
Документация

Выпуск 6.6.7

Руководство разработчика XSQUARE - XRAD 6.6.7

июл. 10, 2026

1	Предисловие	1
1.1	Для кого это руководство	1
1.2	Требования к разработчику	1
2	Быстрый старт	2
2.1	Знакомство с XRAD	2
2.2	Что такое XRAD?	2
2.3	Как работает XRAD?	3
3	Основные концепции	4
3.1	Концепция приложения	4
3.2	Рендеринг страницы и обработка ввода	5
3.3	Сессии и их состояние	5
3.4	Управление значениями сессии	6
4	Настройки	7
4.1	Основные	7
4.2	Источники данных	9
4.3	Глобальные переменные	10
4.4	Глобальные процессы	12
4.5	Схемы авторизации	15
4.6	Схемы аутентификации	17
5	Страницы	20
5.1	Создание новой страницы	21
5.2	Параметры страницы	21
5.3	Работа с папками	22
5.4	Редактор страниц	23
5.5	Модальная страница	25
5.6	Безопасность страниц	27
5.7	События	28
6	Регионы	31
6.1	Параметры регионов	33
6.2	Расположение компонентов на странице	34
6.3	Расположение компонентов на сетке	35

7	Формы	37
7.1	Создание формы	37
7.2	Элементы ввода	38
7.3	Кнопки	44
8	Компоненты	49
8.1	Обертка	50
8.2	Хлебные крошки	50
8.3	Контейнер для кнопок	51
8.4	Карточки	51
8.5	График	53
8.6	Чат	57
8.7	Форма	58
8.8	HTML-код (HTML)	58
8.9	Навигационная панель	59
8.10	Таблица	59
8.11	Отчет	63
8.12	Вкладки	67
8.13	Дерево	68
8.14	Визард (Wizard)	69
8.15	Календарь	70
8.16	DataGrid	72
8.17	Плитки	78
8.18	Канбан	81
9	Процессинг	85
9.1	Процессы	86
9.2	Условия выполнения процесса	88
9.3	Выполнение по запросу (Request)	88
9.4	Управление транзакцией	88
9.5	Закрытие диалога	88
9.6	Ветви	89
9.7	Валидация	90
9.8	Типы валидаторов	91
9.9	Ошибка валидации	93
9.10	Системные валидаторы	93
9.11	Условия для выполнения валидации	93
9.12	Пропуск валидаций.	94
9.13	Вызовы REST	94
9.14	Процесс DataGrid	95
9.15	Загрузки	96
9.16	Схемы аутентификации	97
9.17	Схемы авторизации	98
10	Работа со списками	99
10.1	Что такое список	99
10.2	Создание, редактирование, удаление списков	99
10.3	Создание элементов списка	101
11	Управление пользователями	104
11.1	Создание, редактирование, удаление пользователей	104

Руководство разработчика XSQUARE - XRAD описывает как использовать среду разработки XRAD для построения веб приложений.

1.1 Для кого это руководство

Руководство предназначено для разработчиков, целью которых является построение веб-приложения с датацентрической архитектурой (основа - база данных) с помощью инструментов разработки XSQUARE RAD, далее XRAD. В руководстве описано как использовать инструменты для построения, отладки, управления и развертывания приложения.

1.2 Требования к разработчику

Для разработки приложений необходимы базовые знания концепции работы реляционных баз данных, SQL, основы HTML и JavaScript (js).

В данном разделе Вы познакомитесь с общей концепцией XRAD. Данный раздел описывает основные компоненты и возможности среды разработки.

2.1 Знакомство с XRAD

XRAD обеспечивает разработчика всеми инструментами для построение приложения в единой, расширяемой платформе на основе нескольких возможных СУБД: Postgresql, Mysql, Mssql, Oracle, Firebird.

2.2 Что такое XRAD?

В современном мире для разработки веб-приложения часто требуется наличие нескольких разработчиков, каждый из которых отвечал бы за отдельные компоненты системы.

Обычно веб-приложение разделяют на следующие компоненты:

- пользовательский интерфейс (frontend)
- компонент для взаимодействия с базой данных (backend)
- связующий компонент, который отвечает за различную логику приложения (middleware)

Все эти компоненты взаимозависимы друг от друга и без четко определенных границ ответственности могут нарушать логику работы приложения. Данные границы и зоны ответственности определяет архитектор приложения, человек от которого требуется высокий уровень компетенции в понимании работы каждого из компонентов. Не стоит также забывать про администрирование такой системы и поддержание работоспособности, что требует наличия системного администратора.

Наличие такой команды однозначно увеличивает цену и уменьшает скорость разработки приложения.

Платформа XRAD предлагает совершенно другой подход к разработке.

XRAD - это платформа для быстрой разработки приложений, разработанная ООО «Хи-Квадрат» (Xsquare Rapid Application Development).

XRAD - является декларативной средой для разработки и развертывания веб-приложений, ориентированных на базы данных. Благодаря встроенным функциям, таким как темы пользовательского интерфейса, элементы управления навигацией, обработчики форм и гибкие отчеты, XRAD значительно ускоряет процесс разработки приложений.

2.3 Как работает XRAD?

XRAD использует стандартную 3-уровневую архитектуру, в которой запросы отправляются из браузера через сервер приложений в базу данных. Вся обработка, манипулирование данными и бизнес-логика выполняются в базе данных. После того, как база данных обработает код, результаты будут переданы веб-сервером в виде структуры JSON, на основе которой веб-приложение отобразит страницу.

Эта архитектура гарантирует доступ к данным практически с нулевой задержкой, высочайшую производительность и горизонтальную масштабируемость «из коробки».

Для эффективной работы в XRAD разработчики должны понимать некоторые ключевые понятия:

- как происходит управление дизайном пользовательского интерфейса;
- как происходят процессы обработки и отображения страниц;
- что такое сессии, транзакции и как с ними работать.

3.1 Концепция приложения

Что такое приложение?

Веб-приложение, которое разрабатывается с помощью XRAD - это HTML интерфейс, который существует поверх объектов базы данных: таблиц и процедур. Приложение логически и физически разделяется на две базы данных: база данных, отвечающая за бизнес-логику (APP_DB) и база данных с набором метаданных интерфейса приложения (XRAD_DB). Сервер приложений PGHS - объединяет данные двух баз и предоставляет конечному пользователю-веб интерфейс, который отображает бизнес-данные и процессы в соответствии с тем, как было описано разработчиком. Один экземпляр PGHS может обращаться к неограниченному количеству баз данных APP_DB и только к одной базе данных XRAD_DB.

Для конечного пользователя приложение - это набор страниц, которые отображают и позволяют вводить данные посредством различных компонентов.

Что такое страница?

Страница - это «строительный блок» приложения. Приложение может содержать всего одну, но обычно множество таких блоков. Каждая страница может содержать множество как простых элементов - кнопки и поля для ввода, так и более сложные - разнообразные отчеты, графики, таблицы. Элементы на странице группируются с помощью специального контейнера - региона. Страницы могут включать в себя логику обработки данных, называемую процессами. Связь страниц друг с другом обеспечивают ссылки или ветви.

3.2 Рендеринг страницы и обработка ввода

Для просмотра страниц разрабатываемого приложения необходимо вызвать страницу приложения XRAD по ссылке, на которой размещается приложение. Когда вы запускаете приложение, XRAD вызывает два ключевых процесса:

showPage - процесс рендеринга страницы, который собирает все атрибуты страницы (включая регионы, кнопки и элементы) в один управляющий файл json. После чего среда исполнения передаёт эту управляющую структуру в веб-браузер клиента, на котором происходит отрисовка описанных элементов приложения. За отрисовку компонентов на клиенте отвечает клиентская среда исполнения XRAD. Когда вы запрашиваете страницу, переходя по ссылке - XRAD вызывает процесс showPage.

processPage - процесс обработки страницы. Отвечает за обработку введенных данных, включая в себя выполнения процессов, проверок данных и переходов на другие страницы. Процесс вызывается после отправки страницы на сервер, путем нажатия кнопки или вызова соответствующего метода jsAPI. После отправки все данные указанные в элементах страницы записываются в сессию и подставляются в соответствующие процессы.

Транзакции

При вызове процессов showPage или processPage инициируется транзакция в БД с бизнес-данными. Эта транзакция существует на протяжении всего времени выполнения процессов showPage или processPage. В случае успешного выполнения **всех** процессов транзакция завершается вызовом commit, за исключением случая, когда на уровне процесс установлен атрибут **Фиксация ТХ**. Если возникает ошибка в процессах обработки данных - транзакция будет завершена вызовом rollback, что отменит все изменения на момент начала выполнения процесса. В связи с этим разработчику необходимо учитывать, что данные, которые будут переданы в базу данных посредством выполнения того или иного процесса будут доступны только при успешном завершении транзакции. Доступность данных в момент выполнения той или иной процедуры полностью зависит от уровня изолированности транзакции, настроенного в БД с бизнес-данными. По умолчанию уровень изолированности в PostgreSQL установлен в Read committed.

В связи с ограничениями, связанными с подтранзакциями в PostgreSQL, вызов commit в бизнес-процессе приведет к ошибке. Для обеспечения выполнения отдельного процесса вне зависимости от успешности выполнения обработки всей страницы предлагается в настройках процесса указывать вызов commit после его выполнения.

Значения сессии переданные в ходе вызова процессов showPage или processPage, также будут записаны в БД только после успешного выполнения обработки страницы. Однако их значения доступны в ходе выполнения этих процессов. Для передачи значений элементов ввода в бизнес-процессы см. Процессинг страницы.

3.3 Сессии и их состояние

Что такое сессия

Сессия — это период взаимодействия пользователя с приложением. Она начинается, когда пользователь обращается к приложению, и заканчивается, когда пользователь закрывает сессию, выходит из системы или происходит определённый период бездействия (таймаут). Во время сессии приложение хранит информацию о пользователе и его действиях, что позволяет поддерживать контекст взаимодействия и персонализировать отображаемую информацию.

Каждой сессии присваивается уникальный идентификатор. XRAD использует этот идентификатор для хранения и извлечения рабочего набора данных приложения до и после каждого просмотра страницы. Поскольку сессии полностью независимы друг от друга, в базе данных может существовать любое количество сессий одновременно. Пользователь также может запускать несколько экземпляров приложения одновременно в разных браузерах.

Значение идентификатора сессии хранится в cookie браузера. Время жизни данного cookie определяется настройками приложения и дополнительно контролируются средой исполнения PGHS.

Сессии логически и физически отличаются от сессий базы данных, используемых для обслуживания запросов страниц. Пользователь запускает приложение в одном сеансе XRAD от входа до выхода с типичной продолжительностью, измеряемой в минутах или часах. Каждая страница, запрошенная во время сессии, может инициировать создание новой сессии базы данных или использовать повторно открытые сессии базы данных для доступа к ресурсам базы данных. Часто такие сессии базы данных длятся всего доли секунды.

Что такое состояние сессии

Состояние сессии - это механизм который позволяет разработчикам хранить и получать значения для пользователя, даже когда он переходит по разным страницам приложения.

Протокол передачи гипертекста (HTTP), протокол, по которому чаще всего доставляются HTML-страницы, является протоколом без сохранения состояния. Веб-браузер подключается к серверу только на время, необходимое для загрузки полной страницы. Каждый запрос страницы обрабатывается сервером как независимое событие, не связанное с какими-либо запросами страницы, которые произошли ранее или могут произойти в будущем. Чтобы получить доступ к значениям формы, введенным на одной странице, на следующей странице, значения должны быть сохранены в состояние сессии. XRAD предоставляет разработчикам возможность получать и устанавливать значения состояния сессии с любой страницы приложения.

3.4 Управление значениями сессии

При создании интерактивных веб-приложений, управляемых данными, возможность доступа к значениям состояния сеанса и управления ими имеет решающее значение. В XRAD состояние сессии автоматически управляется для каждой страницы, и на него легко ссылаться в статическом HTML или логических элементах управления, таких как процессы или проверки.

Раздел «Настройки» среды разработки содержит:

1. Основные – главные настройки разрабатываемого приложения.
2. Источники данных – список БД, которые могут использоваться в качестве источников данных для приложения.
3. Глобальные переменные – список глобальных переменных.
4. Системные переменные - список глобальных системных переменных, доступных разработчику.
5. Глобальные процессы – список глобальных процессов приложения.
6. Схемы авторизации – список доступных при разработке схем для авторизации пользователей.
7. Схемы аутентификации – список доступных при разработке схем для аутентификации пользователей.
8. Лицензия - основные сведения о лицензии.

4.1 Основные

Основные настройки уровня приложения являются обязательными для заполнения и делятся на следующие группы:

4.1.1 Основные

Параметр	Описание
Наименование приложения	Название приложения. Выводится в заголовке страницы.
Короткое наименование	Отображаемое название приложения в среде разработки.
URL приложения	URL адрес приложения. Используется для осуществления перехода из среды разработки.
Домашняя страница	Основная страница, на которую попадает авторизованный пользователь. В поле ввода необходимо ввести, либо выбрать из списка номер страницы, которая будет использоваться в качестве домашней.
Страница входа	Основная публичная страница, на которую попадает неавторизованный пользователь. Как правило, страница авторизации. В поле ввода необходимо ввести, либо выбрать из списка номер страницы, которая будет использоваться в качестве страницы входа.

4.1.2 Списки

В этом блоке параметров определяются списки уровня приложения.

Параметр	Описание
Панель навигации	Определяет список для меню в заголовке страницы. Выберите из перечня заранее созданных списков в соответствующем разделе.
Меню навигации	Определяет список для основного меню в левом блоке страницы. Выберите из перечня заранее созданных списков в соответствующем разделе.

4.1.3 Сессия

В этом блоке определяются параметры пользовательских сессий.

Параметр	Описание
Время жизни сессии	Определяет время жизни пользовательской сессии, т.е. период в единицах времени, в течение которого активна пользовательская сессия взаимодействия с приложением.
Время жизни сессии (ед. изм.)	Определяет единицу измерения для указанного в параметре «Время жизни сессии» числа. На выбор 4 варианта: секунды, минуты, часы и дни.
Время жизни неактивной сессии	Определяет период, в единицах времени, по истечению которого будет удалена пользовательская сессия, если пользователь не производит никаких действий в приложении.
Время жизни неактивной сессии (ед. изм.)	Определяет единицу измерения для указанного в параметре «Время жизни неактивной сессии» числа. На выбор 4 варианта: секунды, минуты, часы и дни.
Ограничение одной сессии на пользователя	Данный параметр позволяет назначить пользователю эксклюзивную сессию при авторизации в приложении. При получении эксклюзивной сессии, другие сессии данного пользователя будут сразу же завершены.

4.1.4 Статичные ресурсы

В этом блоке определяются внешние статичные ресурсы, которые позволяют изменить отображение пользовательского интерфейса приложения.

Параметр	Описание
CSS файлы	Список файлов .css, содержащие таблицы стилей. Каждая строка определяет относительный либо абсолютный путь, на внутренний или внешний ресурс.
JS файлы	Список файлов .js, содержащие код JavaScript. Каждая строка определяет относительный либо абсолютный путь, на внутренний или внешний ресурс.
Сервер отчетв	Адрес сервера отчётов (xReports) указывается при наличии лицензии. Используется интерактивными компонентами (Отчёт, DataGrid) для выгрузки данных в формате <code>xlsx</code> .

4.1.5 Контроль версий

В этом блоке определяются настройки контроля версий для `xrad`.

Параметр	Описание
URL внешнего репозитория	Укажите URL внешнего репозитория (<code>remote</code>) для совершения операций по синхронизации кода.

4.2 Источники данных

Источники данных – один из важнейших компонентов XRAD, предоставляющий возможность горизонтально масштабировать разрабатываемое приложение. Приложение позволяет создать неограниченное количество источников данных. Источником данных выступает БД, данные которой используются для отображения компонентов приложения. Указывая различные источники данных, разработчик может объединить вывод данных из разных баз на одной странице.

Для того чтобы добавить новый источник данных укажите его в настройках приложения. В настройках указывается только название источника, которое будет использовано для подстановки в соответствующие поля компонентов редактора страниц, а также его тип.

На текущий момент платформа `xrad` поддерживает 5 типов СУБД: PostgreSQL, Oracle, Firebird, Mssql, Mysql.

После добавления источника данных в справочник источников, вы можете добавить собственную логику, которая будет вызываться при инициализации транзакции к источнику и при её завершении. Данный механизм подойдет тем, кто желает выставлять дополнительные параметры подключения, к примеру `ALTER SESSION` для Oracle, либо `set_config` для PostgreSQL.

Внимание! Изменения вышеуказанных полей вступят в силу только после перезагрузки сервисов `pghs` и `xrad`.

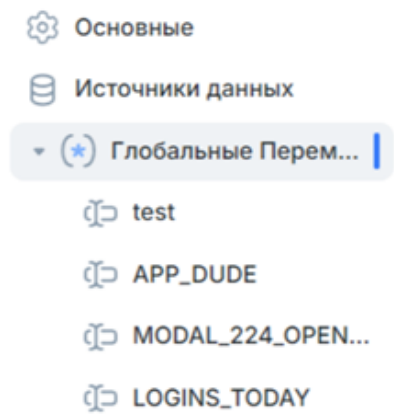
После завершения настроек источника данных в приложении, вам необходимо дополнить соответствующие конфигурационные настройки PGHS и XRAD - `config.json`. При этом наименование источника в конфигурационном файле должно соответствовать введенному в настройках приложения.

Для вступления изменений в силу необходимо перезагрузить сервисы PGHS и XRAD.

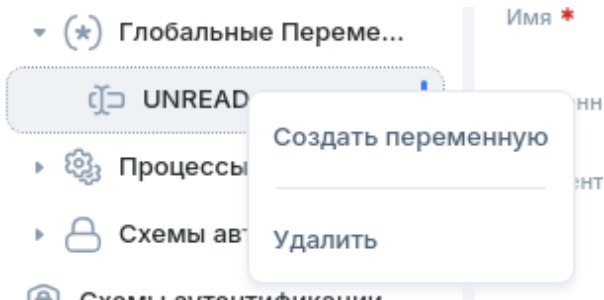
4.3 Глобальные переменные

В этом блоке настроек определяются глобальные переменные приложения.

Глобальные переменные - это переменные, которые доступны и повсеместно используются в приложении для хранения какого-либо значения.



Объявление новой глобальной переменной осуществляется по нажатию ПКМ по списку «Глобальные переменные», либо по любому элементу данного списка.



При создании глобальной переменной необходимо заполнить ее атрибуты:

Параметр	Описание
Имя	единственный обязательный атрибут, определяющий имя создаваемой переменной.
Временная	флаг, который определяет, будет ли переменная временной. Временная переменная не хранится в состоянии сессии и доступа только на время выполнения процессов.
Комментарий	текстовый комментарий.

Глобальная переменная UNREAD_NOTIFICATIONS

Имя *	UNREAD_NOTIFICATIONS
Временная	☐
Комментарий	Непрочитанные уведомления (Глобальные процессы)

После нажатия кнопки «Сохранить» переменная будет занесена в базу данных XRAD и доступна для просмотра, редактирования и удаления. Аналогично созданию происходит редактирование существующих переменных.

Для удаления глобальной переменной необходимо нажать ПКМ по необходимой переменной в списке и выбрать из меню «Удалить».

4.3.1 Предустановленные переменные

Предустановленные переменные - это переменные, которые присутствуют в каждом приложении на базе платформы XRAD. Данные переменные не подлежат редактированию и удалению.

Зарезервированные глобальные переменные:

Временные:

- **REQUEST** - код запроса для выполнения необходимого процесса. Данный параметр определяет какой процесс или группа процессов должны быть выполнены.
- **PAGE** - номер страницы.
- **RESPONSE** - используется для указания выходного параметр при выполнении асинхронных процессов (AJAX).
- **G01 - G10** - встроенные переменный для передачи значений в процессы. Можно использовать при программном вызове процессов с помощью js.
- **CGI_FORWARDED_FOR** - переменная, которая хранит значение заголовка X-Forwarded-For, переданного веб сервером.
- **CGI_FORWARDED_IP** - переменная, которая хранит значение заголовка X-Forwarded-IP, переданного веб сервером.
- **CGI_REAL_IP** - переменная, которая хранит значение заголовка X-Real-IP, переданного веб сервером.

Постоянные:

- **APP_USER** - id авторизованного на данный момент пользователя;
- **SESSION** - уникальный код сессии;
- **APP_USER_GROUPS** - доменные группы, авторизованного на данный момент пользователя.
 - Строка с разделителями в виде ; (точка с запятой);
- **AFTER_LOGIN_URL** - используется для переадресации пользователя на указанную страницу, после выполнения процесса авторизации.

4.4 Глобальные процессы

В этом блоке настроек определяются глобальные процессы для всего приложения.

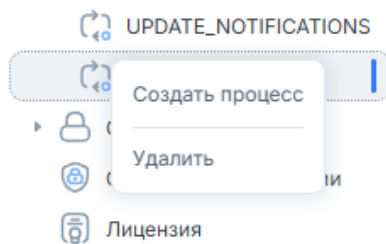
Процесс приложения – это процесс XRAD не имеющий привязки к какой-либо странице. Такие процессы, также как и процессы на страницах, выполняются по запросу от приложения (Request).

4.4.1 Порядок выполнения процессов приложения

При появлении запроса от приложения система отбирает все удовлетворяющие условиям запроса процессы (глобальные и страницные), сортирует в соответствии с их порядковыми номерами и выполняет.

4.4.2 Управление процессами

Для создания или удаления процесса необходимо вызвать контекстное меню, нажав ПКМ на списке процессов и выбрать соответствующее действие.



После заполнения или редактирования параметров процесса необходимо подтвердить внесенные изменения, нажав кнопку «Сохранить».

Процесс UPDATE_NOTIFICATIONS

Основные ▾

Имя *	UPDATE_NOTIFICATIONS
Порядковый номер *	1
Источник данных *	DEFAULT_APP ▾
Тип *	До загрузки страницы ▾
Коммит после выполнения	☐
Текст ошибки	
Sql код *	<pre> 1 SELECT 2 COUNT(1) 3 FROM 4 events.t_notifications </pre>

Элементы ▾

Ввод	
Вывод	UNREAD_NOTIFICATIONS

Условия ▾

Тип Условия *	Always ▾
---------------	----------

Параметры процесса делятся на 3 группы:

1. Основные

Параметр	Описание
Имя	обязательный атрибут, определяющий имя процесса.
Имя запроса	Строка, где указывается значение переменной REQUEST, которое должно быть передано для выполнения запроса.
Порядковый номер	обязательный атрибут, определяющий очередность выполнения процесса.
Источник данных	обязательный атрибут, определяющий в каком источнике данных будет выполнен процесс.
Именованные параметры	Флаг, который указывает необходимость воспринимать входящие параметры как часть текста sql запроса. При установке в false, разработчику потребуются явно указать входящие параметры. Отображается для источника данных отличных от типа Oracle.
Тип	обязательный атрибут, определяющий момент выполнения процесса: • До загрузки страницы • Процессинг • Обратный вызов AJAX
Коммит после выполнения	флаг, определяющий необходимость завершения транзакции
Текст ошибки	Строка с текстом, которая будет отображена, если процесс завершится с ошибкой.
Sql код	Обязательный атрибут, определяющий код SQL-запроса.

2. Элементы

Параметр	Описание
Ввод	Наименования переменных через запятую, содержащих входные параметры. Отображается если признак «Именованные параметры» не установлен.
Вывод	Наименования переменных через запятую, куда будут записаны результаты

3. Условия

Тип условия – определяет условие при котором будет выполнен процесс, действие или событие.

Типы условий

Тип условия – определяет условие при котором будет выполнен процесс, действие или событие.

4.4.3 Типы условий

Разработчику доступно 26 типов условий:

1. Always - выполняется всегда;
2. Exists (SQL query returns at least one row) - позволяет выполнить проверочный SQL-запрос. Условие считается выполненным, если запрос вернул хотя бы 1 строку;
3. NOT Exists (SQL query returns no rows) - позволяет выполнить проверочный SQL-запрос. Условие считается выполненным, если запрос не вернул ни одной строки;
4. Value of Item / Column in Expression 1 Is NOT NULL - встроенное условие на заполненность поля или ячейки таблицы. Условие считается выполненным, если поле ввода или ячейка таблицы содержит какое-либо значение;

5. Value of Item / Column in Expression 1 != Zero - встроенное условие на значение отличное от 0 (нуль) поля ввода или столбца таблицы. Условие считается выполненным, если объект проверки будет содержать значение отличное от 0 (нуль);
6. Value of Item / Column in Expression 1 Is NULL - встроенное условие на значение NULL (пустое значение) поля ввода или столбца таблицы. Условие считается выполненным, если объект проверки содержит NULL (пустое значение);
7. Value of Item / Column in Expression 1 Is NULL or Zero - встроенное условие на значение NULL (пустое значение) или 0 (нуль) поля ввода или столбца таблицы. Условие считается выполненным, если объект проверки содержит NULL (пустое значение) или равно 0 (нуль);
8. Value of Item / Column in Expression 1 = Zero - встроенное условие на значение 0 (нуль) поля ввода или столбца таблицы. Условие считается выполненным, если объект проверки будет содержать значение 0 (нуль);
9. Value of Item / Column in Expression 1 Is NOT null and the Item / Column Is NOT Zero – встроенное условие на заполненность и значение отличное от 0 (нуль) поля ввода или столбца таблицы. Условие считается выполненным, если объект проверки не содержит NULL (пустое значение) и не равно 0 (нуль);
10. SQL Expression - позволяет выполнить SQL-запрос, возвращающий TRUE или FALSE. В поле ввода запро-са указывается только тело запроса, без ключевых слов SELECT и/или FROM. Если необходимо выполнить какой-то подзапрос, то его необходимо обернуть в “(” “)”. Условие считается выполненным, если запрос вернул TRUE;
11. Request = Value - проверяет значение системной глобальной переменной REQUEST на введенное значение. Условие выполняется если значение равняется указанному
12. Request != Value - проверяет значение системной глобальной переменной REQUEST на введенное значение. Условие выполняется если значение не равняется указанному.
13. Request is contained in Value - проверяет значение системной глобальной переменной REQUEST на введенное значение. Условие выполняется если значение содержится в указанной строке (LIKE).
14. Request is NOT contained in Value - проверяет значение системной глобальной переменной REQUEST на введенное значение. Условие выполняется если значение не содержится в указанной строке (NOT LIKE).
15. Item = Value - проверяет соответствие значения элемента ввода на указанное. Условие выполняется если значения соответствуют.
16. Item != Value - проверяет соответствие значения элемента ввода на указанное. Условие выполняется если значения не соответствуют.
17. Item is numeric - проверяет значение элемента ввода. Условие выполняется если значение содержит только цифры или равно null. Общие разделители десятичных и разрядов игнорируются.
18. Item is NOT numeric - проверяет значение элемента ввода. Условие выполняется если значение не является числовым или равно null. Общие разделители десятичных и разрядов игнорируются.
19. Item is alphanumeric - проверяет значение элемента ввода. Условие выполняется если значение содержит только цифры и буквы или равно null.
20. Item is in colon delimited list - проверяет значение элемента ввода. Условие выполняется если значение соответствуют одному из введенных вариантов. Варианты вводятся с помощью разделителя двоеточие.
21. Item is not in colon delimited list - проверяет значение элемента ввода. Условие выполняется если значение не соответствуют ни одному из введенных вариантов. Варианты вводятся с помощью разделителя двоеточие.

22. User is Authenticated - проверяет статус аутентификации пользователя. Условие выполняется если пользователь прошел процесс аутентификации.
23. User Is Public User - проверяет статус аутентификации пользователя. Условие выполняется если пользователь не прошел процесс аутентификации.
24. PL/SQL Expression - выполняет условие на языке PL/SQL. Доступно только для элементов с источником данных типа Oracle. Условие выполняется если выражение возвращает значение TRUE.
25. PL/SQL Function Body - выполняет тело функции на языке PL/SQL. Доступно только для элементов с источником данных типа Oracle. Условие выполняется если функция возвращает значение TRUE. Exists (SQL query returns at least one row) - позволяет выполнить проверочный SQL-запрос. Условие считается выполненным, если запрос вернул хотя бы 1 строку;
26. Never - не выполнять никогда.

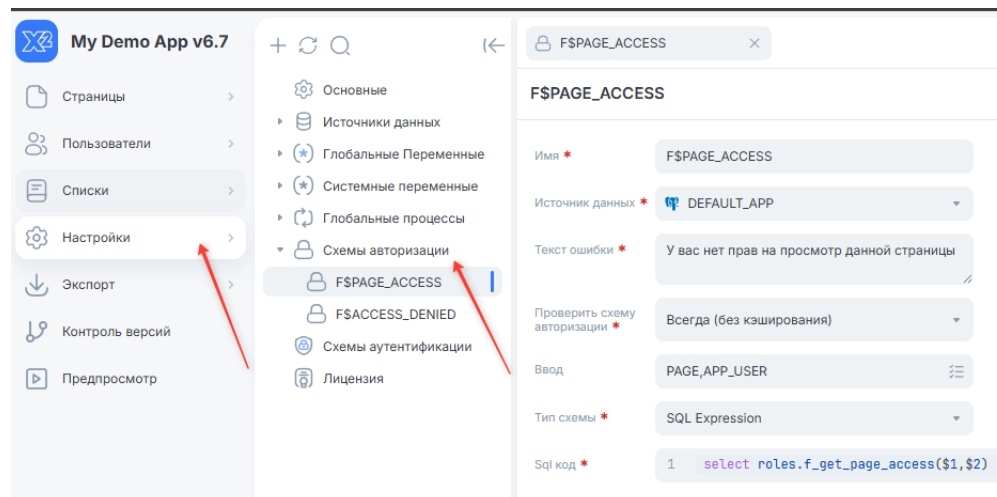
4.5 Схемы авторизации

4.5.1 Общее описание

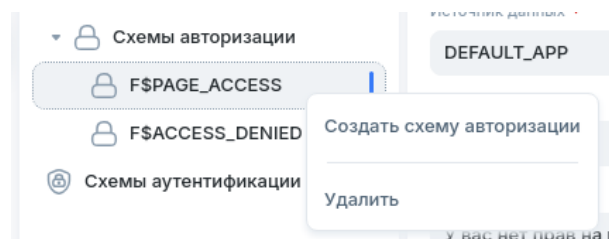
Схема авторизации определяет доступность страницы или компонента страницы конкретному пользователю. Если авторизация проходит успешно, то пользователю отображается запрашиваемая страница или компонент. Если пользователь не проходит авторизацию, то при открытии страницы ему будет выведена ошибка, либо если авторизация проверяется на компоненте, он не будет отображен на странице.

4.5.2 Создание схемы авторизации

Список схем авторизации располагается в разделе «Настройки» -> «Схемы авторизации»:



При нажатии ПКМ на элементах списка «Схемы Авторизации» будет отображено контекстное меню с кнопками для создания новой схемы и удаления выделенной:



Примечание: для перехода на страницу необходимо авторизоваться от лица аккаунта с ролью доступа выше VIEW.

Создание новой схемы авторизации осуществляется по нажатию кнопки «Создать». Откроется форма создания схемы авторизации:

Имя *	FSPAGE_ACCES
Источник данных *	DEFAULT_APP
Ввод	PAGE_APP_USER
Текст ошибки *	У вас нет прав на просмотр данной страницы
Sql код *	1 select roles.f_get_page_access(\$

Атрибуты:

- **Имя** - обязательный атрибут, определяющий имя создаваемой схемы авторизации.
- **Источник данных** - обязательный атрибут, определяющий источник данных для создаваемой схемы авторизации.
- **Именованные переменные** - флаг, который указывает необходимость воспринимать входящие параметры как часть текста sql запроса. При установке в false, разработчику потребуется явно указать входящие параметры. Отображается для источника данных отличных от типа Oracle.
- **Текст ошибки** - обязательный атрибут, определяющий текст ошибки авторизации. Будет отображен в случае, когда авторизация проверяется на уровне страницы.
- **Проверять схему авторизации** - обязательный параметр. Выберите как часто необходимо выполнять указанную авторизацию. Доступны следующие варианты:
 - **Один раз за сессию** - авторизация будет выполнена всего один раз за сессию пользователя. Результат выполнения будет помещен в кэш и будет переиспользован в качестве результата выполнения для других компонентов системы.
 - **Один раз за просмотр страницы** - авторизация будет выполнена в первый раз при рендеринге или обработке страницы. Результат выполнения будет помещен в кэш и будет переиспользован для других компонентов. После выполнения процесса рендеринга или обработки страницы результат будет удален из кэша.
 - **Один раз на компонент** - авторизация будет выполнена один раз для заданного компонента. Результат выполнения будет помещен в кэш и будет переиспользован для этого компонента на все время жизни сессии.
 - **Всегда (без кэширования)** - процесс авторизации будет выполняться всегда.
- **Тип схемы** - Укажите тип выполняемого кода. Доступны следующие варианты:
 - **SQL Expression** - введите логическое выражение на языке sql.
 - **Exists** - условие проверяет наличие возвращаемых строк в sql запросе. Если строки возвращаются - условие считается успешным.
 - **Not Exists** - условие проверяет отсутствие возвращаемых строк в sql запросе. Если строки не возвращаются - условие считается успешным.

- **PL/SQL Function Body** - введите тело анонимной функции на языке PL/SQL. Функция должна возвращать значение типа Boolean.
- **SQL Код** - введите выражение на языке sql, соответствующее выбранному типу схемы.
- **Ввод** - атрибут, определяющий список входных параметров для авторизации. Указывается если не установлен флаг «именованные переменные», либо выбран источник данных типа Oracle.

После нажатия кнопки «Сохранить» и подтверждения введенных данных схема авторизации будет занесена в базу данных и доступна для просмотра, редактирования и удаления.

4.5.3 Редактирование схемы авторизации

Для редактирования схемы авторизации выберите нужную вам схему. После нажатия на схему откроется форма, аналогичная форме создания новой схемы авторизации (см. выше «Создание схемы авторизации»), с информацией о выбранной схеме.

Отредактируйте необходимые вам атрибуты и сохраните их новые значения, нажав кнопку «Сохранить».

Для удаления схемы нажмите ПКМ по схеме, нажмите в меню кнопку «Удалить» и подтвердите действие.

4.6 Схемы аутентификации

4.6.1 Общие сведения

Аутентификация - это процесс проверки подлинности каждого пользователя, который получает доступ к приложению.

Процесс проверки подлинности требует, чтобы пользователь предоставил определенные учетные данные, например, имя пользователя и пароль. Если учетные данные проходят проверку, пользователь получает доступ к приложению, если нет - доступ будет запрещен.

После успешной аутентификации пользователя устанавливается значение глобальной переменной APP_USER (см. Глобальные переменные). Когда пользователь переходит со страницы на страницу, значение APP_USER используется для идентификации пользователя.

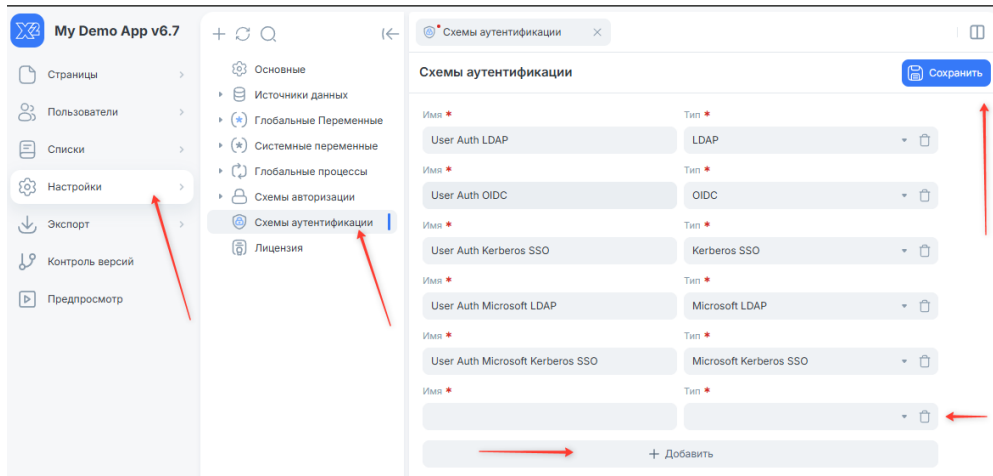
Сервер приложений PGHS поддерживает аутентификацию и авторизацию с использованием следующих схем:

- По логину и паролю - CUSTOM
- Microsoft LDAP
- Microsoft Kerberos SSO
- LDAP
- Kerberos SSO
- Open ID Connect

При разработке приложения в конструкторе XRAD разработчик определяет возможные схемы аутентификации посредством указания имени схемы и ее типа. Таким образом в приложении может быть множество схем одного типа, что позволяет гибко настраивать аутентификацию пользователей. Например, можно реализовать аутентификацию пользователей разных доменов Active Directory.

4.6.2 Создание схемы аутентификации

Страница со схемами аутентификации располагается во вкладке «Настройки» -> «Схемы аутентификации», по нажатию кнопки «Добавить» можно добавить новую схему. Также любую схему можно удалить, нажав кнопку с иконкой корзины, для применения всех изменений необходимо нажать кнопку «Сохранить». Непосредственно параметры каждой схемы аутентификации задаются в файле `auth_config.json` сервера приложений PGHS, с описанием данного файла и параметров можно ознакомиться в документации PGHS.

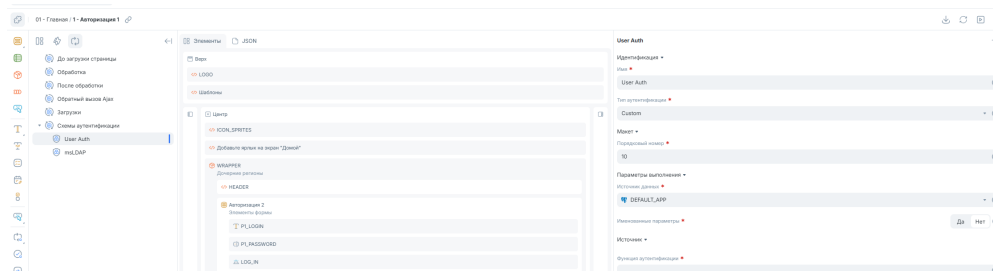


Примечание: для перехода на страницу необходимо авторизоваться от лица аккаунта с ролью доступа выше VIEW.

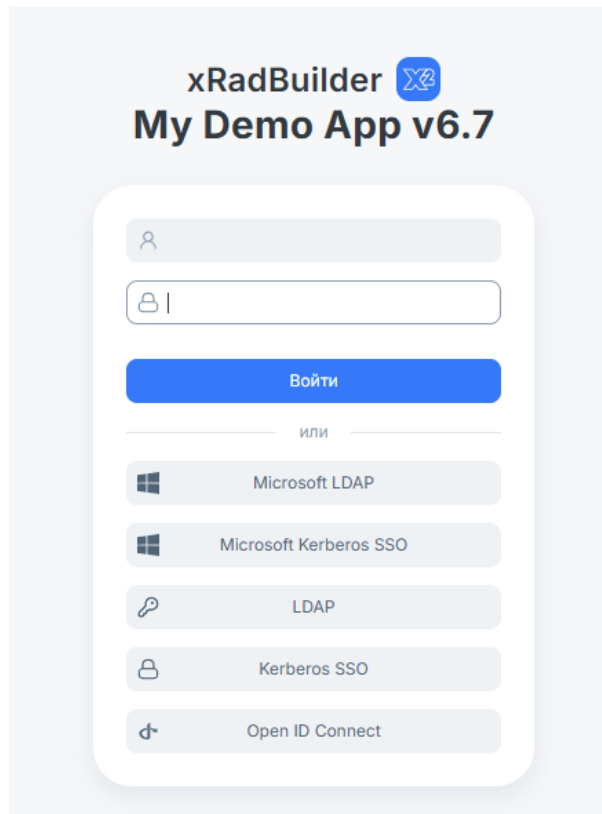
При старте сервер приложений PGHS загружает схемы из файла `auth_config.json` и сопоставляет их по имени и типу с параметрами в БД XRAD.

Для каждой страницы аутентификации разрабатываемого приложения можно создать свой набор способов аутентификации:

Пример 1:

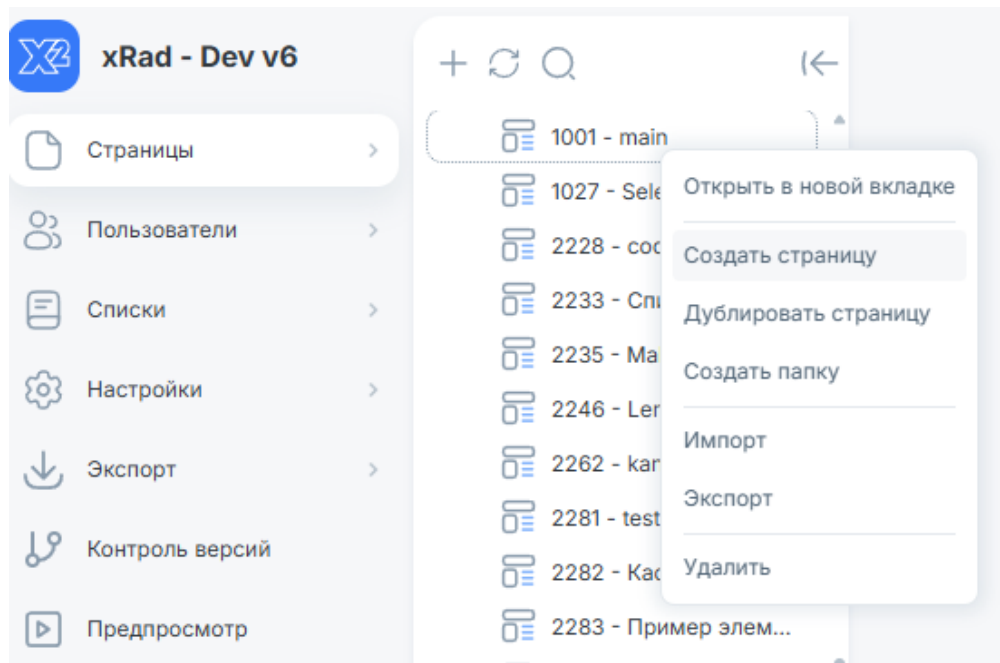


Пример 2:



Страницы

Страница — это основной блок разрабатываемого веб-приложения. Для того, чтобы отобразить список доступных страниц необходимо выбрать в главном меню раздел «Страницы». Все возможные действия со страницами доступны из контекстного меню, вызываемого по нажатию ПКМ. Разработчику доступно создание новой страницы, дублирование, экспорт и удаление текущей страницы, а также создание папок для группировки страниц.



5.1 Создание новой страницы

Создать новую страницу можно с помощью контекстного меню или с помощью кнопки «+». В окне создания страницы необходимо указать атрибуты новой страницы:

- Имя - строковое наименование страницы
- Номер – целочисленный идентификатор страницы
- Вид страницы - шаблон представления страницы
- Папка - выбор папки куда будет помещена страница.

Создать страницу

Имя *	New page
Номер *	555
Вид страницы *	Стандартный
Папка	Стандартный Модальное окно Логин Минималистичный

Доступны следующие виды страниц:

- Стандартный - страница с боковым меню и панелью навигации в заголовке страницы.
- Модальное окно – страница, которая отображается поверх родительской (вызывающей страницы).
- Логин - страница для аутентификации
- Минималистичный – простая страница без бокового меню и с заголовком без панели навигации.

После создания страницы - откроется «Редактор страниц».

5.2 Параметры страницы

Для каждой создаваемой страницы доступен определенный набор параметров:

- Имя – обязательный атрибут, определяющий наименование страницы.
- Папка - название папки, в которой расположена страница.
- Вид страницы – обязательный атрибут, определяющий шаблон представления страницы. Доступны следующие варианты шаблонов: стандартный, модальное окно, логин, минималистичный (описание в разделе создание страницы)
- URL-адреса файлов JS – список URL-адресов файлов JavaScript с кодом, который будет загружен для данной страницы.
- Объявление функций и глобальных переменных JS – используется для определения JS переменных и функций на уровне страницы.
- URL-адреса файлов CSS – список URL-адресов файлов CSS, который будет загружен для данной страницы.
- Встроенный CSS - используется для определения стилей на уровне страницы.
- Ширина - Определяет ширину модального окна. Доступно только для режима страницы «Модальное окно».

- Схема авторизации - определяет доступность страницы конкретному пользователю (См. раздел «Схемы авторизации»).
- Публичная страница - определяет необходимость аутентификации пользователя (См. раздел «Схемы аутентификации»).
- Защита страницы – определяет необходимость защиты параметров в запросе страницы с помощью контрольной суммы.

New page →|

Идентификация ▾

Имя *

New page ?

Папка

01 - Главная ▾ × ≡

Параметры отображения ▾

Вид страницы *

Модальное окно ▾ ?

Параметры окна ▾

Ширина

JavaScript ▾

URL-адреса файлов

Добавить ?

Объявление функций и глобальных переменных JS

Добавить ?

CSS ▾

URL-адреса файлов

Добавить ?

Встроенный CSS

Добавить ?

Безопасность ▾

Схема авторизации

Публичная страница Да Нет ?

Защита страницы * Да Нет ?

5.3 Работа с папками

Папки являются логическим элементом приложения для группировки страниц в модули или блоки. Создать новую папку можно с помощью контекстного меню или с помощью кнопки «+», расположенной вверху панели со списком страниц. Создание вложенной папки возможно только с помощью контекстного меню, которое вызывается по нажатию правой кнопки мыши на родительскую папку.

В окне создания папки необходимо указать наименование, которое должны быть уникально в рамках выбранного уровня вложенности и не должно содержать в имени специального знака «/». Поместить страницу в папку можно с помощью механизма Drag&Drop (перемещение мышкой), либо указав папку в атрибутах страницы. Также с помощью механизма Drag&Drop возможно переместить папку в другую папку.

5.4 Редактор страниц

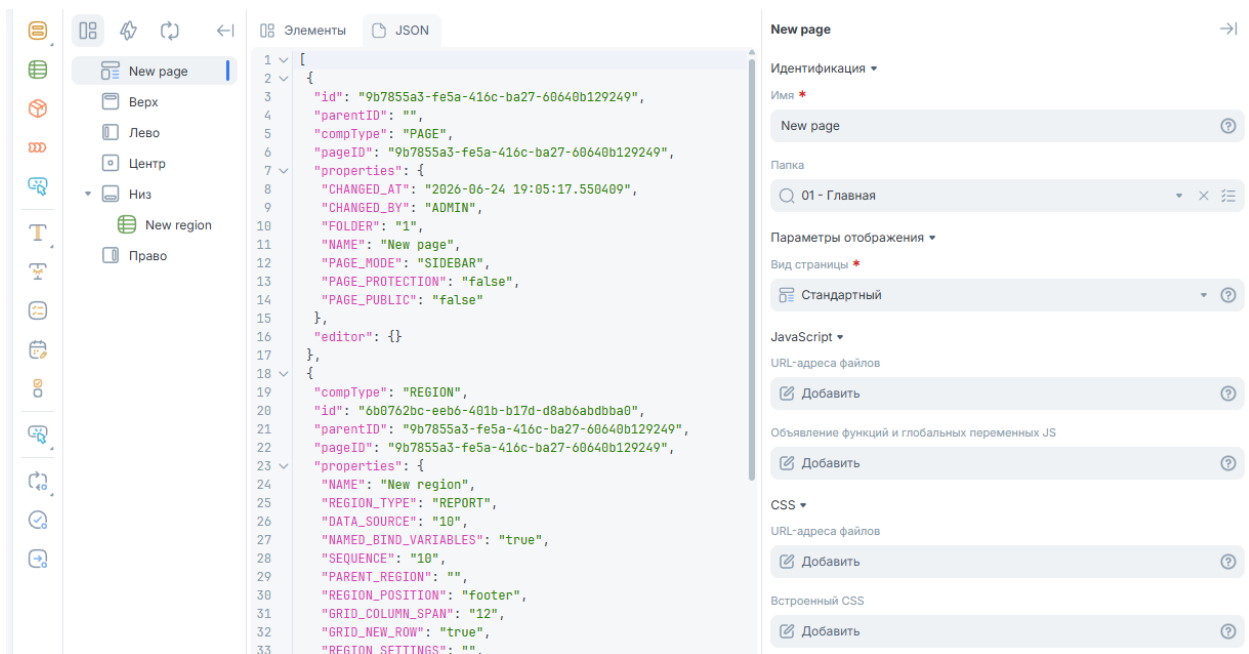
Вся работа со страницей происходит в «Редакторе страниц».

В верхней части Редактора страниц – расположена панель инструментов, где отображается идентификатор и имя страницы, кнопка копирования адреса страницы в буфер обмена, а также кнопки для экспорта, обновления, предпросмотра, блокировки и сохранения изменений страницы.



Ниже панели инструментов располагаются четыре вертикальные области Редактора страниц, логически повторяющие концепцию всей среды разработки:

- Палитра компонентов
- Дерево компонентов страницы
- Область визуального редактора
- Параметры выбранного компонента.



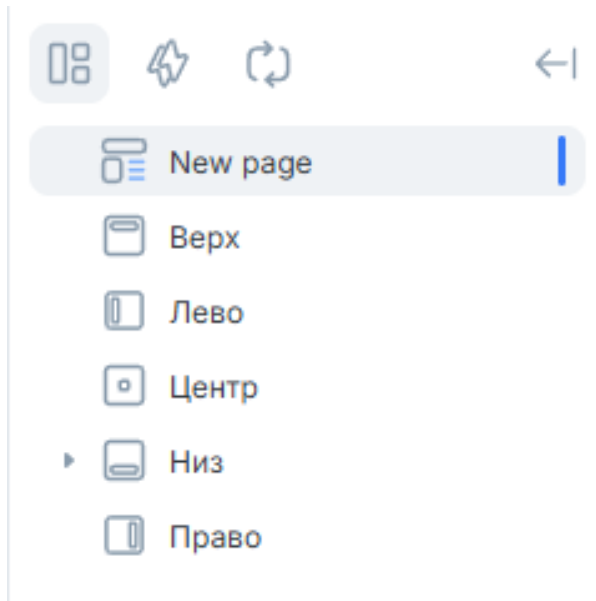
5.4.1 Главное меню редактора страниц

Главное меню состоит из 3-х вкладок:

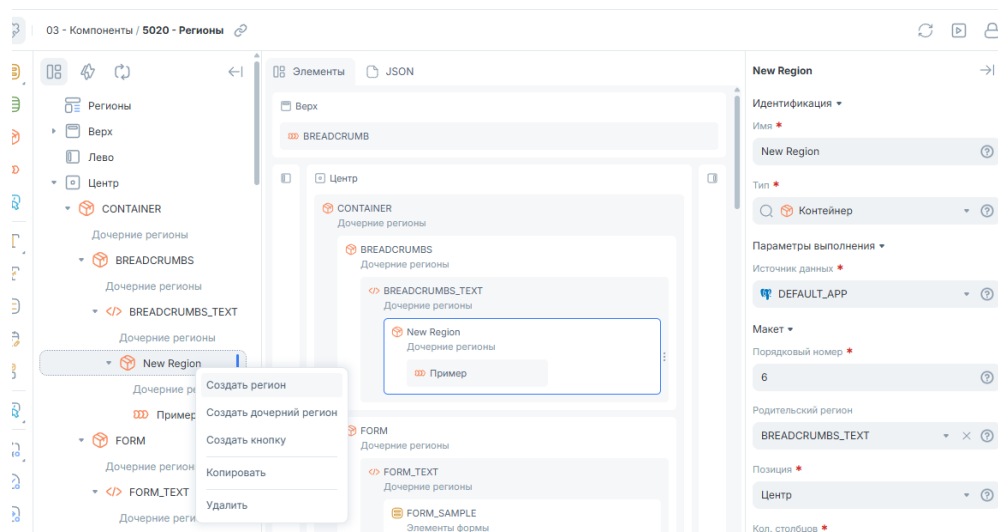
- Элементы
- События
- Процессинг

Элементы

На вкладке Элементы отображается дерево компонентов, из которых состоит страница - регионы, элементы формы и кнопки.



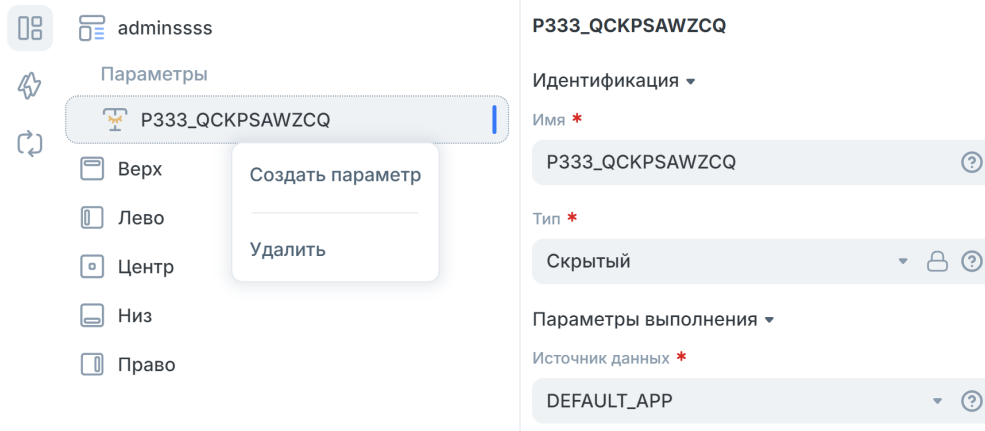
Компоненты могут быть размещены в определенной области страницы (Верх, Лево, Центр, Низ, Право) или могут быть вложены в регионы в определенной позиции региона. Базовая конструкция, которая создается из контекстного меню – это регион. Регион может включать дочерние регионы, кнопки и другие визуальные компоненты, которые определяются типом региона.



Если не выбрана область страницы, то по умолчанию элементы создаются в центральной области (Центр).

Примечание: кнопки могут быть созданы только внутри региона.

Также из контекстного меню имени страницы доступно создание параметров страницы. Данные параметры в отличие от глобальных параметров приложения доступны только в пределах конкретной страницы.



5.5 Модальная страница

Модальная страница представляет собой дочернее окно поверх родительского, расположенное в том же окне браузера. Модальная страница остается активной до тех пор, пока пользователь не завершит работу с ней и не закроет ее. Пользователь не может взаимодействовать с родительской частью страницы, до тех пор, пока модальная страница не будет закрыта.

5.5.1 Вызов модальной страницы

Вызвать модальную страницу возможно 2 способами:

- По ссылке
- Из кода Javascript

Пример вызова модальной страницы с помощью JavaScript:

```
jsAPI.modal.open({
  page: {
    src: '1010',
    query: {
      P1010_ID: jsAPI.getItem('P1000_ID'),
      clear: '1010'
    }
  },
  title: 'Моя модальная страница'
});
```

либо

```
jsAPI.modal.open({page: '/content/1010/?P1010_ID=123&CR=1010'});
```

Если страница защищена чексуммой необходимо использовать второй способ. Чек сумма будет предоставлена автоматически к строке вызова.

JavaScript-код вызова можно использовать как значение ссылки через действие Перенаправление на страницу, так и через событие типа Click на выбранную кнопку.

5.5.2 Закрытие модальной страницы (событие «Отмена»)

Закрытие модальной страницы выполняется через Javascript код, выполняющийся на текущей модальной странице:

```
jsAPI.modal.close()
```

5.5.3 Обновление данных на родительской форме

Для того, чтобы обработать закрытие модальной страницы по необходимому событию необходимо:

Создать **Событие** типа **Dialog Closed** (при закрытии через событие “Отмена”) или **Dialog Accept** (при закрытии через процесс) на элемент, с которого вызывается модальная страница и указать там свойство Код JS.

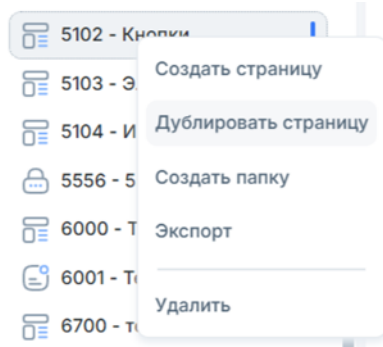
Обработку результата выполнения модальной страницы можно также определить при вызове модальной страницы в блоках onClose, onAccept или onDecline:

```
jsAPI.modal.open({
  page: {
    src: '1010',
    query: {
      P1010_ID: jsAPI.getItem('P1000_ID'),
      clear: '1010'
    }
  },
  title: 'Моя модальная страница'
}, {
  onAccept: function(e, i) {
    jsAPI.setItem('P1000_ID', i.P1010_ID);
  },
  onClose: function(e, i) {
    alert(3);
  }
});
```

5.5.4 Дублирование, удаление, группировка и экспорт страницы

Помимо создания страницы из контекстного меню доступного по ПКМ в списке страниц доступны следующие действия:

- Дублировать страницу – создает копию выбранной страницы. Для новой страницы необходимо указать новое имя, порядковый номер и вид.
- Создать папку – создает папку с заданным именем, которая используется для группировки страниц.
- Экспорт – экспортирует страницу в виде файла *.sql с SQL-кодом.
- Удалить – удаляет выбранную страницу.



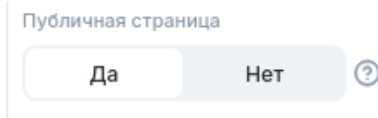
5.6 Безопасность страниц

Далее будут рассмотрены механизмы для обеспечения безопасности и защиты страниц.

5.6.1 Публичная страница

Публичная страница – это страница, для которой не требуется аутентификация и авторизация пользователя.

Установить атрибут публичности можно в настройках страницы через свойство «Публичная страница».



Если свойство «Публичная страница» не установлено или установлено в состояние «Нет», то при запросе страницы отобразится страница для аутентификации пользователя.

Данное свойство будет применено к странице после сохранения изменений через кнопку «Сохранить».

5.6.2 Контрольные суммы

Контрольная сумма используется в случае, когда необходимо обеспечить защиту параметров страницы в URL от изменения. Контрольная сумма формируется для каждой ссылки для конкретного пользователя. Изменение параметров в URL вызовет ошибку проверки контрольной суммы.

Если страница защищена с помощью контрольной суммы, в URL появляется дополнительный GET-параметр «cs» вида:

```
&cs=4825bc2e8de876595d5a36bcc89891b415e2575f45833cc5df786c000e24308
```

Контрольная сумма состоит из:

- набора параметров страницы;
- специальных наборов символов, которые называются salt (соль). Соль формируется для каждой сессии пользователя.

Если пользователь вручную укажет ссылку на страницу, которая защищена контрольной суммой, система распознает эту ссылку и добавит к ней контрольную сумму.

Защита страницы через контрольную сумму настраивается отдельно для каждой страницы.

Примечание

Если страница защищена контрольной суммой, процессы загрузки файлов будут также защищены контрольной суммой.

Защита страницы *

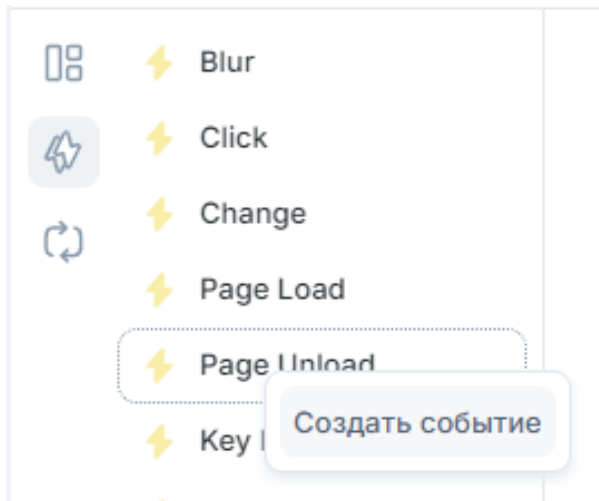
Да

Нет



5.7 События

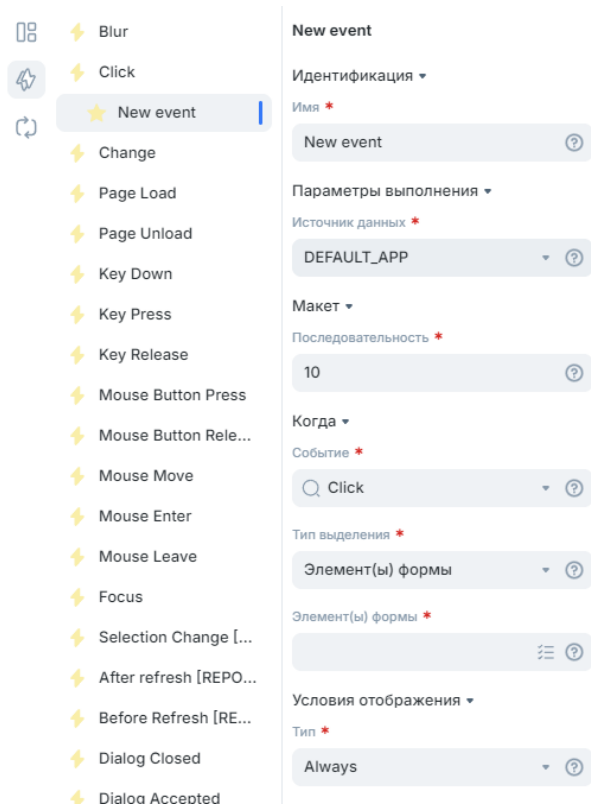
На вкладке **События** отображается список событий и действий, сгруппированных по типам возможных событий. Для создания нового события Разработчику необходимо вызвать контекстное меню с помощью ПКМ.



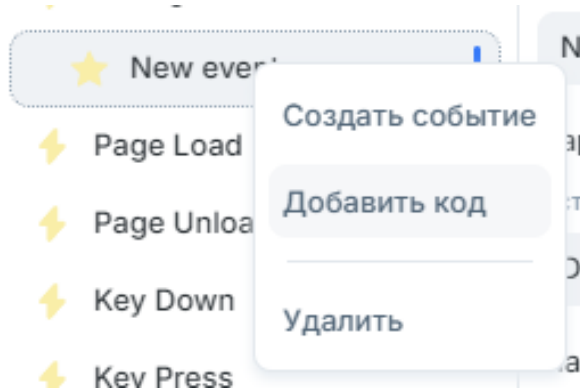
После создания события, необходимо настроить его параметры в области параметров:

- Имя – обязательный атрибут, определяющий наименование события.
- Источник данных – обязательный атрибут, определяющий источник данных для хранения элемента списка.
- Именованные параметры - флаг, который указывает необходимость воспринимать входящие параметры как часть текста sql запроса. При установке в false, разработчику потребуется явно указать входящие параметры. Отображается для источника данных отличных от типа Oracle.
- Последовательность – обязательный атрибут, порядковый номер в соответствии с которым действие будет выполнено.
- Событие – обязательный атрибут, определяющий тип события.
- Тип выделения – обязательный атрибут, определяющий тип элемента страницы для срабатывания события.
- Тип условия отображения – обязательный атрибут, определяющий тип условия, которое должно быть выполнено для того, чтобы это событие могло быть вызвано.

Для применения изменений необходимо сохранить изменения через кнопку в панели инструментов.



К событию можно привязать Действие в виде JavaScript кода. Создание действия осуществляется из контекстного меню по кнопке «Добавить код».

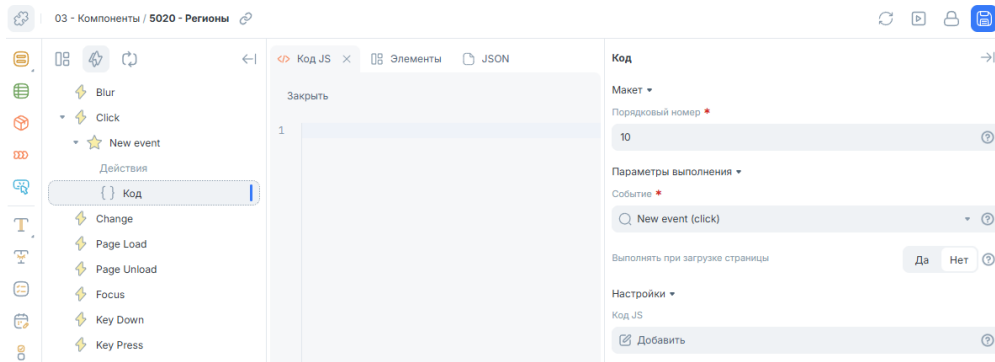


Для создаваемого действия необходимо также определить параметры:

- Последовательность – обязательный атрибут, порядковый номер в соответствии с которым действие будет выполнено.
- Событие – обязательный атрибут, определяет имя события, к которому привязывается код действия.
- Выполнять при загрузке страницы – определяет выполнение действия при загрузке страницы.
- Код JS – обязательный атрибут, определяет JavaScript код, который будет выполнен при наступлении события.

По нажатию кнопки «Добавить» в области вкладок редактора будет открыта вкладка «Код JS» с интерактивным редактором кода. Сохранение изменений JS кода производится в редакторе кода, а

сохранение параметров с помощью кнопки в панели инструментов.



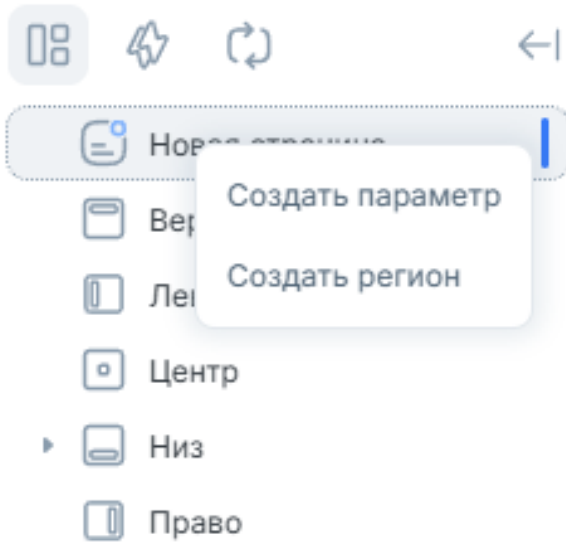
Визуальные компоненты - ключевые элементы пользовательского интерфейса PGHS. Большинство страниц веб-приложений состоит из множества компонентов, которые разработчик группирует в контейнеры.

В XRAD таким универсальным контейнером является **регион**. Регионы могут содержать в себе поля для ввода, кнопки и другие регионы, либо непосредственно выводить данные в виде отчётов, графиков и других способов предоставления информации.

Важно

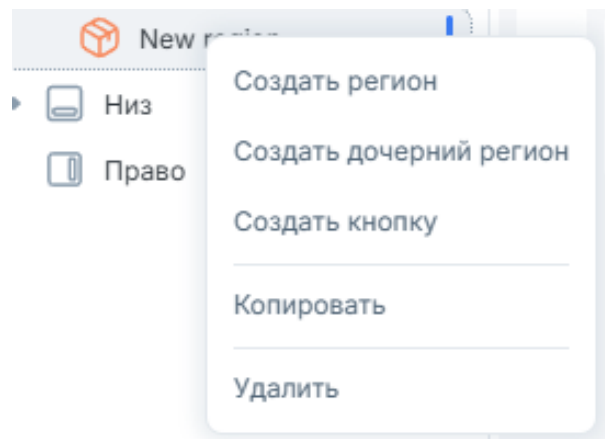
Регион - базовый блок из которого состоят страницы XRAD. Регион представляет из себя область страницы, которая служит контейнером для данных.

Новые регионы создаются через контекстное меню по ПКМ.



Свойства регионов:

1. Имеют типы, которые определяют внешний вид содержимого страницы. Настраивая регионы, разработчики могут вывести на страницу различные компоненты: HTML текст, таблицы, формы, графики и пр.
2. Могут быть родительскими, дочерними или независимыми по отношению друг к другу.
3. Внутри региона может быть создано множество других регионов – дочерних, с вложенностью без ограничения, а также кнопки.



6.1 Параметры регионов

После создания региона необходимо настроить его параметры.

В таблице данного раздела перечислены все параметры регионов. В зависимости от вида региона, их набор может различаться.

Параметр	Тип	Описание
Имя	Текст	Имя региона. Определяет отображение региона в дереве объектов и настройках других компонентов, а также выводится в заголовке региона на странице приложения.
Тип	Список	Тип региона. Определяет набор параметров и наличие особых атрибутов региона, отображение региона на странице приложения.
Источник данных	Список	Название источника данных для данного компонента.
Именованные параметры	Переключатель	Флаг, который указывает необходимость воспринимать входящие параметры как часть текста sql запроса. При установке в false, разработчику потребуется явно указать входящие параметры. Отображается для источника данных отличных от типа Oracle, Mssql.
Последовательность	Число	Порядковый номер региона. Определяет положение региона на сетке внутри области расположения / родительского региона.
Родительский регион	Список	В списке на выбор - страница и регионы, не являющиеся дочерними для данного. В случае выбора страницы, регион будет отображен внутри области расположения, иначе внутри выбранного региона.
Позиция	Список	Выбор области расположения региона.
Количество столбцов	Число	Выбор Automatic, либо число 1-12. Указывает количество ячеек сетки, занимаемых регионом. По умолчанию 12.
Начать новую строку	Переключатель	Определяет положение на сетке по вертикали.
Настройки темы	Окно настроек	Настройка темы региона.
Классы компонентов	Текст	Классы дочерних компонентов внутри региона.
Классы css	Текст	Классы css региона.
Показать заголовков	Переключатель	Определяет, будет ли отображено имя региона в заголовке и визуальная область заголовка (важно для позиции кнопок - "в заголовке").
Складной	Переключатель	Определяет, будет ли регион сворачиваемым.
Развернут	Переключатель	Определяет состояние региона при открытии страницы. Если выбрано, то регион будет отображаться в развернутом состоянии. Параметр доступен только если "Складной" = "да".
Запомнить состояние	Переключатель	Определяет сохранение выбранной пользователем вкладки. При входе на страницу будет отображаться последняя выбранная пользователем вкладка. Параметр доступен только если "Складной" = "да".
Статический идентификатор	Текст	Задаёт идентификатор для региона для дальнейшего обращения к нему из jsAPI.

продолжается на следующей странице

Таблица 1 – продолжение с предыдущей страницы

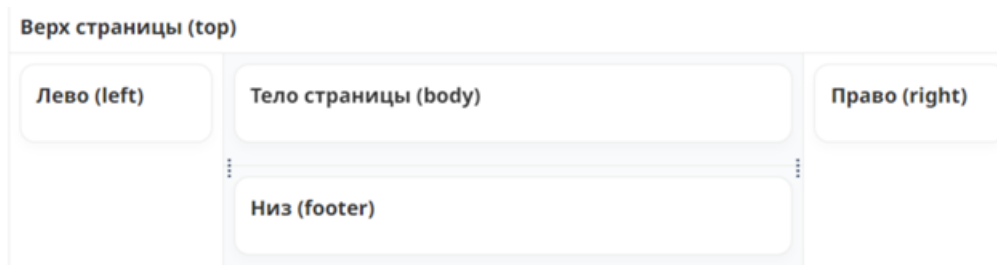
Параметр	Тип	Описание
Источник - Тип	Список	Определяет тип источника данных. Применим для регионов типа HTML. На выбор два варианта: • Статический. В поле исходный текст потребуется ввести текст HTML. • SQL. В поле SQL потребуется ввести запрос к БД.
Источник – HTML код	Область текста	HTML-код. Применим для регионов типа HTML с выбранным типом источника - Статический.
Источник - SQL	Область текста	Запрос к БД. Формат вывода определяется типом региона и его атрибутами. Параметр применим для следующих видов регионов: HTML (с выбранным типом источника SQL) • REPORT • TREE • CHAT • CHART • CALENDAR • DATAGRID
Источник - Исходный вход	Текст / Окно конструктора	Определяет список входных параметров для запроса SQL в поле источник - SQL. Применим для видов регионов, для которых существует параметр Источник - SQL (см. строку выше). Для каждой переменной подстановки в запросе должен быть определён входящий параметр. Может принимать значения глобальных переменных и элементов ввода и выбора страницы. Можно ввести как текстом, так и выбрать в конструкторе.
Источник - список	Список	Привязывает список к региону. На выбор будут представлены все созданные в приложении списки. Параметр применим для следующих видов регионов: • BREADCRUMB • PAGE NAVIGATION • CARDS • WIZARD • TILES
Условия отображения - Тип	Список	Определяет тип условия отображения региона на странице. Подробнее см. в <i>условиях выполнения</i> .

6.2 Расположение компонентов на странице

Для региона обязательным параметром является положение на странице.

В XRAD представлен выбор из пяти вариантов расположения:

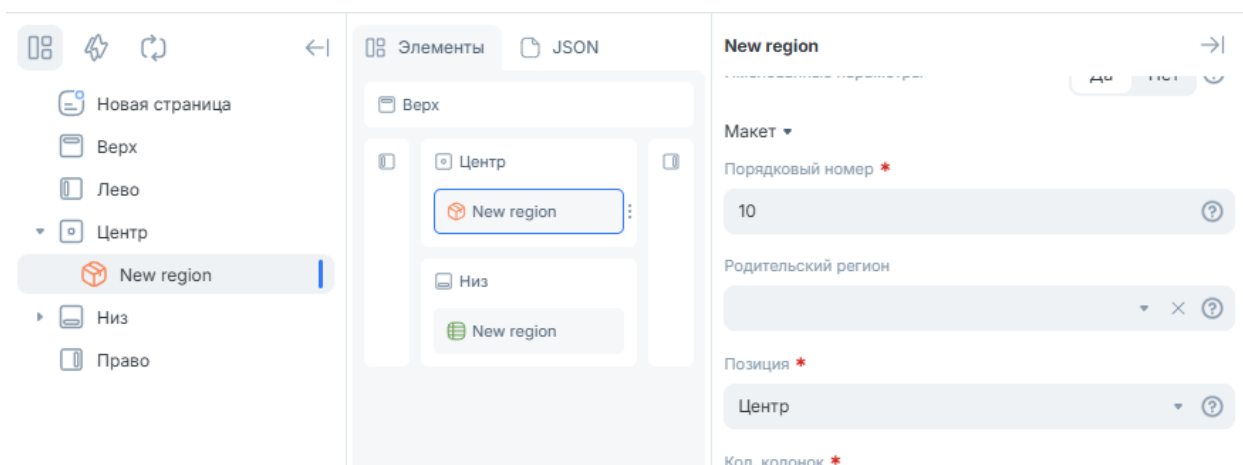
- верх (top). Здесь удобно расположить регион типа “хлебные крошки”, либо другой вариант заголовка страницы.
- лево (left). Хороший вариант расположить навигационную панель, дерево объектов, либо форму фильтров.
- центр (body). Здесь регионы с основными содержимым страницы: отчёты, формы ввода и т.д.
- право (right). Удобный вариант для панели дополнительных инструментов.
- низ (footer). Область для размещения общей информации. На страницах вида модальное окно здесь располагаются кнопки.



После создания, регион можно переместить в другую область страницы, поменяв ему в параметрах свойство “Позиция”(Position).

Предупреждение

Позиция определяет не только расположение элементов на странице, а также порядок прорисовки. Это необходимо учитывать при установке значений по умолчанию для элементов страницы.



6.3 Расположение компонентов на сетке

Визуальные компоненты страницы - регионы и элементы формы (элементы ввода, выбора и кнопки формы) располагаются в сетке из 12 колонок. Сетка для регионов верхнего уровня (в качестве родителя указана страница) определяет положение региона внутри области расположения, для вложенных регионов сетка определяет положение внутри родительского региона, как и для элементов формы.



В XRAD положение компонента на сетке регулируется следующими параметрами:

- Количество столбцов (Column Span). Принимает значение от 1 до 12, либо Automatic. Указывает количество ячеек сетки, занимаемых компонентом. По умолчанию значение 12 для регионов и Automatic для элементов формы.
- Начать новую строку (Start New Row). Переключатель, принимающий значение true / false. Указывает на то, должен ли компонент занять указанное количество ячеек слева начиная с новой строки, или должен продолжить строку, вслед за предыдущим компонентом по порядку, определяемому свойством последовательность.

В одной строке сетки не может быть заполнено более 12 ячеек. В случае, если сумма диапазонов столбцов для компонентов без переноса строки превысит 12, при попытке открыть страницу отобразится ошибка рендеринга.

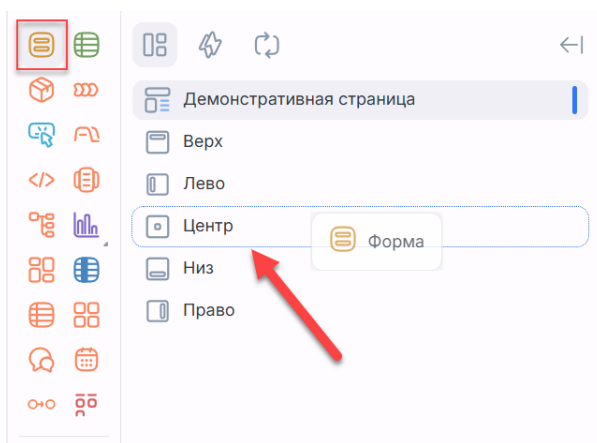
В случае, если сумма компонентов одной строки составит менее 12, оставшиеся ячейки заполнятся, если хотя бы у одного региона в строку указано значение Automatic в количестве столбцов, иначе останется свободное пространство.

Основной задачей любой интерактивной системы является ввод и вывод данных. В данном разделе будет описаны основные средства ввода данных и принципы работы с ними.

7.1 Создание формы

Для создания интерактивных форм необходимо использовать регион с типом «Форма». Платформа разрешает размещать элементы ввода только в данном типе региона.

Для создания формы вы можете воспользоваться панелью инструментов - выбрать компонент «Форма» и перенести в необходимую область страницы:



Либо создать регион воспользовавшись контекстным меню, которое появляется при нажатии правой кнопки мыши в дереве страницы. После чего необходимо будет указать тип региона - «Форма».

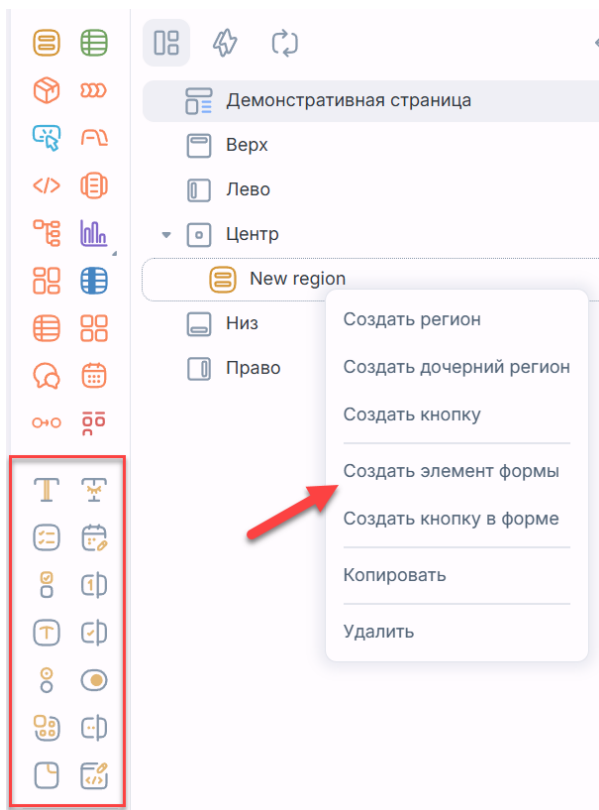
7.2 Элементы ввода

Для взаимодействия с пользователем платформа предоставляет различные элементы ввода. Элементы ввода - это интерактивные элементы управления, предназначенные для получения информации от пользователя. Они помогают ввести данные в нужном формате, осуществить проверки заполненности и полноты данных, а также предоставляют различные функциональные возможности по упрощению самого ввода.

Платформа предоставляет на выбор более 10 типов элементов ввода, которые включают в себя такие типы как «Текстовый ввод», «Числовой ввод», «ввод Даты», «Загрузка файлов» и пр.

7.2.1 Создание элемента ввода

После добавления формы на страницу можно начинать заполнять её элементами ввода. Элементы ввода можно добавить на форму путем вызова контекстного меню и выбора пункта «Создать элемент формы», либо перетащить из палитры компонентов.



7.2.2 Основные атрибуты элемента

Рассмотрим основные атрибуты, которые присутствуют вне зависимости от типа элемента ввода.

Атрибут	Описание
Имя	Для идентификации элемента необходимо указать его имя. Имя должно быть уникальным в разрезе заданной страницы. Для корректной работы приложения настоятельно рекомендуется указывать префикс в формате: RXXX. Где XXX - номер текущей страницы. При создании нового элемента ввода на форме, имя будет заполнено автоматически, с правильно сформированным префиксом. Имя также будет выступать в роли идентификатора элемента в HTML.
Тип	Необходимо указать тип элемента. По умолчанию проставляется в элемент типа «Текст». См. далее для описания типов элементов.
Именованные параметры	Флаг, который указывает необходимость воспринимать входящие параметры как часть текста sql запроса. При установке в false, разработчику потребуется явно указать входящие параметры. Отображается для источника данных отличных от типа Oracle, Mssql.
Временная переменная	Флаг отвечает за настройку хранения значения элемента в состоянии сессии. Если указан, значение не хранится в сессии, но доступно в ходе выполнения процессов загрузки и обработки страницы.
Защищенное значение	Если указан данный флаг, значение элемента будет защищено контрольной суммой. В случае если пользователь попытается изменить значение через js и отправить страницу, система выдаст ошибку проверки контрольной суммы. Доступно только для элементов с типом «Скрытый».
Источник данных	Источник данных для данного компонента.
Последовательность	Поле определяет последовательность отображения элемента ввода в данном регионе. Необходимо указать целочисленное значение. При создании нового элемента поле заполняется автоматически - увеличивая значение на 10 от предыдущего элемента.
Регион	Определяет регион, которому принадлежит элемент ввода, можно выбрать регион только с типом «Форма».
Начать новую строку	Указывает необходимость перенести элемент на новую строку внутри сетки региона.
Количество столбцов	Указывает количество занимаемых столбцов внутри сетки региона.
Наименование	Подпись элемента на странице. Отображается только, если отображается элемент.
Не пустой	Определяет необходимость наличия значения внутри элемента при обработке результата ввода. Если значение не было указано пользователю будет отправлено сообщение об ошибке, с требованием заполнить значение для заданного элемента.
Максимальная длина	Определяет максимальное возможное значение длины элемента в символах. Действует для типов элементов, где ожидается ввод с клавиатуры.
Подсказка	Значение для подсказки, которая показывает ожидаемые значения.
Маска ввода	Определяет формат ввода информации в поле. Для форматирования пользовательского ввода используйте следующие идентификаторы: • 9 — цифровой символ • a — символ буквы • * — буквенно-цифровой символ
Настройки JS	Определяет дополнительные настройки элемента ввода. См. подробности далее.
Классы CSS	Позволяет передать дополнительные css классы для настройки внешнего вида компонента.
Текст до элемента	Позволяет ввести дополнительный текст, который будет отображаться перед элементом ввода. Принимает html.
Текст после элемента	Позволяет ввести дополнительный текст, который будет отображаться после элемента ввода. Принимает html.

продолжается на следующей странице

Таблица 1 – продолжение с предыдущей страницы

Атрибут	Описание
Тип настроек по умолчанию	Выберите значение по умолчанию. Тип определяет откуда брать значение. Доступные значения: - Статический - введите значение по умолчанию, которое будет отображаться, если значение сессии для данного элемента не задано - SQL - введите sql выражение, которое вернет значение для элемента - Элемент - значение будет взять из значения элемента приложения, записанного в сессию.
Условия отображения	Определяет условия вывода элемента на форму.
Только чтение	Определяет условия вывода элемента, как доступного только для чтения.
Схема авторизации	Позволяет указать схему авторизации которая должна быть выполнена для отображения компонента на форме.

7.2.3 Типы элементов

Рассмотрим более подробно каждый тип элементов ввода:

Текст

Элемент для текстового ввода пользовательской информации. Дополнительные настройки (Настройки JS) включают в себя следующие возможности:

- **Отправка при нажатии Enter** - производит отправку страницы если пользователь нажал Enter, при выделении элемента ввода.
- **Пропустить валидации** - пропускает валидации при отправке страницы если был нажат Enter.
- **Показать блокировку** - показывает блокировку страницы до фактического выполнения процесса отправки при нажатии на Enter.

Текстовая область

Элемент для текстового ввода многострочного значения. Дополнительные настройки (Настройки JS) включают в себя все возможности элемента с типом «Текст», а также дополнительно:

- **Количество строк** - позволяет задать количество пустых строк, которые будут отображены при отрисовке элемента.

Список

Элемент выбора из списка элементов. Отличительной особенностью данного элемента является необходимость ввода источника значений для отображения. SQL-запрос, который определяет источник данных, должен возвращать две колонки: первая колонка будет отображена пользователю на форме, значение второй колонки будет передано в виде выбранного значения в процесс обработки. Указание количества колонок отличное от двух приведет к ошибке. Список также имеет следующие дополнительные настройки (Настройки JS):

- **Отобразить пустое значение** - позволяет отображать пустое значение в выпадающем списке. В случае выбора пустого значения в процесс будет передано значение NULL.
- **Отображаемое пустое значение** - позволяет задать отображаемое значение для пустого выбора.
- **Возвращаемое пустое значение** - позволяет указать значение, которое необходимо передать в процесс вместо NULL, при выборе пустого значения

- **Кнопка очистки** - позволяет отобразить дополнительную кнопку управления элементом, по нажатию на которую значение сбросится в пустое.

Числовой

Элемент для числового ввода. Позволяет гибко настроить формат ввода числовых значение. При отправке на сервер необходимо преобразовывать значение из строки в число (`to_number`, `convert`, и пр.). Дополнительные настройки включают в себя настройку следующих параметров:

- **decimalPlaces** - Количество десятичных знаков. Определяет количество отображаемых знаков после разделителя десятичного значения.
- **minimumValue** - Минимальное значение. Определяет минимальное значение доступное для ввода.
- **maximumValue** - Максимальное значение. Определяет максимальное значение доступное для ввода.
- **digitGroupSeparator** - Знак разделителя разрядов. Определяет знак который будет использован для разделения групп разрядов числа.
- **decimalCharacter** - Знак разделителя десятичной части числа. Определяет знак который будет использован для разделения десятичной части числа.
- **decimalCharacterAlternative** - Альтернативный знак для десятичного разделителя.
- **overrideMinMaxLimits** - Переопределяет минимальное и максимальное значение. Возможные значения - `ceiling`, `floor`, `ignore`, `invalid`.

Скрытый

Элемент, который не будет отображаться на форме. Используется для передачи параметров на страницу. В отличие от всех других элементов может быть создан вне формы в дереве страницы в узле «Параметры». Элемент имеет уникальную настройку «Защищенное значение» при указании которой значение может быть выставлено только при загрузке страницы. Если пользователи попытается изменить значение через js или путем вызова асинхронных процессов - при отправке значения на сервер, будет выведена ошибка.

Дата

Элемент для выбора значения типа дата и время. Элемент содержит дополнительную кнопку управления, по нажатию на которую раскрывается календарь с возможностью выбора даты и/или времени. Дополнительные настройки включают в себя:

- **Дата** - Позволяет выбрать дату в открывающемся календаре.
- **Время** - Позволяет выбрать время в открывающемся календаре.
- **Минимальная дата** - Позволяет задать минимальную дату. Дата может быть указана как абсолютное значение в формате «DD.MM.YYYY», так и в относительном. Относительные значения задаются от текущей даты и используют следующий формат:
 - `y +/-` количество лет от текущей даты. Пример: `+3y`; `-7y`;
 - `m +/-` количество месяцев от текущей даты. Пример: `+2m`; `-7m`;
 - `w +/-` количество недель от текущей даты. Пример: `+1w`; `-7w`;
 - `d +/-` количество дней от текущей даты. Пример: `+10d`; `-15d`;

Можно использовать несколько директив одновременно: `+1d-2w+3y`.

- **Максимальная дата** - Позволяет задать максимальную дату. Дата может быть указана как абсолютное значение в формате «DD.MM.YYYY», так и в относительном. Относительные значения задаются от текущей даты и используют следующий формат:

- y +/- количество лет от текущей даты. Пример: +3y; -7y;
- m +/- количество месяцев от текущей даты. Пример: +2m; -7m;
- w +/- количество недель от текущей даты. Пример: +1w; -7w;
- d +/- количество дней от текущей даты. Пример: +10d; -15d;d

Можно использовать несколько директив одновременно: +1d-2w+3y.

Радиокнопки

Элемент для отображения радиокнопок. Для задания списка значений используйте выражения типа SQL, которое вернет два столбца. Первый столбец будет содержать отображаемое на форме значение радиокнопки, второй столбец будет содержать значение, передаваемое обработчику форму. Дополнительные настройки включают в себя:

- **Колонки** - Позволяет задать количество колонок на которое будут разбиты значения радиокнопок.
- **Перенос колонок** - Позволяет задать способ переноса значений по колонкам.

Флажки

Элемент для выбора значений по флажкам. Также как для радиокнопок необходимо задать список значений используя SQL выражение. Дополнительные настройки:

- **Разделитель** - Позволяет задать знак разделителя, который будет использован для разделения выбранных значений при передаче значений в процессы. По умолчанию используется знак двоеточие (:))
- **Колонки** - Позволяет задать количество колонок на которое будут разбиты значения флажков.
- **Перенос колонок** - Позволяет задать способ переноса значений по колонкам.

Переключатель

Элемент для отображения переключателя значения. Работает также как радиокнопки, отличием является визуальное оформление. Также как для радиокнопок необходимо задать список значений используя SQL выражение. Дополнительные настройки отсутствуют.

Автозаполнение

Элемент типа список с возможностью автозаполнения. При вводе значения будет производиться поиск по доступным значениям списка, отфильтровывая результат возможных значений. Для заполнения элемента конечному пользователю необходимо выбрать значение из отфильтрованного списка. Также как для радиокнопок необходимо задать список значений используя SQL выражение. Дополнительные настройки отсутствуют.

Пароль

Элемент для ввода пароля. Введенные символы автоматически заменяются на знак *, по требованию пользователя информация может быть отображена по нажатию на кнопку рядом с элементом.

WYSIWYG

Элемент отображается в виде редактора WYSIWYG, который подходит для создания форматированного текста с использованием html.

Множественный выбор

Список элементов с возможностью выбора нескольких значений. Отличительной особенностью данного элемента является необходимость ввода источника значений для отображения. SQL-запрос, который определяет источник данных, должен возвращать две колонки: первая колонка будет отображена пользователю на форме, значение второй колонки будет передано в виде выбранного значения в процесс обработки. Указание количества колонок отличное от двух приведет к ошибке. Список также имеет следующие дополнительные настройки (Настройки JS):

- **Разделитель** - Позволяет задать знак разделителя, который будет использован для разделения выбранных значений при передаче значений в процессы. По умолчанию используется знак двоеточие (:)
- **Поиск** - Позволяет вывести в элементе строку для поиска значений из списка.

Ввод файла

Элемент для загрузки пользовательских файлов. Настройки элемента задают:

- **Несколько файлов** - позволяет загружать несколько файлов.
- **Высота блока** - определяется высоту области drag and drop.
- **Mime типы** - задаёт ограничение по загружаемым типам файлов. Типы файлов должны быть указаны как mime-type через запятую. При указании значения “any” - снимает ограничение по типам.
- **Текст подсказки** - задаёт текст подсказки для пользователя
- **Макс. размер файла** - определяет максимально допустимый размер файла
- **Макс. кол-во файлов** - определяет максимальное количество файлов.

Для обработки загруженных файлов XRAD создаёт временную таблицу:

- **Postgresql** - в схеме pg_temp, используя следующее выражение: CREATE TEMP TABLE IF NOT EXISTS pg_temp.xrad_files (name text, file_name text, file_mime text, file_content bytea).
- **Oracle** - после выполнения скрипта установки для базы данных Oracle, всем схемам будет доступна публичная временная глобальная таблица XRAD_FILES (name varchar2(4000), file_name varchar2(4000), file_mime varchar2(4000), file_content blob).
- **Firebird** - после выполнения скрипта установки для базы данных Firebird, всем пользователям будет доступна публичная временная глобальная таблица XRAD_FILES (name varchar(4000), file_name varchar(4000), file_mime varchar(500), file_content blob).
- **Mysql** - будет создана временная таблица, используя следующее выражение: CREATE TEMPORARY TABLE IF NOT EXISTS xrad_files (name TEXT,file_name TEXT,file_mime TEXT,file_content LONGBLOB).
- **Mssql** - будет создана временная таблица, используя следующее выражение: IF OBJECT_ID('tempdb..#xrad_files') IS NULL BEGIN CREATE TABLE #xrad_files (name NVARCHAR(MAX),file_name NVARCHAR(MAX),file_mime NVARCHAR(255),file_content VARBINARY(MAX)) END.

Колонка name будет содержать значение из элемента ввода файла, переданное на сервер. При загрузке нескольких файлов значение элемента будет передано в виде строки, в которой значения будут разделены знаком двоеточия (:).

7.3 Кнопки

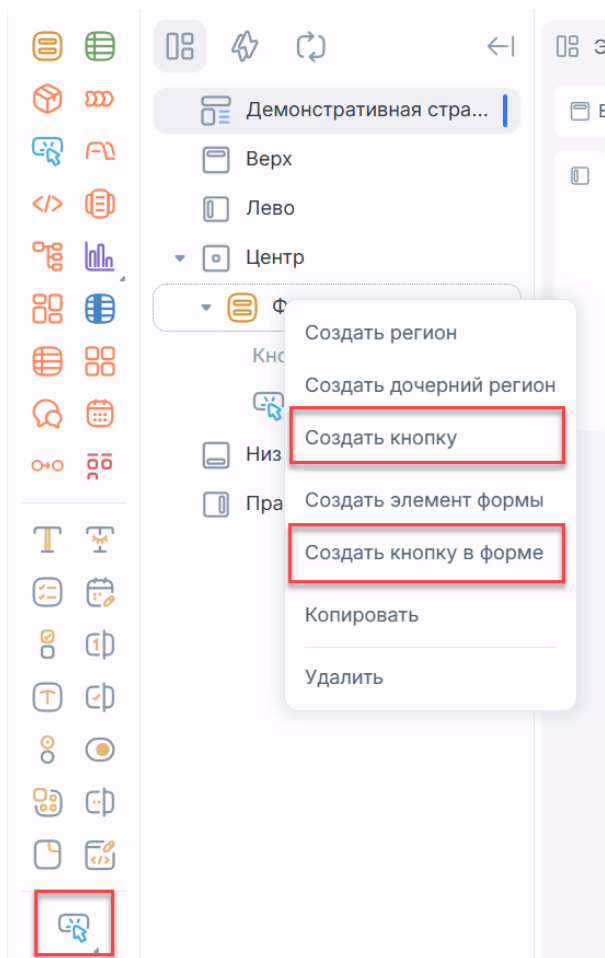
Кнопка - это функциональный интерактивный компонента, который запускает действие или событие на странице. С помощью кнопок пользователи могут подтверждать ввод данных в форму, осуществлять переходы на другие страницы и даже раскрывать дополнительные контекстные меню.

Кнопки могут размещаться в теле любых контейнеров, на форме, а также в заголовках и подвалах любых регионов.

7.3.1 Создание кнопки

Для добавления кнопки на форму можно воспользоваться контекстным меню, которое открывается по нажатию на правую кнопку мыши, при наведении курсора на любой регион. После открытия контекстного меню необходимо выбрать пункт «Создать кнопку». Для региона с типом «Форма» доступна дополнительная опция, которая позволит разместить кнопку непосредственно рядом с элементами ввода. Для создания такой кнопки в контекстном меню региона «Форма» выберите пункт «Создать кнопку в форме».

Также создание доступно с помощью выбора компонента «Кнопка» в палитре компонентов.



7.3.2 Параметры кнопок

После создания кнопки станут доступны следующие настройки компонента:

Параметр	Тип	Описание
Имя	Текст	Имя кнопки для отображения в редакторе страниц
Источник данных	Список	Определяет источник данных к которому будет обращаться система для отрисовки данного компонента.
Именованные параметры	Переключатель	Флаг, который указывает необходимость воспринимать входящие параметры как часть текста sql запроса. При установке в false, разработчику потребуются явно указать входящие параметры. Отображается для источника данных отличных от типа Oracle.
Лейбл	Текст	Определяет текст кнопки, который будет отображен на форме.
Порядковый номер	Число	Определяет порядок расположения кнопки на родительском регионе в рамках местоположения и позиции.
Родительский регион	Список	Определяет регион, на котором кнопка будет отображена
Местоположение	Список	Способ определения позиции кнопки. Для всех родительских регионов, кроме формы имеет одно значение на выбор - REGION. Всего вариантов три: • REGION. Вариант по умолчанию. Кнопка располагается в рамках региона на выбранной позиции по заданному порядку относительно других кнопок позиции. • ITEM. Применимо при родительском регионе типа «Форма». Позиция кнопки определяется относительно привязанного элемента по заданному порядку среди других кнопок той же позиции того же связанного элемента. • FORM. Применимо при родительском регионе типа «Форма». Кнопка располагается на форме, порядок определяется порядком среди других элементов формы

продолжается на следующей странице

Таблица 2 – продолжение с предыдущей страницы

Параметр	Тип	Описание
Позиция	Список	Определяет относительное расположение кнопки. Применимо для вариантов местоположения REGION и ITEM. Для местоположения REGION доступны следующие варианты: • Bottom Fluid. Кнопка растянется на всю ширину вдоль нижней границы региона. • Bottom Left. Кнопка расположится под основным содержимым региона в левой части. • Bottom Right. Кнопка расположится под основным содержимым региона в правой части. • Left Body. Кнопка расположится слева от основного содержимого в теле региона. • Right Body. Кнопка расположится справа от основного содержимого в теле региона. • Top Left. Кнопка расположится в левой части заголовка региона перед его названием. Отобразится только если включен параметр региона “Показать заголовок”. • Top Right. Кнопка расположится в правой части заголовка региона. Отобразится только если включен параметр региона “Показать заголовок”. Для местоположения ITEM варианта два: • Left of Item. Кнопка прикрепится к левой границе элемента ввода. • Right of Item. Кнопка прикрепится к правой границе элемента ввода.
Действие	Список	Поведение при нажатии на кнопку. На выбор четыре варианта: • Отправить страницу. Отправляет форму на сервер, устанавливает значение запроса (REQUEST) в равный параметру «статический идентификатор кнопки». • Перенаправление на Страницу. направляет на страницу, указанную в ссылке. Отправка значений формы при этом не происходит. • Определяется динамическим действием. Вызывает событие Click. • Открыть раскрывающийся список. Нажатие на кнопку вызовет контекстное меню, содержание которого определяется выбранным списком.
Пропустить валидации	Переключатель	Указывает необходимо ли пропустить валидации при сабмите страницы. Пропускает проверки обязательных полей и серверных валидаций, у которых отключена опция «Выполнять всегда». Применимо только для действия «Отправить страницу» и «Определяется динамическим действием».
Показать блокировку	Переключатель	Определяет необходимо ли заблокировать страницу в целях предотвращения повторной отправки страницы пользователем. Может использоваться для блокировки в случае долгого выполнения процессов.
Ссылка	Текст / Окно конструктора	Позволяет задать ссылку на другую страницу, вызвать динамический скрипт, либо настроить в конструкторе ссылку на страницу внутри приложения. Применимо для действия “Перенаправление на Страницу”.

продолжается на следующей странице

Таблица 2 – продолжение с предыдущей страницы

Параметр	Тип	Описание
Источник - Список	Список	Определяет список, отображаемый в контекстном меню. Применимо для действия “Открыть раскрывающийся список”.
Классы	Текст	Перечень CSS-классов кнопки. Здесь определяется цвет кнопки, иконка и др.
Статический идентификатор	Текст	Идентификатор кнопки, передаваемый в переменную REQUEST при нажатии на неё.
Схема авторизации	Список	Определяет схему авторизации которую необходимо выполнить для данного компонента. В случае успешного выполнения компонент будет отображен на страницу.
Условия отображения - Тип	Список	Определяет тип условия отображения кнопки. Подробнее см. в <i>условиях выполнения</i> .

7.3.3 Отображение кнопок

В зависимости от типа местоположения кнопки могут отображаться по-разному. Для наглядности возможные варианты – изображены на скриншотах.

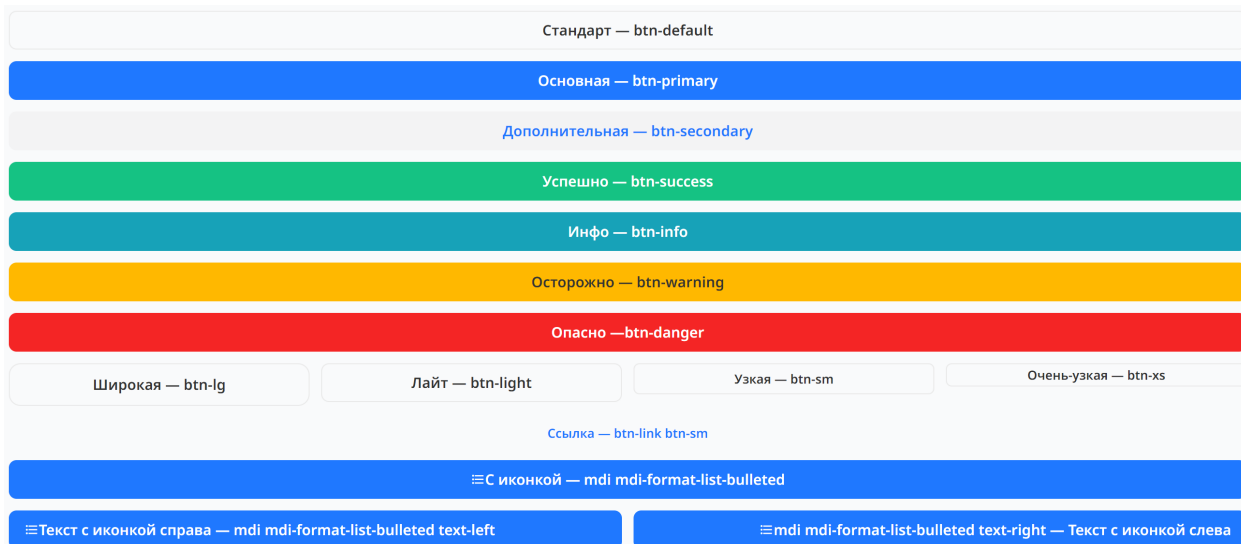
Для местоположения REGION:

Скриншот интерфейса для настройки кнопок в регионе (REGION). В центре находится поле "Позиция дочернего региона". По нему расположены кнопки: "Слева в заголовке", "Справа в заголовке", "Слева в теле региона", "Справа в теле региона", "Слева внизу", "Справа внизу". Внизу находится широкая кнопка "Внизу растянутая".

Для местоположения ITEM:

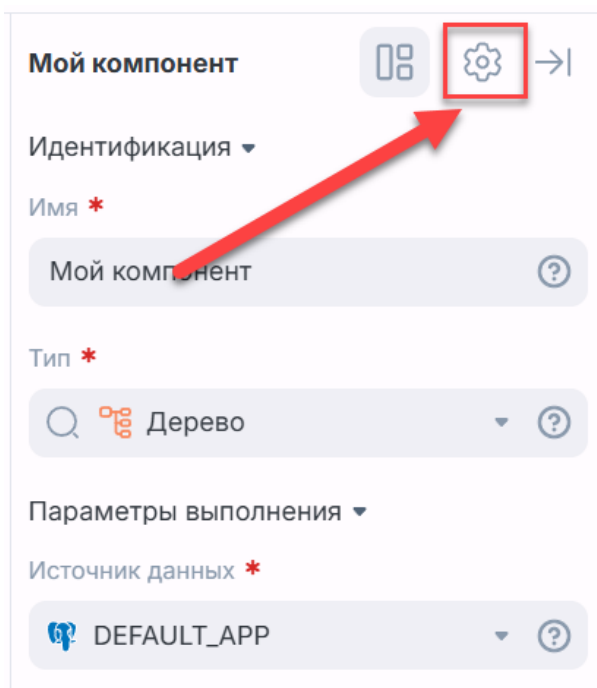
Скриншот интерфейса для настройки кнопок относительно элемента формы (ITEM). В центре находится поле "Элемент ввода без прикрепленных кнопок". По нему расположены кнопки: "Слева от элемента", "Справа от элемента". Внизу находится широкая кнопка.

Для местоположения FORM с демонстрацией различных классов кнопок:



Регион определенного типа, в контексте разработки приложения, называется компонент. В данном разделе описываются все типы регионов и настройки этих типов.

После создания региона и выбора его типа в большинстве случаев система предоставляет возможность настроить компонент. Настройка компонента осуществляется по нажатию на иконку «шестеренка», расположенной в правом верхнем углу редактора атрибутов компонента.

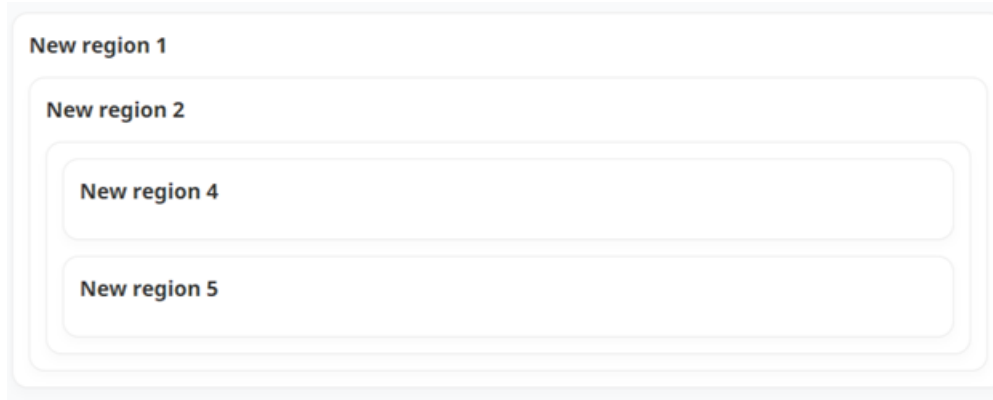


8.1 Обертка

Обертка - основной контейнер для регионов. Используется для группировки и зонирования компонентов или дочерних регионов.

Основными параметрами регионов типа «Обертка» является позиция на странице и отображение имени региона.

Из регионов данного типа можно составлять блоки любой сложности и вложенности.



8.1.1 Настройка компонента

Настройки компонента отсутствуют.

8.2 Хлебные крошки

Хлебные крошки - компонент для вывода навигационной цепочки и заголовка страницы на основе списка.

Для создания компонента проделайте следующие шаги:

1. Выберите страницу.
2. Создайте новый регион в позиции “верх”.
3. Укажите тип региона - BREADCRUMB.
4. Создайте список с типом BREADCRUMB в разделе «Списки».
5. Укажите созданный список в поле Источник - Список.

8.2.1 Создание элемента *списка*

Для вывода навигационной цепочки на компоненте необходимо добавить соответствующий странице элемент в список типа Breadcrumb, привязанный к региону

Главная > Дополнительно

Отчёты

В списке типа BREADCRUMB создайте новый элемент.

В настройках элемента введите:

- родительский элемент - предыдущий узел навигационной цепочки;

- имя - отображаемый заголовок на странице;
- страница - привязка к странице для вывода на ней заголовка;
- ссылка - ссылка на страницу из навигационной цепочки.

Подробнее о работе со списками в п. Управление списками.

8.2.2 Настройка компонента

Настройки компонента отсутствуют.

8.3 Контейнер для кнопок

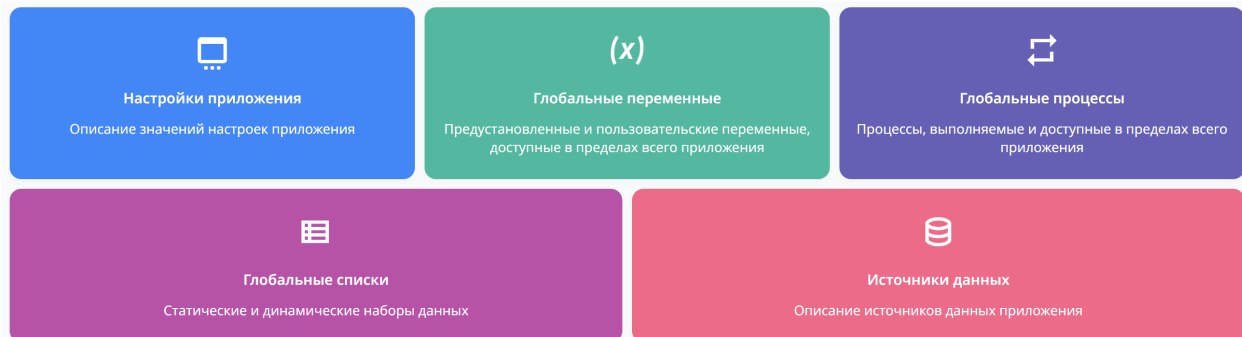
Контейнер для кнопок - компонент для размещения кнопок. Служит для отделения панели управления от основного блока данных. Подробнее про кнопки по [ссылке](#).

8.3.1 Настройка компонента

Настройки компонента отсутствуют.

8.4 Карточки

Карточки - регион для графического представления списка в виде карточек. Карточки имеют собственный цвет, картинку, а также ссылку на страницу, или скрипт.



Источником данных для компонента CARDS является – список.

8.4.1 Создание элемента списка

Для вывода карточки на компоненте необходимо добавить элемент в список, привязанный к региону.

В списке создайте новый элемент и настройте параметры:

- Имя – наименование элемента и отображаемый заголовок;
- Последовательность - численный порядок элемента, определяющий его позицию;
- CSS класс иконки - иконка для карточки;
- Ссылка - ссылка на страницу или скрипт при клике на карточку;
- Атрибут 1 – вспомогательная надпись;
- Атрибут 2 - RGB-цвет карточки в формате #ffffff.

Пример SQL-запроса для создания динамического списка:

```
SELECT
  'Регионы' title,
  'dripicons dripicons-browser' css_icon,
  '/content/5020' target,
  'Демонстрация типов регионов на примерах использования' attribute_01,
  '#FFC62D' attribute_02
```

Примечание: При создании динамического списка для использования его в качестве источника данных региона Карточки требуется соблюдение строгого именования столбцов как в вышеуказанном примере!

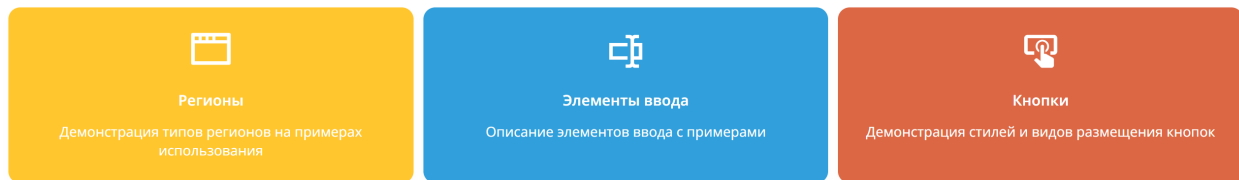
8.4.2 Настройка компонента

Компонент «Карточки» имеет два специальных атрибута, доступных в настройках компонента:

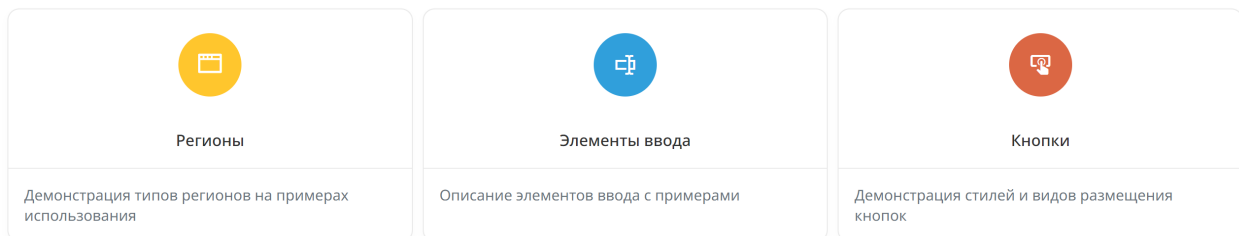
Атрибут	Тип	Описание
Тема	Список	Шаблон, определяющий отображение карточек. На выбор три варианта: • Блочная. • Функциональная. • Базовая.
Колонки	Список	Количество карточек в ряд. В случае если элементов в списке больше, чем указанное здесь число, карточки расположатся в несколько рядов. Принимает значение от одного до двенадцати.

Для наглядности виды шаблонов изображены на скриншотах:

Блочная



Функциональная



Базовая

Регионы 	Элементы ввода 
Демонстрация типов регионов на примерах использования	Описание элементов ввода с примерами
Кнопки 	
Демонстрация стилей и видов размещения кнопок	

8.5 График

График – служит для графического представления числовых данных в виде диаграмм. Компонент поддерживает следующие типы диаграмм:

- Столбчатая – вертикальная
- Столбчатая - горизонтальная
- Кольцевая диаграмма
- Линейная диаграмма
- Диаграмма с областями

Источником данных для компонента служат результаты SQL – запроса.

Пример SQL-запроса:

```
SELECT
  'Рабочие дни' day_kind,      -- Подпись
  247::numeric days           -- Значение
UNION ALL
SELECT
  'Выходные дни',
  118::numeric
```



8.5.1 Настройка компонента

Доступны настройки двух уровней: атрибуты компонента и параметры серий. Рассмотрим подробнее каждую группу.

8.5.2 Атрибуты компонента

Группа специальных атрибутов, характерных для данного вида регионов.

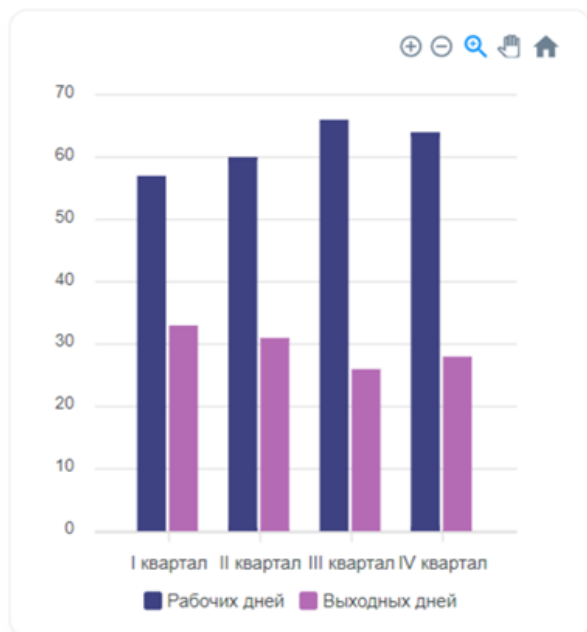
Атрибут	Тип	Описание
Диаграмма - Тип	Список	Определяет внешний вид диаграммы. Доступны варианты: • Столбчатая – вертикальная • Столбчатая - горизонтальная • Кольцевая диаграмма • Линейная диаграмма • Диаграмма с областями
Колонка категории	Список	Определяет какой из столбцов запроса к БД содержит значение для категории диаграммы. Значения серий должны быть сгруппированы по категориям.
Динамическая загрузка	Переключатель	Если установлен, то диаграмма не будет строиться без явного вызова из динамического события. Иначе загружается при открытии страницы.
Ширина	Число	Определяет ширину для компонента.
Высота	Число	Определяет высоту для компонента.

8.5.3 Параметры серий

Параметр	Тип	Описание
Имя	Текст	Определяет отображаемое имя серии. Для дерева объектов и для отображения на диаграмме в приложении. Поскольку в графиках типа кольцевая диаграмма серия одна, для этого типа имя серии отобразится только в дереве объектов.
Последовательность	Число	Определяет порядок серий при выводе в интерфейс. Поскольку в графиках типа кольцевая диаграмма серия одна, отобразится в приложении только первая из них.
Колонка значений	Список	Из выпадающего списка необходимо выбрать один из столбцов запроса, содержащий значение типа numeric, для вывода значения в серию.
Условия отображения - Тип	Список	Определяет тип условия отображения серии. Подробнее см. в условиях выполнения .

Для наглядности виды диаграмм изображены на скриншотах:

Столбчатая



Кольцевая

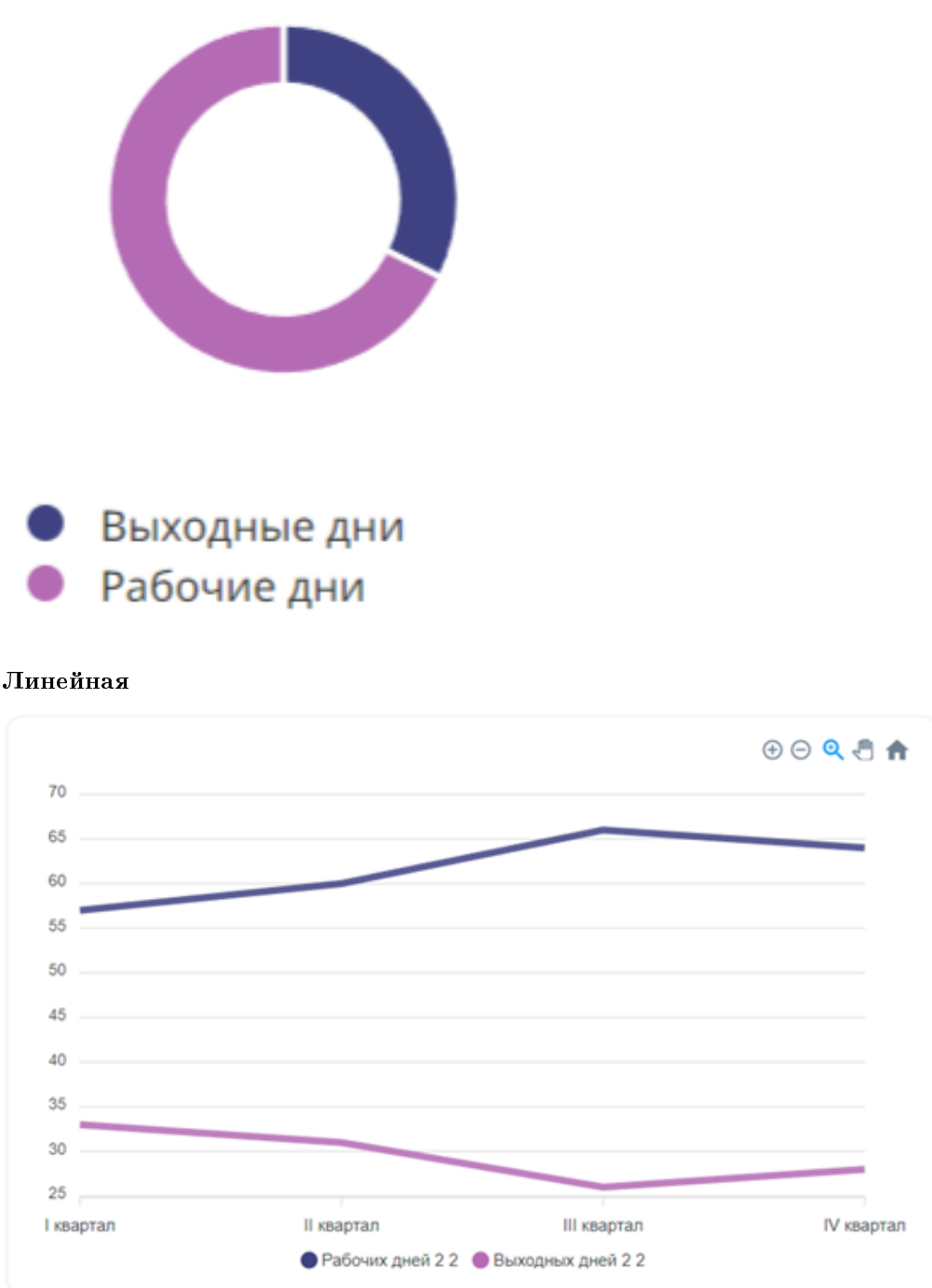
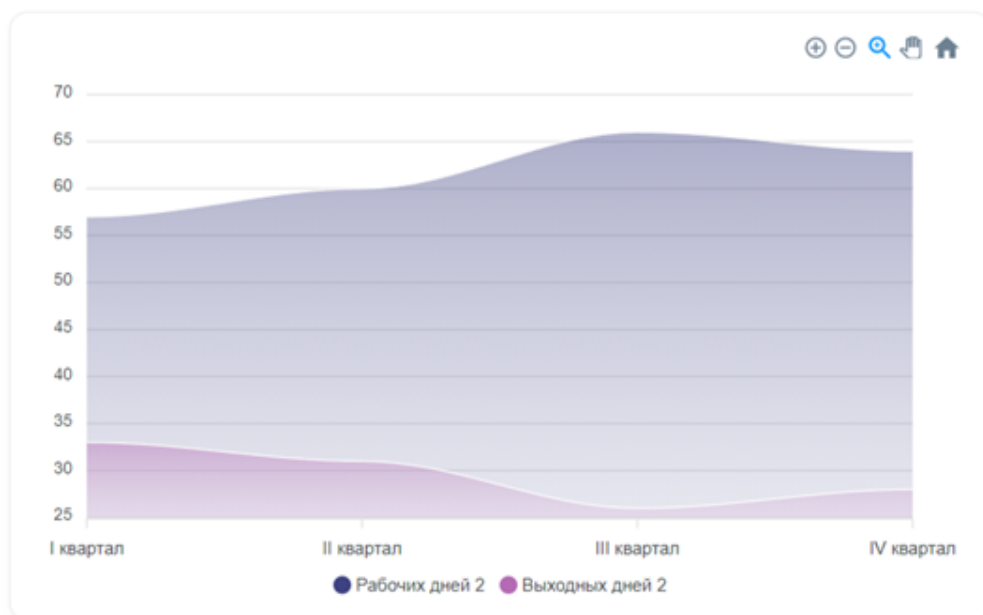


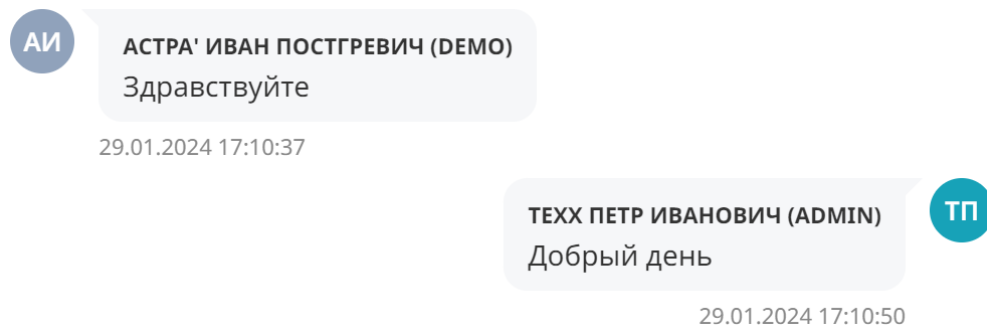
Диаграмма с областями



Графику можно установить пользовательские фильтры данных по элементам формы. Для этого необходимо прописать правильное условие в запросе выборки данных чарта и передать необходимые параметры, которые будут ограничивать выборку.

8.6 Чат

Чат – регион для вывода чатов. В качестве источника данных для компонента является результат SQL-запроса. Таким образом компонент отображает полученный в результате SQL-запроса набор данных в виде пользовательского чата.



В запросе помимо текста сообщения, даты и пользователя, также указывается положение текста и стиль оформления иконки.

Пример SQL-запроса:

```
SELECT
  cm.message_text comment_text, -- Текст сообщения
  TRIM(CONCAT(ul.last_name, ' ',
              ul.first_name, ' ',
              ul.second_name)) || ' (' || ul.username || ')' user_name, -- Имя пользователя
  CASE
    WHEN ul.username = UPPER($1) THEN 'right'
```

(продолжается на следующей странице)

(продолжение с предыдущей страницы)

```

    ELSE 'left'
END align, -- Выравнивание сообщения (left, right)
TO_CHAR(cm.message_date,'DD.MM.YYYY HH24:MI:SS') comment_date, -- Дата отправки
SUBSTR(ul.last_name,1,1)||SUBSTR(ul.first_name,1,1) user_icon, -- Текст в иконке
CASE
    WHEN ul.username = UPPER($1) THEN 'bg-info text-white'
    ELSE NULL
END icon_modifier -- CSS классы иконки
FROM
    chat.t_chat_messages cm
JOIN
    users.t_user_list ul ON cm.id_user = ul.id
WHERE
    cm.id_chat = 1
ORDER BY
    cm.message_date

```

8.6.1 Настройка компонента

Настройки компонента отсутствуют.

8.7 Форма

Форма – компонент для создания разнообразных элементов ввода пользовательской информации. Более подробнее по [ссылке](#).

8.7.1 Настройка компонента

Настройки компонента отсутствуют.

8.8 HTML-код (HTML)

HTML – регион для отображения HTML-кода. Данный тип региона используется для вывода текста, картинок, видео и любых других элементов HTML-кода.

Текст и картинка

Текст



Компонент поддерживает два типа источника данных:

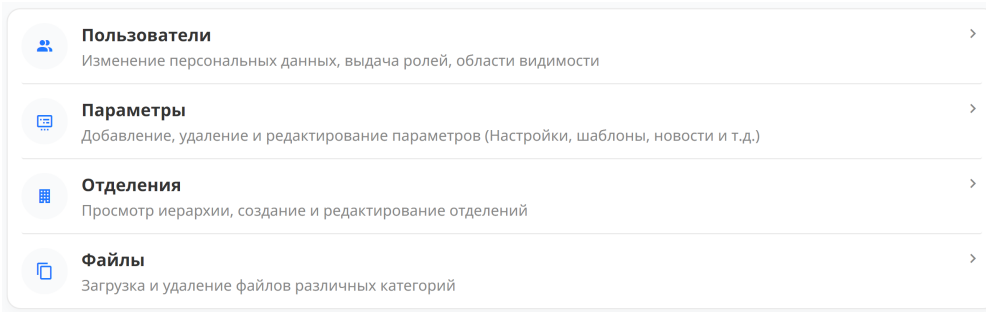
- Статический – HTML-код указывается напрямую в параметре HTML код.
- SQL – HTML-код будет получен в результате выполнения SQL-запроса.

8.8.1 Настройка компонента

Настройки компонента отсутствуют.

8.9 Навигационная панель

Навигационная панель — компонент для размещения навигационной панели на странице. Источником данных служит список, элементы которого отобразятся с иконкой, заголовком-ссылкой и описанием.



Пример SQL-кода для создания динамического списка:

```
SELECT
    n.title name,
    'uil uil-envelope' icon,
    n.message_short attribute_01,
    '/content/9201?P9201_ID=' || n.id target,
    ns.status_name attribute_02
FROM orgn.t_notifications n
ORDER BY n.date_created DESC;
```

Примечание: При создании динамического списка для использования его в качестве источника данных региона Панель навигации требуется соблюдение строгого именования столбцов как в вышеуказанном примере!

8.9.1 Настройка компонента

Настройки компонента отсутствуют.

8.10 Таблица

Таблица — это простой неинтерактивный компонент, предназначенный для вывода табличных данных. Источником данных для компонента является результат выполнения SQL запроса.

Таблица

emp_no ↕	birth_date ↕	first_name ↕	last_name ↕	gender ↕	hire_date ↕
499999	1958-05-01	Sachin	Tsukuda111	M	1997-11-30
499998	1956-09-05	Patricia	Breugel11	M	1993-10-13
499997	1961-08-03	Berhard	Lenart1	M	1986-04-21
499996	1953-03-07	Zito	Baaz1	M	1990-09-27
499995	1958-09-24	Dekang	Lichtner	F	1993-01-12
499994	1952-02-26	Navin	Argence	F	1990-04-24
499993	1963-06-04	DeForest	Mullainathan	M	1997-04-07
499992	1960-10-12	Siamak	Salverda	F	1987-05-10
499991	1962-02-26	Pohua	Sichman	F	1989-01-12
499990	1963-11-03	Khaled	Kohling	M	1985-10-10
499989	1954-05-26	Keiichiro	Lindqvist	M	1993-10-28
499988	1962-09-28	Bangqing	Kleiser	F	1986-06-06
499987	1961-09-05	Rimli	Dusink	F	1998-09-20
499986	1952-07-22	Nathan	Ranta	F	1985-08-11
499985	1964-12-26	Gila	Lukaszewicz	M	1997-02-11

< 1 - 15 >

8.10.1 Настройка компонента

Доступны настройки трёх уровней: параметры региона, атрибуты таблицы и параметры столбцов таблицы. Рассмотрим подробнее каждую группу.

8.10.2 Параметры региона

Для формирования компонента «Таблица» необходимо указать в настройках региона тип региона - «Таблица» и указать SQL запрос, результат выполнения которого будет отображен пользователю. После указания SQL запроса, в области расположенной под регионом в дереве компонентов, система добавит дополнительные компоненты - колонки таблицы. Колонки будут соответствовать колонкам запроса и позволят дополнительно кастомизировать вывод данных.

8.10.3 Атрибуты таблицы

Дополнительные атрибуты компонента позволяют более детально настроить внешний вид таблицы и выбрать шаблон отображения. Для редактирования доступны следующие поля:

Атрибут	Тип	Описание
Включить разбиение на страницы	Переключатель	Атрибут позволяет управлять отображением пагинации компонента. Если указан, конечный пользователь сможет переключать старинцы таблицы.
Включение заголовков	Переключатель	Позволяет управлять отображением названий столбцов таблицы. Если указан - названия будут отображаться.
Шаблон	Список	Позволяет управлять шаблоном таблицы. Для выбора доступно два шаблона: «Стандарт», «Колонка - значение». При выборе «Стандарт» данные будут отображаться в привычном виде таблицы. При выборе «Колонка - значение» таблица будет транспонирована.
Ширина первой колонки	Текст	Указывается только при выборе шаблона «Колонка - значение». Задаёт ширину первой колонки таблицы. Указывается в % или px. Вторая колонка займет оставшуюся ширину компонента.
Ограничение строк	Число	Устанавливает количество строк отчёта на одну страницу пагинации.
Зебра	Переключатель	Переключает отображение акцентуации четных и нечетных строк таблицы.
Подсветка	Переключатель	Отвечает за отображение подсветки строки при наведении на неё курсором.
Отступ в ячейках	Переключатель	Добавляет или убирает отступ в ячейках от содержимого.
Границы строк	Переключатель	Отвечает за отображение визуального разделителя строк таблицы.
Перенос строк в заголовке	Переключатель	Отвечает за перенос строки текста в заголовке при условии, что длина текста не может быть помещена в ячейку заголовка без переноса.
Перенос строк в заголовке	Переключатель	Отвечает за перенос строки текста в ячейках при условии, что длина текста не может быть помещена в ячейку строки без переноса.
Сообщение об отсутствии данных	Область текста	Позволяет указать текст, который будет выведен на экран в случае, если запрос к БД не вернул данных.

8.10.4 Параметры столбцов

Дополнительные настройки каждой колонки таблицы осуществляются при редактировании компонентов «Колонка таблицы»:

Параметр	Тип	Описание
Имя колонки	Текст (только чтение)	Наименование столбца. Определяется псевдонимом столбца запроса источника данных. Поле не редактируется в настройках параметров - оно всегда соответствует наименованию столбца в запросе. Ссылка на значение столбца происходит по этому полю.

продолжается на следующей странице

Таблица 1 – продолжение с предыдущей страницы

Параметр	Тип	Описание
Тип	Список	Определяет набор параметров столбца и его поведение при выводе на страницу. Может принимать следующие значения: • Текст. Простой вывод значения информации из БД. • Ссылка. Позволяет нажатием на элемент внутри ячейки столбца перенаправить пользователя на другую страницу, либо вызвать динамический скрипт. • Скрытый. Столбец репорта будет скрыт в пользовательском интерфейсе. Его значение (например, для передачи в параметры ссылки) будет всё ещё доступно • Флажок. Столбец представляет собой группу чекбоксов для множественного выбора значений отчёта.
Связанный элемент	Текст / Окно кон- структо- ра	Позволяет выбрать элемент ввода, либо ввести глобальную переменную, куда будут переданы значения из ячеек отмеченных чекбоксом строк. Параметр применим для типа столбца - Флажок.
Именованные параметры	Пере- ключа- тель	Флаг, который указывает необходимость воспринимать входящие параметры как часть текста sql запроса. При установке в false, разработчику потребуется явно указать входящие параметры. Отображается для источника данных отличных от типа Oracle, Mssql.
Заголовок	Текст	Заголовок столбца для вывода в пользовательском интерфейсе на странице приложения.
Последовательность	Число	Определяет порядок столбцов при выводе в интерфейс. Изменяя порядок в системе - порядок отображения будет изменен и у конечного пользователя. По умолчанию равен порядку столбцов в запросе при первой генерации. При добавлении столбца в запрос в дальнейшем, новому будет присвоен последний номер, вне зависимости от позиции в запросе.
Выравнивание	Пере- ключа- тель	Выравнивание текста-содержимого столбца. По левому краю, по центру, или по правому краю.
Ссылка	Текст / Окно кон- структо- ра	Позволяет задать ссылку на другую страницу, вызвать динамический скрипт, либо настроить в конструкторе ссылку на страницу внутри приложения. В последнем случае можно выбирать значения не только полей ввода, но и столбцов отчёта. Вместо текста ссылки можно также передать значение строки репорта - такой подход позволяет настраивать ссылку внутри sql-запроса. Параметр применим для Тип столбца - Ссылка.
Значок ссылки	Текст	Позволяет выбрать иконку для вывода в ячейке столбца. Вместо текста ссылки можно также передать значение строки репорта - такой подход позволяет выбрать картинку внутри sql-запроса. Параметр применим для Тип столбца - Ссылка.
Ширина	Текст	Фиксирует ширину столбца.
Формат маски	Текст	Форматирует вывод наложением маски по правилам форматирования SQL-функции to_char для Postgresql и Oracle, FORMAT для mssql, DATE_FORMAT для mysql для типа данных DATE и FORMAT для типа данных NUMBER. Для firebird необходимо указывать пользовательскую функцию форматирования данных - вызов функции будет происходить с подстановкой функции в SQL.

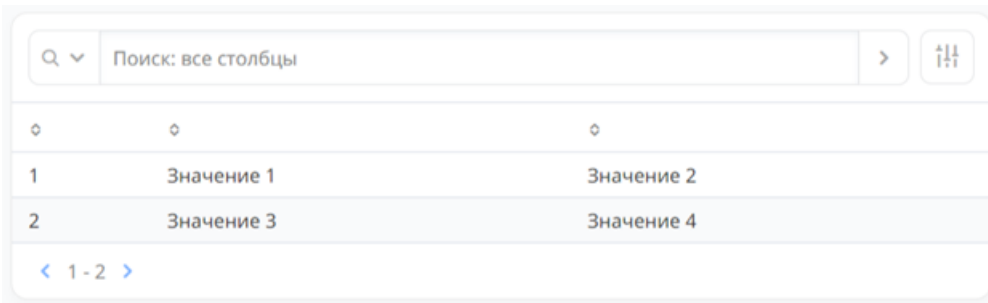
продолжается на следующей странице

Таблица 1 – продолжение с предыдущей страницы

Параметр	Тип	Описание
Включить сортировку	Переключатель	Разрешает сортировать строки по значению столбца.
Выражение HTML	Область текста	Позволяет определить формат вывода в виде HTML. Можно сослаться на значение столбца таблицы через #COLUMN#.
Схема авторизации	Список	Определяет схему авторизации которую необходимо выполнить для данного компонента. В случае успешного выполнения компонент будет отображен на страницу.
Условия отображения - Тип	Список	Определяет тип условия отображения. Подробнее см. в <i>условиях выполнения</i> .

8.11 Отчет

Отчет – интерактивный компонент для вывода данных в виде таблицы. Данные для вывода формируются в результате выполнения SQL-запроса.



Отчет имеет набор встроенных функций, которые позволяют конечному пользователю производить поиск по строкам таблицы, по колонкам, осуществлять фильтры и делать подсветки строк по заданным условиям, создавать пользовательские отчеты. Разработчик со своей стороны может ограничить доступный функционал, либо использовать его для создания отчетов, которые будут отображаться конечным пользователям по умолчанию.

8.11.1 Настройка компонента

Для формирования компонента «Таблица» необходимо указать в настройках региона тип региона - «Таблица» и указать SQL запрос, результат выполнения которого будет отображен пользователю. После указания SQL запроса, в области расположенной под регионом в дереве компонентов, система добавит дополнительные компоненты - колонки таблицы. Колонки будут соответствовать колонкам запроса и позволят дополнительно кастомизировать вывод данных.

8.11.2 Параметры региона

Отличительной особенностью компонента типа «Отчёт» является необходимость обязательного заполнения поля Источник данных - SQL. Необходимо указать SQL-запрос, на основе которого формируется список колонок отчета и выбираются данные для вывода на форму.

8.11.3 Атрибуты региона

Группа специальных атрибутов, характерных для данного вида регионов.

Атрибут	Тип	Описание
Включить поиск	Переключатель	Добавляет элемент поиска по отчёту. Текстовое поле ввода и выпадающий список с возможностью выбрать столбцы репорта для поиска по ним.
Включить разбиение на страницы	Переключатель	Добавляет пагинации - возможность просматривать страницы далее первой.
Включение заголовков	Переключатель	Добавляет заголовки столбцам отчёта.
Действия	Переключатель	Добавляет интерактивные действия для пользователя. Включает в себя возможность выбора колонок, задание последовательности колонок, подсветки, фильтрации и выбора пользовательского шаблона.
Сохранять шаблон	Переключатель	Добавляет возможность сохранять пользовательские отчеты как непубличные шаблоны, которые доступны только пользователю который создал шаблон. Доступно только при выборе «Действия» - «Да».
Сохранять публичный шаблон	Переключатель	Добавляет возможность сохранять пользовательский отчет в виде публичного отчета, который будет доступен всем пользователям. Доступен при выборе «Действия» - «Да».
Авторизация для сохранения публичного отчета.	Список	Позволяет указать схему авторизации, которую должен пройти пользователь для получения доступа к функционалу сохранения своего шаблона как публичного.
Шаблон	Список	Задаёт базовый шаблон отчёта. На текущий момент существует два варианта шаблонов: • Стандарт • Столбец - значение
Ограничение строк	Число	Устанавливает количество строк отчёта на одну страницу пагинации.
Зебра	Переключатель	Переключает отображение акцентуации четных и нечетных строк таблицы.
Подсветка	Переключатель	Отвечает за отображение подсветки строки при наведении на неё курсором.
Отступ в ячейках	Переключатель	Добавляет или убирает отступ в ячейках от содержимого.
Границы строк	Переключатель	Отвечает за отображение визуального разделителя строк таблицы.
Перенос строк в заголовке	Переключатель	Отвечает за перенос строки текста в заголовке при условии, что длина текста не может быть помещена в ячейку заголовка без переноса.
Перенос строк в заголовке	Переключатель	Отвечает за перенос строки текста в ячейках при условии, что длина текста не может быть помещена в ячейку строки без переноса.
Разрешить выгрузку	Переключатель	Позволяет сохранить отображаемый на страницы отчет в формате CSV либо Excel. Сохраненный отчет будет включать в себя фильтры определенные шаблоном.
Формат выгрузки	Множественный выбор	Позволяет указать формат в который можно выгрузить отчет. Для выбора доступны значения: CSV, Excel. Возможность выгрузки в Excel определяется наличием действующего модуля xReports + установленной ссылкой на сервер отчетов в настройках приложения.
8.11. Разделитель CSV Отчет	Текстовое поле	Определяет символ который будет использован в качестве разделителя для значений колонок в CSV документе. Доступно только при выборе «Формат выгрузки» - «CSV».
Кодировка	Список	Определяет кодировку в которой будет выгружен файл CSV. Доступно только при выборе «Формат выгрузки» - «CSV».

8.11.4 Параметры столбцов

Сгенерированный список столбцов отобразится в дереве объектов сразу после успешной валидации запроса в поле Источник данных SQL. Каждый из столбцов является самостоятельным объектом с индивидуальными параметрами. В таблице перечислен полный набор параметров компонента столбец отчёта.

Параметр	Тип	Описание
Имя столбца	Текст (только чтение)	Наименование столбца. Определяется алиасом столбца запроса источника данных. Поле не редактируется в настройках параметров - оно всегда соответствует наименованию столбца в запросе. Ссылка на значение столбца происходит по этому полю.
Тип	Список	Определяет набор параметров столбца и его поведение при выводе на страницу. Может принимать следующие значения: • Текст. Простой вывод значения информации из БД. • Ссылка. Позволяет нажатием на элемент внутри ячейки столбца перенаправить пользователя на другую страницу, либо вызвать динамический скрипт. • Скрытый. Столбец репорта будет скрыт в пользовательском интерфейсе. Его значение (например, для передачи в параметры ссылки) будет всё ещё доступно • Флажок. Столбец представляет собой группу чекбоксов для множественного выбора значений отчёта.
Связанный элемент	Текст / Окно конструктора	Позволяет выбрать элемент ввода, либо ввести глобальную переменную, куда будут переданы значения из ячеек отмеченных чекбоксом строк. Параметр применим для типа столбца - Флажок.
Именованные параметры	Переключатель	Флаг, который указывает необходимость воспринимать входящие параметры как часть текста sql запроса. При установке в false, разработчику потребуется явно указать входящие параметры. Отображается для источника данных отличных от типа Oracle, Mssql.
Заголовок	Текст	Заголовок столбца для вывода в пользовательском интерфейсе на странице приложения.
Порядковый номер	Число	Определяет порядок столбцов. По умолчанию равен порядку столбцов в запросе при первой генерации. При добавлении столбца в запрос в дальнейшем, новому будет присвоен последний номер, вне зависимости от позиции в запросе. ВНИМАНИЕ! Для отчета порядок вывода столбцов определяется в пользовательском интерфейсе через функционал «Действия».
Выравнивание	Переключатель	Выравнивание текста-содержимого столбца. По левому краю, по центру, или по правому краю.
Ссылка	Текст / Окно конструктора	Позволяет задать ссылку на другую страницу, вызвать динамический скрипт, либо настроить в конструкторе ссылку на страницу внутри приложения. В последнем случае можно выбирать значения не только полей ввода, но и столбцов отчёта. Вместо текста ссылки можно также передать значение строки репорта - такой подход позволяет настраивать ссылку внутри sql-запроса. Параметр применим для Тип столбца - Ссылка.

продолжается на следующей странице

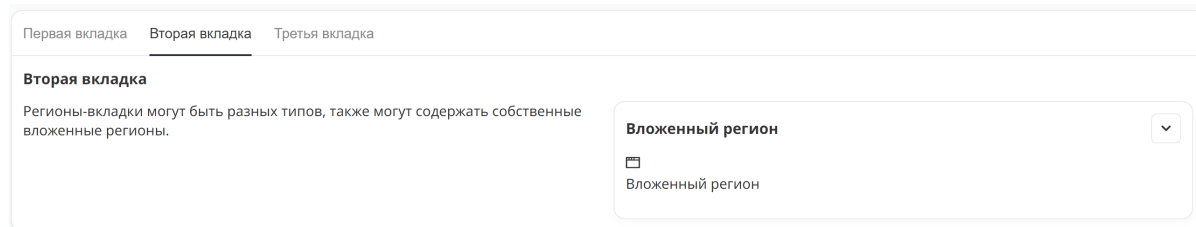
Таблица 2 – продолжение с предыдущей страницы

Параметр	Тип	Описание
Значок ссылки	Текст	Позволяет выбрать иконку для вывода в ячейке столбца. Вместо текста ссылки можно также передать значение строки репорта - такой подход позволяет выбрать картинку внутри sql-запроса. Параметр применим для Тип столбца - Ссылка.
Ширина	Текст	Фиксирует ширину столбца.
Формат маски	Текст	Форматирует вывод наложением маски по правилам форматирования SQL-функции <code>to_char</code> для Postgresql и Oracle, <code>FORMAT</code> для mssql, <code>DATE_FORMAT</code> для mysql для типа данных DATE и <code>FORMAT</code> для типа данных NUMBER. Для firebird необходимо указывать пользовательскую функцию форматирования данных - вызов функции будет происходить с подстановкой функции в SQL.
Включить сортировку	Переключатель	Разрешает сортировать строки по значению столбца.
Поиск по столбцам	Переключатель	Разрешает осуществлять поиск по значению столбца, если поиск включен в атрибутах отчёта.
Выражение HTML	Область текста	Позволяет определить формат вывода в виде HTML. Можно сослаться на значение столбца репорта.
Схема авторизации	Список	Определяет схему авторизации которую необходимо выполнить для данного компонента. В случае успешного выполнения компонент будет отображен на страницу.
Условия отображения - Тип	Список	Определяет тип условия отображения. Подробнее см. в <i>условиях выполнения</i> .

8.12 Вкладки

Вкладки – контейнер для группировки дочерних регионов на переключаемых вкладках. После создания на странице компонента с типом «Вкладки» необходимо добавить дочерние регионы с помощью контекстного меню. Каждый дочерний регион отображается в виде вкладки.

На вкладках отобразятся заголовки дочерних регионов, даже если для самих регионов заголовков отключен.

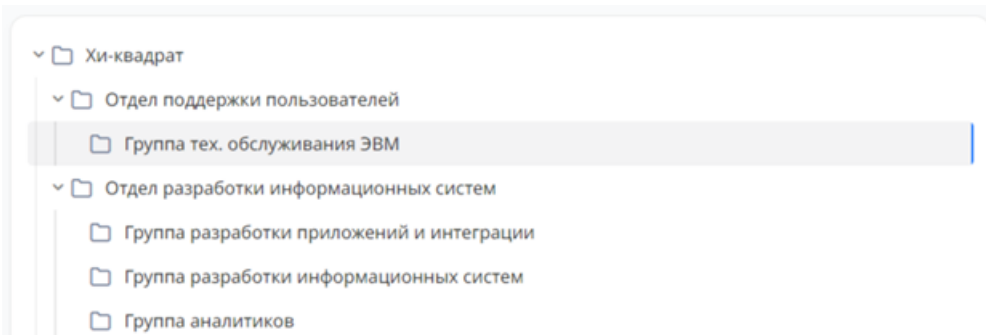


8.12.1 Настройки компонента

Параметры компонента стандартные для региона.

8.13 Дерево

Дерево – регион для вывода древовидной структуры данных. В качестве источника данных используется SQL-запрос. Система автоматически построит древовидную структуру по указанной выборке, при условии наличия ссылок на родительский ключ.



Пример SQL-запроса для компонента «Дерево»:

```
SELECT
  t.id,      -- ID элемента
  t.id_fk,   -- ID родительского элемента
  t.title,   -- Подпись элемента
  t.link     -- Ссылка, по которой будет осуществлен переход после нажатия на элемент
FROM
  dep.f_get_tree_with_scope(1) t;
```

8.13.1 Настройки компонента

Доступны настройки двух уровней: параметры региона и атрибуты дерева.

8.13.2 Параметры региона

Параметры компонента стандартные для региона.

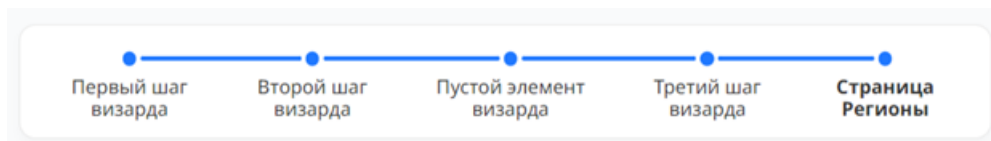
8.13.3 Атрибуты региона

Атрибут	Тип	Описание
Столбец РК	Список	Идентификатор узла. Из выпадающего списка необходимо выбрать один из столбцов запроса. Столбец запроса должен представлять из себя идентификатор узла, на который могут ссылаться дочерние строки.
Столбец FK	Список	Идентификатор родительского узла. Из выпадающего списка необходимо выбрать один из столбцов запроса.
Колонка заголовка	Список	Отображаемый заголовок узла. Из выпадающего списка необходимо выбрать один из столбцов запроса.
Колонка с ссылкой	Список	Колонка запроса которая содержит ссылку, по которой необходимо перейти при нажатии на имя узла.
Ссылка	Текст / Окно кон- структо- ра	Ссылка куда будет произведено перенаправление после нажатия на узел дерева. Можно указать если не указан атрибут «Колонка с ссылкой».
Высота области прокрутки	Текстовое поле	Определяет высоту компонента. Указывается в % или px.
Связанный элемент выбранного узла	Выбор элемента	Определяет элемент который будет указывать значение текущего выбранного узла. Если элемент указан и на момент загрузки страницы имеет значение, активный узел дерева будет соответствовать значению элемента. Такое же поведение будет при вызове jsAPI.component.refresh. Данный атрибут удобно использовать, если необходим более тонкий контроль выбора узла.
Сообщение об отсутствии данных	Текстовое поле	Определяет сообщение пользователю, которое будет выведено в случае отсутствия данных.

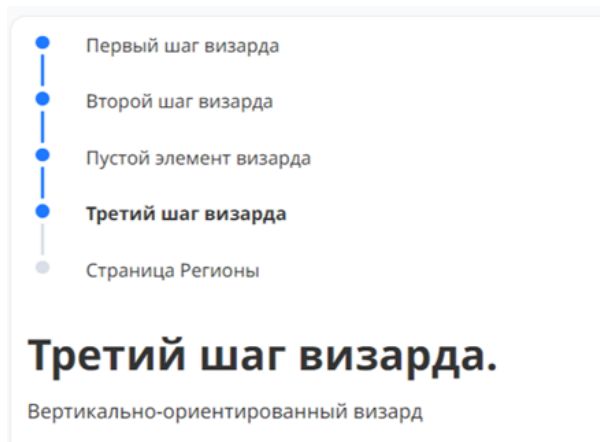
8.14 Визард (Wizard)

Wizard – компонент для графического представления последовательности действий на основе списка. Компонент поддерживает два вида направленности:

Горизонтальная



Вертикальная



8.14.1 Настройка компонента

Доступны настройки двух уровней: параметры региона и атрибуты визарда.

8.14.2 Параметры региона

Параметры компонента стандартные для региона.

8.14.3 Атрибуты региона

Атрибут	Тип	Описание
Разрешить переход по нажатию	Переключатель	Определяет, будет ли осуществляться переход на связанную страницу при клике на узле визарда.
Отображать шаги	Список	Доступны несколько вариантов отображения заголовков шагов визарда: • Все. Заголовки каждого элемента списка будут отображаться на визарде • Текущий. На визард будет выведен только заголовок текущего шага, остальные будут безымянными узлами. • Никогда. На визард не будут выводиться заголовки узлов.
Направление	Список	Пространственная направленность визарда. На выбор два варианта: • Горизонтально • Вертикально

8.15 Календарь

Календарь — компонент для отображения данных с меткой времени начала и конца события в виде интерактивного календаря. Источником данных для компонента является результат SQL-запроса.

Пример SQL-запроса

```
SELECT
  c.id,           -- ID строки для добавления в ссылку
  c.title,        -- Заголовок события
  c.start_ev,     -- Начало периода события
  c.end_ev,       -- Окончание периода события
```

(продолжается на следующей странице)

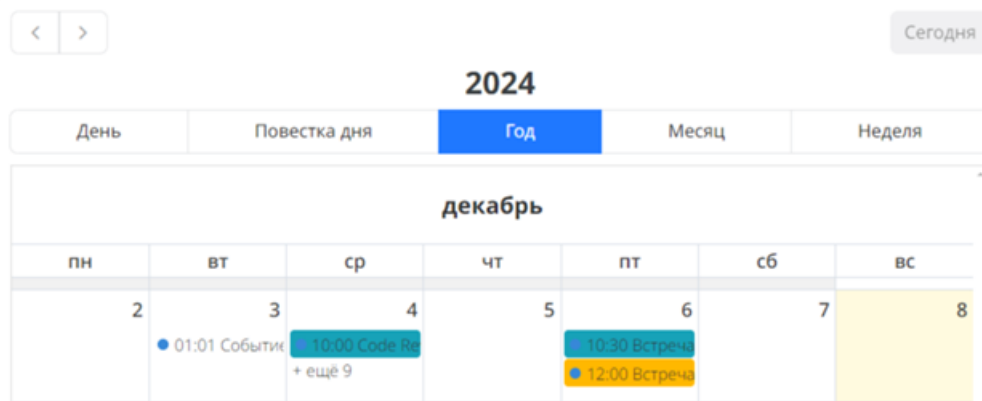
(продолжение с предыдущей страницы)

```

c.color_class -- HEX-код цвета
FROM
events.t_calendar c;

```

Сам календарь можно настроить, изменив отображение выходных, выбрав необходимые для показа представления (День, Неделя, Месяц, Год, Повестка дня), а также высоту календаря и отображение подсказки о событии при наведении курсора. Событию календаря можно установить цвет и ссылку, по которой будет произведен переход после нажатия (в ссылку можно добавить id записи).



8.15.1 Настройка компонента

Доступны настройки двух уровней: параметры региона и атрибуты календаря.

8.15.2 Параметры региона

Параметры компонента стандартные для региона.

8.15.3 Атрибуты региона

Атрибут	Тип	Описание
Колонка с заголовком	Список	Один из столбцов источника данных SQL можно назначить в качестве заголовка отображаемого на календаре события.
Колонка с началом периода	Список	Один из столбцов источника данных SQL можно назначить в качестве времени начала отображаемого на календаре события.
Колонка с окончанием периода	Список	Один из столбцов источника данных SQL можно назначить в качестве времени окончания отображаемого на календаре события.
CSS колонка	Список	Один из столбцов источника данных SQL можно назначить для определения стиля отображаемого на календаре события.
Высота	Число	Размер компонента по вертикали.
Представления	Множественный выбор	Перечень доступных уровней представления календаря. Выбранные варианты будут предложены пользователю в интерфейсе компонента. Список представлений: • Год • Месяц • Неделя • День • Повестка дня
Отображать время	Переключатель	Если установлен, отображаться будет не только дата, но и время события.
Показывать подсказку	Переключатель	Если установлен, наименование события будет отображаться на всплывающей подсказке, при наведении на ячейку в календаре.
Отображать выходные	Переключатель	Если установлен, будут отображаться все дни недели.
Ссылка на просмотр	Текст / Окно конструктора	Определяет ссылку, на которую будет перенаправлен пользователь при клике на событие в календаре.

8.16 DataGrid

DataGrid – интерактивный компонент, аналогичный «Отчет», но с возможностью редактирования данных. Записи в DataGrid можно добавлять, изменять и удалять. Внимание!!! Для осуществления редактирования необходимо указать первичный ключ в настройках колонки компонента.

🔍

Поиск: все столбцы

>

⌵

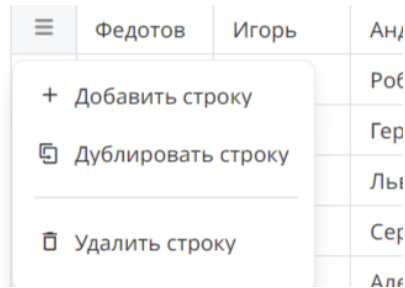
⌶

+

Сохранить

☰	Фамилия	Имя	Отчество	Телефон	Дата ро...	Электр...	Тип кли...	☐
☰	Горелов	Виктор	Лукич	899912...	07.08.1...	info@xs...	Физ. ли...	☒
☰	Романов	Кирилл	Фёдоро...	899912...	23.07.1...	info@xs...	Физ. ли...	☐
☰	Калмык...	Илья	Роберт...	899912...	14.12.1...	info@xs...	Физ. ли...	☐

DataGrid предоставляет возможность манипулировать данными строк - вы можете не только удалять и дублировать строки, но также осуществлять ручную выборку с возможностью переноса данных в популярные редакторы типа Excel.



8.16.1 Настройка компонента

Доступны настройки двух уровней: параметры региона и атрибуты таблицы.

8.16.2 Параметры региона

Параметры компонента стандартные для региона.

8.16.3 Атрибуты региона

Атрибут	Тип	Описание
Редактирование - разрешено	Переключатель	Определяет возможность редактирования содержимого таблицы.
Разрешенные операции	Множественный выбор	Список доступных операций в таблице: - Вставка - Обновление - Удаление Настройка разрешенных операций доступна если разрешено редактирование.
Колонка операций со строкой	Список	Позволяет указать колонку таблицы, которая будет определять какие операции доступны для строки. Если значение колонки содержит символ «U» - данную строку разрешено обновлять. Если значение колонки содержит символ «D» - данную строку разрешено удалять. Если содержать «UD» в любой последовательности - разрешены обе операции.
Потерянные обновления	Список	Атрибут определяет способ, которым предотвращается потеря данных при совместной работе нескольких пользователей с одним и тем же массивом данных. Доступные значения: Значение строки. Перед отправкой изменений система вычисляет хэш измененной строки и сравнивает с хэшем, полученном при загрузке данных. Если хэши расходятся, система не позволит произвести обновление (кто-то другой уже поменял строку).
Авторизация на редактирование	Список	Определяет какая схема авторизации пользователей используется для допуска к выбранной операции в компоненте.
Отобразить NULL как	Текст	Определяет, что будет выведено в ячейке таблицы в случае отсутствия значения (Null).
Панель инструментов – Показывать панель	Переключатель	Определяет отображение панели инструментов компонента.
Панель инструментов – Кнопки панели	Множественный выбор	Определяет набор кнопок на панели инструментов, если включено ее отображение. Доступны следующие компоненты панели инструментов: - Столбцы поиска - Поле поиска - Действия - Сохранить
Разрешить выгрузку	Переключатель	Позволяет сохранить отображаемый на страницы отчет в формате CSV либо Excel. Сохраненный отчет будет включать в себя фильтры определенные шаблоном.
Формат выгрузки	Множественный выбор	Позволяет указать формат в который можно выгрузить отчет. Для выбора доступны значения: CSV, Excel. Возможность выгрузки в Excel определяется наличием действующего модуля xReports + установленной ссылкой на сервер отчетов в настройках приложения.
Разделитель CSV	Текстовое поле	Определяет символ который будет использован в качестве разделителя значений колонок в CSV документе. Доступен только при выборе «Формат выгрузки» - «CSV».
Кодировка	Список	Определяет кодировку в которой будет выгружен файл CSV. Доступен только при выборе «Формат выгрузки» - «CSV».
Имя файла	Текстовое поле	Позволяет ввести название файла выгрузки отчета. Расширение файла будет подставлено автоматически.
8116. Таблица DataGrid	Текст / Окно конструктора	Позволяет указать шаблон по умолчанию.

8.16.4 Параметры столбцов

Сгенерированный список столбцов отобразится в дереве объектов сразу после успешной валидации запроса в поле Источник данных SQL. Каждый из столбцов является самостоятельным объектом с индивидуальными параметрами. Также DataGrid позволяет добавить специальные служебные колонки. В таблице перечислен полный набор параметров компонента столбец DataGrid.

Параметр	Тип	Описание
Имя столбца	Текст (только чтение)	Наименование столбца. Определяется алиасом столбца запроса источника данных. Поле не редактируется в настройках параметров - оно всегда соответствует наименованию столбца в запросе. Ссылка на значение столбца происходит по этому полю.
Тип	Список	Определяет набор параметров столбца и его поведение при выводе на страницу. Может принимать следующие значения: • Выбор строк. Служебный тип колонки DataGrid, которая не привязана к колонке из SQL. Добавляет функциональную возможность выбрать строку или несколько строк для совершения массовых операций с ними. • Действия. Служебный тип колонки DataGrid. Позволяет отобразить контекстное меню для работы со строкой. • Текст. Данный тип колонки позволяет вводить и отображать текстовые данные. Принцип действия схож с текстовым элементом формы. • Числовой. Позволяет отобразить и форматировать числовые данные. Принцип действия схож с числовым элементом формы. • Скрытый. Столбец будет скрыт в пользовательском интерфейсе. Его значение (например, для передачи в параметры ссылки) будет всё ещё доступно. • Список. Столбец позволяет выбрать значение из фиксированного списка, определяемого sql запросом, который указывается в настройках компонента. Принцип действия аналогичен списочному элементу формы. • Дата. Данный тип предоставляет возможности для работы с данными типа «Дата». При редактировании информации пользователю будет доступен интерактивный выбор даты из календаря, аналогично элементу формы «Дата». • Множественный выбор. Столбец позволяет выбрать несколько значений из списка. Аналогичен элементу «Множественный выбор» формы. • Радиокнопки. Столбец отображает данные в виде радиокнопок. Принцип действия аналогичен элементу «Радиокнопки» формы. • Флажок. Столбец представляет собой группу чекбоксов для множественного выбора значений. • Переключатель. Столбец представляет элемент бинарного выбора значения. • Ссылка. Позволяет нажатием на элемент внутри ячейки столбца перенаправить пользователя на другую страницу, либо вызвать динамический скрипт.
Именованные параметры	Переключатель	Флаг, который указывает необходимость воспринимать входящие параметры как часть текста sql запроса. При установке в false, разработчику потребуется явно указать входящие параметры. Отображается для источника данных отличных от типа Oracle, Mssql.

продолжается на следующей странице

Таблица 3 – продолжение с предыдущей страницы

Параметр	Тип	Описание
Заголовок	Текст	Заголовок столбца для вывода в пользовательском интерфейсе на странице приложения.
Выравнивание заголовка	Переключатель	Выравнивание текста заголовка столбца. По левому краю, по центру, или по правому краю.
Порядковый номер	Число	Определяет порядок столбцов. По умолчанию равен порядку столбцов в запросе при первой генерации. При добавлении столбца в запрос в дальнейшем, новому будет присвоен последний номер, вне зависимости от позиции в запросе. ВНИМАНИЕ! Для отчета порядок вывода столбцов определяется в пользовательском интерфейсе через функционал «Действия».
Выравнивание	Переключатель	Выравнивание текста-содержимого столбца. По левому краю, по центру, или по правому краю.
Ссылка	Текст / Окно конструктора	Позволяет задать ссылку на другую страницу, вызвать динамический скрипт, либо настроить в конструкторе ссылку на страницу внутри приложения. В последнем случае можно выбирать значения не только полей ввода, но и столбцов отчёта. Вместо текста ссылки можно также передать значение строки репорта - такой подход позволяет настраивать ссылку внутри sql-запроса. Параметр применим для Тип столбца - Ссылка.
Значок ссылки	Текст	Позволяет выбрать иконку для вывода в ячейке столбца. Вместо текста ссылки можно также передать значение строки репорта - такой подход позволяет выбрать картинку внутри sql-запроса. Параметр применим для Тип столбца - Ссылка.
Формат маски	Текст	Форматирует вывод наложением маски по правилам форматирования SQL-функции <code>to_char</code> для Postgresql и Oracle, <code>FORMAT</code> для mssql, <code>DATE_FORMAT</code> для mysql для типа данных <code>DATE</code> и <code>FORMAT</code> для типа данных <code>NUMBER</code> . Для firebird необходимо указывать пользовательскую функцию форматирования данных - вызов функции будет происходить с подстановкой функции в SQL.
Не пустой	Переключатель	Если DataGrid является редактируемым, добавляет встроенную проверку на наличие данных при обновлении или вставки строки.
Базовый тип	Список	Определяет базовый типа данных колонки. Используется для определения набора операций, доступных для интерактивного взаимодействия с колонкой.
Колонка с данными	Список	Атрибут проставляет соответствие между колонкой и данными из запроса. Указывается наименование колонки значение которой будет выступать в виде источника данных для элемента.
Первичный ключ	Переключатель	Определяет является ли указанный элемент первичным ключом. Если DataGrid является редактируемым необходимо указать хотя бы один элемент, «колонка с данными» которого соответствует первичному ключу и может быть использована для явной идентификации строки.

продолжается на следующей странице

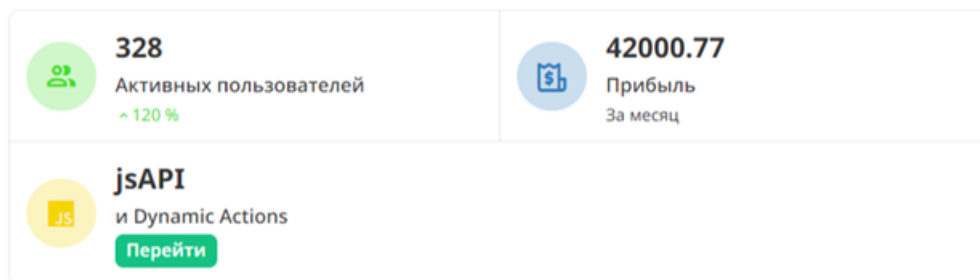
Таблица 3 – продолжение с предыдущей страницы

Параметр	Тип	Описание
Значение по умолчанию	Список	Выберите значение по умолчанию. Тип определяет откуда брать значение. Доступные значения: - Статический - введите значение по умолчанию, которое будет отображаться, если значение сессии для данного элемента не задано - SQL - введите sql выражение, которое вернет значение для элемента - Элемент - значение будет взять из значения элемента приложения, записанного в сессию.
Схема авторизации	Список	Определяет схему авторизации которую необходимо выполнить для данного компонента. В случае успешного выполнения компонент будет отображен на страницу.
Условия отображения - Тип	Список	Определяет тип условия отображения. Подробнее см. в <i>условиях выполнения</i> .
Только чтение	Список	Определяет условия вывода элемента, как доступного только для чтения.

8.17 Плитки

Плитки – служит для отображения статистической информации. Источником данных для компонента «Плитки» является результат SQL-запроса. Для настройки внешнего вида плиток используются определенные параметры SQL-запроса, например:

```
SELECT
'42000.77' title,      -- Главная подпись (заголовок)
'Прибыль' text,       -- Подпись, описывающая заголовок
'#2C7ABB' color,      -- HEX-код цвета оформления плитки
'За месяц' descr,     -- Вспомогательная подпись под основным текстом
'uil uil-bill' icon   -- CSS класс иконки
```



8.17.1 Настройка компонента

Доступны настройки двух уровней: параметры региона и атрибуты таблицы.

8.17.2 Параметры региона

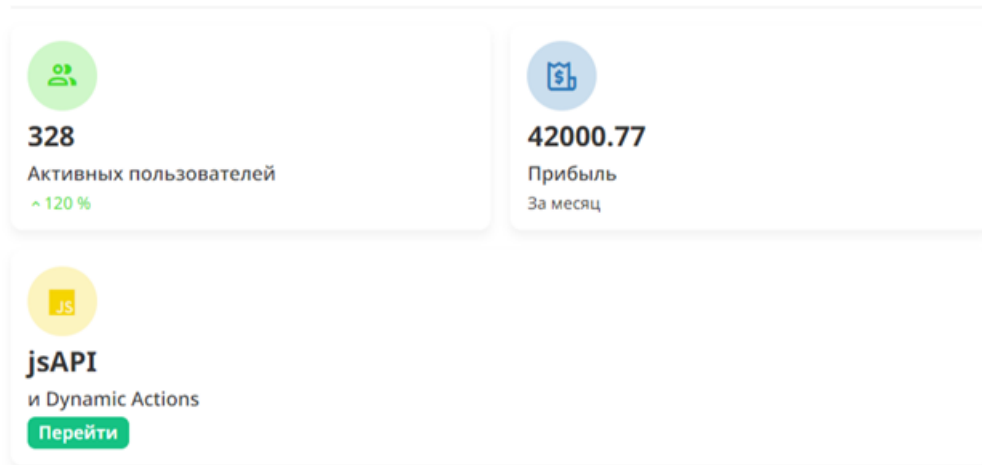
Отличительной особенностью компонента типа «Плитки» является необходимость обязательного заполнения поля Источник - SQL. Необходимо указать SQL-запрос, на основе которого формируется список и внешний вид плиток.

8.17.3 Атрибуты региона

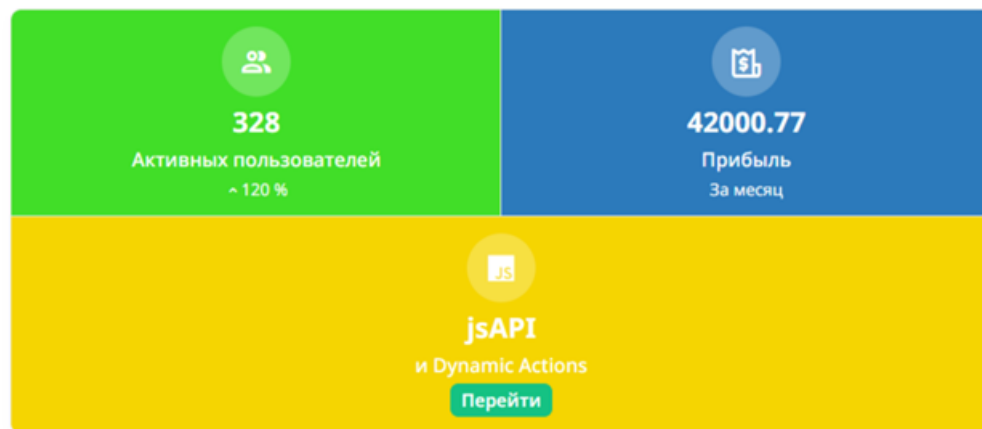
Атрибут	Тип	Описание
Тема	Список	Определяет predetermined тему отображения плиток: • Классическая - отображает плитки с иконкой сверху. Цветовая настройка в данном варианте определяет цвет иконки. • Контрастная – аналогична классической, но цветовая настройка определяет цвет фона плитки. • Горизонтальная - компактное отображение с иконкой слева. Цветовая настройка определяет цвет иконки. • Горизонтально-контрастная – компактное отображение, где цветовая настройка определяет фон плитки
Колонки	Список	Определяет настройки сетки для отображения компонента. При указании параметра «auto» компонент самостоятельно распределит все элементы по строкам. Ширина каждого элемента в строке будет одинакова. Параметр «auto-float» работает также как и «auto», но ширина каждого элемента будет соответствовать содержанию компонента.
Вид	Список	Определяет внешний вид компонента. • Сетка – плоское отображение без промежутков между плитками. • Раздельный – отображение плиток с промежутками и тенями
Выравнивание	Переключатель	Определяет тип выравнивания содержимого в плитке: • По левому краю • По центру • По правому краю
Контрастные иконки	Переключатель	Определяет контрастный режим отображения. В контрастном режиме иконки отображаются белым цветом на цветном фоне.
Колонка с заголовком	Список	Определяет наименование параметра для отображения заголовка плитки.
Колонка с текстом	Список	Определяет наименование параметра для отображения основного текста плитки.
Колонка с описанием	Список	Определяет наименование параметра для отображения дополнительного текста плитки.
Колонка с иконкой	Список	Определяет наименование параметра для отображения иконки плитки.
Колонка с цветом	Список	Определяет наименование параметра для управления цветом плитки. Необходимо указывать цветовое значение в формате hex.

Для плиток реализовано несколько разных тем, по аналогии с компонентом «Карточки». Для текста внутри плитки также можно выбрать разные варианты выравнивания: слева, справа и по центру. Помимо тем отображения плиток есть также два вида расположения плиток: в виде цельной сетки элементов и в виде отдельных элементов. Для наглядности возможные варианты – изображены на скриншотах:

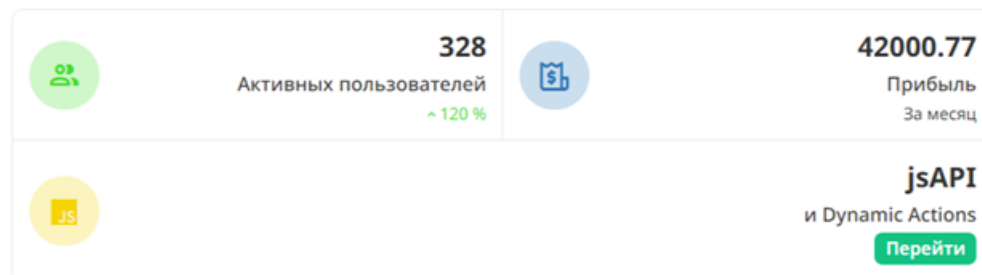
Классическая тема в 3 колонки с раздельными плитками



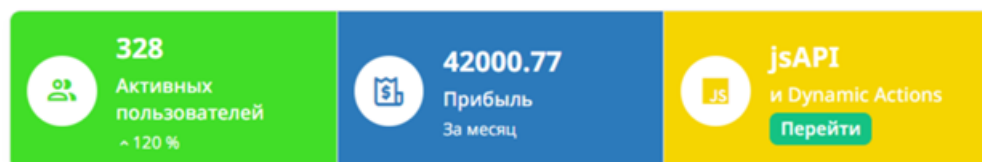
Контрастная тема в 2 колонки с выравнением по центру



Горизонтальная тема в 1 колонку с выравнением справа

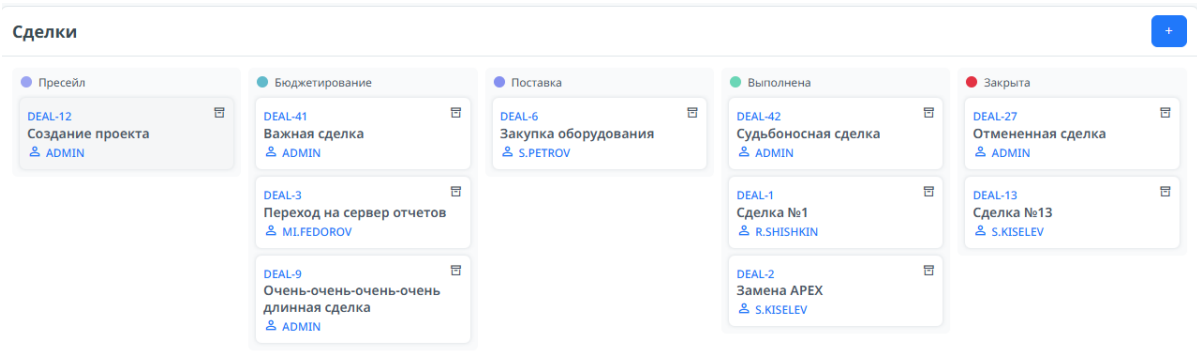


Горизонтально-контрастная тема с контрастными иконками



8.18 Канбан

Канбан – интерактивный компонент, который позволяет визуализировать данные в виде карточек, разбитых на колонки. Компонент подходит для решения задач связанных с управлением рабочими процессами.



Компонент позволяет настроить внешний вид карточек, а также внедрить динамическую систему колонок для отображения. Логика перемещения карточек управляется специальным процессом Канбан.

8.18.1 Настройка компонента

Доступны настройки трех уровней: параметры региона, атрибуты канбан и настройки элементов карточки.

8.18.2 Параметры региона

Для создания компонента Канбан необходимо выбрать тип региона - «Канбан» и ввести SQL код в поле источник данных. После успешной валидации SQL кода система сформирует перечень элементов карточки. Каждый столбец SQL запроса будет соответствовать визуальному элементу выводимой на экран карточки, а содержимое карточки будет содержать значения строки запроса.

8.18.3 Атрибуты региона

Настройки колонок доски канбан осуществляются на вкладке атрибуты региона. Канбан требует указать второй SQL запрос на основании которого будут формироваться колонки доски. Визуальные настройки колонок осуществляются за счет проставления соответствия настройки компонента и колонки запроса. Настройки включают в себя:

Атрибут	Тип	Описание
Ключ колонки в карточке	Список	Атрибут в котором необходимо указать наименование колонки из запроса в параметрах региона. По данной колонке будет определяться к какой колонки доски канбан относится карточка. Значение колонки должно соответствовать значению указанному в атрибуте «Идентификатор колонки» для построения успешной связи.
Редактирование - разрешено	Переключатель	Определяет возможность редактирования содержимого таблицы.
Разрешенные операции	Множественный выбор	Список доступных операции в таблице: Обновление Настройка разрешенных операций доступна если разрешено редактирование.
Потерянные обновления	Список	Атрибут определяет способ, которым предотвращается потеря данных при совместной работе нескольких пользователей с одним и тем же массивом данных. Доступные значения: Значение строки. Перед отправкой изменений система вычисляет хэш измененной строки и сравнивает с хэшем, полученном при загрузке данных. Если хэши расходятся, система не позволит произвести обновление (кто-то другой уже поменял строку).
Авторизация на редактирование	Список	Определяет какая схема авторизации пользователей используется для допуска к выбранной операции в компоненте.
Именованные параметры	Переключатель	Флаг, который указывает необходимость воспринимать входящие параметры как часть текста sql запроса. При установке в false, разработчику потребуется явно указать входящие параметры. Отображается для источника данных отличных от типа Oracle, Mssql.
Тип запроса	Список	Определяет тип запроса который будет выполнен для построения колонок доски канбан. Для Oracle доступен SQL и PL/SQL Block, для остальных типов источников данных - SQL.
SQL	Текстовая область	Указывается запрос который будет формировать колонки доски канбан. Необходимо указать как минимуму 2 колонки: идентификатор колонки и её название.
Идентификатор колонки	Список	Указывается наименование колонки из SQL запроса, которая является идентификатором. Значение данной колонки будет использовано для связи карточек с колонками канбан доски.
Заголовок колонки	Список	Указывается наименование колонки из SQL запроса, которая будет содержать значение для заголовка колонки доски канбан.
Цвет колонки	Список	Позволяет задать цвет колонки канбан. Значение указанной колонки должно содержать значение цвета указанного в hex формате.

8.18.4 Настройки элементов карточки

Сгенерированный список элементов карточки отобразится в дереве объектов сразу после успешной валидации запроса в поле Источник данных SQL. Каждый столбец SQL запроса будет соответствовать определенному элементу карточки, что позволяет гибко настроить внешний вид и функциональность карточки.

Параметр	Тип	Описание
Имя	Текст	Отображаемое имя элемента карточки в среде разработки. По умолчанию соответствует псевдониму столбца.
Тип	Список	Определяет типа элемента карточки. Может принимать следующие значения: - Заголовок. Значение колонки из запроса будет представлено в виде заголовка карточки. Заголовок карточки выделяется жирным(bold) шрифтом. - Текст. Значение колонки выводится в виде простого текста. - Ссылка. Позволяет нажатием на элемент внутри карточки перенаправить пользователя на другую страницу, либо вызвать динамический скрипт. - Скрытый. Столбец будет скрыт в пользовательском интерфейсе. Его значение (например, для передачи в параметры ссылки) будет всё ещё доступно. - Иконка. Позволяет задать иконку для отображения внутри карточки. Значение столбца должно содержать css класс иконки.
Именованные параметры	Переключатель	Флаг, который указывает необходимость воспринимать входящие параметры как часть текста sql запроса. При установке в false, разработчику потребуется явно указать входящие параметры. Отображается для источника данных отличных от типа Oracle, Mssql.
Порядковый номер	Число	Определяет порядок элемента. По умолчанию равен порядку столбцов в запросе при первой генерации. При добавлении столбца в запрос в дальнейшем, новому будет присвоен последний номер, вне зависимости от позиции в запросе. Изменение порядкового номера меняет положение элемента карточки в пользовательском интерфейсе.
Начать новую строку	Переключатель	Позволяет отделить элемент карточки от предыдущего разделив их «новой строкой».
Формат маски	Текст	Форматирует вывод наложением маски по правилам форматирования SQL-функции to_char для Postgresql и Oracle, FORMAT для mssql, DATE_FORMAT для mysql для типа данных DATE и FORMAT для типа данных NUMBER. Для firebird необходимо указывать пользовательскую функцию форматирования данных - вызов функции будет происходить с подстановкой функции в SQL.
Ссылка	Текст / Окно кон- структо- ра	Позволяет задать ссылку на другую страницу, вызвать динамический скрипт, либо настроить в конструкторе ссылку на страницу внутри приложения. В последнем случае можно выбирать значения не только полей ввода, но и столбцов отчёта. Вместо текста ссылки можно также передать значение строки репорта - такой подход позволяет настраивать ссылку внутри sql-запроса. Параметр применим для Тип столбца - Ссылка.
Значок ссылки	Текст	Позволяет выбрать иконку для вывода в ячейке столбца. Вместо текста ссылки можно также передать значение строки репорта - такой подход позволяет выбрать картинку внутри sql-запроса. Параметр применим для Тип столбца - Ссылка.
Базовый тип	Список	Определяет базовый типа данных колонки. Используется для определения набора операций, доступных для интерактивного взаимодействия с колонкой.
Колонка с данными	Список	Атрибут проставляет соответствие между визуальным элементом и данными из запроса. Указывается наименование колонки значение которой будет выступать в виде источника данных для элемента.

продолжается на следующей странице

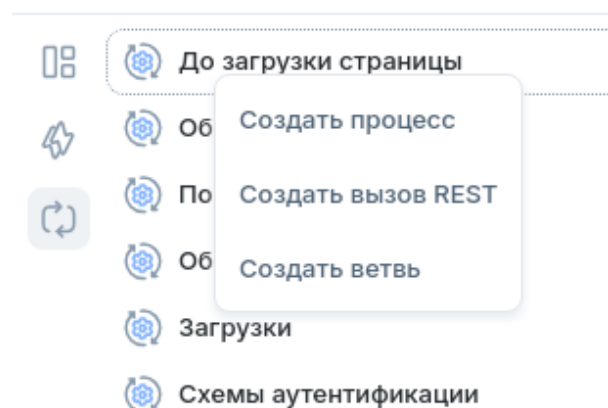
Таблица 4 – продолжение с предыдущей страницы

Параметр	Тип	Описание
Первичный ключ	Переключатель	Определяет является ли указанный элемент первичным ключом. Если канбан является редактируемым необходимо указать хотя бы один элемент, «колонка с данными» которого соответствует первичному ключу и может быть использована для явной идентификации строки.
Схема авторизации	Список	Определяет схему авторизации которую необходимо выполнить для данного компонента. В случае успешного выполнения компонент будет отображен на страницу.
Условия отображения - Тип	Список	Определяет тип условия отображения. Подробнее см. в <i>условиях выполнения</i> .

Процессинг

Для полноценной работы приложения необходимо не только выводить данные на формы и отчеты, но и обрабатывать данные, которые вводят пользователи. В данном разделе описывается блок «Процессинг» в редакторе страницы. Данный блок отвечает за обработку пользовательских данных на разных этапах работы со страницей.

На вкладке **Процессинг** отображается список обработчиков запросов, сгруппированных по виду процессов или типу вызова обработчика. Состав контекстного меню доступного по нажатию ПКМ определяется типом группы:



Процессинг предоставляет следующие виды обработчиков запросов:

- Процесс - это основной механизм взаимодействия приложения с базой данных. Выполнение процессов происходит при поступлении от приложения различных запросов (Request). При появлении запроса от приложения все процессы соответствующего типа выполняются в порядке очереди (Sequence).
- Ветвь – процесс по выполнению условия (Server-side Condition) которого произойдет перенаправление на другую страницу приложения или ссылку.
- Валидация – процесс проверки корректности введенных пользователем данных.

- Вызов REST - процесс который позволяет вызвать сторонний REST API. Позволяет приложению обрабатывать и осуществлять запросы к сторонним сервисам. При этом вызов будет осуществлять сервер приложений PGHS.
- Процесс Data Grid - процесс для обработки измененных данных в DATAGRID.
- Процесс Kanban - процесс для обработки изменений в компоненте Kanban.
- Загрузка – процесс, позволяющий загрузить файл из базы на клиент.
- Схема аутентификации – процесс аутентификации пользователя.

Процессы группируются также по типу процесса:

- До загрузки страницы – процессы, которые выполняются перед загрузкой страницы. Их основное предназначение – подготовить компоненты для взаимодействия с пользователем (например, заполнить поля формы значениями из БД). До загрузки страницы могут быть выполнены следующие виды обработчиков: процессы, вызовы REST и ветви.
- Обработка – процессы данного вида выполняются при отправке формы на сервер (submit).
- После обработки - процессы данного вида выполняются после обработки страницы. Например, переходы на другие страницы по нажатию кнопок.
- Обратный вызов Ajax - процессы данного вида позволяют взаимодействовать с БД по запросу от JS-скрипта.

Ответ от процессов Ajax должен обрабатываться разработчиком в JS-скрипте. Остальные виды процессов могут самостоятельно взаимодействовать с пользователем, сообщая об успешности выполнения запроса или о возникшей ошибке.

9.1 Процессы

Для создания обработчика типа процесс необходимо выбрать соответствующий пункт контекстного меню по ПКМ и заполнить параметры создаваемого процесса.

До загрузки страницы

Процессы

New process

Обработка

Валидации

New validation

Процессы

New process

После обработки

Обратный вызов Ajax

Загрузки

Схемы аутентификации

New process

Идентификация ▾

Имя *
New process

Параметры выполнения ▾

Тип процесса *
Перед загрузкой страницы

Источник данных *
DEFAULT_APP

Имя запроса

Принять модальный Да Нет

Фиксация TX * Да Нет

Параметр	Тип	Описание
Имя	Текст	Имя процесса.
Имя запроса	Строка	Определяет имя запроса. Если указано, то процесс будет выполняться только тогда, когда значение глобальной переменной REQUEST будет соответствовать введенному значению.
Тип процесса	Список	Тип процесса. Тип влияет на порядок вызова кода. В XRAD предусмотрены следующие типы процессов: • До загрузки страницы • Обработка • Обратный вызов Ajax
Источник данных	Список	Название источника данных для данного процесса.
Именованные параметры	Переключатель	Флаг, который указывает необходимость воспринимать входящие параметры как часть текста sql запроса. При установке в false, разработчику потребуются явно указать входящие параметры. Отображается для источника данных отличных от типа Oracle.
Принять модальный	Переключатель	Определяет необходимость закрытия модального окна после выполнения Процесса. Поле доступно только для модальных страниц.
Возвращаемые значения	Текст	Перечень имен элементов ввода, значения которых можно передать на родительскую форму после закрытия модального диалога. Поле доступно при выборе флага «Принять модальный».
Фиксация TX	Переключатель	Определяет необходимость вызывать COMMIT транзакции после успешного завершения процесса. После вызова COMMIT все изменения, внесенные в базу процессами, которые выполнялись до этого процесса, будут записаны в базу. После чего XRAD откроет новую транзакцию для выполнения последующих процессов.
Тип	Список	Список с возможным типом выполняемого кода.
Источник - SQL	Область текста	Определяет SQL-запрос, который необходимо выполнить.
Параметры запроса	Текст / Окно конструктора	Определяет элементы, которые необходимо передать в исполняемый запрос в качестве переменных.
Выходные параметры	Текст / Окно конструктора	Определяет выходные элементы. Если процессу необходимо установить значения сессионных переменных, необходимо указать их в данном поле.
Последовательность	Число	Определяет последовательность выполнения процесса в списке процессов одного типа.
Сообщение об успехе	Область текста	Определяет сообщение, которое будет отображено пользователю при успешном выполнении процесса. Если вызывается цепочка процессов - будет выведено последнее сообщение об успешном исполнении.
Сообщение об ошибке	Область текста	Определяет сообщение об ошибке.
Условия отображения - Тип	Список	Определяет тип условия отображения региона на странице. Подробнее см. в <i>условиях выполнения</i> .

Примечание

При появлении запроса от приложения сначала выполняются все удовлетворяющие условиям запроса глобальные процессы в соответствии с их порядковыми номерами, а затем процессы текущей открытой страницы.

9.2 Условия выполнения процесса

Для выполнения процесса есть 2 вида условий:

1. выполнение по запросу (Request);
2. условие на стороне сервера (Server-side condition).

9.3 Выполнение по запросу (Request)

При работе приложение постоянно посылает различные запросы. Будь то submit страницы или запрос из JS-скрипта. В зависимости от источника запроса исполняются процессы различного типа, но их всех объединяет вид исполнения – выполняются все процессы одного типа в порядке очереди. Для того, чтобы на запрос не были выполнены не нужные процессы используется поле Request Name. Если поле Request Name пустое, то процесс будет выполнен при любом запросе со стороны приложения, если тип процесса подходит под тип запроса. Однако, если указать конкретное имя запроса в поле Request Name, то процесс будет выполнен только в том случае, если имя запроса, посланного приложением, совпадет с именем в поле Request Name.

Того же эффекта можно достичь, используя условие на стороне сервера (Server-side Condition) с типом SQL Expression, а в качестве входного поля указать поле REQUEST.

9.4 Управление транзакцией

При выполнении цепочки запросов к базе данных иногда возникает ситуация, когда возникновение ошибки в очередном шаге не должно приводить к отмене результатов предыдущих шагов. Эта ситуация разрешается завершением транзакции при успешном завершении процесса. Чтобы завершить транзакцию необходимо процессу выставить флаг фиксации транзакции (Commit TX).

Если данный флаг не выставлен, то, в случае возникновения ошибки в любом из процессов, будут отменены все изменения, внесенные другими процессами, у которых так же не выставлен флаг завершения транзакции. Таким образом имеется возможность разбивать большое действие на группы более мелких действий, ошибки в которых не будут влиять на действия, выполненные в предыдущих группах.

9.5 Закрытие диалога

В работе приложений часто используются модальные окна для выполнения некоторых действий. Например: создание пользователя. При нажатии на кнопку “Создать” выполняется некоторый, привязанный к кнопке, процесс. В таких случаях задача модального окна выполнена и окно можно закрыть. Для облегчения задачи у процессов имеется специальный флаг. При выставленном флаге в случае успешного выполнения процесса модальное окно будет автоматически закрыто с результатом Акцепт.

При успешном выполнении на запрос нескольких процессов модальное окно будет закрыто автоматически только в случае, если у последнего из цепочки процессов выставлен флаг Принять модальный (Accept Modal).

9.6 Ветви

При работе приложения бывают ситуации, при которых необходимо произвести автоматический переход на некоторую страницу при загрузке выбранной или после выполнения какого-либо процесса. Для этих целей используются ветви - переходы между страницами (Branch).

Если ветвь расположена в блоке, то переход на указанную в поле Link страницу будет осуществлён при загрузке страницы, на которой расположен Branch при выполнении условия в блоке Server-side Condition. Для безусловного перенаправления в блоке Server-side Condition необходимо оставить значение Always, для отключения перенаправления – Never.

Если Branch расположен в блоке After Processing, то переход на указанную в поле Link страницу будет осуществлён при завершении всех процессов при выполнении условия в блоке Server-side Condition. Для безусловного перенаправления в блоке Server-side Condition необходимо оставить значение Always, для отключения перенаправления – Never.

New branch

Идентификация ▾

Имя *

New branch

?

Параметры выполнения ▾

Источник данных *

DEFAULT_APP

▾

?

Место выполнения *

Перед загрузкой стра...

▾

?

Макет ▾

Последовательность *

10

?

Поведение ▾

Тип *

URL-адрес перенапра...

▾

?

Ссылка ▾

Ссылка *

☰

?

Рассмотрим уникальные параметры для процессов типа ветвь:

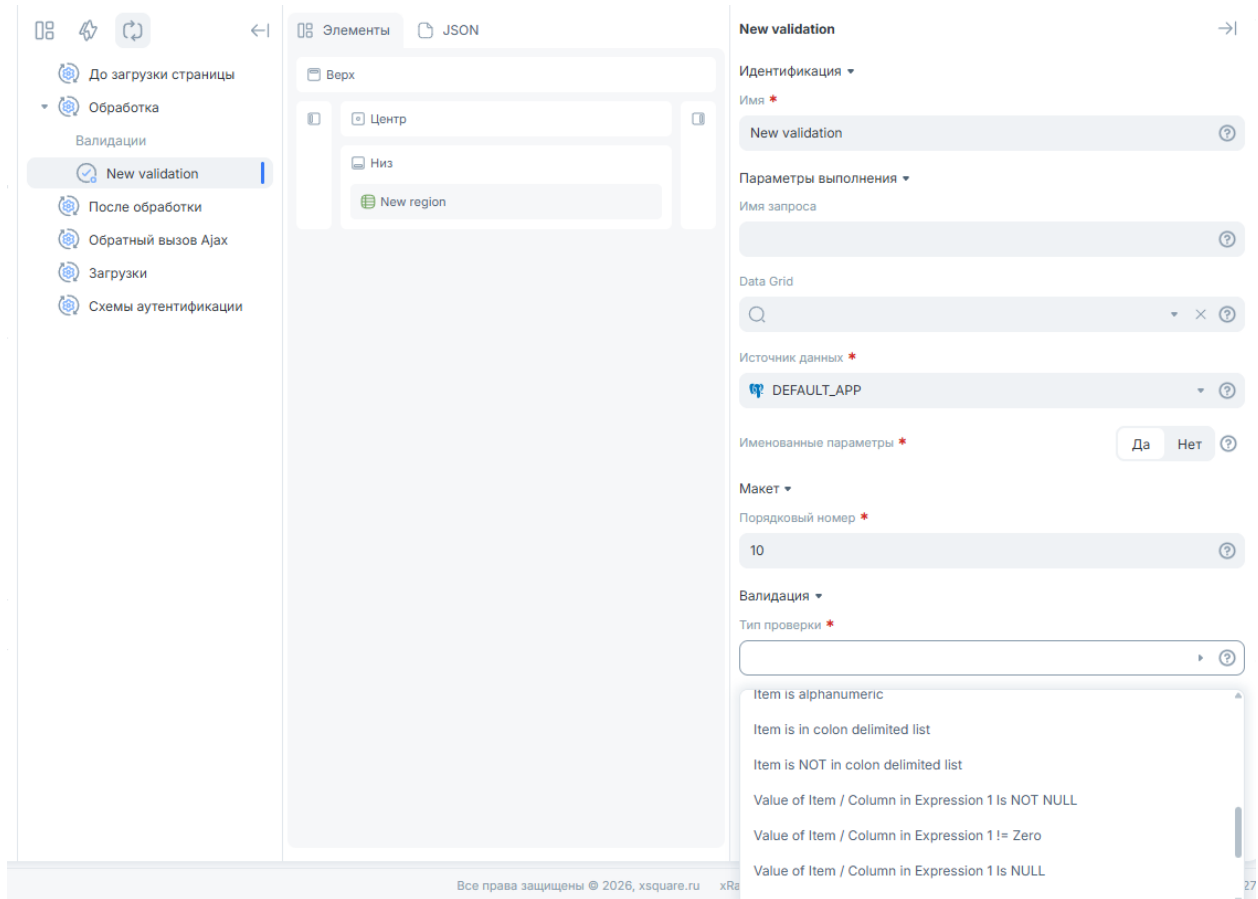
Параметр	Тип	Описание
Имя	Текст	Имя ветви.
Имя запроса	Текст	Определяет имя запроса. Если указано, то процесс будет выполняться только тогда, когда значение глобальной переменной REQUEST будет соответствовать введенному значению.
Источник данных	Список	Определяет источник данных для данной ветви.
Именованные параметры	Переключатель	Флаг, который указывает необходимость воспринимать входящие параметры как часть текста sql запроса. При установке в false, разработчику потребуется явно указать входящие параметры. Отображается для источника данных отличных от типов Oracle.
Место выполнения	Список	Определяет место выполнения перехода. • Перед загрузкой страницы - переход будет выполнен перед загрузкой страницы. Данный пункт удобно использовать в случае, если необходимо реализовать перенаправления пользователя в зависимости от каких-либо условий. • После обработки - перенаправление будет вызвано после завершения выполнения процессов.
Последовательность	Число	Определяет последовательность выполнения процесса в списке процессов одного типа.
Поведение - Тип	Список	Определяет тип перехода страницы для выполнения: • URL-адрес перенаправления • Функция, возвращающая URL
Ссылка	Текст / Окно кон- структо- ра	Определяет ссылку на страницу, если задан тип перехода «URL-адрес перенаправления»
Источник - SQL	Область текста	Определяет SQL-запрос, который возвращает URL
Параметры запроса	Текст / Окно кон- структо- ра	Определяет элементы, которые необходимо передать в исполняемый запрос в качестве переменных.
Элемент формы, определяющий страницу	Список	Определяет элемент, значение которого будет взято в качестве номера страницы для осуществления перехода. Доступно при выборе «Страницы по значению элемента».
Условия отображения - Тип	Список	Определяет тип условия отображения региона на странице. Подробнее см. в <i>условиях выполнения</i> .

9.7 Валидация

Для создания полноценного приложения требуется взаимодействие с пользователем. Все действия и данные пользователя необходимо сохранять в базу данных. Однако, пользователь может умышленно или непреднамеренно вводить некорректные данные. Для того чтобы этого избежать, в XRAD предусмотрена проверка данных – валидация. Валидация – это проверка корректности введенных пользователем данных в поля ввода формы.

9.8 Типы валидаторов

В XRAD процесс валидации – это вызов различных функций и процедур (далее - валидаторы), которые выполняются перед тем как данные будут отправлены на сервер для обработки.



Разработчику доступно 24 типа валидаторов:

1. Exists (SQL query returns at least one row) – позволяет выполнить проверочный SQL-запрос. Условие считается выполненным, если запрос вернул хотя бы 1 строку;
2. NOT Exists (SQL query returns no rows) – позволяет выполнить проверочный SQL-запрос. Условие считается выполненным, если запрос не вернул ни одной строки;
3. Value of Item / Column in Expression 1 Is NOT NULL – встроенное условие на заполненность поля или ячейки таблицы. Условие считается выполненным, если поле ввода или ячейка таблицы содержит какое-либо значение;
4. Value of Item / Column in Expression 1 != Zero – встроенное условие на значение отличное от 0 (нуль) поля ввода или столбца таблицы. Условие считается выполненным, если объект проверки будет содержать значение отличное от 0 (нуль);
5. Value of Item / Column in Expression 1 Is NULL – встроенное условие на значение NULL (пустое значение) поля ввода или столбца таблицы. Условие считается выполненным, если объект проверки содержит NULL (пустое значение);
6. Value of Item / Column in Expression 1 Is NULL or Zero – встроенное условие на значение NULL (пустое значение) или 0 (нуль) поля ввода или столбца таблицы. Условие считается выполненным, если объект проверки содержит NULL (пустое значение) или равно 0 (нуль);

7. Value of Item / Column in Expression 1 = Zero – встроенное условие на значение 0 (нуль) поля ввода или столбца таблицы. Условие считается выполненным, если объект проверки будет содержать значение 0 (нуль);
8. Value of Item / Column in Expression 1 Is NOT null and the Item / Column Is NOT Zero – встроенное условие на заполненность и значение отличное от 0 (нуль) поля ввода или столбца таблицы. Условие считается выполненным, если объект проверки не содержит NULL (пустое значение) и не равно 0 (нуль);
9. SQL Expression – позволяет выполнить SQL-запрос, возвращающий TRUE или FALSE. В поле ввода запроса указывается только тело запроса, без ключевых слов SELECT и/или FROM. Если необходимо выполнить какой-то подзапрос, то его необходимо обернуть в “(” “)”. Условие считается выполненным, если запрос вернул TRUE;
10. Request = Value - проверяет значение системной глобальной переменной REQUEST на введенное значение. Условие выполняется если значение равняется указанному
11. Request != Value - проверяет значение системной глобальной переменной REQUEST на введенное значение. Условие выполняется если значение не равняется указанному.
12. Request is contained in Value - проверяет значение системной глобальной переменной REQUEST на введенное значение. Условие выполняется если значение содержится в указанной строке (LIKE).
13. Request is NOT contained in Value - проверяет значение системной глобальной переменной REQUEST на введенное значение. Условие выполняется если значение не содержится в указанной строке (NOT LIKE).
14. Item = Value - проверяет соответствие значения элемента ввода на указанное. Условие выполняется если значения соответствуют.
15. Item != Value - проверяет соответствие значения элемента ввода на указанное. Условие выполняется если значения не соответствуют.
16. Item is numeric - проверяет значение элемента ввода. Условие выполняется если значение содержит только цифры или равно null. Общие разделители десятичных и разрядов игнорируются.
17. Item is NOT numeric - проверяет значение элемента ввода. Условие выполняется если значение не является числовым или равно null. Общие разделители десятичных и разрядов игнорируются.
18. Item is alphanumeric - проверяет значение элемента ввода. Условие выполняется если значение содержит только цифры и буквы или равно null.
19. Item is in colon delimited list - проверяет значение элемента ввода. Условие выполняется если значение соответствует одному из введенных вариантов. Варианты вводятся с помощью разделителя двоеточие.
20. Item is not in colon delimited list - проверяет значение элемента ввода. Условие выполняется если значение не соответствует ни одному из введенных вариантов. Варианты вводятся с помощью разделителя двоеточие.
21. PL/SQL Expression - выполняет условие на языке PL/SQL. Доступно только для элементов с источником данных типа Oracle. Условие выполняется если выражение возвращает значение TRUE.
22. PL/SQL Function Body - выполняет тело функции на языке PL/SQL. Доступно только для элементов с источником данных типа Oracle. Условие выполняется если функция возвращает значение TRUE.
23. Function returning Error Text – позволяет выполнить SQL-запрос возвращающий текст ошибки. В поле запроса необходимо указать полноценный запрос возвращающий строку. Проверка считается пройденной успешно, если запрос вернул NULL. В противном случае будет отображен возвращенный запросом текст ошибки;

24. Regular Expression – позволяет проверить значение поля формы по регулярному выражению. В поле Item указывается проверяемое поле, а в поле Value регулярное выражение. Проверка считается пройденной успешно, если значение поля Item будет соответствовать регулярному выражению.

Также имеется отдельный тип валидации – это флаг, который определяет необходимость заполнения поля формы. Для любого поля ввода можно поставить флаг, что поле обязательно для заполнения. В таком случае, даже если не сформировать иных валидаторов, при попытке отправить данные на сервер будет проведена проверка на заполненность поля данными.

Вышеуказанные валидаторы могут применяться как к данным формы, так и к данным введенным в интерактивный компонент DataGrid. Для осуществления данной проверки необходимо указать связанный DataGrid в соответствующем поле. В таком случае имена колонок могут использоваться как входящие параметры для валидатора.

Подробнее условия выполнения см. в *Типы условий*.

9.9 Ошибка валидации

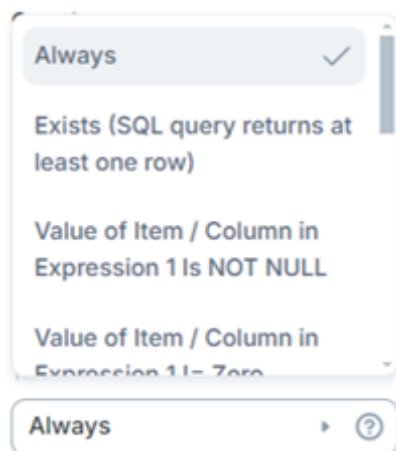
В случае возникновения исключения в любом из присутствующих на форме валидаторе отправка на сервер блокируется и пользователю выдается заданное разработчиком предупреждение. Если валидатор привязан к полю ввода, то дополнительно будет подсвечено поле ввода, на котором произошло исключение.

9.10 Системные валидаторы

Если поле отмечено как обязательное, но оно не заполнено, то в этом случае сработает системный валидатор и пользователь будет уведомлен о необходимости заполнить обязательные поля всплывающей подсказкой. Если на поле ввода требуется добавить дополнительную проверку, то необходимо создать валидатор для данного поля. В этом случае сначала будут вызваны системные валидаторы, а затем пользовательские.

9.11 Условия для выполнения валидации

Так как валидации выполняются автоматически при submit страницы, то часто возникают ситуации, когда необходимо ограничить их выполнение. Для этого в системе XRAD предусмотрен блок условий, по которым выполняются валидации. Подробнее смотри в *Типы условий*.



9.12 Пропуск валидаций.

Управление исполнением валидаций также можно управлять с помощью специального флага «Пропустить валидации», который указывается в атрибутах элемента «Кнопка». При указании данного флага будут пропущены все системные валидаторы, а также те что выполняются на сервере приложений. Если необходимо пропустить только часть, в атрибутах валидации необходимо указать флаг «Всегда выполнять». При указании данного флага валидация будет выполняться вне зависимости от атрибута кнопку, но при этом все равно будут учитываться условия выполнения и имя запроса.

9.13 Вызовы REST

Вызов REST - процесс, который позволяет вызвать внешний REST API.

Рассмотрим уникальные для данного типа процесса параметры:

Параметр	Тип	Описание
Источник - URL	Текст	Определяет URL-адрес для выполнения запроса. URL необходимо указывать полностью включая протокол.
HTTP-метод	Список	Определяет HTTP-метод для выполнения запроса: • GET • POST • PUT • DELETE • PATCH
HTTP-заголовки	Окно конструктора	Определяет заголовки HTTP. Заголовки могут включать в себя статические значения, а также принимать значения элементов формы.
Параметр запроса URL	Окно конструктора	Определяет параметры запроса, которые будут добавлены в URL. Значения параметров могут быть как статическими, так и взятыми из элемента ввода.
Тип тела запроса	Список	Определяет тип тела запроса. Возможные варианты: • Отсутствует • Статичное значение • Form-encoded • SQL
Обработка ответа	Список	Определяет тип обработчика ответа сервиса: • Отсутствует • Назначить переменные • SQL-обработчик
Назначение переменных	Окно конструктора	Определяет соответствие элементов формы и элементов ответа в формате JSON. В качестве аргумента обработчика ответа можно указать путь JMES. В случае если путь вернет просто значение JSON (строка, число, булево значение) элемент будет содержать это значение. Если JMES путь возвращает составное значение (объект или массив) в элемент будет передано вернувшееся значение в формате JSON.
Обработка в SQL	Окно конструктора	Определяет SQL-обработчик.
Входные параметры	Текст / Окно конструктора	Определяет элементы, которые необходимо передать в SQL-запрос в качестве переменных.

9.14 Процесс DataGrid

Для обработки данных введенных в интерактивных компонент DataGrid используется специальный процесс. В рамках данного процесса указывается как именно должны обрабатываться строки, измененные или добавленные пользователем. Процесс позволяет ввести собственный код который должен вызываться для удаления, обновления и вставки строк. Для обработки пользовательских значений процессу доступна информация о каждой ячейки измененных строк. Доступ к данным в этом случае осуществляется по имени столбца.

Помимо общих полей с обычным процессом, выделяются следующие атрибуты:

Параметр	Тип	Описание
Data Grid	Список	Указывается регион с типом DataGrid, для которого должен быть применен данный процесс.
Предотвращать потерянное обновление	Переключатель	Определяет необходимость выполнения операций для предотвращения потерянных обновлений. Потерянные обновления возникают когда два пользователя редактируют одну и ту же строку. При указании данного параметра система будет распознавать такие случаи и выводить ошибку.
Удалить	Область текста	Определяет код процесса, который необходимо выполнить в случае удаления пользователем строки.
Вставка	Область текста	Определяет код процесса, который необходимо выполнить в случае вставки пользователем новой строки. Для того чтобы новая строка сразу отобразилась в DataGrid, необходимо чтобы процесс возвращал колонки, которые помечены как «Первичный ключ».
Обновление	Область текста	Определяет код процесса, который необходимо выполнить в случае обновления строки.
Условия отображения - Выполнить	Список	Позволяет задать условие выполнения с учетом данных переданных из DataGrid. Возможные варианты: • Один раз - условие выполняется единожды. Результат выполнения будет применен ко всем переданным строкам • Для каждой строки - условие выполняется для каждой строки.

9.15 Загрузки

Процесс загрузки файла от веб-приложения. При создании процесса загрузки необходимо задать параметры:

Параметр	Тип	Описание
Имя	Текст	Имя процесса загрузки.
Источник данных	Список	Определяет источник данных для данного процесса.
Имя запроса	Текст	Определяет имя запроса. Если указано, то процесс будет выполняться только тогда, когда значение глобальной переменной REQUEST будет соответствовать введенному значению.
Последовательность	Число	Определяет последовательность выполнения процесса в списке процессов одного типа.
Источник - SQL	Область текста	Определяет SQL-запрос, который возвращает URL.
Колонка с наименованием	Список	Определяет колонку, содержащую название файла.
Колонка с файлом	Список	Определяет колонку, содержащую данные файла.
Mime Type Колонка	Список	Определяет колонку содержащую mime тип файла.
Content Disposition	Список	Определяет содержание заголовка Content-Disposition: • Attachment • Inline Заголовок отвечает за то как необходимо отобразить содержимое файла в браузере. Если указано attachment - файл будет загружен на компьютер пользователя. Если указано inline – содержимое файла будет отображено в браузере.

9.16 Схемы аутентификации

Для ограничения доступа к непубличным страницам приложения используются страницы аутентификации. Способы аутентификации определяются процессами аутентификации, которые в свою очередь ссылаются на глобальные схемы аутентификации.

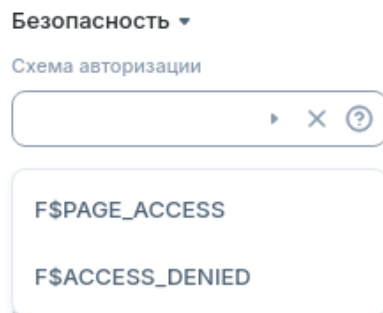
Рассмотрим уникальные параметры процесса аутентификации:

Параметр	Тип	Описание
Тип аутентификации	Список	Определяет тип аутентификации. Поддерживаемые типы аутентификации: • Custom • Ldap • Microsoft Ldap • Kerberos SSO • Microsoft Kerberos SSO • OIDC
Аутентификация	Список	Определяет имя процесса аутентификации из списка глобальных схем аутентификации в соответствии с выбранным типом.
Источник - Логин	Список	Определяет элемент формы, который будет использоваться для ввода логина. Атрибут доступен для схем Ldap и Microsoft Ldap.
Источник - Пароль	Список	Определяет элемент формы, который будет использоваться для ввода пароля. Атрибут доступен для схем Ldap и Microsoft Ldap.
Функция аутентификации	Область текста	Определяет SQL-запрос, который будет выполнен для осуществления аутентификации. Функция должна возвращать значение типа boolean. Первый параметр функции должен быть логин пользователя. Данное поле доступно только при выборе типа аутентификации «Custom».
Функция пост-логина	Область текста	Определяет SQL-запрос, который будет выполнен после успешной авторизации пользователя. Возвращаемые значения функции будут установлены в новую сессию аутентифицированного пользователя.
Сообщение об ошибке	Текст	Позволяет указать текст ошибки, которые будут выведены пользователю в случае неуспешной аутентификации.

9.17 Схемы авторизации

Определить доступность страницы конкретному пользователю можно в редакторе страниц через свойство «Безопасность - Схема авторизации».

В выпадающем списке можно увидеть доступные схемы авторизации.



Выбранная схема авторизации будет применена к странице после сохранения изменений через кнопку «Сохранить».

Если у пользователя нет прав на просмотр страницы, то будет отображена сообщение об ошибке, указанное в схеме авторизации.

В разрабатываемых приложениях часто используются такие элементы как меню, путь к текущей странице (breadcrumb), панели навигации и другое. Для вывода в них информации можно использовать как динамически формируемый набор значений (с помощью SQL-запроса), так и статичный – строго определенный список.

10.1 Что такое список

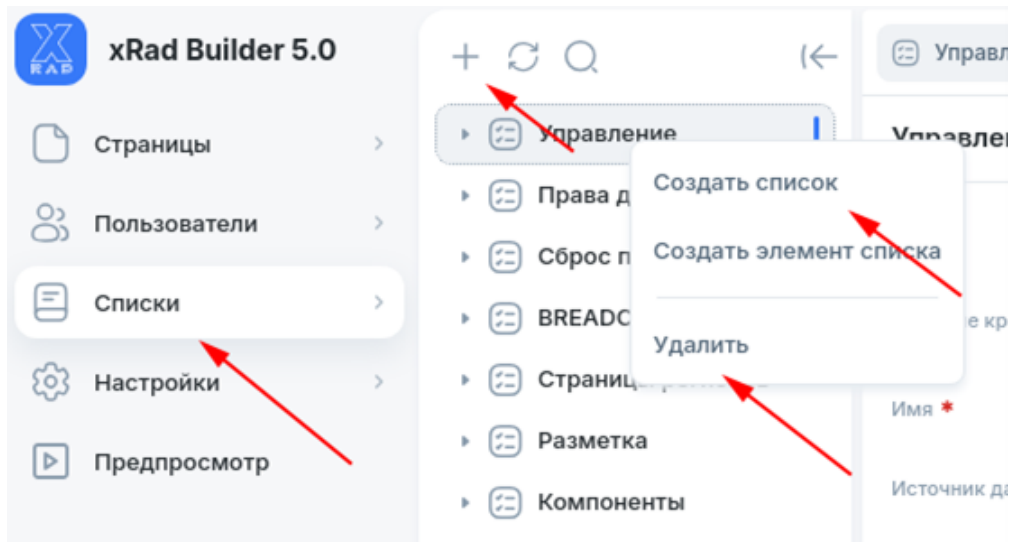
Список – это набор однотипных данных.

В XRAD существует 3 типа списков:

1. Статичный (Static) – список создается и редактируется разработчиком в среде XRAD.
2. Динамический (Based on Query) – список создается на основе SQL-запроса. Может меняться пользователем через интерфейс веб-приложения.
3. Breadcrumb – специальный тип, предназначенный для вывода пути до страницы приложения.

10.2 Создание, редактирование, удаление списков

Для создания списка необходимо выбрать в главном меню раздел «Списки» и нажать кнопку «+» в подменю. Также для создания или удаления списка можно использовать контекстное меню, нажав ПКМ на области, где перечислены списки и выбрать соответствующее действие.



Создание нового списка происходит во всплывающем окне, где необходимо указать:

- Имя – обязательный атрибут, определяющий имя списка.
- Источник данных – обязательный атрибут, определяющий БД для хранения списка.
- Тип списка – Static, Based on Query, Breadcrumb

При создании динамического списка (Based on Query) необходимо указать SQL-запрос, на основе которого будет строиться список и, по необходимости, переменные, которые используются в запросе.

Создать список [X]

Имя *
New1

Источник данных *
DEFAULT_APP

Тип *
Based on Query

Sql код *

```

1 SELECT 1 sequence
2       , 'Первая страница' title
3       , '/content/5025' link
4       , '5025' condition_type

```

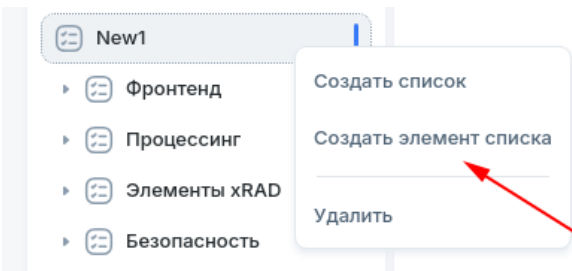
Переменные

Отмена Сохранить

После нажатия кнопки «Сохранить» будет создан новый список, к которому можно добавить элементы списка.

10.3 Создание элементов списка

Для создания элементов списка необходимо с помощью ПКМ вызвать контекстное меню и выбрать «Создать элемент списка».



В новой вкладке заполнить атрибуты нового элемента списка.

Новый элемент списка New1

Сохранить

Элемент списка ▾

Родительский элемент

Имя *

Источник данных *

DEFAULT_APP ▾

Порядковый номер *

Css класс иконки

Подсказка

Условие для активного состояния

Отделён

☐

Назначение ▾

Ссылка

Условия ▾

Тип Условия

Пользовательские атрибуты ▾

Атрибут 1

Атрибут 2

Родительский элемент - родительский пункт списка. Предназначен для формирования многоуровневого меню.

- Имя – обязательный атрибут, определяющий имя элемента списка.
- Источник данных – обязательный атрибут, определяющий источник данных.
- Порядковый номер - обязательный атрибут, определяющий порядок отображения элемента в списке.
- Css класс иконки – определяет отображаемую иконку.
- Подсказка – определяет текстовую подсказку элемента.
- Активен для страниц – список номеров страниц, на которых данный элемент списка будет выделен как активный.

- Отделен – Проставляет разделитель между пунктами, если список используется для формирования навигационного меню, находящегося сверху страницы.
- Ссылка – действия, которое будет выполняться при выборе элемента списка
- Тип условия – определяет тип условия, которое должно быть выполнено для перехода по ссылке.

Атрибуты 1, 2 - для некоторых вариантов вывода так же можно использовать дополнительные атрибуты списков. Например, для компонента Page Navigation в Attribute 1 можно указать что будет отображаться под наименованием элемента, а в Attribute 2 – что будет отображаться в правой части.

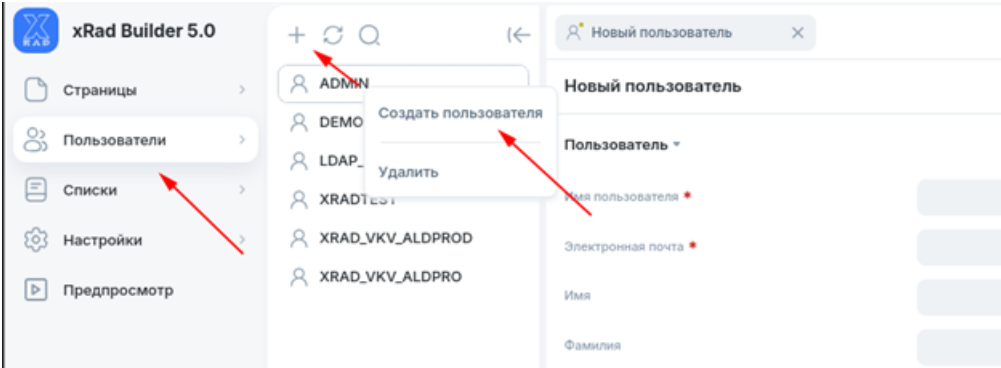
Управление пользователями

XRAD предоставляет возможность управлять пользователями напрямую в среде разработки. Внутренние пользователи среды разработки проходят аутентификацию по логину и паролю (схема CUSTOM), либо сопоставляются по имени с внешними пользователями, которые проходят аутентификацию в соответствии с выбранной схемой. Например, пользователь с именем DEVELOPER может войти в систему по логину и паролю, а может быть сопоставлен с пользователем DEVELOPER из Active Directory или OIDC. Внутренние пользователи XRAD имеют роль, которая определяет права доступа. Процессы создания и редактирования пользователей довольно минималистичны, но имеют весь необходимый функционал. Определены три основные роли, с помощью которых можно разграничить зоны доступа для пользователей:

- **ADMIN** - роль администратора. ADMIN имеет полные права, поэтому следует с особой осторожностью выдавать данную роль. Пользователь с данной ролью может редактировать, создавать и удалять других пользователей. Имеет право создать другого пользователя с ролью ADMIN. Данную роль следует назначать строго администратору, для разработчиков достаточно прав роли DEVELOPER.
- **DEVELOPER** - роль разработчика. DEVELOPER может просматривать и изменять Страницы, создавать и удалять их. Имеет право редактировать Списки и Настройки. Имеет право просматривать Пользователей, но не имеет права изменять их (в том числе и своего пользователя). Данная роль подходит для разработчиков, так как пользователем с данной ролью доступны все инструменты кроме управления Пользователями.
- **VIEW** - роль с минимальным набором привилегий. VIEW может просматривать Страницы и их содержимое, а также Списки и Настройки (но не их содержимое!). Не может просматривать Пользователей. Не имеет права создавать Страницы, не имеет права изменять настройки своего пользователя. Данная роль отлично подходит для пользователей, которые хотят познакомиться с XRAD.

11.1 Создание, редактирование, удаление пользователей

Для создания пользователя необходимо выбрать в главном меню раздел «Пользователи» и нажать кнопку «+» в подменю. Также для создания или удаления пользователя можно вызвать контекстное меню, нажав ПКМ на области, где перечислены пользователи и выбрать соответствующее действие.



После заполнения или редактирования параметров выбранной учетной записи необходимо подтвердить внесенные изменения, нажав кнопку «Сохранить».

Параметр	Описание
Имя пользователя	обязательный атрибут, определяющий имя создаваемого пользователя, используется в качестве логина.
Электронная почта	обязательный атрибут, определяющий адрес электронной почты пользователя.
Имя, Фамилия	атрибуты для внесения имени и фамилий человека, закрепленного за данным пользователем.
Роль	обязательный атрибут, определяющий одну из возможных ролей пользователя.
Заблокирован	флаг, определяющий блокировку пользователя в системе.

Обязательно необходимо указать и подтвердить пароль пользователя, а также можно выставить флаг «Требовать изменение пароля» при необходимости смены пароля после первичного входа. Примечание: при редактировании параметров пользователя изменение поля «Имя пользователя» – недоступно.