
<SessionID?></SessionID>
```

```
<!--Optional:-->
```

```
<Password?></Password>
```

```
<!--Optional:-->
```

```
<Limit?></Limit>
```

```
<!--Zero or more repetitions:-->
```

```
<TicketNumber?></TicketNumber>
```

```
<!--Zero or more repetitions:-->
```

```
<Title?></Title>
```

```
<!--Zero or more repetitions:-->
```

```
<Queues?></Queues>
```

```
<!--Zero or more repetitions:-->
```

<QueueIDs>?</QueueIDs>

<!--Optional:-->

<UseSubQueues>?</UseSubQueues>

<!--Zero or more repetitions:-->

<Types>?</Types>

<!--Zero or more repetitions:-->

<TypeIDs>?</TypeIDs>

<!--Zero or more repetitions:-->

<States>?</States>

<!--Zero or more repetitions:-->

<StateIDs>?</StateIDs>

<!--Zero or more repetitions:-->

<StateType>?</StateType>

<!--Zero or more repetitions:-->

<StateTypeIDs>?</StateTypeIDs>

<!--Zero or more repetitions:-->

<Priorities>?</Priorities>

<!--Zero or more repetitions:-->

<PriorityIDs>?</PriorityIDs>

<!--Zero or more repetitions:-->

<Services>?</Services>

<!--Zero or more repetitions:-->

<ServiceIDs>?</ServiceIDs>

<!--Zero or more repetitions:-->

<SLAs>?</SLAs>

<!--Zero or more repetitions:-->

<SLAIDs>?</SLAIDs>

<!--Zero or more repetitions:-->
<Locks?></Locks>
<!--Zero or more repetitions:-->
<LockIDs?></LockIDs>
<!--Zero or more repetitions:-->
<OwnerIDs?></OwnerIDs>
<!--Zero or more repetitions:-->
<ResponsibleIDs?></ResponsibleIDs>
<!--Zero or more repetitions:-->
<WatchUserIDs?></WatchUserIDs>
<!--Zero or more repetitions:-->
<CustomerID?></CustomerID>
<!--Zero or more repetitions:-->
<CustomerUserLogin?></CustomerUserLogin>
<!--Zero or more repetitions:-->
<CreatedUserIDs?></CreatedUserIDs>
<!--Zero or more repetitions:-->
<CreatedTypes?></CreatedTypes>
<!--Zero or more repetitions:-->
<CreatedTypeID?></CreatedTypeID>
<!--Zero or more repetitions:-->
<CreatedPriorities?></CreatedPriorities>
<!--Zero or more repetitions:-->
<CreatedPriorityIDs?></CreatedPriorityIDs>
<!--Zero or more repetitions:-->
<CreatedStates?></CreatedStates>
<!--Zero or more repetitions:-->

```
<CreatedStateIDs>?</CreatedStateIDs>

<!--Zero or more repetitions:-->

<CreatedQueues>?</CreatedQueues>

<!--Zero or more repetitions:-->

<CreatedQueueIDs>?</CreatedQueueIDs>

<!--Zero or more repetitions:-->

<DynamicField>

<Name>?</Name>

<!--You have a MANDATORY CHOICE of the next 6 items at this level-->

<!--Optional:-->

<Equals>?</Equals>

<!--Optional:-->

<Like>?</Like>

<!--Optional:-->

<GreaterThan>?</GreaterThan>

<!--Optional:-->

<GreaterThanEquals>?</GreaterThanEquals>

<!--Optional:-->

<SmallerThan>?</SmallerThan>

<!--Optional:-->

<SmallerThanEquals>?</SmallerThanEquals>

</DynamicField>

<!--Optional:-->

<Ticketflag>

<!--Optional:-->

<Seen>?</Seen>

</Ticketflag>
```

<!--Optional:-->
<From>?</From>
<!--Optional:-->
<To>?</To>
<!--Optional:-->
<Cc>?</Cc>
<!--Optional:-->
<Subject>?</Subject>
<!--Optional:-->
<Body>?</Body>
<!--Optional:-->
<FullTextIndex>?</FullTextIndex>
<!--Optional:-->
<ContentSearch>?</ContentSearch>
<!--Optional:-->
<ConditionInline>?</ConditionInline>
<!--Optional:-->
<ArticleCreateTimeOlderMinutes>?</ArticleCreateTimeOlderMinutes>
<!--Optional:-->
<ArticleCreateTimeNewerMinutes>?</ArticleCreateTimeNewerMinutes>
<!--Optional:-->
<ArticleCreateTimeNewerDate>?</ArticleCreateTimeNewerDate>
<!--Optional:-->
<ArticleCreateTimeOlderDate>?</ArticleCreateTimeOlderDate>
<!--Optional:-->
<TicketCreateTimeOlderMinutes>?</TicketCreateTimeOlderMinutes>
<!--Optional:-->

```
<ATicketCreateTimeNewerMinutes>?</ATicketCreateTimeNewerMinutes>

<!--Optional:-->

<TicketCreateTimeNewerDate>?</TicketCreateTimeNewerDate>

<!--Optional:-->

<TicketCreateTimeOlderDate>?</TicketCreateTimeOlderDate>

<!--Optional:-->

<TicketLastChangeTimeOlderMinutes>?</TicketLastChangeTimeOlderMinutes>

<!--Optional:-->

<TicketLastChangeTimeNewerMinutes>?</TicketLastChangeTimeNewerMinutes>

<!--Optional:-->

<TicketLastChangeTimeNewerDate>?</TicketLastChangeTimeNewerDate>

<!--Optional:-->

<TicketLastChangeTimeOlderDate>?</TicketLastChangeTimeOlderDate>

<!--Optional:-->

<TicketChangeTimeOlderMinutes>?</TicketChangeTimeOlderMinutes>

<!--Optional:-->

<TicketChangeTimeNewerMinutes>?</TicketChangeTimeNewerMinutes>

<!--Optional:-->

<TicketChangeTimeNewerDate>?</TicketChangeTimeNewerDate>

<!--Optional:-->

<TicketChangeTimeOlderDate>?</TicketChangeTimeOlderDate>

<!--Optional:-->

<TicketCloseTimeOlderMinutes>?</TicketCloseTimeOlderMinutes>

<!--Optional:-->

<TicketCloseTimeNewerMinutes>?</TicketCloseTimeNewerMinutes>

<!--Optional:-->

<TicketCloseTimeNewerDate>?</TicketCloseTimeNewerDate>
```

```
<!--Optional:-->
<TicketCloseTimeOlderDate>?</TicketCloseTimeOlderDate>
<!--Optional:-->
<TicketPendingTimeOlderMinutes>?</TicketPendingTimeOlderMinutes>
<!--Optional:-->
<TicketPendingTimeNewerMinutes>?</TicketPendingTimeNewerMinutes>
<!--Optional:-->
<TicketPendingTimeNewerDate>?</TicketPendingTimeNewerDate>
<!--Optional:-->
<TicketPendingTimeOlderDate>?</TicketPendingTimeOlderDate>
<!--Optional:-->
<TicketEscalationTimeOlderMinutes>?</TicketEscalationTimeOlderMinutes>
<!--Optional:-->
<TTicketEscalationTimeNewerMinutes>?</TTicketEscalationTimeNewerMinutes>
<!--Optional:-->
<TicketEscalationTimeNewerDate>?</TicketEscalationTimeNewerDate>
<!--Optional:-->
<TicketEscalationTimeOlderDate>?</TicketEscalationTimeOlderDate>
<!--Optional:-->
<ArchiveFlags>?</ArchiveFlags>
<!--Zero or more repetitions:-->
<OrderBy>?</OrderBy>
<!--Zero or more repetitions:-->
<SortBy>?</SortBy>
<!--Zero or more repetitions:-->
<CustomerUserID>?</CustomerUserID>
</TicketSearch>
```

2.1.5. TicketHistoryGet

Эта операция используется для получения всех записей истории из заявки или заявок.

Возможные атрибуты:

```
<TicketHistoryGet>
<!--You have a MANDATORY CHOICE of the next 2 items at this level-->
<!--Optional:-->
<UserLogin>?</UserLogin>
<!--Optional:-->
<SessionID>?</SessionID>
<!--Optional:-->
<Password>?</Password>
<!--1 or more repetitions:-->
<TicketID>?</TicketID>
</TicketHistoryGet>
```

Примеры:

1. Конфигурация Веб-сервиса

Ниже приведен базовый, но полный файл конфигурации веб-сервиса в формате YAML для использования всех операций Ticket Connector с сетевым транспортом SOAP. Чтобы использовать его в RSD, вам необходимо скопировать содержимое, сохранить его в файл и назвать его GenericTicketConnectorSOAP.yml, а затем импортировать его в RSD на экране веб-служб в панели администратора, нажав кнопку «Добавить веб-службу». " на обзорном экране, а затем нажмите кнопку "Импортировать веб-службу" на экране добавления.

Debugger:

DebugThreshold: debug

TestMode: 0

Description: Ticket Connector SOAP Sample

FrameworkVersion: 3.4.x git

Provider:

Operation:

SessionCreate:

Description: Creates a Session

MappingInbound: {}

MappingOutbound: {}

Type: Session::SessionCreate

TicketCreate:

Description: Creates a Ticket

MappingInbound: {}

MappingOutbound: {}

Type: Ticket::TicketCreate

TicketUpdate:

Description: Updates a Ticket

MappingInbound: {}

MappingOutbound: {}

Type: Ticket::TicketUpdate

TicketGet:

Description: Retrieves Ticket data

MappingInbound: {}

MappingOutbound: {}

Type: Ticket::TicketGet

TicketSearch:

Description: Search for Tickets

MappingInbound: {}

MappingOutbound: {}

Type: Ticket::TicketSearch

TicketHistoryGet:

Description: Retrieves history of a Ticket

MappingInbound: {}

MappingOutbound: {}

Type: Ticket::TicketHistoryGet

Transport:

Config:

MaxLength: 100000000

NameSpace: <https://radiantsystem.ru/>

Type: HTTP::SOAP

RemoteSystem: "

Requester:

Transport:

Type: "

Аналогичный пример можно сделать для сетевого транспорта REST, веб-сервиса REST используют операции HTTP, такие как «POST», «GET», «PUT», «PATCH» и т. д. Эти операции в сочетании с путем URI, называемым ресурсом, определяют универсальный RSD. Интерфейсная операция или Invoker (в зависимости от способа связи).

В следующем примере используется ресурс /Session для SessionCreate, ресурс /Ticket для TicketSearch и TicketCreate и ресурс /Ticket/{TicketID} для TicketGet и TicketUpdate (где {TicketID} — фактическое значение TicketID заявки, например, /Ticket/123). Чтобы использовать его в RSD, вам необходимо скопировать содержимое, сохранить его в файл и назвать его GenericTicketConnectorREST.yml, а затем импортировать его в RSD на экране веб-служб в панели администратора, нажав кнопку «Добавить веб-службу». " на обзорном экране, а затем нажмите кнопку "Импортировать веб-службу" на экране добавления.

Debugger:

DebugThreshold: debug

TestMode: '0'

Description: Ticket Connector REST Sample

FrameworkVersion: 3.4.x git

Provider:

Operation:

SessionCreate:

Description: Creates a Session

MappingInbound: {}

MappingOutbound: {}

Type: Session::SessionCreate

TicketCreate:

Description: Creates a Ticket

MappingInbound: {}

MappingOutbound: {}

Type: Ticket::TicketCreate

TicketGet:

Description: Retrieves Ticket data

MappingInbound: {}

MappingOutbound: {}

Type: Ticket::TicketGet

TicketSearch:

Description: Search for Tickets

MappingInbound: {}

MappingOutbound: {}

Type: Ticket::TicketSearch

TicketUpdate:

Description: Updates a Ticket

MappingInbound: {}

MappingOutbound: {}

Type: Ticket::TicketUpdate

TicketHistoryGet:

Description: Retrieves history of a Ticket

MappingInbound: {}

MappingOutbound: {}

Type: Ticket::TicketHistoryGet

Transport:

Config:

KeepAlive: "

MaxLength: '100000000'

RouteOperationMapping:

SessionCreate:

RequestMethod:

- POST

Route: /Session

TicketCreate:

RequestMethod:

- POST

Route: /Ticket

TicketGet:

RequestMethod:

- GET

Route: /Ticket/:TicketID

TicketSearch:

RequestMethod:

- GET

Route: /Ticket

TicketUpdate:

RequestMethod:

- PATCH

Route: /Ticket/:TicketID

TicketHistoryGet:

RequestMethod:

- GET

Route: /TicketHistory/:TicketID

Type: HTTP::REST

RemoteSystem: "

Requester:

Transport:

Type: "

2. Запрашивающая сторона Perl SOAP

Следующий код представляет собой сценарий Perl, который может подключаться к RSD через универсальный интерфейс. Для выполнения операций, предоставляемых Ticket Connector, он использует два модуля Perl CPAN SOAP::Lite и Data::Dumper. Пожалуйста, убедитесь, что ваша среда способна использовать эти модули, прежде чем пытаться запустить скрипт.

```
#!/usr/bin/perl -w # --
```

```
# radiant.SOAPRequest.pl - sample to send a SOAP request to RSD Generic Interface  
Ticket Connector
```

```

# Copyright (C) 2001-2022 RSD AG, https://radiantsystem.ru/ # --
# This program is free software: you can redistribute it and/or modify
# it under the terms of the GNU General Public License as published by
# the Free Software Foundation, either version 3 of the License, or
# (at your option) any later version. #
# This program is distributed in the hope that it will be useful,
# but WITHOUT ANY WARRANTY; without even the implied warranty of
# MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the
# GNU General Public License for more details. #
# You should have received a copy of the GNU General Public License
# along with this program. If not, see https://www.gnu.org/licenses/gpl-3.0.txt. # --

use strict;

use warnings;

# use ../ as lib location use File::Basename;

use FindBin qw($RealBin); use lib dirname($RealBin);

use SOAP::Lite;

use Data::Dumper;

# ---

# Variables to be defined.

# this is the URL for the web service # the format is

#                                     <HTTP_TYPE>://<RSD_FQDN>/nph-
genericinterface.pl/Webservice/<WEB_SERVICE_NAME> # or

```

```
# <HTTP_TYPE>:://<RSD_FQDN>/nph-
genericinterface.pl/WebserviceID/<WEB_SERVICE_ID>

my $URL = 'http://localhost/rsd/nph-genericinterface.pl/Webservice/GenericTicketConnector';

# this name space should match the specified name space in the SOAP transport for
the web service.

my $NameSpace = https://radiantsystem.ru/;

# this is operation to execute, it could be TicketCreate, TicketUpdate, TicketGet,
TicketSearch

# or SessionCreate. and they must to be defined in the web service. my $Operation =
'TicketCreate';

# this variable is used to store all the parameters to be included on a request in XML
format. Each

# operation has a determined set of mandatory and non mandatory parameters to work
correctly. Please

# check the RSD Admin Manual in order to get a complete list of parameters. my
$xmlData = '

<UserLogin>some user login</UserLogin>

<Password>some password</Password>

<Ticket>

<Title>some title</Title>

<CustomerUser>some customer user login</CustomerUser>

<Queue>some queue</Queue>

<State>some state</State>

<Priority>some priority</Priority>

</Ticket>

<Article>

<Subject>some subject</Subject>

<Body>some body</Body>
```

```

<ContentType>text/plain; charset=utf8</ContentType>

</Article> ';

# ---

# create a SOAP::Lite data structure from the provided XML data structure.

my $SOAPData = SOAP::Data

->type( 'xml' => $XMLData );

my $SOAPObject = SOAP::Lite

->uri($NameSpace)

->proxy($URL)

->$Operation($SOAPData);

# check for a fault in the soap code.

if ( $SOAPObject->fault ) {

print $SOAPObject->faultcode, " ", $SOAPObject->faultstring, "\n";

}

# otherwise print the results.

else {

# get the XML response part from the SOAP message.

my $XMLResponse = $SOAPObject->context()->transport()->proxy()-
>http_response()-
>content();

# deserialize response (convert it into a perl structure).

my $Deserialized = eval {

SOAP::Deserializer->deserialize($XMLResponse);

};

```

```

# remove all the headers and other not needed parts of the SOAP message.

my $Body = $Deserialized->body();

# just output relevant data and no the operation name key (like
TicketCreateResponse).

for my $ResponseKey ( keys %{$Body} ) {

print Dumper( $Body->{$ResponseKey} );

}

}

```

3. Запросчик Perl REST

Следующий код представляет собой сценарий Perl, который может подключаться к RSD через универсальный интерфейс. Для выполнения операций, предоставляемых Ticket Connector, он использует три модуля Perl CPAN JSON, REST::Client и Data::Dumper. Пожалуйста, убедитесь, что ваша среда способна использовать эти модули, прежде чем пытаться запустить скрипт.

```

#!/usr/bin/perl # --

# Copyright (C) 2001-2022 RSD AG, https://radiantsystem.ru/ # --

# This program is free software: you can redistribute it and/or modify
# it under the terms of the GNU General Public License as published by
# the Free Software Foundation, either version 3 of the License, or
# (at your option) any later version. #

# This program is distributed in the hope that it will be useful,
# but WITHOUT ANY WARRANTY; without even the implied warranty of
# MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the
# GNU General Public License for more details. #

# You should have received a copy of the GNU General Public License
# along with this program. If not, see https://www.gnu.org/licenses/gpl-3.0.txt. # --

```

```

use strict;

use warnings;

use utf8;

## nofilter(TidyAll::Plugin::RSD::Perl::Dumper) # use ../ as lib location

use File::Basename;

use FindBin qw($RealBin);

use lib dirname($RealBin);

use JSON;

use REST::Client;

# This is the HOST for the web service the format is:
# <HTTP_TYPE>://<RSD_FQDN>/nph-genericinterface.pl
my $Host = 'http://localhost/rsd/nph-genericinterface.pl';

my $RestClient = REST::Client->new(
{
host => $Host,
}
);

# These are the Controllers and Providers the format is:
# /Webservice/<WEB_SERVICE_NAME>/<RESOURCE>/<REQUEST_VALUE>
# or
# /WebserviceID/<WEB_SERVICE_ID>/<RESOURCE>/<REQUEST_VALUE> #
# See the documentation on how to setup Providers. #

# This example will retrieve the Ticket with the TicketID = 1 (<REQUEST_VALUE>)
my $GetControllerAndRequest =
'/Webservice/GenericTicketConnectorREST/Ticket/1';

```

```

# This example is the base URL for Ticket Create

my          $CreateControllerAndRequest          =
'/Webservice/GenericTicketConnectorREST/Ticket';

# This example will update the Ticket with the TicketID = 1 (<REQUEST_VALUE>)

my          $UpdateControllerAndRequest          =
'/Webservice/GenericTicketConnectorREST/Ticket/1';

# This is the base URL for Ticket Search

my          $SearchControllerAndRequest          =
'/Webservice/GenericTicketConnectorREST/Ticket';

# This is the base URL for Ticket history with the TicketID = 1 (<REQUEST_VALUE>)

my          $HistoryControllerAndRequest          =
'/Webservice/GenericTicketConnectorREST/TicketHistory/1';

# TicketGet Example

# See the documentation of RSDGenericInterfaceREST on how to setup

#          - webservice

#          - transport

#          - operations

my $GetParams = {

  UserLogin => "some agent user login",          # to be filled
  with valid agent login Password => "some agent user password",
  # to be filled with valid agent password

};

# Build GetParams as part of the URL for REST-GET requests my $QueryParams =
$RestClient->buildQuery( %{$GetParams} );

```

```

$GetControllerAndRequest .= $QueryParams;

$RestClient->GET($GetControllerAndRequest);

my $GetResponseCode = $RestClient->responseCode(); if ( $GetResponseCode ne
'200' ) {

print "Get request failed, response code was: $GetResponseCode\n";

}

else {

# If the request was answered correctly, we receive a JSON string here. my
$ResponseContent = $RestClient->responseContent();

my $Data = decode_json $ResponseContent;

# Just to print out the returned Data structure: use Data::Dumper;

print "Get response was:\n"; print Dumper($Data);

}

# TicketSearch Example

# See the documentation of RSDGenericInterfaceREST on how to setup

# - webservice

# - transport

# - operations

my $SearchParams = {

  UserLogin => "some agent user login",           # to be filled with valid agent login

```

```

Password => "some agent user password",           # to be filled with valid agent
password Queues => ['Raw'],

};

# Build SearchParams as part of the URL for REST-GET requests

$QueryParams = $RestClient->buildQuery( %{$SearchParams} );

$searchControllerAndRequest .= $QueryParams;

$RestClient->GET($searchControllerAndRequest);

# If the host isn't reachable, wrong configured or couldn't serve the requested page:
my $searchResponseCode = $RestClient->responseCode();

if ( $searchResponseCode ne '200' ) {

print "Search request failed, response code was: $searchResponseCode\n";

}

else {

# If the request was answered correctly, we receive a JSON string here.

my $responseContent = $RestClient->responseContent();

my $data = decode_json $responseContent;

# Just to print out the returned Data structure:

use Data::Dumper;

print "Search Response was:\n";

print Dumper($data);

}

```

```

# TicketCreate Example

# See the documentation of RSDGenericInterfaceREST on how to setup

# - webservice

# - transport

# - operations

my $CreateOrUpdateParams = {

  UserLogin => "some agent user login",           # to be filled
  with valid agent login Password => "some agent user password",
  # to be filled with valid agent password

  Ticket => {

    Title          => 'some ticket title',

    Queue          => 'Raw',

    Lock           => 'unlock',

    Type           => 'Unclassified',

    State          => 'new',

    Priority       => '3 normal',

    Owner          => 'some agent user login',

    CustomerUser  => 'customer-1',

  },

  Article => {

    Subject        => 'some subject',

    Body           => 'some body',

    ContentType   => 'text/plain; charset=utf8',

  },

};

my $CreateJSONParams = encode_json $CreateOrUpdateParams;

my @CreateRequestParam = (

  $CreateControllerAndRequest,

```

```

$CreateJSONParams

);

# We have to use REST-POST requests in order to send UserLogin and Password
correctly

# though other REST methods would fit better.

$RestClient->POST(@CreateRequestParam);

# If the host isn't reachable, wrong configured or couldn't serve the requested page:

my $CreateResponseCode = $RestClient->responseCode();

if ( $CreateResponseCode ne '200' ) {
    print "Create request failed, response code was: $CreateResponseCode\n";
}
else {

# If the request was answered correctly, we receive a JSON string here.
my $ResponseContent = $RestClient->responseContent();
my $Data = decode_json $ResponseContent;
# Just to print out the returned Data structure:
use Data::Dumper;
print "Create Response was:\n";
print Dumper($Data);
}

# TicketUpdate Example

```

```

# See the documentation of RSDGenericInterfaceREST on how to setup

# - webservice

# - transport

# - operations

my $UpdateJSONParams = encode_json $CreateOrUpdateParams;

my @UpdateRequestParam = (

$updateControllerAndRequest,

$updateJSONParams

);

# We have to use REST-PATCH requests in order to send UserLogin and Password
correctly # though other REST methods would fit better.

$RestClient->PATCH(@UpdateRequestParam);

# If the host isn't reachable, wrong configured or couldn't serve the requested page:
my $UpdateResponseCode = $RestClient->responseCode();

if ( $UpdateResponseCode ne '200' ) {

print "Update request failed, response code was: $UpdateResponseCode\n";

}

else {

# If the request was answered correctly, we receive a JSON string here.

my $ResponseContent = $RestClient->responseContent();

my $Data = decode_json $ResponseContent;

# Just to print out the returned Data structure:

use Data::Dumper;

print "Update response was:\n";

```

```

print Dumper($Data);

}

# TicketHistoryGet Example

# See the documentation of RSDGenericInterfaceREST on how to setup

# - webservice

# - transport

# - operations

my $HistoryParams = {

    UserLogin => "some agent user login", # to be filled
    with valid agent login Password => "some agent user password",
    # to be filled with valid agent password TicketID => [1],

};

# Build SearchParams as part of the URL for REST-GET requests

$queryParams = $RestClient->buildQuery( %{$HistoryParams} );

$HistoryControllerAndRequest .= $queryParams;

$RestClient->GET($HistoryControllerAndRequest);

# If the host isn't reachable, wrong configured or couldn't serve the requested page:

my $HistoryResponseCode = $RestClient->responseCode();

if ( $HistoryResponseCode ne '200' ) {

    print "History request failed, response code was: $HistoryResponseCode\n";

}

```

```

else {

# If the request was answered correctly, we receive a JSON string here.

my $ResponseContent = $RestClient->responseContent();

my $Data = decode_json $ResponseContent;

# Just to print out the returned Data structure:

use Data::Dumper;

print "History Response was:\n";

print Dumper($Data);

}

```

4. cURL Примеры для REST запросов

В приведенном выше примере настройки REST для Generic Ticket Connector имеем следующее:

Для Ticket Create: используйте POST метод в /Ticket path.

Для Ticket Search: используйте GET метод в /Ticket path.

Для Ticket Update: используйте метод PATCH в /Ticket/{TicketID} пути (где {TicketID} является шаблоном представленным: TicketID в настройке транспорта)

Для Ticket Get: используйте метод Get в /Ticket/{TicketID} пути (где {TicketID} является шаблоном представленным: TicketID в настройке транспорта)

Для Ticket History Get: используйте метод GET для пути /TicketHistory/{TicketID} (где {TicketID} — это шаблон, представленный: TicketID в конфигурации транспорта).

4.1. Создать новую заявку

cURL Команда:

```

shell> curl "http://localhost/rsd/nph-genericinterface.pl/Webservice/
GenericTicketConnectorREST/Ticket?UserLogin=agent&Password=123" -H
"Content-Type:

application/json" -d '{"Ticket":{"Title":"REST Create Test", "Type": "Unclassified
", "Queue":"Raw", "State":"open", "Priority":"3 normal", "CustomerUser":

```

```
\\"customer\\"},\\"Article\\":{\\"Subject\\":\\"Rest Create Test\\",\\"Body\\":\\"This is only a test
\\",\\"ContentType\\":\\"text/plain; charset=utf8\\"}" -X POST
```

Ответ:

```
{
  "ArticleID":5484,
  "TicketNumber":"1001936",
  "TicketID":"1686"
}
```

4.2. Получить подробности заявки

cURL Команда:

```
curl "http://localhost/rsd/nph-genericinterface.pl/Webservice/GenericTicketConnectorREST/Ticket/1686?UserLogin=agent&Password=123"
```

Ответ:

```
{
  "Ticket": [
    {
      "Age": 777,
      "PriorityID": 3,
      "ServiceID": "",
      "Type": "Unclassified",
      "Responsible": "root@localhost",
      "StateID": 4,
```

"ResponsibleID": 1,
"ChangeBy": 2,
"EscalationTime": 0,
"Changed": "2014-06-30 19:08:14",
"OwnerID": 2,
"RealTillTimeNotUsed": 0,
"GroupID": 1,
"Owner": "agent",
"CustomerID": "RSD",
"TypeID": 1,
"Created": "2014-06-30 19:08:12",
"Priority": "3 normal",
"UntilTime": 0,
"EscalationUpdateTime": 0,
"QueueID": 2,
"Queue": "Raw",
"State": "open",
"Title": "REST Create Test",
"CreateBy": 2,
"TicketID": 1686,
"StateType": "open",
"EscalationResponseTime": 0,
"UnlockTimeout": 0,
"EscalationSolutionTime": 0,
"LockID": 1,
"TicketNumber": "1001936",
"ArchiveFlag": "n",

```
"Lock": "unlock",  
"CreateTimeUnix": 1404173292,  
"SLAID": "",  
"CustomerUserID": "customer"  
}  
]  
}
```

4.3. Обновить заявку

cURL Команда:

```
curl "http://localhost/rsd/nph-genericinterface.pl/Webservice/GenericTicketConnectorREST/Ticket/1686?UserLogin=agent&Password=123" -H "Content-Type: application/json" -d '{"Ticket": {"Queues": {"Postmaster": ""}}}' -X PATCH
```

Ответ:

```
{  
  "TicketNumber": "1001936", "TicketID": "1686"  
}
```

4.4. Поиск заявок

cURL Команда:

```
curl "http://localhost/rsd/nph-genericinterface.pl/Webservice/GenericTicketConnectorREST/Ticket?UserLogin=agent&Password=123&Queue=Postmaster"
```

Ответ:

```
{  
  "TicketID": [  
    "1686",  
    "102",  
    "100",  
    "1"  
  ]  
}
```

4.5. Получить сведения об истории заявок

cURL Команда:

```
curl "http://localhost/rsd/nph-genericinterface.pl/Webservice/GenericTicketConnectorREST/TicketHistory/1686?UserLogin=agent&Password=123"
```

Ответ:

```
{  
  "TicketHistory": [  
    {  
      "History": [  
        {  
          "CreateBy":1, "HistoryType":"NewTicket", "TicketID":"1",
```

```

"CreateTime":"2017-06-08 22:44:48", "TypeID":1,

"OwnerID":1,

"QueueID":1,

"Name":"New Ticket [2015071510123456] created.", "StateID":1,

"PriorityID":3, "ArticleID":"1", "HistoryTypeID":1

}

],

"TicketID":"1"

}

]

}

```

12. Демон RSD (The RSD Daemon)

Демон RSD — это независимый набор системных процессов, которые планируют и выполняют задачи в фоновом режиме, либо на регулярной основе, либо по событиям. RSD Daemon необходим для правильной работы системы.

В предыдущих версиях RSD (с 3.1 по 4) существовал другой процесс, называемый RSD Scheduler, который выполнял лишь часть работы, выполняемой демоном RSD в RSD 5. Этот старый процесс заменен демоном RSD, который был переписан с нуля, чтобы сделать его более стабильным, масштабируемым и надежным, чем его предшественник.

Демон RSD способен обрабатывать до 10 задач одновременно и может работать совместно с другими демонами RSD на разных внешних серверах в кластерной среде.

В режиме ожидания RSD Daemon состоит из шести процессов:

- Главный демон (bin/radiant.Daemon.pl)

Этот процесс отвечает за запуск и продолжение работы других дочерних демонов.

- Демон рабочего задания (Kernel/System/Daemon/DaemonModules/SchedulerTaskWorker.pm)

Этот демон выполняет все задачи, которые есть в списке, по принципу "первый вход - первый выход". Он может обрабатывать одновременные задачи, создавая собственные дочерние процессы, и проверять список задач несколько раз в секунду. Список задач может быть заполнен демонами диспетчера задач, обработчиками событий и другими частями системы. Его основная миссия - как можно быстрее справиться со всеми задачами в списке.

- Демон диспетчера будущих задач (Kernel/System/Daemon/DaemonModules/Scheduler-FutureTaskManager.pm)

Этот демон проверяет наличие неповторяющихся задач, (например, когда инициатор универсального интерфейса пытается связаться с сервером, но не может, задача может быть настроена на выполнение в ближайшие 5 минут). В нужное время он посылает ему демона Task Worker.

- Демон диспетчера задач Cron (Kernel/System/Daemon/DaemonModules/Scheduler-CronTaskManager.pm)

Этот демон вычисляет время следующего выполнения всех повторяющихся задач (например, очистка кеша один раз в неделю). Такого рода задачи указаны в файле SysConfig. В нужное время для каждой задачи он отправляет необходимую информацию рабочему демону задач (Task Worker) для их выполнения.

Примечание

Если определение времени выполнения задачи изменено в SysConfig, демону может потребоваться до часа, чтобы автоматически принять это изменение. В качестве альтернативы демон RSD можно перезапустить, чтобы немедленно применить изменения.

- Демон диспетчера задач универсального агента (Kernel/System/Daemon/DaemonModules/SchedulerGenericAgentTaskManager.pm)

Этот демон ищет задания универсального агента, хранящиеся в базе данных, для которых есть расписание по времени (отбрасывая все остальные задания универсального агента, для которых установлено значение "выполняется по событиям"). Когда приходит время запустить задание универсального агента, он отправляет информацию о задании демону рабочего задания (Task Worker) для обработки задания.

- Демон диспетчера синхронизации конфигурации системы (Kernel/System/Daemon/DaemonModules/SystemConfigurationSyncManager.pm)

Этот демон проверяет, что файл конфигурации Kernel/Config/Files/ZZZAAuto.pm соответствует информации о развертывании из базы данных, в противном случае файл обновляется, и подает сигнал главному демону перезапустить все модули демона с обновленной конфигурацией.

Кроме того, если какой-либо другой файл .pm в Kernel/Config/Files/ будет изменен во время работы даэмон, он также подаст сигнал главному демону перезапустить все модули демона.

Примечание

Количество активных процессов зависит от количества задач, которые Демон RSD выполняет одновременно за определенный промежуток времени.

По умолчанию каждый демон записывает все сообщения об ошибках в отдельный файл, расположенный в /opt/radiant/var/log/Daemon/*.log. Эти журналы хранятся в системе в течение определенного периода. Чтобы изменить это поведение и/или также

регистрировать сообщения об ошибках, не являющиеся таковыми, пожалуйста, обновите настройки SysConfig в Daemon -> Core::Log.

Когда задача по какой-либо причине не может быть успешно выполнена, предопределенному получателю отправляется электронное письмо с сообщением о проблеме. Содержание электронного письма включает сообщения об ошибках и трассировку (если таковая имеется).

Демон RSD - это автоматизированный процесс, который обычно не требует вмешательства человека. Однако можно запросить его статус и при необходимости запустить или остановить его.

Чтобы быть уверенным, что демон RSD запущен, существует задание Cron, которое постоянно проверяет, что процесс активен. Главный демон готов работать даже без подключения к базе данных, поэтому совершенно безопасен, если задача Cron для его запуска выполняется еще до процесса базы данных при запуске системы, а также он устойчив к разрывам соединения с базой данных.

Если по какой-либо причине демон RSD необходимо остановить (например, во время обслуживания системы), все необработанные задачи сохраняются, и как только процесс запускается снова, он продолжается со всеми ожидающими задачами. Для повторяющихся задач он выполнит только последний экземпляр задачи (если ее время выполнения было во время режима ожидания).

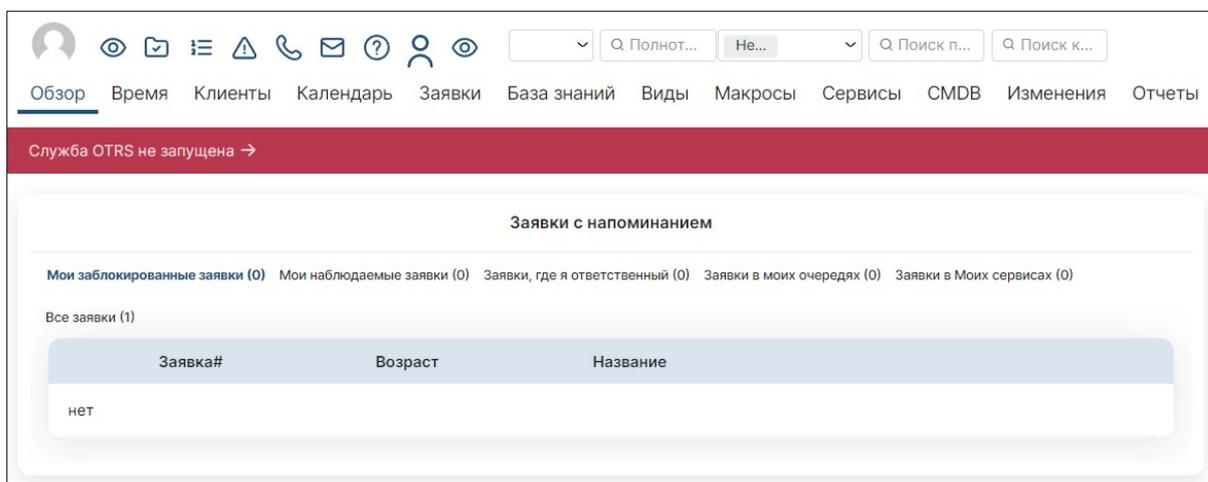
12.1. Графический интерфейс демона RSD

Демон RSD не отображается в графическом пользовательском интерфейсе RSD до тех пор, пока он не прекратит работу.

Когда система обнаруживает, что демон RSD не запущен, определенной группе пользователей (по умолчанию "admin") отправляется уведомление.

Чтобы отключить уведомление (не рекомендуется), изменить или добавить группы уведомлений, пожалуйста, отредактируйте параметр Frontend::NotifyModule###800-Daemon-Check в SysConfig.

Рисунок 4.129. Уведомление демона



При нажатии на уведомление система отображает окно наложения, объясняющее шаги по запуску демона RSD.

Рисунок 4.130. Запустить демон



12.2. Интерфейс командной строки демона RSD

Инструменты командной строки RSD Daemon позволяют управлять основными процессами демона (Запуск / остановка) или запрашивать его статус. Существуют также инструменты для получения более подробной информации о других четырех дочерних демонах.

1. Основные инструменты Демона

Для запуска, остановки или запроса статуса демона `bin/radiant.Daemon.pl` используется скрипт.

Пример 4.30. Пример запуска демона RSD

```
shell> cd /opt/radiant/  
  
shell> RSD_HOME/bin/radiant.Daemon.pl start
```

Доступные опции

- **start** - для запуска процесса демона RSD.
- **stop** - для остановки процесса демона RSD.
- **status** - для запроса статуса процесса демона RSD.
- **start --debug** - для запуска процесса демона RSD в режиме отладки

В этом режиме каждый демон сообщает разные сообщения в зависимости от выполненных действий. Этот режим не рекомендуется для производственных сред.

- **stop --force** - для остановки процесса демона RSD, сокращающего время ожидания завершения дочерних процессов.

Принудительная остановка сокращает время ожидания главного демона для успешной остановки других дочерних процессов с 30 секунд (обычно) до 5 секунд (принудительно).

2. Другие инструменты Демона

Чтобы перечислить все настроенные дочерние демоны, которые главный демон должен запустить и продолжить работать, используйте консольную команду: `Maint::Daemon::List`.

Пример 4.31. Пример для перечисления всех настроенных демонов

```
shell> cd /opt/radiant/  
  
shell> bin/radiant.Console.pl Maint::Daemon::List
```

Чтобы перечислить подробную информацию обо всех демонах, используйте консольную команду: `Maint::Daemon::Summary`.

Пример 4.32. Пример для отображения сводки всех задач демона

```
shell> cd /opt/radiant/  
  
shell> bin/radiant.Console.pl Maint::Daemon::Summary
```

5. Настройка

1. Списки Управления Доступом(ACLs)

1.1. Введение

Начиная с RSD 2.0, списки управления доступом (ACLs) можно использовать для управления доступом к заявкам, модулям, очередям и пр., или для влияния на действия с заявками (закрытие, перемещение и т. д.) в определенных ситуациях. ACL можно использовать для дополнения существующей системы разрешений ролей и групп. С использованием ACL могут быть построены элементарные workflows, основанные на атрибутах заявки.

В основном, ACL используются для ограничения возможных действий над заявками, основанные на заданном наборе правил.

ACL могут быть непосредственно вставлены в файл `Kernel/Config.pm` file, однако, теперь, мы не рекомендуем этот способ, так как в RSD теперь существует специальный раздел Списки управления доступом ACL в панели администратора, позволяющий сохранять ACL в базе данных на первом шаге создания и, затем, синхронизировать их в файл, по готовности.

Этот раздел содержит ряд примеров ACL, который показывает процесс задания параметров ACL и краткое справочное руководство по наиболее важным параметрам ACL.

Предупреждение

К пользователю `'root@localhost'` по умолчанию ACL не применяется

1.2 Описание

Текст ACL состоит из двух больших разделов: «Matching/Настройки условий» и «Change/Изменить настройки». В разделе «Настройки условий» указываются атрибуты которые проверяются на соответствие заданным образцам и при несовпадении ACL не применяется. Раздел «Изменить настройки» содержит правила, ограничивающие возможные опции заявки при выполнении заданных условий.

Разделы Условий

- Properties

Этот раздел содержит параметры сопоставления, которые могут меняться "на лету". Например, во время создания заявки данные заявки динамически изменяются по мере того, как агент устанавливает информацию. Если ACL содержит проверку значения атрибута заявки, то только при совпадении введенного/выбранного значения с образцом в ACL он может быть активирован и повлиять на значения других атрибутов заявки, но как только будет выбрано другое значение, ACL не окажет никакого влияния.

- PropertiesDatabase

Этот раздел похож на «Properties», но не принимает изменений в атрибутах заявки, которые не сохранены в базе данных, это означает, что изменение атрибута без ввода заявки в базу не будут иметь никакого эффекта. Этот раздел не используется для экранов создания заявок (поскольку заявки еще не созданы в базе данных). То есть раздел оперирует с данными ранее созданной заявки.

Разделы Действий

- Possible

Раздел «Possible» сужает список доступных действий или значений до указанных в этом разделе.

- PossibleAdd

Элементы в разделе «PossibleAdd» добавляют недостающие элементы, которые были ограничены другими ACL.«PossibleAdd» используется только вместе с другими списками ACL, имеющими разделы Possible или PossibleNot.

- PossibleNot

Этот раздел используется для удаления определенных элементов/значений из списка доступных на текущий момент. Он может использоваться самостоятельно или вместе с другими списками ACL с разделами Possible или PossibleAdd.

Чтобы сделать разработку ACL проще и мощнее, существует набор так называемых модификаторов для атрибутов в каждом разделе. Эти модификаторы объяснены ниже:

Модификаторы

- [Not]

Этот модификатор используется для исключения значения, например: '[Not]2 low' в данном случае речь идет о приоритетах заявки и будет иметь тот же результат, что и перечисление: '1 very low', '3 normal', '4 high', '5 very high'.

- [RegExp]

Этот модификатор служит для определения регулярного выражения для проверки соответствия нескольким значениям, например, '[RegExp]low' для приоритетов будет аналогично списку '1 very low', '2 low'.

- [regex]

Аналогичен по действию с [RegExp], но нечувствителен к регистру.

- [NotRegExp]

Отрицательные регулярные выражения, например, '[NotRegExp]low' для списка приоритетов аналогично списку из '3 normal', '4 high', '5 very high'.

- [Notregex]

Аналогичен по действию с [NotRegExp], но нечувствителен к регистру.

1.3 Примеры

Следующие примеры показывают оба способа представления ACL - в графическом и текстовом виде.

Пример 5.1. Список управления доступом (ACL), который позволяет перемещать в очереди только заявки с приоритетом 5.

Этот пример показывает базовую структуру ACL. Во-первых, необходимо указать имя. В данном случае это "100-Example-ACL". Обратите внимание, что списки ACL будут отсортированы по номерам перед выполнением, поэтому вам следует использовать имена с осторожностью.

Затем, следует секция "Properties", которая отфильтровывает ваши заявки. Все определенные здесь критерии будут применяться к заявке, чтобы определить, должен ли применяться ACL или нет. В нашем примере, заявка проверяется на принадлежность к очереди "Raw" и наличии приоритета "5 very high". Это также зависит от изменений в форме ввода (т.е. если заявка находится в очереди "Raw" и имеет приоритет "3 normal", то в этот момент соответствия нет, но, затем, при открытии формы «Приоритет» и выборе из выпадающего списка значения "5 very high", соответствие наступит).

Наконец, раздел "Possible" определяет изменения экрана. В нашем случае из доступных очередей на экране заявка можно выбрать только очередь "Alert".

Рисунок 5.1. ACL 100-Example-ACL

▼ Изменить структуру ACL

Настройки условий

- ▼ Properties
 - ▼ Ticket
 - Queue:
 - 2-я линия x Exact match [input] ⊕
 - Priority:
 - 5 very high x Exact match [input] ⊕
 - ⊕ [input]
 - ⊕ [input]
 - [input]

Изменить настройки

- ▼ Possible
 - ▼ Ticket
 - Queue:
 - Alert x Exact match [input] ⊕
 - ⊕ [input]
 - ⊕ [input]
 - [input]

```
# ticket acl
```

```
$Self->{TicketAcl}->{'100-Example-ACL'} = { # match properties
```

```
Properties => {
```

```
# current ticket match properties Ticket => {
```

```
Queue => ['2-я линия'],
```

```
Priority => ['5 very high'],
```

```
}
```

```
},
```

```
# return possible options (white list) Possible => {  
  
# possible ticket options (white list) Ticket => {  
  
Queue => ['Alert'],  
  
},  
  
},  
  
};
```

Пример 5.2. Список управления доступом (ACL), который позволяет перемещать в другие очереди только заявки с приоритетом 5, уже имеющиеся в базе данных.

Этот пример очень похож на предыдущий, но в этом случае только заявки, уже имеющиеся в базе данных, в очереди "Raw" и с приоритетом "5 very high" будут соответствовать критерию. Такой тип ACL не учитывает изменения в формах редактирования заявки, до того, как заявка будет обновлена в базе данных.

Рисунок 5.2. ACL 102-Example-ACL

▼ Изменить структуру ACL

Настройки условий

- ▼ (−) PropertiesDatabase
 - ▼ (−) Ticket
 - (−) Queue:
 - Raw x Exact match ▾
 - (−) Priority:
 - 5 very high x Exact match ▾
 -
 -
 -

Изменить настройки

- ▼ (−) Possible
 - ▼ (−) Ticket
 - (−) Queue:
 - Alert x Exact match ▾
 -
 -
 -

```
# ticket acl
$Self->{TicketAcI}->{'102-Example-ACL'} = { # match properties
PropertiesDatabase => {
# current ticket match properties Ticket => {
Queue => ['Raw'],
Priority => ['5 very high'],
}
},
```

```
# return possible options (white list) Possible => {  
  
# possible ticket options (white list) Ticket => {  
  
Queue => ['Alert'],  
  
},  
  
},  
  
};
```

Пример 5.3. Список управления доступом (ACL), отключающий закрытие заявок в очереди «Raw» и скрывающий кнопку «Закреть».

Здесь вы можете увидеть, как поле заявки (состояние) может быть отфильтровано с более чем одним возможным значением для выбора. Можно также ограничить возможные действия, выполняемые над конкретной заявкой. В этом случае, заявка не может быть закрыта.

Рисунок 5.3. ACL 102-Second-Example-ACL

▼ Изменить структуру ACL

Настройки условий

- ▼ (-) Properties
 - ▼ (-) Ticket
 - Queue:
 - Raw x Exact match ▾
 - Priority:
 - 5 very high x Exact match ▾
 -
 -
 -

Изменить настройки

- ▼ (-) Possible
 - ▼ (-) Ticket
 - State:
 - new x open x pending reminder x
 - Exact match ▾
 -
 -
 - ▼ (-) PossibleNot
 - ▼ (-) Action
 - AgentTicketClose x
 -
 -

```

$Self->{TicketAcl}->{'102-Second-Example-ACL'} = { # match properties
Properties => {
# current ticket match properties Ticket => {
Queue => ['Raw'],
}
}

```

```
},  
  
# return possible options (white list) Possible => {  
  
# possible ticket options (white list) Ticket => {  
  
State => ['new', 'open', 'pending reminder'],  
  
},  
  
},  
  
# return also not possible options (black list) PossibleNot => {  
  
# not possible action options Action => [ 'AgentTicketClose' ],  
  
},  
  
};
```

Пример 5.4. Список управления доступом (ACL), который удаляет навсегда возможность установить состояние "closed successful".

Этот пример показывает, как можно определить отрицательные фильтры (будет удалено состояние «закрыто успешно»). Вы также можете видеть, что не заданные условия для «Properties» для заявки будет означать, что оно верно для всех заявок, т.е. ACL применяется всегда. Это может быть полезно, если вы хотите скрыть определенные значения по умолчанию и включить их только в особых случаях (например, если агент находится в определенной группе).

Рисунок 5.4. ACL 103-Third-ACL-Example

▼ Изменить структуру ACL

Настройки условий

Изменить настройки

- ▼ Possible
 - ▼ Ticket
 - State:
 -
 -
 -
 -

```

$Self->{TicketAcl}->{'103-Third-ACL-Example'} = { # match properties

Properties => {

# current ticket match properties (match always)

},

# return possible options PossibleNot => {

# possible ticket options Ticket => {

State => ['closed successful'],

},

},

};

```

Пример 5.5. Список управления доступом (ACL), который отображает только Hardware-сервисы для заявок, которые были созданы в очередях, начинающихся с "HW".

Этот пример также показывает, как можно использовать регулярные выражения для сопоставления заявок и фильтрации доступных опций.

Рисунок 5.5. ACL 104-Only-Hardware-Services-for-HW-Queues-ACL

▼ Изменить структуру ACL

Настройки условий

- ▼ (-) Properties
 - ▼ (-) Ticket
 - Queue:
 - [RegExp]^HW x Regular expression v
 -
 -
 -

Изменить настройки

- ▼ (-) Possible
 - ▼ (-) Ticket
 - Queue:
 - [RegExp]Hardware x Regular expression v
 -
 -
 -

```
$Self->{TicketAcl}->{'104-Only-Hardware-Services-for-HW-Queues-ACL'} = { # match
properties
```

```
# note we don't have "Ticket => {" because there's no ticket yet Properties => {
```

```
Queue => {
```

```
Name => [['RegExp]^HW'],
```

```
}
```

```
},
```

```
# return possible options Possible => {
```

```
# possible ticket options Ticket => {
```

```
Service => [['RegExp]Hardware'],
```

```
},  
},  
};
```

Пример 5.6. Список управления доступом (ACL) для ограничения доступа к Процессу в интерфейсе клиента по CustomerID.

Рисунок 5.6. ACL 105-Disallow-Process-For-CustomerID

The screenshot shows a web interface for configuring an ACL. It is divided into two main sections: 'Настройки условий' (Condition Settings) and 'Изменить настройки' (Change Settings).
Under 'Настройки условий', there is a tree view: 'Properties' (expanded) -> 'CustomerUser' (expanded) -> 'UserCustomerID:' (expanded). Below this, there is a list of conditions. The first condition is 'TheCustomerID' with a value of 'Exact match'. There are also empty input fields and dropdown menus for adding more conditions.
Under 'Изменить настройки', there is a tree view: 'Possible' (expanded) -> 'Process' (expanded). Below this, there is a list of settings. The first setting is 'P14' with a value of 'Exact match'. There are also empty input fields and dropdown menus for adding more settings.

```
$Self->{TicketAcl}->{"105-Disallow-Process-For-CustomerID"} = { 'Possible' => {},  
'PossibleNot' => { 'Process' => [  
'P14'  
]
```

```

},
'Properties' => { 'CustomerUser' => {
'UserCustomerID' => [ 'CustomerID'
]
}
},
'PropertiesDatabase' => {}, 'StopAfterMatch' => 0
};

```

1.4. Ссылка

В приведенном ниже примере приведен список всех параметров, которые можно использовать для ACL.

Подробное описание того, как использовать ACL для обработки заявок, см. в разделе о ACL в документации ProcessManagement.

Пример 5.7. Ссылка отображает все возможные важные настройки ACL

```

# ticket acl

$Self->{TicketAcl}->{'200-ACL-Reference'} = {

# match properties (current values from the form)

Properties => {

# the used frontend module

Frontend => {

Action => ['AgentTicketPhone', 'AgentTicketEmail'],

},

# the logged in agent

User => {

UserLogin => ['some login'],

Group_rw => [

```

```

'hotline',
],
Role => [
'admin',
],
# ...
},

# the logged in customer
CustomerUser => {
  UserLogin => ['some login'],
  UserCustomerID => ['some customer id'],
  Group_rw => [
    'hotline',
  ],
  # ...
},

# process properties
Process => {
  ProcessEntityID => ['Process-
9c378d7cc59f0fce4cee7bb9995ee3eb'],
  # the Process that the current ticket is part of
  ActivityEntityID => ['Activity-
f8b2fdebe54eeb7b147a5f8e1da5e35c'],
  # the current Activity of the ticket
  ActivityDialogEntityID => ['ActivityDialog-aff0ae05fe6803f38de8fff6cf33b7ce'],
  # the current ActivityDialog that the Agent/Customer is using

```

```
},

# ticket properties

Queue => {
  Name      => ['Raw'],
  QueueID   => ['some id'],
  GroupID   => ['some id'],
  Email     => ['some email'],
  RealName  => ['RSD System'],
  # ...
},

Service => {
  ServiceID => ['some id'],
  Name      => ['some name'],
  ParentID  => ['some id'],
  # ...
},

Type => {
  ID       => ['some id'],
  Name    => ['some name'],
  # ...
},

Priority = {
  ID       => ['some id'],
  Name    => ['some name'],
  # ...
},
```

```
SLA = {  
  
  SLAID          => ['some id'],  
  
  Name           => ['some name'],  
  
  Calendar => ['some calendar'],  
  
  # ...  
  
},  
  
State = {  
  
  ID             => ['some id'],  
  
  Name           => ['some name'],  
  
  TypeName => ['some state type name'],  
  
  TypeID        => ['some state type id'],  
  
  # ...  
  
},  
  
Owner => {  
  
  UserLogin => ['some login'],  
  
  Group_rw => [  
  
    'some group',  
  
  ],  
  
  Role => [  
  
    'admin',  
  
  ],  
  
  # ...  
  
},  
  
Responsible => {  
  
  UserLogin => ['some login'],  
  
  Group_rw => [  
  
    'some group',
```

```

],
Role => [
'admin',
],
# ...
},
DynamicField => {
# Names must be in DynamicField_<field_name> format.
# Values in [ ... ] must always be the untranslated internal data keys #
#   specified in the dynamic field definition and
#   not the data values shown to the user.
DynamicField_Field1          => ['some value'],
DynamicField_OtherField      => ['some value'],
DynamicField_TicketFreeText2 => ['some value'], # ...
},
# alternatively, ticket properties can be specified in the ticket hash
Ticket => {
Queue                        => ['Raw'],
State                        => ['new', 'open'],
Priority                      => ['some priority'],
Lock                         => ['lock'],
CustomerID                   => ['some id'],
CustomerUserID => ['some id'],
Owner                        => ['some owner'],
DynamicField_Field1 => ['some value'],
DynamicField_MyField => ['some value'], # ...
},

```

```

},

# match properties (existing values from the database)
PropertiesDatabase => {
# See section "Properties", the same config can be used here. # ...
}

# reset possible options (white list)
Possible => {
# possible ticket options (white list)
Ticket => {
Queue => ['Hotline', 'Coordination'],
State => ['some state'],
Priority => ['5 very high'],
DynamicField_Field1 => ['some value'],
DynamicField_MyField => ['some value'], # ...
NewOwner => ['some owner'],
OldOwner => ['some owner'], # ...
},

# Limit the number of possible ActivityDialogs the Agent/Customer
# can use in a process ticket.
ActivityDialog => ['AD1', 'AD3'],

# Limit the number of possible Processes that can be started
Process => ['Process-9c378d7cc59f0fce4cee7bb9995ee3eb',
'Process-12345678901234567890123456789012'],

```

```

# possible action options (white list)

Action => [

'AgentTicketBounce',

'AgentTicketPhone'.           # only used to show/hide the Split action '

AgentLinkObject',           # only used to show/hide the Link action # ...

],

},

# add options (white list)

PossibleAdd => {

# See section "Possible" # ...

},

# remove options (black list)

PossibleNot => {

# See section "Possible" # ...

},

};

```

Примечание

При сопоставлении списков ACL, если отправлен параметр CustomerUserID, механизм ACL сравнивает определенные списки ACL, используя предоставленный CustomerUserID, чтобы собрать сведения о пользователе-клиенте для заполнения хэша CustomerUser, а также переопределяет информацию о покупателе в хэше заявки для проверки в Properties. С другой стороны, эти вычисления также выполняются для части PropertiesDatabase, но с использованием Ticket Customer в качестве основы для сбора данных.

Обратите внимание, что в клиентском интерфейсе CustomerUserID всегда отправляется вместе с текущим зарегистрированным пользователем клиента.

Помните, что на экранах поиска (AgentTicketSearch и CustomerTicketSearch)

- единственно доступные атрибуты для ACL это динамические поля. Это означает, что для этих экранов вы не можете ограничить любые другие атрибуты, такие как Тип заявки, Состояние, Очередь и т.д.

Начиная с RSD 4, параметр «Action» теперь является не хэш-функцией, а ссылкой на массив и может быть использован в любом из разделов «Изменение» с использованием любого из модификаторов.

2. Управление Процессами

1. Введение

Эта функция RSD позволяет моделировать процессы (workflow) в системе управления заявками. Основная идея заключается в том, чтобы задать повторяющиеся процессы и делегировать части работы различным сотрудникам, а также управлять продолжением работы по процессу, в зависимости от определенных условий.

2. Пример процесса

Давайте рассмотрим пример, чтобы все было более наглядно и понятно. Мы будем рассматривать процесс покупки книги:

2.1. Запись спроса

Перед размещением заказа будет зафиксирована потребность в литературе со стороны сотрудника. В нашем примере необходима следующая книга:

Title: Prozessmanagement für Dummies Autor: Thilo Knuppertz

ISBN: 3527703713

2.2. Утверждение менеджером

Начальнику отдела сотрудника необходимо определиться с приказом. В случае отказа, причина должна быть записана менеджером. В случае одобрения заказ передается в отдел закупок.

2.3. Обрабатывается отделом закупок

Перед отделом закупок стоит задача выяснить, где книгу можно заказать с лучшими условиями. Если ее нет в наличии, это можно указать в заказе. В случае успешного заказа закупка фиксирует поставщика, цену и дату доставки

2.4. Обрабатывается почтовым отделом

Посылка придет в компанию. Приемный отдел проверяет посылку и фиксирует дату получения. Теперь сотрудник будет проинформирован о том, что его заказ прибыл и готов к выдаче.

3. Реализация примера

Если мы предположим, что заявка действует в этом рабочем процессе как сопроводительный документ, который может получать примечания об изменениях, у нас уже есть четкая картина заявка процесса.

Проанализировав предыдущий пример, мы можем выделить следующие пункты:

- Возможности регистрации данных, назовем их Activity Dialogs/Диалоги Активности,
- Проверки, которые могут автоматически реагировать на изменение данных, назовем *Переходы*,
- Изменения, которые могут быть применены для обработки заявки после успешного перехода обработки заявки, назовем их *Transition Actions/Действия перехода*.

Также нам нужен дополнительный элемент, который может быть не таким очевидным:

- Возможность предлагать более одного диалогового окна действий. В нашем примере это необходимо, когда у менеджера должен быть выбор между «Утвердить» и «Отклонить». Назовем это *Activity/Активность*.

Теперь, введя понятия Activities/Активность, Activity Dialogs/Диалоги Активности, Transitions/Переходы и Transition Actions/Действия перехода, мы имеем все необходимые инструменты для построения отдельных этапов нашего примера. Что мы еще не определили, так это область, в которой для каждого потока работ/ workflow был бы задан порядок шагов. Назовем это Process/Процесс. Чтобы иметь возможность ссылаться на все эти сущности/понятия позже, дадим им краткие имена/аббревиатуры в скобках. Эти обозначения основаны на внутреннем механизме идентификации, называемом EntityIDs.

EntityIDs согласовываются с одной или двумя буквами (в зависимости от части процесса), а затем порядковый номер, например:

- Процесс: 'P1', 'P2' ... 'Pn'.
- Деятельность: 'A1', 'A2' ... 'An'.
- Диалог Активности: 'AD1', 'AD2' ... 'ADn'.
- Переход: 'T1', 'T2' ... 'Tn'.
- Переход действий: 'TA1', 'TA2' ... 'TAn'.

Перед созданием процесса и его частей необходимо подготовить систему. Нужно определить некоторые Очереди, Клиентов и Динамические Поля, а также установить некоторые опции в файле SysConfig.

Создать следующие Очереди:

- Управление
- Сотрудники
- Покупка
- Почтовое отделение

Создать следующих Пользователей:

- Менеджер
- Сотрудник

Создать следующие Динамические Поля:

- Title

Метка	Title
Тип	Текст
Объект	Заявка

- Author

Метка	Author
Тип	Текст
Объект	Заявка

- ISBN

Метка	ISBN
Тип	Текст
Объект	Заявка

- Status

Метка	Status
Тип	Выпадающий список
Объект	Заявка

Возможные значения	<ul style="list-style-type: none"> • Утверждение • Отказано в утверждении • Утвержденный • Заказ отменен • Заказ доставлен • Посылка получена
--------------------	---

Примечание: Используйте именно эти возможные значения для "Ключ" и "Значение" при создании динамических полей.

- Supplier

Метка	Supplier
Тип	Текст
Объект	Заявка

- Price

Метка	Price
Тип	Текст
Объект	Заявка

- DeliveryDate

Метка	Delivery date
Тип	Дата
Объект	Заявка

- DateOfReceipt

Метка	Date Of Receipt
-------	-----------------

Тип	Дата
Объект	Заявка

Устанавливать следующие параметры SysConfig:

- 'Ticket::Responsible': Yes
- 'Ticket::Frontend::AgentTicketZoom####ProcessWidgetDynamicFieldGroups':

Ключ:	Содержание:
Book	Title, Author, ISBN
General	Status
Order	Price, Supplier, DeliveryDate
Shipment	DateOfReceipt

- 'Ticket::Frontend::AgentTicketZoom####ProcessWidgetDynamicField':

Ключ:	Содержание:
Author	1
DateOfReceipt	1
DeliveryDate	1
ISBN	1
Price	1
Status	1

Supplier	1
Title	1

Теперь давайте перейдем к описанию реального процесса. На следующем шаге мы определим конкретные объекты, которые нам необходимы.

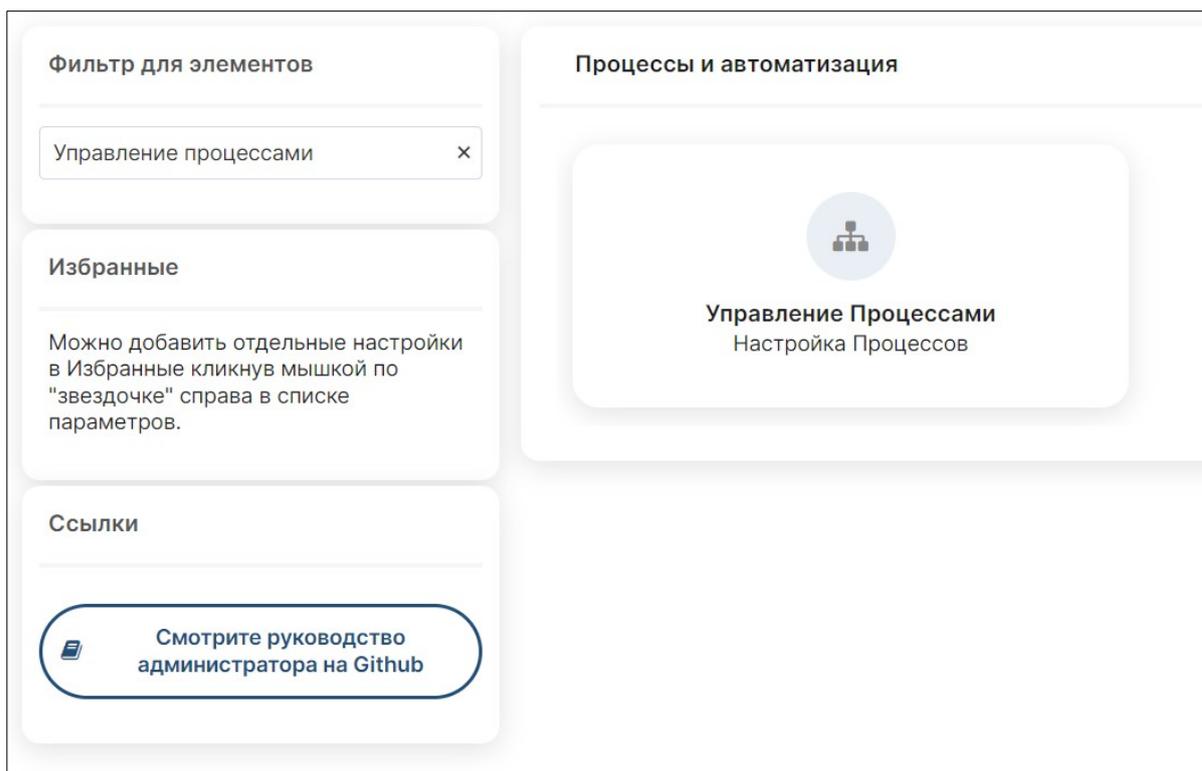
3.1. Процесс (в качестве контейнера)

Для создания нового процесса необходимо перейти к пункту Управление процессами на панели Администрирования, после нажатия на который мы попадем на экран Обзора процессов. После создания процесса мы получим возможность создания всех его объектов (всех составляющих процесса).

Примечание

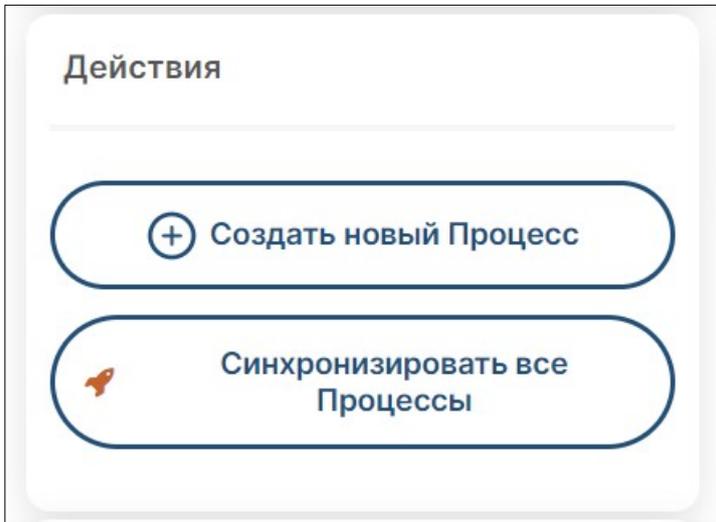
Деятельности, Диалоги Активности, Переходы и Действия переходов, определенные в одном процессе, будут доступны для всех процессов системы.

Рисунок 5.7. Страница Администратора RSD - Управление Процессами



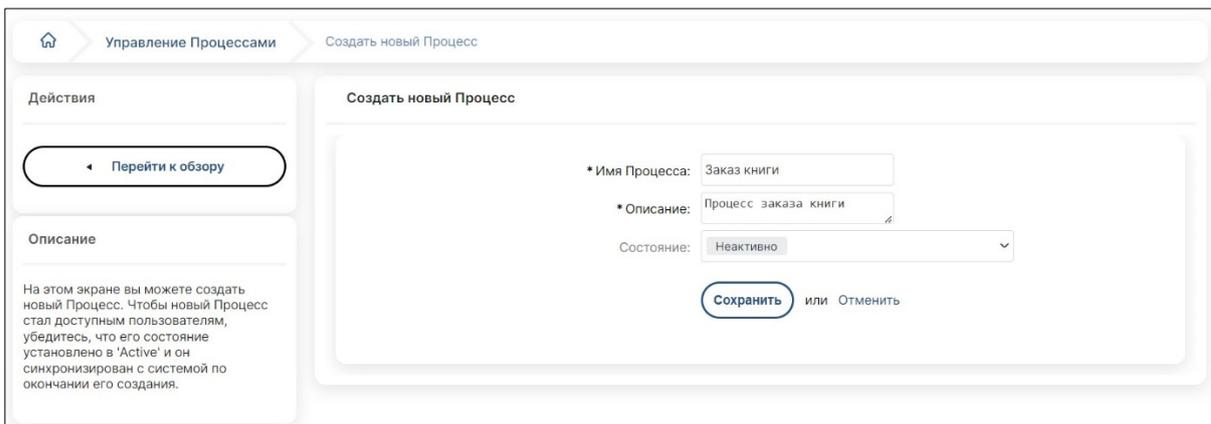
Нажмите на действие "Создать Новый Процесс" в окне Действий.

Рисунок 5.8. Кнопка «Создать новый Процесс»



Заполните информацию о процессе, установите имя процесса и описание, мы оставим состояние процесса как «неактивное», пока не завершим все задачи. Сохраните процесс.

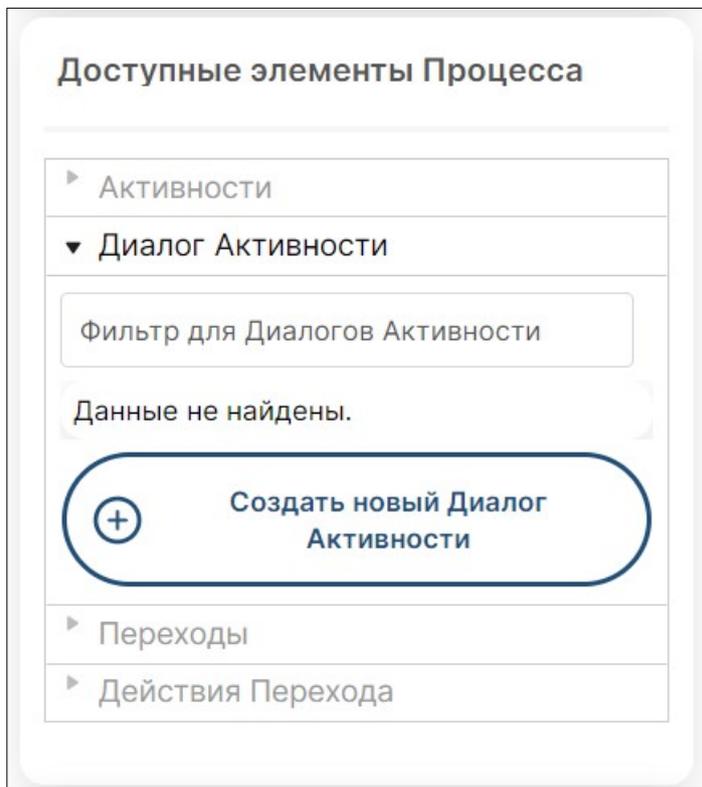
Рисунок 5.9. Добавление нового процесса



3.2. Диалоги Действий

Кликните по имени процесса в окне Обзора процессов, затем в списке "Доступные элементы процесса" нажмите на "Диалоги Активности" (это действие откроет список опций для диалогов активности и скроет другие списки, к нему не относящиеся, создав эффект гармошки), затем нажмите на "Создать новый Диалог Активности".

Рисунок 5.10. Кнопка создания нового Диалога Активности



В открывшемся всплывающем окне заполните поля «Имя Диалога Активности», а также поля «Описание (краткое)». Для этого примера мы оставим все остальные поля по умолчанию. Чтобы назначить поля диалогу действий, просто перетащите необходимое поле из списка «Доступные поля» в список «Назначенные поля». Порядок в списке «Назначенные поля» — это порядок полей на экране. Чтобы изменить порядок, просто перетащите поле, чтобы переставить его в правильном месте.

Рисунок 5.11. Добавить новый Диалог Активности

▼ Диалог Активности

* Имя Диалога Активности:

Доступно в:

* Описание (краткое):

Описание (полное):

Права доступа:

Требуется блокировка:

Текст подсказки кнопки Отправить:

Название кнопки Отправить:

▼ Поля

Вы можете назначить поля для этого Диалога Активности перетаскиванием элементов мышью из левого списка в правый. Порядок элементов в списке изменяется перетаскиванием элементов (drag 'n' drop)

НАЗНАЧЕННЫЕ ПОЛЯ

ДОСТУПНЫЕ ПОЛЯ

- DynamicField_CalendarFinish
- DynamicField_CalendarStart

После того как поля помещены в список "Назначенные поля", другое всплывающее окно покажет некоторые данные о поле. Мы оставим параметры по умолчанию, и только для полей типа Сообщение мы должны убедиться, что для поля «Канал связи» установлено значение «Внутренний» и поле «Виден клиенту» не активно.

Рисунок 5.12. Редактировать элементы поля (Сообщение)

Редактировать атрибуты поля: Сообщение ✕

Описание (краткое):

Описание (полное):

Значение по умолчанию:

Канал связи:

Виден клиенту:

Затраченное время:

Отобразить:

После того, как все поля будут назначены, нажмите кнопку отправки на главном всплывающем экране, чтобы сохранить изменения.

В этом примере мы будем использовать поле «Сообщение» для комментариев, но другим вариантом может быть создание динамического поля типа TextArea/Длинный текст, остальные поля, упомянутые в строках ниже, — это динамические поля, которые мы определили ранее.

Обратите внимание, что на этом экране все динамические поля имеют префикс «DynamicField_», как и «DynamicField_Title». Не путайте с полем «Title», которое является заголовком/темой заявки.

Создайте следующие Диалоги Активности:

- «Запись запроса» (AD1)

Диалог активности, содержащий все необходимые поля для сбора данных по заказу (название, автор и ISBN), а также поле «Статус» с возможностью выбора «Утверждение».

- «Утверждение отклонено» (AD2)

Диалог активности с полем комментария (Сообщение) и полем «Статус» с возможностью выбора «Отказано в утверждении».

- «Утверждено» (AD3)

Здесь нам просто нужно поле «Статус» с опцией «Одобрено».

- «Заказ отклонен» (AD4)

Диалог активности, позволяющий при покупке отклонить невыполнимый заказ (заказать на складе). Здесь нам также понадобится поле комментария и поле «Статус» с опцией «Заказ отклонен».

- «Заказ доставлен» (AD5)

Диалог активности с полями «Поставщик», «Цена» и «Дата поставки для покупки» и поле «Статус» с опцией «Заказ доставлен».

- «Посылка получена» (AD6)

Диалог активности для почтового отделения с полем «Дата получения» и полем «Статус» с опцией «Посылка получена».

Чтобы ограничить поле «Состояние» для каждого диалогового окна активности, нам нужно добавить несколько ACL в Kernel/Config.pm или в новый файл Perl, расположенный в Kernel/Config/Files.

```
$Self->{TicketAcl}->{'P1-AD1-1'} = { Properties => {  
  
  Process => {  
  
    ActivityDialogEntityID => ['AD1'],  
  
  },  
  
  },  
  
  Possible => {  
  
    Ticket => {  
  
      DynamicField_Status => ['Approval'],  
  
    },  
  
  },  
  
};  
  
$Self->{TicketAcl}->{'P1-AD2-1'} = { Properties => {
```

```
Process => {
  ActivityDialogEntityID => ['AD2'],
},
},
Possible => {
  Ticket => {
    DynamicField_Status => ['Approval denied'],
  },
},
};

$Self->{TicketAcl}->{'P1-AD3-1'} = { Properties => {
  Process => {
    ActivityDialogEntityID => ['AD3'],
  },
},
  Possible => {
    Ticket => {
      DynamicField_Status => ['Approved'],
    },
  },
};

$Self->{TicketAcl}->{'P1-AD4-1'} = { Properties => {
  Process => {
    ActivityDialogEntityID => ['AD4'],
  },
},
};
```

```

},
Possible => {
Ticket => {
DynamicField_Status => ['Order denied'],
},
},
};

$Self->{TicketAcl}->{'P1-AD5-1'} = { Properties => {
Process => {
ActivityDialogEntityID => ['AD5'],
},
},
Possible => {
Ticket => {
DynamicField_Status => ['Order placed'],
},
},
};

$Self->{TicketAcl}->{'P1-AD6-1'} = { Properties => {
Process => {
ActivityDialogEntityID => ['AD6'],
},
},
Possible => {
Ticket => {

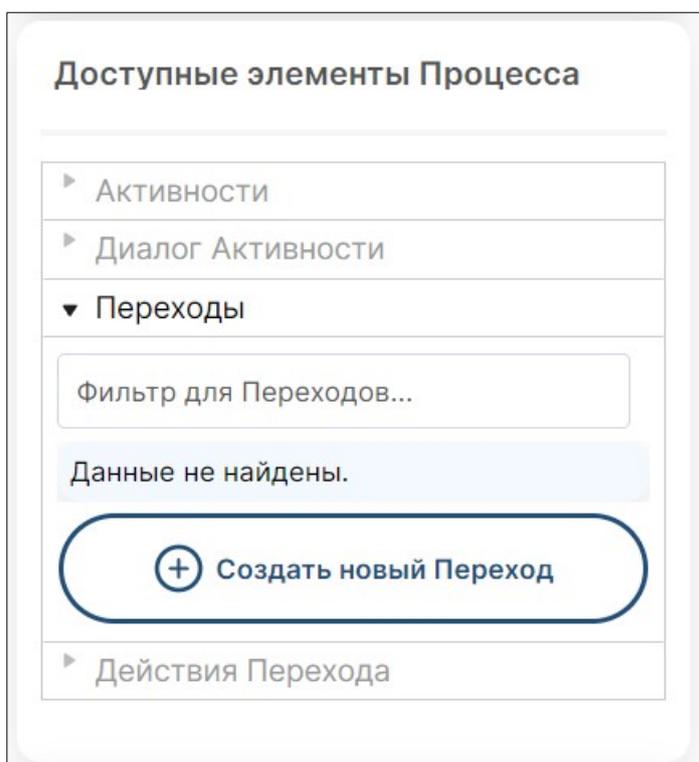
```

```
DynamicField_Status => ['Shipment received'],  
  
},  
  
},  
  
};
```

3.3. Переходы

В «Доступных элементах процесса» нажмите «Переходы», затем нажмите «Создать новый переход».

Рисунок 5.13. Кнопка «Создать новый переход»



В открывшемся всплывающем окне заполните «Имя перехода», затем, в условиях, для этого примера мы будем использовать только одно условие и только одно поле, для обоих мы можем оставить тип связи как «и», и мы будем использовать поле значение типа соответствия как "String".

Рисунок 5.14. Добавить новый Переход

Создать новый Переход
 Отменить и закрыть

▼ Переход

* Имя Перехода:

▼ Условия

Условия могут действовать только на заполненных полях.

Тип связи между Условиями:

Условие 1

Тип связи:

Поля ⊕

Название: Тип: Значение: ⊖

[Добавить новое Условие](#)

[Сохранить](#)

После того, как все условия установлены, нажмите кнопку «Отправить», чтобы сохранить изменения. Создайте следующие переходы:

- «Утверждение» (Т1)

Переход, который проверяет, установлено ли в поле «Статус» значение «Утверждение».

- «Утверждение отклонено» (Т2)

Переход, который проверяет, установлено ли в поле «Статус» значение «Отказано в утверждении».

- «Утверждено» (Т3)

Переход, который проверяет, установлено ли в поле «Статус» значение «Утверждено».

- «Заказ отклонен» (Т4)

Переход, который проверяет, установлено ли в поле «Статус» значение «Заказ отклонен».

- «Заказ размещен» (Т5)

Переход, который проверяет, установлено ли в поле «Статус» значение «Заказ размещен».

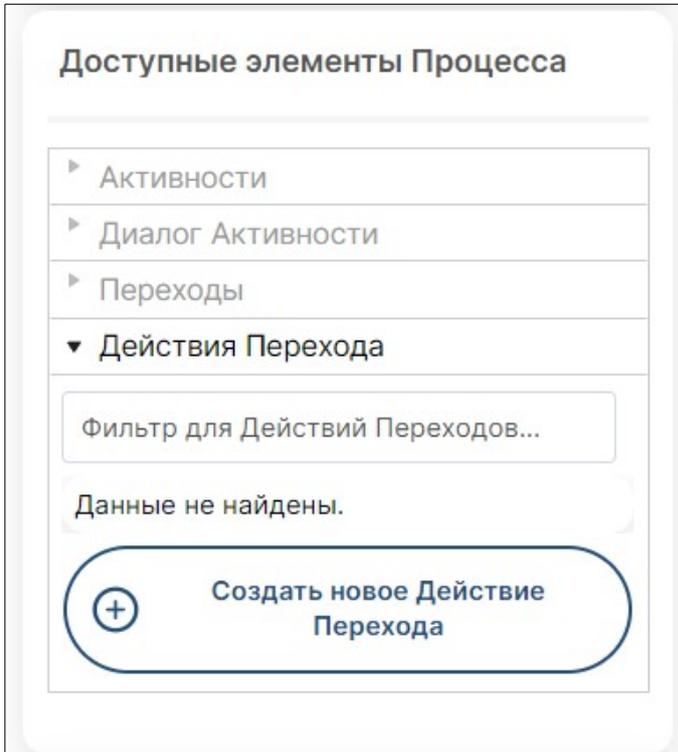
- «Посылка получена» (Т6)

Переход, который проверяет, установлено ли в поле «Статус» значение «Посылка получена».

3.4. Действия перехода

Нажмите «Действия перехода» в «Доступных элементах процесса», затем нажмите «Создать новое действие перехода».

Рисунок 5.15. Кнопка «Создать новое действие перехода»



В открывшемся всплывающем окне заполните «Имя действия перехода» и «Модуль действия перехода», затем добавьте обязательные и необязательные имена и значения параметров.

Все модули действий перехода расположены в Kernel/System/ProcessManagement/TransitionAction, и ниже приведен список объединенных действий перехода, включенных в этот выпуск:

- DynamicFieldSet
- TicketArticleCreate
- TicketCreate
- TicketCustomerSet
- TicketLockSet
- TicketOwnerSet
- TicketQueueSet
- TicketResponsibleSet
- TicketServiceSet
- TicketSLASet
- TicketStateSet

- TicketTitleSet
- TicketTypeSet

Каждый модуль имеет свои собственные и различные параметры. Пожалуйста, просмотрите документацию по модулю, чтобы узнать обо всех обязательных и необязательных параметрах.

Примечание

Начиная с RSD 4.0.1, параметры теперь необязательно имеют фиксированные значения, но, вместо этого, они могут наследовать их из оригинальной заявки используя формат: <RSD_Ticket_property>.

Начиная с версии RSD 4.0.6, поддерживается формат <RSD_TICKET_property>, старый формат тоже может использоваться, но не рекомендуется, так как в следующих версиях он будет удален.

Рисунок 5.16. Добавить новые Действия Перехода

После того, как все параметры и значения установлены, нажмите кнопку отправки, чтобы сохранить изменения. Создайте следующие действия перехода:

- «Переместить процессную заявку в очередь «Управление»» (TA1)

Предполагается, что это действие будет выполнено при применении Перехода «Утверждение» (T1).

- «Изменить ответственного за заявку на «Менеджера»» (TA2)

Выполняется при применении Перехода «Утверждение» (T1).

- «Переместить процессную заявку в очередь «Сотрудники»» (TA3).

Выполняется, когда:

- Применен переход «Отказано в утверждении» (T2)

- Применен Переход "Заказ отклонен" (Т4)
- Применен Переход "Отгрузка получена" (Т6)
- «Изменить ответственного за заявку на «Сотрудника»» (ТА4)

Выполняется, когда:

- Применен переход «Отказано в утверждении» (Т2)
- Применен Переход "Заказ отклонен" (Т4)
- Применен Переход "Отгрузка получена" (Т6)
- «Переместить процессную заявку в очередь «Покупка»» (ТА5)

Выполняется при применении перехода «Утверждено» (Т3).

- «Переместить процессную заявку в очередь «Почтовое отделение»» (ТА6)

Выполняется при применении перехода «Оформлен заказ» (Т5).

- «Закрыть заявку успешно» (ТА7)

Выполняется, когда:

- Применен переход "Посылка получена" (Т6)
- «Закрыть заявку неуспешно» (ТА8)

Выполняется, когда:

- Применен переход «Отказано в утверждении» (Т2)
- Применен переход "Заказ отклонен" (Т4)

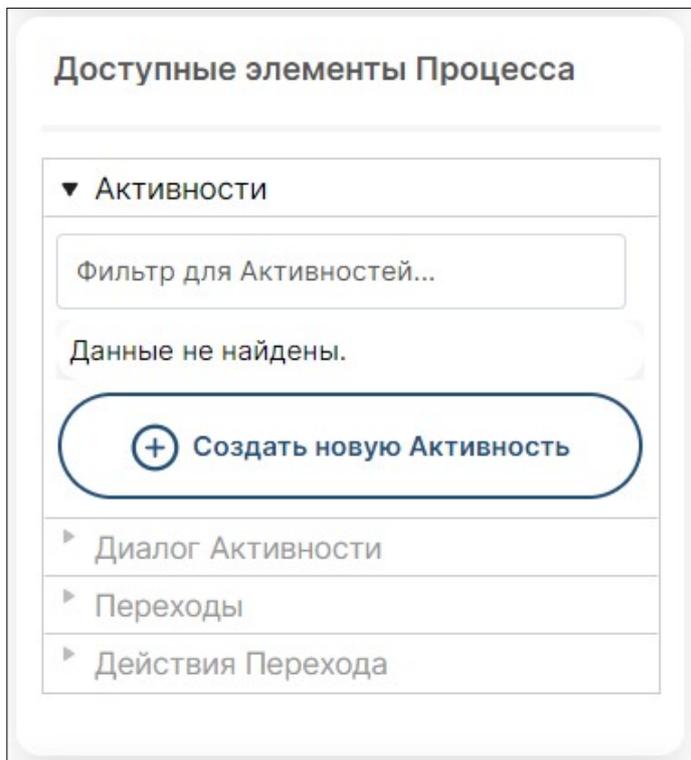
Как видите, есть места, где должны выполняться одни и те же Действия Переходов. Таким образом было бы логично сделать возможным связывать Transition Actions/Действия Переходов свободно с Transitions/Переходами, для их повторного использования.

3.5. Активности

Мы выбрали подход, при котором активности рассматриваются как корзина, которая может содержать один или несколько диалогов активности.

Нажмите «Активности» в «Доступных элементах процесса», затем нажмите «Создать новую активность».

Рисунок 5.17. Кнопка «Создать новую активность»



В открывшемся всплывающем окне заполните «Имя Активности», затем перетащите необходимые диалоги действий из списка «Доступные диалоги действий» и поместите их в список «Назначенные диалоги действий». Эти диалоги будут представлены (на экране масштабирования заявок) в том же порядке, в котором они определены на этом экране, переводя их сверху вниз и слева направо.

Этот порядок особенно важен в первом действии, так как первое диалоговое окно действия для этого действия является единственным, которое представлено при запуске процесса.

Создайте следующие активности:

- «Запись спроса» (A1)

Содержит диалог действий «Запись запроса» (AD1)

- «Утверждение» (A2)

Содержит диалоговые окна действий «Отказано в утверждении» (AD2), а также «Утверждено» (AD3).

- «Заказ» (A3)

Содержит диалоговые окна действий «Заказ отклонен» (AD4), а также «Заказ доставлен» (AD5).

- «Поступление» (A4)

Содержит диалог действий «Посылка получена» (AD6)

- «Процесс завершен» (A5): это действие без возможных диалогов действий. Он будет установлен после «Отказано в утверждении», «Отказано в заказе» или «Отгрузка получена» и представляет собой конец процесса.

Теперь мы можем видеть, что Деятельность точно определяет состояние процесса Заявки. После успешного Перехода Заявка перемещается из одного вида деятельности в другой.

3.6. Схема процесса заказа книги

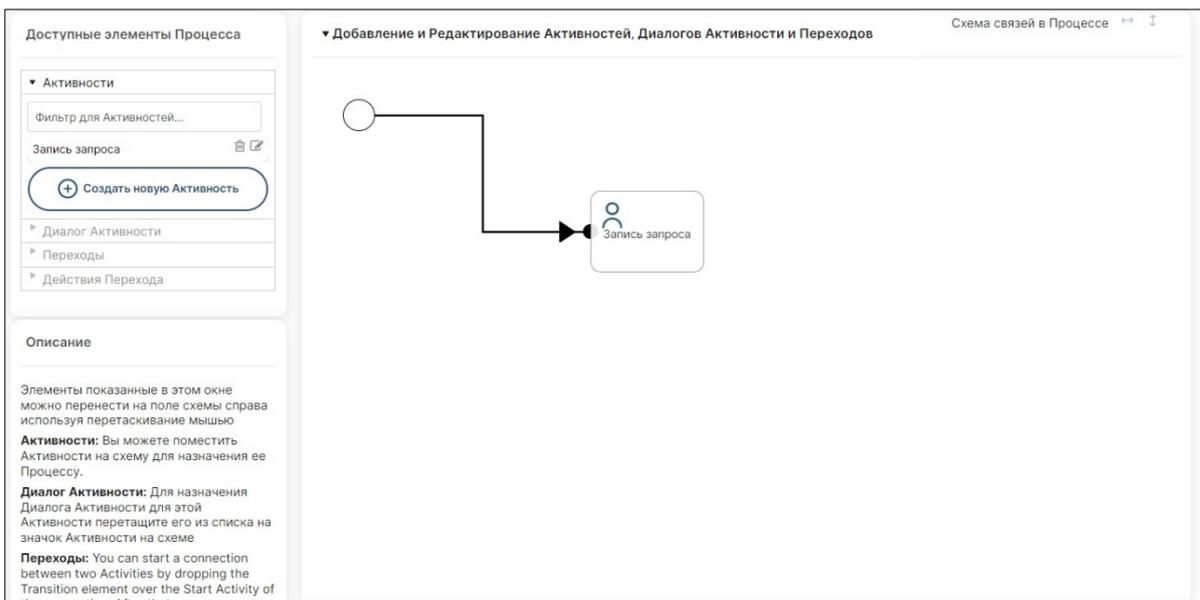
Давайте завершим наш пример последней недостающей частью головоломки - процессом как описанием потока. В нашем случае это весь рабочий процесс заказа. Другие процессы могут быть заказом канцелярских товаров или совершенно другими процессами.

Процесс имеет начальную точку, состоящую из начального действия и диалогового окна начального действия. Для любого нового заказа книги первым отображаемым экраном является диалоговое окно начала действия (первое диалоговое окно действия для первого действия). Если это будет выполнено и сохранено, будет создана заявка на обработку, и она сможет следовать настроенному рабочему процессу.

Процесс также содержит указания о том, как процессная заявка может перемещаться по процессу. Назовем это «Путь». Он состоит из начального действия, одного или нескольких переходов (возможно, с действиями перехода) и других действий.

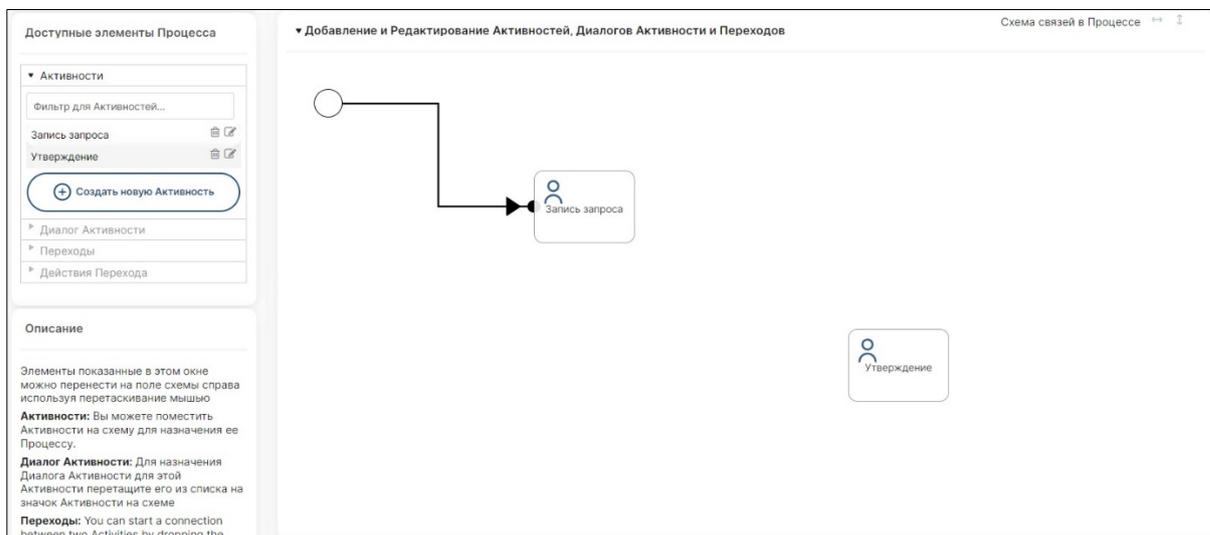
Предполагая, что действия уже назначили свои диалоги действий, перетащите действие из «гармошки» (в левой части экрана) и поместите его в область схемы (под информацией о процессе). Обратите внимание, что стрелка от начала процесса (зеленый кружок) к действию размещается автоматически. (Это первое действие, и его первое диалоговое окно действия — это первый экран, который будет отображаться при запуске процесса).

Рисунок 5.18. Перетащите первую активность на схему



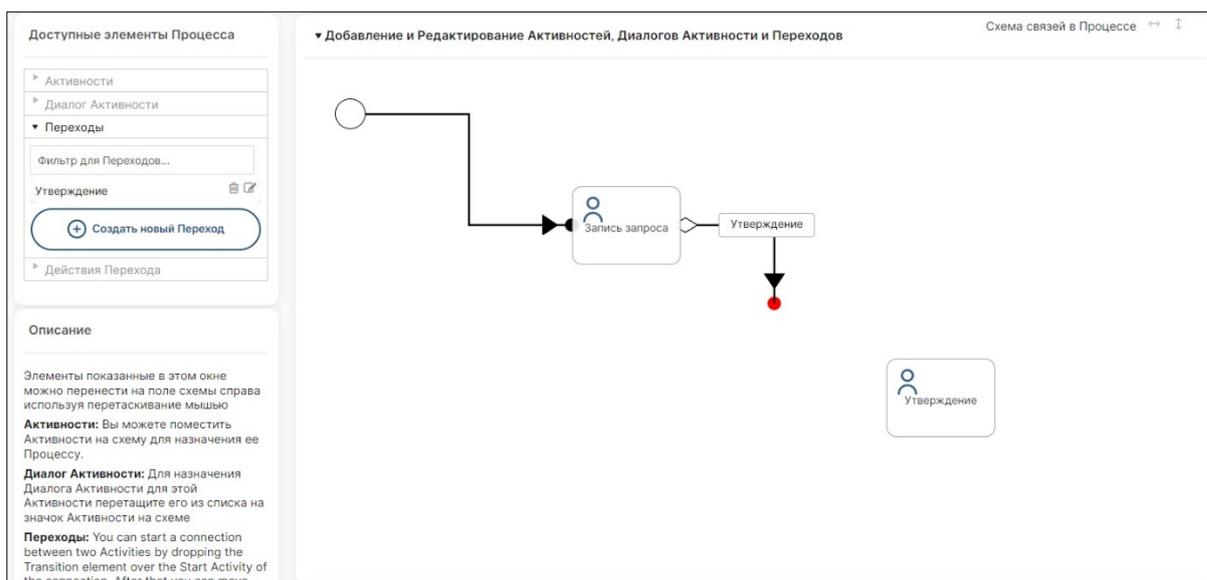
Затем перетащите еще одну активность на схему. Теперь у нас будет две активности на холсте. Первый подключен к начальной точке, а второй не имеет соединений. Вы можете навести указатель мыши на каждое действие, чтобы открыть их собственные диалоговые окна действий.

Рисунок 5.19. Перетащите вторую активность на холст



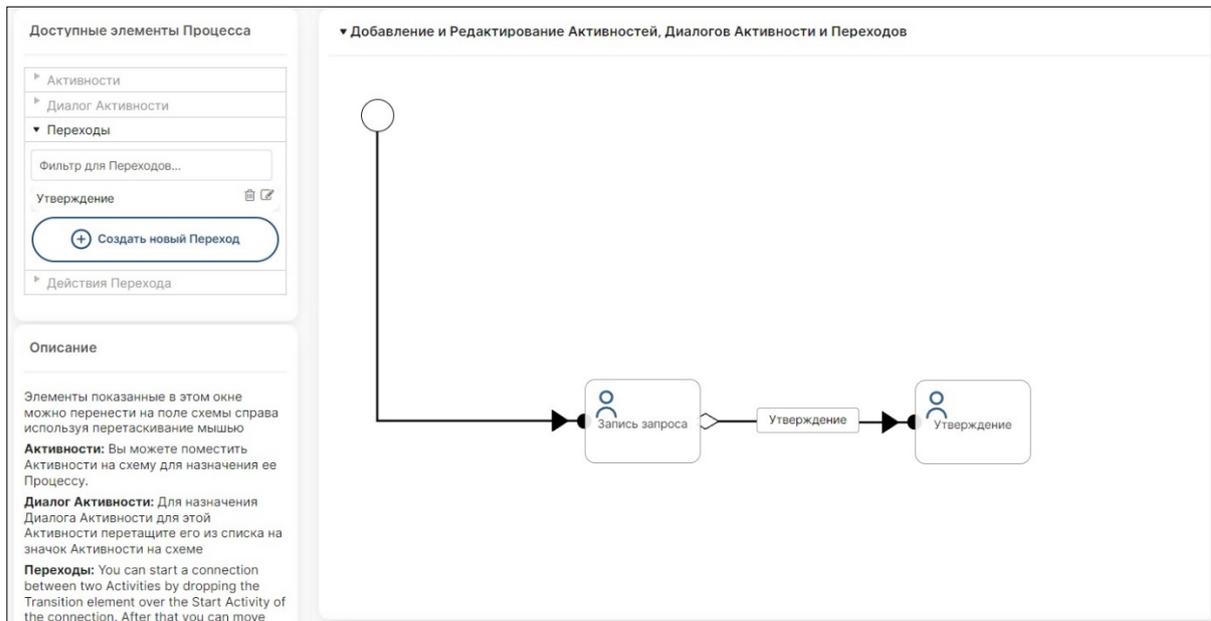
Затем давайте создадим «Путь» (соединение) между этими двумя Действиями, для этого мы будем использовать Переходы. Нажмите «Переходы» в «гармошке», перетащите переход и поместите его внутрь первого действия. Обратите внимание, что действие меняет свой цвет, указывая на то, что переход присоединен. Как только переход будет удален, конечная точка стрелки перехода будет помещена рядом с начальной точкой процесса. Перетащите конечную точку стрелки перехода внутрь другого действия, чтобы создать связь между действиями.

Рисунок 5.20. Перетащите переход на схему



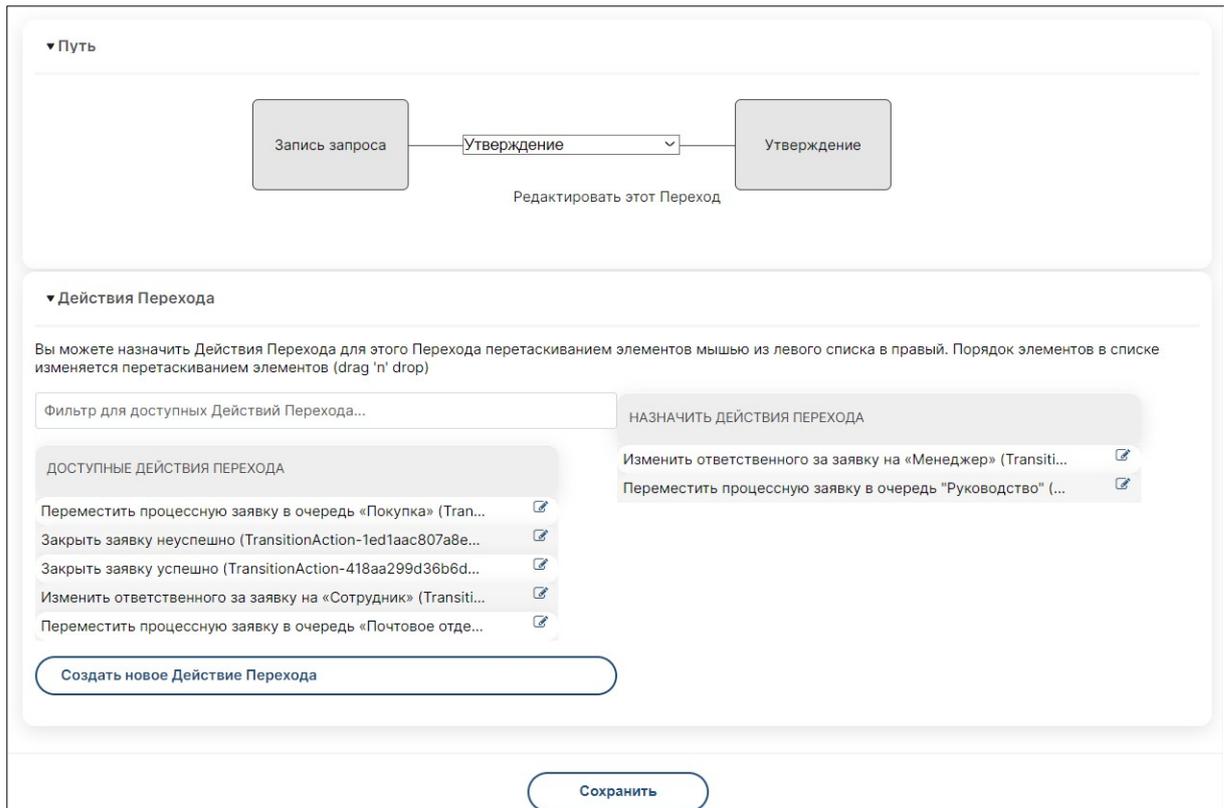
Теперь, когда «Путь» между действиями определен, нам нужно назначить действия перехода переходу, дважды щелкните метку перехода (на схеме), это откроет новое всплывающее окно.

Рисунок 5.21. Соедините действия с помощью переходов



Перетащите необходимые действия перехода из списка доступных действий перехода и поместите их в список назначенных действий перехода и нажмите кнопку «Отправить».

Рисунок 5.22. Назначение действий перехода



Затем вернитесь на главный экран редактирования процесса, нажмите кнопку «Сохранить» под схемой, чтобы сохранить все остальные изменения.

Завершите «путь», добавив следующие Активности, Переходы и Действия Перехода:
Запись запроса до «Утверждения»

- Начальная точка: Активность: «Запись запроса» (A1)
- Возможный переход: «Утверждение» (T1)
 - Если условие этого действия выполнено, заявка перейдет к действию: «Утверждение» (A2).
 - Кроме того, выполняются следующие действия перехода:
 - «Переместить процессную заявку в очередь «Управление»» (TA1)
 - «Изменить ответственного за заявку на «менеджера»» (TA2)

Активность: «Запись запроса» (A1) — это определенный шаг процессной заявки, где есть возможность для Перехода: «Утверждение» (T1). Если это применимо, заявка переместится к следующему действию: «Утверждение» (A2) и к следующим Действиям Перехода: «Переместить процессную заявку в очередь «Управление»» (TA1) и «Изменить ответственного за заявку на «менеджера»». (TA2). В действии: «Утверждение» (A2) доступны диалоги действий: «Утверждение отклонено» (AD2) и «Утверждено» (AD3).

Утверждение

- Начальная точка: Активность «Утверждение» (A2)
- Возможные переходы:
 - «Утверждение отклонено» (T2)
 - Если это соответствует, процессная заявка переместится в Активность: «Процесс завершен» (A5).
 - Кроме того, выполняются следующие действия перехода:
 - "Переместить процессную заявку в очередь «Сотрудники»» (TA3)
 - «Изменить ответственного за заявку на «Сотрудника»» (TA4)
 - «Закрыть заявку unsuccessfully» (TA8)
 - «Утверждено» (T3)
 - Если это соответствует, процессная заявка переместится в действие: «Заказ» (A3).
 - Кроме того, выполняется следующее действие перехода:
 - «Переместить процессную заявку в очередь «Покупка»» (TA5)

Мы можем видеть, что из текущего действия, которое определяет шаг заявки процесса, существует одна или несколько возможностей для перехода, которые имеют ровно одно целевое действие (и, возможно, одно или несколько действий перехода).

Заказ

- Начальная точка: Активность «Заказ» (A3)
- Возможные переходы:
 - «Заказ отклонен» (T4)
 - Если это соответствует, процессная заявка переместится в действие: «Процесс завершен» (A5).
 - Кроме того, выполняются следующие действия перехода:
 - «Переместить процессную заявку в очередь «Сотрудники»» (TA3)

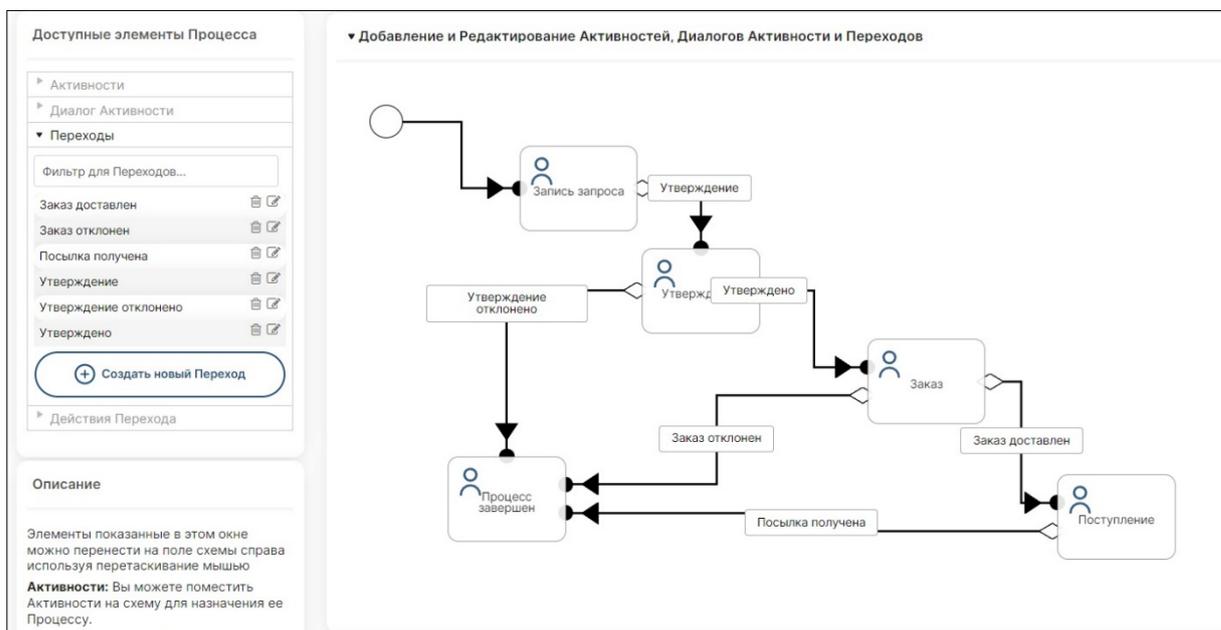
- «Изменить ответственного за заявку на «Сотрудника»» (TA4)
- «Закрывать заявку unsuccessfully» (TA8)
- "Заказ доставлен" (T5)
 - Если это соответствует, процессная заявка переместится в действие: «Поступление» (A4).
 - Кроме того, выполняется следующее действие перехода:
 - «Переместить процессную заявку в очередь «Почтовое отделение»» (TA6)

Поступление

- Начальная точка: Активность «Поступление» (A4)
- Возможные переходы:
 - «Посылка получена» (T6)
 - Если это соответствует, процессная заявка переместится в действие: Если это соответствует «Процесс завершен» (A5).
 - Кроме того, выполняются следующие действия перехода:
 - «Переместить процессную заявку в очередь «Сотрудники»» (TA3)
 - «Изменить ответственного за заявку на «Сотрудника»» (TA4)
 - «Закрывать заявку успешно» (TA7)

Полный путь процесса заказа книг будет выглядеть следующим образом:

Рисунок 5.23. Законченная/полная схема процесса заказа книги.



После завершения пути процесса нажмите кнопку «Сохранить» в нижней части схемы, а затем нажмите кнопку «Синхронизировать все процессы». Это соберет всю информацию о процессах из базы данных и создаст файл кеша (на языке Perl). Этот файл кеша на самом деле является конфигурацией процессов, которую система будет использовать для создания или использования заявок на процессы.

Любое изменение, внесенное в процесс (в графическом интерфейсе), потребует повторной синхронизации файла кеша, чтобы изменение отразилось в системе.

Также можно импортировать весь процесс из файла YAML, но все же необходимо создать все динамические поля, пользователей, очереди и т. д., которые необходимы каждому процессу перед импортом.

Обратите внимание, что, если процесс требует использования ACL, их также необходимо установить вручную.

Ниже приведен полный файл YAML для примера процесса заказа книг:

```
---
```

```
Activities:
```

```
A1:
```

```
ActivityDialogs:
```

```
- AD1
```

```
ChangeTime: 2012-11-23 14:49:22
```

```
Config:
```

```
ActivityDialog:
```

```
1: AD1
```

```
CreateTime: 2012-11-23 11:49:38
```

```
EntityID: A1
```

```
ID: 151
```

```
Name: Recording the demand
```

```
A2:
```

```
ActivityDialogs:
```

```
- AD2
```

```
- AD3
```

```
ChangeTime: 2012-12-13 00:55:12
```

```
Config:
```

```
ActivityDialog:
```

```
1: AD2
```

```
2: AD3
```

CreateTime: 2012-11-23 11:50:11

EntityID: A2

ID: 152

Name: Approval

A3:

ActivityDialogs:

- AD4

- AD5

ChangeTime: 2012-11-23 18:12:14

Config:

ActivityDialog:

1: AD4

2: AD5

CreateTime: 2012-11-23 11:50:35

EntityID: A3

ID: 153

Name: Order

A4:

ActivityDialogs:

- AD6

ChangeTime: 2012-11-23 18:12:35

Config:

ActivityDialog:

1: AD6

CreateTime: 2012-11-23 11:51:00

EntityID: A4

ID: 154

Name: Incoming

A5:

ActivityDialogs: []

ChangeTime: 2012-11-23 11:51:33

Config: {}

CreateTime: 2012-11-23 11:51:33

EntityID: A5

ID: 155

Name: Process complete

ActivityDialogs:

AD1:

ChangeTime: 2012-12-06 02:16:21

Config:

DescriptionLong: "

DescriptionShort: Recoding the demand

FieldOrder:

- DynamicField_Author
- DynamicField_ISBN
- DynamicField_Title
- DynamicField_Status

Fields:

DynamicField_Author:

DefaultValue: "

DescriptionLong: "

DescriptionShort: "

Display: 1

DynamicField_ISBN:

DefaultValue: "
DescriptionLong: "
DescriptionShort: "
Display: 1
DynamicField_Status:
DefaultValue: "
DescriptionLong: "
DescriptionShort: "
Display: 1
DynamicField_Title:
DefaultValue: "
DescriptionLong: "
DescriptionShort: "
Display: 1
Interface:
 - AgentInterface
Permission: "
RequiredLock: 0
SubmitAdviceText: "
SubmitButtonText: "
CreateTime: 2012-11-23 14:34:43
EntityID: AD1
ID: 154
Name: Recording the demand
AD2:
ChangeTime: 2012-11-23 14:57:41
Config:

DescriptionLong: "

DescriptionShort: Approval denied

FieldOrder:

- Article
- DynamicField_Status

Fields:

Article:

Config:

CommunicationChannel: 'Internal'

IsVisibleForCustomer: '0'

DefaultValue: "

DescriptionLong: "

DescriptionShort: "

Display: 1

DynamicField_Status:

DefaultValue: "

DescriptionLong: "

DescriptionShort: "

Display: 1

Interface:

- AgentInterface

Permission: "

RequiredLock: 0

SubmitAdviceText: "

SubmitButtonText: Deny Request

CreateTime: 2012-11-23 14:36:39

EntityID: AD2

ID: 155

Name: Approval denied

AD3:

ChangeTime: 2012-12-14 03:14:23

Config:

DescriptionLong: "

DescriptionShort: Approved

FieldOrder:

- DynamicField_Status

Fields:

DynamicField_Status:

DefaultValue: "

DescriptionLong: "

DescriptionShort: "

Display: 1

Interface:

- AgentInterface

Permission: "

RequiredLock: 0

SubmitAdviceText: "

SubmitButtonText: Approve Request

CreateTime: 2012-11-23 14:37:35

EntityID: AD3

ID: 156

Name: Approved

AD4:

ChangeTime: 2012-11-23 14:58:52

Config:

DescriptionLong: "

DescriptionShort: Order rejected

FieldOrder:

- Article
- DynamicField_Status

Fields:

Article:

Config:

CommunicationChannel: 'Internal'

IsVisibleForCustomer: '0'

DefaultValue: "

DescriptionLong: "

DescriptionShort: "

Display: 1

DynamicField_Status:

DefaultValue: "

DescriptionLong: "

DescriptionShort: "

Display: 1

Interface:

- AgentInterface

Permission: "

RequiredLock: 0

SubmitAdviceText: "

SubmitButtonText: Reject Order

CreateTime: 2012-11-23 14:38:48

EntityID: AD4

ID: 157

Name: Order rejected

AD5:

ChangeTime: 2012-12-06 02:20:12

Config:

DescriptionLong: "

DescriptionShort: Order placed

FieldOrder:

- DynamicField_DeliveryDate
- DynamicField_Price
- DynamicField_Supplier
- DynamicField_Status

Fields:

DynamicField_DeliveryDate:

DefaultValue: "

DescriptionLong: "

DescriptionShort: "

Display: 1

DynamicField_Price:

DefaultValue: "

DescriptionLong: "

DescriptionShort: "

Display: 1

DynamicField_Status:

DefaultValue: "

DescriptionLong: "

DescriptionShort: "
Display: 1
DynamicField_Supplier:
DefaultValue: "
DescriptionLong: "
DescriptionShort: "
Display: 1
Interface:
- AgentInterface
Permission: "
RequiredLock: 0
SubmitAdviceText: "
SubmitButtonText: Place Order
CreateTime: 2012-11-23 14:41:28
EntityID: AD5
ID: 158
Name: Order placed
AD6:
ChangeTime: 2012-11-23 14:42:43
Config:
DescriptionLong: "
DescriptionShort: Shipment received
FieldOrder:
- DynamicField_DateOfReceipt
- DynamicField_Status
Fields:
DynamicField_DateOfReceipt:

DefaultValue: "
DescriptionLong: "
DescriptionShort: "
Display: 1
DynamicField_Status:
DefaultValue: "
DescriptionLong: "
DescriptionShort: "
Display: 1
Interface:
- AgentInterface
Permission: "
RequiredLock: 0
SubmitAdviceText: "
SubmitButtonText: "
CreateTime: 2012-11-23 14:42:43
EntityID: AD6 I
D: 159
Name: Shipment received
Process:
Activities:
- A1
- A2
- A3
- A4
- A5
ChangeTime: 2012-12-06 02:31:59

Config:

Description: The process to order a book

Path:

A1:

T1:

ActivityEntityID: A2

TransitionAction:

- TA2
- TA1

A2:

T2:

ActivityEntityID: A5

TransitionAction:

- TA3
- TA4
- TA8

T3:

ActivityEntityID: A3

TransitionAction:

- TA5

A3:

T4:

ActivityEntityID: A5

TransitionAction:

- TA3
- TA4
- TA8

T5:

ActivityEntityID: A4

TransitionAction:

- TA6

A4:

T6:

ActivityEntityID: A5

TransitionAction:

- TA3

- TA4

- TA7

A5: {}

StartActivity: A1

StartActivityDialog: AD1

CreateTime: 2012-11-23 11:45:12

EntityID: P1

ID: 94

Layout:

A1:

left: 172

top: 63

A2:

left: 402

top: 156

A3:

left: 649

top: 255

A4:

left: 774

top: 391

A5:

left: 194

top: 410

Name: Book ordering

State: Active

StateEntityID: S1

TransitionActions:

- TA1

- TA2

- TA3

- TA4

- TA8

- TA5

- TA3

- TA4

- TA8

- TA6

- TA3

- TA4

- TA7

Transitions:

- T1

- T2

- T3

- T4

- T5

- T6

TransitionActions:

TA1:

ChangeTime: 2012-11-23 16:01:37

Config:

Config:

Queue: Management

Module: Kernel::System::ProcessManagement::TransitionAction::TicketQueueSet

CreateTime: 2012-11-23 15:50:59

EntityID: TA1

ID: 61

Name: Move the process ticket into the "Management" queue

TA2:

ChangeTime: 2012-11-23 16:02:12

Config:

Config:

Responsible: manager

Module:

Kernel::System::ProcessManagement::TransitionAction::TicketResponsibleSet

CreateTime: 2012-11-23 15:58:22

EntityID: TA2

ID: 62

Name: Change ticket responsible to "manager"

TA3:

ChangeTime: 2012-11-24 14:27:02

Config:

Config:

Queue: Employees

Module: Kernel::System::ProcessManagement::TransitionAction::TicketQueueSet

CreateTime: 2012-11-23 16:02:54

EntityID: TA3

ID: 63

Name: Move the process ticket into the "Employees" queue

TA4:

ChangeTime: 2012-11-23 16:04:06

Config:

Config:

Responsible: Employee

Module:

Kernel::System::ProcessManagement::TransitionAction::TicketResponsibleSet

CreateTime: 2012-11-23 16:04:06

EntityID: TA4

ID: 64

Name: Change ticket responsible to "Employee"

TA5:

ChangeTime: 2012-12-06 02:18:34

Config:

Config:

Queue: Purchasing

Module: Kernel::System::ProcessManagement::TransitionAction::TicketQueueSet

CreateTime: 2012-11-23 16:04:54

EntityID: TA5

ID: 65

Name: Move process ticket into the "Purchasing" queue

TA6:

ChangeTime: 2012-12-06 02:18:48

Config:

Config:

Queue: Post office

Module: Kernel::System::ProcessManagement::TransitionAction::TicketQueueSet

CreateTime: 2012-11-23 16:06:20

EntityID: TA6

ID: 66

Name: Move process ticket into the "Post office" queue

TA7:

ChangeTime: 2012-12-06 02:29:55

Config:

Config:

State: closed successful

Module: Kernel::System::ProcessManagement::TransitionAction::TicketStateSet

CreateTime: 2012-12-06 02:29:27

EntityID: TA7

ID: 67

Name: Close ticket successfully

TA8:

ChangeTime: 2012-12-06 02:31:12

Config:

Config:

State: closed unsuccessful

Module: Kernel::System::ProcessManagement::TransitionAction::TicketStateSet

CreateTime: 2012-12-06 02:31:12

EntityID: TA8

ID: 68

Name: Close ticket unsuccessfully

Transitions:

T1:

ChangeTime: 2012-11-23 15:12:20

Config:

Condition:

1:

Fields:

DynamicField_Status:

Match: Approval

Type: String

Type: and

ConditionLinking: and

CreateTime: 2012-11-23 11:53:52

EntityID: T1

ID: 94

Name: Approval

T2:

ChangeTime: 2012-11-23 15:12:50

Config:

Condition:

1:

Fields:

DynamicField_Status:

Match: Approval denied

Type: String

Type: and

ConditionLinking: and

CreateTime: 2012-11-23 11:54:26

EntityID: T2

ID: 95

Name: Approval denied

T3:

ChangeTime: 2012-11-23 15:13:29

Config:

Condition:

1:

Fields:

DynamicField_Status:

Match: Approved

Type: String

Type: and

ConditionLinking: and

CreateTime: 2012-11-23 11:54:54

EntityID: T3

ID: 96

Name: Approved

T4:

ChangeTime: 2012-11-23 15:14:08

Config:

Condition:

1:

Fields:

DynamicField_Status:

Match: Order denied

Type: String

Type: and

ConditionLinking: and

CreateTime: 2012-11-23 11:55:25

EntityID: T4

ID: 97

Name: Order denied

T5:

ChangeTime: 2012-11-23 18:30:33

Config:

Condition:

1:

Fields:

DynamicField_Status:

Match: Order placed

Type: String

Type: and

ConditionLinking: and

CreateTime: 2012-11-23 11:56:15

EntityID: T5

ID: 98

Name: Order placed

T6:
ChangeTime: 2012-11-23 15:15:30
Config:
Condition:
1:
Fields:
DynamicField_Status:
Match: Shipment received
Type: String
Type: and
ConditionLinking: and
CreateTime: 2012-11-23 11:56:48
EntityID: T6
ID: 99
Name: Shipment received

4. Справочник по конфигурации процесса

4.1. Процесс

Процесс моделирует путь рабочего процесса. Путьевые точки на этом пути могут быть действиями или переходами, мы поговорим об этом позже.

Конфигурация процесса

Конфигурацию процесса можно выполнить в файле Kernel/Config.pm, но настоятельно рекомендуется создавать новые файлы, такие как Kernel/Config/Files/MyProcess.pm. Обратите внимание, что графический интерфейс генерирует файл Kernel/Config/File/ZZZProcessManagement. Не используйте это имя файла, иначе оно будет перезаписано при синхронизации процессов. Давайте посмотрим пример конфигурации процесса (из файла кеша процесса):

```
$Self->{'Process'} = {  
  
'P1' => {
```



```
State => 'Active',

StartActivity=> 'A2',

StartActivityDialog => 'AD2',

Path => {

'A2' => {

'T3' => {

ActivityEntityID => 'A4',

},

},

},

}

};
```

Name

Имя процесса, которое может быть выбрано агентом при создании новой процессной заявки.

CreateTime

Время создания процесса.

CreateBy

UID пользователя, создавшего процесс.

ChangeTime

Время, когда процесс был изменен.

ChangeBy

UID пользователя, внесшего последнее изменение в процесс.

State

Определяет состояние процесса. Возможные значения:

- Активировать все процессы, которые можно использовать в новых процессных заявках.
- FadeAway – это процессы, которые больше нельзя выбирать для новых заявок, но существующие заявки по-прежнему могут использовать этот процесс.
- Неактивные процессы деактивируются и не могут использоваться для новых или существующих заявок.

StartActivity

При создании новой процессной заявки необходимо определить StartActivity. Как только заявка будет создана, это действие будет установлено и использовано в качестве основы для первых проверок перехода.

StartActivityDialog

Для новых процессных заявок должен быть определен StartActivityDialog. Это будет показано при создании новой процессной заявки (после выбора процесса). На данный момент заявка еще не существует, она будет создана после отправки StartActivityDialog.

Path

Путь содержит структуру действий и возможные переходы между ними для текущего процесса. А также действия по переходу, которые происходят при переходе. Это контролирует путь, по которому может пройти процессная заявка. Пример:

```
'A1' => {  
  
  'T1' => {  
  
    ActivityEntityID => 'A2',  
  
  },  
  
  'T2' => {  
  
    ActivityEntityID => 'A3',  
  
  },  
  
  'T3' => {  
  
    ActivityEntityID => 'A4', TransitionAction => ['TA1', 'TA2'],  
  
  },  
  
},
```

Если процессная заявка находится в действии A1, у него есть три возможных способа перейти к другому действию. В Переходах от T1 до T3 определяются условия, которым должна соответствовать процессная заявка, чтобы перейти к другой Активности.

Если в этом случае все значения процессной заявки и ее динамических полей, необходимые для Перехода T2, верны, то заявка будет перемещена из Активности A1 в A3. После отправки Диалога Активности или внесения любых других изменений в заявку она будет проверена на предмет возможных переходов из текущей Активности. Если возможны несколько переходов, будет использоваться первый из них (на основе числовой сортировки идентификаторов переходов).

Кроме того, можно назначать переходные действия для переходов в конфигурации пути. Это модули, которые выполняются после успешного перехода. Они должны быть указаны в виде массива, как в примере, о деталях мы поговорим позже.

4.2. Активность

Действие содержит один или несколько диалогов действий и моделирует шаг в процессе. Все Диалоговые окна текущего Действия отображаются в обзоре заявки и могут использоваться до тех пор, пока не будут выполнены условия Перехода.

Конфигурация активности

Давайте рассмотрим пример конфигурации активности:

```
$Self->{'Process::Activity'} =  
  
{  
  
'A1' => {  
  
  Name           => 'Activity 1 optional', CreateTime => '16-02-2012 13:37:00',  
  
  CreateBy       => '1',  
  
  ChangeTime => '17-02-2012 13:37:00',  
  
  ChangeBy       => '1', ActivityDialog => {  
  
    1 => 'AD1',  
  
  },  
  
  },  
  
'A2' => {  
  
  Name           => 'Activity 2 optional', CreateTime => '16-02-2012 13:37:00',  
  
  CreateBy       => '1',  
  
  ChangeTime => '17-02-2012 13:37:00',  
  
  ChangeBy       => '1', ActivityDialog => {  
  
    1 => 'AD5',  
  
    2 => 'AD6',  
  
    3 => 'AD1',  
  
  },  
  
  },  
  
}
```

```
};
```

Name

Название активности.

CreateTime

Время, когда она была создана.

CreateBy

UID пользователя, создавшего Активность.

ChangeTime

Время, когда Активность была изменена.

ChangeBy

UID последнего пользователя, изменившего активность.

ActivityDialog

Диалог Активности содержит список Диалогов Активности, доступных в этой Активности. Все Диалоги Активности текущей Активности отображаются в обзоре заявки. Их порядок задается порядком в конфигурации, здесь AD5 отображается перед AD6 и AD1.

4.3. Диалог Активности

Диалог Активности— это отдельный экран, который можно использовать в различных Активностях.

Конфигурация диалога действий

Рассмотрим пример конфига:

```
$Self->{'Process::ActivityDialog'} = {  
  'AD1' => {  
    Name => 'ActivityDialog 1 optional',  
    DescriptionShort => 'Basic info',  
    DescriptionLong => 'Please insert the necessesary basic information for IT orders',  
    CreateTime => '28-02-2012 13:37:00',  
    CreateBy => '1',  
    ChangeTime => '29-02-2012 13:37:00',
```

```
ChangeBy => '1',

Fields => {

PriorityID => {

DescriptionShort => 'Priority ID',

DescriptionLong => 'Enter the priority here',

Display    => 2,

},

},

FieldOrder => [ 'PriorityID' ],

SubmitAdviceText => 'Note: If you submit the form...',

SubmitButtonText => 'Send request',

},

'AD2' => {

Name => 'ActivityDialog 2 optional',

DescriptionShort => 'Basic info',

DescriptionLong => 'Please insert the necessesary basic information for Book orders',

CreateTime => '28-02-2012 13:37:00',

CreateBy => '1',

ChangeTime => '29-02-2012 13:37:00',

ChangeBy => '1',

Fields => {

StateID => {

DescriptionShort => 'State ID',

DescriptionLong => 'Enter the state here',

Display => 2,

DefaultValue => '2',

},

},

},
```

```

Queue => {
  DescriptionShort => 'Queue ID',
  DescriptionLong => 'Enter the queue here',
  Display => 2,
  DefaultValue => 'Raw',
},
Title => {
  DescriptionShort => 'Title',
  DescriptionLong => 'Enter the title here',
  Display => 1,
  DefaultValue => 'Default Title',
},
DynamicField_Anzahl => {
  DescriptionShort => 'Amount',
  DescriptionLong => 'Enter the amount here',
  Display => 2,
  DefaultValue => '4',
},
},
FieldOrder => [ 'DynamicField_Anzahl', 'StateID', 'Queue', 'Title' ],
SubmitAdviceText => 'Note: If you submit the form...',
SubmitButtonText => 'Send request',
},
};

```

Name

Имя Диалога Активности.

CreateTime

Время, когда он был создан.

CreateBy

UID пользователя, создавшего этот диалог активности.

ChangeTime

Время последнего изменения.

ChangeBy

UID последнего пользователя, изменившего этот диалог активности.

Fields

Содержит все поля, которые могут отображаться в этом Диалоге Активности. В настоящее время можно использовать следующие поля:

```
Title State StateID Priority PriorityID Lock LockID Queue QueueID
Customer
CustomerID CustomerNo CustomerUserID Owner
OwnerID Type TypeID SLA SLAID
Service ServiceID Responsible ResponsibleID PendingTime
DynamicField_ $FieldName # for all dynamic fields
```

Пример конфигурации одного поля:

```
StateID => {
  DescriptionShort => 'State ID', DescriptionLong => 'Enter the state here', Display
    => 2,
  DefaultValue          => '2',
},
```

Поле Сообщение является особым случаем. Если он присутствует в конфигурации «Поля», диалоговое окно «Действие» будет содержать полный редактор Richtext с темой и обработкой вложений. Введенный текст будет добавлен к заявке как заметка. Давайте рассмотрим поле Сообщение на примере конфигурации:

```
Article => {
  DescriptionShort => 'Please insert your comment here.', DescriptionLong => "",
```

```
Display      => 1,  
Config      => {  
  CommunicationChannel => 'Internal' IsVisibleForCustomer => '0' LabelSubject => ",  
  LabelBody      => ",  
},  
},
```

Давайте посмотрим на параметры конфигурации поля:

1. DescriptionShort

Необязательное краткое описание, отображаемое с заголовком поля.

2. DescriptionLong

Необязательное более длинное описание поля, которое отображается, когда мышь находится над полем, например, совет о том, как заполнить поле.

3. Display

Определяет, отображается ли поле и/или обязательно. Возможные значения:

- 0: поле невидимо. Это может быть полезно, если значения полей должны устанавливаться автоматически. В этом случае настроенное по умолчанию значение будет сохранено.
- 1: поле видимое, но необязательное.

2: поле видимое и обязательное для заполнения. Следующие поля могут быть только невидимыми или обязательными:

```
QueueID  
Queue  
State  
StateID  
Lock  
LockID  
Priority  
PriorityID  
Type
```

TypeID

Если поля сконфигурированы как необязательные и пользователь не вводит никаких значений, значение по умолчанию будет сохранено при отправке Диалога Активности пользователем.

4. Значение по умолчанию

Для полей с идентификатором (например, QueueID, OwnerID), система использует идентификатор в базе данных в качестве значения. Для других полей без "ID" (таких как Queue, Owner), значение по умолчанию должно содержать собственно значение. Например:

```
Queue => {  
  DescriptionShort => 'Queue', DescriptionLong => 'Enter the queue here', Display  
    => 2,  
  DefaultValue      => 'Raw',  
},
```

FieldOrder

Здесь настраивается порядок отображения полей. ВАЖНО: Здесь также необходимо настроить невидимые поля, поскольку при сохранении будут учитываться только настроенные поля. Поля, которые не настроены, не будут сохранены.

SubmitAdviceText

Необязательный текст, который будет отображаться прямо над кнопкой отправки для дополнительной помощи или текста совета.

SubmitButtonText

Необязательный пользовательский текст для кнопки отправки.

4.4. Переход

Переход решает - на основе настраиваемых условий - какой путь в Процессе будет выбран, т.е. в какую Активность можно переместить процессную заявку.

Конфигурация перехода

Давайте посмотрим пример:

```

$Self->{'Process::Transition'} = {
'T1' => {
Name => 'Transition 1',
CreateTime => '14-03-2012 13:37:00', # optional
CreateBy => '1', # optional
ChangeTime => '15-03-2012 13:37:00', # optional
ChangeBy => '15-03-2012 13:37:00', # optional
Condition => {
Cond1 => {
Fields => {
StateID => {
Type => 'String',
Match => '1',
},
},
},
},
},
},
'T2' => {
Name          => 'Transition 2 optional',
CreateTime => 'DATE', # optional
CreateBy => 'USERID', # optional
ChangeTime => 'DATE', # optional
ChangeBy => 'USERID', # optional
Condition => {
Cond1 => {

```

```
Queue=> 'Raw',  
  
DynamicField_Farbe => '2',  
  
DynamicField_Anzahl => '1',  
  
},  
  
},  
  
},  
  
};
```

Name

Название перехода.

CreateTime

Время, когда он был создан.

CreateBy

UID пользователя, создавшего этот переход.

ChangeTime

Время последнего изменения.

ChangeBy

UID последнего пользователя, изменившего этот Переход.

Condition

Содержит все условия, необходимые для того, чтобы этот Переход вступил в силу.
Пример:

```
Condition => {  
  
Type => 'and', Cond1 => {  
  
Type      => 'and', Fields => {  
  
StateID => {  
  
Type => 'String', Match => '1',  
  
},  
  
DynamicField_Marke => { Type => 'String', Match => 'VW',
```

```
},  
  
},  
  
Cond2 => {  
  
  Type => 'and',  
  
  Fields => {  
  
    Queue => {  
  
      Type => 'String', Match => 'Raw',  
  
    },  
  
  },  
  
},  
  
},  
  
},
```

Рассмотрим подробнее конфигурирование Условий.

1. Type in Condition

Указывает, как различные элементы условия связаны друг с другом. Возможные значения:

- and: это значение по умолчанию. Все условия должны быть соблюдены, чтобы переход вступил в силу.
- or: Должно совпасть хотя бы одно условие.
- xor: должно совпасть ровно одно условие, не более.

2. Cond1

Это имя одного из условий примера. Имя выбирается произвольно. Условия вычисляются в порядке очередности.

3. Type in Cond

Определяет, как отдельные проверяемые поля этого условия связаны друг с другом. Возможные значения:

- and: это значение по умолчанию. Все проверяемые поля должны удовлетворять условию, чтобы результат был - "верно".
- or: хотя бы одно из полей должно удовлетворять условию.
- xor: не более одного поля должно соответствовать условию.

4. Fields

Задаёт конкретные поля, значения которых необходимо проверить. Из нашего примера:

```
Fields => {  
  StateID => {  
    Type => 'String', Match => '1',  
  },  
}
```

5. StateID

Пример имени поля. Можно использовать следующие поля заявки:

```
Title State StateID Priority  
PriorityID  
Lock LockID Queue QueueID Customer CustomerID CustomerNo  
CustomerUserID Owner  
OwnerID Type TypeID SLA SLAID  
Service ServiceID Responsible ResponsibleID  
DynamicField_ $FieldName # for all DynamicFields
```

При проверке поля с идентификатором (например, SLAID) используется идентификатор поля из базы данных, для других полей (например, SLA) используется текущее значение.

6. Type

Определяет тип проверяемого поля. Возможные значения:

- 'String/Строка': Сравнивает значение поля со строкой, заданной в 'Match'. Верно, если их значения полностью совпадают.
- 'Hash': Сравнивает значение поля (hash) со значением hash заданным в 'Match'. Все значения должны совпадать.
- 'Array/Список': Сравнивает значение поля (array) со списком, заданным в 'Match'. Оба списка должны быть одинаковы.
- 'Regex': Значение поля проверяется с помощью регулярного выражения. Важным является, что 'Match' содержит *qr{}* как базовое условие. Между скобками может быть записано реальное регулярное выражение.
- 'Module': Позволяет использовать perl модуль для проверки условия. Если он возвращает, в качестве результата 1, результат проверки верен. Пример модуля можно найти в `Kernel/System/ProcessManagement/TransitionValidation/ValidateDemo.pm`.

4.5. Действия перехода

Действия Перехода - это действия, которые могут быть выполнены после успешного Перехода (когда процессная заявка переходит от одной Активности к другой). Эти Действия Перехода могут быть использованы для выполнения различных изменений в заявке, например, изменение Очереди или Владельца заявки. Вы также можете создать свои собственные Действия Переходов для осуществления других сложных изменений.

Конфигурация Действия Перехода

Давайте рассмотрим пример:

```
$Self->{'Process::TransitionAction'} = { 'TA1' => {  
  
  Name      => 'Queue Move',  
  
  Module                                         =>  
'Kernel::System::ProcessManagement::TransitionAction::TicketQueueSet', Config =>  
{  
  
  Queue => 'Junk', UserID => 123,  
  
  },  
  
  },  
  
  };
```

Name

Имя Действия Перехода.

Module

Указывает модуль Perl, который будет использоваться.

Config

Этот параметр содержит все настройки, необходимые для модуля. Его содержание зависит от конкретного используемого модуля Действий Переходов. Подробности см. в документации к отдельным модулям. В нашем примере необходимо указать только очередь. Тем не менее, мы также отправляем параметр UserID, используя параметр UserID. Действие Перехода будет выполнено от лица пользователя с заданным UserID.

Использование UserID в параметре Config Действия Перехода допускается всеми действиями перехода (начиная с RSD 3.2.4). В этом примере это может быть особенно важно, если пользователь, инициирующий переход, не имеет прав на перемещение заявки в очередь Junk, в то время как у пользователя с UserID 123 они могут быть.

Повторное использование модулей Действий Переходов.

Чтобы использовать модули Действий Переходов несколько раз, просто укажите несколько Действий Переходов в вашей конфигурации. Пример:

```

$Self->{'Process::TransitionAction'} = { 'TA1' => {
    Name      => 'Queue Move Junk',

    Module                                         =>
'Kernel::System::ProcessManagement::TransitionAction::TicketQueueSet', Config =>
    {
        Queue => 'Junk',

    },

},

'TA2' => {

    Name      => 'Queue Move Raw',

    Module                                         =>
'Kernel::System::ProcessManagement::TransitionAction::TicketQueueSet', Config =>
    {
        Queue => 'Raw',

    },

},

};

```

Здесь тот же модуль используется для перемещения процессной заявки в очередь Raw, а в другой раз - для перемещения ее в очередь Junk. Действие перехода, которое должно использоваться для конкретного перехода, определяется параметром пути из конфигурации процесса.

Доступные действия по переходу

RSD поставляется с несколькими действиями перехода, которые можно использовать в ваших процессах. Здесь вы можете найти их документацию и то, как их необходимо настроить.

1. Набор динамических полей

Задает одно или несколько динамических полей в процессной заявке. Пример:

```

$Self->{'Process::TransitionAction'} = { 'TA1' => {
    Name      => 'Set DynamicField MasterSlave to Master and Approved to 1',

```

```

Module                                     =>
'Kernel::System::ProcessManagement::TransitionAction::DynamicFieldSet', Config
=> {

    MasterSlave => 'Master', Approved      => '1',

},

},

};

```

Name задает имя конфигурируемого Действия Перехода.

MasterSlave и Approved приведены как пример имен динамических полей. Значения полей (Master и 1) будут установлены этим Действием Перехода.

2. TicketArticleCreate

Создает сообщение/заметку и может быть использован для создания заметок или почтовых ответов. Параметры %DataPayload зависят от заданного канала связи, по умолчанию используется Internal. Пример:

```

$Self->{"Process::TransitionAction"} = {

    'TA1' => {

        Name => 'Article Create Note',

        Module                                     =>
'Kernel::System::ProcessManagement::TransitionAction::TicketArticleCreate',

        Config => {

            Config => { # (required) agent|system|customer # 0 or 1

                SenderType => 'agent', # Internal|Phone|Email|..., default: Internal

                IsVisibleForCustomer => 1,

                CommunicationChannel => 'Internal',Internal

                %DataPayload, # some parameters depending of each communication channel

            },

        },

    },

};

```

Ниже приведена настройка параметра %DataPayload для каналов связи на основе MIME (электронная почта, внутренняя и телефонная связь).

SenderType => 'agent', # agent|system|customer

ContentType => 'text/plain; charset=ISO-8859-15', # or optional Charset & MimeType
(e.g. 'text/html; charset=UTF-8')

Subject => 'some short description', # required

Body => 'the message text', # required

HistoryType => 'OwnerUpdate', #
EmailCustomer|Move|AddNote|PriorityUpdate|WebRequestCustomer|...

HistoryComment => 'Some free text!'

From => 'Some Agent <email@example.com>', # not required but useful

To => 'Some Customer A <customer-a@example.com>', # not required but useful

Cc => 'Some Customer B <customer-b@example.com>', # not required but useful

ReplyTo => 'Some Customer B <customer-b@example.com>', # not required

InReplyTo => '<asdasdasd.12@example.com>', # not required but useful

References not => '<asdasdasd.1@example.com> <asdasdasd.12@example.com>',
required but useful

NoAgentNotify=> 0, # if you don't want to send agent notifications

AutoResponseType => 'auto reply', #auto reject|auto follow up|auto reply/new ticket|auto remove

ForceNotificationToUserID => '1,43,56', # if you want to force somebody

ExcludeNotificationToUserID => '43,56',

if you want full exclude somebody from notifications,

will also be removed in To: line of article,

higher prio as ForceNotificationToUserID

ExcludeMuteNotificationToUserID => '43,56',

the same as ExcludeNotificationToUserID but only the

sending gets muted, agent will still shown in To:

line of article

Name указывает имя сконфигурированного действия Действия Перехода. Его можно выбрать произвольно, но оно должно отражать цель настроенного действия.

SenderType определяет тип отправителя заметки/сообщения. Возможные значения: агент, система, клиент.

IsVisibleForCustomer определяет, должна ли заметка/сообщение отображаться в пользовательском интерфейсе.

CommunicationChannel определяет тип создаваемой заметки/сообщения. Возможные значения: Электронная почта, Внутренняя и Телефон. Этот список можно расширить, установив новые каналы связи через пакет RSD.

ContentType определяет тип содержимого заметки/сообщения. Возможные значения: text/plain, charset=ISO-8859-15 или любая другая допустимая кодировка или mime тип.

Тема определяет название заметки/сообщения. Обязательна.

Тело определяет содержание заметки/сообщения. Обязательно.

HistoryType/Тип записи Истории задает типы/названия записей истории заявки. Возможные значения: AddNote, Archive-FlagUpdate, Bounce, CustomerUpdate, EmailAgent, EmailCustomer, EscalationResponseTimeNotifyBefore, EscalationResponseTimeStart, EscalationResponseTimeStop, EscalationSolutionTimeNotifyBefore, EscalationSolutionTimeStart, EscalationSolutionTimeS-top, EscalationUpdateTimeNotifyBefore, EscalationUpdateTimeStart, EscalationUpdateTimeStop, LoopUp, Forward, Forward , Merged, Misc, Move, NewTicket, OwnerUpdate, PhoneCallAgent, PhoneCallCustomer, PriorityUpdate, Remove, ResponsibleUpdate, SendAgentNotification, SendAnswer, SendAutoFollowUp, SendAutoReject, SendAutoReply, SendCustomerNotification, ServiceUpdate, SetPendingTime, SLAUpdate, StateUpdate, Subscribe, SystemRequest, TicketDynamicFieldUpdate, TicketLinkAdd , TicketLinkDelete, TimeAccounting, TypeUpdate, Unlock, Unsubscribe, WebRequestCustomer.

HistoryComment определяет содержимое записи истории.

From, To, Cc и ReplyTo берут адреса электронной почты с учетом правил, указанных выше.

InReplyTo и References принимают идентификаторы сообщений электронной почты.

NoAgentNotify — если установлено значение 1, уведомление Агенту по электронной почте не будет отправлено.

AutoResponseType может принимать следующие значения: auto follow up/автоотклик, auto reject/автоотказ, auto remove/автоудаление, auto reply/автоответ, auto reply/new ticket/автоответ-новая заявка.

ForceNotificationToUserID, ExcludeNotificationToUserID, ExcludeMuteNotificationToUserID могут принимать список UserIDs, которые либо всегда уведомляются, либо не уведомляются, либо указаны как уведомленные, но фактически им не было отправлено уведомление по электронной почте.

3. TicketCreate

Создает заявку с сообщением, новая заявка может быть связана с процессной заявкой. Пример:

```
$Self->{'Process::TransitionAction'} = {  
  'TA1' => {  
    Name      => 'Ticket Create',  
    Module => 'Kernel::System::ProcessManagement::TransitionAction::TicketCreate',  
    Config => {  
  
    # ticket required:
```

```

Title      => 'Some Ticket Title',
Queue      => 'Raw',                # or QueueID => 123,
Lock       => 'unlock',
Priority    => '3 normal', # or PriorityID => 2,
State      => 'new',                # or StateID => 5,
CustomerID => '123465',
CustomerUser => 'customer@example.com',
OwnerID    => 'someuserlogin',     # or OwnerID => 123,

# ticket optional:
TN          => $TicketObject->TicketCreateNumber(), # optional
Type        => 'Incident',          # or TypeID => 1, not required
Service     => 'Service A',         # or ServiceID => 1, not
required
SLA         => 'SLA A',              # or SLAID => 1, not required
ResponsibleID => 123,                # not required
ArchiveFlag => 'y',                  # (y|n) not required
PendingTime => '2011-12-23 23:05:00', # optional (for pending states)
PendingTimeDiff => 123,              # optional (for
pending states)

# article required:
SenderType  => 'agent',              # agent|system|customer
CommunicationChannel => 'Internal'    # Internal|Phone|Email|...,
default: Internal
IsVisibleForCustomer => '0'

```

```
%DataPayload, # some parameters depending
of each communication channel

# article optional:

TimeUnit          => 123

# other:

DynamicField_NameX => $Value,

LinkAs            => $LinkType, # Normal, Parent, Child, etc.
(respective original ticket)

UserID           => 123, # optional, to override the UserID from the logged user

},

},

};
```

Name задает имя настраиваемого Действия Перехода. Его можно выбрать произвольно, но желательно, чтобы оно отражало цель действия.

Title - Название заявки.

Queue или QueueID задает имя или идентификатор очереди для использования в новой заявке.

Lock или LockID задает статус блокировки для заявки.

Priority или PriorityID задает приоритет или его идентификатор для использования в новой заявке.

State или StateID задает состояние или его идентификатор для использования в новой заявке.

CustomerID, идентификатор клиента/компании клиента для указания в новой заявке.

CustomerUser, логин клиента, назначаемый новой заявке.

Owner или OwnerID задает логин или идентификатор агента, который будет новым владельцем заявки.

TN - пользовательский номер новой заявки.

Type или TypeID задает название или id типа заявки для использования в новой заявке.

Service или ServiceID задает название или id Сервиса для использования в новой заявке.

SLA or SLAID задает название или id SLA для использования в новой заявке.

ResponsibleID - ID агента, назначаемого ответственным за новую заявку.

PendingTime/Время напоминания - предопределенная дата для установки времени ожидания заявки, если состояние заявки относится к типу состояния - отложить.

PendingTimeDiff - динамическая дата (разница, выраженная в секундах от текущей даты/времени) для установки времени ожидания заявки, если состояние заявки относится к типу состояния – отложить

SenderType/Тип отправителя задает тип отправителя для заметки/сообщения. Возможные значения: агент, система, клиент.

IsVisibleForCustomer определяет, должна ли заметка/сообщение отображаться в пользовательском интерфейсе.

CommunicationChannel определяет тип создаваемой заметки/сообщения. Возможные значения: Электронная почта, Внутренняя и Телефон. Этот список можно расширить, установив новые каналы связи через пакет RSD.

Пожалуйста, проверьте дополнительные параметры для разных каналов заметок/сообщений.

TimeUnit — время, затраченное на текущую заметку/сообщение о заявке, выраженное в секундах, минутах, часах и т. д.

DynamicField_NameX, где DynamicField_ — обязательный префикс, а NameX — имя динамического поля, которое будет установлено в новой заявке (на уровне заявки, а не на уровне заметки).

LinkAs нужен для определения отношения к оригинальной заявке, с точки зрения новой заявки, например, Normal/Обычная, Parent/Родитель, Child/Потомок и т. д.

Смарт-теги RSD, такие как <RSD_CUSTOMER_BODY> или <RSD_CUSTOMER_REAL NAME>, теперь поддерживаются в действии перехода TicketCreate. Эти смарт-теги можно использовать для создания новой заявки и вставки данных из процессной заявки в эту дочернюю заявку. Использование смарт-тегов RSD идентично использованию текстовых шаблонов в уведомлениях о заявках.

4. TicketCustomerSet

Устанавливает клиента для процессной заявки. Пример:

```
$Self->{'Process::TransitionAction'} = { 'TA1' => {  
  Name      => 'Customer Set Customer to test',  
  Module => 'Kernel::System::Process::TransitionAction::TicketCustomerSet', Config =>  
  {  
  No        => 'test',
```

```

User      => 'client-user-123', # or in other words

# CustomerID      => 'client123',

# CustomerUserID => 'client-user-123',

},

},

};

```

Name задает имя конфигурируемого Действия Перехода.

No или CustomerID задает Customer ID клиента.

User или CustomerUserID задает имя пользователя-клиента.

5. TicketLockSet

Устанавливает состояние блокирования процессной заявки. Пример:

```

$Self->{'Process::TransitionAction'} = { 'TA1' => {

Name      => 'Set Lock to lock',

Module => 'Kernel::System::ProcessManagement::TransitionAction::TicketLockSet',
Config => {

Lock      => 'lock', # or

LockID => 2,

},

},

};

```

Name задает имя конфигурируемого Действия Перехода.

Lock/Блокировка - задает новый статус блокировки для процессной заявки.

LockID задает внутренний ID нового статуса блокировки.

6. TicketOwnerSet

Изменяет владельца этой процессной заявки. Пример:

```

$Self->{'Process::TransitionAction'} = { 'TA1' => {
Name      => 'Owner Set root@localhost',

Module => 'Kernel::System::ProcessManagement::TransitionAction::TicketOwnerSet',
Config => {

Owner => 'root@localhost', # or

OwnerID => 1,

},

},

};

```

Name задает имя конфигурируемого Действия Перехода.

Owner/Владелец задает логин нового владельца.

OwnerID задает внутренний ID нового владельца.

7. TicketQueueSet

Перемещает заявку в указанную очередь. Пример:

```

$Self->{'Process::TransitionAction'} = { 'TA1' => {

Name      => 'Queue Move Raw',

Module                                     =>
'Kernel::System::ProcessManagement::TransitionAction::TicketQueueSet', Config =>
{

Queue => 'Raw', # or

# QueueID => '2',

},

},

};

```

Name задает имя конфигурируемого Действия Перехода.

Queue задает имя очереди для перемещения.

QueueID задает внутренний ID для очереди для перемещения.

8. TicketResponsibleSet

Изменяет владельца процессной заявки. Пример:

```
$Self->{'Process::TransitionAction'} = { 'TA1' => {  
    Name      => 'Responsible Set root@localhost', Module =>  
    'Kernel::System::ProcessManagement::TransitionAction::TicketResponsibleSet',  
    Config => {  
        Responsible => 'root@localhost', # or  
        ResponsibleID => 1,  
    },  
},  
};
```

Name задает имя конфигурируемого Действия Перехода.

Responsible задает логин нового ответственного.

ResponsibleID задает внутренний ID нового ответственного.

9. TicketServiceSet

Назначает сервис процессной заявке. Заявка должна иметь клиента и сервис должен быть назначен этому клиенту. Пример:

```
$Self->{'Process::TransitionAction'} = { 'TA1' => {  
    Name      => 'Set MyService service',  
    Module                                         =>  
    'Kernel::System::ProcessManagement::TransitionAction::TicketServiceSet', Config =>  
    {  
        Service      => 'MyService', # or  
        ServiceID => 123,  
    },  
},  
};
```

Name задает имя конфигурируемого Действия Перехода.

Service задает новый сервис для процессной заявки. Требуется задать полное имя сервиса (например, GrandFatherService::FatherService::SonService).

ServiceID задает внутренний ID нового сервиса.

10. TicketSLASet

Задаёт SLA для процессной заявки. Заявке должен быть уже назначен сервис и SLA должен быть привязан к этому сервису. Пример:

```
$Self->{'Process::TransitionAction'} = { 'TA1' => {  
  
  Name      => 'Set MySLA SLA',  
  
  Module => 'Kernel::System::ProcessManagement::TransitionAction::TicketSLASet',  
  Config => {  
  
    SLA      => 'MySLA',  
  
    # or  
  
    SLAID => 123,  
  
  },  
  
  },  
  
};
```

Name задает имя конфигурируемого Действия Перехода.

SLA задает новое соглашение об уровне сервиса для процессной заявки.

SLAID задает внутренний ID нового SLA.

11. TicketStateSet

Изменяет состояние процессной заявки. Пример:

```
$Self->{'Process::TransitionAction'} = { 'TA1' => {  
  
  Name      => 'Set State to open',  
  
  Module => 'Kernel::System::ProcessManagement::TransitionAction::TicketStateSet',  
  Config => {  
  
    State      => 'open', # or  
  
    StateID => 4,  
  
  },  
  
};
```

```
PendingTimeDiff => 123,  
  
},  
  
},  
  
};
```

Name задает имя конфигулируемого Действия Перехода.

State задает новое состояние процессной заявки.

StateID задает внутренний ID нового состояния.

PendingTimeDiff используется только для состояний с ожиданием, задает относительную разность в секундах (относительно заданного времени выполнения Действия Перехода) для установки времени ожидания для заявки (например, 3600 означает что устанавливается ожидание в 1 час после заданного времени выполнения Действия Перехода).

12. TicketTitleSet

Задает Тему процессной заявки. Пример:

```
$Self->{'Process::TransitionAction'} = { 'TA1' => {  
  
  Name      => 'Set Ticket Title to Ticket-title',  
  
  Module => 'Kernel::System::ProcessManagement::TransitionAction::TicketTitleSet',  
  Config => {  
  
    Title => 'Ticket-title',  
  
  },  
  
  },  
  
};
```

Name задает имя конфигулируемого Действия Перехода.

Title задает новую тему заявки.

13. TicketTypeSet

Задает Тип процессной заявки. Пример:

```
$Self->{'Process::TransitionAction'} = { 'TA1' => {  
  
  Name      => 'Set Ticket Type to default',
```

```

Module => 'Kernel::System::ProcessManagement::TransitionAction::TicketTypeSet',
Config => {

Type      => 'default', # or

#TypeID => '1',

},

},

};

```

4.6. Списки Контроля Доступа (ACLs)

С помощью ACL, вы можете ограничить выбор значений в процессной заявке. См. также справочник ACL для описания полного синтаксиса ACL для заявок.

1. Настройка ACL

ACL могут быть заданы только в Kernel/Config.pm (Правильнее с помощью редактора ACL в SysConfig). Пример:

```

$Self->{TicketAcI}->{'001-ACL-ProcessProperties'} = { Properties => {

Process => {

ProcessEntityID           => ['P1'],

ActivityEntityID          => ['A1'], ActivityDialogEntityID => ['AD1'],

}

},

Possible => {

ActivityDialog => ['AD1', 'AD3'],

},

PossibleNot => { ActivityDialog => ['AD3'],

},

};

```

2. 001-ACL-ProcessProperties

Имя правила ACL. Для более полной информации обратитесь к руководству ACL.

3. Process

Это раздел, который используется для проверки того, должен ли быть применен ACL. Если параметр имеет заданное значение, правило применяется. Могут быть использованы следующие значения:

1. ProcessEntityID

ID процесса для текущего процесса. Совпадает, если заявка назначена этому процессу.

2. ActivityEntityID

ID Активности, которой назначена процессная заявка в текущий момент.

3. ActivityDialogEntityID

ID Диалога Активности, открытого в настоящий момент для процессной заявки.

4. Possible and PossibleNot Activity Dialog/Диалога Активности

Здесь вы можете задать список идентификаторов Диалогов Активности. Этот список ограничивает Диалоги Активности, которые могут быть предложены пользователю при просмотре заявки.

Possible показывает список допустимых Диалогов Активности. Установка выше, допускает только 'AD1' и 'AD3' из списка заданных Диалогов Активности.

PossibleNot отображает список неразрешенных Диалогов Активности. В примере выше, настройка удаляет 'AD3' из списка заданных Диалогов Активности.

Если оба параметра Possible и PossibleNot заданы, список заданных Диалогов Активности сначала фильтруется, используя Possible, оставляя 'AD1' и 'AD3' в нашем примере. Затем применяется фильтр PossibleNot, который удаляет 'AD3', оставляя таким образом только 'AD1', как доступный пользователю для использования Диалог Активности.

Если совпадают несколько правил ACL, пересечение всех соответствующих правил будет рассчитано для определения возможных Диалогов Активности. Пример:

Настроенные Диалоги Активности: 'AD1', 'AD2', 'AD3', 'AD4', 'AD5', 'AD6', 'AD7'.

```
$Self->{TicketAcl}->{'001-ACL-Status'} = { Properties => {  
  
    Ticket => {  
  
        Status => 'new',  
  
    }  
  
    },  
  
    Possible => {
```

```

ActivityDialog => ['AD1', 'AD2', 'AD3', 'AD6', 'AD7'],
},
};

$Self->{TicketAcl}->{'002-ACL-Queue'} = { Properties => {
Ticket => {
Queue => ['Raw']
}
},
Possible => {
ActivityDialog => ['AD2', 'AD3', 'AD4', 'AD7'],
},
};

$Self->{TicketAcl}->{'003-ACL-Priority'} = { Properties => {
Ticket => {
Priority => ['3 normal']
}
},
PossibleNot => {
ActivityDialog => ['AD3', 'AD4'],
},
};

```

Если процессная заявка имеет состояние new, находится в очереди 'Raw' и имеет приоритет '3 normal', то все три правила ACL будут совпадать.

Первое правило сокращает список Диалогов Активности с 'AD1', 'AD2', 'AD3', 'AD4', 'AD5', 'AD6', 'AD7' до 'AD1', 'AD2', 'AD3', 'AD6', 'AD7' и запрещает 'AD4' and 'AD5'.

Второе правило еще сокращает список оставшихся Диалогов Активности. В нашем примере, остаются 'AD2', 'AD3', 'AD7'.

Третье правило еще более уменьшает список с помощью PossibleNot. 'AD3' удаляется из списка. 'AD4' не будет удален, т.к. он не на первом месте в списке. И, наконец, 'AD2' и 'AD7' останутся, как доступные для выбора пользователем Диалоги Активности.

Также, можно ограничить список Процессов, которые будут отображаться на экране "Новая процессная заявка", способ аналогичен ограничению списка Диалогов Активности, за одним исключением: ACL должны базироваться на данных пользователя. То есть, список ACL привязывается к пользователю, создающему заявку.

Смотрите примеры ниже:

```
$Self->{TicketAcl}->{'200-ACL-Process'} = { # match properties  
  
Properties => { User => {  
  UserID => [2, 3],  
},  
},  
  
Possible => {  
  Process => ['P1', 'P2', 'P3'],  
},  
  
PossibleNot => { Process => ['P4'],  
},  
};
```

```
$Self->{TicketAcl}->{'201-ACL-Process'} = { # match properties  
  
Properties => { User => {  
  Group_rw => ['MyGroup'],  
},  
},  
  
Possible => {  
  Process => ['P1', 'P2', 'P3'],  
}
```

```
},  
PossibleNot => { Process => ['P4'],  
},  
};
```

```
$Self->{TicketAcl}->{'202-ACL-Process'} = { # match properties  
Properties => { User => {  
Role => [ 'MyRole' ],  
},  
},  
Possible => {  
Process => ['P1', 'P2', 'P3'],  
},  
PossibleNot => { Process => ['P4'],  
},  
};
```

6. Настройка производительности

Ниже представлен список методов повышения производительности для вашей установки RSD, включая настройку, кодирование, использование памяти и многое другое.

1. RSD

Существует несколько вариантов повышения производительности RSD.

1. TicketIndexModule

Есть два типа модулей для индексирования заявок в обзоре очередей:

Kernel::System::Ticket::IndexAccelerator::RuntimeDB

Это опция по умолчанию, и она будет генерировать каждый обзор очереди "на лету" из таблицы заявок. У вас не будет проблем с производительностью, пока в вашей системе не будет около 60 000 открытых заявок.

Kernel::System::Ticket::IndexAccelerator::StaticDB

Самый мощный модуль, его следует использовать, когда у вас более 80 000 открытых заявок. Он использует дополнительную таблицу `ticket_index`, которая будет заполнена ключевыми словами на основе данных заявок. Используйте `bin/radiant.Console.pl Maint::Ticket::QueueIndexRebuild` для генерации начального индекса после переключения бэкендов.

Вы можете изменить используемый модуль `IndexAccelerator` через `SysConfig`.

2. Индекс поиска заявок

RSD использует специальный поисковый индекс для выполнения полнотекстового поиска по полям в заметках из разных каналов связи.

Чтобы создать начальный индекс, используйте `bin/radiant.Console.pl Maint::Ticket::FulltextIndex`

`--rebuild`.

Примечание

Фактическое индексирование заметок происходит с помощью задания демона RSD в фоновом режиме. В то время как заметки, которые только что были добавлены в систему, помечаются для индексации немедленно, может случиться так, что их индекс будет доступен в течение нескольких минут.

Есть несколько доступных опций для точной настройки поискового индекса:

`Ticket::SearchIndex::IndexArchivedTickets`

Определяет, будут ли архивированные заявки включены в индекс поиска (по умолчанию отключено). Желательно, чтобы индекс был небольшим в больших системах с архивированными заявками. Если этот параметр отключен, архивированные заявки не будут найдены при полнотекстовом поиске.

`Ticket::SearchIndex::Attribute`

Атрибут `WordCountMax` определяет максимальное количество слов, которые будут обработаны для создания индекса. Например, в индексе поиска заметок хранятся только первые 1000 слов основного текста заметки. Атрибуты `WordLengthMin` и `WordLengthMax` используются в качестве границ длины слова. В индексе поиска заметок сохраняются только слова с длиной между этими двумя значениями.

`Ticket::SearchIndex::Filters`

Определены три фильтра по умолчанию:

- Первый фильтр удаляет специальные символы, такие как: `, & < > ? " ! * | ; [] () + $ ^ =`

- Второй фильтр удаляет слова, которые начинаются или заканчиваются одним из следующих символов: ' : .
- Третий фильтр удаляет слова, которые не содержат символы: a-z, A-Z, 0-9, _

`Ticket::SearchIndex::StopWords`

Для некоторых языков определены так называемые стоп-слова. Эти стоп-слова будут пропущены при создании поискового индекса.

3. Хранение заметок (электронная почта, телефон и внутренние заметки)

Существует два различных внутренних модуля для хранения телефонных, электронных и внутренних заметок (настраивается с помощью `Ticket::Article::Backend::MIMEBase::ArticleStorage`):

`Kernel::System::Ticket::Article::Backend::MIMEBase::ArticleStorageDB`

Этот модуль по умолчанию будет хранить вложения в базе данных.

Примечание

Не используйте его с большими установками.

Плюсы: работает с несколькими интерфейсными серверами.

Минусы: требует много места для хранения в базе данных.

`Kernel::System::Ticket::Article::Backend::MIMEBase::ArticleStorageFS`

Используйте этот модуль для хранения вложений в локальной файловой системе.

Примечание

Рекомендуется для больших установок.

Плюсы: это быстро!

Минусы: если у вас несколько интерфейсных серверов, то вы должны убедиться, что файловая система является общей для всех серверов. Поместите его в общий ресурс NFS или, что предпочтительнее, в SAN или аналогичное решение.

Примечание

Вы можете быстро переключаться с одного бэкенда на другой. Вы можете переключить серверную часть в `SysConfig`, а затем запустить утилиту командной строки `bin/radiant.Console.pl Admin::Article::StorageSwitch`, чтобы поместить заметки из базы данных в файловую систему или наоборот. Вы можете использовать параметр `--target`, чтобы указать целевой сервер. Обратите внимание, что весь процесс может занять значительное время, в зависимости от количества имеющихся у вас заметок и доступной мощности процессора и/или пропускной способности сети.

```
shell> bin/radiant.Console.pl Admin::Article::StorageSwitch --target ArticleStorageFS
```

Сценарий: Переключение серверной части хранилища с базы данных на файловую систему.

Если вы хотите сохранить старые вложения в базе данных, вы можете активировать опцию SysConfig Ticket::Article::Backend::MIMEBase::CheckAllStorageBackends, чтобы убедиться, что RSD все еще найдет их.

4. Архивирование заявок

Поскольку RSD можно использовать в качестве системы проверки подлинности, удаление закрытых заявок может оказаться не очень хорошей идеей. Поэтому мы внедрили функцию, которая позволяет вам архивировать заявки.

Заявки, соответствующие определенным критериям, могут быть помечены как "архивированные". Доступ к этим заявкам невозможен, если вы выполняете обычный поиск заявок или выполняете обычное задание агента. Самой системе больше не нужно иметь дело с огромным количеством заявок, поскольку при использовании RSD учитываются только "последние" заявки. Это может привести к огромному увеличению производительности в больших системах.

Чтобы воспользоваться функцией архивирования, просто выполните следующие действия:

1. Активируйте архивную систему в SysConfig

На странице администратора перейдите в SysConfig и выберите группу Ticket. В Core::Ticket вы найдете опцию Ticket::ArchiveSystem, для которой по умолчанию установлено значение "нет". Измените этот параметр на "да" и сохраните это изменение.

2. Определите задание GenericAgent

На странице администратора выберите GenericAgent и добавьте туда новое задание.

a. Настройки задания

Укажите имя задания архивации и выберите соответствующие параметры для планирования этого задания.

b. Фильтр заявок

Фильтр заявок - это поиск заявок, соответствующих выбранным критериям. Возможно, было бы хорошей идеей архивировать только те заявки в закрытом состоянии, которые были закрыты несколькими месяцами ранее.

c. Действие заявок

В этом разделе установите для поля с надписью "Архивировать выбранные заявки" значение "архивировать заявки".

d. Сохраните задание

В конце страницы вы найдете опцию сохранения задания.

e. Затронутые заявки

Система отобразит все заявки, которые будут заархивированы при выполнении задания универсального агента.

3. Поиск заявок

При поиске заявок система по умолчанию выполняет поиск заявок, которые не заархивированы. Если вы также хотите выполнять поиск по архивным заявкам, просто добавьте "поиск по архиву" при определении критериев поиска.

5. Кэш

RSD кэширует много временных данных в /opt/radiant/var/tmp. Пожалуйста, убедитесь, что при этом используется высокопроизводительная файловая система / хранилище. Если у вас достаточно оперативной памяти, вы также можете попробовать поместить этот каталог на виртуальный диск следующим образом:

```
shell> /opt/radiant/bin/radiant.Console.pl Maint::Session::DeleteAll shell>
/opt/radiant/bin/radiant.Console.pl Maint::Cache::Delete shell> sudo mount -o
size=16G -t tmpfs none /opt/radiant/var/tmp

# добавить постоянную точку монтирования в /etc/fstab
```

Примечание

Обратите внимание, что это будет непостоянное хранилище, которое будет потеряно при перезагрузке сервера. Все ваши сеансы (если вы храните их в файловой системе) и данные вашего кеша будут потеряны.

Существует также централизованный кеш-сервер на основе memcached, который можно приобрести у Radiant System.

2. База данных

Проблемы с БД зависят от используемой базы данных. Изучите документацию к вашей базе данных или проконсультируйтесь с вашим администратором базы данных.

1. MySQL

Если для MySQL-таблиц используется MyISAM (который используется по умолчанию) и из таблицы была удалена большая часть данных, или же производилось множество действий над таблицей, которая содержит строки переменной длины (таблицы, которые содержат поля типов VARCHAR, BLOB или TEXT), то нужно дефрагментировать файл (таблицу), или другими словами выполнить команду "оптимизировать".

Нужно попытаться сделать это, если сервер `mysqld` использует много процессорного времени. Оптимизация таблиц `ticket_history` и `article` (см Сценарий ниже).

```
shell> mysql -u user -p database mysql> optimize table ticket;

mysql> optimize table ticket_history; mysql> optimize table article;
```

Сценарий: Оптимизация таблиц базы данных.

2. PostgreSQL

PostgreSQL лучше всего настроить, изменив файл `postgresql.conf` в вашем каталоге данных PostgreSQL.

Если производительность по-прежнему неудовлетворительна, мы рекомендуем вам присоединиться к списку рассылки PostgreSQL Performance (<http://www.postgresql.org/community/lists/>), и задавайте там вопросы. Люди из списка PostgreSQL очень дружелюбны и, вероятно, смогут помочь.

3. Веб-сервер

Конечно, вы должны использовать `mod_perl 2.0` (<http://perl.apache.org/>). Это намного быстрее (~*100), чем чистый CGI, но для этого требуется больше оперативной памяти.

1. Предустановленное соединение с базой данных

Вы можете предварительно установить подключения к базе данных при запуске веб-сервера. Это экономит время (см. `README.webserver`).

2. Предварительно загруженные модули - `startup.pl`

Используйте сценарий запуска `scripts/apache2-perl-startup.pl` чтобы предварительно загруженные/скомпилированные модули Perl на вашем веб-сервере `mod_perl` были быстрее и занимали меньше памяти (см. `README.webserver`).

3. Перегрузка Perl-модулей во время обновления с диска

По умолчанию `Apache::Reload` используется в `scripts/apache2-httpd.include.conf`. Отключите его, и вы получите на 8% больше скорости. Но не забудьте перезапустить веб-сервер, если вы устанавливаете какие-либо модули через диспетчер пакетов RSD или какие-либо значения в вашей `SysConfig` или в `Kernel/Config.pm`.

4. `mod_gzip/mod_deflate`

Если ваша пропускная способность невелика, используйте `mod_deflate` для Apache2. Если у вас есть html-страница размером 45 кб, `mod_gzip/mod_deflate` сжимает ее примерно до 7 кб. Недостатком является то, что это увеличивает нагрузку на серверную часть.