

СИСТЕМА
ГАЛАКТИКА
РАСПИСАНИЕ УЧЕБНЫХ
ЗАНЯТИЙ

РУКОВОДСТВО АДМИНИСТРАТОРА



2025

АННОТАЦИЯ

Руководство предназначено для администраторов системы **Галактика Расписание учебных занятий** и содержит описание задач и действий по установке, администрированию и поддержанию работоспособности системы.

Издание 09.2025

Оглавление

1. Назначение системы.....	6
1.1. Термины и сокращения.....	8
2. Общие принципы работы системы.....	9
3. Развертывание системы.....	12
3.1. Лицензирование.....	12
3.1.1. Общие сведения	12
3.1.2. Генерация локального ключа лицензирования	12
3.1.3. Создание файла лицензии	13
3.1.4. Схема установки и запуска конечного приложения	14
3.1.5. Установка и настройка сервиса лицензий	15
3.1.6. Запуск и удаление сервиса лицензий из командной строки	16
3.1.7. Возможные проблемы и их устранение	17
3.2. Win. Основное решение.....	19
3.2.1. Требования к программному и техническому обеспечению	19
3.2.2. Основные варианты развертывания	20
3.2.2.1. Развертывание системы на клиентских компьютерах.....	21
3.2.2.2. Развертывание системы на сервере.....	21
3.2.2.3. Лицензионное использование.....	22
3.2.3. Установка системы	22
3.2.3.1. Обычная установка.....	23
3.2.3.2. Выборочная установка.....	24
3.2.3.3. Полная установка.....	25
3.2.4. Начало использования системы	26
3.2.4.1. Генерация БД	26
3.2.4.2. Аутентификация входа	27
3.2.4.3. Загрузка демонстрационных данных.....	28
3.2.5. Обновление системы	28
3.2.6. Удаление системы	29
3.3. Web-служба расписания.....	29
3.3.1. Настройка IIS	30
3.3.2. Дополнительная настройка сервера	33
3.3.3. Настройка на БД	34
3.3.4. Настройка взаимодействия пользовательского web-приложения с web-службой системы	35
3.3.5. Настройка для кроссдоменных AJAX-запросов	35
3.3.6. Запуск	35
3.4. Web. Просмотр расписания.....	36
3.4.1. Настройка IIS	36
3.4.2. Настройка на БД	36
3.4.3. Запуск	38
3.5. Web. Заявки на бронирование/дооснащение аудиторий и запреты преподавателей.....	38
3.6. Мобильный клиент.....	38
3.7. Настройка конфигурационного файла.....	39
3.7.1. Соединение с БД	39
3.7.2. Ссылка на сервис лицензирования	40
3.7.3. Путь к лог-файлу приложения	41
3.7.4. Путь к файлам модели пользователя	42
3.7.5. Путь к файлам модели сборки	42
3.7.6. Отображение диагностическая информация	43

3.7.7.	Настройка аутентификации	43
3.7.8.	Настройка выбора БД	43
4.	Безопасность приложения и настройка прав доступа	45
4.1.	Ролевая схема администрирования	45
4.2.	Предустановленные роли	50
4.3.	Ведение пользователей	52
4.3.1.	Создание/редактирование пользователя	52
4.3.2.	Приостановка действия учетной записи пользователя	53
4.3.3.	Удаление пользователя	54
4.3.4.	Предустановленные пользователи	54
4.3.5.	Особенности настройки доступа пользователей к аудиторному фонду	54
4.4.	Аутентификация пользователей	55
5.	Настройка параметров системы	58
5.1.	Общие параметры	58
5.2.	Параметры	60
5.3.	Работа с ускорителем	61
6.	Импорт данных	63
6.1.	Импорт данных из xml-файлов	63
6.1.1.	Файл настройки (corTable.xml)	64
6.1.2.	Объекты импорта	65
6.1.2.1.	Список объектов	65
6.1.2.2.	Пример файла для импорта (ContentOfLoad.xml)	66
6.1.3.	Подготовка и выгрузка данных	67
6.1.4.	Импорт данных	69
6.1.5.	Импорт из Галактики ERP	71
6.1.5.1.	Экспорт данных	71
6.1.5.2.	Импорт данных	74
6.2.	Импорт данных из ПКД	74
7.	Настройка типовых отчетов	82
7.1.	Шаблоны отчетов	82
7.2.	Дизайнер отчетов	84
7.2.1.	Структура шаблона	85
7.2.2.	Основные элементы управления	85
8.	Технологические операции	89
8.1.	Очистка БД	89
8.2.	Экспорт данных	89
8.3.	Корректировка данных	90
8.4.	Переход к новому учебному году	90
9.	Работа с редактором модели приложения	92
9.1.	Назначение и состав модели	92
9.2.	Структура модели	93
9.3.	Правила работы с моделью	94
9.4.	Восстановление первоначального вида	95
9.5.	Типовые пользовательские изменения	95

9.5.1. Настройка списочных форм	96
9.5.2. Настройка карточных форм	98
9.5.3. Настройка форм для выбора	99
10. Возможные проблемы и способы их устранения.....	101
11. Приложения.....	104
11.1. Контракты web-службы.....	104
11.2. Содержание объектов импорта.....	140

1. Назначение системы

Система **Галактика Расписание учебных занятий (РУЗ)** предназначена для автоматизации процесса составления расписаний учебных занятий в образовательных учреждениях высшего и среднего профессионального образования. Основными пользователями системы предполагаются сотрудники учебно-методических подразделений образовательных учреждений, ответственные за формирование расписания.

Система **Галактика Расписание учебных занятий** может использоваться как самостоятельное приложение, так и в комплексе с другими продуктами корпорации **Галактика**, (например с **Галактика Управление Вузом**), а также с системами других разработчиков. Таким образом, данные для расписания могут вводиться непосредственно в системе **Галактика Расписание учебных занятий** или импортироваться из внешних систем.

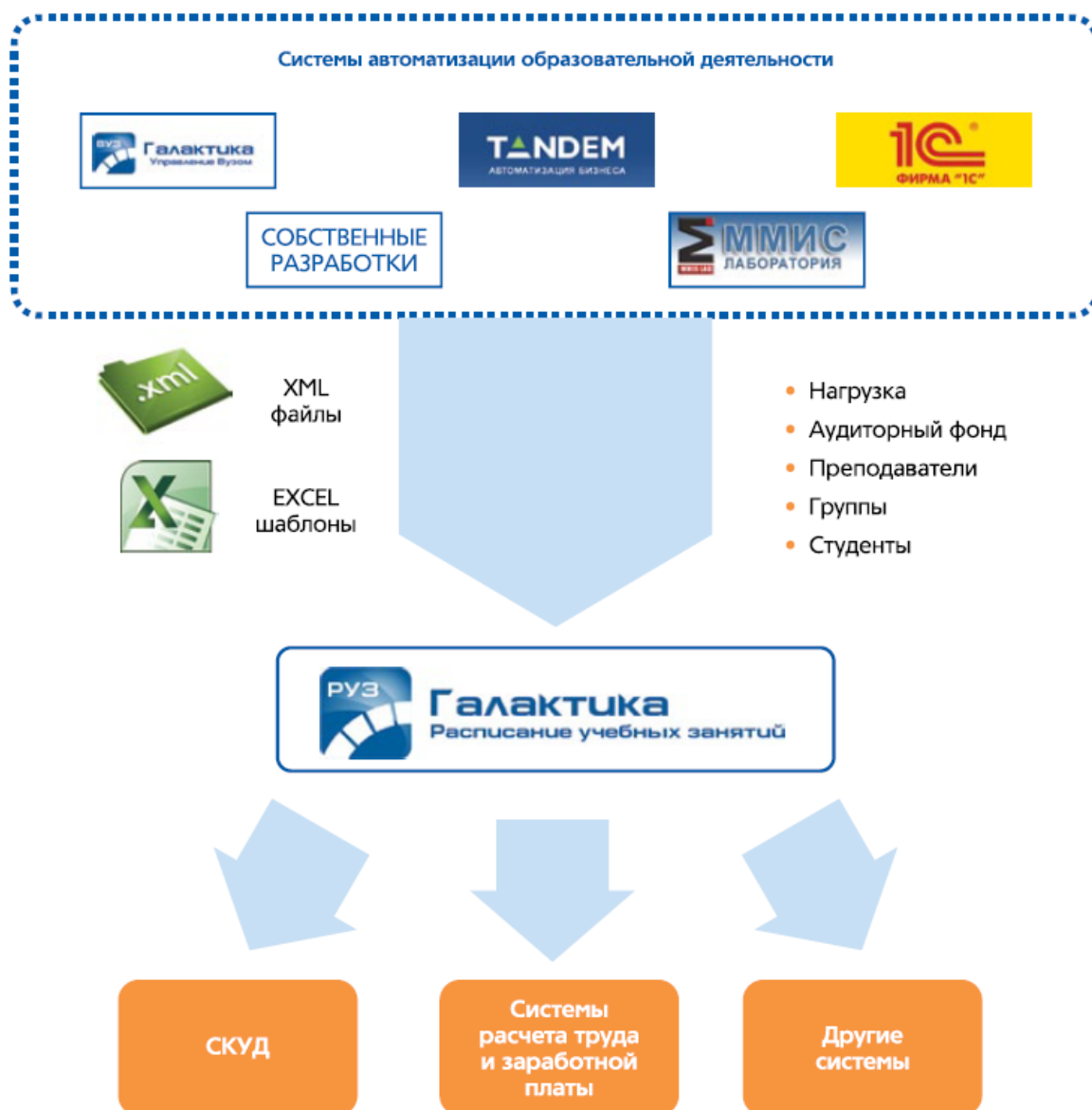


Рис. 1. Интеграция системы Галактика Расписание учебных занятий

Возможности системы **Галактика Расписание учебных занятий**:

Ведение всей справочной информации, необходимой для построения расписания (аудиторный фонд, преподаватели, факультеты, специальности и т. д.);

Инструмент формирования расписания учебных занятий:

- Эффективный конструктор расписания;
- Автоматическое формирование опорного расписания;
- Формирование расписания в автоматическом, ручном или комбинированном режимах;

Развитая система функциональных настроек:

- Контроль привязки аудиторного фонда к дисциплинам, видам работ, кафедрам, факультетам;
- Настройка приоритетов использования ресурсов в расписании (приоритеты дисциплин, преподавателей, сложность дисциплин и видов работ);
- Учет и контроль временной удаленности зданий, их приоритетов при назначении занятий;
- Поддержка различных объединений контингента (подгруппа; группа; поток), контроль их пересечения при составлении расписания;
- Гибкая система запретов использования ресурсов в расписании, включая пожелания преподавателей, доступность и бронирование аудиторного фонда;
- Поддержка нескольких распорядков звонков для различных зданий, расписаний, дней недели;
- Учет сменности в расписании;
- Регулирование продолжительности занятий;

Технология учета планов прохождения дисциплин:

- Ведение и использование графиков прохождения дисциплин;
- Технологические карты последовательности изучения дисциплин;
- Соблюдение требований интервала между определенными занятиями;
- Соблюдение требований медиа- и компьютерной оснащенности аудиторий для отдельных занятий;

Мощная аналитическая подсистема:

- Контроль, расчет и визуализация показателей эффективности расписания;
- Расчет интегральных показателей качества, сравнение расписаний;
- Подготовка и визуализация данных по загруженности аудиторий, преподавателей;
- Графическое представление информации по балансам часов при заменах в расписаниях, по аудиторному фонду, профессорско-преподавательскому составу;

Развитая система отчетности:

- Формирование отчетов с типовыми формами вывода сводных сведений из всех расписаний в разрезе аудиторий, групп, преподавателей;
- Прямой экспорт данных в Excel в специальные формы-шаблоны;
- Встроенный дизайнер отчетных форм

1.1. Термины и сокращения

Перечень используемых сокращений:

- **БД** — база данных;
- **ПКД** — промежуточный кластер данных в БД РУЗ, используемый для прямого взаимодействия с внешними системами;
- **РУЗ** — расписание учебных занятий;
- **СУБД** — система управления базами данных;
- **JSON** — JavaScript Object Notation, текстовый формат обмена данными, основанный на JavaScript;
- **IIS** — Internet Information Services, проприетарный набор серверов для нескольких служб интернета от компании Microsoft;
- **.NET** — программная платформа компании Microsoft;
- **REST** — Representational State Transfer, метод взаимодействия компонентов распределённого приложения в интернет-сети;
- **XAF** — eXpressApp Framework, технологическая платформа компании Developer Express для ускоренного создания .NET-приложений промышленного класса;
- **Xafari** — бизнес-платформа, основанная на платформе XAF и расширяющая ее возможности;
- **XPO** — eXpress Persistent Object, технология представления объектной модели в реляционной БД.

2. Общие принципы работы системы

Система **Галактика Расписание учебных занятий** реализована на основе программных и технологических платформ .Net, XAF, XPO, Xafari.

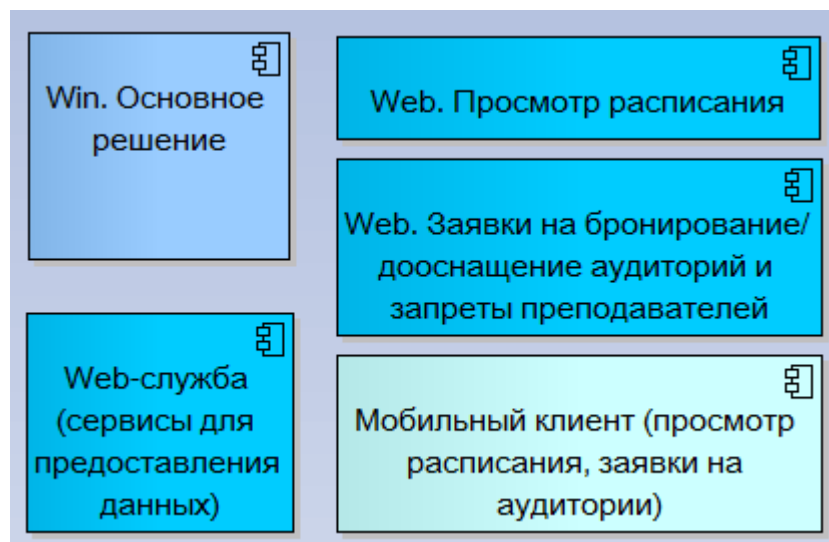


Рис. 2. Структура решения

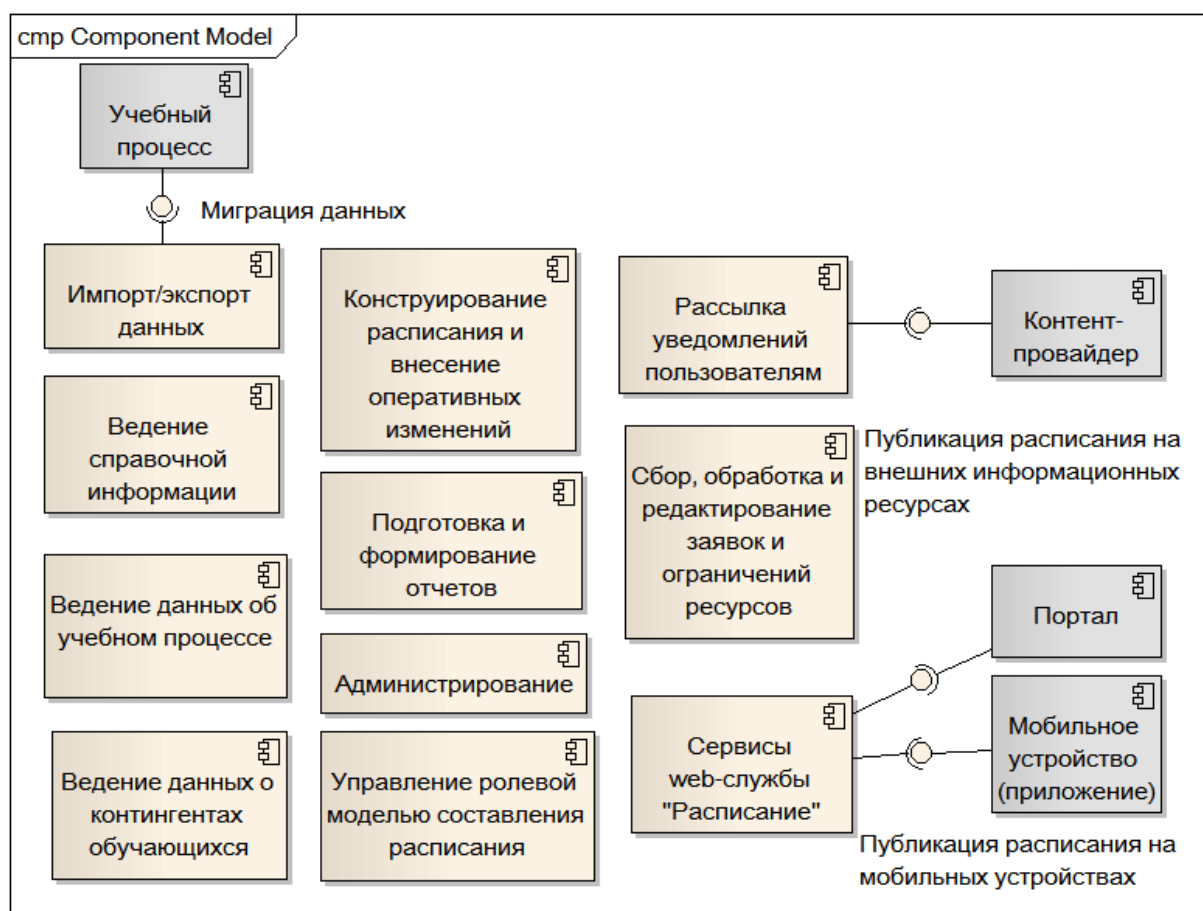


Рис. 3. Диаграмма компонентов системы

Основное решение реализовано в настольном windows-исполнении.

Web-служба **расписания** — набор сервисов для предоставления данных внешним приложениям, взаимодействие осуществляется на основе метода REST и формата JSON.

Подробное описание используемых контрактов служб приведено в [приложении](#)¹⁰⁴. Кроме того, описание при запущенной службе можно увидеть в браузере, введя соответствующий адрес.

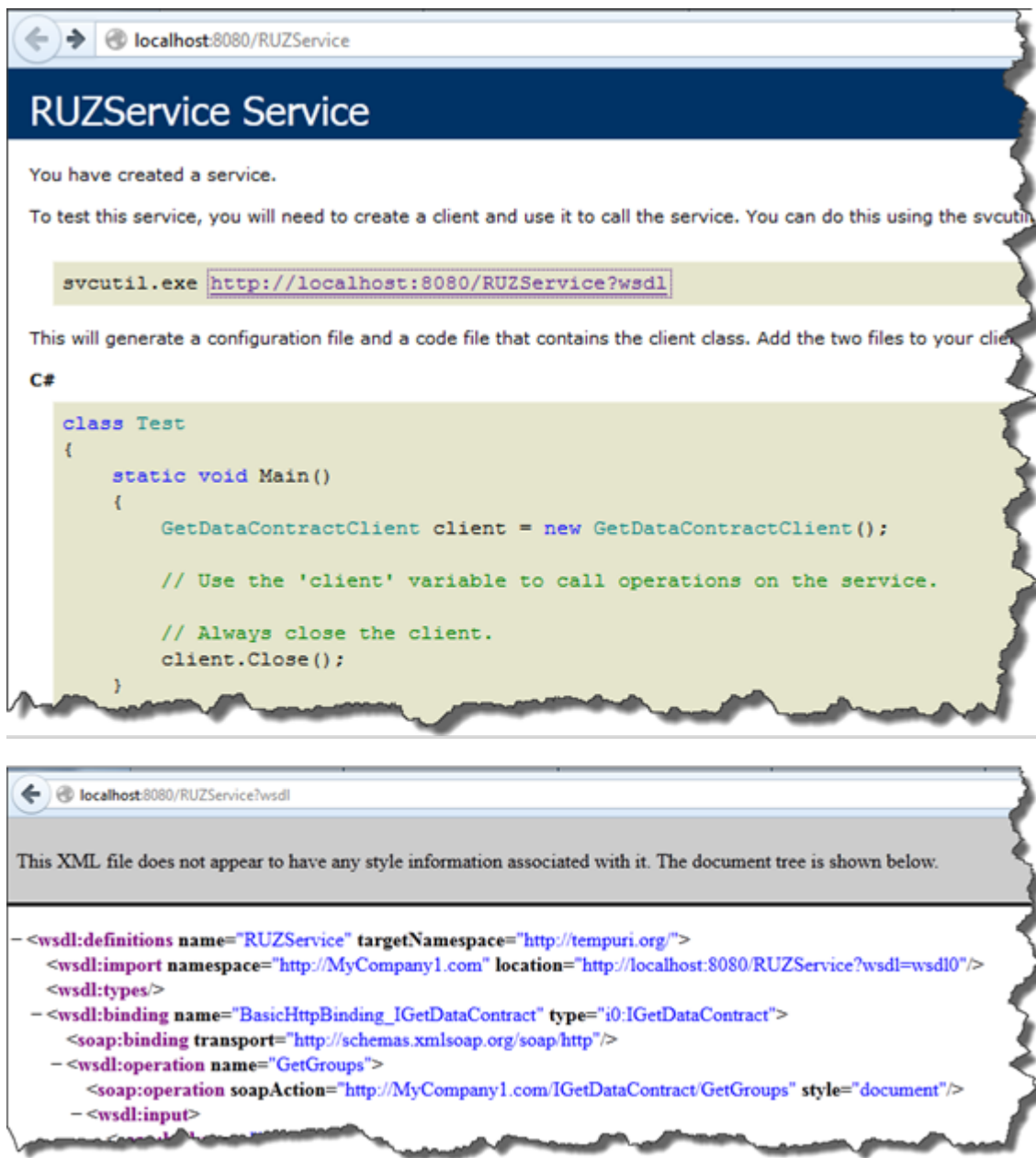


Рис. 4. Просмотр описания контрактов служб

Web-приложение "Просмотр расписания" обеспечивает возможность просмотра готового расписания в web-интерфейсе, не требует авторизации.

Web-приложение "Заявки на бронирование/дооснащение аудиторий и запреты преподавателей" обеспечивает возможность в web-интерфейсе оформлять заявки на бронирование/дооснащение аудиторий и ведение ограничений преподавателей по использованию их в расписании, требует авторизации.

Мобильный клиент обеспечивает возможность просмотра готового расписания на мобильном устройстве и оформление заявки на бронирование/дооснащение аудиторий, поддерживает платформы Android, iOS и Windows Phone.

3. Развертывание системы

3.1. Лицензирование

3.1.1. Общие сведения

Система лицензирования в многопользовательском режиме состоит из элементов:

- Сервер лицензий - это компьютер в локальной сети, на котором установлено программное обеспечение сервиса лицензий. Сервером лицензий может выступать тот же компьютер, на котором установлена система **Галактика Расписание учебных занятий**.
- На сервере лицензий находится лицензионный файл. Этот файл выписывается поставщиком конечного приложения на основе локального ключа, созданного с помощью специального приложения, являющегося частью системы **Галактика Расписание учебных занятий**. Путь к лицензионному файлу прописывается в конфигурационном файле сервиса лицензий.
- На рабочей станции или терминальном сервере установлено конечное приложение.
- На рабочей станции или терминальном сервере в папке, где установлено конечное приложение, должен находиться лицензионный файл (*.xlic) (для версий, начиная с 3.5.0.X, лицензионный файл и его компоненты вынесены в отдельный каталог LiscenseServer) в котором прописан адрес сервиса лицензий, или адрес сервиса лицензий может указываться в конфигурационном файле приложения.

Настроенный и запущенный сервис лицензий – это обязательный элемент системы при работе в многопользовательском режиме. Без него конечное приложение работать не будет. Клиентское приложение должно иметь доступ к сервису лицензий по протоколу TCP/IP.

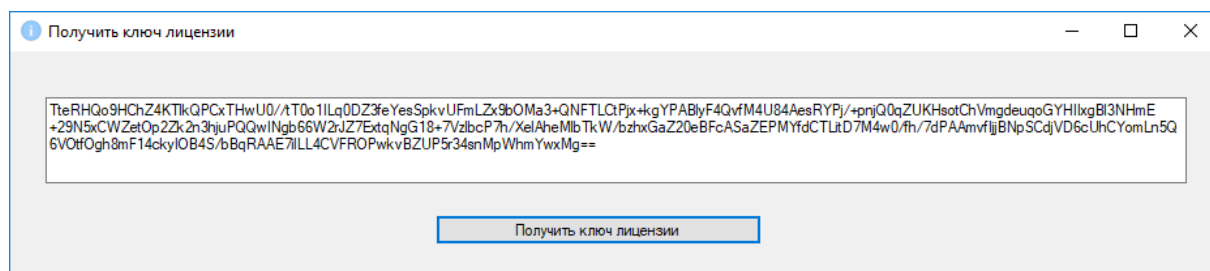
В локальном (однопользовательском, демонстрационном) режиме лицензионный файл располагается непосредственно в папке приложения или встроен в само приложение.

3.1.2. Генерация локального ключа лицензирования

Для того, чтобы поставщик прикладного решения системы мог выписать лицензионный файл, необходимо на сервере лицензий создать специальный локальный ключ лицензии, на основе которого будет в дальнейшем создан основной файл лицензии.

Локальный ключ лицензии создается с помощью приложения **GetMachineID.exe** входящего в состав установщика РУЗ.

Файл приложения  **GetMachineID.exe** необходимо запустить из папки **Application**, после того как появится пустое окно, нажать кнопку **Получить ключ лицензии**.



В окне появится сгенерированный с помощью специального алгоритма ключ. Алгоритм создания ключа использует привязку к сетевым IP-адресам сервера и MAC-адресам сетевой карты, поэтому воспользоваться им можно только на том сервере, с которого выполнен запуск программы **GetMachineID.exe**.

Ключ необходимо скопировать и любым способом передать поставщику прикладного решения, например, по электронной почте ruz@galaktika-it.ru. В письме необходимо указать полное название учебного заведения, а также вложить предыдущий файл лицензии Galaktika.RUZ.XLIC, если он уже выписывался ранее.

3.1.3. Создание файла лицензии

Лицензионный файл - это текстовый файл, который предоставляется поставщиком (лицензиаром) заказчику (лицензиату). В лицензионном файле указана информация о продукте, типе пакета поставки (основном модуле), способе использования и ограничениях на количество пользователей. Эти параметры определяются договором между поставщиком и заказчиком.

Лицензионный файл создается поставщиком прикладного решения системы **Галактика Расписание учебных занятий** на основе переданного ему локального ключа лицензии.

Лицензионный файл может содержать:

- название учебного заведения
- название лицензируемого продукта
- информацию о привязке к месту своего использования (адрес сервиса лицензий и т.д.)
- дату окончания действия лицензии
- дату окончания срока сопровождения
- другие лицензионные ограничения

Важно! При попытке изменить лицензионный файл, файл станет некорректным и не будет восприниматься системой лицензирования. Если по каким-то причинам Вам требуется изменить лицензионный файл, то вы должны обратиться к поставщику ПО для получения нового файла.

Лицензионный файл имеет примерно следующий вид:

```
Format=XI
ID=11
Licensee= ЗАО «Тестовая организация»
Licensor=ЗАО «Галактика Центр»
Product=Galaktika.EAM
Server=dpp-822.topsoft.local:33444
Signature=WX8gojAiBO3gt9W/nl04yfqBFwXRx6GqzO8bRVqEUeIsL+Q3rguC2vsHdYP
YkE+os0zmqQ2ND/hi
Y0/QCI+VoG5N9zPR8uAOlg5mnVV6LVxiA527YyUUeTq10f0elq5Di2qYIMi8wuYsFfx3l
hYF24XUJTNkqCit/JEDfeJueN8qwQDbPI2UgIQRo/03pRH2RYXhX9Yv7A3q0inxQ==#
Выпуска лицензии 2(1992452610060513)@3
SupportExpired=15/07/2014
Token=BgAAABSU0ExAAgAAAEAAQBbXYh4kmrQGIt3gQlFSX1Eh2D7wTwKj0YOBpq
zK7VU8rKWk613EQm8u
WJuoOrgAgXHXzN5DIIfXiLeziSmWAqskRVmXn2R2t14i0hgtzwap0cGp0nIzwgYOTVZK
VIFvkCq/9Ny4dj6aVmR7xMbydAHDA5f/D5KtlrK1BDbMkBfIX6TgPYJFil4tZAHt7C5eqi
5XD+/mwhw8pH2us3ZKgINbHwVuf12AG2hQgDqPNgNIG2cWUui#10077@3
-ConcurentConnections-
```

Лицензионный файл может быть передан на любом носителе, а также выслан по почте, как вложение. Лицензионный файл может иметь любое имя, но, как правило, имеет название **Galaktika.RUZ.XLIC**. При работе в многопользовательском режиме лицензионный файл должен находиться на сервере лицензий. В локальном (однопользовательском) режиме лицензионный файл располагается непосредственно в папке приложения. Для версий старше 3.5.0.X лицензионный файл и его компоненты перенесены в отдельный каталог LiscenseServer.

3.1.4. Схема установки и запуска конечного приложения

Многопользовательский режим

Для того, чтобы установить и запустить конечное приложение, необходимо:

1. Получить у поставщика инсталлятор системы.
2. Создать локальный лицензионный ключ с помощью утилиты **GetMachimeID.exe**.
3. Передать поставщику решения локальный лицензионный ключ.
4. Получить у поставщика лицензионный файл.
5. Развернуть сервис лицензий на одном компьютере (сервере лицензий) в локальной сети.
6. На сервере лицензий расположить лицензионный файл, сконфигурировать и запустить сервис лицензирования с помощью командной строки.
7. Запомнить или записать адрес сервиса лицензий (имя компьютера сервера лицензий и номер порта).
8. Установить на рабочие станции (или терминальный сервер) конечное приложение, указывая при каждой установке адрес сервиса лицензий или указать адрес сервиса лицензий в конфигурационном файле приложения вручную.

Для корректной работы конечное приложение должно получить информацию о лицензии. В случае многопользовательской работы конечное приложение должно иметь подключение к сервису лицензий. Каждое конечное приложение имеет свой конфигурационный файл. Название конфигурационного файла состоит из имени исполняемого файла (включая расширение **exe**) и добавления к нему расширения **.config**. Например, если основной исполняемый файл конечного приложения называется **Galaktika.EAM.Win.exe**, то его конфигурационный файл будет располагаться в том же каталоге и называться **Galaktika.EAM.Win.exe.config**.

Система лицензирования использует следующую информацию из конфигурационного файла приложения (все параметры являются необязательными, при их отсутствии используется значение по умолчанию):

Раздел configuration\appSettings:

```
<appSettings>
  <add key="XafariLicenseFile" value="License.xlic"/>
  <add key="XafariLicenseServer" value="localhost:33444 "/>
  <add key="XafariDebug" value="no"/>
</appSettings>
```

Параметр **XafariLicenseFile** определяет лицензионный файл, используемый конечным приложением. Допустимо указывать полный путь к файлу или просто имя файла, который в этом случае ищется в том же каталоге, что и конфигурационный файл. Если

параметр *XafariLicenseFile* не задан, то лицензионный файл ищется с именем *License.xlic*.

Параметр *XafariLicenseServer* задает адрес сервиса лицензий. Адрес сервиса лицензий состоит из полного имени компьютера, на котором установлен сервис лицензий и номера порта сервиса лицензий. Например, *dpp-822.topsoft.local:33444*. При отсутствии этого параметра, система лицензирования пытается подключиться по порту 33444 к сервису лицензирования, установленному на локальном компьютере.

Параметр *XafariDebug* используется исключительно в разработке и предназначен для отключения автоматического освобождения лицензии при останове приложения и обработки некоторых исключений, вызванных остановками приложения при его отладке. Использование этого параметра в процессе реальной работы приложения настоятельно не рекомендуется, так как вместо корректных сообщений об ошибке приложение будет зависать, и на время зависания лицензия будет считаться захваченной. Для включения режима отладки задайте значение параметра “XafariDebug” в “yes” как для сервиса лицензий, так и в конечном приложении.

9. Запустить конечное приложение и убедиться, что не возникает ошибок подключения к сервису лицензирования.

Если по каким-то причинам необходимо заменить лицензионный файл, нужно:

1. Создать новый локальный лицензионный ключ с помощью утилиты *GetMachimeID.exe*.
2. Передать поставщику решения локальный лицензионный ключ.
3. Получить от поставщика решения новый лицензионный файл.
4. Остановить сервис лицензий.
5. Заменить лицензионный файл в папке, в которой установлен сервер лицензий.
6. Запустить сервер лицензий.

Если по каким-то причинам необходимо перенести сервис лицензий на другой компьютер, то необходимо повторить все шаги по установке и настройке.

Локальный или демонстрационный режим

Система лицензирования конечного приложения не пытается подключиться к сервису лицензий в случае, если локальный лицензионный файл найден и в нем указан параметр:

LicenseMode=Local

или

LicenseMode=EvaluationLocal

В этом случае конечное приложение работает в однопользовательском режиме на основе локального лицензионного файла.

В этом случае система лицензирования работает в демонстрационном режиме с существенно ограниченной функциональностью. Конкретный перечень ограничений демонстрационного режима зависит от конечного приложения, вплоть до полного запрета запуска.

3.1.5. Установка и настройка сервиса лицензий

Важно! Для установки и управления сервисом лицензий Вы должны обладать администраторскими привилегиями на данном компьютере (сервере лицензий).

При работе в локальном (однопользовательском, демонстрационном) режиме лицензионный файл располагается в папке конечного приложения и установка сервиса лицензий не требуется.

ПО сервиса лицензий поставляется в составе конечного приложения и состоит из следующих файлов:

- ***Xafari.Licensing.Light.exe*** – исполняемый файл сервиса лицензий
- ***Xafari.Licensing.Light.exe.config*** – конфигурационный файл сервиса лицензий

Лицензионный файл может иметь любое имя. Как правило, используется расширение ****.xlic***. По умолчанию система лицензирования ищет лицензионный файл в папке установки сервиса с именем ***Galaktika.RUZ.xlic*** для версий старше 3.5.0.X. Начиная с версии 3.5.0.X, лицензионный файл и его компоненты вынесены в отдельный каталог ***LicenseServer***.

Конфигурационный файл сервиса лицензий

Файл с именем ***Xafari.Licensing.Light.exe.config*** расположен в том же каталоге, где развернут сервис лицензий. В этом конфигурационном файле в разделе **<appSettings>** указывается адрес сервиса лицензий и путь к лицензионному файлу, например:

```
<appSettings>
  <add key="XafariLicenseFile" value="D:
\Galaktika\License.xlic" />
  <add key="XafariLicenseServer" value="by01-
945.topsoft.local:33444" />
  <add key="XafariDebug" value="no"/>
</appSettings>
```

Параметр ***XafariLicenseFile*** определяет лицензионный файл, используемый сервисом лицензий. Допустимо указывать полный путь к файлу или просто имя файла, который в этом случае ищется в том же каталоге, что и конфигурационный файл.

Параметр ***XafariLicenseServer*** задает адрес сервиса лицензий.

Адрес сервера лицензий – это полное имя компьютера, на котором установлен сервер с указанием имени порта **<имя компьютер>:<порт>**. Адрес сервиса должен быть задан в конфигурационном файле сервера лицензий.

Параметр ***XafariDebug*** используется исключительно в разработке и предназначен для отключения некоторых ошибок, вызванных остановками приложения при его отладке. Использование этого параметра в процессе реальной работы приложения настоятельно не рекомендуется, так как вместо корректных сообщений об ошибке приложение будет зависать, и на время зависания лицензия будет считаться захваченной. Для включения режима отладки задайте значение параметра ***XafariDebug*** как ***yes*** - как для сервиса лицензий, так и в конечном приложении.

3.1.6. Запуск и удаление сервиса лицензий из командной строки

Важно! Все параметры сервиса лицензий берутся из конфигурационного файла *Xafari.Licensing.Light.exe.config* сервиса лицензий.

При запуске исполняемого файла ***Xafari.Licensing.Light.exe*** из командной строки в окне консоли будет отображен список поддерживаемых параметров командной строки:

- **-i** - Инсталлировать и стартовать сервис (если сервис с таким же именем уже существует, то предварительно он будет удален из системы)
- **-u** - Удалить сервис из системы

Например, для установки сервиса лицензий необходимо выполнить команду:

```
> Xafari.Licensing.Light.exe -i
```

Кроме того, сервис лицензий является службой Windows и им можно управлять посредством стандартных команд Windows (например, net.exe, PowerShell.exe)

3.1.7. Возможные проблемы и их устранение

При некорректной настройке, отсутствии допустимой лицензии или попытках обойти систему контроля лицензий возникают ошибки. По возможности система лицензирования сообщает максимально детальную информацию о произошедшей ошибке.

Ниже указаны наиболее часто встречающиеся проблемы описаны способы их устранения.

При невозможности самостоятельно устранить ошибку, необходимо обратиться в службу поддержки поставщика ПО, предоставив детальную информацию о проблеме, включая:

- Описание последовательности действий, приведших к ошибке
- Лицензионный файл
- Имя компьютера сервера лицензий и информацию об установленных на нем программах
- Конфигурационный файл сервиса лицензий
- Конфигурационный файл конечного приложения
- Скриншоты и тесты ошибок
- Log-файлы сервиса и приложения

Превышено количество одновременных подключений

Сообщение

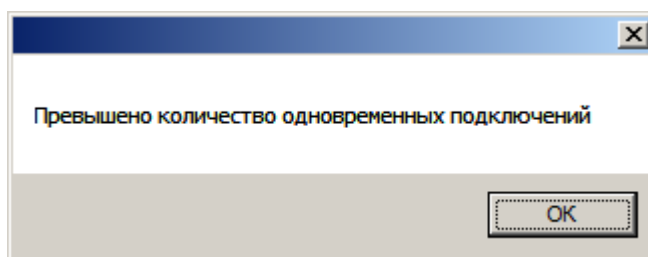


Рис. 6. Превышено количество одновременных подключений

Когда возникает

В момент запуска конечного приложения

Что означает

Пользователь пытается зайти в систему, когда в приложении уже работает максимально допустимое количество пользователей. Все конкурентные лицензии, количество которых указано в лицензионном файле, уже захвачены.

Причины и устранение

Необходимо дождаться, пока кто-то из пользователей покинет систему

Статистика подключений пользователей, в т.ч. при неудачной попытке подключения, отображается в [Журнале логирования](#)⁵⁷.

Нет доступа к сервису лицензий

Сообщение

Не удалось получить лицензию с сервера лицензий по адресу:localhost:33444

License server error

Когда возникает

В момент запуска конечного приложения

Что означает

Сервер лицензий не доступен

Причины и устранение

Неверная настройка адреса сервиса лицензий. Задать правильный адрес в конфигурационном файле приложения.

Доступ к сервису лицензий блокируется брандмауэром Windows. Попробовать запустить приложение на том же компьютере, где установлен сервис лицензий. Если приложение запускается, то необходимо отключить или настроить брандмауэр Windows.

Сервер лицензий по данному адресу не запущен или в нем возникают ошибки. Настроить и запустить сервер лицензий.

Сервис лицензий не запускается

Сообщение

Лицензионный файл не найден

Лицензионный файл некорректен

Активная лицензия не найдена

Ключ не найден

Срок действия лицензии истек. Пожалуйста, обратитесь к производителю за обновленной лицензией

Обнаружено наличие нескольких экземпляров лицензионного сервера для одного и того же криптоконтейнера.

Период сопровождения закончился. Вы не имеете права использовать продукты, выпущенные после окончания периода сопровождения.

Возникла непредвиденная ошибка при проверке лицензии. Возможно, производились попытки обойти лицензионную защиту.

Когда возникает

В момент запуска сервиса лицензий

Что означает

Возникли проблемы проверки корректности лицензии в соответствии с текстом сообщений.

Причины и устранение

Не найден программный ключ, указанный в лицензионном файле – необходимо проверить наличие лицензионного файла и его название

Лицензионный файл отсутствует или поврежден – установить на сервер лицензий лицензионный файл, полученный от поставщика

Некорректные настройки системы лицензирования – задать правильные настройки в конфигурационном файле сервиса лицензий. При невозможности самостоятельно устранить ошибку необходимо обратиться в службу поддержки поставщика ПО.

3.2. Win. Основное решение

3.2.1. Требования к программному и техническому обеспечению

Требования к программному обеспечению

Требования к программному обеспечению [сервера](#):

- Windows Server 2003/2008/2012;
- .NET Framework v.4.5.1 и выше;
- СУБД.

Требования к программному обеспечению [рабочей станции](#):

- Windows (XP SP3, 7, 8/8.1, Server 2003/2008/2012);
- .NET Framework v.4.5.1 и выше;
- клиент СУБД (устанавливается при необходимости).

При использовании СУБД MS SQL Server возможно применение SQL Server 2005/2008/2012/2014/2016.

При использовании СУБД Oracle клиент Oracle должен быть той же разрядности что и ОС, т. е. если ОС 32-разрядная, то и клиент Oracle должен быть 32-разрядным, если ОС 64-разрядная, то и клиент Oracle должен быть 64-разрядным (Oracle 10g, Oracle 11g, Oracle 11R2).

При использовании Postgres Pro возможно применение версий 9, 10, 11.

Требования к техническому обеспечению

Требования к техническому обеспечению [сервера СУБД](#):

- процессоры, объем ОЗУ, объем и количество жестких дисков выбираются в зависимости от предполагаемой нагрузки;
- две сетевые карты 100Mb/1Gb — Ethernet; установленный и сконфигурированный протокол TCP/IP;
- рекомендуется дисковая подсистема с поддержкой технологии RAID;
- предпочтительно использование дисков, обеспечивающих повышенную скорость случайного чтения (SAS, SCSI) с высокой скоростью вращения (10000-15000 rpm);
- память должна поддерживать технологию контроля четности;
- бесперебойное питание;
- требования к аппаратному обеспечению должны соответствовать рекомендуемым требованиям, предъявляемым для установки используемой СУБД.

Требования к техническому обеспечению [рабочей станции](#):

- процессор с архитектурой Intel I13 и выше с тактовой частотой не менее 1,8 GHz и частотой внутренней шины не менее 400 MHz;
- ОЗУ — рекомендуется 8 Гб и выше;
- манипулятор типа "мышь" или сенсорный экран;
- клавиатура;
- монитор цветного изображения SVGA с поддержкой видеорежима с глубиной цвета не ниже HiColor (65536 цветов). Рекомендуемое разрешение — не менее 1920x1080 пикселей;
- сетевые карты (при серверном использовании). Минимальные требования = 100Мб. Рекомендуемые требования = 1Гб.

Рекомендации по архитектуре развертывания

При развертывании системы рекомендуется:

- продукт разворачивать на сервере приложения;
- доступ к приложению осуществлять с рабочих станций пользователей.

В этом случае:

- требования к техническим характеристикам рабочей станции могут быть снижены;
- при обновлении версии продукта потребуется его обновление только на сервере приложения.

3.2.2. Основные варианты развертывания

Возможны два варианта развертывания системы *Галактика Расписание учебных занятий*:

- однопользовательский;
- многопользовательский.

Однопользовательский режим предполагает установку системы на локальном рабочем месте пользователя. Этот вариант при установке не требует специальных знаний администрирования (в случае использования СУБД MS SQL) и для его реализации достаточно выполнить установку посредством запуска инсталлятора. Целесообразно применение для ознакомительных целей.

При многопользовательском (серверном) развертывании возможны следующие варианты:

- [система устанавливается на каждом клиентском компьютере](#)^[21], настраивается доступ к общей БД;
- [система устанавливается на общем сервере](#)^[21], при этом все пользователи запускают систему с сервера из общего каталога.

Вариант серверного развертывания должен выполняться подготовленным специалистом (администратором), т. к. при этом требуются специальные знания, навыки и опыт администрирования используемой СУБД.

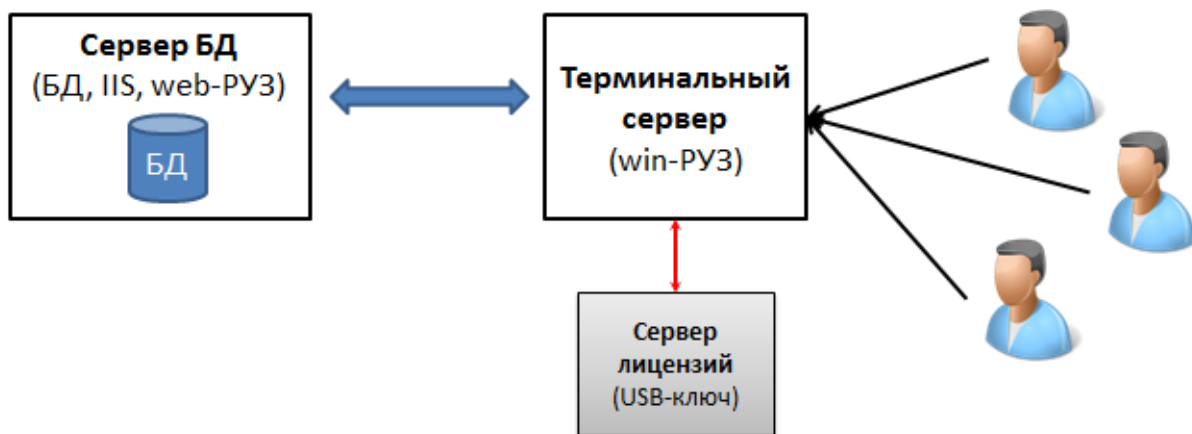


Рис. 7. Типовой вариант архитектуры развертывания системы

Типовой вариант серверного развертывания системы представлен на рисунке выше. При таком использовании рекомендуется на терминальном сервере планировать на каждого пользователя, работающего с РУЗ, не менее 1 Гб оперативной памяти при одновременной работе.

3.2.2.1. Развертывание системы на клиентских компьютерах

Этот вариант развертывания менее предпочтителен для многопользовательской работы.

При таком варианте развертывания на каждом компьютере пользователя необходимо:

- установить клиентское приложение с помощью инсталлятора;
- настроить в конфигурационном файле строку соединения с БД (не требуется в случае однопользовательского использования при установке на локальном рабочем месте с установленной там же СУБД MS SQL).

Установка обновлений системы и патчей должна выполняться на всех компьютерах пользователей.

3.2.2.2. Развертывание системы на сервере

Этот вариант развертывания предпочтителен для многопользовательской работы.

При таком варианте развертывания необходимо:

- установить клиентское приложение с помощью инсталлятора на сервере;
- настроить в конфигурационном файле строку соединения с БД;
- раскомментировать элементы конфигурационного файла:


```
<add key="TraceLogLocation" value="CurrentUserApplicationDataFolder"/>
<add key="UserModelDiffsLocation" value="CurrentUserApplicationDataFolder"/>
```
- открыть доступ к папке с установленным продуктом;
- сообщить пользователям путь на общедоступную папку с приложением;
- выполнить настройку пользователей средствами СУБД и средствами системы **Галактика Расписание учебных занятий**.

Установка обновлений системы и патчей выполняется только на сервере.

3.2.2.3. Лицензионное использование

Вместе с продуктом поставляется:

- сервер лицензий;
- лицензионный файл.

Исполнение лицензионного соглашения обеспечивается сервером лицензий с установленным программным ключом.

Сервер лицензий — это один из компьютеров в сети, на котором установлено и функционирует ПО сервера лицензий, и, как правило, находится лицензионный файл.

В **лицензионном файле** (License.xlic) содержатся данные о продукте, типе пакета поставки, ограничения на количество конкурентных подключений пользователей и другие параметры, определенные договором между поставщиком и заказчиком.

Место расположения лицензионного файла указывается при инсталляции сервера лицензий и может быть изменено в процессе использования продукта с помощью специальной утилиты "Менеджер лицензий". Порядок работы с утилитой изложен в соответствующей инструкции.

 1) Редактирование содержания лицензионного файла не допускается. Любые его корректировки приводят к его невалидности и, как следствие, к отказу в запуске приложения.

2) При смене наименования образовательного учреждения необходимо запросить новую лицензию и предварительно (до ее использования) изменить в системе наименование с помощью функции **Сервис > Общие параметры**.

3.2.3. Установка системы

Установка системы **Галактика Расписание учебных занятий** выполняется с помощью инсталлятора Galaktika.RUZ-x.x.x.x.msi.

Для установки продукта выполните следующее:

- запустите на исполнение файл инсталлятора Galaktika.RUZ-x.x.x.x.msi;
- следуйте всем указаниям инсталлятора.

Варианты установки:

- [[Обычная](#)^[23]];
- [[Выборочная](#)^[24]];
- [[Полная](#)^[25]].

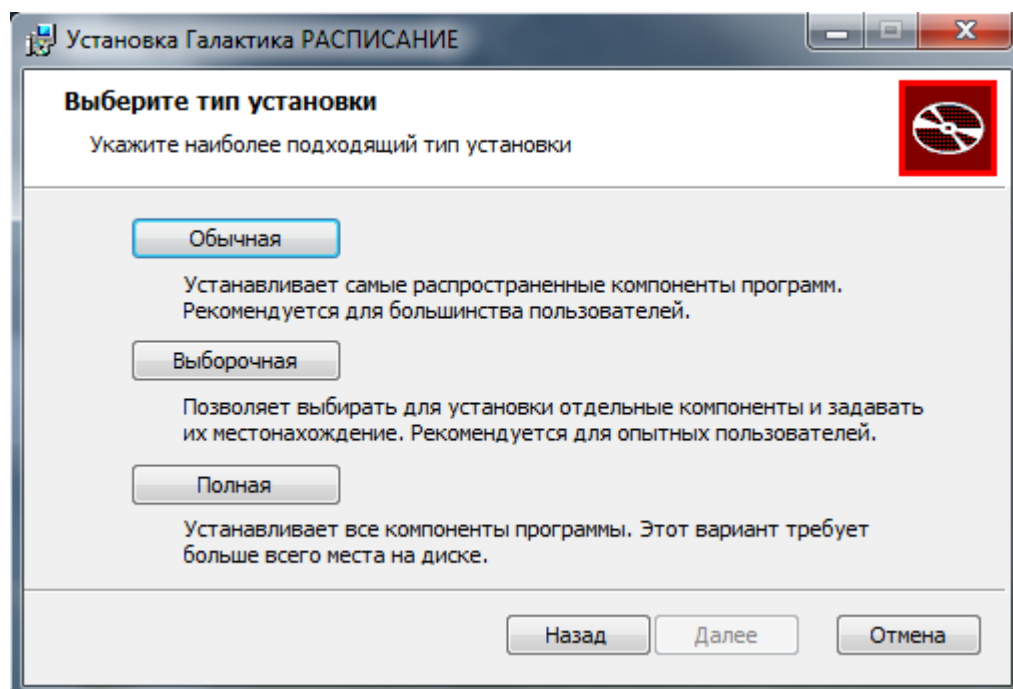


Рис. 8. Выбор варианта установки системы

Порядок установки предполагает предварительную установку сервера лицензий. Сервер лицензий доступен при вариантах установок [Выборочная] и [Полная].

3.2.3.1. Обычная установка

Обычная установка обеспечивает инсталляцию клиентского приложения. После выбора этого типа установки необходимо дополнительно указать:

- имя (IP-адрес) сервера лицензий;
- номер порта сервера лицензий.

В строках ввода указываются данные, полностью совпадающие с содержанием аналогичных параметров, заданных при установке сервера лицензий (см. [выборочный](#)^[24] или [полный](#)^[25] тип установки).

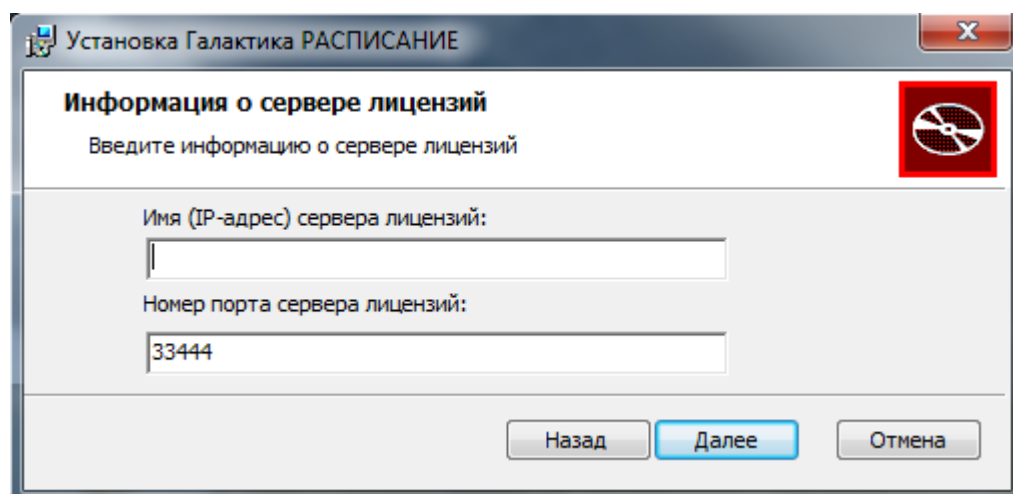


Рис. 9. Информация о сервере лицензий

После завершения установки в меню Пуск > Все программы появится раздел Галактика РАСПИСАНИЕ.

В этом разделе будут созданы ярлыки:

- *Расписание учебных занятий* — для запуска системы;
- *Руководство пользователя* — для открытия документа с описанием системы.

💡 Установка выполняется в каталог `C:\%ProgramFiles(x86)%\Galaktika`. Для возможности изменения каталога установки выберите тип установки *[Выборочная]*.

3.2.3.2. Выборочная установка

Выборочная установка обеспечивает инсталляцию:

- сервера лицензий;
- клиентского приложения.

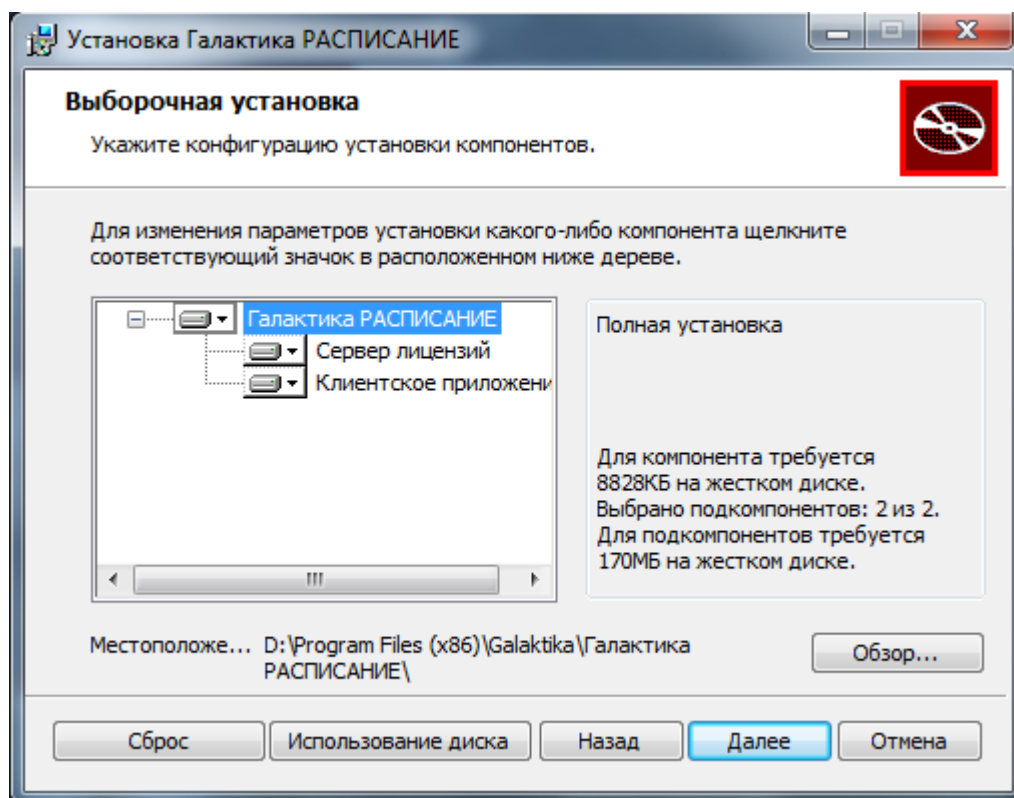


Рис. 10. Выбор компонентов для установки

По умолчанию выбраны для установки оба пункта. Для отказа от инсталляции какого-либо из пунктов необходимо выполнить соответствующее действие.

Для выбора каталога установки необходимо в дереве устанавливаемых компонентов установить курсор на уровень *Галактика РАСПИСАНИЕ* и нажать кнопку *[Обзор]*.

При переходе к следующему окну инсталлятора необходимо дополнительно указать:

- номер порта сервера лицензий;
- лицензионный файл.

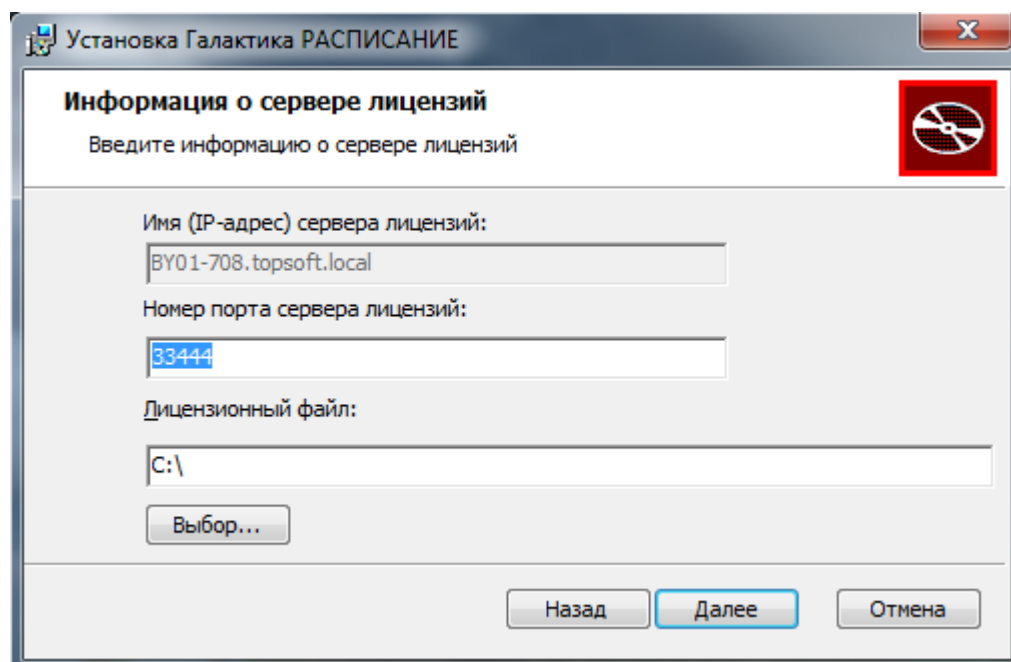


Рис. 11. Информация о сервере лицензий

Имя (IP-адрес) сервера лицензий при этом определяется автоматически и недоступно для изменения.

Для выбора лицензионного файла необходимо воспользоваться кнопкой [Выбор].

 Файл лицензий должен быть доступен системному пользователю ("Система"/"System"). Именно от этого пользователя работает менеджер лицензий.

После завершения установки в меню *Пуск > Все программы* появятся разделы, соответствующие выбранным пунктам установки.

3.2.3.3. Полная установка

Полная установка обеспечивает инсталляцию:

- сервера лицензий;
- клиентского приложения.

 Перед запуском установки рекомендуется скопировать лицензионный файл в каталог сервера.

При переходе к следующему окну инсталлятора необходимо дополнительно указать (аналогично [выборочной](#)^[24] установке):

- номер порта сервера лицензий;
- лицензионный файл.

Имя (IP-адрес) сервера лицензий при этом определяется автоматически и недоступно для изменения.

 Рекомендуется записать (сохранить) это имя — оно понадобится при установке клиентского приложения.

Для выбора лицензионного файла необходимо воспользоваться кнопкой [Выбор].

После завершения установки в меню *Пуск > Все программы* появится раздел *Галактика РАСПИСАНИЕ*, в котором будут созданы ярлыки:

- *Расписание учебных занятий* — для запуска системы;
- *Руководство пользователя* — для открытия документа с описанием системы;
- *Менеджер лицензий* — для запуска менеджера лицензий;
- *Менеджер лицензий Руководство пользователя* — для открытия специальной инструкции по использованию данного продукта.

 Установка выполняется в каталог *C:\%ProgramFiles(x86)%\Galaktika*. Для возможности изменения каталога установки выберите тип установки *[Выборочная]*.

3.2.4. Начало использования системы

3.2.4.1. Генерация БД

Генерация БД для приложения выполняется, после того как в конфигурационном файле настроена [строка соединения с БД](#)^[39].

Генерация БД для MS SQL Server и PostgreSQL

Для генерации БД необходимо запустить приложение **Галактика Расписание учебных занятий** и положительно ответить на запрос системы о создании/обновлении БД. Если БД с наименованием, указанным в конфигурационном файле, уже существует, то будет произведено обновление структуры существующей базы данных, в противном случае будет создана новая БД.

Условия генерации/обновления структуры БД:

- Если в строке соединения с БД указан доступ к SQL Server от имени Windows-пользователя, то пользователь, от имени которого запускается приложение, должен обладать правами на создание БД и на создание/изменение объектов БД (таблицы, индексы, процедуры и т. п.) на SQL Server.
- Если в строке соединения с БД задан SQL-пользователь для подключения к серверу, то этот пользователь должен обладать правами на создание БД и на создание/изменение объектов БД (таблицы, индексы, процедуры и т. п.) на SQL Server.

При успешном выполнении операции произойдет запуск приложения, никаких сообщений об ошибках не будет выдано.

Если приложению не удастся создать БД, то рекомендуется создать пустую БД на сервере вручную, а затем запустить приложение.

 Генерация БД для PostgreSQL осуществляется аналогично MS SQL Server.

Генерация БД для Oracle

 В настоящее время при работе с СУБД Oracle все пользователи системы **Галактика Расписание учебных занятий** работают с СУБД Oracle от имени одного и того же Oracle-пользователя (элемент *User ID* в строке соединения с БД). Таблицы приложения создаются в схеме этого пользователя. У пользователя должны быть права на создание таблиц, индексов, процедур и на работу с данными в таблицах.

Для генерации БД необходимо:

- создать нового пользователя в нужной БД Oracle (для этого пользователя будет создана сразу и схема);
- в конфигурационном файле в строке соединения с БД в элементах **User ID** и **Password** задать имя этого пользователя и пароль;
- запустить приложение, согласившись на создание/обновление БД (при запуске приложения одновременно будет сгенерирована БД).

При успешном выполнении операции произойдет запуск приложения, никаких сообщений об ошибках не будет выдано.

Генерация БД в XML-файле

У пользователя, от имени которого запускается программа, должны быть права на запись в каталоге приложения (в каталоге, где будет располагаться XML-файл данных).

Для генерации БД необходимо запустить приложение, согласившись на создание/обновление БД (при запуске приложения одновременно будет сгенерирована БД).

При успешном выполнении операции произойдет запуск приложения, никаких сообщений об ошибках не будет выдано.

3.2.4.2. Аутентификация входа

При запуске приложения необходимо пройти аутентификацию. Для этого в окне *Вход в систему* необходимо указать тип аутентификации: *Windows* или *Стандартная*.

При Windows-аутентификации имя пользователя специально не задается и берется автоматически из Active Directory ОС.

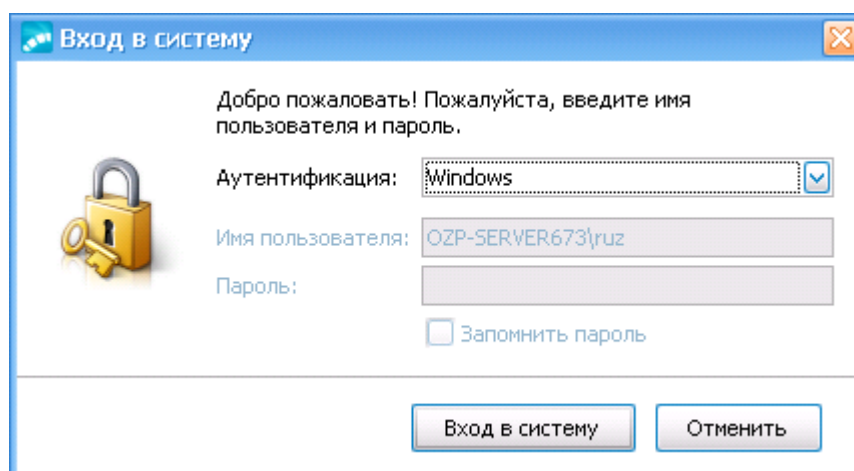


Рис. 12. Windows-аутентификация

В случае стандартной аутентификации необходимо указать имя пользователя и пароль.

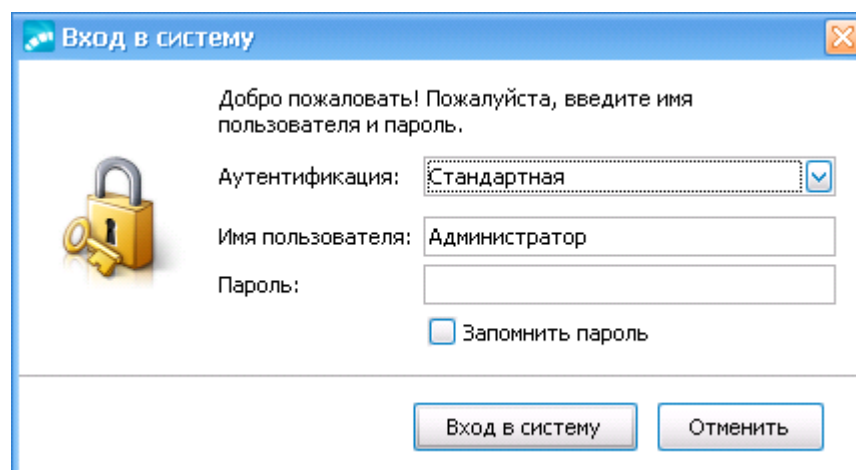


Рис. 13. Стандартная аутентификация

При установке системы или при создании новой базы данных автоматически генерируются [типовые пользователи](#)^[54] с [типовыми ролевыми правами](#)^[50].

 *Пользователь Администратор создается с предустановленным паролем "galaktika", после первого входа в систему пароль пользователя Администратор необходимо изменить. Начиная с версии РУЗ 3.x пустой пароль не допускается, если используется Стандартная аутентификация, перед обновлением системы необходимо обязательно задать непустой пароль пользователя Администратор.*

При однопользовательском использовании системы на локальном рабочем месте равнозначно без дополнительных настроек может использоваться для первого входа как Windows-аутентификация, так и стандартная под типовым пользователем (пароль не требуется).

При серверном использовании системы необходима [предварительная настройка](#)^[55] пользователей системным администратором и администратором приложения **Галактика Расписание учебных занятий**.

3.2.4.3. Загрузка демонстрационных данных

Предусмотрена возможность загрузки демонстрационных данных в систему **Галактика Расписание учебных занятий**. Демонстрационные данные поставляются в файлах <папка приложения>\Import*.xml.

Для загрузки демонстрационных данных необходимо выполнить импорт из XML-файлов:

- выбрать в главном меню системы **Галактика Расписание учебных занятий** функцию *Операции > Импорт*;
- выбрать для импорта все файлы из каталога <папка приложения>\Import.

После завершения импорта в БД будут находиться демонстрационные исходные данные, на основании которых далее можно конструировать расписание.

3.2.5. Обновление системы

Обновление системы поставляется в виде нового инсталляционного файла.

Порядок обновления:

- сохранить конфигурационные файлы системы из каталога ...\\Application (Galaktika.Win.exe.config, Galaktika.RUZ.Service.exe.config, Xafari.Licensing.Light.Activation.exe.config, Xafari.Licensing.Light.exe.config);
- создать резервную копию БД;
- в случае переименования образовательного учреждения и получения нового лицензионного файла — изменить в системе наименование с помощью функции *Сервис > Общие параметры*;
- [удалить](#)^[29] текущую версию системы;
- [установить](#)^[22] новую версию системы с помощью нового инсталляционного файла;
- восстановить сохраненные ранее конфигурационные файлы.

 В случае необходимости при первом запуске новой версии системой будет выдан запрос на подтверждение обновления БД. При положительном ответе все действия будут выполнены автоматически и система будет переведена в штатный режим эксплуатации.

3.2.6. Удаление системы

Удаление системы **Галактика Расписание учебных занятий** выполняется из меню *Пуск > Панель управления > Установка и удаление программ*. Для Windows 7 — из меню *Пуск > Панель управления > Программы и компоненты*. Для Windows 10 — из меню *Пуск > Параметры > Приложения*.

В окне *=Установка и удаление программ=* установите курсор на пункт **Галактика РАСПИСАНИЕ** и выполните для него команду *Удалить*.

После удаления системы на диске может остаться папка, в которую она была установлена. В ней могут находиться временные файлы, созданные во время работы приложения. Эту папку и файлы следует удалить вручную.

3.3. Web-служба расписания

Последовательность действий при развертывании web-службы приложения:

- 1) Получить каталог web-службы Galaktika.RUZ.Service, содержащий необходимые файлы для развертывания.
- 2) На сервере, где будет расположена web-служба:
 - убедиться что установлены компоненты: ASP.NET 4.0 или выше, HTTP Activation (при необходимости установить);
 - проверить наличие web-сервера, например IIS (Internet Information Services) (при необходимости установить);
 - выполнить [настройку IIS](#)^[30];
 - выполнить [дополнительные настройки на сервере](#)^[33];
 - настроить [конфигурационный файл web-службы](#)^[34];
 - [запустить web-службу](#)^[35].

См. также приложение "[Контракты web-службы](#)"^[104].

3.3.1. Настройка IIS

Настройка IIS осуществляется с помощью "Диспетчера служб IIS":

Сначала следует создать пул приложения (имя может быть произвольным).

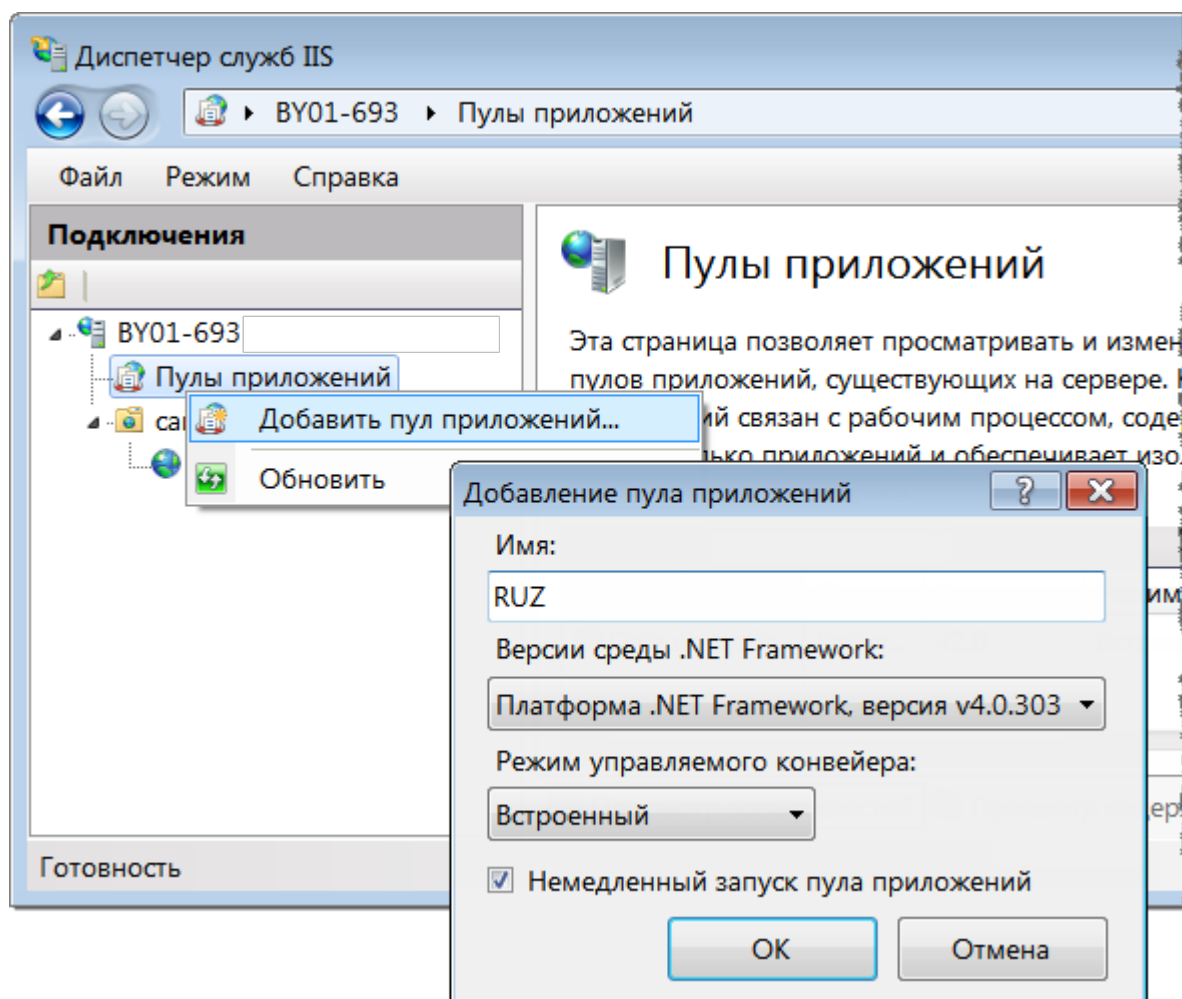


Рис. 14. Создание пула

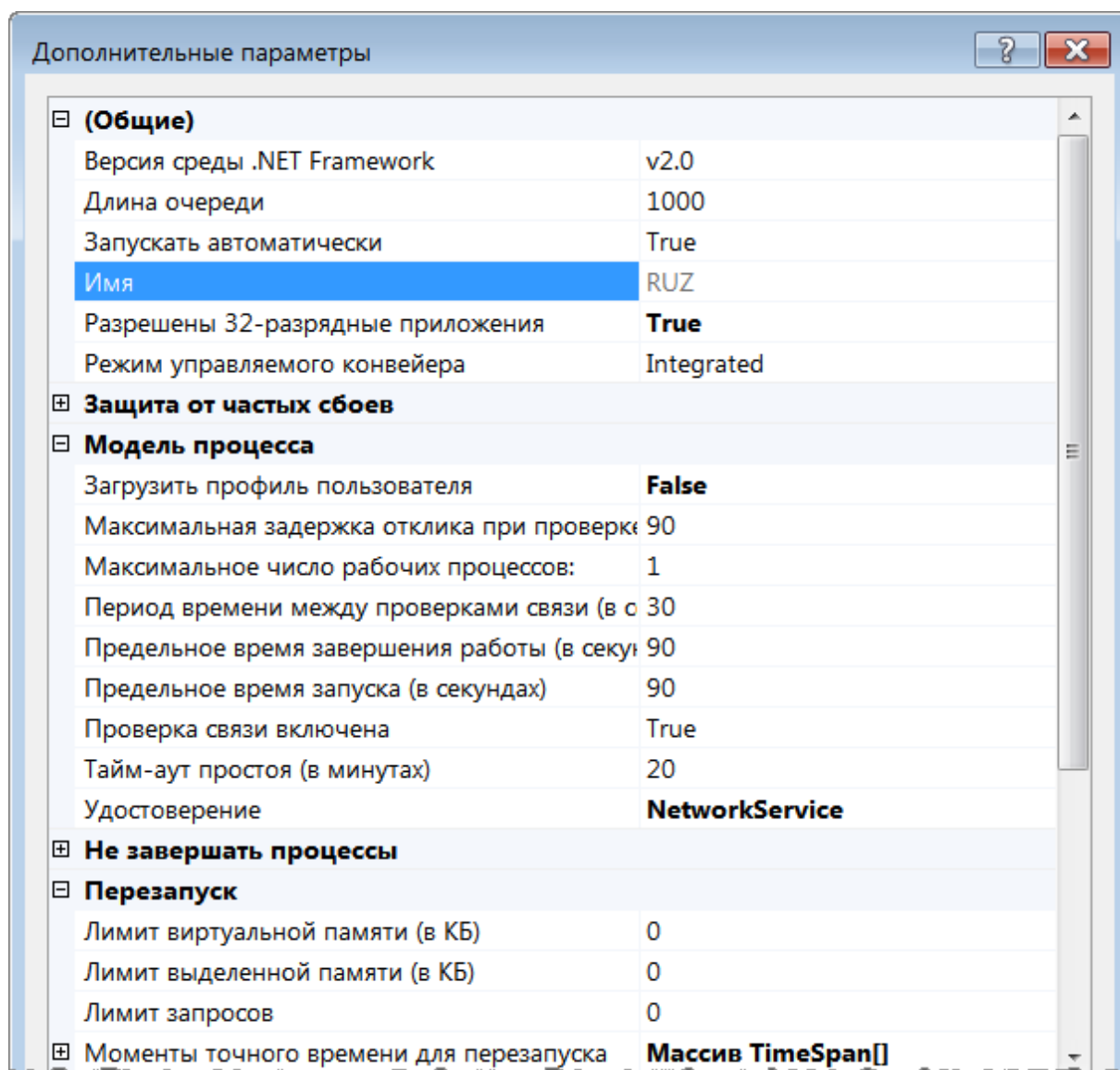


Рис. 15. Дополнительные параметры пула

Затем необходимо добавить web-приложение, указав для него созданный ранее пул.

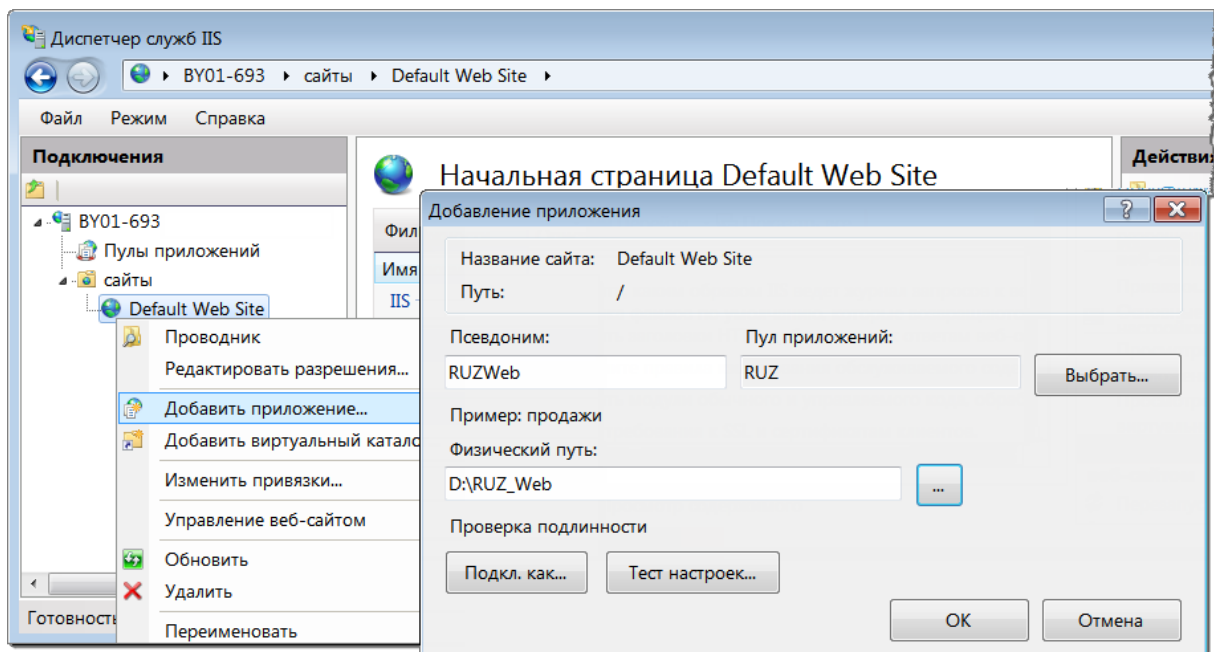


Рис. 16. Добавление web-приложения

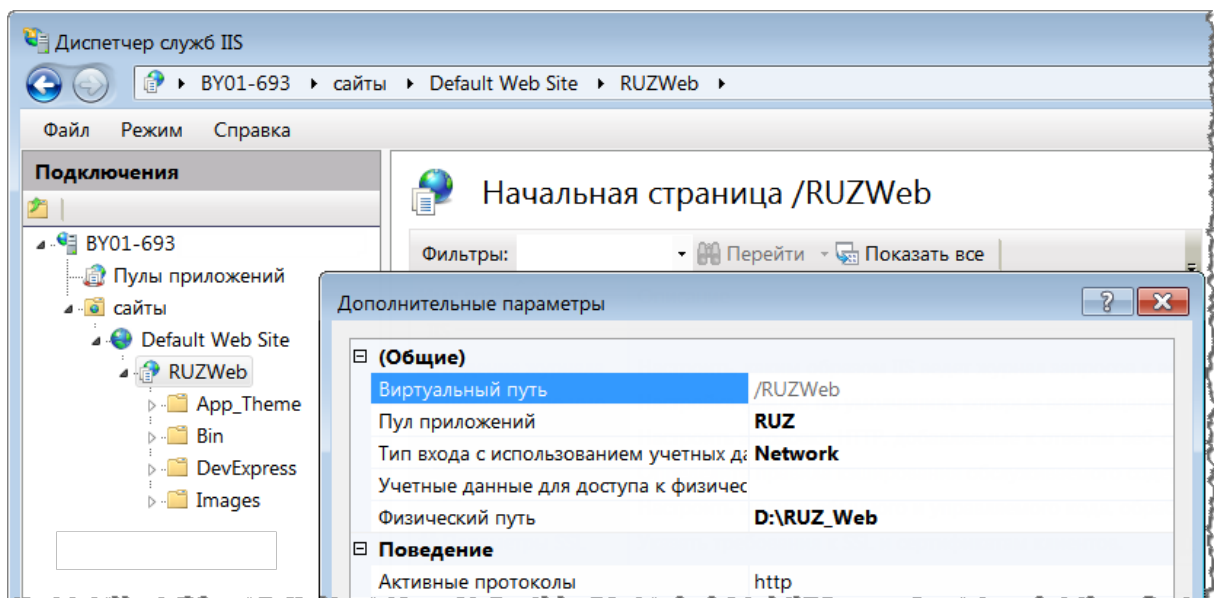


Рис. 17. Дополнительные параметры web-приложения

Далее следует открыть доступ к папке с сайтом, добавив пользователю IIS_IUSRS возможность записи и изменения содержимого данной папки.

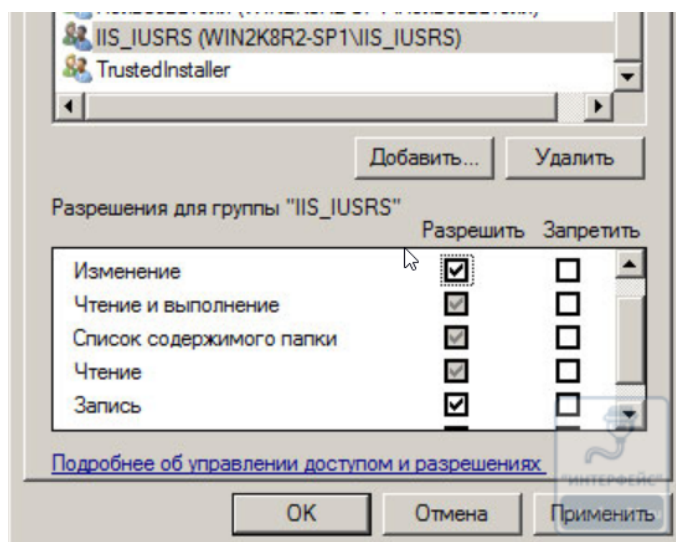


Рис. 18. Настройка прав доступа

На завершающем этапе необходимо выполнить настройку параметра **Удостоверение** (Identity), выбрав учетную запись *LocalSystem*.

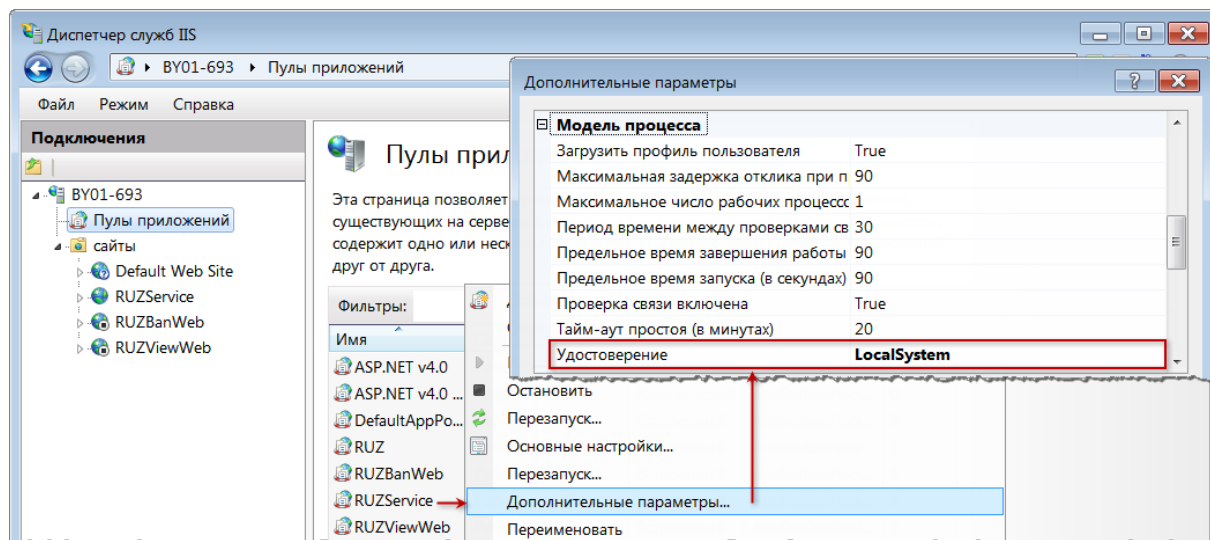


Рис. 19. Настройка дополнительных параметров для web-службы

Более подробную информацию о настройке IIS можно найти на сайтах:

- <http://msdn.microsoft.com/ru-ru/library/bb763174%28v=vs.90%29.aspx>;
- http://interface31.ru/tech_it/2012/05/windows-server-nastraivaem-veb-server-iis.html.

3.3.2. Дополнительная настройка сервера

На сервере, где расположена web-служба РУЗ, дополнительно необходимо:

- проверить открытость порта, через который выполняется связь с web-службой — для открытия необходимо выполнить команду `netsh http add urlacl url=http://+:8080/RUZService user=DOMAIN\user`, например: `netsh http add urlacl url=http://+:8080/RUZService user=topsoft.local\ivanov`;
- выполнить регистрацию приложений ASP.NET в службах IIS — в командной строке выполнить: `aspnet_regiis -i`, например: `C:\Windows\Microsoft.NET\Framework64\v4.0.30319\aspnet_regiis -i`;

- из каталога web-службы РУЗ запустить файл runcert.bat — создается сертификат, используемый при обмене данными между мобильным клиентом РУЗ и сервисами web-службы;
- настроить для системной учетной записи сервера СУБД права на БД РУЗ (для MSSQL это пользователь типа "NT AUTHORITY\SYSTEM"), указав серверные роли доступа к БД РУЗ public и db_owner.

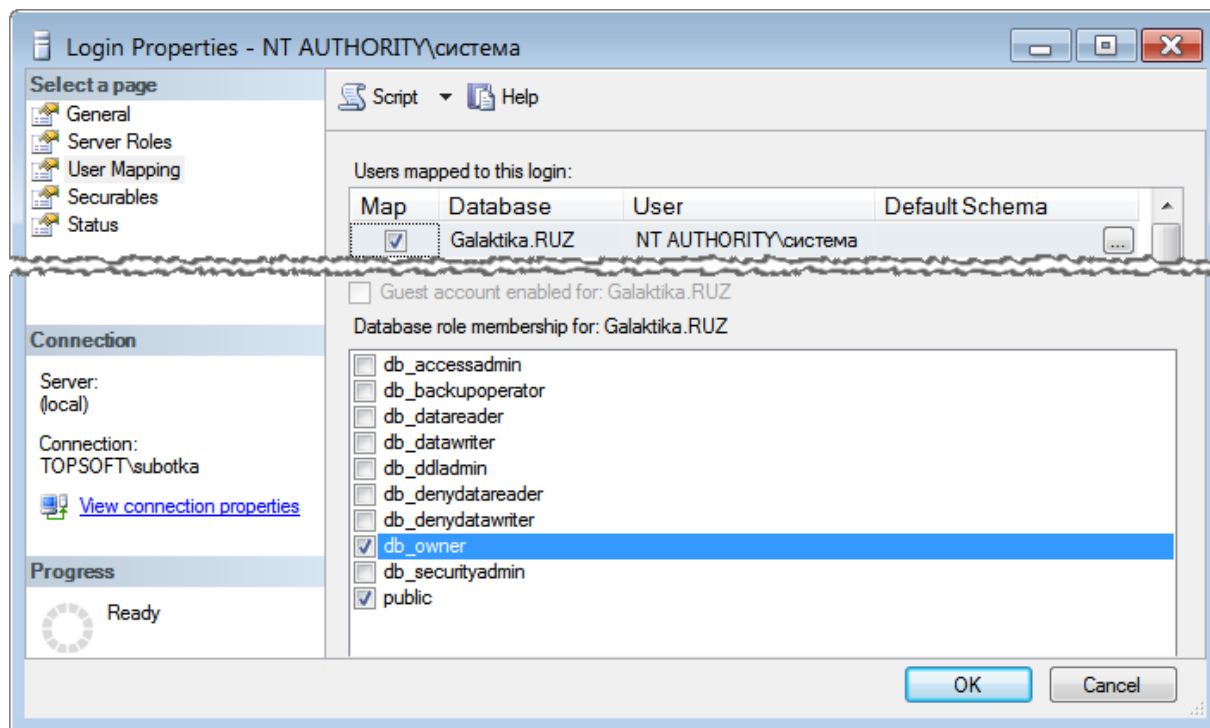


Рис. 20. Настройка пользователя для web-службы

3.3.3. Настройка на БД

Web-служба РУЗ располагается, как правило, в том же каталоге, что и загрузочный модуль приложения **Галактика Расписание учебных занятий** и там же запускается. При работе служба использует конфигурационный файл Web.config, в котором указывается строка соединения [connectionStrings](#) с БД расписания. [Параметры](#)³⁹ данной строки полностью идентичны аналогичной строке для приложения **Галактика Расписание учебных занятий**.

Примечание

Строка `<add... >` в секции `<connectionStrings>` должна быть только одна и идентична строке в конфигурационном файле Galaktika.Win.exe.config, например:

```
<connectionStrings>
  <add name="ConnectionString" connectionString="Integrated
    Security=SSPI;Pooling=false;Data Source=(local);
    Initial Catalog=Galaktika.RUZ" />
</connectionStrings>
```

Комментирование строк осуществляется с помощью тегов `<!-- ... -->`.

3.3.4. Настройка взаимодействия пользовательского web-приложения с web-службой системы

При настройке взаимодействия пользовательского web-приложения, использующего сервисы web-службы РУЗ, необходимо на сервере, где расположено пользовательское web-приложение:

- установить web-сервер, например IIS (Internet Information Services);
- разместить пользовательское web-приложение в виртуальном каталоге IIS;
- при необходимости выполнить настройку IIS;
- выполнить настройки пользовательского web-приложения в своем конфигурационном файле (в частности, указать размещение web-службы РУЗ, например: `<endpoint address="http://localhost:8080/RUZService" binding="basicHttpBinding">`, где localhost — это IP-адрес этого сервера).

3.3.5. Настройка для кроссдоменных AJAX-запросов

Если необходимо вызывать сервисы через кроссдоменные AJAX-запросы из доменов, отличных от домена, в котором развернута web-служба, то нужно корректно прописать значение параметра **Access-Control-Allow-Origin**. Для этого следует указать в качестве значения имя домена, из которого будут вызываться запросы. Если таких доменов несколько, то можно указать их имена через пробел или повторить параметр несколько раз. Если необходимо открыть доступ с любых доменов, то в качестве значения указывается символ "*", однако это не рекомендуется в силу появления изъянов в безопасности работы web-службы. Следует также помнить, что при использовании значения "*" не будут доступны POST-запросы-сервисы web-службы (например, сервис авторизации). Во избежание такого рода коллизий рекомендуется все проекты размещать в одном домене или настраивать доступ через Проху.

```
<system.webServer>
  <httpProtocol>
    <customHeaders>
      <!-- При отсутствии этого параметра кроссдоменные AJAX-запросы
           не будут выполняться. При необходимости использовать CORS -
           прописать в value: домен, из которого выполняются запросы;
           * - если надо открыть для всех доменов. -->
      <!--<add name="Access-Control-Allow-Origin" value="MyDomen"/>-->
      <remove name="X-Powered-By"/>
    </customHeaders>
  </httpProtocol>
  <modules runAllManagedModulesForAllRequests="false"/>
</system.webServer>
```

3.3.6. Запуск

Запуск web-службы осуществляется стандартным способом средствами IIS.

Для просмотра контрактов службы можно ввести в браузере строку вида:

http://localhost/RUZService.svc

Если приложение запускается с web-сервера, можно использовать localhost, иначе необходимо указать IP-адрес web-сервера и порт. Здесь *RUZService* — это псевдоним web-службы (задается при настройке IIS).

3.4. Web. Просмотр расписания

Примерная последовательность действий при развертывании web-приложения:

- 1) Получить каталог web-приложения Galaktika.RUZ.View.WebApp (каталог с необходимыми файлами для развертывания).
- 2) На сервере, где будет расположено web-приложение:
 - установить web-сервер, например IIS (Internet Information Services);
 - при необходимости [выполнить настройку IIS](#)^[36];
 - [настроить конфигурационный файл](#)^[36] web-приложения;
 - [запустить web-приложение](#)^[38].

3.4.1. Настройка IIS

Настройка IIS web-приложения "Просмотр расписания" осуществляется аналогично настройке [web-службы расписания](#)^[30].

При развертывании решения в IIS важно указать необходимые способы и параметры аутентификации для сайта.

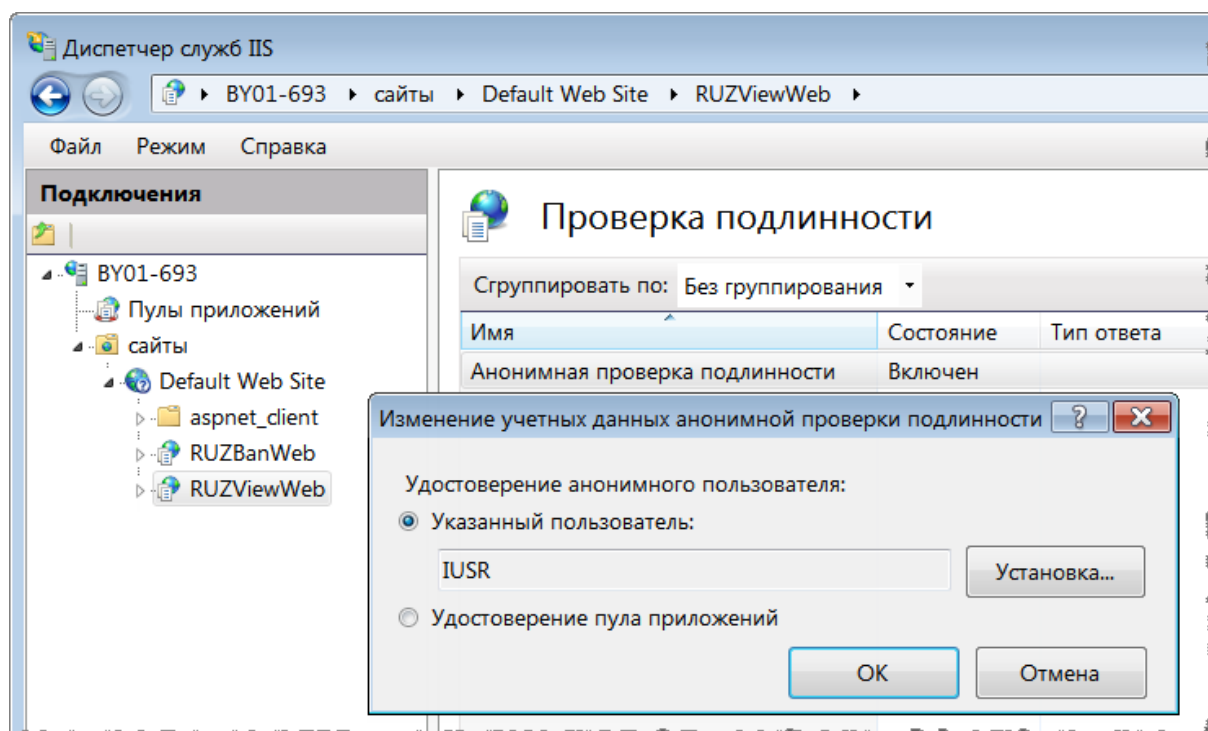


Рис. 21. Настройка способов аутентификации в IIS

3.4.2. Настройка на БД

Все параметры запуска приложения настраиваются в конфигурационном файле.

Web.config — это стандартный конфигурационный файл приложения на платформе .NET. Он расположен в каталоге с web-приложением. Редактирование файла выполняется с помощью любого редактора.

В конфигурационном файле необходимо настроить строку соединения с базой данных (БД) системы **Галактика Расписание учебных занятий**. Параметры соединения задаются в элементе:

```
<add name="ConnectionString" connectionString="Integrated
Security=SSPI; Pooling=false; Data Source=server01\SQL2008R2;
Initial Catalog=Galaktika.RUZ" />
```

В атрибуте `connectionString` задается строка соединения с БД. Приложение поддерживает работу с основными современными БД, в частности:

- MS SQL Server;
- Oracle;
- PostgreSQL;
- XML-файл.

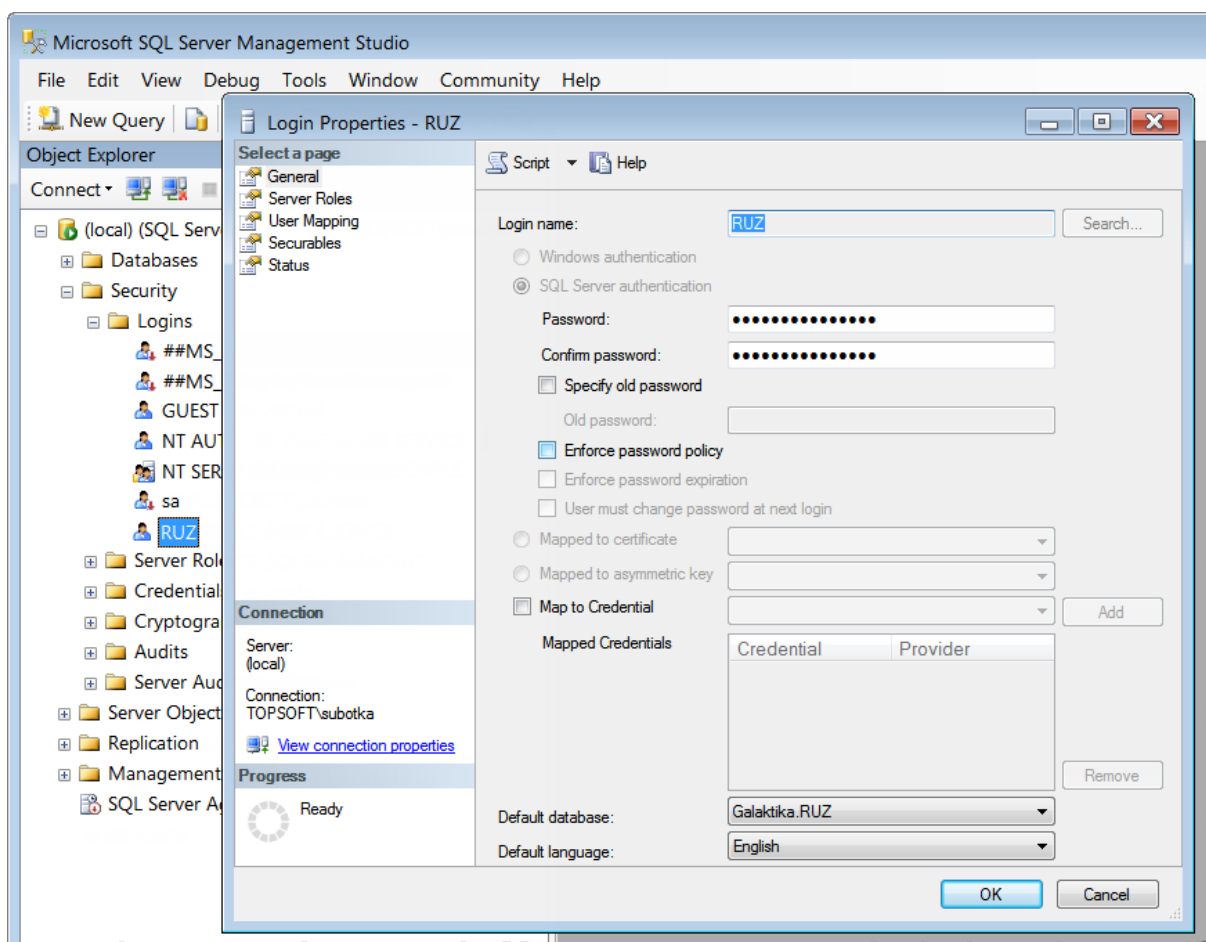
По умолчанию в конфигурационном файле задается строка соединения с БД MS SQL Server.

Пример конфигурационной строки для соединения с БД MS SQL Server Galaktika.RUZ, расположенной на сервере `serverName.domain.local`:

```
<add name="ConnectionString" connectionString="Persist Security
Info=False; User ID=RUZ; Password=123; Data Source=
serverName.domain.local; Initial Catalog=Galaktika.RUZ"/>
```

Для возможности доступа к БД необходимо указать пользователя, который будет иметь доступ к БД (в приведенном примере это пользователь *RUZ* с паролем *123*).

На сервере *serverName.domain.local* в MS SQL Server необходимо создать пользователя *RUZ* и дать ему права на доступ к БД Galaktika.RUZ.



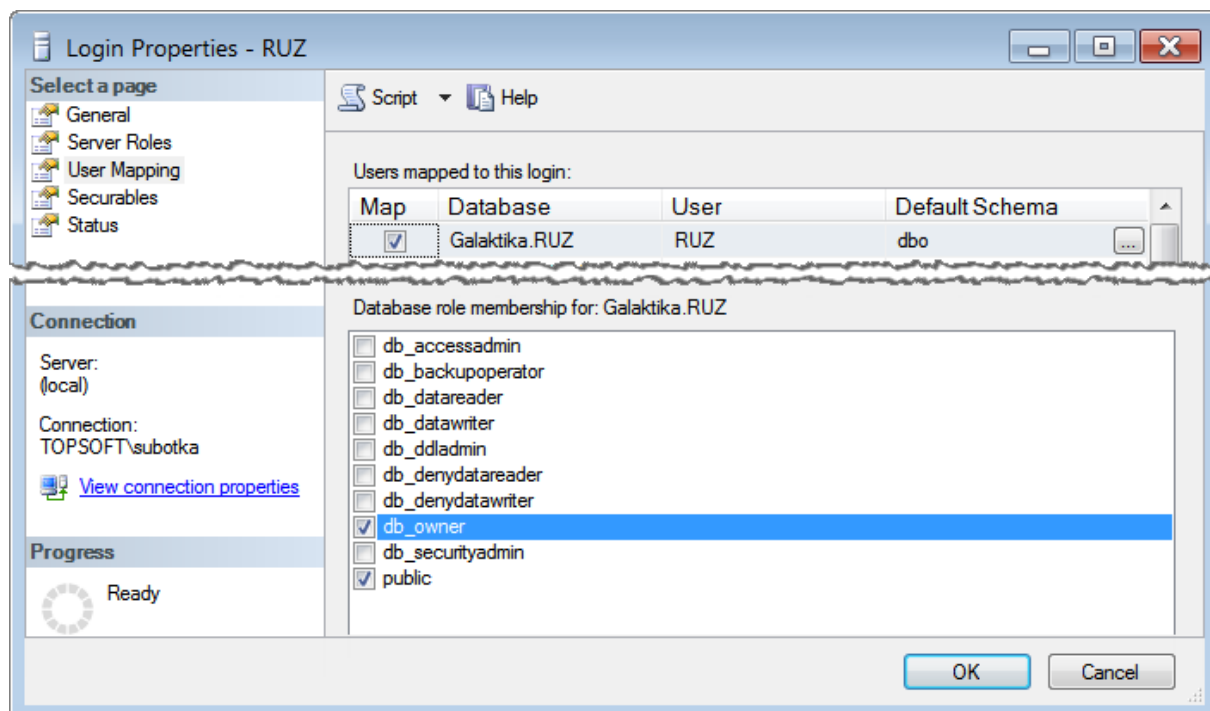


Рис. 23. Настройка прав доступа к БД в MS SQL Server

См. также [прочие параметры соединения с БД](#) ³⁹.

3.4.3. Запуск

Для запуска приложения следует ввести в браузере строку вида:

http://localhost/RUZViewWeb

Если приложение запускается с web-сервера, можно использовать localhost, иначе необходимо указать IP-адрес web-сервера.

RUZViewWeb — это псевдоним web-приложения (задается при настройке IIS).

3.5. Web. Заявки на бронирование/дооснащение аудиторий и запреты преподавателей

Порядок действий при разворачивании web-приложения "Заявки на бронирование/дооснащение аудиторий и запреты преподавателей" полностью идентичен разворачиванию web-приложения "Просмотр расписания". Отличия состоят лишь в каталоге приложения Galaktika.RUZ.Ban.WebApp и наименовании приложения-сайта RUZBanWeb.

3.6. Мобильный клиент



Мобильный клиент недоступен в базовой поставке.

Мобильный клиент разработан для платформ Android, iOS и Windows Phone. Поддерживаются клиентские устройства с операционными системами:

- Apple, iOS 6+;
- Android, Android 4+;
- Windows Phone, Windows Phone 8+.

Взаимодействие мобильного клиента с РУЗ осуществляется через web-сервисы расписания, поэтому для поддержания его работоспособности необходимо предварительно развернуть и настроить web-службу РУЗ.

Для использования мобильного приложения на мобильном устройстве необходимо скачать приложение с AppStore, GoogleMarket или WindowsStore — в зависимости от используемой в мобильном устройстве операционной системы.

После скачивания приложения при первом запуске потребуется выполнить настройки, в частности, ввести адрес сервера web-службы РУЗ.

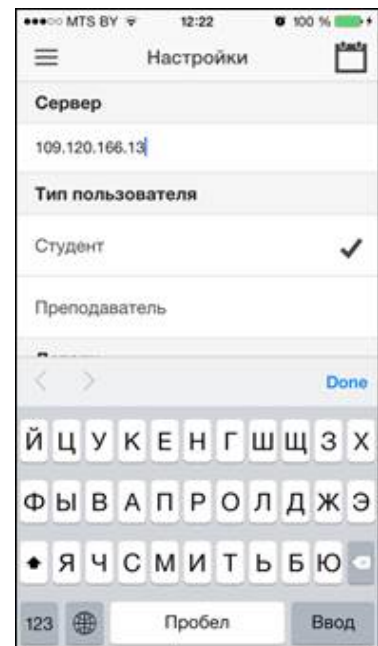


Рис. 24. Настройка мобильного приложения

3.7. Настройка конфигурационного файла

3.7.1. Соединение с БД

Параметры соединения с БД системы **Галактика Расписание учебных занятий** задаются в элементе:

```
<add name="ConnectionString" connectionString="Integrated
Security=SSPI;
Pooling=false;
Data Source=server01\SQL2008R2;
Initial Catalog=Galaktika.RUZ" />
```

В атрибуте `connectionString` задается строка соединения с БД. Приложение поддерживает работу со следующими БД:

- MS SQL Server;
- Oracle;
- Postgres Pro;
- XML-файл.

Строка соединения с БД на MS SQL Server

Строка для соединения с БД на MS SQL Server имеет следующий вид:

```
Integrated Security=SSPI;Pooling=false;Data Source=.;Initial Catalog=Galaktika.RUZ
```

Параметры:

- **Data Source** — имя инстанса SQL Server;
- **Initial Catalog** — имя БД;
- **Integrated Security=SSPI** — соединение с БД выполняется от имени пользователя, под которым запущено приложение;
- **User ID** — имя SQL пользователя, под которым выполнять соединение;
- **Password** — пароль SQL пользователя, под которым выполнять соединение.

В строке соединения используется либо **Integrated Security=SSPI** либо пара **User ID**; **Password**.

Строка соединения с БД на Oracle

Строка для соединения с БД на Oracle имеет следующий вид:

```
XpoProvider=Oracle;Data Source=ORCL_RUZ;User ID=RUZ;Password=1
```

Параметры:

- **Data Source** — имя настройки доступа к БД;
- **User ID** — имя пользователя Oracle, под которым выполнять соединение;
- **Password** — пароль пользователя Oracle, под которым выполнять соединение.

Строка соединения с БД на PostgreSQL

Строка для соединения с БД на PostgreSQL имеет следующий вид:

```
XpoProvider=Postgres;Server=127.0.0.1;User ID=RUZ;Password=1;Database=Galaktika.RUZ;Encoding=UNICODE
```

Параметры:

- **Data Source** — имя настройки доступа к БД;
- **User ID** — имя пользователя PostgreSQL, под которым выполнять соединение;
- **Password** — пароль пользователя PostgreSQL, под которым выполнять соединение.

Строка соединения с БД в XML-файле

Строка для соединения с БД в XML-файле имеет следующий вид:

```
XpoProvider=XmlDataSet;Data Source=Galaktika.RUZ.Demo.xml;read only=false;
```

Параметры:

- **Data Source** — имя XML-файла, в котором хранятся данные.

3.7.2. Ссылка на сервис лицензирования

Ссылка на сервер лицензирования системы *Галактика Расписание учебных занятий* задаются в элементе:

```
<add key="XafariLicenseServer" value="192.168.1.160:33445" />
```

В атрибуте **value** задается адрес сервиса лицензий. Адрес сервиса лицензий состоит из полного имени компьютера, на котором установлен сервис лицензий и номера порта сервиса лицензий. Например, `dpp-822.topsoft.local:33444`.

3.7.3. Путь к лог-файлу приложения

В лог-файл записываются операции, выполняемые системой и конечным пользователем во время работы приложения. Если операция приводит к ошибке или к исключению, соответствующая информация также записывается в лог-файл и это позволяет локализовать и понять ошибку. Файл не пересоздается при каждом запуске приложения, информация в нем накапливается.

Протокол работы приложения записывается в файл `eXpressAppFramework.log`. Файл сохраняется:

- в каталоге, указанном в ключе **GlobalLocationForLogAndModels**:

```
<add key="GlobalLocationForLogAndModels" value="D:\\PathToFolder1"/>
```

- иначе в каталоге, определяемой переменной, указанной в ключе **TraceLogLocation**:

```
<add key="TraceLogLocation" value="ApplicationFolder"/>
```

Value может принимать значения: **ApplicationFolder**, **CurrentUserApplicationDataFolder** (соответствует пути `C:\Documents and Settings\<пользователь>\Local Settings\Galaktika\Galaktika.RUZ\х.х.<сборка>.0`).

- если перечисленные ключи отсутствуют, лог-файл не сохраняется.

Сохранение лог-файла в каталог приложения требует открытия этого каталога на запись для пользователей системы. В большинстве случаев это нецелесообразно, поэтому при развертывании системы на сервере рекомендуется использовать отдельный каталог, определяемый перечисленными ключами.

Администратор может настроить уровень детализации информации, записываемой в лог-файл. В конфигурационном файле для этого предназначен раздел **system.diagnostics**:

```
<system.diagnostics>
  <trace autoflush="true" indentsize="3" />
  <switches>
    <!-- Use the one of predefined values: 0-Off, 1-Errors, 2-
Warnings, 3-Info, 4-Verbose. The default value is 3. -->
    <add name="eXpressAppFramework" value="1"/>
    <add name="XPO" value="1"/>
    <add name="Ranet" value="1"/>
    <add name="ApsModuleInfo" value="2"/>
    <add name="ApsModuleServiceInfo" value="4"/>
    <add name="WomInfo" value="3"/>
    <add name="WomSqlInfo" value="3"/>
  </switches>
</system.diagnostics>
```

Ключ **eXpressAppFramework** определяет настройки протоколирования XAF, ключ **XPO** определяет настройки протоколирования бизнес-объектов. Для каждого ключа можно задать следующие уровни детализации информации:

- 0 — протоколирование отключено;
- 1 — протоколируются только ошибки;
- 2 — протоколируются ошибки и предупреждения;
- 3 — в дополнение к ошибкам и предупреждениям протоколируются системные сообщения и действия пользователя;

- 4 — протоколируются все возможные события.

 *Со временем лог-файл увеличивается в объеме, поэтому целесообразно периодически его удалять или резервировать в отдельное хранилище.*

3.7.4. Путь к файлам модели пользователя

Файлы `Model.User.<Имя пользователя>.xafml` и `Model.User.<Имя пользователя>_ru.xafml` (например: `Model.User.Администратор.xafml`, `Model.User.Администратор_ru.xafml`) хранят изменения настроек визуальной части приложения, сделанные пользователем.

Путь к этим файлам задается в элементе:

```
<add key="UserModelDiffsLocation"
value="CurrentUserApplicationDataFolder"/>
```

По умолчанию файлы расположены в каталоге приложения.

В качестве пути к файлам можно указать либо полный путь, либо специальную переменную `CurrentUserApplicationDataFolder` (соответствует пути `C:\Documents and Settings\<пользователь>\Galaktika\Galaktika.RUZ\х.х.<сборка>.0`).

 *Если используется вариант развертывания приложения на сервере в общей папке, обязательно следует использовать переменную `CurrentUserApplicationDataFolder` для задания пути в конфигурационном файле (`<add key="UserModelDiffsLocation" value="CurrentUserApplicationDataFolder"/>`).*

В указанном каталоге сохраняется также файл `LogonParameters`, в который записываются параметры последней аутентификации пользователя.

 *Рассматриваемые файлы или каталог их расположения должны быть доступны пользователю для записи/изменения.*

В случае наличия в конфигурационном файле ключа `GlobalLocationForLogAndModels` перечисленные файлы сохраняются в каталоге, заданном в этом ключе:

```
<add key="GlobalLocationForLogAndModels" value="D:\\PathToFolder1"/>
```

3.7.5. Путь к файлам модели сборки

При первом запуске приложения формируется файл модели сборки `ModelAssembly.dll`. Файл может располагаться в каталоге приложения (по умолчанию) или по адресу, заданному с помощью параметра `GlobalLocationForModelAssembly`:

```
<add key="GlobalLocationForModelAssembly" value="D:\\
\\PathToFolder2"/>
```

В первом случае каталог приложения должен быть доступен для записи тому пользователю, который запускает первый раз систему, в т.ч. при ее обновлении. Установку системы или ее обновление может выполнять только администратор. Если после этих действий запускать приложение, то администраторские права на каталог приложения позволят сохранить в нем файл модели сборки. Такой вариант использования позволяет, в том числе, избежать коллизий с отсутствием сохраненной резервной копии БД, которую рекомендуется делать перед обновлением версии, т.к. пользователь, не являющийся администратором, не сможет запустить обновленную версию до тех пор, пока это не сделает администратор.

 *Перед обновлением версии рекомендуется делать резервную копию БД, т.к. при первом запуске обновленной версии выполняется, как правило, автоматическое обновление структуры БД и самих данных.*

3.7.6. Отображение диагностическая информация

Диагностическая информация о правилах, действиях, представлениях доступна пользователю при работе с приложением в случае установления в конфигурационном файле ключа `EnableDiagnosticActions` в разделе `appSettings`:

```
<appSettings>
  <add key="EnableDiagnosticActions" value="True" />
</appSettings>
```

В результате в приложении на панели инструментов станет доступна группа функций *Диагностика*.

Более подробно о содержимом отображаемой диагностической информации можно посмотреть в документации по XAF.

3.7.7. Настройка аутентификации

В обычном режиме использования приложения для прохождения [аутентификации](#)²⁷ пользователю предлагается окно с выбором типа аутентификации — windows или стандартная.

 *Web-приложение для просмотра расписания не требует аутентификации.*

При необходимости отключения окна выбора типа аутентификации в конфигурационный файл необходимо добавить ключ `OnlyWindowsAuthentication`:

```
<appSettings>
  <add key="OnlyWindowsAuthentication" value="True" />
</appSettings>
```

В этом случае будет автоматически использоваться windows-аутентификация.

3.7.8. Настройка выбора БД

Система **Галактика Расписание учебных занятий** может быть запущена в режиме обучения, когда каждый пользователь-обучающийся работает со своей БД автономно. Для этого необходимо:

- в конфигурационном файле прописать строку `<add key="CanChangeDatabase" value="true"/>`:

```
<appSettings>
  <add key="CanChangeDatabase" value="true"/>
</appSettings>
```

- в окне авторизации указать нужные БД, имя пользователя и пароль.

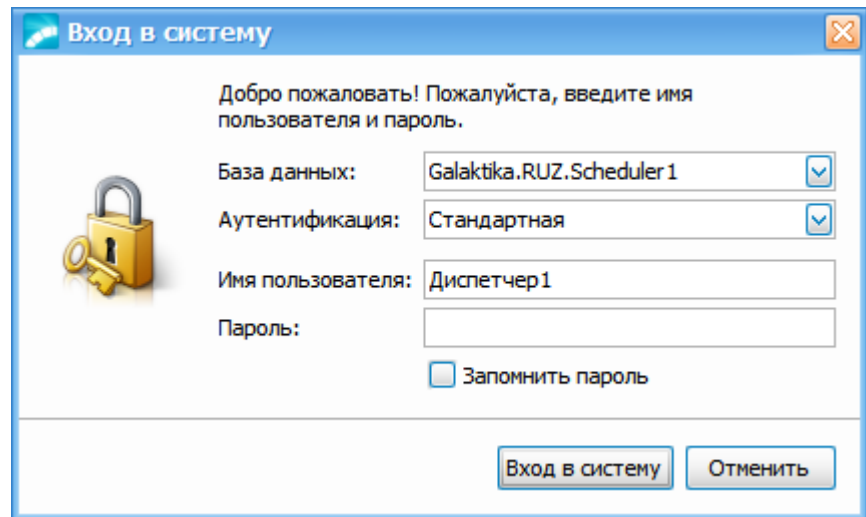


Рис. 25. Авторизация

Примечания:

1. В конфигурационном файле в разделе **ConnectionString** должна быть корректно настроена строка соединения с БД (должно обеспечиваться подсоединение к БД).
2. БД, указываемая в окне авторизации, должна быть предварительно подготовлена/создана и для нее должен быть заведен указываемый пользователь. Если такой БД нет в СУБД, то она будет создана автоматически с предустановленными значениями некоторых справочников, включая пользователей.

4. Безопасность приложения и настройка прав доступа

4.1. Ролевая схема администрирования

Для разграничения прав доступа в системе используется ролевая схема администрирования, обеспечивающая комплексную стратегию безопасности, которая включает в себя наборы разрешений, контроль их соблюдения, технологию проверки подлинности и т. д. Роли предназначены для определения групп пользователей в базе данных.

Определение ролевых групп и присущих им наборов прав осуществляется с помощью функции *Администрирование > Роли*.

При создании роли указывается ее имя.

Параметр *Административная* определяет доступ к функциям навигационной панели *Администрирование > Роли* и *Пользователи*, а также полный доступ к данным и действиям в системе.

Параметр *Разрешить редактирование модели* определяет доступ к функции общесистемного меню *Инструменты > Редактировать модель* ⁹².

При установленном параметре *Проверять кафедру аудитории (для диспетчера кафедры)* для диспетчера кафедры доступны только те аудитории, у которых установлена привязка к доступным диспетчеру *кафедрам* ⁵³.

При установленном параметре *Проверять факультет аудитории (для диспетчера факультета)* для диспетчера факультета доступны только те аудитории, у которых установлена привязка к доступным диспетчеру *факультетам* ⁵³ (или входящим в их состав кафедрам).

На вкладке *Пользователи* добавляются пользователи, которым назначается эта роль.

На вкладке *Права доступа к действиям* определяется возможность выполнения пользователем того или иного действия в системе. Здесь применяется один уровень доступа — *Да/Нет*.

На вкладке *Права доступа к данным* определяются права доступа (чтение, запись, создание, удаление, навигация) к определенным объектам (справочникам, нагрузке, расписанию и проч.). Для быстрого определения прав можно выделить необходимые записи и воспользоваться функциями  *Разрешить* и  *Запретить*, которые позволяют установить/снять флажки как на отдельные действия, так и на все сразу (полный доступ). При нажатии **Enter** на записи открывается окно, в котором возможно задание сложных условий управления доступом к полям и объектам с помощью критериев.

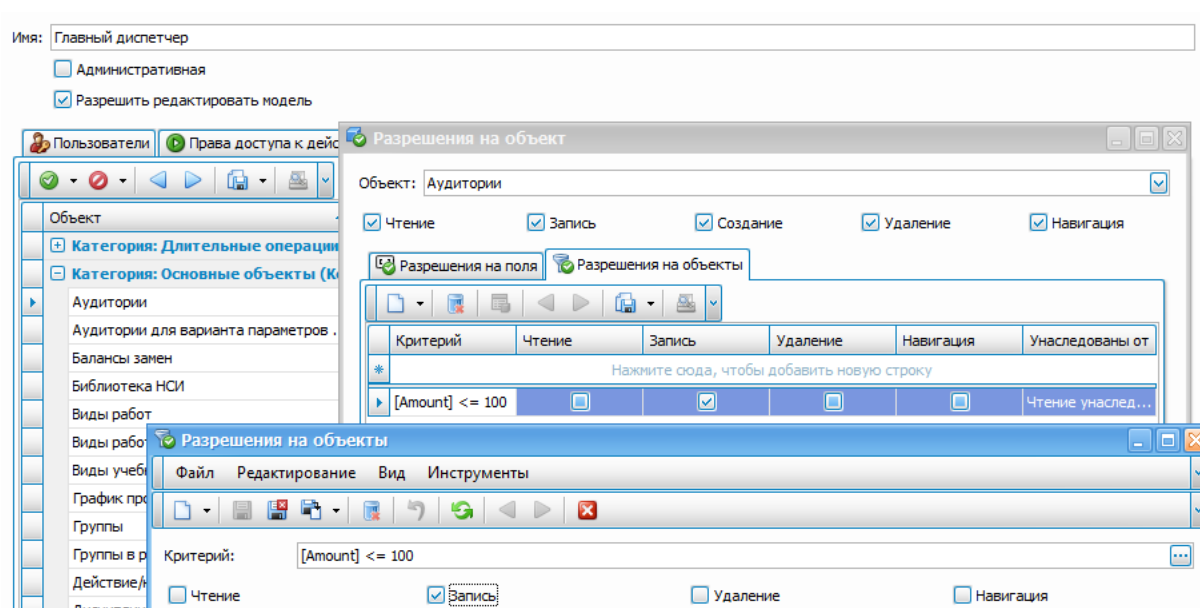


Рис. 26. Критерий доступа к объекту

Критерий может создаваться с помощью специального конструктора или вручную. Подходить к созданию критериев следует обдуманно, учитывая некоторые нюансы, если необходимо:

- открыть доступ к просмотру/навигации какого-либо справочника;
- закрыть права на редактирование/создание/удаление;
- открыть доступ к **выбору** элементов из этого справочника в зависимых от него объектах.

В последнем случае необходимо корректно прописать доступ на редактирование к необходимым зависимым коллекциям. На рисунке ниже приведен пример настройки разрешений для объекта *Виды работ*.

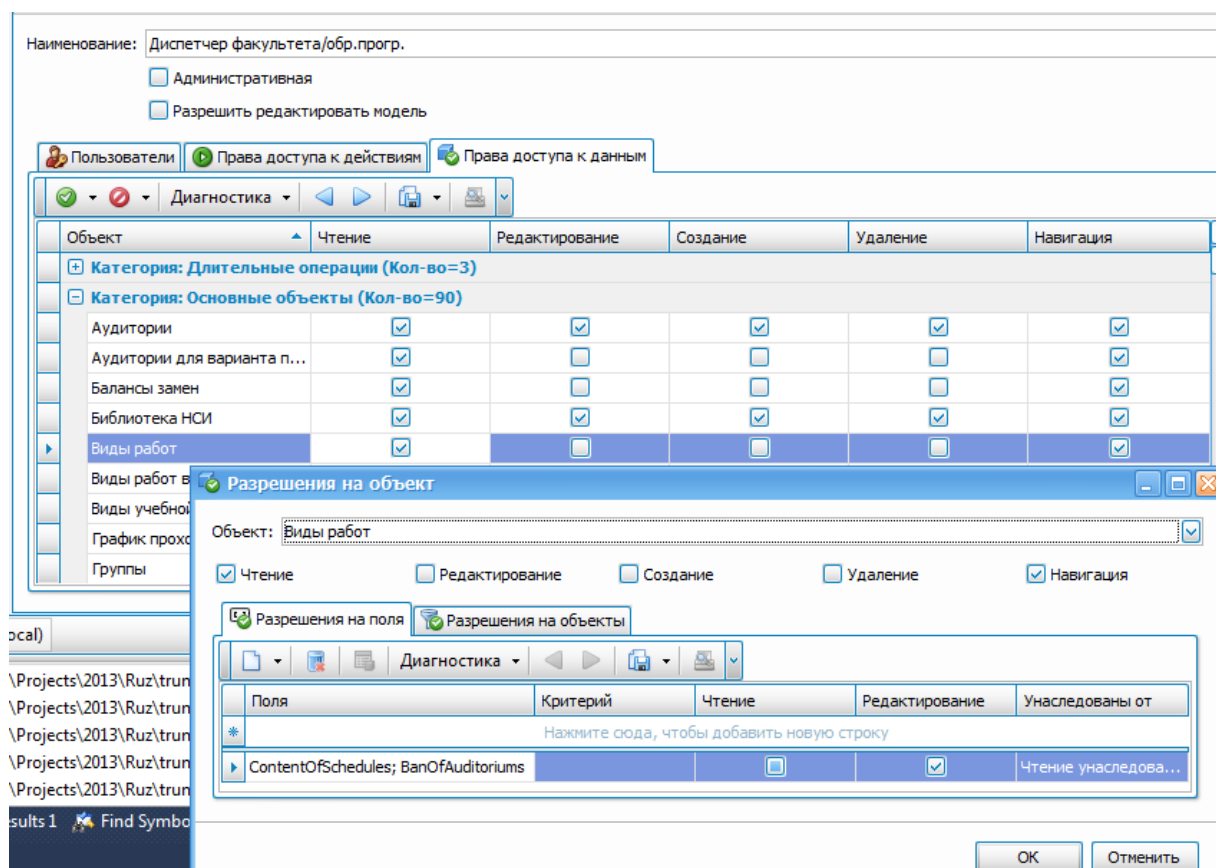


Рис. 27. Доступ к редактированию зависимых коллекций

Для перечисления зависимых коллекций следует воспользоваться приведенной ниже таблицей.

Табл. 1. Зависимые коллекции объектов

Объект	Зависимые коллекции
Аудитории	KindOfWorkInAuditoriums DisciplineInAuditoriums RestrictionOnAuditoriums BanOfAuditoriums RepeatBanOfAuditoriums ContentOfSchedules ChangeLogAuditoriumOlds ChangeLogAuditoriumNews ReplacedLessons ContentOfTechnologicalCards AuditoriumInTypicalWeekA
Библиотека НСИ	ScheduleInDocNSIs
Виды работ	ContentOfSchedules KindOfWorkInAuditoriums RestrictionOnAuditoriums ContentOfLoads ChangeLogKindOfWorkOlds ChangeLogKindOfWorkNews ReplacedLessons ContentOfTechnologicalCards BanOfAuditoriums

Объект	Зависимые коллекции
Виды учебной деятельности	KindOfWorks
Группы	SubGroups StaffOfStreams GroupInSchedules BanOfGroups RepeatBanOfGroups ContentOfSchedules ContentOfLoads ReplacedLessons StaffGroups CrossingContingents CrossingContingentCrossings TypicalAnalysisOfTheSchedule_Parametres
Дисциплины	PriorityOfDisciplineInSchedules ContentOfSchedules ContentOfLoads DisciplineInAuditoriums RestrictionOnAuditoriums ChangeLogDisciplineOlds ChangeLogDisciplineNews ReplacedLessons DisciplineOfSpecialitys TechnologicalCards
Должности	Lecturers
Запреты группы	BanOfGroupInSchedules
Запреты для пар	BanOfLessonInSchedules
Запреты зданий	BanBuildingInSchedules
Запреты преподавателей	ContentBanOfLecturers RepeatBanOfLecturers BanOfLecturerInSchedules
Запреты/заявки аудиторий	BanOfAuditoriumInSchedules
Здания	Auditoriums BuildingInSchedules BuildingDocs BuildingInBanLecturers BanBuildings RepeatBanBuildings BanOfAuditoriums
Календ. период	Schedules DocNSIs
Кафедры/департ.	Disciplines Auditoriums Lecturers Groups ContentOfLoads
Курсы	Schedules Groups Streams

Объект	Зависимые коллекции
	Curriculums
Нерабочие дни	UnworkDayInSchedules
Обучающиеся	StaffOfSubGroups StaffOfStreams StaffGroups
Пары	CompoundOfGangs ContentOfSchedules BanOfLessons ContentBanOfLecturers BanOfAuditoriums BanOfGroups ReplacedLessons BanBuildings
Подгруппы	StaffOfStreams ContentOfSchedules ContentOfLoads ReplacedLessons CrossingContingents CrossingContingentCrossings
Потоки	ContentOfLoads ContentOfSchedules ReplacedLessons CrossingContingents CrossingContingentCrossings
Преподаватели	BanOfLecturers ContentOfSchedules ContentOfLoads ChangeLogLecturerOlds ChangeLogLecturerNews ReplacedLessons
Расписания занятий	ReplacedLessons
Сетки пар	ContentTableOfLessons Buildings
Смены	CompoundOfGangs GangInSchedules Groups
Содержание нагрузок	ContentOfSchedules DenormalizationOfLoads AuditoriumPreferreds
Специализации	Groups Curriculums
Специальности	Specializations Groups Schedules DisciplineOfSpecialitys Curriculums
Типы аудиторий	Auditoriums BanOfAuditoriums

Объект	Зависимые коллекции
Требования	RequirementOfSchedules
Факультет/образовательная программа	Schedules Chairs Auditoriums Groups Streams RUZUserFaculties Curriculums
Формы обучения	Schedules Groups Streams
Циклы дисциплин	ContentOfLoads

Примечания:

1. При назначении прав необходимо соблюдать согласованность между правами доступа к данным и правами доступа к действиям. Например, если закрыть доступ на создание, редактирование, удаление для объекта *Аудитории*, то при этом автоматически закроется доступ к стандартным действиям создания, редактирования и удаления записи в списке и форме редактирования. Однако при открытом доступе к действиям в списке будет доступно множественное редактирование записей.
2. Пользователь может обладать несколькими ролями. В этом случае при расчете прав доступа к действию или к объекту к учету принимается максимальное из прав.

4.2. Предустановленные роли

По умолчанию в системе предустановлены следующие роли:

- *Administrator*;
- *Главный диспетчер*;
- *Диспетчер факультета/образовательной программы*;
- *Диспетчер кафедры/департамента*;
- *Диспетчер филиала*;
- *Преподаватель*;
- *Студент*.

Каждая роль является самодостаточной. При необходимости администратор системы может скорректировать права для роли.

Администратор

Роль *Administrator* предназначена для администратора системы. Признак *Административная* обеспечивает полный доступ ко всем данным и ко всем действиям.

Главный диспетчер

Роль *Главный диспетчер* предназначена для пользователей, исполняющих обязанности главного диспетчера. Для этой роли предустановлен доступ ко всем

действиям, за исключением прав по администрированию и изменению справочника *Академкалендарь*.

Критериально также отрегулирован доступ к возможности изменения преподавателя в назначенных занятиях для неутвержденных расписаний (создано разрешение на редактирование для всех полей, кроме *Преподаватель* (конструкция **;Lecturer*), и отдельное разрешение для поля *Преподаватель*, см. рисунок).

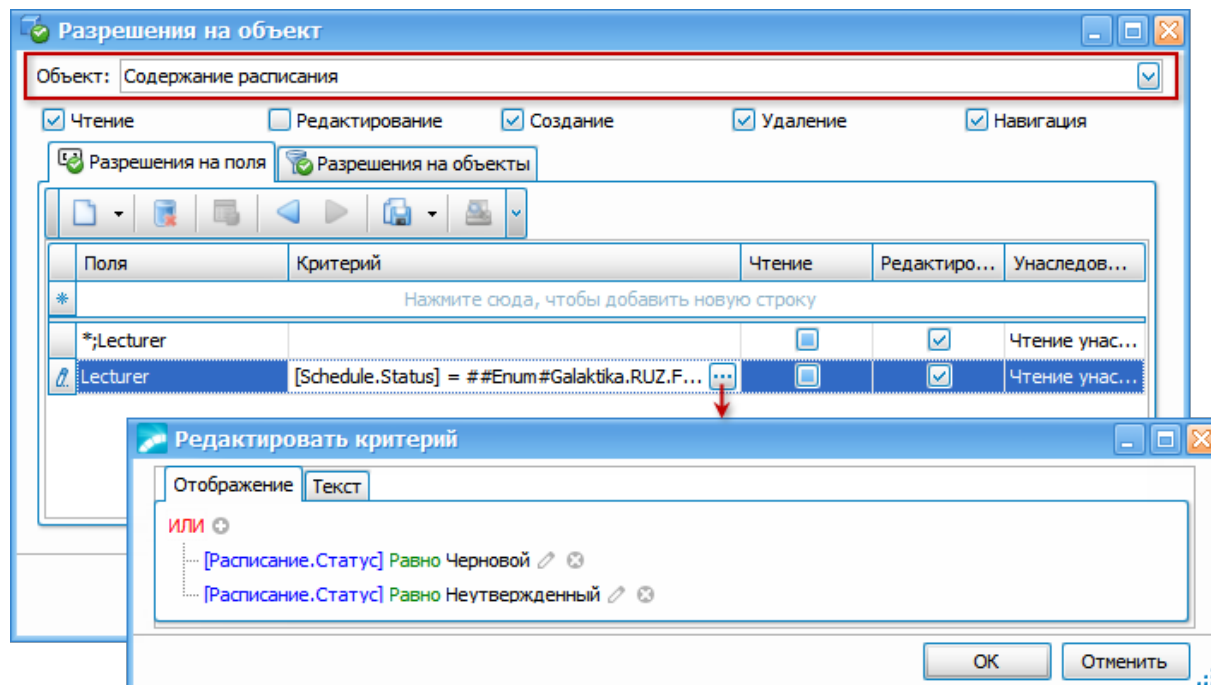


Рис. 28. Критерии доступа к полю "Преподаватель"

Диспетчер факультета/образовательной программы

Для роли *Диспетчер факультета/образовательной программы* предустановлены следующие разрешения:

- просмотр/чтение всех справочников;
- ведение аудиторного фонда;
- ведение ограничений для преподавателей;
- ведение ограничений по дисциплинам, видам работ для аудиторий;
- ведение ограничений для учебных групп;
- подача и согласование заявок на бронирование аудиторий и дооснащение аудиторий;
- ведение библиотеки нормативно-справочной информации;
- формирование и редактирование расписаний;
- формирование отчетов;
- выполнение оповещений.

Диспетчер кафедры/департамента

Для роли *Диспетчер кафедры/департамента* предустановлены следующие разрешения:

- просмотр/чтение всех справочников;

- ведение ограничений для преподавателей;
- ведение ограничений по дисциплинам, видам работ для аудиторий;
- подача и согласование заявок на бронирование и дооснащение аудиторий;
- формирование отчетов;
- выполнение оповещений.

Диспетчер филиала

Роль *Диспетчер филиала* может назначаться только тем пользователям, у которых уже назначена роль *Диспетчер факультета/образовательной программы*. Она является, своего рода, ограничителем, т.к. разрешает доступ пользователю только к объектам назначенному ему филиала. Также, для корректной работы данной роли, предварительно необходимо настроить факультеты/филиалы:

- 1) добавить головной факультет (допускается создание фиктивного подразделения), который будет объединять в себе совокупность факультетов/обр. программ;
- 2) подчиненным факультетам/обр. программам проставить в поле Институт/Факультет головное подразделение.

После назначения пользователю роли и филиала/факультета, ему будут доступны объекты (преподаватели, группы, сетки пар и т.п.) только назначенного ему филиала/факультета.

4.3. Ведение пользователей

4.3.1. Создание/редактирование пользователя

С помощью функции *Администрирование > Пользователи* формируется список пользователей системы.

 При создании новой БД приложения, пользователь, выполняющий эту операцию, автоматически регистрируется в БД и ему назначается роль *Administrator*.

Пользователь

Имя пользователя: Email:

ФИО:

Тип:

Преподаватель:

Телефон:

Студент:

☐ Сменить пароль при первом входе

☐ VIP-пользователь

☒ Активен

☐ Скрывать недоступные здания

☐ Разрешено редактировать только учебные группы

☐ Возможность редактировать скрытые здания

ID пользователя:

Примечание:

Роли ☒ Доступные кафедры/департ. ☒ Доступные факультеты/обр.прогр. ☒ Доступные здания

Диагностика

Наименование	Административ...	Разрешить редактировать модель
Users	☐	☐
Главный диспетчер	☐	☐

При внесении в систему нового пользователя следует указать его имя (логин), под которым он будет входить в систему, телефон, Email.

Если установлен параметр **Сменить пароль при первом входе**, то пользователю будет выдано соответствующее окно при первом запуске системы. При смене пароля система безопасности не разрешает использовать старый пароль.

Параметр **VIP-пользователь** используется в последующем для выделения важности действий, выполненных пользователем (например, подсветка заявок на аудитории).

Параметр **Активен** позволяет включать/отключать доступ пользователя к БД.

Если установлен параметр **Скрывать недоступные здания**, пользователю будут видны только аудитории зданий, внесенных на вкладке Доступные здания.

Параметр **Возможность редактировать скрытые здания** позволяет включать/отключать доступ пользователя к параметру **Скрытое здание** в карточке справочника **Здания**.

Далее следует добавить Роли пользователя. На основании этих ролей определяется значение поля **Тип** — берется предопределенная в системе роль с наибольшими правами (например, если указаны роли *Главный диспетчер* и *Преподаватель*, то тип примет значение *Главный диспетчер*). Если указана созданная пользовательская роль, то тип принимает значение *Преподаватель*. Если для роли (системной или пользовательской) указан признак *Административная*, то тип принимает значение *Администратор*.

Если для пользователя указаны роли *Преподаватель* или *Студент*, то в верхней панели становятся доступными соответствующие поля для привязки пользователя к конкретному преподавателю или студенту. Если такая связь установлена, то *Преподаватель* имеет право создавать, редактировать, удалять только свои запреты (график работы).

Для пользователей с ролями *Главный диспетчер* или *Диспетчер факультета/образовательной программы* предназначены две дополнительные вкладки, на которых формируются списки доступных пользователю факультетов/образовательных программ и кафедр/департаментов.

Если установлен доступ к факультету/образовательной программе, то считается открытым доступ ко всем кафедрам/департаментам этого объекта данных.

Если установлен доступ к факультету/образовательной программе, у которых имеются подчиненные факультеты/образовательные программы (связь по полю **Институт/факультет**), то считается открытым доступ и ко всем этим объектам.

Если у пользователя с ролью *Главный диспетчер* не указаны доступные факультеты/образовательные программы, то считаются доступными все.

Для пользователя с ролью *Диспетчер кафедры/департамента* доступна вкладка Доступные кафедры/департаменты, на которой определяется перечень подразделений, за которые отвечает данный пользователь.

Пользователь может обладать несколькими ролями. В этом случае при расчете права доступа к действию или к объекту к учету принимается максимальное из прав.

4.3.2. Приостановка действия учетной записи пользователя

В некоторых случаях возникает необходимость временно заблокировать вход определенного пользователя в систему. Для этого следует снять признак **Активен** в

учетной записи пользователя. При попытке входа неактивного пользователя в систему будет выдана ошибка и вход будет запрещен.

Для снятия блокировки следует снова установить признак *Активен*. В этом случае пользователь может продолжить работу с системой в обычном режиме.

Если учетная запись более не нужна, ее рекомендуется удалить.

4.3.3. Удаление пользователя

Для удаления пользователя используется стандартная функция  *Удалить*.

 Для [предустановленных пользователей](#)⁵⁴ операция удаления недоступна.

4.3.4. Предустановленные пользователи

По умолчанию в системе созданы следующие пользователи:

- *Администратор* с ролью *Administrator*;
- *ДиспетчерГл* с ролью *Главный диспетчер*;
- *ДиспетчерФ* с ролью *Диспетчер факультета/образовательной программы*;
- *ДиспетчерК* с ролью *Диспетчер кафедры/департамента*;
- *Преподаватель* с ролью *Преподаватель*;
- *Студент* с ролью *Студент*.

Эти пользователи могут использоваться при работе с системой, если при загрузке выбирается аутентификация *Стандартная*. Для каждого из них по умолчанию установлена одна самодостаточная роль. При необходимости *Администратор* системы может наделить пользователя несколькими ролями. В этом случае при расчете прав доступа к действию или к объекту к учету принимается максимальное из прав.

Для удобства создания однотипных пользователей может использоваться функция клонирования.

4.3.5. Особенности настройки доступа пользователей к аудиторному фонду

Доступ пользователей к ведению аудиторного фонда обеспечивается типовыми правами по доступу к действиям и доступу к данным.

Доступ пользователей к использованию аудиторного фонда определяется на основании:

- привязки аудиторий к факультету/образовательной программе;
- прав доступа пользователя к факультету/образовательной программе;
- прав доступа пользователя к зданию.

В результате для пользователей с ролями *Главный диспетчер*, *Диспетчер факультета/образовательной программы*, *Диспетчер кафедры/департамента* аудитория доступна (видима), если:

- аудитория является общедоступной, т. е. нет привязки к факультету/образовательной программе, нет привязки к кафедре/департаменту и при этом выключены [параметры](#)⁴⁵ **Проверять факультет аудитории** (для

диспетчера факультета) или **Проверить кафедру аудитории** (для диспетчера кафедры);

- аудитория привязана к факультету/образовательной программе и этот факультет/образовательная программа указаны на соответствующей вкладке окна редактирования прав доступа пользователя;
- аудитория привязана к факультету/образовательной программе, подчиненным другому факультету/образовательной программе (связь по полю **Институт/факультет**), и последний (последняя) указаны на соответствующей вкладке окна редактирования прав доступа пользователя;
- аудитория привязана к кафедре/департаменту, которые указаны на соответствующей вкладке окна редактирования прав доступа пользователя;
- не установлен параметр **Скрывать недоступные здания** или параметр установлен, но аудитория относится к зданию, указанному на вкладке Доступные здания.

Для пользователей с ролью *Преподаватель* аудитория доступна (видима), если:

- аудитория является общедоступной;
- аудитория привязана к кафедре/департаменту, к которым относится преподаватель;
- не установлен параметр **Скрывать недоступные здания** или параметр установлен, но аудитория относится к зданию, указанному на вкладке Доступные здания.

Для пользователей с ролью *Студент* аудитория доступна (видима), если:

- аудитория является общедоступной;
- аудитория привязана к факультету/образовательной программе или кафедре/департаменту, к которым относится этот студент (для этого используются одноименные параметры академической группы студента);
- не установлен параметр **Скрывать недоступные здания** или параметр установлен, но аудитория относится к зданию, указанному на вкладке Доступные здания.

4.4. Аутентификация пользователей

Подсистема поддерживает два типа аутентификации: стандартная и windows.

Тип аутентификации указывается при [запуске системы](#)^[27].

В первом случае конечный пользователь проходит аутентификацию на основании учетных данных (имя пользователя и пароль), указанных в окне входа в систему. Пользователь, чьи учетные сведения совпадают с соответствующей записью в базе данных приложения, проходит проверку подлинности.

Windows-аутентификация использует службу WindowsActiveDirectory для получения информации о пользователе. Если учетная запись присутствует в системе и соответствующий объект существует в базе данных приложения, пользователь успешно проходит аутентификацию.

При серверном использовании системы осуществляется предварительная настройка пользователей системным администратором и администратором приложения **Галактика Расписание учебных занятий**.

Системный администратор прописывает средствами СУБД в системе безопасности СУБД пользователей для Windows-аутентификации с необходимыми для конкретного пользователя типовыми для СУБД серверными ролями и ролями доступа к БД решения.

Для обычного windows-пользователя достаточны приведенные на рисунках ниже права в системе безопасности СУБД.

Администратор приложения прописывает средствами приложения **Галактика Расписание учебных занятий** в системе администрирования приложения тех же пользователей для Windows-аутентификации, наделяя конкретного пользователя необходимыми ему типовыми ролевыми правами в рамках доступа к функциям и данным приложения **Галактика Расписание учебных занятий**.

При Windows-аутентификации при первом входе:

- в случае отсутствия пользователя в системе безопасности СУБД будет выдано сообщение о невозможности соединения с БД;
- в случае отсутствия пользователя в БД приложения он будет добавлен в список пользователей этой БД автоматически, но наделен минимальными правами, исключающими возможность использования приложения по назначению. При необходимости администратор приложения может скорректировать права пользователя.

 После развертывания системы рекомендуется список типовых пользователей привести в соответствие потребностям (переименовать, удалить).

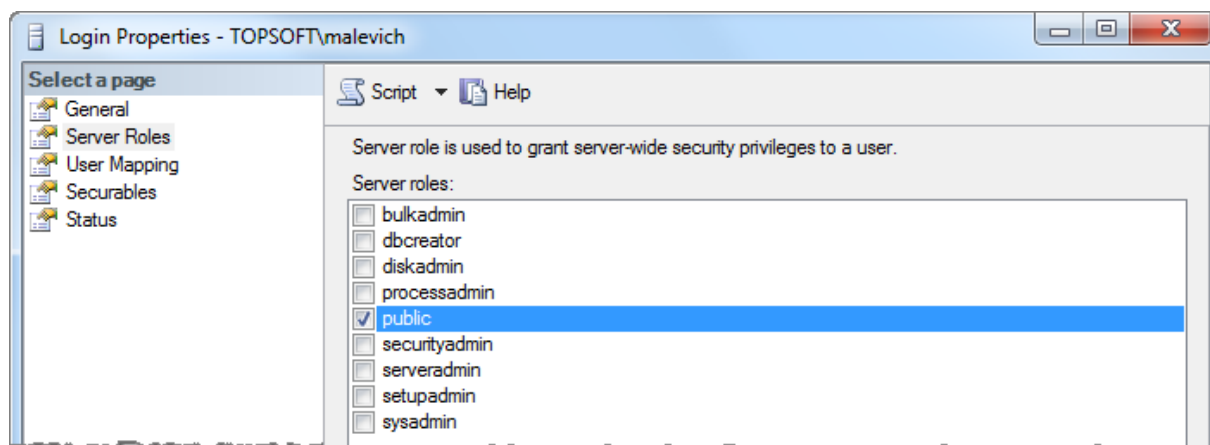


Рис. 30. Серверные роли windows-пользователя

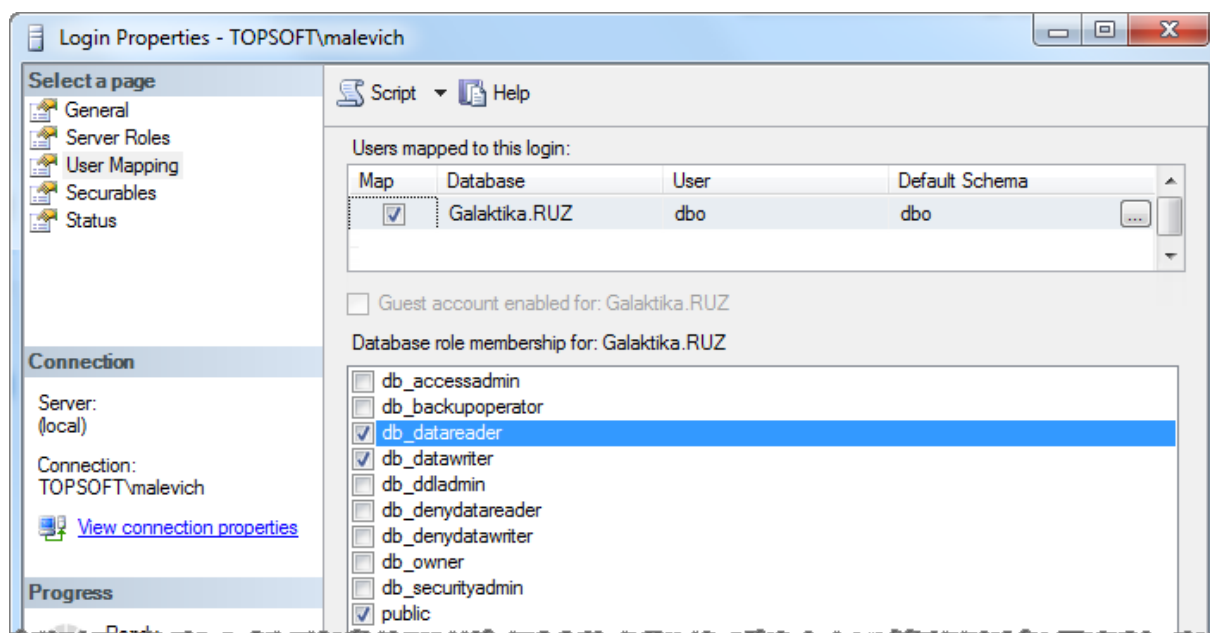


Рис. 31. Серверные права доступа к БД для windows-пользователя

Журнал логирования (Администрирование > Журнал логирования) позволяет вести учет подключенных пользователей, в том числе при неудачной попытке подключения.

Возможен **Экспорт** Журнала логирования в другие форматы (*.csv, *.html, *.xlsx и т.п.).

5. Настройка параметров системы

5.1. Общие параметры

Поведение системы может корректироваться посредством настройки ее параметров, доступ к которым осуществляется с помощью функции *Сервис > Общие параметры*.

Образовательное учреждение — заполняется автоматически при создании БД на основании лицензионного файла. При смене наименования образовательного учреждения необходимо:

- получить новый лицензионный файл;
- активировать лицензию (см. соответствующую инструкцию);
- изменить наименование учреждения в общих параметрах в соответствии с лицензионным файлом;
- проверить запуск системы.

Дискр. "Загрузка преподавателей" (мин), Дискр. для "Загруженность аудиторий" (мин) — определяют интервалы для временной шкалы, выводимой в окнах просмотра дневной загрузки преподавателей/аудиторий (функции панели инструментов расписания *Загрузка преподавателей > Дневная* и *Загруженность аудиторий > Дневная* соответственно).

Для web-службы учитывать расписания в статусах: Активный, Неутвержденный, Черновой — указываются статусы расписаний, которые будут учитываться в сервисах web-службы. По умолчанию web-службой выбираются занятия из расписаний в статусах **Активный** и **Неутвержденный**.

Параметры оповещения — в этом разделе указываются параметры оповещений об изменениях в расписании и бронировании аудиторий:

- Если параметр **Журналирование оповещений** установлен, то все оповещения об изменении в расписаниях, действиях с расписаниями, появлении заявок на бронирование/дооснащение аудиторий или изменении их статуса фиксируются в **Журнале уведомлений**.
- Параметры **Отправлять EMail, Отправлять SMS и Отправлять Push** управляют возможностью дублирования оповещений по соответствующим каналам.
- Параметр **Персонализация расписания до обучающегося** определяет конкретизацию опубликованных расписаний до конкретного студента. Суть этого параметра состоит в следующем. У занятия имеется параметр-ресурс — контингент обучающихся. Это группа, поток, подгруппа, но не отдельный обучающийся. Для получения расписания указывается искомая группа. В результирующем наборе будут присутствовать занятия:
 - назначенные для этой группы;
 - занятия потоков, в составе которых есть эта группа или любая подгруппа этой группы;
 - подгрупповые занятия подгрупп этой группы.

Обучающийся может входить только в одну академическую группу. Если он учится на нескольких специальностях, то это "разные" студенты. Таким образом, в расписании группы возможно будут "лишние" подгрупповые или потоковые занятия,

которые к конкретному обучающемуся отношения непосредственно не имеют. В случае необходимости отображать исключительно занятия, касающиеся конкретного обучающегося:

- в настройках системы устанавливается указанный параметр;
 - на форме запроса расписания в дополнительном параметре **Обучающийся** выбирается конкретный обучающийся;
 - из программно формируемого списка занятий по группе обучающегося автоматически исключаются занятия, в контингент которых этот обучающийся не входит.
- Параметр **Язык публикации** позволяет опубликовать расписание на выбранном иностранном языке. Если значение параметра не указано, то публикация осуществляется на русском языке.

Параметры выгрузки — при установленном параметре **Не передавать email обучающихся и преподавателей** во всех [методах веб-службы РУЗ¹⁰⁴](#), возвращающих e-mail студентов и преподавателей, параметры *email* и *lecturerEmail* будут иметь пустое значение.

Параметры SMS-уведомления — в этом разделе указываются параметры рассылки SMS-сообщений об изменениях в расписании занятий. На данный момент для апробации отправки SMS-сообщений используется сервис <http://smssc.ru/>.

Параметры EMail-уведомления — в этом разделе указываются параметры рассылки EMail-сообщений об изменениях в расписании занятий.

Рассылка SMS- и EMail-уведомлений осуществляется из:

- расписания (конструктор расписания, журнал изменений, действия/комментарии);
- отчета *Журнал изменений*;
- заявок на бронирование/дооснащение аудиторий.

Порядок вывода информации о занятии по умолчанию — в этом разделе задается последовательность вывода данных о занятии в сетке расписания конструктора. При необходимости эти параметры могут быть переопределены в карточке конкретного расписания.

Параметры конструктора:

- **Подсветка ячеек только после выбора всех ресурсов** — недоступность ячеек в сетке расписания (подсветка серым фоном) рассчитывается только после выбора и загрузки, и аудитории; параметр влияет на быстродействие назначения занятия;
- **Подсветка в ячейках занятости преподавателя, аудитории** — при выборе загрузки занятость преподавателя и аудитории подсвечивается в сетке расписания соответствующим фоном (преподавателя — бежевым цветом, аудитории — зеленым);
- **Учитывать при подсветке пересечение контингентов** — при определении недоступности ячеек в сетке расписания учитывается занятость контингента или его элементов в других расписаниях; параметр влияет на быстродействие назначения занятия для потоковых нагрузок;
- **Подсветка позиции занятия до замены** — в сетке расписания розовым фоном подсвечивается позиция, в которой находилось занятие до замены.

Трактовка ограничений ресурсов в отношении аудитории — в этой группе параметров указывается жесткость ограничений на проведение занятий в аудитории в соответствии с *Дисциплиной*, *Видом работы* или разрезом *Дисциплина — вид*

работы: (под символом Δ понимается дисциплина, вид работы или разрез "дисциплина — вид работы" — в зависимости от параметра):

- *Только в этой аудитории только Δ* — в конкретной аудитории может проводиться только эта дисциплина и никакие другие дисциплины в данной аудитории проводиться не могут (аналогично для вида работ и разреза "дисциплина — вид работы");
- *Δ только в этой аудитории* — дисциплина может проводиться только в конкретной аудитории, но при этом в данной аудитории могут проводиться и другие дисциплины (аналогично для вида работ и разреза "дисциплина — вид работы");



Непосредственно сами ограничения задаются в справочниках аудиторий, видов работ, дисциплин.

Учитывать при назначении занятий запреты из общего перечня (независимо от включения в расписание) — отмеченные флажками запреты учитываются в расписании, даже если не включены в него, остальные запреты необходимо добавлять в расписания вручную.

Форматы отображения информации — в этой группе параметров указывается, какие данные и в каком виде по умолчанию отображать в расписании. При необходимости параметры могут быть переопределены в самом расписании. Эти параметры используются также при публикации расписаний на сайтах.



Значение *Наименование (язык)* для *Дисциплины* позволяет отображать в расписании язык преподавания дисциплины.

График прохождения с учетом последовательности прохождения дисциплин — значения *Среднее количество часов в неделю*, *Максимальное количество часов в неделю* используются в интерфейсе *График прохождения с учетом последовательности прохождения дисциплин* для сравнения при распределении часов нагрузки по календарному периоду.

Параметры для нагрузок — при установленном флаге *Запретить создание дубликатов нагрузок* происходит проверка на уникальность при создании нагрузки.

Параметры импорта:

- **Импорт по GID или UID** — возможность выбора идентификатора импорта из ПКД;
- **Api Token** — параметр, используемый при работе методов [setbanofauditorium](#)^[113] и [deletebanofauditorium](#)^[130].

5.2. Параметры

Параметры учебного года задаются с помощью функции *Сервис > Параметры*.

В различных образовательных учреждениях в различные временные периоды учебный год может начинаться в разное время. Начало учебного года влияет на номер недели, используемый в конструкторе расписания и в отчетах.

Для каждого *Года* обучения задаются параметры *Начало* (учебного года), *Продолжительность академического часа*, *Перерыв между часами в паре*. Последние два параметра используются в системе по умолчанию при формировании недельной сетки пар и в других процессах.

5.3. Работа с ускорителем

Модуль Xafari.Accelerator - специальный модуль, разработанный для решения проблем с производительностью при работе с моделью приложения. Так применение Ускорителя позволяет существенно сократить время запуска приложения.

По умолчанию ускоритель только подключен к приложению, чтобы начать его использовать, необходимо его запустить. Для этого администратору приложения доступна специальная команда *Запустить Ускоритель* из меню **Инструменты**.



Для работы ускорителя требуется, выполнение приложения в режиме .NET 4.5.1+.

Запуск Ускорителя

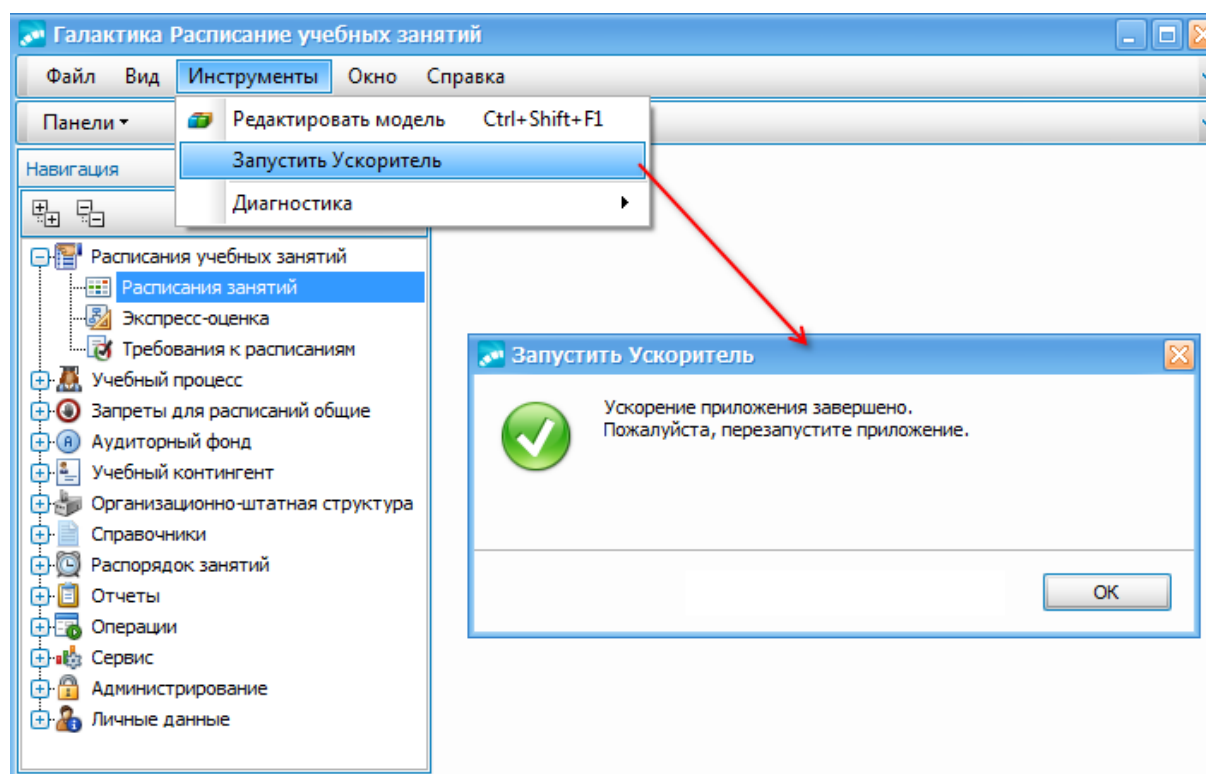


Рис. 32. Запуск ускорителя

Ускоритель начинает работу только после перезапуска приложения.

Как это работает?

Работа ускорителя заключается в том, что он выгружает полную неизменяемую часть модели приложения для текущей комбинации модулей и их версий в отдельный xml-файл. Также могут быть получены и файлы локализации, если локализация присутствует. В дальнейшем ускоритель модифицирует модель приложения так, чтобы неизменяемая часть модели приложения строилась только на полученном файле.

Остановка Ускорителя

Для остановки Ускорителя администратору приложения доступна команда *Отключить Ускоритель* из меню **Инструменты**.

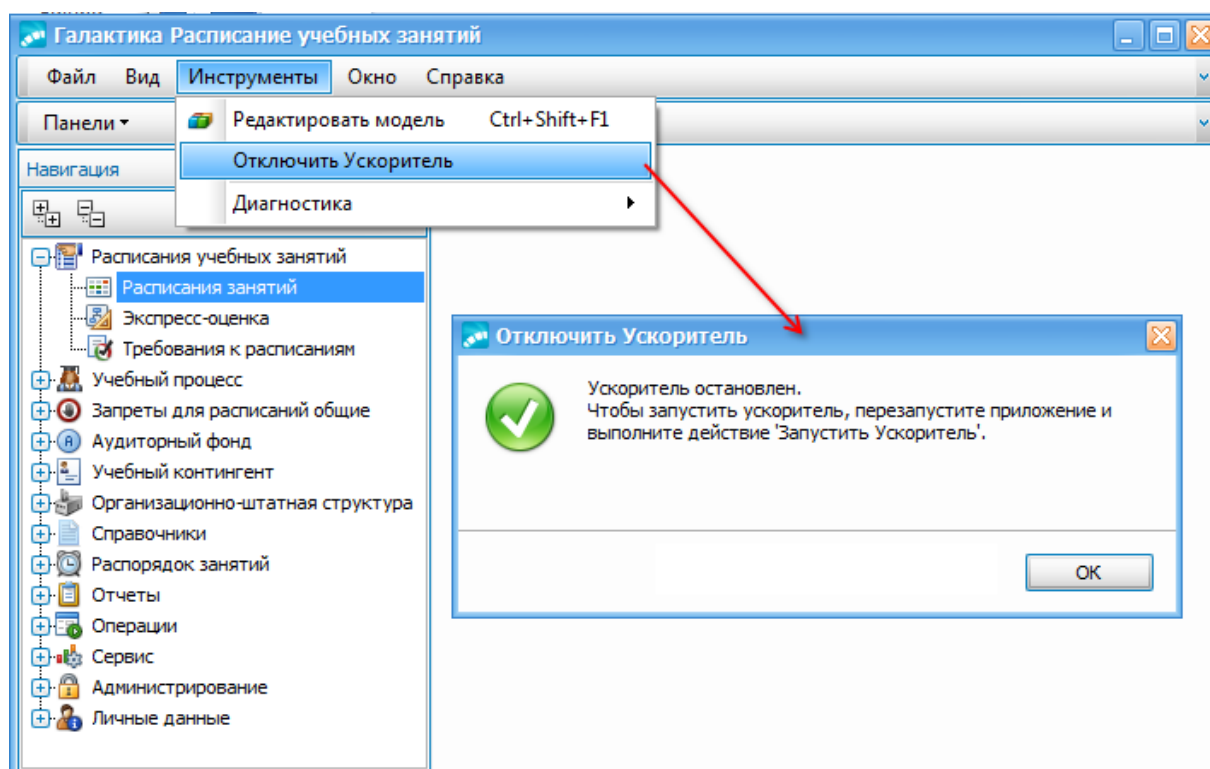


Рис. 33. Остановка ускорителя

При следующем запуске приложения ускоритель будет выключен.

6. Импорт данных

6.1. Импорт данных из xml-файлов

Подсистема импорта системы *Галактика Расписание учебных занятий* (РУЗ) обеспечивает загрузку большинства данных, необходимых для формирования занятий: справочники, профессорско-преподавательский состав, аудиторный фонд, контингент обучающихся, нагрузка расписания.

Технология импорта характеризуется следующими моментами:

- формат импортируемых файлов — xml;
- структура, состав, последовательность импорта определяется файлом настроек [corTable.xml](#)^[64];
- процедура импорта вызывается по требованию пользователя и является уникальной операцией (не может выполняться одновременно более одной процедуры импорта);
- процесс загрузки журналируется с фиксацией и отображением состояния, временных параметров, ошибок, предупреждений;
- данные, некорректные с точки зрения расписания и которые нельзя исправить без потери качества, не загружаются;
- возможен режим обновления данных с установлением соответствия и "выравниванием" с ранее загруженными данными;
- наличие идентификаторов (ID) для загружаемых данных является обязательным — по ним осуществляется последующее установление связей и обновление, тип ID — "длинное целое" (long, Long64, – 9 223 372 036 854 775 808 .. 9 223 372 036 854 775 807);
- [перечень импортируемых объектов](#)^[65] фиксирован, импорт осуществляется в фиксированной на программном уровне последовательности, состав для загрузки определяется оператором;
- для объектов, имеющих ссылочные поля, импортируются только корректные записи, т. е. те записи, связанные поля которых ссылаются на уже существующие в БД РУЗ данные;
- при импорте осуществляется проверка данных на соответствие требованиям РУЗ:
 - контроль длины строковых полей;
 - соблюдение типа, диапазона числовых полей;
 - проверка правил уникальности;
 - контроль обязательности полей.

Общая схема взаимодействия РУЗ с системой автоматизации учебного процесса представлена на рисунке ниже.

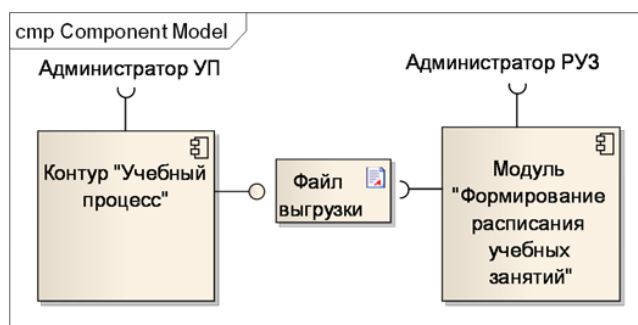


Рис. 34. Схема взаимодействия системы автоматизации учебного процесса и РУЗ

Этапы импорта данных:

1. Подготовка загрузочных файлов с данными в соответствии с требованиями (выполняется средствами подсистемы автоматизации учебного процесса).
2. Импорт (проверка) данных.
3. Анализ результатов импорта.
4. Исправление входных данных и повторный импорт (при необходимости).

6.1.1. Файл настройки (corTable.xml)

Ниже приведены фрагменты файла настройки (corTable.xml). Изменяться могут только подсвеченные поля. Их наименование должно быть идентичным в соответствующих файлах с данными.

```
<?xml version="1.0" encoding="WINDOWS-1251"?>
<FileConf>
<ListTable>
  <table name="Faculty.xml" value="34" name_ru="Факультеты"/>
  <table name="CycleOfDiscipline.xml" value="33" name_ru="Циклы дисциплин"/>
  <table name="KindOfActivity.xml" value="32" name_ru="Виды учебной деятельности"/>
  <table name="TypeOfAuditorium.xml" value="31" name_ru="Типы аудиторий"/>
  <table name="FormOfEducation.xml" value="30" name_ru="Формы обучения"/>
  <table name="Post.xml" value="29" name_ru="Должности"/>
  <table name="Building.xml" value="28" name_ru="Корпуса"/>
  ...
</ListTable>
<Data>
  <Object name="Faculty" type="Faculty">
    <prop key="Code" value="Code"/>
    <prop key="Name" value="Name"/>
    <prop key="Institut" value="Institut"/>
    <prop key="Abbr" value="Abbr"/>
  </Object>
  <Object name="CycleOfDiscipline" type="CycleOfDiscipline">
    <prop key="Code" value="Code"/>
    <prop key="Name" value="Name"/>
    <prop key="Abbr" value="Abbr"/>
    <prop key="TypeOfPlanAge" value="TypeOfPlanAge"/>
  </Object>
  ...
</Data>
</FileConf>
```

6.1.2. Объекты импорта

6.1.2.1. Список объектов

Список файлов для загрузки приведен в таблице (по ссылкам осуществляется переход к содержанию соответствующего объекта). Кодовые обозначения файлов загрузки фиксированы и не могут изменяться пользователем.

Табл. 2. Объекты импорта

№	Наименование	Код
1	Образовательные программы (Факультеты) ¹⁴⁰	Faculty
2	Циклы дисциплин ¹⁴¹	CycleOfDiscipline
3	Виды учебной деятельности ¹⁴¹	KindOfActivity
4	Типы аудиторий ¹⁴¹	TypeOfAuditorium
5	Формы обучения ¹⁴¹	FormOfEducation
6	Должности ¹⁴²	Post
7	Корпуса ¹⁴²	Building
8	Специальности ¹⁴²	Speciality
9	Специализации ¹⁴²	Specialization
10	Виды работ ¹⁴³	KindOfWork
11	Типы кафедр ¹⁴³	ChairType
12	Департаменты (Кафедры) ¹⁴³	Chair
13	Дисциплины ¹⁴⁴	Discipline
14	Преподаватели ¹⁴⁴	Lecturer
15	Аудитории ¹⁴⁵	Auditorium
16	Группы ¹⁴⁶	Group
17	Обучающиеся ¹⁴⁷	Student
18	Составы учебных групп ¹⁴⁸	StaffEduGroup
19	Составы групп ¹⁴⁸	StaffGroup
20	Подгруппы ¹⁴⁸	SubGroup
21	Составы подгрупп ¹⁴⁸	StaffOfSubGroup
22	Потоки ¹⁴⁹	Stream
23	Составы потоков ¹⁴⁹	StaffOfStream
24	Периоды обучения ¹⁵⁰	Semester
25	Содержание нагрузки расписания ¹⁵⁰	ContentOfLoad
26	ГППД ¹⁵²	DenormalizationOfLoad

№	Наименование	Код
27	Предпочтительные аудитории ¹⁵³	AuditoriumPreferred
28	Учебные планы ¹⁵³	Curriculum
29	Расстояния между зданиями ¹⁵⁴	DistanceBetweenBuilding
30	Виды дисциплин ¹⁵⁴	DisciplineKind
31	Типы дисциплин ¹⁵⁴	DisciplineType
32	Аспекты преподавания языковых дисциплин ¹⁵⁴	AspectOfTeachingLangDis
33	Состав аспектов преподавания ¹⁵⁴	StaffOfAspectOfTeachingLangDis
34	Преподаваемые дисциплины ¹⁵⁵	StaffOfDisciplineLecturer
35	Преподаваемые языки ¹⁵⁵	StaffOfLanguageLecturer
36	Ресурсы на языках ¹⁵⁵	ResourceLanguage
37	Языки подгрупп ¹⁵⁶	StaffOfLanguageSubGroup

6.1.2.2. Пример файла для импорта (ContentOfLoad.xml)

Ниже приведен фрагмент файла для импорта нагрузки (ContentOfLoad.xml).

```
<?xml version="1.0" encoding="WINDOWS-1251"?>
<!--Copyright ЗАО "Корпорация Галактика"-->
<Data_Root>
  <Descript ExpSet_Name="ContentOfLoad" ExpSet_Code="023">
    <![CDATA[]]>
  </Descript>
  <Data>
    <Collection child_tags="Object" name="Data.ContentOfLoad" caption="Нагрузка">
      <Object name="Информатика Лекция" id="123458005" class_id="ContentOfLoad">
        <Collection child_tags="prop_value" name="Prop_Values" caption="Свойства">
          <prop_value value="" prop_name="YearOfEducation"/>
          <prop_value value="01.09.2013" prop_name="DateFrom"/>
          <prop_value value="31.12.2013" prop_name="DateTo"/>
          <prop_value value="72" prop_name="Amount"/>
          <prop_value value="0" prop_name="Unit"/>
          <prop_value value="72" prop_name="AmountInUnit"/>
          <prop_value value="0" prop_name="SignOfContinuity"/>
          <prop_value value="2" prop_name="TypeOfContingent"/>
          <prop_value value="" prop_name="ID_Stream" rlt_class=""/>
          <prop_value value="123457729" prop_name="ID_Group" rlt_class=""/>
          <prop_value value="" prop_name="ID_SubGroup" rlt_class=""/>
          <prop_value value="123457476" prop_name="ID_KindOfWork" rlt_class=""/>
          <prop_value value="123457516" prop_name="ID_Discipline" rlt_class=""/>
          <prop_value value="123457602" prop_name="ID_Lecturer" rlt_class=""/>
          <prop_value value="" prop_name="ID_Chair" rlt_class=""/>
          <prop_value value="" prop_name="ID_CycleOfDiscipline" rlt_class=""/>
          <prop_value value="" prop_name="PackageNumber"/>
          <prop_value value="" prop_name="ID_Auditorium" rlt_class=""/>
          <prop_value value="" prop_name="ID_Semester" rlt_class=""/>
          <prop_value value="" prop_name="Remotely"/>
        </Collection>
      </Object>
      ...
    </Collection>
  </Data>
</Data_Root>
```

```
</Data>  
</Data_Root>
```

6.1.3. Подготовка и выгрузка данных

Для удобства подготовки данных для импорта при отсутствии системы автоматизации учебного процесса может использоваться специальный excel-шаблон ForImport.xlsm, поставляемый вместе с продуктом. Схема действий пользователя в этом случае представлена на рисунке ниже.

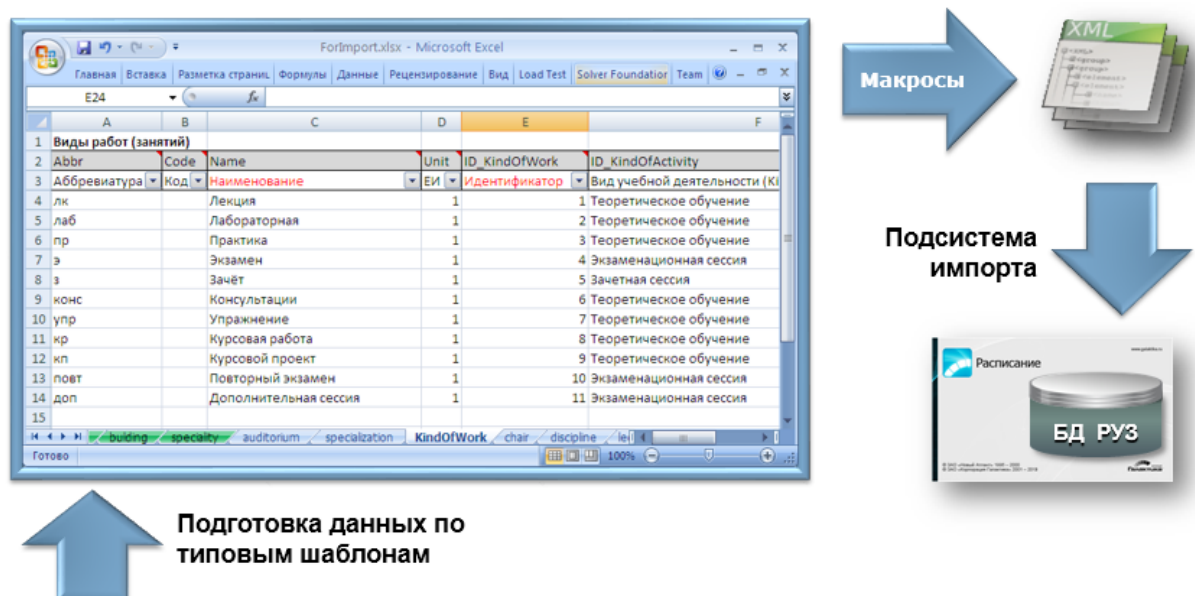


Рис. 35. Схема подготовки и импорта данных с использованием среды Excel

Файл состоит из листов, каждый из которых соответствует определенному справочнику и имеет фиксированную структуру.

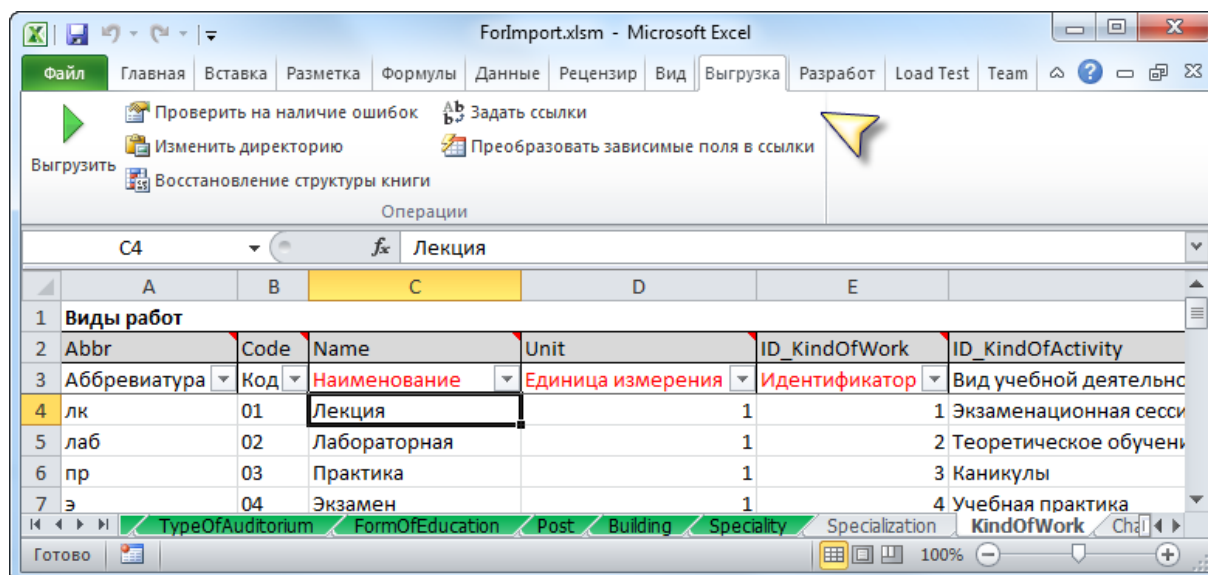


Рис. 36. Файл-шаблон для импорта

Листы оформлены в соответствии со следующими правилами:

- красным цветом выделены названия обязательных полей;

- серым фоном выделены дополнительные поля, используемые только для идентификации записей из связанных листов, а не для выгрузки;
- каждое поле в заголовке имеет описание (всплывающую подсказку), содержащее тип поля, признак обязательности и пояснения (если они есть);
- заголовок закреплен сверху листа и запрещен для редактирования;
- листы-справочники, не имеющие ссылок на другие справочники, подсвечены зеленым цветом.

Внешние поля, т. е. имеющие ссылки на связанные записи из других листов, делятся на 2 типа: основные и дополнительные.

Основные внешние поля доступны для копирования/вставки/удаления и работы с маркером автозаполнения.

Дополнительные поля подсвечиваются серым цветом. Для отображения/скрытия таких полей (при их наличии) следует дважды кликнуть на заголовок основного внешнего поля. Значения дополнительных полей меняются в зависимости от значений основного поля и недоступны для редактирования.

Для выполнения операций, связанных с выгрузкой, используется специальное меню **Выгрузка**:

- **Выгрузить** — обеспечивает запуск процесса выгрузки данных из Excel в xml-файлы. По умолчанию xml-файлы сохраняются в папку result, находящуюся в папке с файлом-шаблоном. Если такой папки не существует, она будет создана.

При запуске функции открывается окно настройки параметров импорта:

- **Вывод лога** — сообщения журнала выводятся **На экран** или **В текстовый файл**.
- **Проверки** — в процессе выгрузки данные обязательно проверяются на **Соответствие типов**, имеется возможность проверки данных на **Уникальность**.
- **Разбиение данных** — для больших объемных листов предоставляется возможность **Разбивать данные на файлы**. Объем данных для каждого файла определяется параметром **В файл выгружать не более ... записей** (по умолчанию — 1000 записей). В этом случае к названию 2-го и последующих файлов добавляется номер пакета (2, 3 и т. д.).

Примечание. При импорте xml-файлов в систему **Галактика Расписание учебных занятий** к обработке принимаются файлы, имя которых совпадает с именем, заданным в файле [corTable.xml](#)⁶⁴

- **Последовательность действий** — при выполнении функции позволяет осуществлять **Преобразование зависимых полей в ссылки**. **Проверка на наличие ошибок** и **Выгрузка** выполняются в обязательном порядке.

При обнаружении ошибки в процессе выгрузки выводится соответствующая запись в журнал. Строка, в которой обнаружена ошибка, не выгружается, а ошибочные данные выделяются другим цветом.

 Для выполнения выгрузки необходимо, чтобы файл-шаблон ForImport.xlsm и файл настроек corTable.xml располагались в одном каталоге.

- **Проверить на наличие ошибок** — используется для проверки введенных данных, при этом осуществляются журнализация и подсветка ошибок.
- **Изменить директорию** — позволяет настроить каталог выгрузки для текущего сеанса (без сохранения для последующих).

- *Восстановление структуры книги* — обеспечивает восстановление структуры книги и листов. Предназначена для ситуаций, когда нужно отменить выбор внешнего поля либо скрыть дополнительные поля.
- *Задать ссылки* — используется для группового указания ссылки. Доступна, если выделены ячейки в пределах одного столбца и этот столбец является основным внешним.
- *Преобразовать зависимые поля в ссылки* — обеспечивает преобразование для ссылочных полей введенных вручную значений в ссылки. Выполняется аналогично функции *Выгрузить* для всех внешних полей. Не преобразует два типа текста: некорректный (не найденный в зависимой таблице) и неоднозначный (текст, не уникальный в зависимой таблице), соответствующие предупреждения выводятся в журнал. Все ошибочные данные подсвечиваются светло-зеленым цветом. После преобразования автоматически заполняются дополнительные внешние поля.

6.1.4. Импорт данных

Импорт данных в систему *Галактика Расписание учебных занятий* осуществляется с помощью функции *Операции > Импорт данных*. При ее запуске открывается окно параметров импорта.

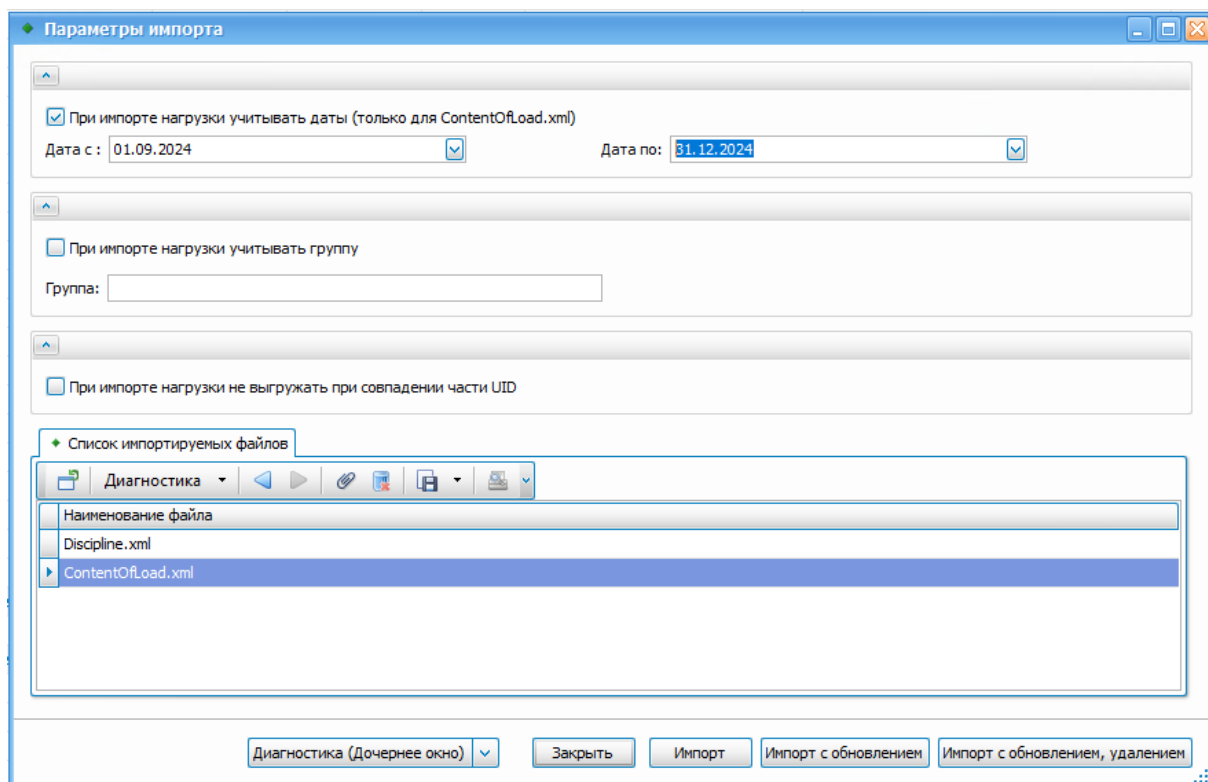


Рис. 37. Параметры импорта

При импорте нагрузки (ContentOfLoad.xml) существует возможность учитывать даты и группу (предварительно проставить соответствующие флажки).

Если в *Общих параметрах* указан [Импорт по UID](#)⁶⁰, существует возможность использовать параметр *При импорте нагрузки не выгружать при совпадении части UID*. При установленном флаге в случае, если в РУЗ уже есть нагрузка с аналогичной частью идентификатора UID (часть идентификатора, отвечающего за нагрузку, а не за преподавателя), то такая нагрузка не будет загружена в РУЗ.

Добавление импортируемых файлов осуществляется с помощью кнопки  на панели инструментов или комбинации клавиш **Ctrl+N**.

Запуск импорта осуществляется с помощью кнопок:

- **[Импорт]** — первичный и повторный импорт данных. В последнем случае происходит добавление отсутствующих в справочниках данных;
- **[Импорт с обновлением]** — при повторном импорте происходит обновление уже имеющихся и добавление отсутствующих в справочниках данных;
- **[Импорт с обновлением, удалением]** — при повторном импорте происходит обновление уже имеющихся, добавление отсутствующих в справочниках данных и удаление из справочников отсутствующих в файлах импорта данных.

Процесс импорта сопровождается ведением протокола загрузки. В него помещаются все сообщения о нарушениях корректности импортируемой информации: дублирование данных (нарушение уникальности); нарушение границ диапазонов полей; нарушение правил сравнения объектовых записей; нарушение размерности текстовых полей; некорректность ссылочных данных.

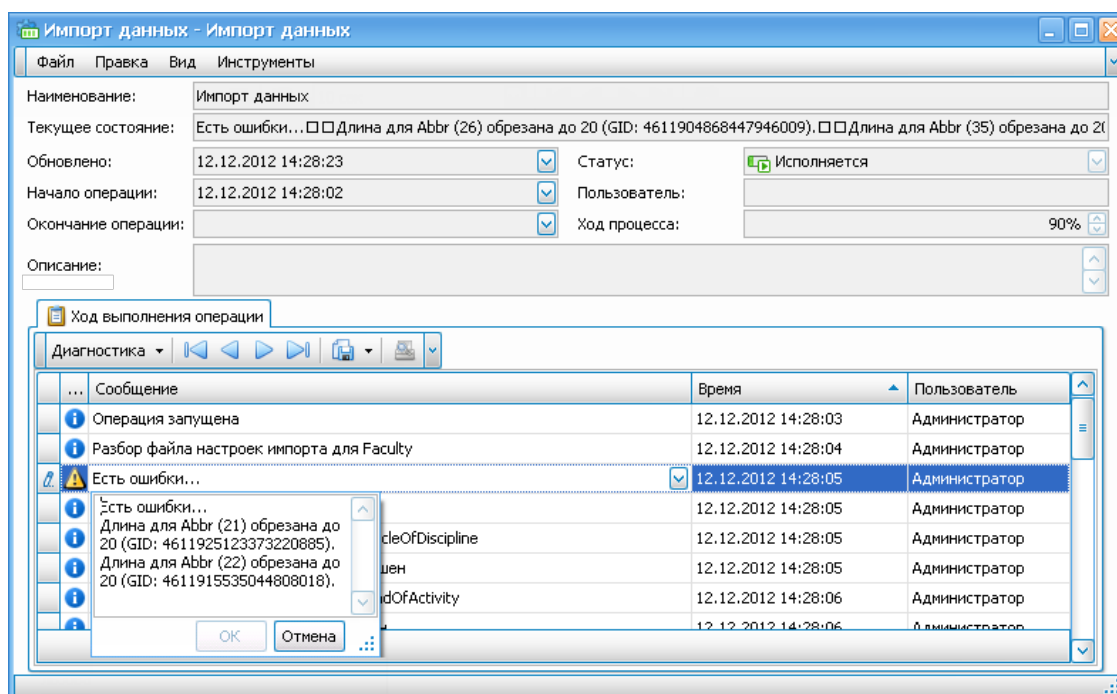


Рис. 38. Проверка корректности данных

 В импортируемых данных может присутствовать не вся информация, необходимая для составления расписания, поэтому часть данных придется заполнить вручную.

При импорте данных в справочниках заполняется поле **Внешний ID** — позволяет настроить (посмотреть) связь импортируемых данных и ранее загруженных. Поле может также использоваться при [подготовке данных](#)⁶⁷ для импорта с использованием шаблона ForImport.xlsm.

После загрузки данных необходимо произвести расчет пересеченности контингента (п 4.4 Руководства пользователя системы Галактика расписание учебных занятий) с помощью функции **Учебный контингент > Пересеченность контингента**

В сформированной сводной форме выполнить команду **Рассчитать**.

Пересеченность контингента - Галактика Расписание учебных занятий [Версия: 3.7.22 | БД: RUZ.TEST.2022]

Файл Редактирование Вид Инструменты Окно Справка

Детализация Панели

Навигация

Содержание нагрузок Группы Поток Подгруппы Расписания занятий Пересеченность контингента

Год ...	Форма об...	Факультет/обр.прогр.	Курс	Поток	Группа	Подгруппа	Пер...
2021	Очная фо...	Институт бизнес-коммуник...	1		1-СД-11		1-С
2021	Очная фо...	Институт бизнес-коммуник...	1		1-СД-10		1-С
2021	Очная фо...	Институт бизнес-коммуник...	1		1-МГ-48		1-М
2021	Очная фо...	Институт бизнес-коммуник...	1		1-МГ-32		1-М
2021	Очная фо...	Институт бизнес-коммуник...	1		1-МГ-25		1-М
2021	Очная фо...	Институт бизнес-коммуник...	1			1-СД-11/10	1-С
2021	Очная фо...	Институт бизнес-коммуник...	1	1-СД-10; 1-СД-11#Деловые...			
2021	Очная фо...	Институт бизнес-коммуник...	1			1-СД-11/9	1-С
2021	Очная фо...	Институт бизнес-коммуник...	1	1-СД-10; 1-СД-11#Деловые...			
2021	Очная фо...	Институт бизнес-коммуник...	1			1-СД-11/8	1-С
2021	Очная фо...	Институт бизнес-коммуник...	1	1-СД-10; 1-СД-11#Деловые...			
2021	Очная фо...	Институт бизнес-коммуник...	1			1-СД-11/7	1-С
2021	Очная фо...	Институт бизнес-коммуник...	1	1-СД-10; 1-СД-11#Деловые...			

Рис. 39. Расчет пересеченности контингента

6.1.5. Импорт из Галактики ERP

Рассмотрим вариант, когда планирование учебного процесса осуществляется в системе автоматизации **Галактика ERP**, включая распределение нагрузок.

Если к этому моменту имеются еще неопределенные данные (преподаватели, группы, аудитории), то в соответствующие справочники вводится обезличенная информация. Например, вводятся фиктивные группы, преподаватели типа *Вакансия* и эти данные используются при назначении нагрузок. Это необходимо, чтобы при выгрузке данных для системы **Галактика Расписание учебных занятий** максимально был сформирован файл для нагрузки расписания. В противном случае эту нагрузку придется вводить вручную.

Далее осуществляется [выгрузка данных](#)⁷¹ из **Галактики ERP** в xml-файлы, а затем [загрузка данных](#)⁷⁴ из этих файлов в систему **Галактика Расписание учебных занятий**.

 В импортируемых из **Галактики ERP** данных может присутствовать не вся информация, необходимая для составления расписания, поэтому после импорта часть данных придется заполнить вручную.

6.1.5.1. Экспорт данных

Для экспорта данных из системы **Галактика ERP** следует поместить файл EIO.cfg в папку EXE/xml (папка установки системы **Галактика ERP**).

 Файл EIO.cfg поставляется вместе с обновлениями **Галактики ERP** для учебного процесса (например, U_StProc).

В модуле **Обмен бизнес-документами** выбирается функция XML > Типы данных для экспорта-импорта и проводится инициализация данных.

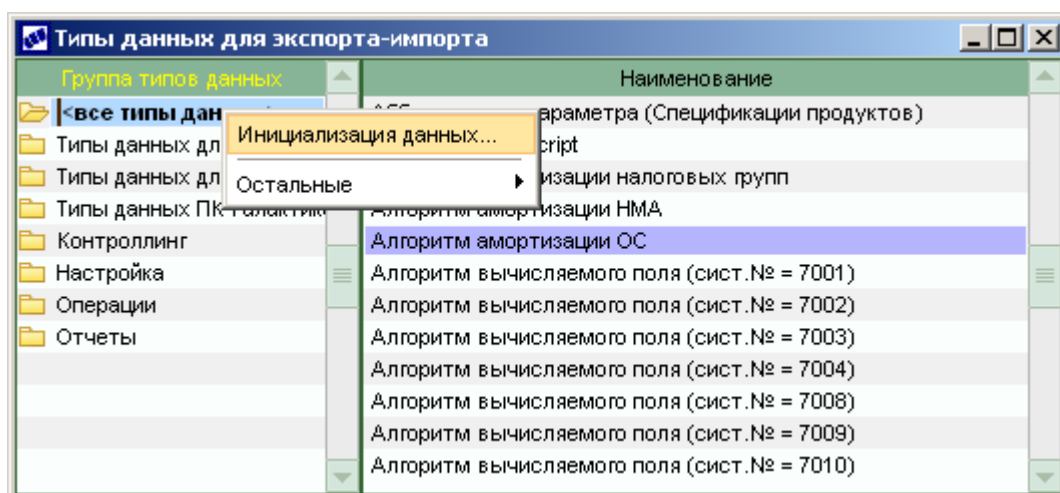


Рис. 40. Инициализация данных

С помощью функции *XML > Экспорт данных в XML* создаются настройки экспорта:

- Faculty (факультет);
- Chair (кафедра);
- CycleOfDiscipline (цикл компонентов/дисциплин);
- KindOfActivity (виды учебной деятельности);
- KindOfWork (виды работ);
- Post (должности);
- Discipline (дисциплины);
- FormOfEducation (форма обучения);
- Speciality (специальность);
- Specialization (специализация);
- TypeOfAuditorium (тип аудитории);
- Lecturer (преподаватели);
- Auditorium (аудитории);
- Building (корпуса);
- Student (обучающиеся);
- Group (группа);
- SubGroup (подгруппа);
- StaffOfSubGroup (состав подгруппы);
- Stream (поток);
- StaffOfStream (состав потока);
- ContentOfLoad (нагрузка).

В каждой настройке для группы параметров *ТД для экспорта* выбираем *Тип данных для экспорта*, соответствующий настройке.

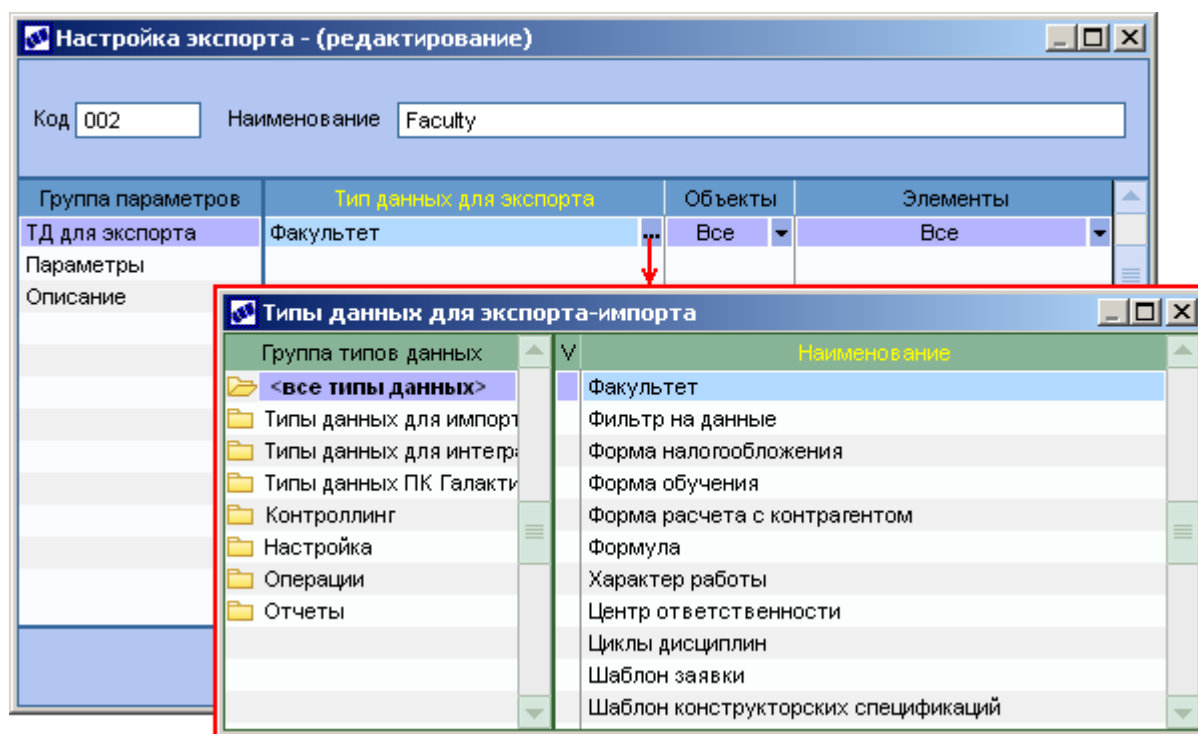


Рис. 41. Настройка экспорта

Экспорт осуществляется из списка настроек с помощью соответствующей кнопки.

Внимание! Приведенные ниже типы данных должны экспортироваться именно в указанной последовательности:

- 1) *Stream (поток)*;
- 2) *StaffOfStream (состав потока)*;
- 3) *ContentOfLoad (нагрузка)*.

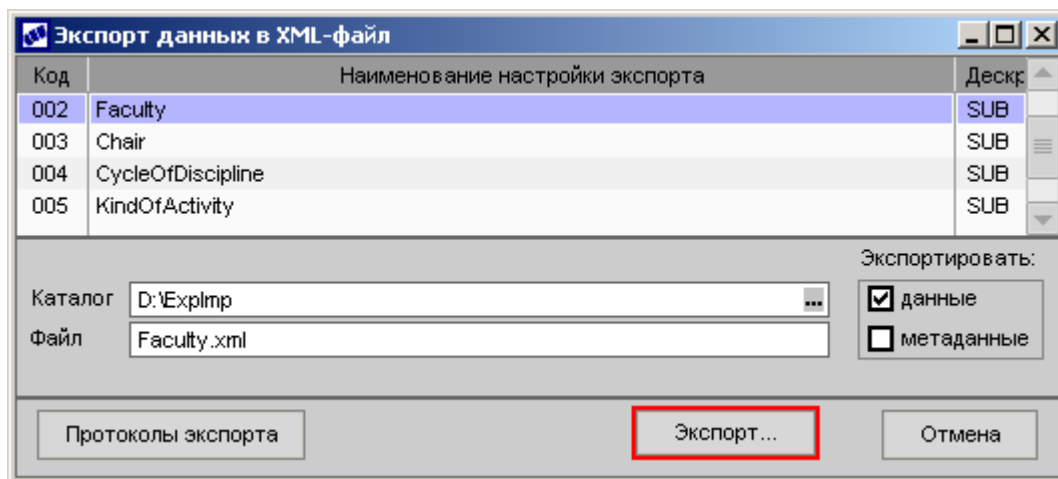


Рис. 42. Экспорт

После завершения этой операции анализируются финальные сообщения и осуществляется визуальный анализ сформированных файлов. Например, нулевая размерность файла означает отсутствие данных (наименование файла указывает на соответствующий справочник или категорию информации). Таким образом, можно выявить некоторые проблемы еще до операции загрузки информации в РУЗ.

6.1.5.2. Импорт данных

После выгрузки данных из системы **Галактика ERP** осуществляется [импорт](#)^[69] этих данных в систему **Галактика Расписание учебных занятий**.

Операция импорта сопровождается журналированием процесса и выводом сообщений о ходе процесса, предупреждениях, ошибках. Журнал рекомендуется сохранять по завершении операции для последующего использования при устранении ошибок. Информационные сообщения типа предупреждения (например, обрезана длина строки) носят рекомендательный характер и не оказывают существенного влияния на дальнейшие действия. При анализе журнала особое внимание следует обращать на ошибки, указывающие на несохранение записей в БД системы **Галактика Расписание учебных занятий**. Здесь необходимо понять из текста причину и принять решение о критичности этой ошибки и способе устранения (игнорирования):

- исправить ситуацию в системе **Галактика ERP** и повторить процедуру экспорта-импорта;
- исправить (дополнить) ситуацию средствами системы ведения РУЗ.

Следует учитывать следующее обстоятельство. Данные обладают связностью и незагрузка объекта некоторого справочника может быть причиной невозможности загрузки последующих данных. Например, нарушение уникальности в справочнике должностей (Post.xml) повлечет незагрузку преподавателей (Lecturer.xml), находящихся на повторяющихся должностях и далее не позволит осуществить импорт по ним записей загрузки расписания (ContentOfLoad.xml).

6.2. Импорт данных из ПКД

В системе **Галактика Расписание учебных занятий** предусмотрена возможность импорта данных из внешних информационных систем с использованием [промежуточного кластера данных](#) (ПКД). При этом данные из внешней системы экспортируются напрямую в ПКД, размещаемый внутри БД РУЗ в виде дополнительных таблиц.

Возможен импорт данных с использованием идентификаторов числового типа (GID), либо строкового типа (UID). Выбор типа идентификатора осуществляется в [Общих параметрах](#)^[60] системы. Для всех импортируемых объектов должны использоваться идентификаторы одного типа.

Табл. 3. Таблицы ПКД

№	Описание	Таблица
1	Образовательные программы (Факультеты)	iFaculty
2	Циклы дисциплин	iCycleOfDiscipline
3	Виды учебной деятельности	iKindOfActivity
4	Типы аудиторий	iTypeOfAuditorium
5	Формы обучения	iFormOfEducation
6	Должности	iPost
7	Корпуса	iBuilding
8	Специальности	iSpeciality
9	Специализации	iSpecialization
10	Виды работ	iKindOfWork

№	Описание	Таблица
11	Департаменты (Кафедры)	iChair
12	Дисциплины	iDiscipline
13	Преподаватели	iLecturer
14	Аудитории	iAuditorium
15	Группы	iGroup
16	Обучающиеся	iStudent
17	Составы групп	iStaffGroup
18	Подгруппы	iSubGroup
19	Составы подгрупп	iStaffOfSubGroup
20	Потоки	iStream
21	Составы потоков	iStaffOfStream
22	Периоды обучения	iSemester
23	Содержание нагрузки расписания	iContentOfLoad
24	ГППД	iDenormalizationOfLoad
25	Предпочтительные аудитории	iAuditoriumPreferred
26	Учебные планы	iCurriculum
27	Расстояния между зданиями	iDistanceBetweenBuilding
28	Ресурсы на языках	iResourceLanguage

[Содержание импортируемых объектов](#)¹⁴⁰ почти такое же, как и при импорте данных с использованием шаблона ForImport.xlsm, ниже приведены лишь две отличающиеся таблицы.

Табл. 4. Содержание нагрузки расписания (iContentOfLoad)

Name	Code	Type	P	M
Год обучения	YearOfEducation	int		
Дата начала	StartDate	date		X
Дата завершения	EndDate	date		X
Количество ак. часов ¹	Amount	float		X
Тип контингента ²	TypeOfContingent	int		X
ID потока ³	ID_Stream	long		
UID потока ³	UID_Stream	char(2000)		
ID группы ⁴	ID_Group	long		
UID группы ⁴	UID_Group	char(2000)		
ID подгруппы ⁵	ID_SubGroup	long		
UID подгруппы ⁵	UID_SubGroup	char(2000)		
ID вида работы	ID_KindOfWork	long		X
UID вида работы	UID_KindOfWork	char(2000)		X
ID дисциплины	ID_Discipline	long		X

Name	Code	Type	P	M
UID дисциплины	UID_Discipline	char(2000)		X
ID дисциплины в плане	ID_DisciplineInPlan	long		
UID дисциплины в плане	UID_DisciplineInPlan	char(2000)		
ID преподавателя	ID_Lecturer	long		X
UID преподавателя	UID_Lecturer	char(2000)		X
ID кафедры	ID_Chair	long		
UID кафедры	UID_Chair	char(2000)		
ID цикла дисциплины	ID_CycleOfDiscipline	long		
UID цикла дисциплины	UID_CycleOfDiscipline	char(2000)		
ID нагрузки расписания	ID_ContentOfLoad	long	X	X
UID нагрузки расписания	UID_ContentOfLoad	char(2000)	X	X
ID специализации	ID_Specialization	long		
UID специализации	UID_Specialization	char(2000)		
Единица измерения (ЕИ) ⁶	Unit	int		X
Количество в ЕИ	AmountUnit	int		X
Признак непрерывности ⁷	SignOfContinuity	boolean		
Номер пакета ⁸	PackageNumber	int		
ID аудитории ⁹	ID_Auditorium	long		
ID семестра	ID_Semester	long		
Читается дистанционно ¹⁰	Remotely	boolean		
Тип дисциплины в нагрузке ¹¹	DisciplineTypeLoad	int		
Язык преподавания ¹²	ID_Language	long		

¹ Заполняется всегда (в академических часах).

² 1 — подгруппа, 2 — группа, 3 — поток.

³ Заполняется, если TypeOfContingent = 3.

⁴ Заполняется, если TypeOfContingent = 2.

⁵ Заполняется, если TypeOfContingent = 1.

⁶ 0 — часы (лекции, практики и т. п.), 1 — единицы (экзамен, зачет, практика и т. п.).

⁷ Значение true устанавливается для нагрузок, подразумевающих неделимость (экзамен, практика т. п.).

⁸ Устанавливается значение 0, если нагрузка непaketная, иначе — номер, одинаковый для всех нагрузок, входящих в пакет.

⁹ Обязательно указывается для пакетных нагрузок; корректность накладок аудиторий в одном пакете — это задача методиста (в РУЗ это принимается за аксиому); для непaketных нагрузок в случае необходимости может указываться несколько [предпочтительных аудиторий](#)¹⁵³.

¹⁰ Если этот флаг установлен, то в поле "ID аудитории" указывается обязательно ссылка на аудиторию "Дистанционная" (она должна быть предустановлена в этом

справочнике). При импорте будет выполняться попытка установить соответствие по "ID аудитории" и по наименованию "Дистанционная".

¹¹ 0 — обычная обязательная дисциплина (по умолчанию), 1 — дисциплина по выбору, 2 — факультатив.

¹² 1 — русский (по умолчанию), 2 — английский.

Дополнительные правила:

- 1) StartDate < EndDate.
- 2) StartDate, EndDate не могут быть шире диапазона [Semester.StartDate, Semester.EndDate].
- 3) YearOfEducation должен быть согласован с Semester.

Правила уникальности:

- StartDate, ID_Discipline, ID_KindOfWork, ID_Lecturer, ID_Stream;
- StartDate, ID_Discipline, ID_KindOfWork, ID_Lecturer, ID_Group;
- StartDate, ID_Discipline, ID_KindOfWork, ID_Lecturer, ID_SubGroup.

 **Важно:**

1) В правилах не учитываются записи, если есть поле со значением null.

2) В правилах все перечисленные поля обязательны, кроме контингентных (ID_Stream, ID_Group, ID_SubGroup).

3) В зависимости от TypeOfContingent заполняется только одно из контингентных полей.

Табл. 5. Ресурсы на языках (iResourceLanguage)

Name	Code	Type	P	M
ID ресурса на языке	ID_ResourceLanguage	long	X	X
ID языка ¹	ID_Language	long		X
Объект ресурса ²	ObjectApply	int		X
ID объекта ресурса ³	ID_ObjectOID	long		X
Аббревиатура	Abbr	chair(80)		
Наименование	Name	chair(200)		X

¹ 1 — русский (по умолчанию), 2 — английский.

² 1 — Discipline (дисциплина), 2 — Lecturer (преподаватель), 3 — Post (должность), 4 — KindOfWork (Вид работы), 5 — Building (здание), 6 — Auditorium (аудитория), 7 — Stream (поток), 8 — Group (группа), 9 — Subgroup (подгруппа).

³ Содержит iDiscipline.ID_Discipline для ObjectApply = 1, iLecturer.ID_Lecturer для ObjectApply = 2 и т. д.

 В таблицу iResourceLanguage необходимо заносить наименования/аббревиатуры ресурсов только на иностранных языках, т. к. на базовом (русском) языке эти значения содержатся в основных сущностях.

Порядок импорта

Импорт данных из ПКД предполагает следующую последовательность действий:

1. Экспорт данных из внешней системы в ПКД.

2. Настройка в *Общих параметрах* [идентификатора импорта](#)⁶⁰.
3. Настройка параметров импорта.
4. Импорт данных из ПКД в основные объекты РУЗ.
5. Анализ результатов импорта.
6. Исправление во внешней системе незагруженных данных и повторение процедуры экспорта-импорта.

Экспорт осуществляется средствами внешней системы. При этом экспортируемые для РУЗ данные должны удовлетворять следующим общим требованиям:

- состав определяется ограниченным перечнем объектов;
- выполняются требования по корректности, согласованности, целостности данных объектов выгрузки и данных пакета выгрузки в целом, требования по согласованности с данными предыдущих выгрузок;
- экспортируются только измененные во внешней системе данные с момента последней выгрузки;
- экспортируемые данные записываются в объекты ПКД в режиме добавления, не изменяя и не удаляя существующих записей;
- экспортируемые данные не должны нарушать логику уже составленных в РУЗ расписаний;
- идентификация записей (связей) осуществляется по идентификаторам типа "длинное целое" (long, Int64, - 9 223 372 036 854 775 808 .. 9 223 372 036 854 775 807).

Для выполнения этапов 3–5 в системе **Галактика Расписание учебных занятий** предназначена функция меню *Операции > Импорт данных из ПКД*. При ее запуске открывается журнал, в котором фиксируются все сеансы импорта данных. В локальном меню и на панели инструментов содержатся функции, управляющие процессом импорта.

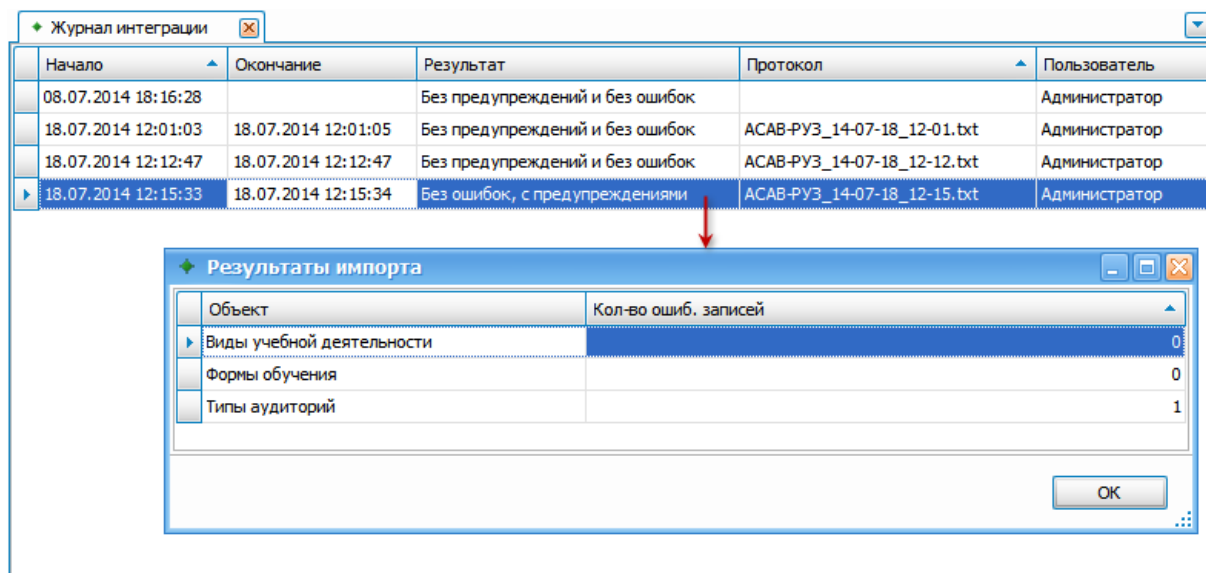


Рис. 43. Журнал интеграции

С помощью функции *Параметры импорта* задаются настройки:

- **Правило очистки:**

- В процессе импорта — удаление из ПКД ошибочных данных, которые невозможно импортировать в систему, осуществляется автоматически непосредственно в процессе импорта данных;
- По требованию — удаление ошибочных данных осуществляется вручную с помощью функции  *Очистка ошибочных данных*;
- **EMail-уведомление, EMail ответственного** — при наличии ошибок импорта позволяют отправлять уведомление на указанный электронный адрес;
- **Каталог размещения файлов** — указывается каталог, в который будут помещаться протоколы импорта;
- **Флаг активности загрузки** - если установлен в "1", то процедура импорта данных из ПКД не запустится. Выставляется вручную для исключения возможных ошибок в случае, когда не завершен процесс загрузки данных в ПКД и по расписанию может запуститься процесс импорта данных из ПКД. Для установки флага записать в поле iParametr.ActiveLoad значение "1", для снятия - "0". Также можно установить и снять флаг активности в пункте меню "Настройка импорта из ПКД" РУЗ;
- **Автоматическое создание резервных копий ДО/ПОСЛЕ импорта** - устанавливаются соответствующие флаги и указывается каталог на сервере для сохранения резервных копий.

Для просмотра информации о количестве записей, находящихся в ПКД и подлежащих импорту, используется функция  *Данные для импорта*.

Запуск процедуры импорта данных из ПКД в основные таблицы БД осуществляется с помощью функции *Импортировать*.

Перечень импортируемых объектов определяется файлом настроек [corTable.xml](#)^[64] (включен в состав системы).

При импорте обрабатываются только новые записи, при этом осуществляется проверка данных на соответствие требованиям РУЗ:

- контроль длины строковых полей;
- соблюдение типа, диапазона числовых полей;
- проверка правил уникальности;
- контроль обязательности полей;
- другие требования.

В процессе импорта возможны следующие операции в основных объектах (таблицах) БД РУЗ:

- добавление импортируемой записи, если такая запись не найдена;
- полное обновление записи, если такая запись уже существует;
- удаление записи, если импортируемая запись отмечена специальным флагом и это не противоречит целостности существующих данных и завершенности связанного бизнес-процесса.

 *Установление соответствия импортируемой записи объекта с уже существующими в основном объекте БД РУЗ выполняется по идентификаторам (первичным ключам).*

Обработанные без ошибок записи удаляются из ПКД (если установлено соответствующее **Правило очистки**). Ошибочные данные не импортируются, для

них указывается ссылка на запись журнала и заполняется специальное поле результата.

В процессе импорта данных из ПКД БД формируется:

- протокол (log-файл);
- запись в журнале интеграции.

Просмотреть протокол можно с помощью кнопки [\[Показать историю выполнения\]](#) в окне процесса импорта или открыв в проводнике соответствующий log-файл. В протокол выводится информация:

- момент, стадия, импортируемый объект;
- момент, наименование конфликта, идентифицирующие параметры обрабатываемой записи.

Каждый сеанс импорта фиксируется в специальном объекте ПКД — в журнале интеграции.

Табл. 6. Журнал интеграции (iLog)

Name	Code	Type
ID журнала импорта	ID_Log	long
Пользователь	RUZUser	long
Начало	StartDate	datetime
Окончание	EndDate	datetime
Протокол	LogName	char(100)
(е) Результат ¹	Rezult	int

¹ Возможные значения: 0 — без предупреждений и без ошибок; 1 — без ошибок, с предупреждениями; 2 — с ошибками.

Для регулирования обновления данных в ПКД при загрузке в его объекты данных, а также для отслеживания результатов импорта в каждом из импортируемых объектов используются следующие служебные поля.

Табл. 7. Служебные поля объектов ПКД

Name	Code	Type
Служебный флаг ¹	TempFlag	smallint
(е) Результат ²	Rezult	smallint
Ссылка на журнал интеграции	iLog	long
Текстовый идентификатор	UID	char(2000)

¹ Возможные значения: 0 — новая или обновляемая запись, значение по умолчанию; 1 — запись для удаления; 2 — запись обработана

² Возможные значения:

- 0 — значение по умолчанию;
- 10 — добавлена без ошибок; 11 — добавлена с предупреждениями; 12 — не добавлена из-за ошибок;
- 20 — обновлена без ошибок; 21 — обновлена с предупреждениями; 22 — не обновлена из-за ошибок;

- 30 — удалена без ошибок; 31 — удалена с предупреждениями; 32 — не удалена из-за ошибок.

Операция импорта из ПКД может инициироваться двумя способами:

- вручную при вызове функции *Операции > Импорт данных из ПКД > Импортировать*;
- при вызове из командного файла.

Варианты запуска операции из командной строки:

runcmd /c runimport /logon AuthenticationType:ActiveDirectory

runcmd /c runimport /logon AuthenticationType:Standard UserName:<указать пользователя> Password:<указать пароль>

Здесь:

- *runcmd* — наименование утилиты;
- *runimport* — зарегистрированное в приложении наименование команды импорта из ПКД.

В первом варианте используется windows-аутентификация, во втором — стандартная.

Операция таким способом может быть запущена из планировщика заданий ОС.

 *Используемая для активации операций из командного файла специальная утилита `runcmd.exe` поставляется вместе с приложением и располагается в каталоге приложения. Для вывода на экран краткой справки по использованию утилиты можно использовать один из ключей: `/?`, `/h`, `/help`. При работе утилиты используется конфигурационный файл `runcmd.exe.config`. В нем, в частности, должна быть корректно указана строка [соединения с базой данных](#)^[39] (ключ `ConnectionString`).*

В результате выполнения операции, помимо записи в журнал интеграции и формирования протокола:

- формируется запись в журнале уведомлений для администратора системы о результате операции в случае запуска из командного файла;
- автоматически программно формируются уведомления для диспетчеров факультетов/образовательных программ при выявлении касающихся их изменений в содержании нагрузки.

7. Настройка типовых отчетов

7.1. Шаблоны отчетов

Инструментарий решения предоставляет пользователям возможность редактирования существующих шаблонов отчетов или создания новых. Для работы с шаблонами используется *встроенный дизайнер отчетов*, вызвать который можно из отчета с помощью функции  *Показать дизайнер отчетов* на вкладке Шаблоны.

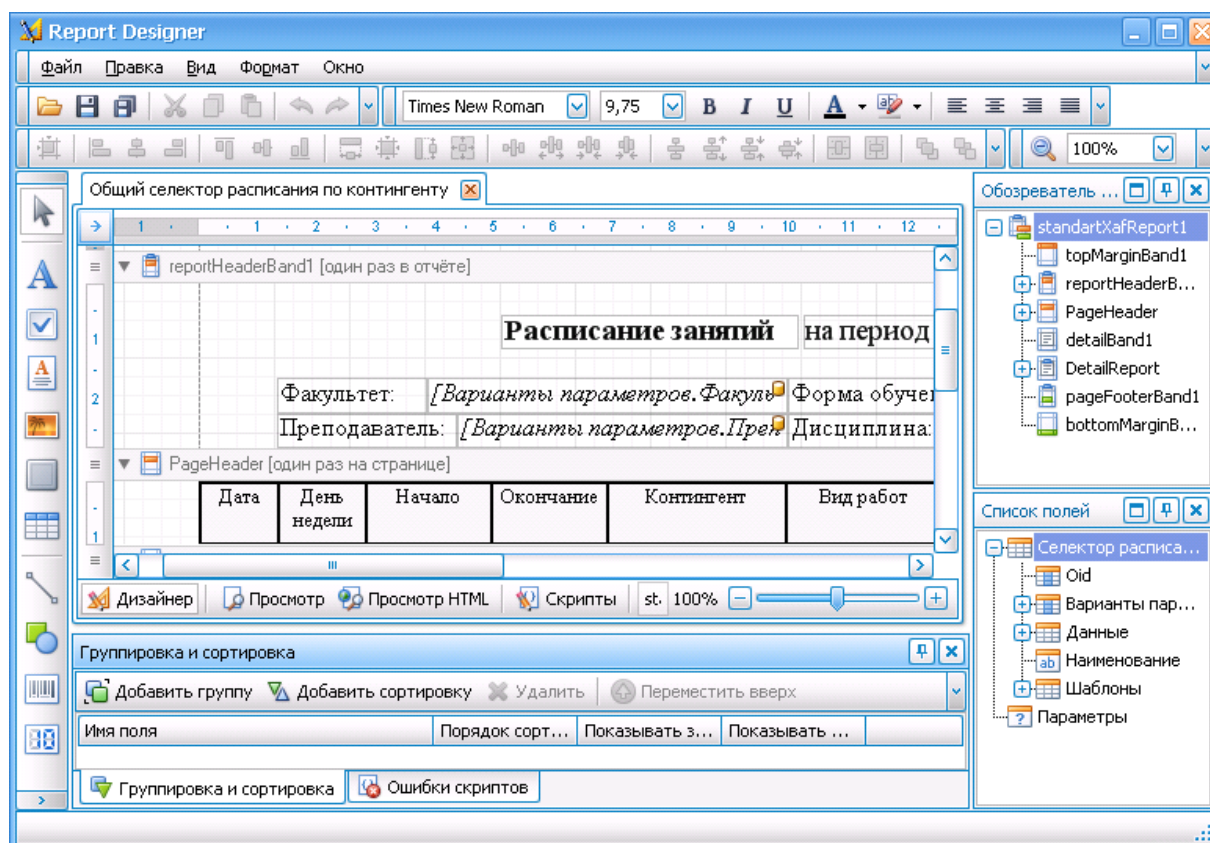


Рис. 44. Дизайнер отчетов

 *Шаблоны печати используют источники данных. У каждого типового отчета свой источник данных. У пользователя нет возможностей для создания своих источников, поэтому при редактировании существующих шаблонов или создании новых шаблонов можно использовать только ту информацию, которая имеется в указанных наборах. При необходимости вывода в печатных формах сведений, отсутствующих в этих наборах, необходимо воспользоваться внешними построителями отчетов с прямым подсоединением к БД расписания.*

Порядок редактирования существующих шаблонов:

- 1) сохранение резервной исходной копии шаблона;
- 2) редактирование шаблона (см. [дизайнер отчетов](#)⁸⁴);
- 3) проверка и сохранение шаблона или восстановление исходной копии.

Для **сохранения копии шаблона** необходимо:

- выбрать нужный отчет и перейти на вкладку Шаблоны;

- вызвать дизайнер отчетов;
- сохранить шаблон с помощью функции *Отчет задания* (вызывается нажатием элемента  в левом верхнем углу отчета) > *Экспорт разметки в файл*.

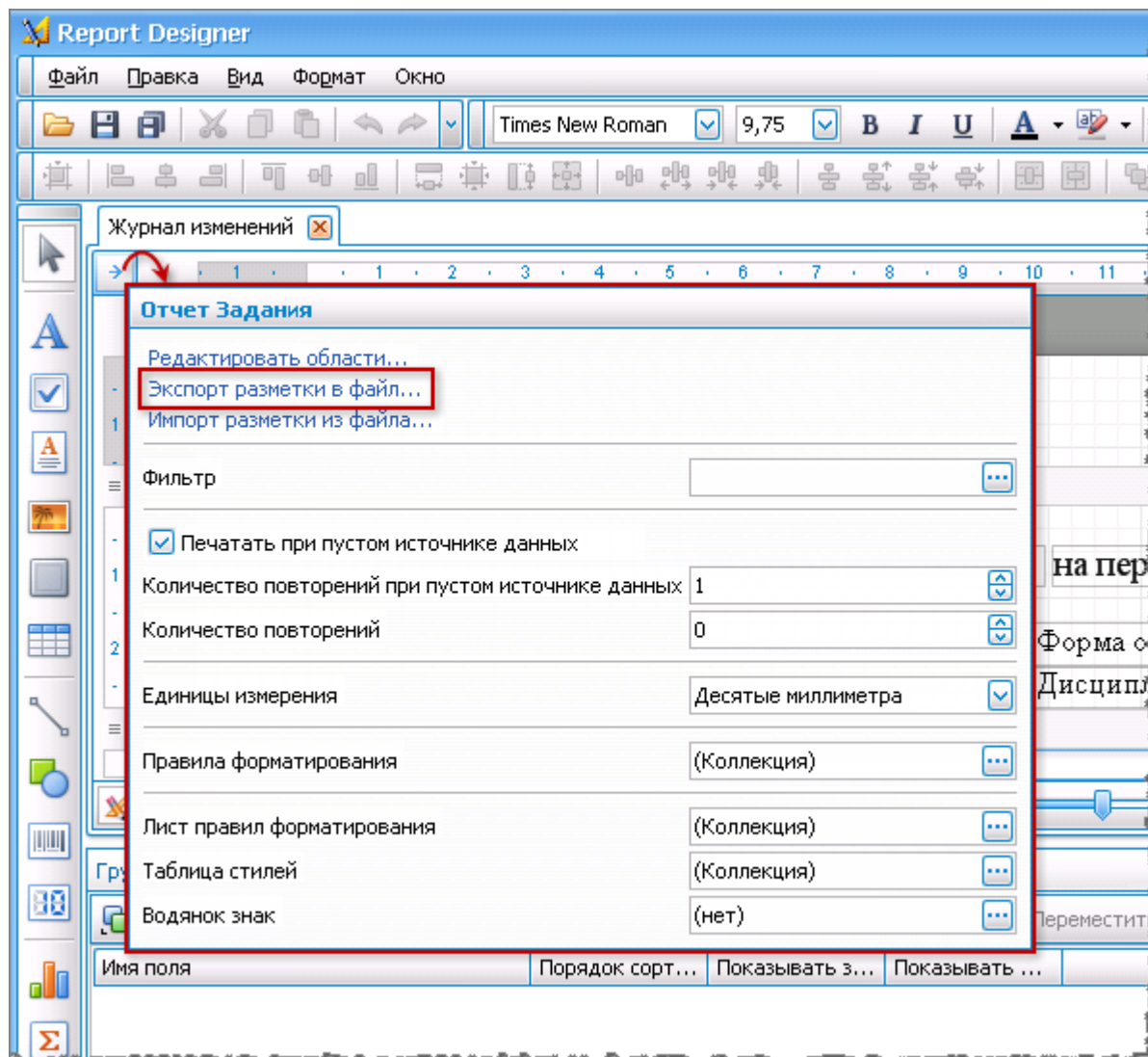


Рис. 45. Сохранение шаблона

 Копию шаблона, поставляемого с решением, рекомендуется сохранить для возможности восстановления отчета в исходном виде.

Порядок создания новых отчетных форм:

- 1) выбор наиболее подходящего в качестве основы существующего шаблона и его сохранение (см. выше);
- 2) создание нового шаблона;
- 3) импорт в шаблон сохраненной копии основы;
- 4) редактирование и сохранение шаблона.

Для **создания нового шаблона** необходимо:

- выбрать нужный отчет и перейти на вкладку *Шаблоны*;
- выполнить функцию  *Создать* (**Ctrl+N**) и заполнить в открывшемся мастере отчетов параметры *Название отчета* (не должно совпадать с уже

существующими) и **Тип данных** (должен совпадать с типом отчета, в котором создается шаблон, например, если был открыт отчет *Типовая неделя для групп*, то такое же значение должно быть указано и в качестве типа данных);

- нажать кнопку [Завершить].

Для **импорта шаблона** из файла следует воспользоваться функцией дизайнера отчетов *Отчет задания > Импорт разметки из файла*.

После загрузки шаблона следует переименовать название отчета, а также проверить, чтобы тип данных соответствовал отчету, в котором был создан или открыт для редактирования шаблон.

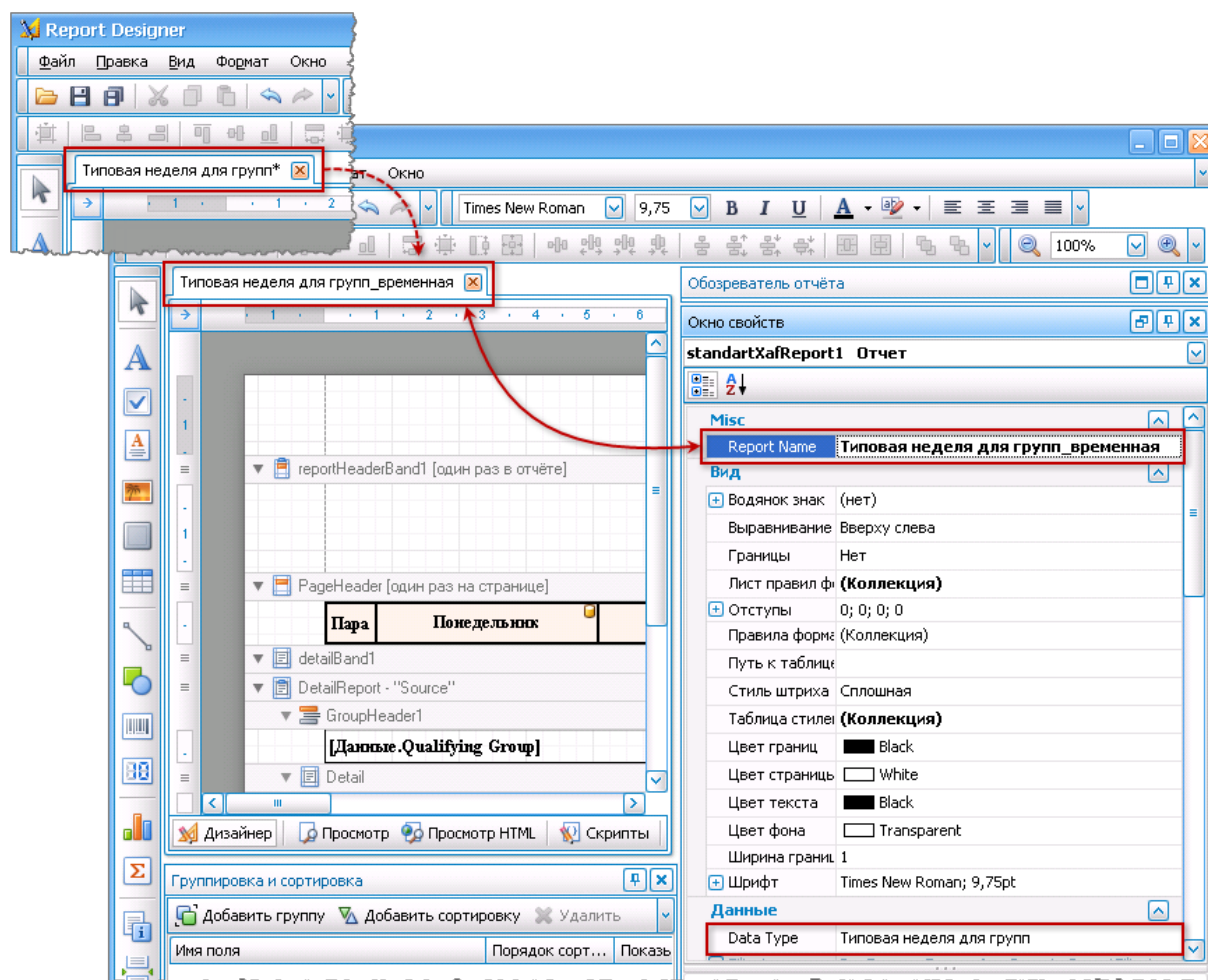


Рис. 46. Корректировка наименования отчета

7.2. Дизайнер отчетов

Встроенный дизайнер отчетов представляет собой типовой инструментарий для этого класса строителей. Назначение большинства функций понятно из их наименования. Функциональность дизайнера весьма обширна и, несмотря на интуитивно понятный интерфейс, для работы с ним необходимы определенные знания, навыки, аккуратность и опыт. В подразделах "[Структура шаблона](#)"⁸⁵ и "[Основные элементы управления](#)"⁸⁵ изложены только некоторые основные моменты по созданию (редактированию) шаблонов отчетов.

7.2.1. Структура шаблона

Новый шаблон содержит три основные области-полосы (Header, Detail и Footer), которые предоставляют пространство для размещения на них элементов управления и определяют параметры их отображения, порядок отрисовки и сколько раз они предоставляются (в случае с привязкой отчета). Некоторые полосы могут отображаться с подсказкой. Например, для областей Header и Footer отображаются подсказки "один раз в отчете" и "один раз на странице" соответственно.



Рис. 47. Исходный вид нового шаблона

Кроме указанных выше областей можно добавлять/удалять и другие области, приведенные в таблице (по ссылкам можно перейти на англоязычный сайт с более подробным описанием).

Область	Описание
TopMarginBand	верхнее поле
ReportHeaderBand	отображение вводных сведений, например, титульной страницы для отчета
PageHeaderBand	отображение номеров страниц или заголовка таблицы
GroupHeaderBand	определение критериев группировки и отображение информации в начале группы записей (для получения дополнительной информации используется Grouping Data)
DetailBand	основная область, в которой отображаются данные-записи из связанного источника данных
DetailReportBand	вложенный отчет
GroupFooterBand	сводная информация для группы или подвал группы (нижний колонтитул группы)
ReportFooterBand	отображение какой-либо завершающей информации по всему отчету
PageFooterBand	отображение завершающей информации страницы отчета, например, номера страницы или нижнего колонтитула страницы
BottomMarginBand	нижнее поле

7.2.2. Основные элементы управления

Наиболее часто используемые элементы управления в шаблоне и основные принципы их использования представлены в таблице ниже (по ссылкам можно перейти на англоязычный сайт с описанием этих элементов).

Иконка	Наименование	Иконка	Наименование
	Текст		Цифровая подпись
	Флажок		График
	Панель		Сводная таблица
	Сложный текст		Информация
	Таблица		Разрыв страницы
	Линия		Кросс-линия
	Фигура		Кросс-рамка
	Картинка		Вложенный отчет
	Штрих-код		

Для **добавления элементов** в шаблон необходимо:

- выбрать нужный элемент на панели инструментов;
- кликнуть левой кнопкой мыши по области, на которую нужно добавить элемент.

Для **установления связи** элемента с данными источника данных отчета необходимо:

- добавить элемент и выделить его;
- в верхнем правом углу элемента нажать на стрелочку  и в выпадающем списке выбрать данные, которые необходимо отобразить в элементе.

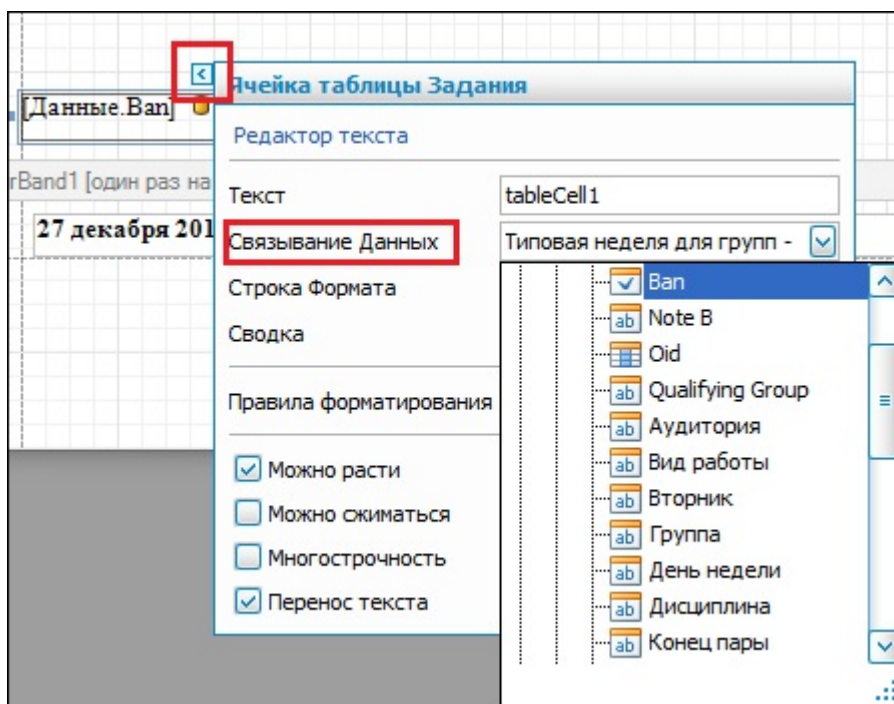


Рис. 48. Связывание данных

Редактирование элементов осуществляется через настроечные параметры, правила форматирования и т. п. в окне свойств.

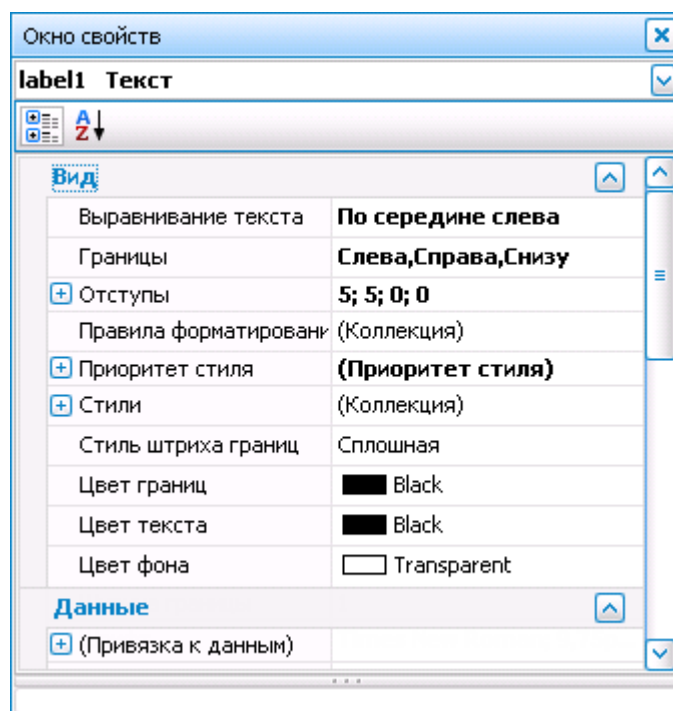


Рис. 49. Окно свойств

Правила форматирования:

- выбрать элемент и нажать на стрелочку  в верхнем правом углу элемента;
- перейти к коллекции **Правила форматирования**;
- если имеются **Доступные правила** и они удовлетворяют требованиям, необходимо выделить правило и переместить его в **Применяемые правила**; в противном случае следует нажать на кнопку [Редактировать] и создать/отредактировать правило, задав условие и параметры форматирования (т. е. как будут отображаться данные в случае выполнения условия).

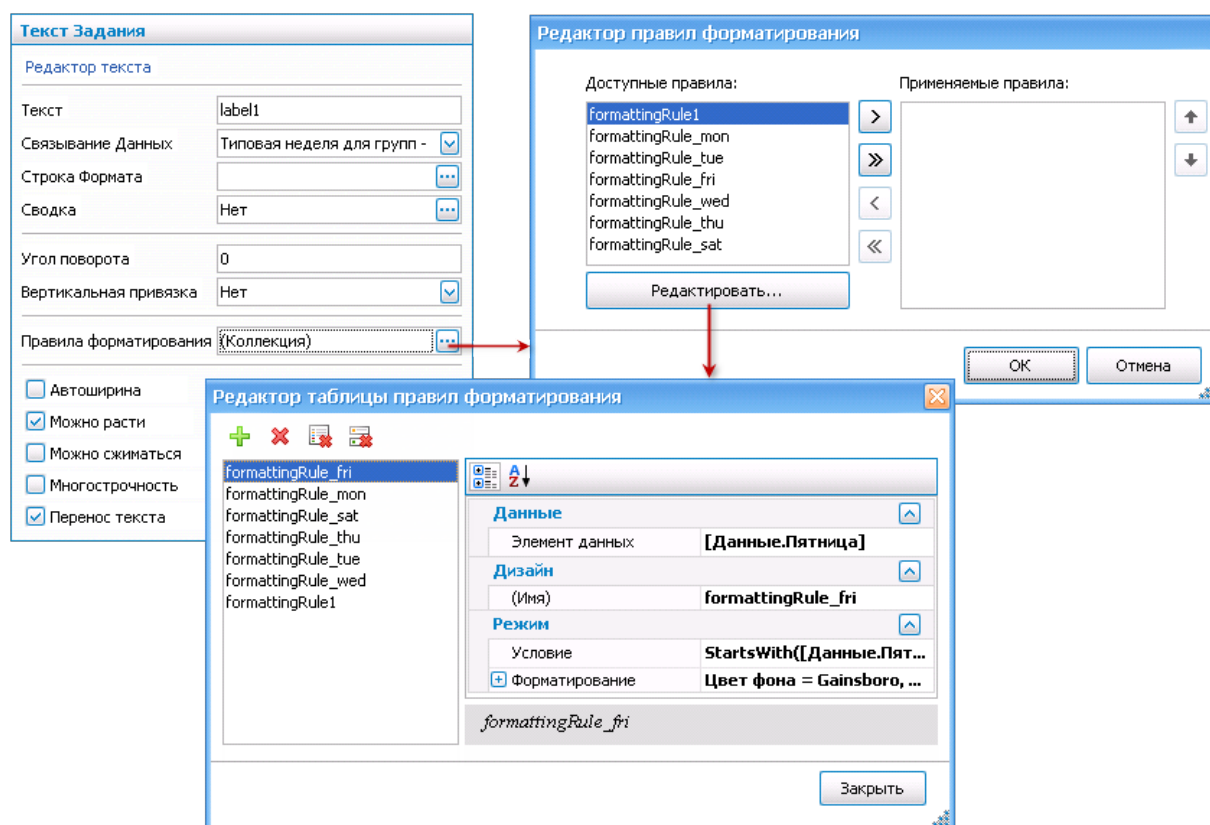


Рис. 50. Правила форматирования

8. Технологические операции

8.1. Очистка БД

Возможны следующие способы удаления данных из объектов системы:

- удаление одиночной записи — из списочных или карточных форм;
- удаление группы записей — при выделении нескольких записей в списочных формах или при выделении занятий в конструкторе;
- удаление группы записей-занятий — при использовании функций специального удаления для расписания;
- удаление всех записей объекта — при использовании функции очистки данных.

Функция *Очистка БД* используется, как правило, при необходимости полной или частичной перезагрузки данных (например, при переходе к новому учебному году).

При вызове функции открывается окно, в котором представлен перечень всех возможных объектов для очистки, сгруппированных аналогично меню системы. В поле **Выбор** следует отметить необходимые объекты. При выборе/отмене выбора группы объектов помечаются все подчиненные объекты.

Очистка БД осуществляется при выборе функции *Выполнить* на панели инструментов. При этом следует внимательно проанализировать список выделенных объектов для удаления — он автоматически может быть расширен объектами, связанными с выделенными вручную.

Операция выполняется с использованием механизма длительных операций и сопровождается визуализацией хода ее выполнения. Дождитесь, пока поле **Статус** не примет значение *Завершен*, а затем просмотрите в нижней части окна протокол на предмет наличия ошибок.

8.2. Экспорт данных

С помощью этой функции осуществляется пообъектный экспорт данных из системы *Галактика Расписание учебных занятий* в xml-файлы. Операция может быть полезна, например, при переносе данных из одной БД в другую, при подготовке данных для нового учебного года и т. п.

Перечень экспортируемых объектов фиксирован и отображается при запуске функции.

Для экспорта следует выделить необходимые объекты, вызвать функцию  **Экспорт** (в локальном меню или на панели инструментов) и указать каталог экспорта.

Маппирование (сопоставление) полей выполняется на основе файла corTable.xml. При установке системы он размещается в каталоге приложения.

В приведенном ниже примере для объекта Auditorium в параметре **key** указывается наименование, под которым из БД будет выгружаться поле с именем, указанным в параметре **value**. Таким образом, изменять в файле настроек можно только значения для параметра **key**.

```

<Object name="Auditorium" type="Auditorium">
  <prop key="Number" value="Number"/>
  <prop key="Name" value="Name"/>
  <prop key="Abbr" value="Abbr"/>
  <prop key="ID_Faculty" value="Faculty"/>
  <prop key="ID_Chair" value="Chair"/>
  <prop key="ID_Building" value="Building"/>
  <prop key="Amount" value="Amount"/>
  <prop key="ID_TypeOfAuditorium" value="TypeOfAuditorium"/>
  <prop key="Square" value="Square"/>
  <prop key="Equipment" value="Equipment"/>
  <prop key="TableType" value="TableType"/>
  <prop key="Darken" value="Darken"/>
  <prop key="Conditioner" value="Conditioner"/>
  <prop key="Height" value="Height"/>
  <prop key="Length" value="Length"/>
  <prop key="Width" value="Width"/>
  <prop key="ComputerEquipment" value="ComputerEquipment"/>
  <prop key="MediaFuture" value="MediaFuture"/>
  <prop key="ComputerAmount" value="ComputerAmount"/>
  <prop key="MonitorDiagonal" value="MonitorDiagonal"/>
  <prop key="Overlay" value="Overlay"/>
  <prop key="Note" value="Note"/>
</Object>

```

Рис. 51. Пример маппирования полей для импорта/экспорта

8.3. Корректировка данных

Функция используется в частных случаях по указанию разработчиков для проверки и точечной корректировки данных при обнаружении в БД пользователей некоторых недочетов.

8.4. Переход к новому учебному году

По окончании учебного года часть введенных в систему данных становится неактуальной (нагрузка, расписания и проч.). Чтобы старые данные не мешали работе, планирование учебного процесса на новый учебный год рекомендуется осуществлять в новой БД, а старую — "законсервировать".

 Чтобы иметь возможность просмотра старой БД, следует сохранить соответствующую версию системы, более не обновляя ее. Для работы с новой БД нужно еще раз выполнить установку системы и обновлять ее по мере выхода новых версий. Не забудьте также указать в конфигурационном файле путь на новую БД.

Возможно несколько вариантов создания новой БД.

Вариант 1:

- Создается пустая база данных. Для этого в конфигурационном файле указывается новое имя БД (например, к прежнему названию можно добавить номер нового учебного года). После запуска системы будет создана новая БД с предустановленными значениями в некоторых справочниках.
- Осуществляется [импорт данных](#)^[69] из внешних источников (в том числе с использованием шаблона ForImport.xlsm).

Вариант 2:

- Создается копия старой БД, переименовывается и в конфигурационном файле указывается путь на эту базу.
- БД очищается от ненужной информации следующими способами:
 - с помощью функции *Сервис* > [Очистка БД](#)⁸⁹ (отмеченные объекты удаляются целиком);
 - с помощью стандартной функции удаления записей (**Ctrl+D**);
 - с помощью функции *Специальное удаление*, используемой для удаления расписаний.
- В БД вносятся новые данные.

Вариант 3:

- Из старой БД осуществляется экспорт необходимой информации с помощью стандартных функций экспорта данных.
- Полученные данные вносятся в шаблон ForImport.xlsm (при необходимости корректируются), выгружаются в xml-файлы.
- Создается пустая база данных (аналогично варианту 1).
- Осуществляется импорт данных в новую БД из xml-файлов.
- БД дополняется необходимыми данными.

9. Работа с редактором модели приложения

9.1. Назначение и состав модели

Система *Галактика Расписание учебных занятий* разработана на основе платформы XAF (eXpressApp Framework). Визуальный пользовательский интерфейс, включая его содержание, параметры, поведение и т. п., хранится в специальном хранилище данных — модели приложения (Application Model).

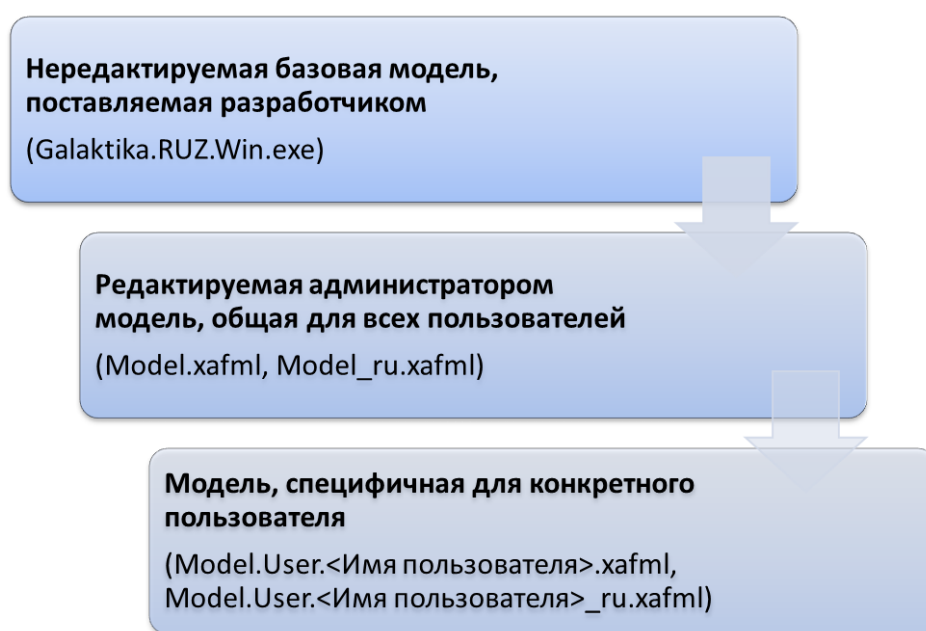


Рис. 52. Схема модели приложения

Модель состоит из следующих уровней:

- 1) **Базовая модель приложения, поставляемая разработчиком.** Эта модель создается разработчиками, жестко прошита в коде, ее нельзя изменить без перекомпиляции приложения.
- 2) **Редактируемая модель приложения, общая для всех пользователей.** Редактируется, как правило, только администратором системы. Для этого используется утилита DevExpress.ExpressApp.ModelEditor.vXX.X.exe (при установке устанавливается в каталог приложения, в качестве входных параметров для утилиты необходимо передать конфигурационный файл). Изменения, сделанные на этом уровне, применяются для всех пользователей.
- 3) **Модель, специфичная для конкретного пользователя.** При запуске приложения для каждого пользователя создается своя модель. Расположение файлов модели этого уровня регулируется ключами [конфигурационного файла](#)⁴². В модели сохраняются настройки форм, устанавливаемые пользователем при работе с приложением, положение окон, содержание и т. п. Для редактирования других параметров модели предназначена специальная функция *Инструменты > Редактировать модель*. Доступ пользователей к этой функции устанавливается в настройках [роли](#)⁴⁵ с помощью параметра *Разрешить редактировать модель*. Пользовательская модель может быть удалена (будут удалены произведенные пользователем индивидуальные настройки). При запуске системы

пользовательская модель будет создана заново. Для отмены отдельных изменений, внесенных в модель через редактор модели, необходимо воспользоваться соответствующей [функцией редактора](#)⁹⁵.

9.2. Структура модели

Информация в модели приложения организована в виде дерева. Основные узлы модели:

- *ActionDesign* — позволяет настроить поведение действий (*Actions*) и контроллеров (*Controllers*), например доступность при определенных условиях.
- *BOModel* — позволяет настроить бизнес-объекты, например добавить новый бизнес-объект или добавить реквизит в существующий бизнес-объект.
- *CreatableItems* — позволяет настроить поведение для действия *Новый*.
- *DetailViewItems* — позволяет определить элементы управления, используемые на форме редактирования.
- *ImageSources* — используется приложением при загрузке изображений. Здесь можно указать источники загрузки изображений и определить порядок загрузки.
- *Localization* — позволяет локализовать визуальные элементы приложения, такие как сообщения, исключения, надписи на кнопках и т. д.
- *NavigationItem* — позволяет определить двухуровневую структуру основного меню. В приложении для настройки меню сейчас используется узел *Xafari*.
- *Options* — содержит общие настройки приложения.
- *Validation* — содержит контекст (*Contexts*) и правила (*Rules*), используемые в приложении. Например, можно настроить уникальность значения кода в справочнике.
- *ViewItems* — содержит визуальные элементы, доступные на детальных формах.
- *Views* — содержит настройку расположения визуальных элементов на формах.
- *Xafari* — содержит настройку основного меню приложения (так называемого списка АРМов) и другие настройки платформы Xafari.

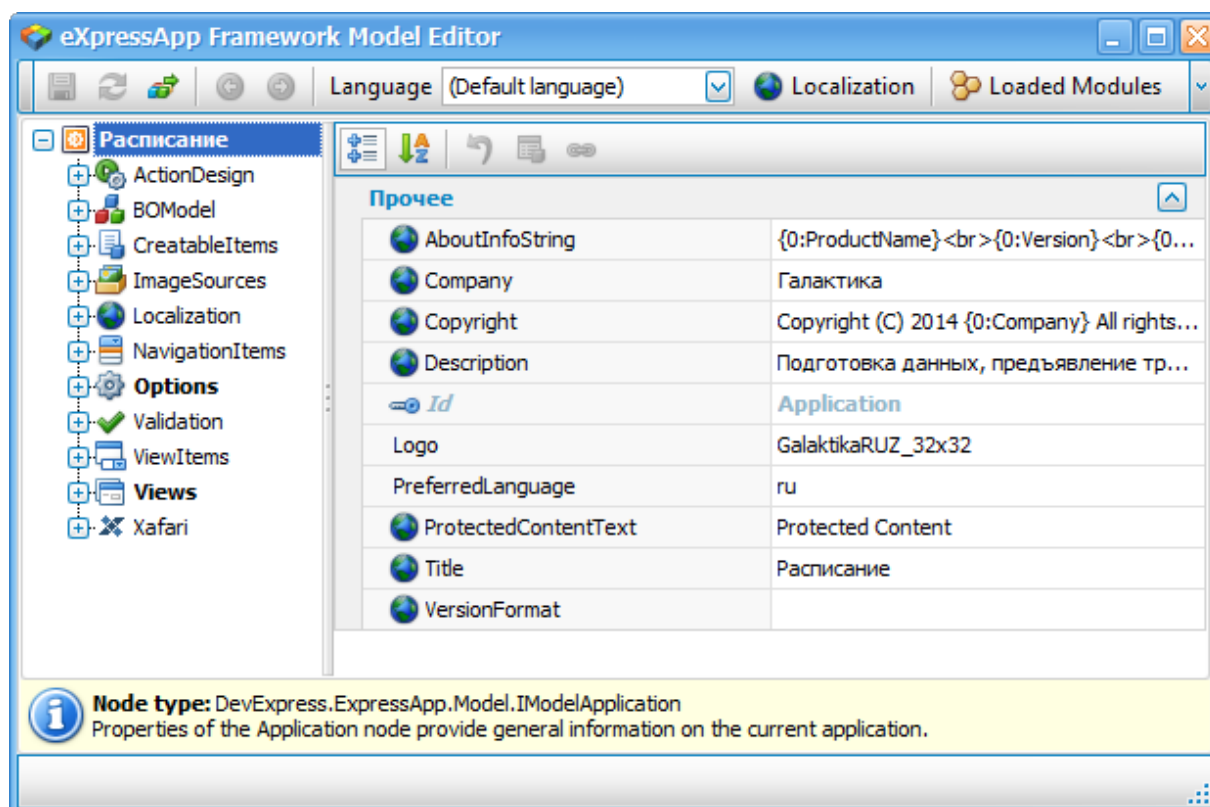


Рис. 53. Структура модели приложения

9.3. Правила работы с моделью

Каждый узел, также как и любой другой элемент, имеет определенный набор свойств, через изменение которых достигается требуемый результат. Различные элементы модели имеют различный набор свойств. При позиционировании на элементе в структуре модели или на свойстве элемента в нижней части формы отображается подсказка.

Свойства элементов сгруппированы по группам, определяющим внешний вид, данные, макет, поведение, прочее, формат и др.

Панель инструментов окна редактора модели содержит ряд стандартных функций: *Save*, *Reload*, *Back*, *Forward*, *Search* и др. Если на панели инструментов в качестве языка (*Language*) указано значение (*Default language*), то изменения сохраняются в файле *Model...xafml*. Если выбран язык локализации *ru*, то изменения сохраняются в файле *Model..._ru.xafml*.

Для изменения локализации наименований элементов модели может использоваться специальный интерфейс, вызываемый по функции *Localization*. Здесь доступно прямое редактирование локализованных значений в поле ***Translated Value***, а также ряд полезных функций поиска (**Ctrl+F**), фильтрации (конструктор фильтра), сортировки и т. п.

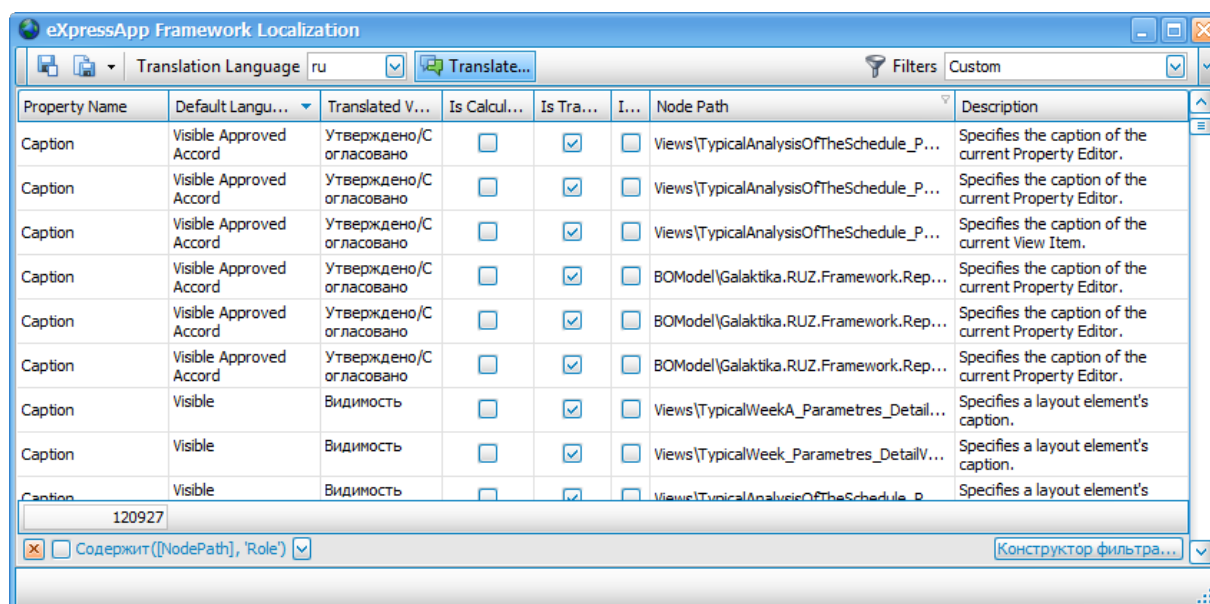


Рис. 54. Локализация элементов модели

Дополнительные сведения по работе с редактором модели можно найти в документации, расположенной по адресу: <https://documentation.devexpress.com/#Xaf/CustomDocument2582>.

9.4. Восстановление первоначального вида

Вносить изменения в модель приложения через редактор модели без особой необходимости не рекомендуется. Все изменения должны быть продуманы и пользователь должен представлять их последствия и дальнейшую логику поведения приложения в новых условиях.

При внесении изменений в какой-либо элемент модели его наименование выделяется жирным шрифтом. Для отмены внесенных изменений необходимо воспользоваться функцией локального меню *Reset Differences*.

9.5. Типовые пользовательские изменения

Для визуализации данных системы в большинстве случаев используются следующие формы или их сочетания:

- список;
- форма редактирования;
- список для выбора;
- наборная форма.

 Подробное описание различных видов форм приведено в руководстве пользователя.

Основные настройки форм могут быть выполнены пользователем в режиме runtime. Эти настройки будут сохранены в его собственной модели и не повлияют на изменение других пользовательских моделей. Порядок действий при этом изложен в руководстве пользователя. Аналогичные и дополнительные действия можно выполнить через редактирование модели.

Для того чтобы изменения коснулись всех пользователей, необходимо отредактировать общую модель. Это делается только через редактор модели. Далее каждый пользователь по-прежнему может вносить свои специфические настройки или редактировать свою модель.

Для идентификации наименования формы, которую понадобилось перенастроить, целесообразно в конфигурационном файле включить диагностику (`<add key="EnableDiagnosticActions" value="True" />`) и после запуска приложения на искомой форме вызвать функцию *Информация о представлении*.

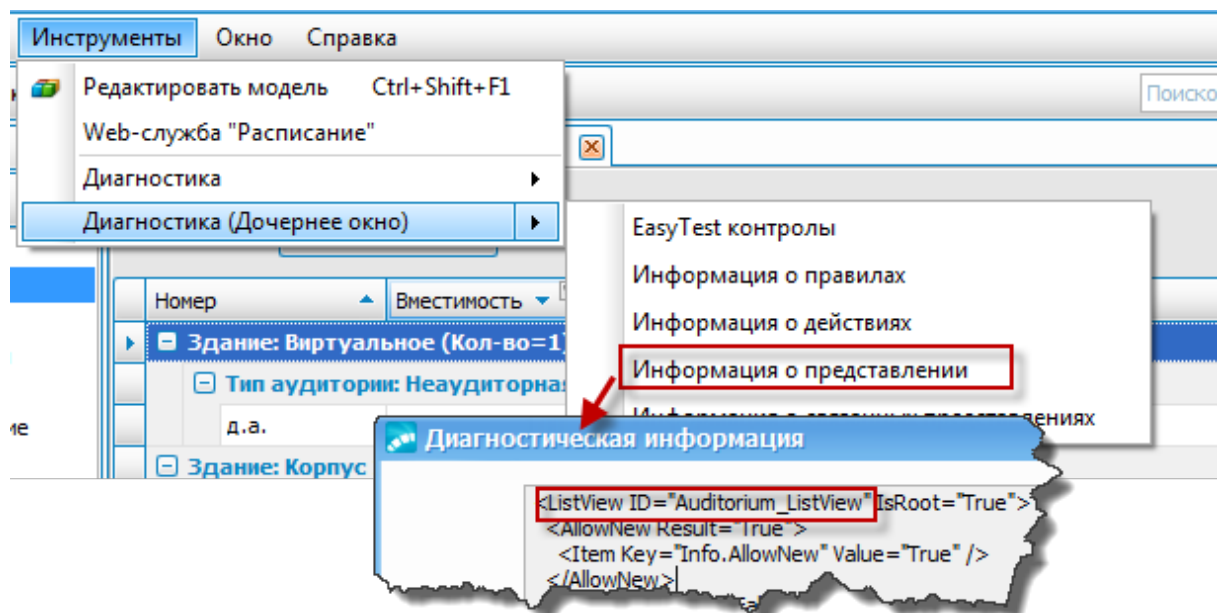


Рис. 55. Диагностическая информация

Для идентификации наименования карточных форм и вложенных списочных форм используется аналогичная функция *Диагностика > Информация о преставлении*.

9.5.1. Настройка списочных форм

Основные списочные формы приложения размещены в узле *Views > Galaktika.RUZ.Framework.Data*. Они имеют постфикс *_ListView*.

Основные параметры настройки поведения формы:

- *IsGroupPanelVisible* — флаг отображения области группировки;
- *IsFooterVisible* — флаг отображения footer-области;
- *AutoExpandAllGroups* — флаг автоматического раскрытия всех групп списка.

Здание ▼ Тип аудитории ▲

Область группировки

Номер	Вместимость	Факультет/обр.прогр.	Аббревиатура	Доп. накл.
Здание: Виртуальное (Кол-во=1)				
Тип аудитории: Неаудиторная (Кол-во=1)				
д.а.	1 000		д.а.	☒
Здание: Корпус 1 (Кол-во=8)				
Тип аудитории: Лекционная (Кол-во=4)				
K1-104	200		ауд. 104	☐
K1-102	30	ФАКУЛЬТЕТ ФИЗИКО-МАТЕМАТИЧЕСКИЙ	ауд. 102	☒
K1-103	30	ФАКУЛЬТЕТ ФИЗИКО-МАТЕМАТИЧЕСКИЙ	ауд. 103	☐
K1-101	20	ФАКУЛЬТЕТ ФИЗИКО-МАТЕМАТИЧЕСКИЙ	ауд. 101	☐
Тип аудитории: Семинарская (Кол-во=4)				
K1-203	30	ФАКУЛЬТЕТ ФИЗИКО-МАТЕМАТИЧЕСКИЙ	K1-203	☐
66	Сумма=3 432	Footer-область		

Рис. 56. Элементы списочной формы

Auditorium_ListView

- Bands
- ChartSettings
- Columns
 - Number
 - Building
 - Amount
 - TypeOfAuditorium
 - Abbr
 - Faculty
 - Overlay
 - Chair
 - ComputerAmount
 - ComputerEquipment
 - Conditioner

Поведение

MinWidth	-1
AllowDelete	True
AllowDragAndDrop	False
AllowEdit	False
AllowNew	True
AutoExpandAllGroups	True
AutoRefresh	False
AutoRefreshTimeout	10
FreezeColumnIndices	False
GroupSummary	
IsFooterVisible	True
IsGroupPanelVisible	True
IsNeedSaveState	True

Рис. 57. Настройка поведения списочной формы

Перечень и последовательность колонок списка определяется конструкцией *Columns*.

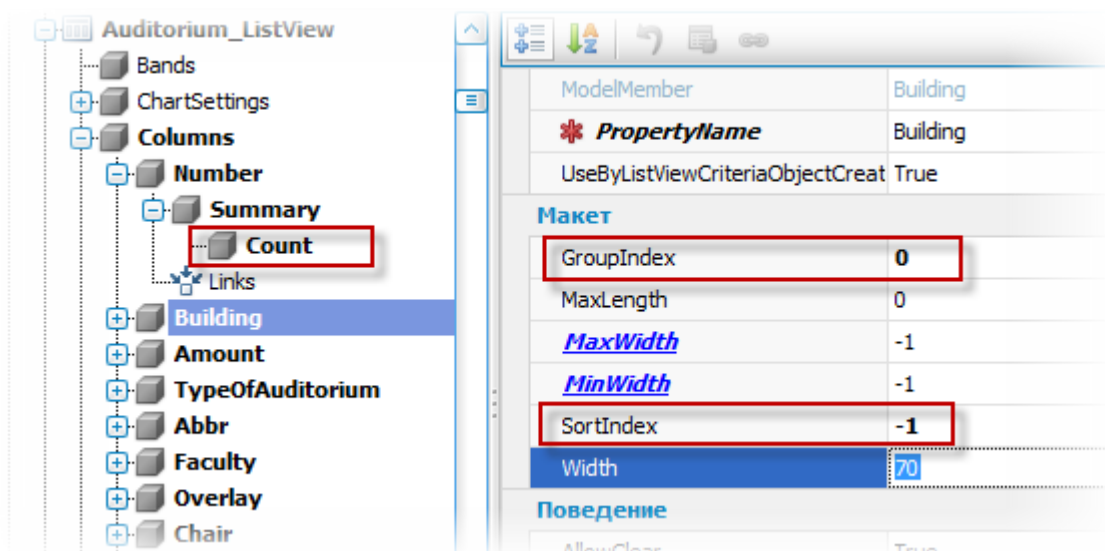


Рис. 58. Настройка содержания списка

Для изменения последовательности удобно использовать комбинации клавиш **Alt+Up/Down**.

Для скрытия поля необходимо установить для него свойство **Index** = -1.

Если по полю должна выполняться группировка, то для свойства **GroupIndex** необходимо установить отличное от -1 значение. Если группировка выполняется по нескольким полям, то последовательность группировки определяется значениями этого свойства.

Сортировка и ее последовательность определяются свойством **SortIndex**.

9.5.2. Настройка карточных форм

Основные карточные формы приложения размещены в узле *Views > Galaktika.RUZ.Framework.Data*. Они имеют постфикс *_DetailView*.

Для настройки расположения визуальных элементов на форме выберите узел *Layout*. Вместо списка атрибутов в правой панели для этого узла отображается дизайнер формы.

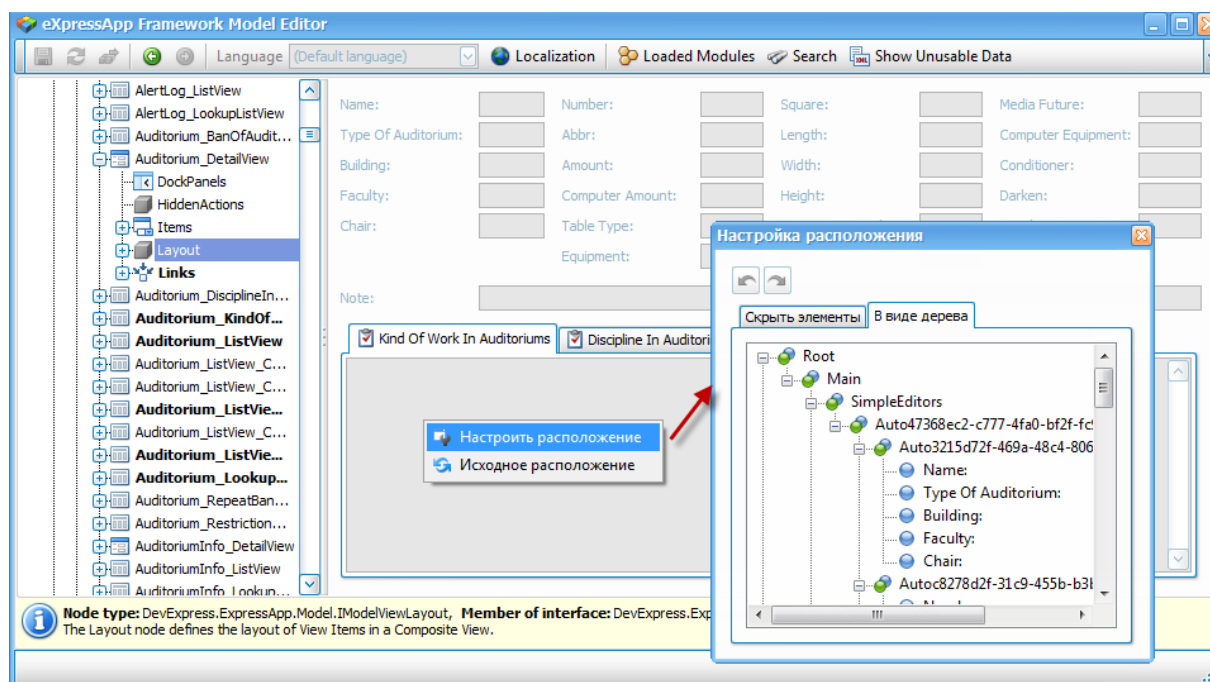


Рис. 59. Настройка карточной формы

Чтобы изменить расположение визуальных элементов необходимо вызвать из контекстного меню функцию *Настроить расположение*. Далее можно использовать технологию Drag&Drop для перемещения визуальных элементов. Чтобы убрать визуальный элемент, необходимо перетянуть его на форму настройки. В дальнейшем этот визуальный элемент снова можно будет вернуть на форму таким же способом.

Для дополнительного изучения возможностей настройки визуального представления карточной формы можно обратиться к документации, расположенной по адресу: <https://documentation.devexpress.com/#WindowsForms/CustomDocument2307>.

9.5.3. Настройка форм для выбора

Основные формы приложения для выбора размещены в узле *Views > Galaktika.RUZ.Framework.Data*. Они имеют постфикс *_LookupListView*.

Форма для выбора может быть представлена в виде обычного списка или списка с возможностью поиска. Вариант отображения формы выбора и ее поведение зависят от свойства **LookupEditorMode**, устанавливаемого для этого поля в **карточной** форме:

- *Auto* — вариант отображения списка (с возможностью поиска или без) определяется автоматически и зависит от количества записей в объекте выбора и настроек модели. Если в объекте для выбора количество записей не более указанного для свойства **LookupSmallCollectionItemCount** (узел *Options*), то отображается обычный список, иначе — список с возможностью поиска.
- *AllItems* — отображается полный перечень значений для выбора, панель поиска отсутствует. При большом объеме справочника это будет замедлять отображение данных.
- *Search* — отображается пустой список и доступна панель поиска. Данные для выбора отображаются только после ввода искомого фрагмента текста в поисковой строке и нажатии кнопки [Вперед] (выводятся только те данные справочника, в которых присутствует искомый фрагмент). Если в строке поиска ничего не введено и нажата кнопка [Вперед], то отображаются все данные.

- *AllItemsWithSearch* — отображается полный перечень значений для выбора, при этом имеется возможность поиска необходимых записей.

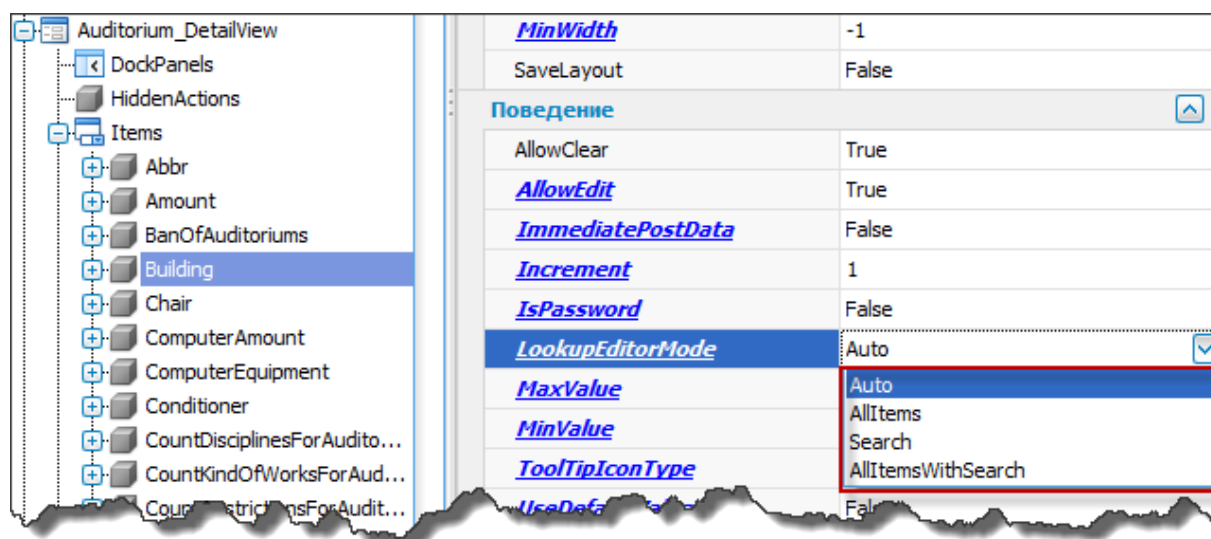


Рис. 60. Режимы отображения форм для выбора

Настройка свойств содержания, внешнего вида, поведения и т. п. формы для выбора аналогична настройке [списочной](#)⁹⁶ формы.

10. Возможные проблемы и способы их устранения

Особенности запуска в Windows 7, Windows 10

При запуске приложения под ОС Windows 7 или Windows 10 с включенной проверкой учетных записей могут возникать ошибки, если приложение запускается не от имени администратора.

Запуск приложения от имени администратора возможен следующими способами:

- 1) в свойствах ярлыка приложения установить признак *Запуск от имени администратора*.
- 2) на ярлыке приложения щелкнуть правой кнопки мышки и выбрать функцию *Запуск от имени администратора*.

Нет доступа к серверу лицензий

Сервис лицензий недоступен.

Необходимо проверить наличие установленного сервиса лицензий, его настройку, отсутствие блокировки брандмауэром Windows и выполнить его запуск.

Превышено количество одновременных подключений

При попытке очередным пользователем использования приложения возникло нарушение количества конкурентных подключений, указанного в файле лицензий.

Необходимо дождаться завершения работы какого-либо из уже работающих пользователей.

Статистика подключений пользователей, в т.ч. при неудачной попытке подключения, отображается в [Журнале логирования](#)^[57].

Ошибка проверки лицензии

При выполнении очередной проверки установленных лицензионным файлом правил лицензирования обнаружено нарушение корректности лицензионного файла или недоступность сервиса лицензий.

Необходимо проверить идентичность лицензионного файла полученному при поставке, настройки в конфигурационном файле, активность сервиса лицензий. При необходимости устранить причину.

Подробнее о возможных проблемах и способах их устранения, связанных с лицензированием, см. соответствующую инструкцию.

Приложение не запускается, и нет сообщений об ошибке

Наиболее частой причиной может быть ошибка в синтаксисе конфигурационного файла Galaktika.RUZ.Win.exe.config, т. е. некорректный XML-файл. Следует внимательно просмотреть config-файл на предмет его корректности.

БД не существует или ее версия старше. Выполнить создание или обновление?

Возникает при запуске приложения в следующих случаях:

- при первом запуске обновленной версии приложения;

- если не удалось подключиться к БД.

При выборе положительного варианта в диалоге в первом случае выполняется программное обновление структуры и содержания БД, заложенное в обновленную версию и приложение запускается по завершении этой операции.

 До запуска обновленной версии приложения следует сделать резервную копию БД. Если это не было сделано, то при получении данного сообщения следует отказаться от продолжения запуска и сделать резервную копию.

Во втором случае при положительном выборе в диалоге будет получено нижеуказанное сообщение.

Не удалось подключиться к базе данных. Пожалуйста, обратитесь к системному администратору

Возможные причины:

- при запуске приложения не установлено соединение с БД;
- в процессе работы с приложением возникла ошибка на уровне СУБД.

В первом случае необходимо:

- убедиться в корректности синтаксиса и содержания конфигурационного файла Galaktika.RUZ.Win.exe.config (см. [настройки конфигурационного файла](#)^[39]);
- проверить готовность "окружения" к соединению с БД: активность сервера БД и его процесса-браузера; корректность настройки сервера БД, в частности, указание статического или динамического портов; наличие в брэндмауэре исключений для соединений по статическому порту или по программе-серверу для динамических.

Во втором случае необходимо открыть log-файл приложения и посмотреть детализацию причины ошибки. При необходимости выполнить действия, указанные ниже в пункте "Невозможно локализовать ошибку".

При запуске приложения на БД Oracle ошибка ORA-12154

При запуске приложения открывается окно с сообщением: "ORA-12154: TNS: невозможно разрешить заданный идентификатор соединения".

Не удастся найти идентификатор соединения с БД в файле tnsnames.ora (Oracle). Следует корректно настроить соединение с БД на клиенте.

Иногда при настроенном соединении клиента Oracle на БД ошибка все равно возникает. В этом случае в строке соединения с БД в config-файле можно напрямую прописать соединение с БД Oracle, например:

```
<add name="ConnectionString"
connectionString="XpoProvider=Oracle;Data Source=(DESCRIPTION =
  (ADDRESS_LIST =
    (ADDRESS = (PROTOCOL = TCP) (HOST = ORUAP-SERVER) (PORT = 1521))
  )
  (CONNECT_DATA =
    (SERVICE_NAME = ORCL_RUZ)
  )
);User ID=RUZ;Password=1" />
```

 Правила описания соединения с БД Oracle смотрите в документации по Oracle.

При запуске приложения на БД Oracle ошибка `BadImageFormatException`

При запуске приложения открывается окно с сообщением: *"Попытка загрузить клиентские библиотеки Oracle привела к созданию `BadImageFormatException`. Такое поведение наблюдается в 64-разрядном режиме, если установлены 32-разрядный клиентские компоненты Oracle"*.

Разрядность клиента Oracle не соответствует разрядности ОС. Если ОС 32-разрядная, установите клиент Oracle 32-разрядный. Если ОС 64-разрядная, установите клиент Oracle 64-разрядный.

Невозможно локализовать ошибку

Если возникает ошибка, с которой самостоятельно справиться не получается, то следует обратиться в службу техподдержки. При этом необходимо:

- описать последовательность действий, приведших к проблемной ситуации;
- приложить снимки экранов;
- приложить файл `eXpressAppFramework.log`;
- сформулировать вопрос.

11. Приложения

11.1. Контракты web-службы

Табл. 8. Контракты web-службы

№	Контракт	Наименование контракта
1	groups ^[105]	Предоставление списка групп
2	lecturers ^[107]	Предоставление списка преподавателей
3	auditoriums ^[107]	Предоставление списка аудиторий
4	typeofauditoriums ^[108]	Предоставление списка типов аудиторий
5	kindofworks ^[108]	Предоставление списка видов работ
6	buildings ^[109]	Предоставление списка зданий
7	faculties ^[109]	Предоставление списка факультетов
8	chairs ^[110]	Предоставление списка кафедр
9	streams ^[110]	Предоставление списка потоков
10	subgroups ^[111]	Предоставление списка подгрупп
11	messages ^[111]	Предоставление полного текста PUSH-сообщения по его ID
12	notifications ^[112]	Предоставление списка уведомлений пользователей РУЗ об изменении статусов заявок на бронирование/дооснащение аудиторий
13	banofauditoriums ^[112]	Получение списка заявок на бронирование/дооснащение аудиторий (интервал не более 180 дней)
14	setbanofauditorium ^[113]	Размещение заявки на бронирование/дооснащение аудитории от имени пользователя
15	token	Исключен
16	getavailableauditoriums ^[116]	Подбор свободной аудитории по заданным параметрам
17	staffofgroup ^[117]	Предоставление списка состава группы
18	staffofstream ^[117]	Предоставление списка состава потоков
19	lessons ^[117]	Предоставление списка занятий и групповых мероприятий из запретов/заявок аудиторий в заданном интервале времени по указанным входным параметрам (интервал дат не более 180 дней)
20	personlessons ^[120]	Предоставление списка персонального расписания занятий и групповых мероприятий из запретов/заявок аудиторий в заданном интервале времени по указанным входным параметрам (интервал дат не более 180 дней)
21	v2/personlessons ^[124]	Предоставление списка или количества занятий персонального расписания занятий и групповых мероприятий из запретов/заявок аудиторий в заданном

№	Контракт	Наименование контракта
		интервале времени по указанным входным параметрам (интервал дат не более 180 дней), вариант 2
22	lessonscaendar ¹²⁷	Предоставление списка персонального расписания занятий и групповых мероприятий из запретов/заявок аудиторий в заданном интервале времени по указанным входным параметрам в формате .ics (интервал дат не более 180 дней)
23	semester ¹²⁸	Предоставление Академкалендаря
24	tableoflesson ¹²⁹	Предоставление Сетки пар
25	contenttableoflesson ¹²⁹	Предоставление занятий в сетках пар
26	lecturersgroups ¹²⁹	Предоставление списка всех связей преподаватель - группа/поток/ подгруппа
27	deletebanofauditorium ¹³⁰	Удаление заявки на бронирование/дооснащение аудитории
28	students ¹³¹	Предоставление списка студентов
29	discipline ¹³¹	Предоставление списка дисциплин
30	contentofload ¹³²	Предоставление списка нагрузок
31	denormalizationofload ¹³³	Предоставление графика прохождения дисциплины
32	studentgroups ¹³⁴	Предоставление списка связей студент - группа
33	resourceLanguage ¹³⁴	Предоставление списка наименований объектов на иностранном языке
34	objectApply ¹³⁵	Предоставление списка всех типов объектов
35	languages ¹³⁵	Предоставление списка языков
36	speciality ¹³⁵	Предоставление списка специальностей
37	specialization ¹³⁶	Предоставление списка специализаций
38	posts ¹³⁶	Предоставление списка должностей
39	ContentOfScheduleLog ¹³⁶	Предоставление списка занятий, в которые были внесены изменения в заданном интервале времени
40	staffOfEduGroup ¹³⁸	Предоставление списка связей академическая группа - учебная группа
41	staffofsubgroup ¹³⁹	Предоставление списка состава подгруппы

Формат передаваемых/принимаемых дат: **yyyy.mm.dd**.

Формат передаваемого/принимаемого времени: **hh:mm**.

1. Предоставление списка групп

URL ресурса: `http://servername/ruzservice.svc/groups?facultyOid={facultyOid}&findtext={findText}&kindeducation={kindEducation}&isscheduleactive={isScheduleActive}&isschedule={isSchedule}&GID={GID}`.

HTTP-метод: GET.

Возвращает список: Да.

Формат ответа: JSON.

Входные параметры:

- Int facultyOid — определяет факультет, группы которого следует предоставить; по умолчанию используется значение 0 — предоставляет список всех групп.
- Int KindEducation (необязательный параметр) — уровень образования: 0 — Бакалавриат, 1 — Магистратура, 2 — Специалитет, 3 — Аспирантура, 4 — Дополнительное профессиональное образование, 5 — Дополнительное образование, 6 — Среднее профессиональное образование.
- IsScheduleActive (необязательный параметр) — признак необходимости проверки наличия у группы расписания в статусе *Активный* (в выходной список групп будут отобраны только такие группы); значения: true, false.
- IsSchedule (необязательный параметр) — признак необходимости проверки, есть ли для группы занятия хоть в одном расписании в статусе *Активный* или *Неутвержденный* или бронирования (в выходной список групп будут отобраны только такие группы); значения: true, false.
- Long GID (необязательный параметр) — внешний id ; если он задан, то поиск группы осуществляется по нему (Group.GID=GID).

Возвращаемые значения:

- int FormOfEducationGid — внешний id формы обучения;
- int FormOfEducationOid — id формы обучения;
- int SpecialityGid — внешний id специальности;
- int SpecialityOid — id специальности;
- int YearOfEducation — год обучения;
- int amount — фактическое количество студентов в группе;
- int chairGid — внешний id кафедры;
- int chairOid — id кафедры;
- int course — курс;
- string faculty — факультет;
- int facultyGid — внешний id факультета;
- int facultyOid — id факультета;
- string formOfEducation — форма обучения;
- char groupGUID — дополнительный идентификатор группы в формате UUID;
- int groupGid — внешний id группы;
- int groupOid — id группы;
- string groupSpeciality_uid — UID специальности группы;
- string groupSpecialization_name — наименование специализации группы;
- int groupSpecialization_oid — OID специализации группы;
- string groupSpecialization_uid — UID специализации группы;
- string groupUID — текстовый id группы;
- int kind — вид группы (0 - Академическая, 1 - Учебная);

- int kindEducation — уровень образования;
- string name — наименование группы;
- string number — номер группы;
- int plannedamount — плановое количество студентов в группе;
- string speciality — специальность.

2. Предоставление списка преподавателей

URL ресурса: `http://servername/ruzservice.svc/lecturers?chairid={value}`.

HTTP-метод: GET.

Возвращает список: Да.

Формат ответа: JSON.

Входные параметры:

Int chairOid — определяет кафедру, преподавателей которой следует предоставить, по умолчанию используется значение 0 — предоставляет список всех преподавателей.

Возвращаемые значения:

- bool availability — доступность преподавателя (1 - Да, 0 - преподаватель в данный момент не работает);
- string Chair — кафедра;
- int chairGid — внешний id кафедры;
- int chairOid — id кафедры;
- string [email](#)^[59] — электронная почта преподавателя;
- string FIO — Ф.И.О. преподавателя полностью;
- char lecturerCustomUID — дополнительный идентификатор преподавателя, отличие от person в типе данных (char);
- char lecturerGUID — дополнительный идентификатор преподавателя в формате UUID;
- int lecturerGid — внешний id преподавателя;
- int lecturerOid — id преподавателя;
- string lecturerUID — идентификатор преподавателя в виде строки, используется в случае, если id содержит символы, например в табельном номере;
- string lecturer_rank — должность преподавателя;
- int person — идентификатор преподавателя, как физического лица (персоны);
- string shortFIO — Ф.И.О. преподавателя (имя, отчество - инициалами).

3. Предоставление списка аудиторий

URL ресурса: `http://servername/ruzservice.svc/auditoriums?buildingoid={value}`.

HTTP-метод: GET.

Возвращает список: Да.

Формат ответа: JSON.

Входные параметры:

Int buildingOid — определяет здание, аудитории которого следует предоставить; по умолчанию используется значение 0 — предоставляет список всех аудиторий.

Возвращаемые значения:

- int auditoriumOid — id аудитории;
- int auditoriumGid — внешний id аудитории;
- char auditoriumGUID — дополнительный идентификатор аудитории в формате UUID;
- string auditoriumUID — текстовый идентификатор аудитории;
- string Building — название здания;
- int buildingOid — id здания;
- int buildingGid — внешний id здания;
- string TypeOfAuditorium — тип аудитории;
- int TypeOfAuditoriumOid — id типа аудитории;
- int hideInCapacity — флаг, скрывать аудиторию для отображения или не скрывать (0 - не скрывать, 1 - скрывать);
- string Name — наименование аудитории;
- string Number — номер аудитории.
- int amount — вместимость аудитории.

4. Предоставление списка типов аудиторий

URL ресурса: <http://servername/ruzservice.svc/typeofauditoriums>.

HTTP-метод: GET.

Возвращает список: Да.

Формат ответа: JSON.

Возвращаемые значения:

- int TypeOfAuditoriumOid — id типа аудитории;
- int TypeOfAuditoriumGid — внешний id типа аудитории;
- string Name — наименование;
- string Abbr — аббревиатура;
- string Code — код;
- int hideInCapacity — флаг, скрывать данный тип аудиторий для отображения или не скрывать.

5. Предоставление списка видов работ

URL ресурса: <http://servername/ruzservice.svc/kindofworks>.

HTTP-метод: GET.

Возвращает список: Да.

Формат ответа: JSON.

Возвращаемые значения:

- int kindOfWorkOid — id вида работ;
- int kindOfWorkGid — внешний id вида работ;
- string kindOfWorkUID — текстовый идентификатор вида работ;
- string Name — наименование;
- string Abbr — аббревиатура;
- string Code — код;
- int complexity — сложность вида работ;
- string Unit — единица измерения.

6. Предоставление списка зданий

URL ресурса: <http://servername/ruzservice.svc/buildings>.

HTTP-метод: GET.

Возвращает список: Да.

Формат ответа: JSON.

Возвращаемые значения:

- string Abbr — аббревиатура;
- string Address — адрес;
- int Availableforgetfreeauditorium — флаг Доступно для подбора свободной аудитории (0 - нет, 1 - да);
- int buildingGid — внешний id здания;
- int buildingOid — id здания;
- string buildingUID — текстовый идентификатор здания;
- int hidden — флаг Скрытое здание (0 - нет, 1 - да);
- double latitude — широта здания;
- double longitude — долгота здания;
- string Name — наименование;
- string note — примечание;
- int tableofflesson — сетка пар;
- int floor — этажность здания.

7. Предоставление списка факультетов

URL ресурса: <http://servername/ruzservice.svc/faculties>.

HTTP-метод: GET.

Возвращает список: Да.

Формат ответа: JSON.

Возвращаемые значения:

- int facultyOid — id факультета;

- int facultyGid — внешний id факультета;
- string facultyUID — текстовый идентификатор факультета;
- string Name — наименование;
- string Abbr — аббревиатура;
- string Institute — институт;
- string Code — код.

8. Предоставление списка кафедр

URL ресурса: `http://servername/ruzservice.svc/chairs?facultyoid={value}`.

HTTP-метод: GET.

Возвращает список: Да.

Формат ответа: JSON.

Входные параметры:

Int facultyoid — определяет факультет, кафедры которого следует предоставить; по умолчанию используется значение 0 — предоставляет список всех кафедр.

Возвращаемые значения:

- int chairOid — id кафедры;
- int chairGid — внешний id кафедры;
- string chairUID — текстовый идентификатор кафедры;
- string Name — наименование;
- string Abbr — аббревиатура;
- string Faculty — факультет;
- int facultyOid — id факультета;
- int facultyGid — внешний id факультета;
- string Code — код.

9. Предоставление списка потоков

URL ресурса: `http://servername/ruzservice.svc/streams`.

HTTP-метод: GET.

Возвращает список: Да.

Формат ответа: JSON.

Возвращаемые значения:

- int Amount — количество обучающихся в потоке;
- int FormOfEducationGid — внешний id формы обучения;
- int FormOfEducationOid — id формы обучения;
- string StreamUID — текстовый идентификатор потока;
- string Abbr — аббревиатура;
- string Course — курс;

- string Faculty — факультет;
- int facultyGid — внешний id факультета;
- int facultyOid — id факультета;
- string FormOfEducation — форма обучения;
- string Name — наименование;
- int streamOid — id потока;
- int streamGid — внешний id потока;
- int YearOfEducation — год обучения.

10. Предоставление списка подгрупп

URL ресурса: `http://servername/ruzservice.svc/subgroups?isschedule={isSchedule}`.

HTTP-метод: GET.

Возвращает список: Да.

Формат ответа: JSON.

Входные параметры:

- IsSchedule (необязательный параметр) — признак необходимости проверки, есть ли для группы, к которой относится данная подгруппа, занятия хоть в одном расписании в статусе *Активный* или *Неутвержденный* или бронирования (в выходной список подгрупп будут отобраны только такие подгруппы); значения: true, false.

Возвращаемые значения:

- int subgroupOid — id подгруппы;
- int subgroupGid — внешний id подгруппы;
- char subgroupGUID — дополнительный идентификатор подгруппы в формате UUID;
- string Group — наименование группы;
- int groupOid — id группы;
- int groupGid — внешний id группы;
- char groupGUID — дополнительный идентификатор группы в формате UUID;
- string Name — наименование;
- string Abbr — аббревиатура.

11. Предоставление полного текста PUSH-сообщения по его ID

URL ресурса: `http://servername/ruzservice.svc/messages?id={messageId}`.

HTTP-метод: GET.

Возвращает список: Нет.

Формат ответа: JSON.

Входные параметры:

Int id — id сообщения, переданного в Push-уведомлении (обязательный параметр).

Возвращаемые значения:

Тип String. Полный текст сообщения.

12. Предоставление списка уведомлений пользователей РУЗ об изменении статусов заявок на бронирование/дооснащение аудиторий

URL ресурса: `http://servername/ruzservice.svc/notifications?days={days}`.

HTTP-метод: GET.

Возвращает список: Да.

Формат ответа: JSON.

Входные параметры:

Int days — обязательный параметр. Последние N дней, за которые требуется предоставить список уведомлений пользователей РУЗ об изменении статусов заявок на бронирование/дооснащение аудиторий.

Возвращаемые значения:

- string userType — тип пользователя, оформившего заявку;
- string senderEmail — электронный почтовый ящик пользователя, создавшего заявку;
- string note — краткое примечание;
- string noteFull — полное примечание;
- string dateCreate — дата создания заявки;
- int notificationOid — id уведомления.

13. Получение списка заявок на бронирование/дооснащение аудиторий (интервал не более 180 дней)

URL ресурса: `http://servername/ruzservice.svc/banofauditoriums?fromdate={fromDate}&todate={toDate}`.

HTTP-метод: GET.

Возвращает список: Да.

Формат ответа: JSON.

Входные параметры:

- string fromDate — начало интервала;
- string toDate — окончание интервала.

Возвращаемые значения:

- int kindBanOfAuditorium — вид заявки: 0 — бронирование, 1 — дооснащение;
- string auditorium — наименование аудитории;
- int auditoriumOid — id аудитории;
- char auditoriumGUID — дополнительный идентификатор аудитории в формате UUID;
- string dateOfNest — дата создания заявки;
- string timeBeg — время начала заявки;
- string timeEnd — время окончания заявки;

- string dateApproved — время согласования заявки;
- int typeOfLimitation — тип ограничения: 0 — обязательно, 1 — желательно, 2 — не учитывать;
- int typeOfSource — тип источника: 0 — УМП, 1 — ППС, 2 — мобильное устройство, 3 — другой;
- int status — статус заявки: 0 — черновой, 1 — утвержденный, 2 — на согласовании, 3 — отклонено;
- int typeOfAuditoriumOid — id типа аудитории;
- string typeOfAuditorium — наименование типа аудитории;
- int kindOfWorkOid — id вида работы;
- string kindOfWork — наименование вида работы;
- int buildingOid — id здания;
- string building — наименование здания;
- int tableType — тип столов: 0 — столы, 1 — парты, 2 — круглые, 3 — амфитеатр;
- string note — примечание;
- int equipment — дополнительное оборудование: 0 — нет, 1 — проектор, 2 — интерактивная доска;
- bool computerEquipment — компьютерное оборудование;
- bool mediaEquipment — медиаоборудование;
- string contingent — контингент;
- string software — ПО;
- string reason — причина;
- string username — пользователь, оформивший заявку;
- int amount — требуемая вместимость аудитории.
- string lecturers — преподаватели (строка преподавателей из коллекции запрета, разделенных ";", каждый преподаватель отображается по формату согласно установленным текущим общим параметрам);
- int supervision — признак аудиторной консультации: 1 — аудиторная консультация, 0 — нет.

14. Размещение заявки на бронирование/дооснащение аудитории от имени пользователя.

URL ресурса: `http://servername/ruzservice.svc/setbanofauditorium?username={username}&apitoken={apitoken}`.

HTTP метод: POST.

Возвращает список: Нет.

Формат ответа: JSON.

Входные параметры:

- string username — пользователь, оформивший заявку (обязательный параметр);
- string apitoken — значение API Token в Сервис/Общие параметры. (обязательный параметр, если значение API Token задано);

В теле запроса передается объект в формате JSON (см. выше [пункт 13](#)¹¹², "Возвращаемые значения").

- `int kindBanOfAuditorium` — вид заявки: 0 — бронирование, 1 — дооснащение (обязательный параметр);
- `string auditorium` — наименование аудитории;
- `int auditoriumOid` — id аудитории (обязательный параметр);
- `char auditoriumGUID` — дополнительный идентификатор аудитории в формате UUID;
- `string dateOfNest` — дата создания заявки (обязательный параметр);
- `string timeBeg` — время начала заявки (обязательный параметр);
- `string timeEnd` — время окончания заявки (обязательный параметр);
- `string dateApproved` — время согласования заявки;
- `int typeOfLimitation` — тип ограничения: 0 — обязательно, 1 — желательно, 2 — не учитывать;
- `int typeOfSource` — тип источника: 0 — УМП, 1 — ППС, 2 — мобильное устройство, 3 — другой;
- `int typeOfAuditoriumOid` — id типа аудитории;
- `string typeOfAuditorium` — наименование типа аудитории;
- `int kindOfWorkOid` — id вида работы;
- `string kindOfWork` — наименование вида работы;
- `int buildingOid` — id здания;
- `string building` — наименование здания;
- `int tableType` — тип столов: 0 — столы, 1 — парты, 2 — круглые, 3 — амфитеатр;
- `string note` — примечание;
- `int equipment` — дополнительное оборудование: 0 — нет, 1 — проектор, 2 — интерактивная доска;
- `bool computerEquipment` — компьютерное оборудование;
- `bool mediaEquipment` — медиаоборудование;
- `string contingent` — контингент;
- `string software` — ПО;
- `string reason` — причина;
- `string username` — пользователь, оформивший заявку;
- `int amount` — требуемая вместимость аудитории;
- `int supervision` — признак аудиторной консультации: 1 — аудиторная консультация, 0 — нет;
- `int status` — меняет статус создаваемой заявки на бронирование следующим образом: 0 — черновой, 1 — утвержденный, 2 — на согласовании;
- `int lecturers` — внешний id преподавателя (поле GID из таблицы lecturer);
- `string lecturersuid` — внешний id преподавателя в виде строки (поле UID из таблицы lecturer);

- char lecturerGUID — дополнительный идентификатор преподавателя в формате UUID;
- url1 — адрес гиперссылки 1;
- url1_description — текст гиперссылки 1;
- url2 — адрес гиперссылки 2;
- url2_description — текст гиперссылки 2;

 *Аудитория задается через один из параметров: auditoriumOid или auditoriumGUID. Сначала проверяется параметр auditoriumOid, и если auditoriumOid не передан, то аудитория определяется по auditoriumGUID.*

 *При сохранении параметра contingent (контингент) выполняется разбор строки с целью выделения групп. Предполагается, что в этом случае строка формируется пользователем как совокупность номеров групп, разделенных символом ";", например: 151;121;13-П1. Если разбор проходит успешно, то в коллекцию "Группы" заявки заносятся идентифицированные группы с последующим отображением этих заявок в расписаниях групп в качестве мероприятий.*

 *Преподаватель задается через один из параметров: lecturers, lecturersuid или lecturerGUID. Если преподавателей несколько, можно указать несколько значений, разделенных символом ";". Если задан параметр lecturers, то значение ищется в lecturer.Gid, если не найдено – поиск повторяется в lecturer.CustomUID. Если задан параметр lecturersuid, то значение ищется в lecturer.UID. Аналогично, если задан параметр lecturerGUID, то преподаватель определяется по lecturer.GUID.*

Возвращаемые значения:

- byte Code — код ошибки/успешного выполнения;
- string Description — описание;
- int lessonOid — идентификатор занятия созданного бронирования

Коды ошибок/успешного выполнения и их описание:

- Code: 0, Description: "ОК".
- Code: 1, Description: "Неправильно введены типы полей."
- Code: 2, Description: "Неправильно задан вид запрета аудитории."
- Code: 3, Description: "Аудитория не существует."
- Code: 4, Description: "Ошибка даты заявки."
- Code: 5, Description: "Ошибка даты согласования заявки."
- Code: 6, Description: "Ошибка времени начала."
- Code: 7, Description: "Ошибка времени окончания."
- Code: 8, Description: "Не заполнено описание."
- Code: 9, Description: "Тип аудитории не существует."
- Code: 10, Description: "Здание не существует."

- Code: 11, Description: "Тип столов не существует."
- Code:12, Description: "Вид работы не существует."
- Code: 13, Description: "Дополнительное оборудование не существует."
- Code: 14, Description: "Неправильно задан тип ограничения."
- Code: 15, Description: "Пользователь не найден."
- Code: 16, Description: "Токен не совпадает."

16. Подбор свободной аудитории по заданным параметрам

URL ресурса: `http://servername/ruzservice.svc/getavailableauditoriums?dateofnest={dateOfNest}&timebeg={timeBeg}&timeend={timeEnd}&kindbanofauditorium={kindBanOfAuditorium}&typeofauditoriumoid={typeOfAuditoriumOid}&buildingoid={buildingOid}&equipment={equipment}&computerequipment={computerEquipment}&mediaequipment={mediaEquipment}&amount={amount}&tabletype={tableType}`.

HTTP-метод: GET.

Аутентификация: Требуется передачи токена в заголовке "Authorization".

Возвращает список: Да.

Формат ответа: JSON.

Входные параметры:

- string dateOfNest — дата, на которую осуществляется подбор аудитории (обязательный параметр);
- string timeBeg — время начала (обязательный параметр);
- string timeEnd — время окончания (обязательный параметр);
- int kindBanOfAuditorium — вид заявки: 0 — бронирование, 1 — дооснащение;
- int typeOfAuditoriumOid — id типа аудитории;
- int buildingOid — id здания (обязательный параметр);
- int equipment — дополнительное оборудование: 0 — нет, 1 — проектор, 2 — интерактивная доска;
- bool computerEquipment — компьютерное оборудование;
- bool mediaEquipment — медиаоборудование;
- int amount — требуемая вместимость аудитории;
- int tableType — тип столов: 0 — столы, 1 — парты, 2 — круглые, 3 — амфитеатр.

Возвращаемые значения:

- int auditoriumOid — id аудитории;
- char auditoriumGUID — дополнительный идентификатор аудитории в формате UUID;
- string Building — здание;
- int buildingOid — id здания;
- string TypeOfAuditorium — тип аудитории;
- string Number — номер аудитории.

17. Предоставление списка состава группы

URL ресурса: `http://servername/ruzservice.svc/staffofgroup?groupOid={groupOid}&groupGuid={groupGuid}`.

HTTP-метод: GET.

Возвращает список: Да.

Формат ответа: JSON.

Входные параметры:

- `int groupOid` — id группы (обязательный параметр);
- `char groupGUID` — дополнительный идентификатор группы в формате UUID;

Возвращаемые значения:

- `int studentOid` — id студента;
- `int studentGid` — внешний id студента;
- `int studentGUID` — дополнительный идентификатор студента в формате UUID;
- `string fio` — Ф.И.О. (полное);
- `string shortFIO` — Ф.И.О. (только первые буквы слов);
- `string studentUID` — текстовый идентификатор студента.

18. Предоставление списка состава потоков

URL ресурса: `http://servername/ruzservice.svc/staffofstream?streamoid={streamOid}`.

HTTP-метод: GET.

Возвращает список: Да.

Формат ответа: JSON.

Входные параметры:

`Int streamOid` — id потока (необязательный параметр).

Если параметр `streamOid` не указан, метод предоставляет состав всех потоков.

Возвращаемые значения:

- `int GroupGid` — внешний id группы;
- `string GroupNumber` — номер группы;
- `int GroupOid` — id группы;
- `int streamOid` — id потока;
- `int SubgroupGid` — внешний id подгруппы;
- `string SubgroupName` — наименование подгруппы.
- `int SubgroupOid` — id подгруппы;
- `char GroupGUID` — дополнительный идентификатор группы в формате UUID;

19. Предоставление списка занятий и групповых мероприятий из запретов/заявок аудиторий в заданном интервале времени по указанным входным параметрам (интервал дат не более 180 дней)

URL ресурса: `http://servername/ruzservice.svc/lessons?fromdate={fromDate}&todate={toDate}&groupoid={groupOid}&groupGUID={groupGUID}`

&lectureroid={lecturerOid}&lecturerGUID={lecturerGUID}
&auditoriumoid={auditoriumOid}&auditoriumGUID={auditoriumGUID}
&language={language}.

HTTP-метод: GET.

Возвращает список: Да.

Формат ответа: JSON.

Входные параметры:

- string fromDate — начало интервала;
- string toDate — окончание интервала;
- int groupOid — id группы;
- char groupGUID — дополнительный идентификатор группы в формате UUID;
- int lecturerOid — id преподавателя;
- char lecturerGUID — дополнительный идентификатор преподавателя в формате UUID;
- int auditoriumOid — id аудитории;
- char auditoriumGUID — дополнительный идентификатор аудитории в формате UUID;
- int language — язык предоставления расписания: 1 — русский (по умолчанию), 2 — английский; если параметр не указан или имеет отличный от перечисленных вариант, то принимается значение 1;
- int lessonOid — идентификатор занятия.

 *Аудитория задается через один из параметров: auditoriumOid или auditoriumGUID. Сначала проверяется параметр auditoriumOid, и если auditoriumOid не передан, то аудитория определяется по auditoriumGUID. Группа и Преподаватель — аналогично.*

Возвращаемые значения:

- string auditorium — номер аудитории;
- int auditoriumAmount — вместимость аудитории;
- int auditoriumOid — id аудитории;
- char auditoriumGUID — дополнительный идентификатор аудитории в формате UUID;
- string author — имя пользователя-автора занятия;
- string beginLesson — время начала занятия;
- string building — аббревиатура здания проведения занятия;
- int buildingGid — внешний id здания;
- int buildingOid — id здания;
- int contentOfLoadOid — идентификатор нагрузки, соответствующий нагрузке поставленного занятия;
- int contentOfLoadUID — текстовый идентификатор нагрузки, соответствующий нагрузке поставленного занятия;
- int contentTableOfLessonsOid — id занятия в сетке пар;

- string contentTableOfLessonsName — наименование пары;
- date createddate — дата создания занятия в формате RFC;
- string date — дата проведения занятия;
- string dateOfNest — дата проведения занятия в формате Unix timestamp;
- int dayOfWeek — день недели (enum System.DayOfWeek);
- string dayOfWeekString — название дня недели (аббревиатура - Пн, Вт, Ср, Чт, Пт, Сб, Вс);
- string detailInfo — примечание;
- string discipline — наименование дисциплины;
- int disciplineOid — id дисциплины;
- int disciplineinplan — id дисциплины в плане;
- int disciplinetypeLoad — дисциплина в нагрузке;
- int duration — продолжительность занятия;
- string endLesson — время окончания занятия;
- string group — номер группы (для группового занятия);
- int groupOid — id группы (для группового занятия);
- string groupUID — текстовый id группы;
- char groupGUID — дополнительный идентификатор группы в формате UUID;
- int group_facultyoid — id факультета группы (для группового занятия);
- int hideinCapacity — флаг Скрывать для аудитории (1 - скрывать, 0 - показывать);
- bool isBan — флаг, показывающий, является ли занятие бронированием/запретом для аудитории (true/false);
- string kindOfWork — наименование вида работы (лекция, практика и т. п.); для мероприятий из запрета при установленном параметре BanOfAuditorium.Supervision=true выводится "Консультация студентов";
- int kindOfWorkComplexity — сложность вида работ;
- int kindOfWorkOid — id вида работ;
- string kindOfWorkUID — текстовый id вида работ;
- string lecturer — Фамилия и инициалы преподавателя;
- char lecturerCustomUID — дополнительный идентификатор преподавателя, отличие от person в типе данных (char);
- char lecturerGUID — дополнительный идентификатор преподавателя в формате UUID;
- string [lecturerEmail](#)⁵⁹ — Адрес электронной почты преподавателя;
- int lecturerOid — id преподавателя;
- string lecturerUID — идентификатор преподавателя в виде строки, используется в случае, если id содержит символы, например в табельном номере;
- string lecturer_rank — должность преподавателя;
- string lecturer_title — Ф.И.О. преподавателя;
- int lessonNumberEnd — номер пары окончания занятия;

- int lessonNumberStart — номер пары начала занятия;
- int lessonOid — идентификатор занятия
- date modifieddate — дата изменения занятия в формате RFC;
- string note — комментарий;
- string note_description — примечание (при isBan=true);
- string parentschedule — название расписания-родителя;
- string replaces — изменения в занятии
- string stream — наименование потока (для поточного занятия);
- int streamOid — id потока (для поточного занятия);
- int stream_facultyoid — id факультета (для поточного занятия);
- string subGroup — наименование подгруппы (для подгруппового занятия);
- int subgroupOid — id группы (для подгруппового занятия);
- int subgroup_facultyoid — id факультета (для подгруппового занятия);
- int tableofLessonsOid — id сетки пар;
- string tableofLessonsName — наименование сетки пар;
- url1 — адрес гиперссылки 1
- url1_description — текст гиперссылки 1
- url2 — адрес гиперссылки 2
- url2_description — текст гиперссылки 2
- listofLecturers — список преподавателей при isBan=true;

20. Предоставление списка персонального расписания занятий и групповых мероприятий из запретов/заявок аудиторий в заданном интервале времени по указанным входным параметрам (интервал дат не более 180 дней)

URL ресурса: `http://servername/ruzservice.svc/personlessons?fromdate={fromDate}&todate={toDate}&receivertype={receiverType}&groupoid={groupOid}&groupGUID={groupGUID}&GroupNumber={GroupNumber}&lectureroid={lecturerOid}&lecturerGUID={lecturerGUID}&auditoriumoid={auditoriumOid}&auditoriumGUID={auditoriumGUID}&studentoid={studentOid}&studentGUID={studentGUID}&uns={UNS}&email={email}&language={language}&facultyoid={facultyoid}&subGroupOid={subGroupOid}&buildingOid={buildingOid}&personOid={personOid}&disciplineinplan={disciplineinplan}&lessonOid={lessonOid}.`

HTTP-метод: GET.

Возвращает список: Да.

Формат ответа: JSON.

Входные параметры:

- string fromDate — начало интервала;
- string toDate — окончание интервала;
- int receiverType — тип параметра, для которого необходимо произвести поиск расписания занятий: 0 — студент (по умолчанию), 1 — преподаватель, 2 — аудитория, 3 — группа, 4 — здание, 5 - персона или физ. лицо (используется в случае, если необходимо выбрать расписание преподавателя, который работает в

нескольких подразделениях), 6 — факультет, 7—идентификатор дисциплины в плане, 8 — идентификатор занятия, 9 — подгруппа;

- int lecturerOid — id преподавателя (обязательный при условии receiverType = 1);
- char lecturerGUID — дополнительный идентификатор преподавателя в формате UUID;
- int auditoriumOid — id аудитории (обязательный при условии receiverType = 2);
- char auditoriumGUID — дополнительный идентификатор аудитории в формате UUID;
- int buildingOid — id здания (обязательный при условии receiverType = 4);
- int personOid — id персоны (обязательный при условии receiverType = 5);
- int facultyOid — id факультета группы (обязательный при условии receiverType = 6);
- int disciplineinplan — id дисциплины в плане (обязательный при условии receiverType = 7);
- int lessonOid — идентификатор занятия (обязательный при условии receiverType = 8);
- int subGroupOid — id подгруппы (обязательный при условии receiverType = 9);
- int studentOid — id студента;
- char student GUID — дополнительный идентификатор студента в формате UUID;
- string UNS — уникальный номер обучающегося;
- string email — электронный адрес обучающегося;
- int groupOid — id группы;
- char groupGUID — дополнительный идентификатор группы в формате UUID;
- string GroupNumber — номер группы;
- int language — язык предоставления расписания: 1 — русский (по умолчанию), 2 — английский; если параметр не указан или имеет отличный от перечисленных вариант, то принимается значение 1;
- char SP_DisciplineInPlan_ID — идентификатор дисциплины в плане системы SmartPlan (для ReceiverType = 7).

 Если receiverType = 0, обязательно указывается один из параметров: studentOid (при studentOid=0 проверяется параметр studentGUID), email или UNS. По одному из этих параметров в указанной последовательности идентифицируется обучающийся в справочнике обучающихся. Дополнительно может указываться также groupOid (при groupOid=0 проверяется параметр groupGUID), тогда расписание будет уточнено для конкретного обучающегося в пределах заданной группы.

 Если receiverType = 3, обязательно указывается один из параметров: groupOid (при groupOid=0 проверяется параметр groupGUID) или GroupNumber. Если номер группы не является уникальным, то возвращается расписание по всем группам с указанным номером.

 Преподаватель задается через один из параметров: lecturerOid или lecturerGUID. Сначала проверяется параметр lecturerOid, и если lecturerOid не передан, то преподаватель определяется по lecturerGUID. Аудитория — аналогично.

Возвращаемые значения:

- string auditorium — номер аудитории;
- int auditoriumAmount — вместимость аудитории;
- int auditoriumOid — id аудитории;
- char auditoriumGUID — дополнительный идентификатор аудитории в формате UUID;
- int auditoriumfloor — этаж, на котором располагается аудитория;
- string author — имя пользователя-автора занятия;
- string beginLesson — время начала занятия;
- string building — аббревиатура здания проведения занятия;
- int buildingGid — внешний id здания;
- int buildingOid — id здания;
- int contentOfLoadOid — идентификатор нагрузки, соответствующий нагрузке поставленного занятия;
- int contentOfLoadUID — текстовый идентификатор нагрузки, соответствующий нагрузке поставленного занятия;
- int contentTableOfLessonsOid — id занятия в сетке пар;
- string contentTableOfLessonsName — наименование пары;
- date createddate — дата создания занятия в формате RFC;
- string date — дата проведения занятия;
- string dateOfNest — дата проведения занятия в формате Unix timestamp;
- int dayOfWeek — день недели (enum System.DayOfWeek);
- string dayOfWeekString — название дня недели (аббревиатура - Пн, Вт, Ср, Чт, Пт, Сб, Вс);
- string detailInfo — примечание;
- string discipline — наименование дисциплины;
- int disciplineOid — id дисциплины;
- string disciplineUID — идентификатор дисциплины в виде строки;
- int disciplineinplan — id дисциплины в плане;
- int disciplinetypeLoad — дисциплина в нагрузке;
- int duration — продолжительность занятия;
- string endLesson — время окончания занятия;
- string group — номер группы (для группового занятия);
- int groupOid — id группы (для группового занятия);
- string groupSP_Uid — дополнительный текстовый идентификатор группы;
- string groupUID — текстовый id группы;
- char groupGUID — дополнительный идентификатор группы в формате UUID;
- string group_facultyUID — текстовый id факультета группы;

- string group_faculty_SP_Uid — дополнительный текстовый идентификатор факультета группы;
- string group_facultyname — наименование факультета группы;
- int group_facultyoid — id факультета группы (для группового занятия);
- int hideinCapacity — флаг Скрывать для аудитории (1 - скрывать, 0 - показывать);
- bool isBan — флаг, показывающий, является ли занятие бронированием/запретом для аудитории (true/false);
- string kindOfWork — наименование вида работы (лекция, практика и т. п.); для мероприятий из запрета при установленном параметре BanOfAuditorium.Supervision=true выводится "Консультация студентов";
- int kindOfWorkComplexity — сложность вида работ;
- int kindOfWorkOid — id вида работ;
- string kindOfWorkUID — текстовый id вида работ;
- string lecturer — Фамилия и инициалы преподавателя;
- char lecturerCustomUID — дополнительный идентификатор преподавателя, отличие от person в типе данных (char).
- string [lecturerEmail](#)⁵⁹ — Адрес электронной почты преподавателя
- int lecturerOid — id преподавателя;
- char lecturerGUID — дополнительный идентификатор преподавателя в формате UUID;
- string lecturerUID — идентификатор преподавателя в виде строки, используется в случае, если id содержит символы, например в табельном номере;
- string lecturer_rank — должность преподавателя;
- string lecturer_post_Oid — Oid должности преподавателя;
- string lecturer_title — Ф.И.О. преподавателя;
- int lessonNumberEnd — номер пары окончания занятия;
- int lessonNumberStart — номер пары начала занятия;
- int lessonOid — идентификатор занятия
- date modifieddate — дата изменения занятия в формате RFC;
- string note — комментарий;
- string note_description — примечание (при isBan=true);
- int packageNumber — номер пакета нагрузки, по которой поставлено занятие;
- bool openlesson — признак открытой лекции (1 - является открытой лекцией, 0 - не является открытой лекцией);
- int parentSchedule_Status — статус расписания-родителя занятия (0 - Активный, 1 - Неутвержденный, 2 - Черновой, 3 - Архивный);
- string parentschedule — название расписания-родителя;
- string replaces — изменения в занятии;
- char specialization_name — наименование специализации нагрузки занятия;
- int specialization_oid — Oid специализации нагрузки занятия;

- string stream — наименование потока (для поточного занятия);
- int streamOid — id потока (для поточного занятия);
- string streamUId — идентификатор потока в виде строки;
- int stream_facultyoid — id факультета (для поточного занятия);
- string subGroup — наименование подгруппы (для подгруппового занятия);
- char subgroupGUID — дополнительный идентификатор подгруппы в формате UUID;
- int subgroupOid — id группы (для подгруппового занятия);
- string subgroupUId — идентификатор подгруппы в виде строки;
- int subgroup_facultyoid — id факультета (для подгруппового занятия);
- int subgroup_groupOid — Oid группы (для подгруппового занятия);
- string subject — тема занятия;
- int tableofLessonsOid — id сетки пар;
- string tableofLessonsName — наименование сетки пар;
- int typeOfContingent — идентификатор типа контингента (1 - подгруппа, 2 - группа, 3 - поток);
- url1 — адрес гиперссылки;
- url1_description — текст гиперссылки 1;
- url2 — адрес гиперссылки 2;
- url2_description — текст гиперссылки 2;
- listofLecturers — список преподавателей;
- listGroups — список групп;
- listSubGroups — список подгрупп;

21. Предоставление списка или количества занятий персонального расписания занятий и групповых мероприятий из запретов/заявок аудиторий в заданном интервале времени по указанным входным параметрам (интервал дат не более 180 дней), вариант 2

URL псцпса: `http://servername/ruzservice.svc/v2/personlessons?fromdate={fromDate}&todate={toDate}&receivertype={receiverType}&groupoid={groupOid}&groupGUID={groupGUID}&lectureroid={lecturerOid}&lecturerGUID={lecturerGUID}&auditoriumoid={auditoriumOid}&auditoriumGUID={auditoriumGUID}&studentoid={studentOid}&studentGUID={studentGUID}&uns={UNS}&email={email}&iscount={isCount}&language={language}`.

HTTP-метод: GET.

Возвращает объект.

Формат ответа: JSON.

Входные параметры:

- string fromDate — начало интервала;
- string toDate — окончание интервала;
- int receiverType — тип параметра, для которого необходимо произвести поиск расписания занятий: 0 — студент (по умолчанию), 1 — преподаватель, 2 — аудитория, 3 — группа;

- int lecturerOid — id преподавателя (обязательный при условии receiverType = 1);
- char lecturerGUID — дополнительный идентификатор преподавателя в формате UUID;
- int auditoriumOid — id аудитории (обязательный при условии receiverType = 2);
- char auditoriumGUID — дополнительный идентификатор аудитории в формате UUID;
- int studentOid — id студента;
- char student GUID — дополнительный идентификатор студента в формате UUID;
- string UNS — уникальный номер обучающегося;
- string email — электронный адрес обучающегося;
- int groupOid — id группы;
- char groupGUID — дополнительный идентификатор группы в формате UUID;
- string GroupNumber — номер группы;
- bool isCount — признак получения только количества или самой коллекции занятий (true/false);
- int language — язык предоставления расписания: 1 — русский (по умолчанию), 2 — английский; если параметр не указан или имеет отличный от перечисленных вариант, то принимается значение 1.

💡 Если receiverType = 0, обязательно указывается один из параметров: studentOid (при studentOid=0 проверяется параметр studentGUID), email или UNS. По одному из этих параметров в указанной последовательности идентифицируется обучающийся в справочнике обучающихся. Дополнительно может указываться также groupOid (при groupOid=0 проверяется параметр groupGUID), тогда расписание будет уточнено для конкретного обучающегося в пределах заданной группы.

💡 Если receiverType = 3, обязательно указывается один из параметров: groupOid (при groupOid=0 проверяется параметр groupGUID) или GroupNumber. Если номер группы не является уникальным, то возвращается расписание по всем группам с указанным номером.

💡 Преподаватель задается через один из параметров: lecturerOid или lecturerGUID. Сначала проверяется параметр lecturerOid, и если lecturerOid не передан, то преподаватель определяется по lecturerGUID. Аудитория — аналогично.

Возвращаемые значения:

- Если isCount=true, то на выходе объект вида, например:

```
{"Count":Количество,"Lessons":null,"StatusCode":{"Code":  
ЦифровойКод,"Description":"Текст"}}
```

- Если isCount=false, то:

```
{"Count":Количество,"Lessons":[Коллекция объектов  
занятий],"StatusCode":{"Code":ЦифровойКод,"Description":"Текст"}}
```

Поля объекта:

- int Count — количество объектов коллекции;
- поля объекта StatusCode:
 - int Code — код ошибки/успешного выполнения;

- string Description — описание ошибки/успешного выполнения.

Поля объекта занятия:

- string stream — наименование потока (для поточного занятия);
- int streamOid — id потока (для поточного занятия);
- string group — номер группы (для группового занятия);
- int groupOid — id группы (для группового занятия);
- char groupGUID — дополнительный идентификатор группы в формате UUID;
- string subGroup — наименование подгруппы (для подгруппового занятия);
- int subgroupOid — id подгруппы (для подгруппового занятия);
- string date — дата проведения занятия;
- int dayOfWeek — день недели (enum System.DayOfWeek);
- string dayOfWeekString — название дня недели (аббревиатура - Пн, Вт, Ср, Чт, Пт, Сб, Вс);
- string beginLesson — время начала занятия;
- string endLesson — время окончания занятия;
- int duration — продолжительность занятия;
- int contentOfLoadOid — идентификатор нагрузки, соответствующий нагрузке поставленного занятия;
- int contentOfLoadUID — текстовый идентификатор нагрузки, соответствующий нагрузке поставленного занятия;
- string kindOfWork — наименование вида работы (лекция, практика и т. п.); для мероприятий из запрета при установленном параметре BanOfAuditorium.Supervision=true выводится "Аудиторная консультация";
- string discipline — наименование дисциплины;
- int disciplineOid — id дисциплины;
- int disciplineinplan — id дисциплины в плане;
- int disciplinetypeLoad — дисциплина в нагрузке;
- string building — аббревиатура здания проведения занятия;
- string auditorium — номер аудитории;
- int auditoriumOid — id аудитории;
- char auditoriumGUID — дополнительный идентификатор аудитории в формате UUID;
- int auditoriumAmount — вместимость аудитории;
- string lecturer — должность и Ф.И.О. преподавателя;
- int lecturerOid — id преподавателя;
- char lecturerGUID — дополнительный идентификатор преподавателя в формате UUID;
- string lecturerUID — идентификатор преподавателя в виде строки, используется в случае, если id содержит символы, например в табельном номере;

- char lecturerCustomUID — дополнительный идентификатор преподавателя, отличие от person в типе данных (char).
- string author — имя пользователя-автора занятия;
- string note — комментарий;
- date createddate — дата создания занятия в формате RFC;
- string dateOfNest — дата проведения занятия в формате Unix timestamp;
- int hideinCapacity — флаг Скрывать для аудитории (1 - скрывать, 0 - показывать);
- bool isBan — флаг, показывающий, является ли занятие бронированием/запретом для аудитории (true/false);
- int lessonNumberStart — номер пары начала занятия;
- int lessonNumberEnd — номер пары окончания занятия;
- date modifieddate — дата изменения занятия в формате RFC;
- string parentschedule — название расписания-родителя;
- string detailInfo — примечание.

Коды ошибок:

- Code = 20, Description = "Не определены данные по обучающемуся или не найдено соответствие."
- Code = 21, Description = "Не определены данные по группе или не найдено соответствие."
- Code = 22, Description = "Не определены данные по преподавателю или не найдено соответствие."
- Code = 23, Description = "Не определены данные по аудитории или не найдено соответствие."

22. Предоставление списка персонального расписания занятий и групповых мероприятий из запретов/заявок аудиторий в заданном интервале времени по указанным входным параметрам в формате .ics (интервал дат не более 180 дней)

URL ресурса: `http://servername/ruzservice.svc/lessonscalendar?fromdate={fromDate}&todate={toDate}&receivertype={receiverType}&groupoid={groupGUID}&groupoid={groupGUID}&lectureroid={lecturerOid}&lecturerGUID={lecturerGUID}&auditoriumoid={auditoriumOid}&auditoriumGUID={auditoriumGUID}&studentoid={studentOid}&studentGUID={studentGUID}&uns={UNS}&email={email}&language={language}`.

HTTP-метод: GET.

Возвращает список: Нет.

Формат ответа: поток байтов. Заголовок "Content-Disposition" со значением прикрепленного файла, имеющего расширение ics.

Входные параметры:

- string fromDate — начало интервала;
- string toDate — окончание интервала;

- `int receiverType` — тип параметра, для которого необходимо произвести поиск расписания занятий: 0 — студент (по умолчанию), 1 — преподаватель, 2 — аудитория, 3 — группа;
- `int groupId` — id группы (обязательный при условии `receiverType = 3`);
- `char groupGUID` — дополнительный идентификатор группы в формате UUID;
- `int lecturerOid` — id преподавателя (обязательный при условии `receiverType = 1`);
- `char lecturerGUID` — дополнительный идентификатор преподавателя в формате UUID;
- `int auditoriumOid` — id аудитории (обязательный при условии `receiverType = 2`);
- `char auditoriumGUID` — дополнительный идентификатор аудитории в формате UUID;
- `int studentOid` — id студента;
- `char student GUID` — дополнительный идентификатор студента в формате UUID;
- `string UNS` — уникальный номер обучающегося;
- `string email` — электронный адрес обучающегося;
- `int language` — язык предоставления расписания: 1 — русский (по умолчанию), 2 — английский; если параметр не указан или имеет отличный от перечисленных вариант, то принимается значение 1.

 Если `receiverType = 0`, обязательно указывается один из параметров: `studentOid` (при `studentOid=0` проверяется параметр `studentGUID`), `email` или `UNS`. По одному из этих параметров в указанной последовательности идентифицируется обучающийся в справочнике обучающихся. Дополнительно может указываться также `groupId` (при `groupId=0` проверяется параметр `groupGUID`), тогда расписание будет уточнено для конкретного обучающегося в пределах заданной группы.

 Преподаватель задается через один из параметров: `lecturerOid` или `lecturerGUID`. Сначала проверяется параметр `lecturerOid`, и если `lecturerOid` не передан, то преподаватель определяется по `lecturerGUID`. Аудитория — аналогично.

Возвращаемые значения:

См. выше пункт "[Формат ответа](#)¹²⁷" данного метода.

23. Предоставление Академкалендаря

URL ресурса: `http://servername/ruzservice.svc/semester`

HTTP-метод: GET.

Возвращает список: Да.

Формат ответа: JSON.

Возвращаемые значения:

- `string semesterAbbr` — аббревиатура семестра;
- `string semesterDateFrom` — дата начала семестра;
- `string semesterDateTo` — дата окончания семестра;

- int semesterFlag — флаг Признак семестра (0-нет; 1-да);
- int semesterGid — внешний id семестра;
- string semesterName — наименование семестра;
- int semesterOid — id семестра;
- int semesterParentID — id семестра-родителя;
- string semesterUID — текстовый идентификатор семестра.

24. Предоставление Сетки пар

URL ресурса: `http://servername/ruzservice.svc/tableoflesson`

HTTP-метод: GET.

Возвращает список: Да.

Формат ответа: JSON.

- string abbr — аббревиатура сетки пар;
- string code — код;
- string name — наименование сетки пар;
- int tableoflessonoid — id сетки пар;
- int withdifferencesbydayofweek — флаг С отличиями по дням недели - тип сетки пар: 0 - обычная (все дни одинаковые); 1 - с разбивкой по дням недели;
- int yearofeducation — год обучения.

25. Предоставление занятий в сетках пар

URL ресурса: `http://servername/ruzservice.svc/contenttableoflesson`

HTTP-метод: GET.

Возвращает список: Да.

Формат ответа: JSON.

Возвращаемые значения:

- int contenttableoflessonoid — id занятия в сетке пар;
- int duration — продолжительность занятия, АЧ;
- int extdayofweek — номер дня недели (0 – для всех дней недели. 1 – понедельник и т.п.);
- int number — номер пары в день;
- int tableoflesson — id сетки пар;
- string timebeg — время начала занятия;
- string timeend — время окончания занятия.

26. Предоставление списка всех связей преподаватель - группа/поток/подгруппа

URL ресурса: `http://servername/ruzservice.svc/lecturersgroups`

HTTP-метод: GET.

Возвращает список: Да.

Формат ответа: JSON.

Входные параметры:

- string fromDate — начало интервала;
- string toDate — окончание интервала;
- int lecturerOid — id преподавателя;
- char lecturerGUID — дополнительный идентификатор преподавателя в формате UUID;
- string lecturerUID — текстовый идентификатор преподавателя.

 *Преподаватель задается через один из параметров: lecturerOid или lecturerGUID. Сначала проверяется параметр lecturerOid, и если lecturerOid не передан, то преподаватель определяется по lecturerGUID.*

Возвращаемые значения:

- string PersonID — идентификатор персоны;
- string beginLesson — время начала занятия;
- string dateOfNest — дата проведения занятия в формате Unix timestamp
- string disciplineUID — текстовый идентификатор дисциплины;
- int disciplineOid — id дисциплины;
- string endLesson — время окончания занятия;
- int groupOid — числовой id группы;
- string groupUID — текстовый идентификатор группы;
- char groupGUID — дополнительный идентификатор группы в формате UUID;
- int lecturerOid — числовой id преподавателя;
- char lecturerGUID — дополнительный идентификатор преподавателя в формате UUID;
- string lecturerUID — текстовый идентификатор преподавателя.

27. Удаление заявки на бронирование/дооснащение аудитории

URL ресурса: <http://servername/ruzservice.svc/deletebanofauditorium?username={username}&apitoken={apitoken}>.

HTTP метод: POST.

Возвращает список: Нет.

Формат ответа: JSON.

Входные параметры:

- string username — пользователь, удаляющий заявку (обязательный параметр);
- string apitoken — значение API Token в Сервис/Общие параметры. (обязательный параметр, если значение API Token задано);
- int lessonOid — идентификатор занятия удаляемого бронирования (обязательный);
- string reason — примечание, причина отмены, до 500 символов;

Возвращаемые значения:

- byte Code — код ошибки/успешного выполнения;
- string Description — описание;

Коды ошибок/успешного выполнения и их описание:

- Code: 0, Description: "ОК".
- Code: 1, Description: "Ошибка удаления".
- Code: 3, Description: "Пользователь не найден".
- Code: 3, Description: "Токен не совпадает".

В результате выполнения метода статус заявки меняется на "Отклонено".

Методом `deletebanofauditorium` можно удалить только одиночное бронирование (аналогично тому, как метод [setbanofauditorium](#)^[113] создает только одиночное бронирование).

28. Предоставление списка студентов

URL ресурса: `http://servername/ruzservice.svc/Students`.

HTTP-метод: GET.

Возвращает список: Да.

Формат ответа: JSON.

Возвращаемые значения:

- string [email](#)^[59] — электронная почта студента;
- string FIO — Ф.И.О. студента полностью;
- string groupName — наименование группы;
- int studentGid — внешний id студента;
- int studentOid — id студента;
- char student GUID — дополнительный идентификатор студента в формате UUID;
- string studentUID — текстовый идентификатор студента;
- string uns — УНС, уникальный номер студента.

29. Предоставление списка дисциплин

URL ресурса: `http://servername/ruzservice.svc/discipline`.

HTTP-метод: GET.

Возвращает список: Да.

Формат ответа: JSON.

Возвращаемые значения:

- string abbr — аббревиатура дисциплины;
- int chairOid — id кафедры;
- string code — код дисциплины;
- int complexity — сложность дисциплины;
- string disciplineKind — вид дисциплины;

- int disciplineOid — id дисциплины;
- string disciplineType — тип дисциплины;
- long gid — внешний id дисциплины;
- string name — наименование дисциплины;
- int priority — приоритет дисциплины;
- string uid — текстовый id дисциплины.

30. Предоставление списка нагрузок

URL ресурса: `http://servername/ruzservice.svc/contentofload?fromdate={fromDate}&todate={toDate}`.

HTTP-метод: GET.

Возвращает список: Да.

Формат ответа: JSON.

Входные параметры:

- string fromDate — начало интервала;
- string toDate — окончание интервала.

Возвращаемые значения:

- int amount — количество часов;
- int amountUnit — количество в Единицах измерения (ЕИ);
- int auditoriumOid — id аудитории;
- char auditoriumGUID — дополнительный идентификатор аудитории в формате UUID;
- int chairOid — id кафедры;
- int cycleOfDisciplineOid — id цикла дисциплин
- int disciplineInPlan — id дисциплины в плане
- int disciplineOid — id дисциплины;
- string endDate — Окончание;
- int Gid — внешний id нагрузки;
- int groupOid — id группы;
- char groupGUID — дополнительный идентификатор группы в формате UUID;
- int kindOfWorkOid — id вида работ;
- int languageOid — id языка преподавания;
- int lecturerOid — id преподавателя;
- char lecturerGUID — дополнительный идентификатор преподавателя в формате UUID;
- int Oid — id нагрузки;
- string packageNumber — номер пакета;
- bool publicURL — флаг Публиковать URL по нагрузке (true/false);
- bool remotely — флаг Читается дистанционно (true/false);

- int semesterOid — id семестра;
- bool signOfContinuity — флаг Признак непрерывности (true/false);
- string startDate — Начало;
- int streamOid — id потока;
- int subgroupOid — id подгруппы;
- int typeOfContingent — тип Контингента (1 — подгруппа, 2 — группа, 3 — поток);
- string uid — текстовый id нагрузки;
- int unit — Единица измерения (ЕИ) (0 — часы (лекции, практики и т. п.), 1 — единицы (экзамен, зачет, практика и т. п.)).

31. Предоставление графика прохождения дисциплины

URL ресурса: `http://servername/ruzservice.svc/denormalizationofload?semesteroid={semesterOid}&groupoid={groupOid}&lectureroid={lecturerOid}&studentoid={studentOid}&groupGUID={groupGUID}&lecturerGUID={lecturerGUID}&studentGUID={studentGUID}`

HTTP-метод: GET.

Возвращает список: Да.

Формат ответа: JSON.

Входные параметры:

- int semesterOid — id семестра (обязательный параметр);

Обязательным также является использование хотя бы одного из параметров:

- int groupOid — id группы;
- int lecturerOid — id преподавателя;
- int studentOid — id студента;
- char groupGUID — дополнительный идентификатор группы в формате UUID;
- char lecturerGUID — дополнительный идентификатор преподавателя в формате UUID;
- char studentGUID — дополнительный идентификатор студента в формате UUID;



Преподаватель задается через один из параметров: lecturerOid или lecturerGUID. Сначала проверяется параметр lecturerOid, и если lecturerOid не передан, то преподаватель определяется по lecturerGUID. Группа, Студент — аналогично.

Возвращаемые значения:

- string discipline — наименование дисциплины;
- int disciplineOid — id дисциплины;
- int groupOid — id группы;
- char groupGUID — дополнительный идентификатор группы в формате UUID;
- string kindOfWork — наименование вида работы;
- int kindOfWorkOid — id вида работ;
- int streamOid — id потока;

- int subGroupOid — id подгруппы;
- int amount — количество часов;
- int denormalizationOid — id денормализации;
- int discreteness — дискретность;
- string endDate — Окончание;
- string startDate — Начало

32. Предоставление списка связей студент - группа

URL ресурса: `http://servername/ruzservice.svc/studentgroups?studentOid={studentOid}&studentGUID={studentGUID}`

HTTP-метод: GET.

Возвращает список: Да.

Формат ответа: JSON.

Входные параметры:

- int studentOid — id студента;
- char studentGUID — дополнительный идентификатор студента в формате UUID

 Студент задается через один из параметров: *studentOid* или *studentGUID*. Сначала проверяется параметр *studentOid*, и, если *studentOid* не передан, то преподаватель определяется по *studentGUID*.

Возвращаемые значения:

- char groupGUID — дополнительный идентификатор группы в формате UUID;
- int groupOid — числовой id группы;
- char studentGUID — дополнительный идентификатор студента в формате UUID;
- int studentOid — id студента.

33. Предоставление списка наименований объектов на иностранном языке

URL ресурса: `http://servername/ruzservice.svc/resourceLanguage?languageOid={languageOid}&objectapply={objectapply}`

HTTP-метод: GET.

Возвращает список: Да.

Формат ответа: JSON.

Входные параметры:

- languageOid — id языка;
- objectapply — id типа объекта

Возвращаемые значения:

- int Oid — id записи;
- long Gid — внешний id объекта;
- string Uid — текстовый id объекта;
- int languageOid — id языка;
- int objectApply — id типа объекта;

- int objectId — id объекта;
- string Abbr — аббревиатура;
- string Name — наименование.

 Метод не может быть применен для получения списка наименований объектов на русском языке.

34. Предоставление списка всех типов объектов

URL ресурса: http://servername/ruzservice.svc/objectApply

HTTP-метод: GET.

Возвращает список: Да.

Формат ответа: JSON.

Возвращаемые значения:

- int ObjectApply — id типа объекта;
- string Name — наименование.

35. Предоставление списка языков

URL ресурса: http://servername/ruzservice.svc/languages

HTTP-метод: GET.

Возвращает список: Да.

Формат ответа: JSON.

Возвращаемые значения:

- int languageOid — id языка;
- long languageGid — внешний id языка;
- string languageUid — текстовый id языка;
- string Code — код;
- string Abbr — аббревиатура;
- string AbbrEng — аббревиатура на английском языке;
- string Name — наименование;
- string Note — примечание.

36. Предоставление списка специальностей

URL ресурса: http://servername/ruzservice.svc/speciality

HTTP-метод: GET.

Возвращает список: Да.

Формат ответа: JSON.

Возвращаемые значения:

- string abbr — аббревиатура специальности;
- string code — код специальности;
- string name — наименование специальности;
- int specialityGid — внешний ID (GID) специальности;

- int specialityOid — OID специальности;
- string specialityUID — UID специальности.

37. Предоставление списка специализаций

URL ресурса: http://servername/ruzservice.svc/specialization

HTTP-метод: GET.

Возвращает список: Да.

Формат ответа: JSON.

Возвращаемые значения:

- string code — код специализации;
- string name — наименование специализации;
- int specialityGid — Gid специальности-родителя;
- int specialityOid — OID специальности-родителя
- string specialityUID — UID специальности-родителя;
- int specializationGid — внешний ID (GID) специализации;
- int specializationOid — OID специализации;
- string specializationUID — UID специализации.

38. Предоставление списка должностей

URL ресурса: http://servername/ruzservice.svc/posts

HTTP-метод: GET.

Возвращает список: Да.

Формат ответа: JSON.

Возвращаемые значения:

- string abbr — аббревиатура должности;
- string code — код должности;
- string name — наименование должности;
- int postGID — внешний ID (GID) должности;
- int postOid — OID должности;
- string postUID — UID должности.

39. Предоставление списка занятий, в которые были внесены изменения в заданном интервале времени

URL ресурса: http://servername/RUZService/RUZService.svc/ContentOfScheduleLog?
fromdate={fromDate}&todate={toDate}&starttime={modstarttime}
&endtime={modendtime}&student={studentOid}&studentemail={studentemail}
&receivertype={receiverType}

HTTP-метод: GET.

Возвращает список: Да.

Формат ответа: JSON.

Входные параметры:

- string fromDate — начало интервала;

- string toDate — окончание интервала;
- float modstarttime — начало временного периода (время в формате чч-мм) (необязательный параметр);
- float modendtime — окончание временного периода (время в формате чч-мм) (необязательный параметр);
- int receiverType — тип параметра, для которого необходимо произвести поиск изменений: 1 — преподаватель, 2 — дата, 3 — аудитория, 4 — вид работ (необязательный параметр);
- int studentOid — id студента (необязательный параметр);
- string studentemail — электронная почта студента (необязательный параметр);

Возвращаемые значения:

- int contentOfScheduleOid — oid изменённого занятия;
- string dateCreate — дата и время изменения занятия;
- int typeLog — тип изменения (1 – создание, 2 – удаление, 3 - изменение занятия);

Остальные возвращаемые значения: Old (значение параметра до изменения) и New (значение параметра после изменения) полей таблицы ContentOfScheduleLog, описывающей изменения в конкретном занятии:

- int auditoriumNew\Old — oid аудитории;
- string auditoriumNameNew\Old - наименование аудитории;
- string auditoriumNumberNew\Old – номер аудитории;
- int coefDiscardNew\Old — коэффициент списания нагрузки;
- int contentOfLoadNew\Old — идентификатор нагрузки занятия;
- int contentOfTechnologicalCardNew\Old — содержание технологической карты (0 - нет, 1 - да);
- int contentTableOfLessonNew\Old — id занятия в сетке пар;
- string dateOfNestNew\Old — дата проведения занятия в формате Unix timestamp;
- int disciplineNew\Old — id дисциплины;
- string disciplineNameNew\Old — наименование дисциплины;
- int durationNew\Old — продолжительность занятия;
- string endOnNew\Old — дата окончания занятия;
- string endTimeNew\Old — время окончания занятия;
- string groupNameNew\Old — наименование группы;
- int groupNew\Old — id группы;
- string groupNumberNew\Old — номер группы;
- kindOfWorkNew\Old — id вида работ;
- int labelNew\Old — признак изменения занятия в активном расписании (0 – нет, 1 – да);
- string lecturer2FIONew\Old — ФИО преподавателя2;
- int lecturer2New\Old — id преподавателя2;
- string lecturer2PostNew\Old — наименование должности преподавателя2;

- string lecturerFIONew\Old — ФИО преподавателя;
- int lecturerNew\Old — id преподавателя;
- string lecturerPostNew\Old — наименование должности преподавателя;
- int notToTakeContentOfLoadNew\Old — флаг не учитывать в нагрузке (0 – нет, 1 – да);
- string noteNew\Old — примечание;
- int parentScheduleNew\Old — id расписания-родителя;
- int publicURLNew\Old — флаг Публиковать URL (0 – нет, 1 – да);
- string recurrenceInfoXmlNew\Old — системный параметр;
- string recurrencePatternNew\Old — системный параметр;
- int repeatSettingNew\Old — настройки повторения занятия (0 - нет, 1 – раз в неделю, 2 – раз в 2 недели);
- string replacesNew New\Old — описание изменений в карточке занятия;
- int scheduleNew\Old — id расписания;
- string startOnNew\Old — дата начала занятия;
- string startTimeNew\Old — время начала занятия;
- string streamAbbrNew\Old — аббревиатура потока;
- string streamNameNew\Old — наименование потока;
- int streamNew\Old — id потока;
- string subGroupAbbrNew\Old — аббревиатура подгруппы;
- int subGroupNew\Old — id подгруппы;
- string subGroupNameNew\Old — наименование подгруппы;
- string subjectNew\Old — id тема занятия;
- int typeOfContingentNew\Old — тип контингента (1 — подгруппа, 2 — группа, 3 — поток);
- uRL1New\Old — адрес гиперссылки 1;
- uRL1_description New\Old — текст гиперссылки 1;
- uRL2New\Old — адрес гиперссылки 2;
- uRL2_description New\Old — текст гиперссылки 2.

 В отличие от журнала изменений, в который записываются изменения только в расписаниях со статусом активный, метод *ContentOfScheduleLog* предоставляет список изменений для расписаний с любым статусом.

40. Предоставление списка связей академическая группа - учебная группа

URL ресурса: `http://servername/ruzservice.svc/staffofedugroup?groupOid={groupOid}&groupGuid={groupGuid}`

HTTP-метод: GET.

Возвращает список: Да.

Формат ответа: JSON.

Входные параметры:

- int groupOid — id группы (необязательный параметр);
- char groupGUID — дополнительный идентификатор группы в формате UUID (необязательный параметр).

Если параметры не указаны, метод предоставляет состав всех учебных групп.

Возвращаемые значения:

- int Academ_GroupCourse — курс;
- string Academ_GroupFaculty — факультет;
- int Academ_GroupFacultyOid — id факультета;
- string Academ_GroupFormOfEducation — форма обучения;
- int Academ_GroupGUID — дополнительный идентификатор академической группы в формате UUID;
- int Academ_GroupGid — внешний id академической группы;
- int Academ_GroupKindEducation — уровень образования (0 — Бакалавриат, 1 — Магистратура, 2 — Специалитет, 3 — Аспирантура, 4 — Дополнительное профессиональное образование, 5 — Дополнительное образование, 6 — Среднее профессиональное образование);
- string Academ_GroupName — наименование академической группы;
- string Academ_GroupNumber — номер академической группы;
- int Academ_GroupOid — id академической группы;
- int Academ_GroupPlannedamount — плановое количество студентов в академической группе;
- int Academ_GroupUID — внешний id академической группы;
- int Edu_GroupGUID — дополнительный идентификатор учебной группы в формате UUID;
- int Edu_GroupGid — внешний id учебной группы;
- string Edu_GroupNumber — номер учебной группы;
- int Edu_GroupOid — id учебной группы;
- int Edu_GroupUID — внешний id учебной группы.

41. Предоставление списка состава подгруппы

URL ресурса: <http://servername/ruzservice.svc/staffofsubgroup?subgroupOid={subgroupOid}&subgroupGuid={subgroupGuid}>.

HTTP-метод: GET.

Возвращает список: Да.

Формат ответа: JSON.

Входные параметры:

- int subgroupOid — id подгруппы (обязательный параметр);
- char subgroupGUID — дополнительный идентификатор подгруппы в формате UUID;

Возвращаемые значения:

- int studentOid — id студента;
- int studentGid — внешний id студента;

- int studentGUID — дополнительный идентификатор студента в формате UUID;
- string fio — Ф.И.О. (полное);
- string shortFIO — Ф.И.О. (только первые буквы имени и отчества);
- string studentUID — внешний UID студента.

 *Дополнительные необязательные параметры с суффиксами UID_HR, UID_ASAP, UID_MDM, UID_SP, ID_HR_Person, UID_MDM_Person, ID_SP_DisciplineInPlan могут быть использованы в качестве дополнительных идентификаторов по объектам. При необходимости использования обратитесь к разработчику для получения информации по корректному заполнению данных полей.*

11.2. Содержание объектов импорта

Список объектов импорта находится в таблице [Объекты импорта](#)^[65].

Принятые в таблицах обозначения:

- Name — смысловое наименование поля;
- Code — кодовое наименование поля, может изменяться пользователем, но при этом должно быть полностью идентичным с соответствующими метаданными из настроечного файла [corTable.xml](#)^[64];
- Type — тип данных;
- P — идентификатор из той БД, откуда осуществляется экспорт (уникальное значение);
- M — обязательность непустого значения.

Для создания связей между таблицами допускается использовать текстовые и числовые идентификаторы. Обязательным является заполнение одного из идентификаторов ID_объекта или UID_объекта.

Для всех импортируемых объектов должны использоваться идентификаторы одного вида: целочисленные (ID_объекта) или текстовые (UID_объекта).

- ID_объекта — данные с типом long int;
- UID_объекта — данные с типом char(2000).

Табл. 9. Факультеты (Faculty)

Name	Code	Type	P	M
Аббревиатура	Abbr	char(20)		X
Код	Code	char(20)		
Наименование	Name	char(200)		X
ID факультета	ID_Faculty	long	X	X
UID факультета	UID_Faculty	char(2000)	X	X

Правила уникальности:

- Name;
- Abbr.

Табл. 10. Циклы дисциплин (CycleOfDiscipline)

Name	Code	Type	P	M
Аббревиатура	Abbr	char(20)		X
Код	Code	char(20)		
Наименование	Name	char(200)		X
Тип поколения планов	TypeOfPlanAge	long		X
ID цикла дисциплины	ID_CycleOfDiscipline	long	X	X
UID цикла дисциплины	UID_CycleOfDiscipline	char(2000)	X	X

Правила уникальности:

- TypeOfPlanAge, Name, Abbr.

Табл. 11. Виды учебной деятельности (KindOfActivity)

Name	Code	Type	P	M
Аббревиатура	Abbr	char(20)		
Код	Code	char(20)		
Наименование	Name	char(200)		X
ID вида учебной деятельности	ID_KindOfActivity	long	X	X
UID вида учебной деятельности	UID_KindOfActivity	char(2000)	X	X

Правила уникальности:

- Name;
- Abbr.

Табл. 12. Типы аудиторий (TypeOfAuditorium)

Name	Code	Type	P	M
Аббревиатура	Abbr	char(20)		X
Код	Code	char(20)		
Наименование	Name	char(200)		X
ID типа аудитории	ID_TypeOfAuditorium	long	X	X
UID типа аудитории	UID_TypeOfAuditorium	char(2000)	X	X

Правила уникальности:

- Name;
- Abbr.

Табл. 13. Формы обучения (FormOfEducation)

Name	Code	Type	P	M
Аббревиатура	Abbr	char(20)		X
Наименование	Name	char(200)		X
ID формы обучения	ID_FormOfEducation	long	X	X
UID формы обучения	UID_FormOfEducation	char(2000)	X	X

Правила уникальности:

- Name;
- Abbr.

Табл. 14. Должности (Post)

Name	Code	Type	P	M
Аббревиатура	Abbr	char(20)		
Код	Code	char(20)		
Наименование	Name	char(200)		X
ID должности	ID_Post	long	X	X
UID должности	UID_Post	char(2000)	X	X

Правила уникальности:

- Name, Code.

Табл. 15. Корпуса (Building)

Name	Code	Type	P	M
Аббревиатура	Abbr	char(20)		
Наименование	Name	char(200)		X
Филиал	Filial	char(200)		
Адрес	Address	char(200)		
Положение на карте	Position	image		
Этажность	Floor	long		
Количество лифтов	Elevator	long		
Общая площадь, кв. м.	Square	float		
Площадь ауд. фонда, кв. м.	SquareClass	float		
ID корпуса	ID_Building	long	X	X
UID корпуса	UID_Building	char(2000)	X	X

Правила уникальности:

- Name;
- Abbr.

Табл. 16. Специальности (Speciality)

Name	Code	Type	P	M
Код	Code	char(20)		
Наименование	Name	char(200)		X
ID специальности	ID_Speciality	long	X	X
UID специальности	UID_Speciality	char(2000)	X	X

Правила уникальности:

- Name, Code.

Табл. 17. Специализации (Specialization)

Name	Code	Type	P	M
Код	Code	char(20)		
Наименование	Name	char(200)		X
ID специализации	ID_Specialization	long	X	X
ID специальности	ID_Speciality	long		X
UID специализации	ID_Specialization	long	X	X
UID специальности	UID_Speciality	char(2000)		X

Правила уникальности:

- ID_Specialization (или UID_Specialization), Name.

Табл. 18. Виды работ (KindOfWork)

Name	Code	Type	P	M
Аббревиатура	Abbr	char(20)		
Код	Code	char(20)		
Наименование	Name	char(200)		X
Единица измерения ¹	Unit	tinylong		X
ID вида работы	ID_KindOfWork	long	X	X
ID вида учебной деятельности	ID_KindOfActivity	long		
Сложность ²	Complexity	tinylong		X
UID вида работы	UID_KindOfWork	char(2000)	X	X
UID вида учебной деятельности	UID_KindOfActivity	char(2000)		

¹ 0 (по умолчанию) — объем измеряется в часах, 1 — в единицах, 2 — в неделях.

² [0..100].

Правила уникальности:

- ID_KindOfActivity (или UID_KindOfActivity), Name, Abbr.

Табл. 19. Типы кафедр (ChairType)

Name	Code	Type	P	M
Наименование	Name	char(200)		X
ID типа кафедры	ID_ChairType	long	X	X
UID типа кафедры	UID_ChairType	char(2000)	X	X
Языковой доп. тип ¹	LanguageAdditionalType	tinylong		

¹ 0 - не установлено, 1 - языковая.

Правила уникальности:

- ID_ChairType (или UID_ChairType), Name.

Табл. 20. Кафедры (Chair)

Name	Code	Type	P	M
Аббревиатура	Abbr	char(20)		

Name	Code	Type	P	M
Код	Code	char(20)		
Наименование	Name	char(200)		X
ID кафедры	ID_Chair	long	X	X
ID типа кафедры	ID_ChairType	long		
ID факультета	ID_Faculty	long		
UID кафедры	UID_Chair	char(2000)	X	X
UID типа кафедры	UID_ChairType	char(2000)		
UID факультета	UID_Faculty	char(2000)		
Вид подразделения ¹	DepartmentKind	tinylong		

¹ 0 - не установлено, 1- Кафедра, 2- Отделение, 3- Департамент, 4- Сектор, 5- Центр

Правила уникальности:

– ID_Faculty (или UID_Faculty), Name.

Табл. 21. Дисциплины (Discipline)

Name	Code	Type	P	M
Аббревиатура	Abbr	char(20)		
Код	Code	char(20)		
Приоритет ¹	Priority	tinylong		X
Наименование	Name	char(200)		X
ID кафедры	ID_Chair	long		
ID дисциплины	ID_Discipline	long	X	X
ID вида дисциплины	ID_DisciplineKind	long		
ID типа дисциплины	ID_DisciplineType	long		
Сложность ²	Complexity	tinylong		X
UID кафедры	UID_Chair	char(2000)		
UID дисциплины	UID_Discipline	char(2000)	X	X
UID вида дисциплины	UID_DisciplineKind	char(2000)		
UID типа дисциплины	UID_DisciplineType	char(2000)		

¹ [0..100].

² [0..100].

Правила уникальности:

– Name.

Табл. 22. Преподаватели (Lecturer)

Name	Code	Type	P	M
Доступность ¹	Availability	tinylong		X
Приоритет	Priority	tinylong		X
ID должности	ID_Post	long		X

Name	Code	Type	P	M
ID кафедры	ID_Chair	long		X
ID преподавателя	ID_Lecturer	long	X	X
ФИО	FIO	char(200)		X
UID должности	UID_Post	char(2000)		X
UID кафедры	UID_Chair	char(2000)		X
UID преподавателя	UID_Lecturer	char(2000)	X	X
CustomUID	lecturerCustomUID	char(2000)		
Телефон	Phone	char(14)		
E-Mail	eMail	char(40)		
SMS-уведомления	SendSMS	boolean		
E-Mail-уведомления	SendEMail	boolean		
Дата начала контракта	DateContractBeg	date		
Дата окончания контракта	DateContractEnd	date		
VIP	VIP	boolean		
Табельный номер	TabNum	char(200)	X	

¹ 0 — не работает в н. в., 1 (по умолчанию) — работает, 2 — работает при необходимости.

Правила уникальности:

– FIO, ID_Chair (или UID_Chair), ID_Post (или UID_Post).

Табл. 23. Аудитории (Auditorium)

Name	Code	Type	P	M
Аббревиатура	Abbr	char(20)		
Количество (вместимость человек)	Amount	int		X
Наименование	Name	char(200)		
Номер	Number	char(20)		
ID аудитории	ID_Auditorium	long	X	X
ID кафедры	ID_Chair	long		
ID корпуса	ID_Building	long		X
ID факультета	ID_Faculty	long		
ID типа аудитории	ID_TypeOfAuditorium	long		X
UID аудитории	UID_Auditorium	char(2000)	X	X
UID кафедры	UID_Chair	char(2000)		
UID корпуса	UID_Building	char(2000)		X
UID факультета	UID_Faculty	char(2000)		
UID типа аудитории	UID_TypeOfAuditorium	char(2000)		X
Общая площадь, кв.м.	Square	float		
Дополнительное оборудование ¹	Equipment	long		

Name	Code	Type	P	M
Тип столов ²	TableType	long		
Затемнение	Darken	boolean		
Кондиционер	Conditioner	boolean		
Высота	Height	float		
Длина	Length	float		
Ширина	Width	float		
Комп. оборудование	ComputerEquipment	boolean		
Медиаоснащение	MediaFuture	boolean		
Количество компьютеров	ComputerAmount	long		
Диагональ мониторов	MonitorDiagonal	float		
Допустимость накладок ³	Overlay	boolean		
Описание	Note	char(1000)		
Этаж	Floor	int		
VIP	VIP	boolean		
Трансляция с рабочего места	BroadcastWorkPlace	boolean		

¹ 0 — нет, 1 — проектор, 2 — интерактивная доска.

² 0 — столы, 1 — парты, 2 — круглые, 3 — амфитеатр.

³ Значение true обязательно устанавливается для аудитории с наименованием "Дистанционная".

Правила уникальности:

– ID_Building (или UID_Building), Number.

Табл. 24. Группы (Group)

Name	Code	Type	P	M
Год обучения ¹	YearOfEducation	long		
Год поступления ¹	YearOfEntry	long		X
Количество студентов (план)	PlannedAmount	int		X
Наименование	Name	char(200)		
ID группы	ID_Group	long	X	X
ID кафедры	ID_Chair	long		
ID курса	ID_Course	long		X
ID факультета	ID_Faculty	long		X
ID формы обучения	ID_FormOfEducation	long		X
ID направленности/специализации	ID_Specialization	long		
ID направления/специальности	UID_Speciality	long		
UID группы	UID_Group	char(2000)	X	X
UID кафедры	UID_Chair	char(2000)		

Name	Code	Type	P	M
UID курса	UID_Course	char(2000)		X
UID факультета	UID_Faculty	char(2000)		X
UID формы обучения	UID_FormOfEducation	char(2000)		X
UID направленности/специализации	UID_Specialization	char(2000)		
UID направления/специальности	UID_Speciality	char(2000)		
Номер	Number	char(20)		X
Вид ²	Kind	long		X
Тип ³	Type	long		X
Уровень образования ⁴	KindEducation	long		
Описание	Note	char(1000)		

¹ Год задается в формате YYYY (например: 2011, 2012 и т. д.).

² 0 — академическая; 1 — учебная.

³ Заполняется, если Kind = 1; 0 — элективная; 1 — факультативная.

⁴ 0 — Бакалавриат; 1 — Магистратура; 2 — Специалитет; 3 — Аспирантура; 4 — Дополнительное профессиональное образование; 5 — Дополнительное образование; 6 — Среднее профессиональное образование.

Дополнительные правила:

- 1) YearOfEducation, YearOfEntry, ID_Course должны быть согласованы.
- 2) Если ID_Specialization (UID_Specialization) задано, то эта специализация должна входить в коллекцию Speciality.Specializations.
- 3) ID_Chair и ID_Faculty (UID_Chair и UID_Faculty) должны быть согласованы при наличии иерархической подчиненности.

Правила уникальности:

– Faculty, Course, FormOfEducation, YearOfEducation, Number.

Табл. 25. Обучающиеся (Student)

Name	Code	Type	P	M
ID группы ¹	ID_Group	long		X
ID студента	ID_Student	long	X	X
UID группы ¹	UID_Group	char(2000)		X
UID студента	UID_Student	char(2000)	X	X
УНС (уникальный номер студента)	UNS	char(20)	X	
ФИО	FIO	char(200)		X
Телефон	Phone	char(14)		
EMail	eMail	char(40)		
Флаг SMS-уведомления	SendSMS	boolean		
Флаг EMail-уведомления	SendEMail	boolean		

¹ Указывается идентификатор **основной** академической группы, в которую входит студент. Это поле всегда должно быть заполнено!

Правила уникальности:

- ID_Group (или UID_Group), FIO.

Табл. 26. Составы учебных групп (StaffEduGroup)

Name	Code	Type	P	M
ID Академической группы	ID_AcademicGroup	long	X	X
ID учебной группы	ID_EducationGroup	long		X
ID состава учебной группы	ID_StaffEduGroup	long		X
UID Академической группы	UID_AcademicGroup	char(200)		X
UID состава учебной группы	UID_EducationGroup	char(200)	X	X
UID	UID	char(200)		X

Правила уникальности:

- ID_StaffEduGroup (или UID_StaffEduGroup).

Табл. 27. Составы групп (StaffGroup)

Name	Code	Type	P	M
ID группы	ID_Group	long		X
ID состава группы	ID_StaffGroup	long	X	X
ID студента	ID_Student	long		X
UID группы	UID_Group	char(2000)		X
UID состава группы	UID_StaffGroup	char(2000)	X	X
UID студента	UID_Student	char(2000)		X

Правила уникальности:

- ID_Group, ID_Student (или UID_Group, UID_Student).

Табл. 28. Подгруппы (SubGroup)

Name	Code	Type	P	M
Аббревиатура	Abbr	char(20)		
Количество студентов (план)	PlannedAmount	int		X
Наименование	Name	char(200)		X
ID группы	ID_Group	long		X
ID подгруппы	ID_SubGroup	char(2000)	X	X
UID группы	UID_Group	char(2000)		X
UID подгруппы	UID_SubGroup	char(2000)	X	X
Языковая подгруппа	isLanguageGroup	boolean		

Правила уникальности:

- ID_Group (или UID_Group), Name.

Табл. 29. Составы подгрупп (StaffOfSubGroup)

Name	Code	Type	P	M
ID подгруппы	ID_SubGroup	long		X
ID состава подгруппы	ID_StaffOfSubGroup	long	X	X
ID студента	ID_Student	long		X
UID подгруппы	UID_SubGroup	char(2000)		X
UID состава подгруппы	UID_StaffOfSubGroup	char(2000)	X	X
UID студента	UID_Student	char(2000)		X

Правила уникальности:

- ID_SubGroup, ID_Student (или UID_SubGroup, UID_Student).

Табл. 30. Поток (Stream)

Name	Code	Type	P	M
Аббревиатура	Abbr	char(2000)		X
Количество	Amount	int		
Год обучения	YearOfEducation	long		X
Наименование	Name	char(2000)		X
ID курса	ID_Course	long		
ID потока	ID_Stream	long	X	X
ID факультета	ID_Faculty	long		
ID формы обучения	ID_FormOfEducation	long		
UID курса	UID_Course	char(2000)		
UID потока	UID_Stream	char(2000)	X	X
UID факультета	UID_Faculty	char(2000)		
UID формы обучения	UID_FormOfEducation	char(2000)		
Тип постоянства ¹	TypeOfPersistence	tinylong		X

¹ 0 — постоянный, 1 — временный (используется в содержании нагрузок и в содержании расписаний).

Правила уникальности:

- YearOfEducation, Name.

Табл. 31. Составы потоков (StaffOfStream)

Name	Code	Type	P	M
ID состава потока	ID_StaffOfStream	long	X	X
ID потока	ID_Stream	long		X
Тип контингента ¹	TypeOfContingent	int		X
ID группы ²	ID_Group	long		
ID подгруппы ³	ID_SubGroup	long		
ID студента ⁴	ID_Student	long		
UID группы ²	UID_Group	char(2000)		

Name	Code	Type	P	M
UID подгруппы ³	UID_SubGroup	char(2000)		
UID студента ⁴	UID_Student	char(2000)		

¹ 0 — студент, 1 — подгруппа, 2 — группа.

² Заполняется, если TypeOfContingent = 2.

³ Заполняется, если TypeOfContingent = 1.

⁴ Заполняется, если TypeOfContingent = 0.

Правила уникальности:

- ID_Stream, ID_Group (или UID_Stream, UID_Group);
- ID_Stream, ID_SubGroup (или UID_Stream, UID_SubGroup);
- ID_Stream, ID_Student (или UID_Stream, UID_Student).

Табл. 32. Периоды обучения (Semester)

Name	Code	Type	P	M
Аббревиатура	Abbr	char(20)		
Дата завершения	EndDate	date		X
Дата начала	StartDate	date		X
Наименование	Name	char(200)		X
(ф) Признак семестра	Flag	boolean		
ID родителя	ID_Parent	int		X
ID семестра	ID_Semester	char(2000)	X	X
UID родителя	UID_Parent	char(2000)		X
UID семестра	UID_Semester	char(2000)	X	X

Дополнительное правило: StartDate < EndDate.

Правила уникальности:

- Parent, StartDate, EndDate, Name;
- Parent, StartDate, EndDate, Abbr.

Табл. 33. Содержание нагрузки расписания (ContentOfLoad)

Name	Code	Type	P	M
Год обучения	YearOfEducation	int		
Дата начала	StartDate	date		X
Дата завершения	EndDate	date		X
Количество ак. часов ¹	Amount	float		X
Тип контингента ²	TypeOfContingent	int		X
ID потока ³	ID_Stream	long		
ID группы ⁴	ID_Group	long		
ID подгруппы ⁵	ID_SubGroup	long		
ID вида работы	ID_KindOfWork	long		X

Name	Code	Type	P	M
ID дисциплины	ID_Discipline	long		X
ID преподавателя	ID_Lecturer	long		X
ID кафедры	ID_Chair	long		
ID цикла дисциплины	ID_CycleOfDiscipline	long		
ID нагрузки расписания	ID_ContentOfLoad	long	X	X
UID потока ³	UID_Stream	char(2000)		
UID группы ⁴	UID_Group	char(2000)		
UID подгруппы ⁵	UID_SubGroup	char(2000)		
UID вида работы	UID_KindOfWork	char(2000)		X
UID дисциплины	UID_Discipline	char(2000)		X
UID преподавателя	UID_Lecturer	char(2000)		X
UID кафедры	UID_Chair	char(2000)		
UID цикла дисциплины	UID_CycleOfDiscipline	char(2000)		
UID нагрузки расписания	UID_ContentOfLoad	char(2000)	X	X
Единица измерения (ЕИ) ⁶	Unit	int		X
Количество в ЕИ	AmountUnit	int		X
Признак непрерывности ⁷	SignOfContinuity	boolean		
Номер пакета ⁸	PackageNumber	int		
ID аудитории ⁹	ID_Auditorium	long		
ID семестра	ID_Semester	long		
UID аудитории ⁹	UID_Auditorium	char(2000)		
UID семестра	UID_Semester	char(2000)		
Читается дистанционно ¹⁰	Remotely	boolean		

¹ Заполняется всегда (в академических часах).

² 1 — подгруппа, 2 — группа, 3 — поток.

³ Заполняется, если TypeOfContingent = 3.

⁴ Заполняется, если TypeOfContingent = 2.

⁵ Заполняется, если TypeOfContingent = 1.

⁶ 0 — часы (лекции, практики и т. п.), 1 — единицы (экзамен, зачет, практика и т. п.).

⁷ Значение true устанавливается для нагрузок, подразумевающих неделимость (экзамен, практика т. п.).

⁸ Устанавливается значение 0, если нагрузка непaketная, иначе — номер, одинаковый для всех нагрузок, входящих в пакет.

⁹ Обязательно указывается для пакетных нагрузок; корректность накладок аудиторий в одном пакете — это задача методиста (в РУЗ это принимается за аксиому); для непaketных нагрузок в случае необходимости может указываться несколько [предпочтительных аудиторий](#)¹⁵³.

¹⁰ Если этот флаг установлен, то в поле "ID аудитории" указывается обязательно ссылка на аудиторию "Дистанционная" (она должна быть предустановлена в этом справочнике). При импорте будет выполняться попытка установить соответствие по "ID аудитории" и по наименованию "Дистанционная".

Дополнительные правила:

- 1) $StartDate < EndDate$.
- 2) $StartDate$, $EndDate$ не могут быть шире диапазона $[Semester.StartDate, Semester.EndDate]$.
- 3) $YearOfEducation$ должен быть согласован с $Semester$.

Правила уникальности:

- $StartDate$, $ID_Discipline$, $ID_KindOfWork$, $ID_Lecturer$, ID_Stream (или $UID_Discipline$, $UID_KindOfWork$, $UID_Lecturer$, UID_Stream);
- $StartDate$, $ID_Discipline$, $ID_KindOfWork$, $ID_Lecturer$, ID_Group (или $UID_Discipline$, $UID_KindOfWork$, $UID_Lecturer$, UID_Group);
- $StartDate$, $ID_Discipline$, $ID_KindOfWork$, $ID_Lecturer$, $ID_SubGroup$ (или $UID_Discipline$, $UID_KindOfWork$, $UID_Lecturer$, $UID_SubGroup$).



Важно:

- 1) В правилах не учитываются записи, если есть поле со значением *null*.
- 2) В правилах все перечисленные поля обязательны, кроме контингентных (ID_Stream , ID_Group , $ID_SubGroup$ или UID_Stream , UID_Group , $UID_SubGroup$).
- 3) В зависимости от $TypeOfContingent$ заполняется только одно из контингентных полей.

См. [пример файла для импорта](#)⁶⁶.

Табл. 34. График недельного прохождения дисциплин ($DenormalizationOfLoad$)

Name	Code	Type	P	M
Дата завершения	EndDate	date		
Дата начала	StartDate	date		
Дискретность ¹	Discreteness	int		
Количество	Amount	int		
ID денормализации	ID_DenormalizationOfLoad	long	X	X
ID нагрузки расписания	ID_ContentOfLoad	long		X
UID денормализации	UID_DenormalizationOfLoad	char(2000)	X	X
UID нагрузки расписания	UID_ContentOfLoad	char(2000)		X

¹ Продолжительность одного непрерывного занятия; как правило, составляет 2 академических часа. При необходимости указывается другая продолжительность, но при этом:

- $Discreteness \geq 1$;
- $Discreteness \leq Amount$;
- $Amount$ делится без остатка на $Discreteness$.

Правила уникальности:

- $ID_ContentOfLoad$ (или $UID_ContentOfLoad$), $StartDate$.

График недельного прохождения дисциплин — разбиение объема нагрузки (ContentOfLoad.Amount) по календарным неделям периода реализации нагрузки (ContentOfLoad.StartDate .. ContentOfLoad.EndDate).

Требования согласованности:

- 1) $\text{Sum}(\text{DenormalizationOfLoad.ContentOfLoad.Amount}) = \text{ContentOfLoad.Amount}$.
- 2) Отрезки идут последовательно неделя за неделей.
- 3) Каждая неделя — отрезок с понедельника по воскресенье.
- 4) Перекрытие отрезков не допускается.
- 5) Отрезки не выходят за период реализации нагрузки ContentOfLoad.StartDate .. ContentOfLoad.EndDate.

Табл. 35. Предпочтительные аудитории (AuditoriumPreferred)

Name	Code	Type	P	M
Приоритет ¹	Priority	int		
ID предпочтительной аудитории	ID_AuditoriumPreferred	long	X	X
ID аудитории	ID_Auditorium	long		X
ID нагрузки расписания	ID_ContentOfLoad	long		X
UID предпочтительной аудитории	UID_AuditoriumPreferred	char(2000)	X	X
UID аудитории	UID_Auditorium	char(2000)		X
UID нагрузки расписания	UID_ContentOfLoad	char(2000)		X

¹ [0..10]; чем больше значение, тем выше приоритет.

Правила уникальности:

- ID_ContentOfLoad, ID_Auditorium (или UID_ContentOfLoad, UID_Auditorium).

Табл. 36. Учебные планы (Curriculum)

Name	Code	Type	P	M
(е) Уровень образования ¹	KindEducation	int		X
Год обучения	YearOfEducation	int		X
Магистр, программа	MasterProgramm	char(200)		
Магистр, специализация	MasterSpecialization	char(200)		
Наименование	Name	char(200)		X
ID курса	ID_Course	long		X
ID специализации	ID_Specialization	long		
ID специальности	ID_Speciality	long		X
ID учебного плана	ID_Curriculum	long	X	X
ID факультета	ID_Faculty	long		X
UID курса	UID_Course	char(2000)		X
UID специализации	UID_Specialization	char(2000)		
UID специальности	UID_Speciality	char(2000)		X
UID учебного плана	UID_Curriculum	char(2000)	X	X

Name	Code	Type	P	M
UID факультета	UID_Faculty	char(2000)		X

¹ 0 — Бакалавриат, 1 — Магистратура, 2 — Специалитет, 3 — Аспирантура, 4 — Дополнительное профессиональное образование, 5 — Дополнительное образование, 6 — Среднее профессиональное образование.

Дополнительное правило: если ID_Specialization задано, то эта специализация должна входить в коллекцию Speciality.Specializations.

Правила уникальности: YearOfEducation, ID_Faculty (или UID_Faculty), ID_Course (или UID_Course), Name.

Табл. 37. Расстояние между зданиями (*DistanceBetweenBuilding*)

Name	Code	Type	P	M
ID корпуса до	ID_Building2	long		X
ID корпуса от	ID_Building1	long		X
ID удаленности корпуса	ID_DistanceBetweenBuilding	long	X	X
Удаленность (мин)	Distance	int		X
UID корпуса до	UID_Building2	char(2000)		X
UID корпуса от	UID_Building1	char(2000)		X
UID удаленности корпуса	UID_DistanceBetweenBuilding	char(2000)	X	X

Табл. 38. Виды дисциплин (*DisciplineKind*)

Name	Code	Type	P	M
ID Вида дисциплины	ID_DisciplineKind	long		X
Наименование	Name	char(200)		
UID	UID	char(200)		X

Табл. 39. Типы дисциплин (*DisciplineType*)

Name	Code	Type	P	M
ID Типа дисциплины	ID_DisciplineType	long		X
Наименование	Name	char(200)		
UID	UID	char(200)		X

Табл. 40. Аспекты преподавания языковых дисциплин (*AspectOfTeachingLangDis*)

Name	Code	Type	P	M
ID Аспекта преподавания	ID_AspectOfTeachingLangDis	long		X
Наименование	Name	char(200)		
UID	UID	char(200)		X

Табл. 41. Состав аспектов преподавания (*StaffOfAspectOfTeachingLangDis*)

Name	Code	Type	P	M
ID Состав аспектов преподавания	ID_StaffOfAspectOfTeachingLangDis	long		X
ID преподавателя	ID_Lecturer	long	X	X

Name	Code	Type	P	M
UID преподавателя	UID_Lecturer	char(200)	X	
ID Аспекта преподавания	ID_AspectOfTeachingLangDis	long		X
UID Аспекта преподавания	UID_AspectOfTeachingLangDis	char(200)		
UID	UID	char(200)		X

Табл. 42. Преподаваемые дисциплины (StaffOfDisciplineLecturer)

Name	Code	Type	P	M
ID Преподаваемые дисциплины	ID_StaffOfDisciplineLecturer	long		X
ID преподавателя	ID_Lecturer	long	X	X
UID преподавателя	UID_Lecturer	char(200)	X	
ID дисциплины	ID_Discipline	long	X	
UID дисциплины	UID_Discipline	char(200)		X
UID	UID	char(200)		X

Табл. 43. Преподаваемые языки (StaffOfLanguageLecturer)

Name	Code	Type	P	M
ID Преподаваемые языки	ID_StaffOfLanguageLecturer	long		X
ID преподавателя	ID_Lecturer	long	X	X
UID преподавателя	UID_Lecturer	char(200)	X	
ID языка ¹	ID_Language	long	X	
UID языка	UID_Language	char(200)		X
UID	UID	char(200)		X

¹ 1 — русский (по умолчанию), 2 — английский.

Табл. 44. Ресурсы на языках (ResourceLanguage)

Name	Code	Type	P	M
ID ресурса на языке	ID_ResourceLanguage	long	X	X
ID языка ¹	ID_Language	long		X
Объект ресурса ²	ObjectApply	int		X
ID объекта ресурса ³	ID_ObjectOID	long		X
Аббревиатура	Abbr	char(80)		
Наименование	Name	char(200)		X

¹ 1 — русский (по умолчанию), 2 — английский.

² 1 — Discipline (дисциплина), 2 — Lecturer (преподаватель), 3 — Post (должность), 4 — KindOfWork (Вид работы), 5 — Building (здание), 6 — Auditorium (аудитория), 7 — Stream (поток), 8 — Group (группа), 9 — Subgroup (подгруппа).

³ Содержит ID_Discipline для ObjectApply = 1, ID_Lecturer для ObjectApply = 2 и т. д.

💡 В *ResourceLanguage* необходимо заносить наименования/аббревиатуры ресурсов только на иностранных языках, т.к. на базовом (русском) языке эти значения содержатся в основных сущностях.

Табл. 45. Языки подгрупп (StaffOfLanguageSubGroup)

Name	Code	Type	P	M
ID языка подгруппы	ID_StaffOfLanguageSubGroup	long	X	X
ID подгруппы	ID_SubGroup	long		X
UID подгруппы	UID_SubGroup	char(200)	X	
ID языка	Language	long		X
UID языка	UID_Language	char(200)	X	
Количество студентов	StudCount	int		X
Приоритет языка	LangOrder	int		X