

Инструкция администратора Jelata Framework

I. Правила работы с базой данных PostgreSQL:

Рекомендации по созданию таблиц:

- 1) Рекомендуется называть таблицы, схемы и поля таблиц символами, нижнего регистра, латинского алфавита.
Не рекомендуется использовать различные знаки препинания и знаки арифметических операций, а также называть только цифрами.
Не рекомендуется использовать служебные слова языка SQL (insert, select, update, union, declare, desc и т.д.)
- 2) Рекомендуется указывать значение по умолчанию для полей not null.
- 3) Рекомендуется указывать связи с другими таблицами с помощью foreign key.
- 4) Рекомендуется оставлять короткий комментарий для поля таблицы, он будет использоваться, как алиас, по умолчанию при создании компонентов интерфейса.
- 5) Рекомендуется использовать таблицу framework.users для работы пользователями. Для работы с организациями framework.orgs, с ролями framework.roles (в случае, если в проекте это требуется).
- 6) При создании поля с названием userid, при добавлении записи в таблицу, в него будет записываться пользователь, при обновлении он так же будет обновляться.
- 7) Рекомендуется описывать логику и логические проверки в триггерах.

II. Правила работы с компонентами View:

Правила построения запросов:

- 1) Передаваемые параметры из адресной строки являются фильтрами для SQL запроса. Важно понимать, что параметры должны передаваться по указанному в конфигурации алиасу (title), то есть, если вы переименовали алиас поля id в number, то отфильтровать данные вы сможете следующим образом:
правильно: /list/orgs?number=1
не правильно: /list/orgs?id=1
- 2) Можно передавать в запросе специальные параметры, которые будут заменены на сервере:
 - _userid_ - идентификатор текущего пользователя
 - _orgid_ - организация текущего пользователя
 - _orgs_ - организации текущего пользователя, массив из идентификаторов
 - _checked_ - выбранные строки в таблице (если выбран checker в конфигураторе)
- 3) Параметр relation передаётся в случае, если вам необходимо при заполнении другие поля значениями из параметров запроса. Значением параметра должны быть поля таблицы, перечисленные через запятую (поля, как они называются в таблице, col в конфигураторе). Их значения должны быть указаны в одноимённых параметрах запроса.

Например, у нас есть поле "orgtype" с алиасом "Type Organization":

правильно: /getone/org?id=0&relation=orgtype&orgtype=1

неправильно: /getone/org?id=0&relation=Type Organization&Type Organization=1

Пример с двумя полями:

/getone/org?id=0&relation=orgtype,director&orgtype=1&director=John Doe

- 4) Не постоянные параметры для запросов (not CONST) в действиях (Acts) берутся из полученного данных, либо из параметров адресной строки, в случае отсутствия данных в таблице. Для типов table и tiles, где множественный набор данных, параметры берутся из первой строки.

Работа с конфигуратором:

Поля конфигуратора и их применение:

1. column – название поля таблицы
2. title – алиас поля, который будет показан на интерфейсе
3. type – тип поля (подробное описание в разделе «Типы полей»)
4. roles – роли пользователей, из таблицы framework.roles, которым видно поле
5. visible – видимость поля
6. default value – значение по умолчанию, которое будет уходить в условие sql запроса
7. width – ширина поля
8. visible_condition – условия видимости
9. select_condition – условия получения зависимостей, для типов select, multiselect, typehead, multitypehead из связанной таблицы
10. join – применение JOIN вместо LEFT JOIN в SQL запросе
11. updatable – обновлять данные, после редактирования поля
12. class name – название CSS класса поля
13. relation – связанная таблица (по foreign key), также позволяет конфигурировать более глубокие связи
14. relation config – поля из связанной таблицы, при выборе полей, они будут добавляться в SQL запрос и в конфигуратор, так же выбранные поля будут служить для типов select, typehead в полях компонента form full и для фильтра select в компонентах tiles, table

Типы полей:

Form full:

1. text – текстовый input
2. number – числовой input
3. date – input типа дата
4. checkbox – input типа checkbox
5. select – компонент типа Select. Получает на вход value и label. Настраивается в relation и relation config, где необходимо выбрать value (например id), и label (например name) из вспомогательной таблицы. Можно настроить условие запроса в select condition
6. typehead – выполняет поиск, логика такая же как у типа select

7. array -
8. label – не редактируемое поле
9. password – input типа пароль, пароль хэшируется при добавлении в базу
10. autocomplete – автозаполнение, поиск выполняется по указанному полю
11. multiselect – множественный выбор, записывает в таблицу массив из идентификаторов выбранной таблицы, таблицу можно выбрать любую, связь не важна
12. multitypehead – не работает в данной версии
13. file – input типа файл, записывает в базу json с данными о файле

```
[{"thumbnail": "", "original": "", "src": "", "uri": "", "filename": "", "content_type": "", "size": 0}]
```
14. files,- input типа файл с возможностью множественной загрузки
15. image - input типа файл, сохраняет только картинки, в базу записывает json с информацией

```
[{"original": "http://jelata.framework/files/355f6af3-db46-4666-bd18-0b982dd9c720Terminator_Hauptseite.jpg", "src": "http://jelata.framework/files/355f6af3-db46-4666-bd18-0b982dd9c720Terminator_Hauptseite.jpg", "thumbnail": "http://jelata.framework/files/355f6af3-db46-4666-bd18-0b982dd9c720Terminator_Hauptseite.jpg", "size": 23993, "content_type": "image/jpeg", "filename": "Terminator_Hauptseite.jpg", "uri": "/files/355f6af3-db46-4666-bd18-0b982dd9c720Terminator_Hauptseite.jpg"}]
```
16. images - input типа файл с возможностью множественной загрузки картинок
17. gallery – Компонент галерея, служит для показа картинок, сохранённых с помощью типов image, images
18. filelist – Не редактируемый список файлов, сохранённых с помощью типов file, files
19. datetime –компонент для заполнения даты и времени
20. time – для заполнения времени
21. color – показывает цвет в зависимости от значения (т.е. для значения green будет нарисован зелёный квадрат)
22. colorpicker – выбор цвета, в базу запишется название цвета

Table & Tiles:

Для этих компонентов используются лишь некоторые типы, все остальные выводятся как простой текст

1. color – показывает цвет в зависимости от значения (т.е. для значения green будет нарисован зелёный квадрат)
2. colorpicker – показывает цвет в зависимости от значения (т.е. для значения green будет нарисован зелёный квадрат)
3. file – Не редактируемый список файлов, сохранённых с помощью типов file, files
4. files– Не редактируемый список файлов, сохранённых с помощью типов file, files
5. image - Компонент галерея, служит для показа картинок, сохранённых с помощью типов image, images images - Компонент галерея, служит для показа картинок, сохранённых с помощью типов image, images
6. gallery – Компонент галерея, служит для показа картинок, сохранённых с помощью типов image, images

7. `filelist` – Не редактируемый список файлов, сохранённых с помощью типов `file`, `files`
8. `images` - Компонент галерея, служит для показа картинок, сохранённых с помощью типов `image`, `images`
9. `array` – показывает таблицу внутри таблицы, для связей один ко многим