



**Ред Платформа**

**Руководство пользователя**

**46.98898398.506180-01 34 1-1**

Версия 1.6

Москва, 2019

## Перечень сокращений

<b>БД</b>	База данных
<b>ИБ</b>	Информационная безопасность
<b>ОС</b>	Операционная система
<b>ПК</b>	Персональный компьютер
<b>ПО</b>	Программное обеспечение
<b>РП</b>	Ред Платформа
<b>СУБД</b>	Система управления базами данных

## Оглавление

1. Назначение программы .....	6
1.1. Функциональное назначение программы .....	6
1.2. Эксплуатационное назначение программы .....	6
1.3. Состав функций.....	6
2. Условия выполнения программы .....	7
2.1. Минимальный состав аппаратных средств.....	7
2.2. Минимальный состав программных средств .....	7
2.3. Требования к персоналу (пользователю) .....	7
2.4. Предположения о поведении пользователя, связанные с безопасной эксплуатацией РП.....	7
3. Установка и запуск.....	9
3.1. Установка СУБД «Ред База Данных» .....	9
3.1.1. Загрузка дистрибутива СУБД .....	9
3.1.2. Требования к среде исполнения.....	9
3.1.3. Установка СУБД «Ред База Данных» в ОС семейства Windows .....	9
3.1.4. Установка СУБД «Ред База Данных» в ОС семейства Linux.....	10
3.2. Установка Java.....	10
3.3. Установка Apache Tomcat.....	11
3.3.1. Установка и запуск Apache Tomcat на Windows .....	11
3.3.2. Установка и запуск Apache Tomcat на Linux.....	11
3.4. Установка Платформы.....	11
4. Выполнение программы .....	13
4.1. Обозначение элементов управления.....	13
4.2. Загрузка и запуск программы .....	14
4.3. Описание операций .....	15
4.3.1. Подменю «Окна» .....	16
4.3.2. Подменю «Системные объекты» .....	16
4.3.3. «Профиль пользователя» .....	26
4.4. Завершение работы программы .....	27
5. Сообщения пользователю .....	29
6. ncore-api .....	31
6.1. Подключение к проекту .....	31
6.2. Виды API.....	32
6.3. Endpoint, или точка входа .....	32
REST.....	33

JSON-API.....	33
6.4. Ресурсы .....	33
6.5. Резолверы ресурсов .....	33
6.6. Структура URL .....	34
6.7. Авторизация и аутентификация. ApiApplicationManager .....	35
6.8. ApiProcessContext .....	36
6.9. Описание NCORE REST API.....	36
6.9.1. Виды ресурсов REST .....	36
RestDataObjectResource .....	37
RestAliasResource .....	39
6.9.2. Системные ресурсы REST .....	39
@actions .....	39
@reload.....	40

## Введение

В данном документе приведено руководство оператора по применению и эксплуатации Ред Платформы (далее – РП, платформа), предназначенной для автоматизации деловых процессов документооборота потребителей продукта.

В данном документе, в разделе «Назначение программы» указаны сведения о назначении программы и информация, достаточная для понимания функций программы и ее эксплуатации.

В разделе «Условия выполнения программы» указаны условия, необходимые для выполнения программы (минимальный состав аппаратных и программных средств и т.п.).

В разделе «Выполнение программы» указана последовательность действий оператора, обеспечивающих загрузку, запуск, выполнение и завершение программы, приведено описание функций, формата и возможных вариантов операций, с помощью которых оператор осуществляет выполнение программы, а также ответы программы на эти действия.

В разделе «Сообщения оператору» приведены тексты сообщений, выдаваемых в ходе выполнения программы, описание их содержания и соответствующие действия оператора.

# **1. Назначение программы**

## **1.1. Функциональное назначение программы**

Платформа документооборота предназначена для управления документами и другими типами контента, их хранения и обработки в масштабах предприятия. Платформа поддерживает единый жизненный цикл информации различного типа и формата. Платформа соответствует принятым в мировой практике промышленным стандартам качества, надежности и безопасности.

## **1.2. Эксплуатационное назначение программы**

РП является удобной и функциональной средой для ведения электронного документооборота, позволяющая пользователям создавать, изменять и хранить необходимые сведения. РП обеспечивает требуемый уровень безопасности, предъявляемый к системам такого рода. Технологическая платформа реализует управление жизненным циклом документов, доступом к ним, управление пользователями и их ролями. Имеется встроенная подсистема журналирования событий и мониторинга инцидентов ИБ.

## **1.3. Состав функций**

Состав функций РП, доступных конкретному пользователю, определяется объемом полномочий, предоставленных ему администратором в рамках его роли. В общем случае это следующие функции:

- просмотр настроек системы;
- просмотр (редактирование) размеров экранных форм;
- просмотр задач планировщика;
- просмотр отчетов: шаблонов отчетов и печатных форм;
- просмотр обновлений: их порядок и содержание;
- просмотр (редактирование) документооборота: классы, статусы и их настройка, работа с документами;
- просмотр (редактирование) обмена данными.

## **2. Условия выполнения программы**

### **2.1. Минимальный состав аппаратных средств**

Для нормального функционирования РП необходим персональный компьютер в следующей минимальной конфигурации:

- процессор: Pentium IV 1.5GHz и выше;
- память: 1024 Мб и более;
- накопитель: 80 Гб;
- номинальная скорость локальной вычислительной сети: 100 Мбит/сек;
- принтер (если пользователь должен осуществлять печать документов);
- сканер штрих-кодов (если это необходимо).

На рабочем месте пользователя должно быть настроено подключение к сети по протоколу TCP/IP, если хранение БД происходит на отдельном сервере, а не на этом же ПК.

### **2.2. Минимальный состав программных средств**

Запуск РП возможен в двух видах архитектуры: «толстый клиент» и «тонкий клиент». От выбранного типа запуска зависит минимальный состав программных средств.

Для «тонкого клиента» необходимы:

- любая ОС семейства Windows или Linux, имеющая веб-браузер с поддержкой Javascript.

Для «толстого клиента» необходимы:

- любая ОС семейства Windows или Linux, поддерживающая установку;
- JRE версии 8 для ОС семейства Windows;
- OpenJDK 1.8 для ОС семейства Linux.

### **2.3. Требования к персоналу (пользователю)**

Пользователь РП должен иметь минимальные навыки работы с ОС семейства Windows или Linux (в зависимости от выбранной архитектуры), опыт работы со стандартными элементами графического интерфейса приложений.

### **2.4. Предположения о поведении пользователя, связанные с безопасной эксплуатацией РП**

Предполагается, что у пользователя на компьютере с установленным РП, отсутствуют нештатные программные средства, позволяющие осуществить несанкционированную модификацию РП.

Эксплуатация РП осуществляется пользователем согласно руководства

пользователя РП.

В своей деятельности пользователь должен руководствоваться политикой безопасности организации и не нарушать организационно-технических мер, действующих в организации.



### 3. Установка и запуск

Для корректной работы платформы необходимо установить:

- Систему управления базами данных «Ред База Данных», включенную в Единый реестр российских программ для электронных вычислительных машин и баз данных ( <https://reestr.minsvyaz.ru/reestr/58191/> );
- Apache Tomcat 8.5 (<http://tomcat.apache.org/>);
- Java SE 8 (OpenJdk 8);
- Интернет-браузер (Internet Explorer, Mozilla Firefox, Opera, Chrome и т.д.).

#### 3.1. Установка СУБД «Ред База Данных»

##### 3.1.1. Загрузка дистрибутива СУБД

Произвести загрузку СУБД для соответствующей платформы с общедоступного ресурса производителя по адресу:

или из прилагаемого дистрибутива:  
/dist/RedDatabase.

##### 3.1.2. Требования к среде исполнения

Функционирование СУБД возможно на операционных системах семейства Microsoft Windows 7 и новее, а также на операционных системах семейства Linux, BSD Unix, IBM AIX, HP-UX, Sun Solaris и других.

##### 3.1.3. Установка СУБД «Ред База Данных» в ОС семейства Windows

Перед установкой СУБД «Ред База Данных» необходимо определить разрядность используемой операционной системы и на каком языке будет отображаться процесс установки (предусмотрена установка на русском и английском языках). После этого необходимо выбрать соответствующий инсталлятор (win32 или win64, en или ru) и запустить установку СУБД «Ред База Данных» (например, rdb30-windows-x86-3.0.x.x.exe инсталлятор win32, rdb30-windows-x86\_64-3.0.x.x.exe инсталлятор win64).

Инсталляция СУБД «Ред База Данных» осуществляется с помощью стандартного для Windows мастера установки программ. В ходе установки мастер собирает всю необходимую для установки СУБД информацию, производит копирование файлов и регистрацию программных модулей в реестре Windows. Для установки СУБД «Ред База Данных» необходимы права администратора. Для правильного функционирования платформы выберите

Обычный тип установки и далее все параметры по умолчанию. После завершения установки СУБД запустится автоматически, перезагрузка не требуется.

#### 3.1.4. Установка СУБД «Ред База Данных» в ОС семейства Linux

Дистрибутив СУБД «Ред База Данных» поставляется в виде бинарного пакета. Установка СУБД из бинарного пакета возможна как в текстовом, так и в графическом режиме. Для установки необходимо назначить файлу дистрибутива права на выполнение и запустить с правами администратора.

При запуске его из любой графической системы (например KDE) будет вызван мастер установки, который аналогично установке под Windows произведет сбор всей необходимой информации и установит СУБД «Ред База Данных» на Ваш компьютер. Для функционирования платформы выберите обычную установку и далее все параметры по умолчанию.

Подробнее процесс установки СУБД «Ред База данных» описан в гл. 2 Руководства администратора СУБД «Ред База данных» (<https://reddatabase.ru/documentation/>)

Внимание! При установке СУБД «Ред База данных» в архитектуре Classic убедитесь что служба xinetd установлена и работает. После установки СУБД необходимо осуществить перезапуск этой службы.

### 3.2. Установка Java

Платформа функционирует на Java версии 8 и новее. Для запуска потребуется установить Oracle JRE 8, либо OpenJDK 8 или новее. Дистрибутивы Oracle JRE можно бесплатно скачать с сайта Oracle по адресу:

либо взять прилагаемый дистрибутив Oracle JRE 8 для Windows:

/dist/Java.

В процессе установки Вам необходимо указать каталог для установки.

Для установки Java на Linux можно также воспользоваться дистрибутивом Oracle JRE (JDK) либо установить OpenJDK. Установка OpenJDK осуществляется из соответствующего репозитория операционной системы. Пакет как правило носит имя `openjdk-8-jdk` или похожее. Подробную информацию об установке OpenJDK можно получить на сайте .

### 3.3. Установка Apache Tomcat

Для функционирования платформы и запуска тонкого клиента, а также сервера, необходима установка сервера Apache Tomcat 8.5. Дистрибутивы можно скачать по адресу:

Либо из прилагаемого дистрибутива для Windows:

`/dist/Tomcat.`

#### 3.3.1. Установка и запуск Apache Tomcat на Windows

Установка производится в графическом режиме, все параметры необходимо выбирать по умолчанию. Перед установкой Tomcat необходимо убедиться, что установлена Java. В качестве каталога для установки лучше выбрать `C:\tomcat\`, но это необязательно. По умолчанию Tomcat устанавливается в качестве сервиса, но возможен также запуск и останов при помощи командной строки специальными командными файлами `startup.bat` и `shutdown.bat`, располагаемыми в каталоге `C:\tomcat\bin`.

#### 3.3.2. Установка и запуск Apache Tomcat на Linux

Для установки Tomcat на Linux необходимо распаковать архив с дистрибутивом Apache Tomcat в выбранный каталог (например, `/opt/tomcat`).

Запуск и останов сервера Tomcat осуществляются командными файлами `startup.sh` и `shutdown.sh`, которые располагаются в подкаталоге `bin` каталога установки сервера приложений Apache Tomcat.

### 3.4. Установка Платформы

Перед установкой Платформы убедитесь, что установлены Java, СУБД «Ред База Данных» и Apache Tomcat.

Общий порядок установки следующий:

- распаковать архив `ncore-1.6.x.x.zip`;
- скопировать файл БД `ncore-1.6.x.x.fdb` из каталога `/ncore-1.6.x.x/db/` на жесткий диск (например, в каталог `/opt/db/`);

– настроить пользователю, от имени которого запускается СУБД «Ред База Данных» (на Linux это, как правило, `firebird`) необходимые права (чтение и запись) на скопированную БД:

- в каталоге установки СУБД в файле `aliases.conf` указать путь к базе данных (имя алиаса `ncore.head.dev`). Например:

```
ncore.head.dev = /opt/db/ncore-1.6.x.x.fdb
```

– разместить пакет с сервером Платформы ncore-1.6.x.x.war - в каталог webapps сервера приложений Apache Tomcat (/opt/tomcat/webapps/) и запустить сервер Tomcat. В процессе запуска сервера автоматически будет распакован war-файл и создан соответствующий подкаталог в каталоге webapps, содержащий все необходимые файлы для тонкого клиента. После запуска сервера при помощи браузера операционной системы зайти по URL <http://localhost:8080/ncore-1.6.x.x/client>.

Запуск толстого клиента производится путем выполнения командного файла startclient.bat (для Windows систем) или startclient.sh (для Linux систем) из каталога платформы /ncore-1.6.x.x/bin/.











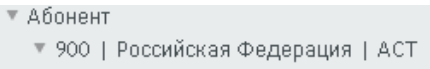
При запросе авторизации при первичном запуске необходимо ввести в поле «Имя пользователя» значение sysdba, а в поле «Пароль» значение masterkey.





## 4. Выполнение программы

### 4.1. Обозначение элементов управления

В таблице 1 представлены термины для обозначения стандартных элементов графического интерфейса приложений.

Таблица 1 – Элементы интерфейса и их описание

Элемент интерфейса	Термин	Примечание
	Кнопка	
	Кнопка вызова справочника	Кнопка вызова справочника
	Кнопка вызова календаря	Кнопка вызова системного календаря
	Экспорт списковой области	Экспорт списковой области формы в табличный редактор
Код: <input type="text" value=""/> *	Поле ввода	
	Раскрывающийся список	
*	Обязательный реквизит	Обязательный реквизит для заполнения
	Кнопка обновления записи	Кнопка обновления записи или списковой записи
	Кнопка создания записи	Кнопка создания новой записи
	Кнопка создания записи с копированием	Кнопка создания записи с копированием
	Кнопка редактирования записи	Кнопка редактирования записи
	Кнопка удаления записи	Кнопка удаления записи
	Кнопка открытия / скрытия панели фильтрации	Кнопка открытия / скрытия панели фильтрации
	Кнопка очистки фильтров	Кнопка очистки панели фильтрации от введенных данных
	Иерархический список	Иерархический раскрывающийся список
<input checked="" type="checkbox"/>	Checkbox	Графический элемент

Элемент интерфейса	Термин	Примечание
 		выбора, может принимать значения:  - активен,  - не определено,  - значение не выбрано

## 4.2. Загрузка и запуск программы

Доступ к функциональным возможностям РП открыт только для зарегистрированных пользователей. Регистрацию выполняет уполномоченный администратор, который назначает пользователю права доступа, роли и сообщает идентификационные данные: учетную запись пользователя и пароль, с которыми он будет регистрироваться в системе.

Для начала работы с РП (архитектура «тонкий клиент») пользователю необходимо открыть браузер и в адресной строке набрать:

`http://ip_адрес_сервера:порт_подключения/ncore/client`

Для начала работы с РП (архитектура «толстый клиент») пользователю необходимо дважды щелкнуть по ярлыку на рабочем столе или выполнить запуск командного файла:

- для ОС Windows: `X:\ncore\client\start-client.bat`
- для ОС Linux: `X:\ncore\client\start-client.sh`

где `X:\` - путь к каталогу, хранящему ЕСМ документооборота.

В появившемся окне входа (рисунок 1) пользователю предлагается ввести регистрационные данные: логин и пароль. Кнопка «Войти» доступна только после того как пользователь ознакомится с мерами информационной безопасности и подтвердит это, поставив галочку в соответствующем поле. В результате будет обеспечен доступ к функциональным возможностям РП в соответствии с полномочиями пользователя, запустившего сеанс работы.

После успешной регистрации в тонком клиенте можно навести курсор мыши на заголовок вкладки браузера, с открытым клиентом РП. Во всплывающем окне при наведении курсора мыши отображается помимо имени пользователя дата и время последнего входа. В толстом клиенте информация о дате и времени последнего входа отображается в заголовке окна.

Также в заголовке окна отображается число неуспешных попыток входа в систему с последнего успешного входа. Для просмотра списка попыток неуспешного входа необходимо перейти в меню «Профиль пользователя» - «Неуспешные попытки входа».

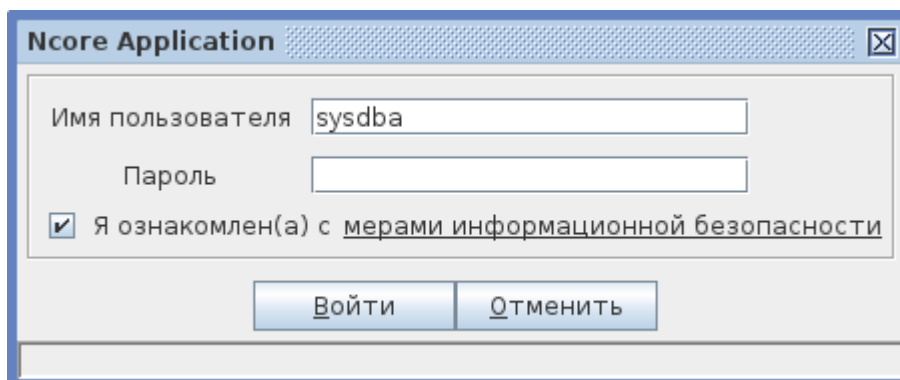


Рисунок 1 – Вход в систему

### 4.3. Описание операций

После успешной регистрации пользователь попадает в главное меню программы (рисунок 2), где, используя кнопки навигации, может выбрать интересующее меню и однократным нажатием кнопки мыши открыть окно подменю.



Рисунок 2 – Главное меню РП

В тонком клиенте, если пользователь не предпринимает каких-либо действий в течении определенного периода времени, то происходит принудительный разрыв сессии и после этого при обращении к какому-либо пункту меню выдается предупреждающее сообщение: «Сессия истекла. Примите к сведению все несохраненные данные, и нажмите здесь для продолжения».

В толстом клиенте по истечении заданного времени выводится предупреждающее сообщение: «Внимание! Время до блокировки сессии истекает в ...». Если в это время произвести какие-либо действия с РП, окно будет закрыто, можно продолжать работать. Если оставить РП без действий, то

произойдет разрыв сессии, все открытые окна будут закрыты, пользователю будет показано окно входа в РП.

#### 4.3.1. Подменю «Окна»

В данном подменю пользователь может выполнить стандартные операции с окнами: свернуть все окна приложения; закрыть все открытые окна; упорядочить окна или восстановить их исходный размер или узнать информацию о версиях используемых компонентов подсистемы (рисунок 3).

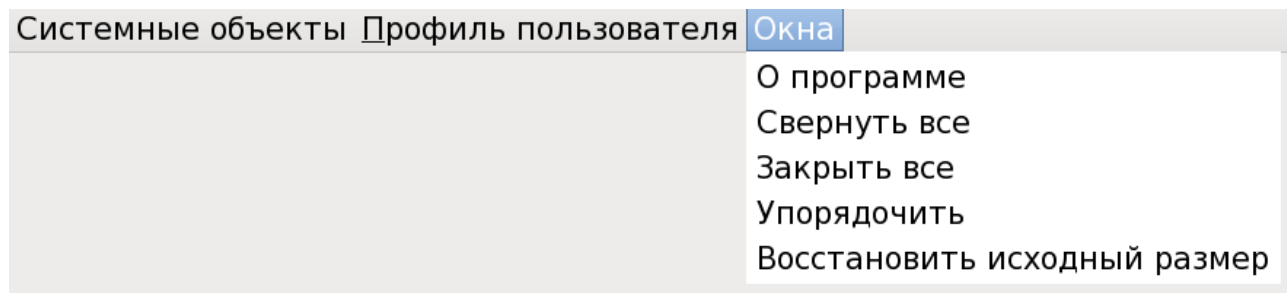


Рисунок 3 – Подменю «Окна»

В подменю «Окна» - «О программе» выводится справочная информация о версии ядра платформы документооборота, версии подсистемы РП, версии СУБД «Ред База Данных», версии java и ОС, на которой функционирует РП.

#### 4.3.2. Подменю «Системные объекты»

##### 4.3.2.1. «СИСТЕМНЫЕ НАСТРОЙКИ»

В данном подменю «Системные объекты» - «Системные настройки» пользователь может увидеть (рисунок 4) системные настройки РП, такие как: настройки почты, настройки интерфейса или общесистемные настройки. Доступ в данное подменю рекомендуется предоставлять только уполномоченным пользователям, а право на редактирование только администраторам РП.

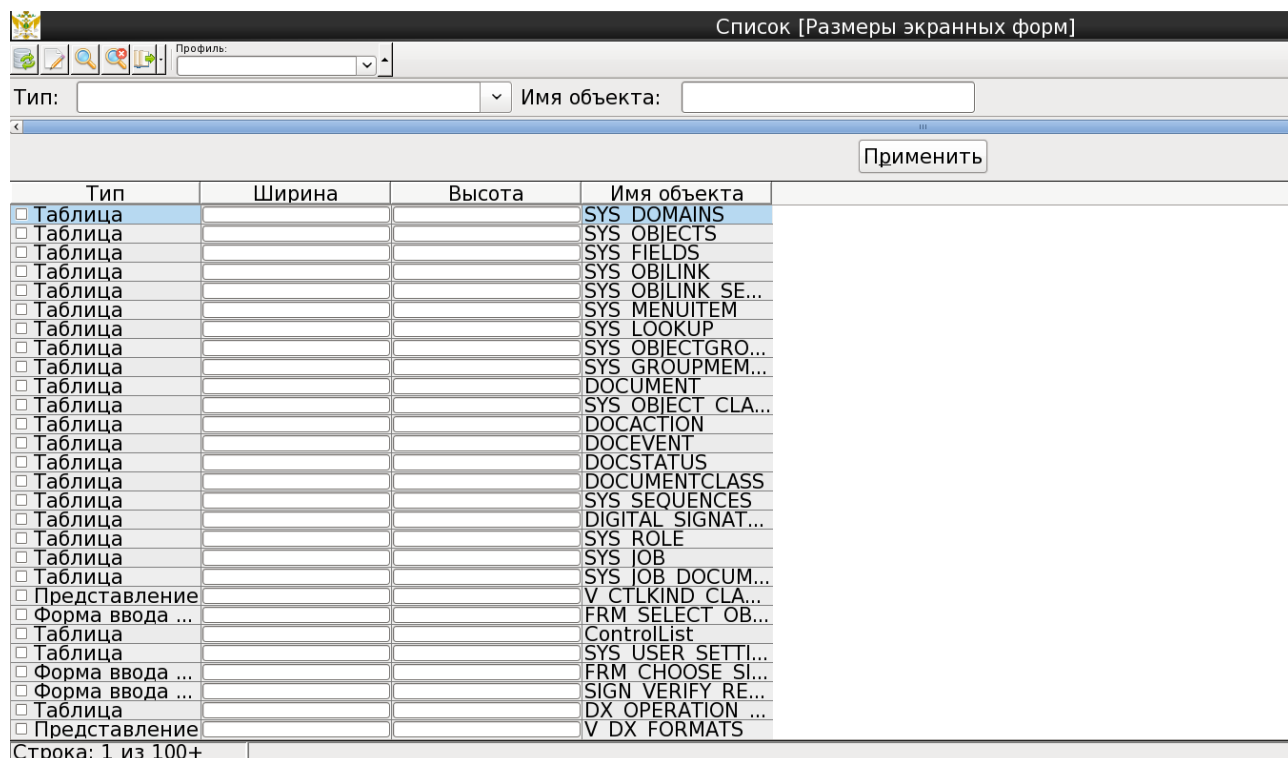
Имя	Описание	Группа	Значение	Пароль
SEC_RDB_PASSWOR...			false	****
SEC_RDB_PASSWOR...			false	****
SEC_RDB_PASSWOR...			false	****
server.ip.mirror			true	****
update.sessions.jour...	Вести журнал поль...	Системные настро...	true	****
interface.web.grid.s...	Показывать строку...	Настройки интерф...	false	****
SEC_LOGIN_FAILURE...	Задержка ответа с...		8	****
tomcat.session.time...	Время до разрыва с...		30	****
SEC_RDB_PASSWOR...	Срок действия пар...		10	****
SEC_LOG_STORE_DA...	Срок хранения соб...		12	****
SEC_SHOW_FORM	Просмотр карточки...		false	****
SEC_CHANGE_DS_SE...	Управление настро...		true	****
SEC_UNCOMPATIBLE...	Выполнение несов...		true	****
SEC_PRIVILEGE_UPG...	Повышение привил...		true	****
SEC_USER_LOGIN	Вход пользователя...		true	****
SEC_UNAPPROVED	Несанкционирован...		false	****
SEC_USER_LOGOUT	Выход пользовател...		true	****
SEC_DOCUMENT_PRI...	Печать документа		false	****
SEC_DOCUMENT_EX...	Выгрузка документа		true	****
SEC_USER_CHANGE	Управление учетно...		false	****
SEC_LOG_STORE_DA...	Срок хранения таб...		12	****
SEC_UNLOAD_TABLE	Выгрузка таблицы		false	****
SEC_USER_UNUSED	Неиспользование У...		false	****
SEC_CLIENT_SESSIO...			15	****
SEC_CLIENT_SESSIO...			15	****
SEC_CLIENT_SESSIO...			3	****
SEC_CLIENT_SESSIO...			4	****
SEC_LOG_STORE_SE...			24	****
SEC_LOG_STORE_CO...			24	****



## Рисунок 4 – Подменю «Системные настройки»

### 4.3.2.2. «РАЗМЕРЫ ЭКРАННЫХ ФОРМ»

В данном подменю «Системные объекты» - «Размеры экранных форм» пользователь может увидеть (рисунок 5) тип объекта, вызываемого как форма, его ширину и высоту (если они были предварительно заданы) и системное имя объекта.



Тип	Ширина	Высота	Имя объекта
<input type="checkbox"/> Таблица			SYS DOMAINS
<input type="checkbox"/> Таблица			SYS OBJECTS
<input type="checkbox"/> Таблица			SYS FIELDS
<input type="checkbox"/> Таблица			SYS OBJLINK
<input type="checkbox"/> Таблица			SYS OBJLINK SE...
<input type="checkbox"/> Таблица			SYS MENUITEM
<input type="checkbox"/> Таблица			SYS LOOKUP
<input type="checkbox"/> Таблица			SYS OBJECTGRO...
<input type="checkbox"/> Таблица			SYS GROUPEMEM...
<input type="checkbox"/> Таблица			DOCUMENT
<input type="checkbox"/> Таблица			SYS OBJECT CLA...
<input type="checkbox"/> Таблица			DOCACTION
<input type="checkbox"/> Таблица			DOCEVENT
<input type="checkbox"/> Таблица			DOCSTATUS
<input type="checkbox"/> Таблица			DOCUMENTCLASS
<input type="checkbox"/> Таблица			SYS SEQUENCES
<input type="checkbox"/> Таблица			DIGITAL SIGNAT...
<input type="checkbox"/> Таблица			SYS ROLE
<input type="checkbox"/> Таблица			SYS JOB
<input type="checkbox"/> Таблица			SYS JOB DOCUM...
<input type="checkbox"/> Представление			V CTLKIND CLA...
<input type="checkbox"/> Форма ввода ...			FRM SELECT OB...
<input type="checkbox"/> Таблица			ControlList
<input type="checkbox"/> Таблица			SYS USER SETTI...
<input type="checkbox"/> Форма ввода ...			FRM CHOOSE SI...
<input type="checkbox"/> Форма ввода ...			SIGN VERIFY RE...
<input type="checkbox"/> Таблица			DX OPERATION ...
<input type="checkbox"/> Представление			V DX FORMATS

Строка: 1 из 100+

Рисунок 5 – Подменю «размеры экранных форм»

### 4.3.2.3. «СБРОСИТЬ КЭШ»

В данном подменю «Системные объекты» - «Сбросить кэш» пользователь может очистить кэш, используемый РП. Никаких дополнительных настроек для выполнения этой операции не требуется.

### 4.3.2.4. «ЗАДАЧИ ПЛАНИРОВЩИКА»

В данном подменю «Системные объекты» - «Планировщик» - «Задачи планировщика» пользователь может увидеть (рисунок 6) запланированные задачи, которые могут быть выполнены уполномоченным пользователем немедленно или просмотреть задачи, которые стоят в очереди выполнения. Для каждой задачи отображается: наименование задачи, подсистема в рамках которой она выполняется, отметка об активности задачи, инструкция для планировщика задач (выражение Cron), отметка о необходимости уведомлений о выполнении и электронный почтовый адрес на который необходимо отправлять уведомления, также срок хранения результатов выполнения задачи.

Пользователь может, используя фильтр, отсортировать и найти нужные задачи по наименованию задачи или подсистемы. Рекомендуется оставлять право составления задач, выполняемых по расписанию, за уполномоченным администратором, предоставляя пользователю только возможность просмотра и выборочного запуска задач, если это необходимо. Для запуска определенной задачи необходимо выбрать определенную задачу (используя кнопки клавиатуры или однократным щелчком правой кнопки мыши на строке), и нажать кнопку «Выполнить задачу» (отображается в виде зеленой шестеренки) в панели инструментов.

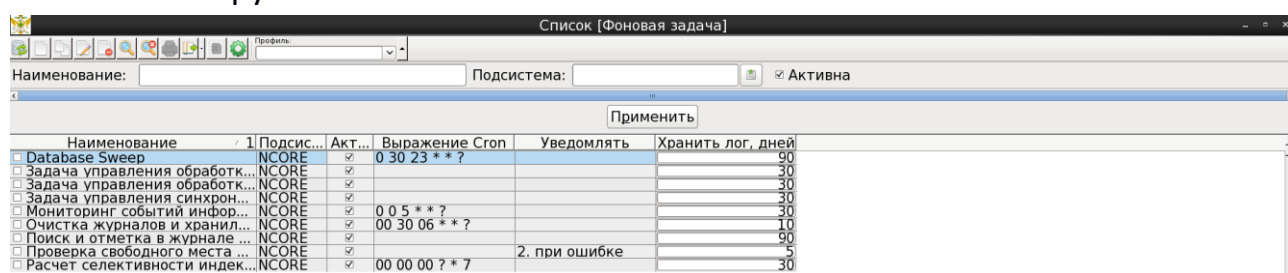


Рисунок 6 – Окно задач подменю «Задачи планировщика»

#### 4.3.2.5. «ШАБЛОНЫ ОТЧЕТОВ»

В данном подменю «Системные объекты» - «Отчеты» - «Шаблоны отчетов» пользователь может создавать, редактировать или удалять шаблоны отчетов, используемые в РП (рисунок 7).

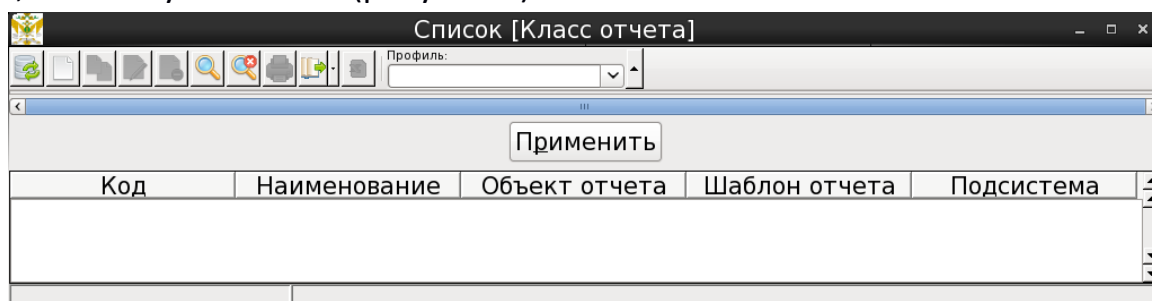


Рисунок 7 – Окно списка шаблонов подменю «Шаблоны отчетов»

В данном окне пользователь видит список уже существующих шаблонов, их код, наименование, объект отчета, шаблон отчета и наименование подсистемы, для которой создан данный шаблон. С помощью кнопок панели инструментов пользователь может создавать новые шаблоны, редактировать существующие, удалять их или осуществлять навигацию по существующим шаблонам задавая нужный фильтр поиска.

При создании нового шаблона или редактировании существующего вызывается окно, представленное на рисунке 8.

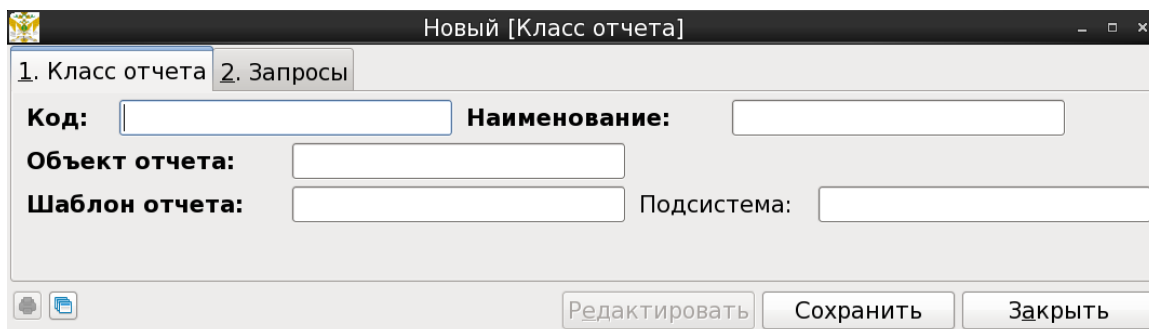


Рисунок 8 - Окно создания (редактирования) шаблона отчета

В этом окне пользователь должен заполнить представленные поля и нажать кнопку «Сохранить». Во вкладке «Запросы» пользователь может создать связанные с данным шаблоном запросы (рисунок 9). Здесь он также может редактировать представленные запросы или удалять их.

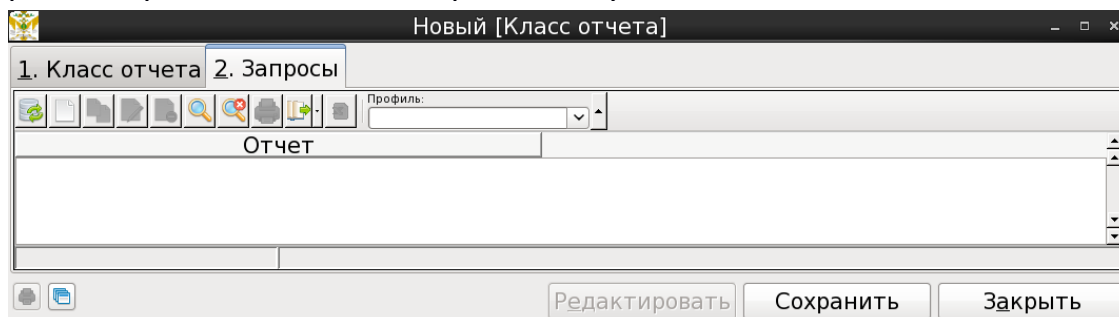


Рисунок 9 - Окно создания (редактирования) запросов шаблона

Создание нового запроса (или редактирование существующего) по нажатию соответствующей кнопки в панели инструментов или двойному щелчку по выбранному запросу, вызовет окно, представленное на рисунке 10.

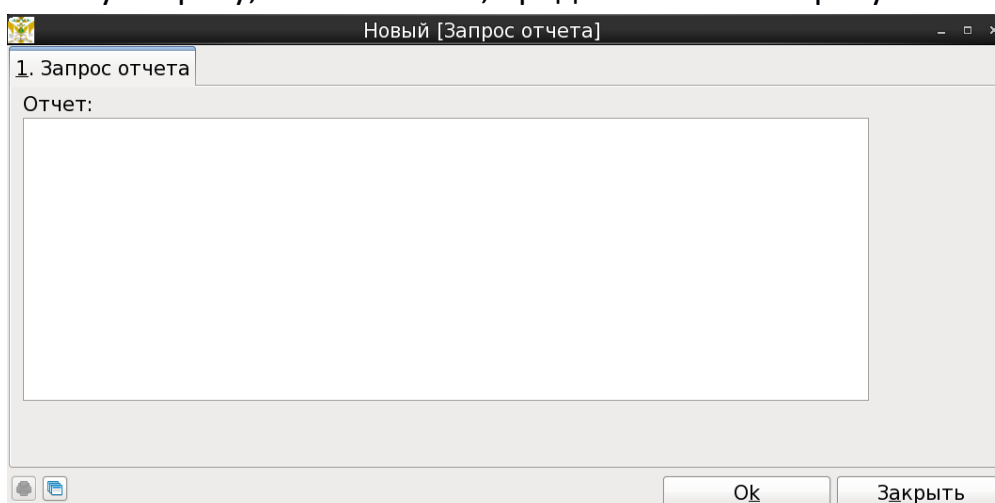


Рисунок 10 – Окно создания (редактирования) запроса шаблона отчета

Здесь пользователь должен заполнить текстовое поле «Отчет:» и нажать кнопку «ОК».

Все операции с окнами начинаются с нажатия кнопки «Редактировать»,

разблокирующей соответствующие кнопки панели инструментов.

Все операции с окнами должны заканчиваться нажатием кнопки «Сохранить», если необходимо сохранить внесенные изменения или «Закреть», если необходимо просто закрыть текущее окно или закрыть окно без сохранения изменений.

#### 4.3.2.6. «ШАБЛОНЫ ПЕЧАТНЫХ ФОРМ»

В данном подменю «Системные объекты» - «Отчеты» - «Шаблоны печатных форм» пользователь может создавать, редактировать или удалять шаблоны печатных форм, используемых в РП (рисунок 11).

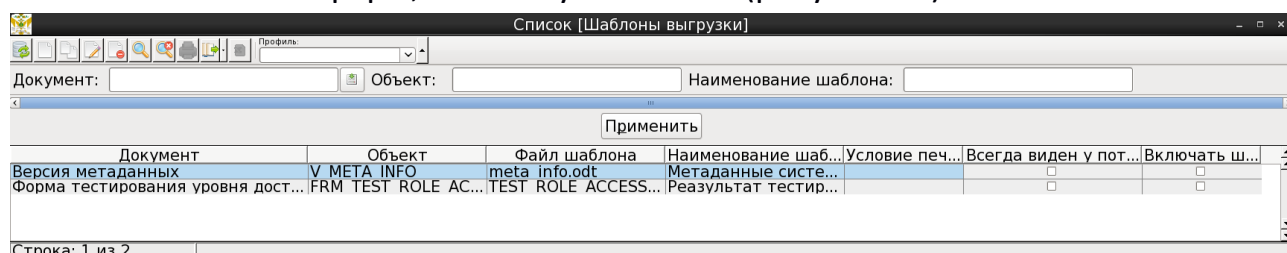


Рисунок 11 – Окно шаблонов печатных форм

В списке шаблонов отображается наименование документа, связанные с ним объект, файл шаблона, наименование шаблона, условие печати, отметка о видимости у потомка и отметка о включении в шаблон штрих-кода.

Создание нового шаблона печатной формы (или редактирование существующего) по нажатию соответствующей кнопки в панели инструментов или двойному щелчку по выбранному документу, вызовет окно, представленное на рисунке 12.

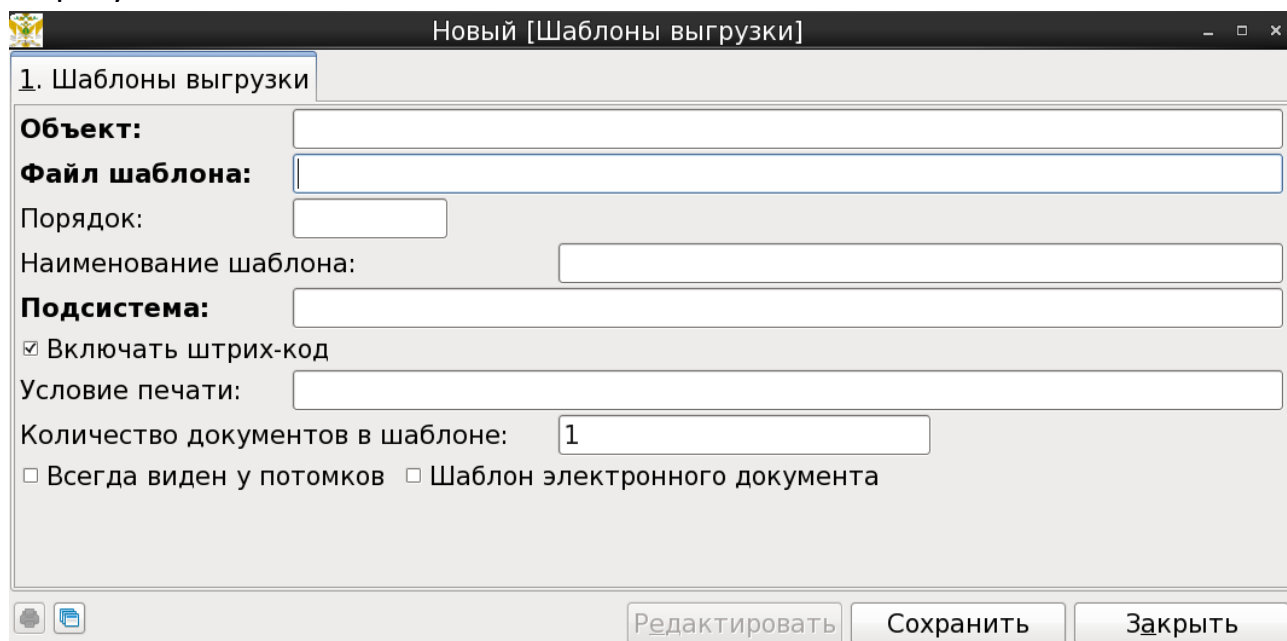


Рисунок 12 – Окно создания или редактирования шаблона печатной формы

В представленной форме пользователю необходимо заполнить все поля и

установить отметки в местах, где необходим выбор, после чего нажать кнопку «Сохранить».

Все операции с окнами начинаются с нажатия кнопки «Редактировать», разблокирующей соответствующие кнопки панели инструментов.

Все операции с окнами должны заканчиваться нажатием кнопки «Сохранить», если необходимо сохранить внесенные изменения или «Заккрыть», если необходимо просто закрыть текущее окно или закрыть окно без сохранения изменений.

#### 4.3.2.7. «ДОКУМЕНТЫ»

В данном подменю «Системные объекты» - «Документооборот» - «Документы» пользователь может создавать, редактировать или удалять документы, используемых в РП (рисунок 13).

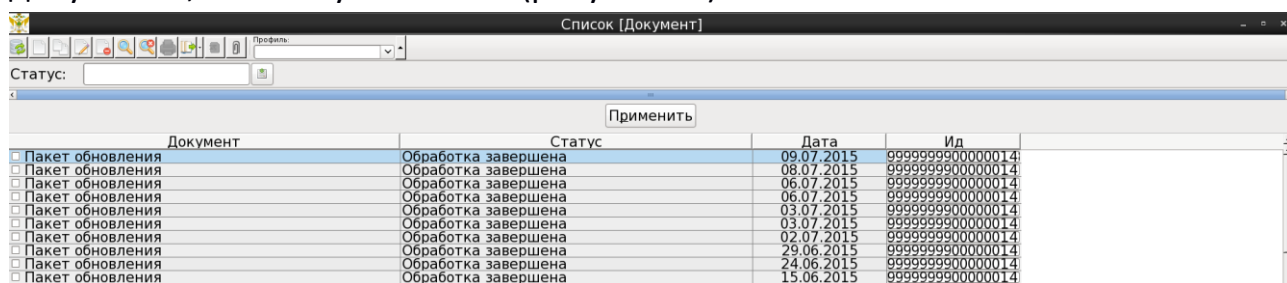


Рисунок 13 – Окно списка документов подменю «Документы»

Под панелью инструментов находится строка фильтра, в которой можно установить статус интересующих пользователя документов. После нажатия кнопки «Применить», произойдет сортировка по выбранному фильтру в списке всех документов и в таблице будут отображены только те документы, что соответствуют заданному фильтру.

Если щелкнуть правой клавишей мыши по заголовку столбца таблицы, то появится дополнительное меню для работы с таблицей (рисунок 14).

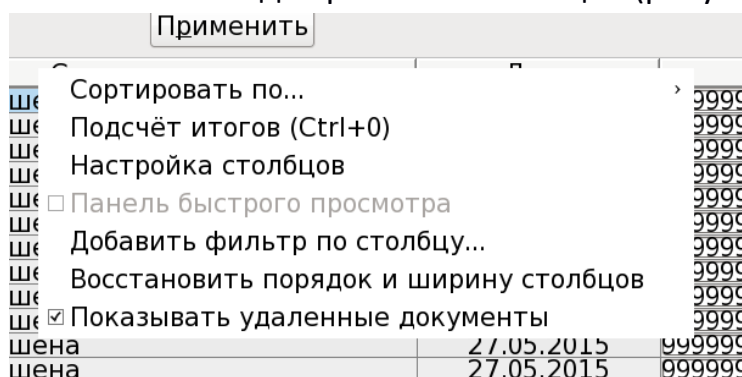


Рисунок 14 – Меню настройки таблицы

В данном меню пользователь может выбрать режимы сортировки строк, подсчитать количество строк в таблице, настроить отображаемые столбцы, добавить фильтры по столбцу, восстановить изначальный порядок и ширину столбцов и указать необходимость отображения удаленных документов.

В окне настройки столбцов (рисунок 15) пользователь может отметить те столбцы, которые необходимо отображать при общем выводе списка документов. После внесения изменений необходимо нажать кнопку «Сохранить» или «Отмена», если изменения не требуются.

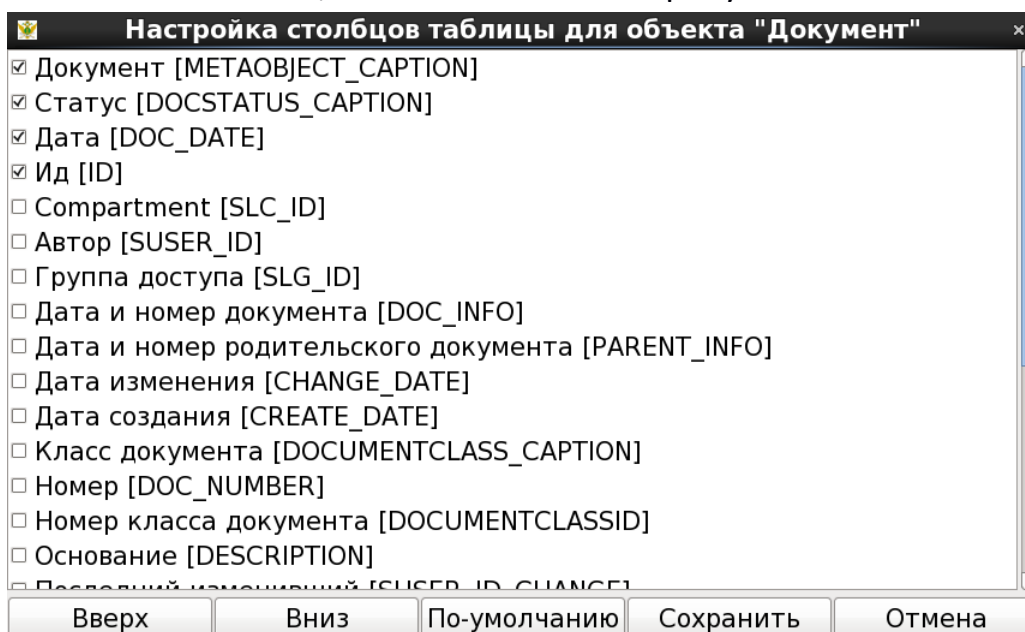


Рисунок 15 – Окно настройки столбцов

В окне «Параметры фильтра» (рисунок 16) пользователь может создать и добавить свой собственный фильтр, по которому будет осуществляться выборка из списка документов. В этом окне задается поле (столбец таблицы), в котором будет осуществляться поиск, условие сравнения и значение, по которому будет производиться сравнение. В поле «Описание:» приводится краткая справка по использованию данного фильтра.

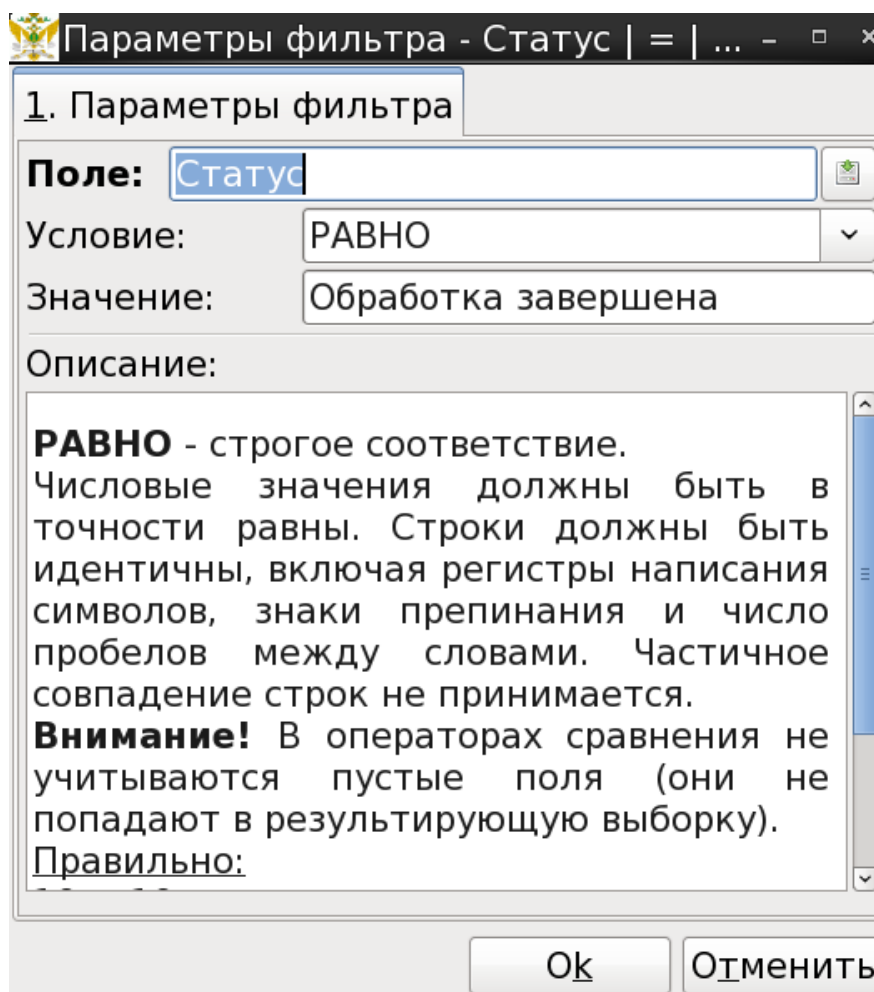


Рисунок 16 – Окно «Параметры фильтра»

После нажатия кнопки «Ок» созданный фильтр появится в окне с таблицей, содержащей список документов (рисунок 17). Для его применения к текущему представлению необходимо нажать кнопку «Применить».

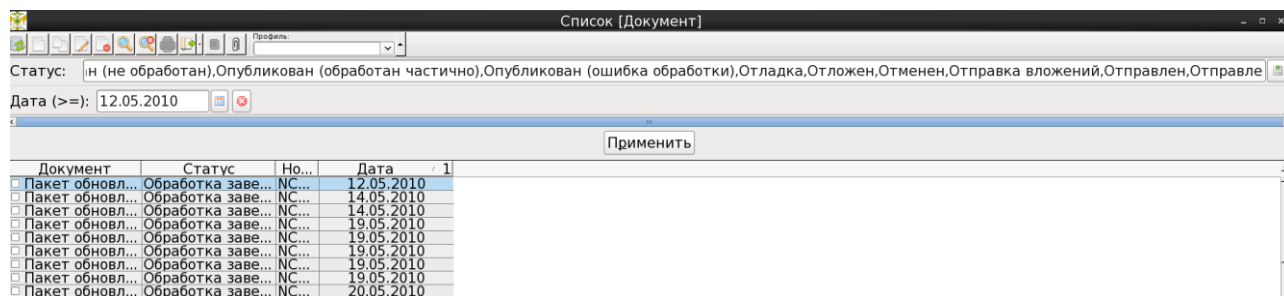


Рисунок 17 – Окно подменю «Документы» после создания дополнительного фильтра

Для сброса сортировки документов по текущему фильтру необходимо нажать кнопку «Удалить фильтр» в строке, описывающей фильтр, и нажать кнопку «Применить».

В нижней части экрана расположена статусная панель (рисунок 18), в которой отображается краткая информация по таблице (общее количество записей и номер выбранной записи) и выбранной строке таблицы (дата и время создания документа, кем был создан документ, дата и время изменения

документа и кем были произведены эти изменения).

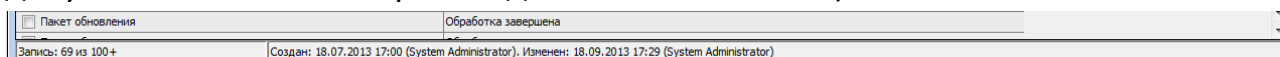


Рисунок 18 – Статусная панель подменю «Документы»

Для создания нового документа необходимо нажать кнопку «Создать» на панели инструментов или функциональную клавишу F9. В появившемся окне (рисунок 19) необходимо выбрать тип создаваемого документа, нажав кнопку выбора, справа от пустого поля типа документа.

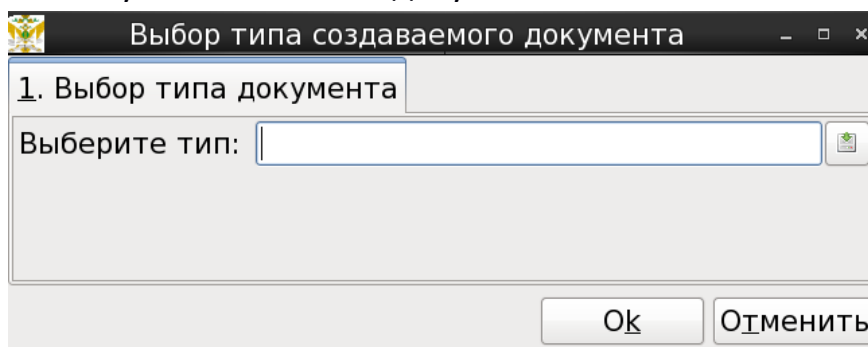


Рисунок 19 – Окно создания документа

После нажатия кнопки выбора типа документа появится новое окно (рисунок 20), в котором пользователю будет предложено выбрать тип документа из имеющихся шаблонов. Шаблоны могут быть сгруппированы, и для раскрытия группы необходимо щелкнуть по значку «+» и открыть вложенный список. Выбрать шаблон можно двумя способами: двойным щелчком левой клавиши мыши по необходимому типу документа или однократным щелчком левой клавиши мыши и нажатием кнопки «Выбрать».

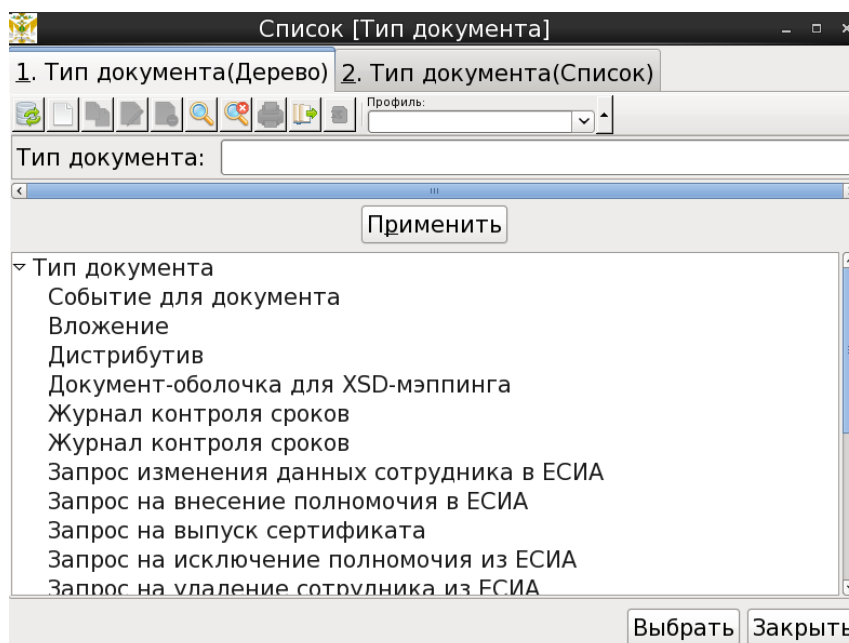


Рисунок 20 – Окно выбора типа создаваемого документа

После успешного выбора типа документа пользователь будет возвращен в



предыдущее окно (рисунок 19), но строка «Выберите тип:» будет заполнена выбранным типом документа. После нажатия кнопки «Ок» пользователю будет вызвано новое окно, в котором будет предложено заполнить требуемые поля. Состав этих полей зависит от выбранного типа документа. На рисунках 21 и 22 приведены варианты создания новых документов типа «Сертификат ЭЦП» и «Сообщение».

Новый [Сертификат ЭЦП]

1. Сведения о сертификате 2. Сертификат

Имя пользователя:

Серийный номер:

Дата выпуска сертификата: . . . :

Дата истечения срока действия сертификата: . . . :

Дата истечения срока действия открытого ключа: . . . :

Полное имя DN пользователя:

Назначение сертификата:

Статус (1)

Рисунок 21 – Создание документа типа «Сертификат ЭЦП»

Новый [Сообщение]

1. Сообщение

Номер:  Наименование агента отправителя:

Наименование подразделения агента отправителя:

Дата: . . .

Наименование соглашения отправителя:

Код сообщения:  Отправка пакета документов

Наименование агента получателя:

Ключ документа отправителя:

Ключ документа получателя:

Наименование подразделения агента получателя:

Наименование соглашения получателя:

Код ошибки:

Заголовок:

Текст сообщения:

Новый

Рисунок 22 – Создание документа типа «Сообщение»

После заполнения всех обязательных полей пользователь должен нажать кнопку «Сохранить» или «Закреть» (зависит от шаблона создаваемого документа), после чего ему будет предложено (рисунок 23) сохранить внесенные изменения.

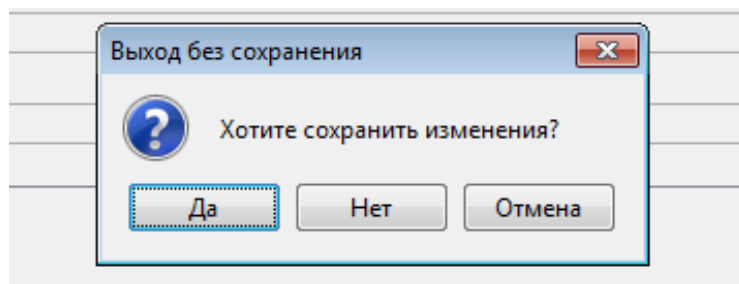


Рисунок 23 – Окно сохранения документа

При необходимости редактирования уже существующего документа, пользователь должен сделать двойной щелчок левой клавиши мыши по необходимому документу или однократным щелчком левой клавиши мыши подсветить строку таблицы и нажать кнопку «Редактировать» в панели инструментов (или использовать функциональную клавишу F4).

Пользователю будет представлено окно аналогичное окнам, изображенных на рисунках 21 и 22 (зависит от шаблона выбранного документа), где он сможет изменить необходимые поля.

Для удаления документа пользователь должен однократным щелчком мыши подсветить строку таблицы и нажать кнопку «Удалить» в панели инструментов (или использовать клавишу DEL). После чего ему будет предложено подтвердить процедуру удаления документа (рисунок 24).

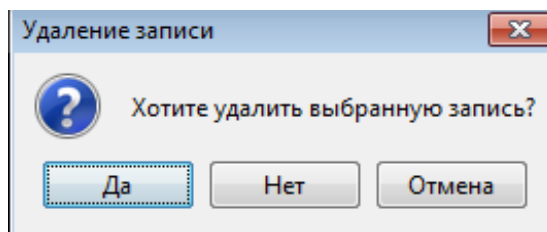


Рисунок 24 – Окно подтверждения удаления документа

#### 4.3.3. «Профиль пользователя»

В данном меню «Профиль пользователя» пользователь может изменить свой пароль. Для этого необходимо перейти в подменю «Профиль пользователя» - «Изменить пароль» (рисунок 25) и ввести в соответствующих полях: старый (текущий) пароль, новый пароль и повторить его в поле «подтверждение». Если новый пароль соответствует заданным метрикам, используемым в РП, произойдет успешная смена пароля. В противном случае будет выведено окно предупреждения, где будет указано, каким критериям не соответствует новый пароль.

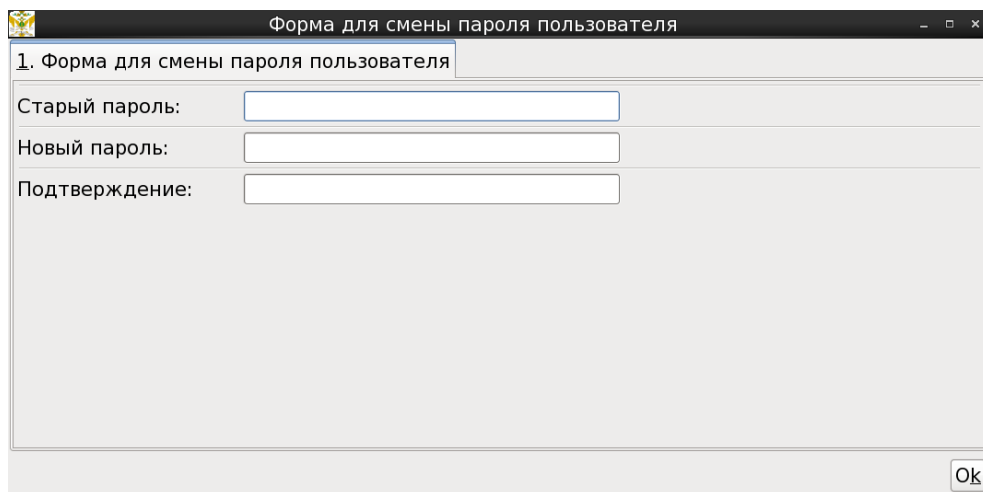


Рисунок 25 – Окно смены пароля

Так же в этом меню пользователь может видеть историю неуспешных попыток ввода пароля с момента последней успешной аутентификации (рисунок 26).

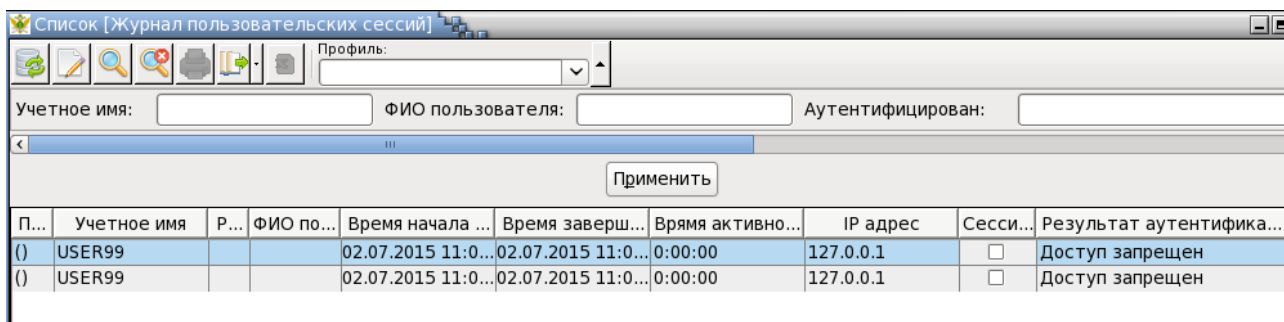


Рисунок 26 – Окно журнала пользовательских сессий меню «Профиль пользователя» - «Неуспешные попытки входа»

#### 4.4. Завершение работы программы

В архитектуре «тонкий клиент» для завершения сеанса работы с РП, пользователь должен нажать кнопку «Выход» в правом верхнем углу или закрыть окно (текущую вкладку) веб-браузера (рисунок 27).

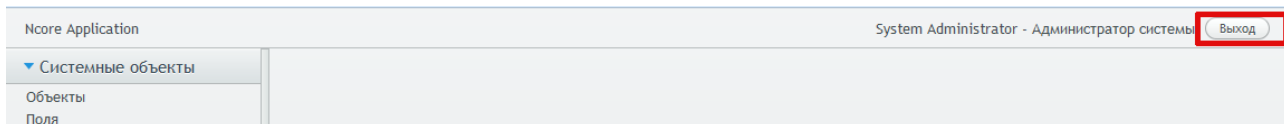


Рисунок 27 – Завершение работы РП в «тонком клиенте»

В архитектуре «толстый клиент» для завершения сеанса работы с РП, пользователь должен закрыть приложения, используя стандартную кнопку закрытия окна.

В целях безопасности сеанс работы с РП может быть завершен автоматически в случае бездействия подсистемы в течение определенного

времени, протяженность которого указывает администратор.

## 5. Сообщения пользователю

При правильном функционировании РП все ошибки, вызванные действиями пользователя, приводят к возникновению окна с сообщением об ошибке (рисунок 26).

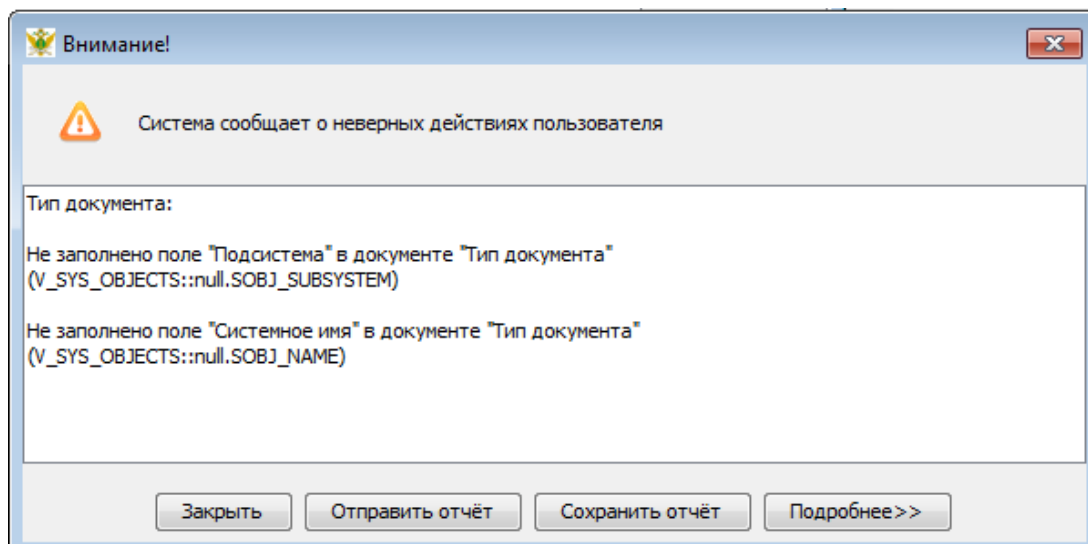


Рисунок 26 – Окно ошибки

В данном окне пользователь получает сведения о том, какие действия вызвали ошибку, и что необходимо предпринять для ее устранения. Он может посмотреть подробные технические сведения, нажав кнопку «Подробнее». Сохранить локальную копию отчета об ошибке нажав кнопку «Сохранить отчет». Отправить отчет разработчику (или иному лицу, в зависимости от системных настроек) по электронной почте нажав кнопку «Отправить отчет». Или закрыть окно кнопкой «Закрыть» и вернуться в предыдущее окно.

Если возникает какая-либо ошибка, которая не может быть устранена пользователем, то возникает окно (рисунок 27), где описан код ошибки и действия, которые может предпринять пользователь. В данном случае пользователю необходимо обратиться к системному администратору или разработчику.

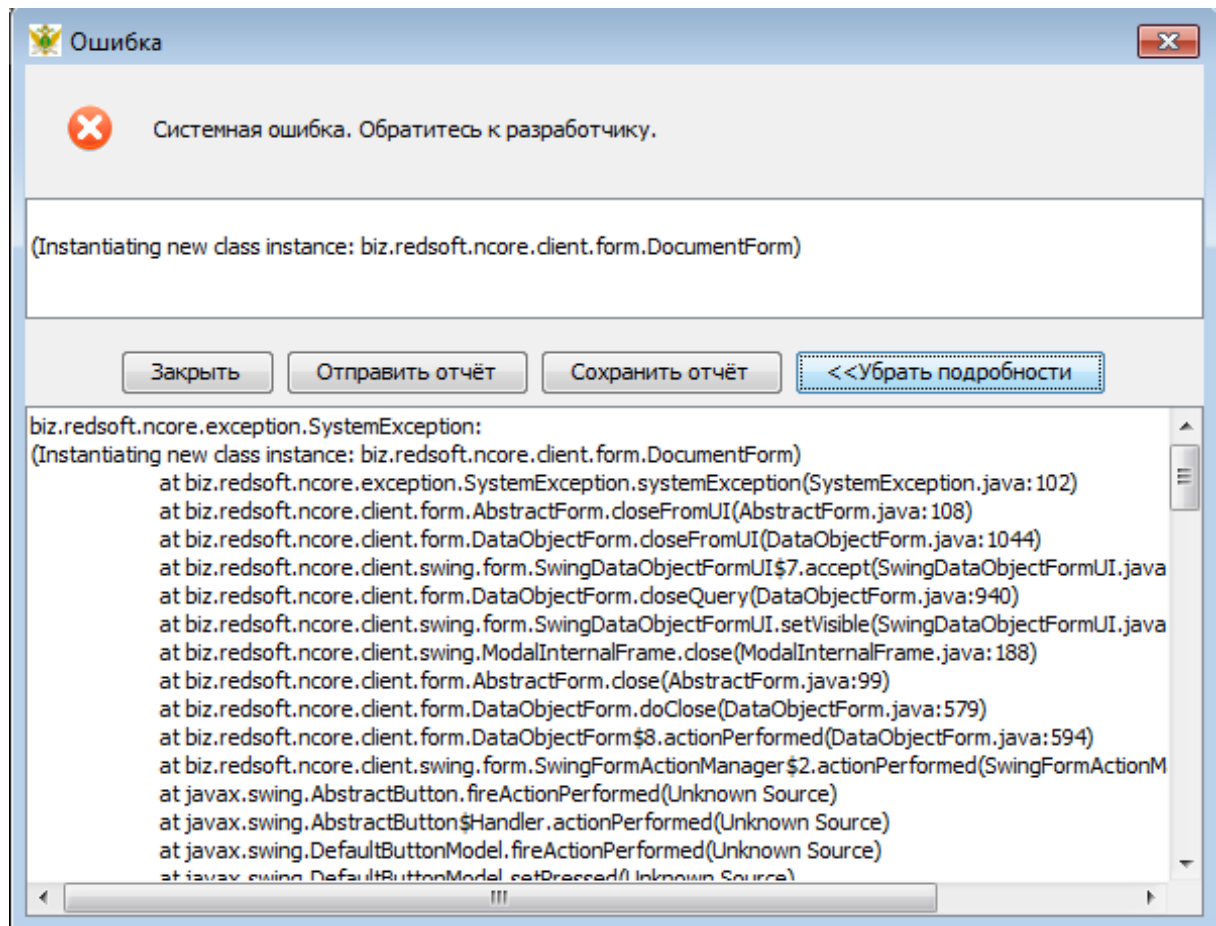


Рисунок 27 – Окно системной ошибки

## 6. ncore-api

API (application programming interface) - это программный интерфейс приложения.

API определяет функциональность, которую предоставляет приложение, при этом API позволяет абстрагироваться от того, как именно эта функциональность реализована.

В данном разделе описывается модуль ncore-api, служащий для создания и конфигурации web-API для прикладных решений на основе РП. Web-API - это API реализованная посредством веб-сервиса или набора веб-сервисов.

Как правило, web-API может выполнять следующие функции:

- удаленное создание, изменение, вставка и удаление данных в системе (CRUD);
- удаленный вызов неких процедур и возврат результата их исполнения (RPC).

API может применяться для следующих нужд:

- создание пользовательского интерфейса к системе (front-end), например, для личных кабинетов;
- пересылка данных или документов между системами (интеграция);
- создание систем централизованного хранения данных, например, НСИ, или серверов единой авторизации пользователей (SSO);
- и многое другое.

### 6.1. Подключение к проекту

Ncore-api реализовано как отдельный модуль. Для подключения модуля требуется добавить зависимость в build.gradle (версию napi прописать в gradle.properties)

```
def napiVersion= project.properties.'biz.redsoft.napi.version'  
compile("biz.redsoft.napi:ncore-api:${napiVersion}") {  
    exclude group: 'biz.redsoft.ncore', module: 'ncore'  
}  
updates "biz.redsoft.napi:ncore-api:${napiVersion}:updates"
```

В register.json добавить зависимость

biz.redsoft.napi

В web.xml заменить

```
<listener-class>biz.redsoft.ncore.dx.init.DxInitializationListener</listener-class>
```

на

```
<listener-class>biz.redsoft.napi.api.ApiInitializationListener</listener-class>
```

Требуется версия ядра не ниже 1.5.96.0

## 6.2. Виды API

На текущий момент в системе присутствует только один вид API - NCORE REST API. Планируется к созданию JSON-API. Рассматривается возможность реализации GraphQL.

## 6.3. Endpoint, или точка входа

Каждый API должен иметь точку входа. Точка входа служит для приема запросов и отправки ответов. Для web-API точка входа определяется URL. В модуле ncore-api для конфигурирования точек входа служит системный объект API\_ENDPOINT и его потомки. Каждый потомок API\_ENDPOINT реализует собственный вид API, так API\_ENDPOINT\_REST служит для реализации REST\_API. Помимо разновидности, основной характеристикой точки входа служат код, версия и путь. Код и версия вместе составляют уникальный ключ точки входа. Параметр путь также должен быть уникальным в рамках приложения, так как он определяет URL, по которому будет доступна точка входа. Путь задается относительно пути к приложению, например если приложению. Например:

Путь к приложению: `http://localhost:8080/napi`

Путь, указанный в объекте API\_ENDPOINT: `/rest/v0/`

Результирующий путь к точке входа: `http://localhost:8080/napi/rest/v0/`

Путь к точке входа должен начинаться и заканчиваться символом '/' (слэш), и должен содержать допустимые в URL символы. По умолчанию путь составляется из кода и версии точки входа по шаблону `"/<код>/<версия>/"`, но может быть изменен. Все запросы, обращенные к пути, начинающемуся с пути точки входа, будут обработаны точкой входа. В связи с этим, пути к точкам входа должны быть уникальными и не перекрывать друг друга.

***NB!*** Путь регистрозависим!

### REST

REST API или RESTful API строится на основе протокола http, и использует его не для передачи не только данных, но и управляющей информации.



Преимуществами REST является высокое быстродействие, универсальность, широкая распространенность, простота применения. Недостатками является отсутствие унификации и стандартов, нераспространенность языков описания API.

## **JSON-API**

*раздел в разработке*

### **6.4. Ресурсы**

Как говорилось выше, основными функциями API являются оперирование данными (CRUD) и удаленный вызов процедур (RPC). Для того, чтобы определить, какая функция должна выполняться при вызове web-API, клиент указывает в URL имя ресурса, к которому хочет обратиться (подробнее см. Структура URL). В зависимости от выбранного ресурса и переданных параметров, сервер выполняет определенные функции, в результате чего может изменяться состояние сервера и возвращаться некоторые данные клиенту. Каждая точка входа обладает своим набором ресурсов, который и определяет функционал точки входа. Настройки ресурсов в psoge-api хранятся в объекте API\_RESOURCE и его потомках. Каждый ресурс характеризуется уникальным именем (в рамках точки входа). Кроме того, ресурсы могут иметь псевдонимы, в этом случае псевдоним заменяет имя и тоже должен быть уникальным в рамках данной точки входа.

### **6.5. Резолверы ресурсов**

На основе настроек ресурсов в зависимости от типа точки входа могут создаваться разные типы ресурсов, это входит в функционал Резолвера ресурсов (см. следующий пункт). Так, резолвер RestResourceResolver на основе настройки ресурса API\_RESOURCE\_DATAOBJECT создает ресурс RestDataObjectResource, а на основе настройки API\_RESOURCE\_ALIAS - RestAliasResource. Все ресурсы имплементируют интерфейс IfApiResource. Для инициализации, хранения и выбора ресурсов применяются классы, имплементирующие интерфейс IfApiResourceResolver. В зависимости от вида точки входа, могут использоваться разные реализации. Все резолверы хранятся в общем кэше ApiResolversCache который инициализируется при старте системы. Резолвер инициализируется точкой входа и помещается в кэш. В качестве ключа кэша используется путь точки входа.

### **6.6. Структура URL**

URL традиционно записывается следующим образом:

<схема>:[//[<логин>:<пароль>@]<хост>[:<порт>]][/]<URL-  
путь>[?<параметры>][#<якорь>]

В этой записи:

*схема:*

схема обращения к ресурсу; в большинстве случаев имеется в виду сетевой протокол

Для REST-API это http или https

*логин:*

имя пользователя, используемое для доступа к ресурсу  
В API не используется

*пароль:*

пароль указанного пользователя  
В API не используется

*хост:*

полностью прописанное доменное имя хоста в системе DNS или IP-адрес хоста в форме четырёх групп десятичных чисел, разделённых точками; числа — целые в интервале от 0 до 255.

*порт:*

порт хоста для подключения

*URL-путь:*

уточняющая информация о месте нахождения ресурса; зависит от протокола. Путь состоит из двух частей: путь точки входа и непосредственно путь к ресурсу. Путь к ресурсу может быть как простым (состоящим из одной строки), так и составным (разделённым с помощью символа "/"). Составной путь применяется при наличии иерархии ресурсов. Как правило, составной путь состоит из наименований ресурсов и ключей ресурсов, например:

/resource1/:id1/resource2/:id2/

*параметры:*

строка запроса с передаваемыми на сервер параметрами. Начинается с символа ?, разделитель параметров — знак &. Пример:  
?параметр\_1=значение\_1&параметр\_2=значение\_2&параметр3=значение\_3

*якорь:*

идентификатор «якоря» (англ.)русск. с предшествующим символом #  
В API не используется

## 6.7. Авторизация и аутентификация. ApiApplicationManager

Основной особенностью ncore-api является то, что оно позволяет осуществлять запросы с указанием имени пользователя и пароля. При этом подключение к базе осуществляется именно под выбранным пользователем, что позволяет в полной мере использовать весь функционал NCORE по разграничению доступа, журналированию и т.п. Для авторизации требуется указывать в запросе заголовок "authorization", содержащего логин и пароль, разделенные двоеточием и упакованные в Base64. Кроме того, в каждой точке входа может быть установлен пользователь по умолчанию, под которым будут осуществляться подключение к базе анонимных пользователей. Этот пользователь должен иметь минимальные привилегии. Поскольку архитектура ncore подразумевает, что подключения к базе данных инкапсулирует в себя объект Application, для каждого обращения к API требуется получить объект ApiApplication. ApiApplication - это максимально легковесная разновидность Application, созданная специально для api. Однако создание ApiApplication и подключения к базе все-таки требует достаточно больших ресурсов.

В ncore-api эта проблема решается за счет использования ApiApplicationManager, который содержит пул ApiApplication. ApiApplicationManager инициализируется при старте сервера. При запросе API точка входа создает настройки подключения, и запрашивает менеджер о наличии ApiApplication с такими настройками. В случае отсутствия свободного ApiApplication, он будет создан. После завершения обработки ApiApplication освобождается и возвращается в пул.

ApiApplicationManager имеет следующие настройки пула:

- *"ncore.api.application.pool.maxWait"* - время ожидания соединения, если пул исчерпан
- *"ncore.api.application.pool.maxActive"* - максимальное количество соединений (на пользователя)
- *"ncore.api.application.pool.maxTotal"* - общее максимальное количество соединений
- *"ncore.api.application.pool.maxIdle"* - максимальное число соединений (на пользователя), которые могут быть неактивными
- *"ncore.api.application.pool.minIdle"* - минимальное число соединений (на пользователя), которые могут быть неактивными
- *"ncore.api.application.pool.numTestsPerEvictionRun"* - количество удаляемых соединений за один интервал опроса
- *"ncore.api.application.pool.minEvictableIdleTimeMillis"* - интервал, по истечении которого неиспользуемое соединение удаляется
- *"ncore.api.application.pool.timeBetweenEvictionRunsMillis"* - периодичность проверки необходимости закрытия неиспользуемых соединений

**NB!** Следует с осторожностью менять настройки менеджера, поскольку неправильные значения могут привести к значительным затратам памяти или потере производительности.

## 6.8. ApiProcessContext

ApiProcessContext создается API непосредственно в начале обработки каждого запроса в API, используется до самого конца обработки запроса и обязательно должен быть закрыт по окончании. Обратите внимание, что по умолчанию при закрытии контекста вызывается rollback. Commit должен быть выполнен вашей точкой входа явно. Основными функциями ApiProcessContext являются:

- хранение некоторых параметров точки входа, таких как размер страницы
- анализ и хранение частей URL запроса, включая параметры.
- создание, хранение и закрытие контекста (biz.redsoft.ncore.system.Context)
- выполнение commit и rollback хранимым контекстом.
- хранение промежуточных результатов обработки ресурсов

**NB!** При создании собственных видов точек входа убедитесь, что вы всегда закрываете ApiProcessContext! Используйте конструкцию try-with-resources.

## 6.9. Описание NCORE REST API

NCORE REST API реализует архитектуру REST, и описывается точкой входа API\_ENDPOINT\_REST.

Классы, реализующие NCORE REST API находятся в пакете biz.redsoft.napi.api.rest. Функции точки входа осуществляет RestServlet. В качестве резолвера ресурсов используется RestResourceResolver.

### 6.9.1. Виды ресурсов REST

Ресурсы создаются на основе настроек (API\_RESOURCE), относящихся к точке входа.

#### • RestDataObjectResource

Создается на основе настройки API\_RESOURCE\_DATAOBJECT. Может содержать поля и/или связи. Данный вид ресурса создается на основе системного объекта, его полей и связей. Каждый ресурс соответствует конкретному системному объекту, а каждое поле или связь - его полю или связи. Наследование не применяется, однако возвращаемые данные могут содержать

название ресурса, соответствующего реальному объекту данных.  
Тип данных:

json

Доступные методы:

GET, POST, PUT, DELETE

Коды ответа:

200, 201, 202, 204, 400, 404, 405, 409, 500

Виды представлений:

Список, экземпляр

Структура выходных данных:

```
{
  "meta": {
    "key": "<ключ экземпляра>",
    "type": "<metaname экземпляра>",
    "resource": "<ресурс, отражающий metaname. Может не совпадать с
запрошенным ресурсом>"
    "docstatusid": "<Код статуса. Является признаком документа>"
  },
  "data": {
    "<поле>": "<значение>"
    ...
  }
  "included": {
    "<код_включения>": {...}
  }
}
```

Блок "included" необязательный, передается только в случае запроса через @include

Структура входных данных (PUT: для экземпляра, POST: для списка):

```
{
  "<поле>": "<значение>"
  ...
}
```

Параметры:

- @offset - количество пропускаемых элементов до начала страницы, по умолчанию 0
- @count - количество элементов на странице, по умолчанию 50
- @fields - перечень возвращаемых полей, перечисляются через точку с запятой, по умолчанию передаются все поля
- @include - перечень возвращаемых подчиненных ресурсов, по умолчанию не передается. Названия подчиненных ресурсов перечисляются через точку с запятой
- @debug - параметр для отладки, при ошибке возвращается стектрейс
- @sort - сортировка

Прочие параметры применяются для фильтрации списков. Параметр передается в форме [условие]<имя\_поля>=<значение>. Доступные условия:

- =
- <>
- &=
- &>
- <
- <=
- >
- >=
- ISNULL
- ISNOTNULL
- DISTINCT
- NOT DISTINCT
- NOT LIKE
- LIKE
- LIKES
- LIKES\_OR
- IN
- NOT IN
- STARTING
- NOT STARTING
- EQUAL\_IGNORE\_CASE
- SIMILAR TO

Условие должно быть защищено URL-преобразованием

Поддержка вложенных ресурсов:

да, на основе связей, а также системные ресурсы.

- **RestAliasResource**

Создается на основе настройки API\_RESOURCE\_ALIAS. Не содержит полей и связей. При обращении передает любой запрос указанному ресурсу.

## 6.9.2. Системные ресурсы REST

В отличие от обычных, системные ресурсы не создаются на основе настроек, а формируются автоматически

- **@actions**

Служит для чтения и выполнения действий документооборота. Применяется только к экземплярам ресурсов, описывающих документы, не к спискам. Может быть запрошен через @includes (включая и списки) Не может быть корневым ресурсом.

Тип данных:

json

Доступные методы:

GET, PUT

Коды ответа:

200, 204, 400, 404, 405, 500

Виды представлений:

Список, экземпляр

Структура выходных данных:

"<имя действия>": "<заголовок>"

Структура входных данных (только для экземпляра):

```
{  
  "<поле>": "<значение>"  
  ...  
}
```

Поддержка вложенных ресурсов:

нет

Параметры:

нет

В качестве входных данных принимает параметры документооборота. Обработка форм ввода параметров на данный момент не поддерживается.

- **@reload**

Корневой ресурс. Применяется для сброса кэша.  
Тип данных:

нет

Доступные методы:

GET

Коды ответа:

200, 500

Виды представлений:

нет