Jelata Framework Administrator Guide

I. *Rules for working with the PostgreSQL database:*

Recommendations for creating tables:

1) It is recommended to call the tables, charts and fields of the tables characters, lowercase, Latin alphabet.
   It is not recommended to use different punctuation marks and signs of arithmetic operations, and also to call only numbers.
   Do not use SQL service words (insert , select , update , union , declare , desc , etc.)
2) It is recommended to specify a default value for the fields non null .
3) It is recommended to specify links to other tables using the foreign key .
4) It is recommended to leave a short comment for the table field, it will be used, as an alias, by default when creating interface components.
5) It is recommended to use the framework framework . users to work with users. To work with organizations framework . orgs , with framework roles . roles (if required by the project).
6) When creating a field called userid , when adding a record to a table, the user will be written to it, and it will also be updated when updated.
7) It is recommended to describe logic and logic checks in triggers.

Ii. *Rules for working with components View:*

Query construction rules:

1) The parameters passed from the address bar are filters for the SQL query. It is important to understand that the parameters must be transmitted by the alias specified in the configuration (title), that is, if you renamed the alias id fields to number , you can filter the data as follows:

   correct: /list/orgs?number=1

   not correct: /list/orgs?id=1

2) You can send in the request special parameters that will be replaced on the server:
   - _ userid _ - current user ID
   - _ orgid _ - organization of the current user
   - _ orgs _ - organizations of the current user, an array of identifiers
   - _ checked _ - selected rows in the table (if the checker is selected in the configurator)
3) The relation parameter is passed if you need to fill in other fields with values from the query parameters. The parameter value must be table fields, separated by commas (fields, as they are called in the table, col in the configurator). Their values must be specified in the query parameters of the same name.

   For example, we have an “ orgtype ” field with an “ Type Organization ” alias :

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

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

   An example with two fields :

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

4) Non-constant parameters for queries (not CONST) in actions (Acts) are taken from the received data, or from the parameters of the address bar, in the absence of

data in the table. For the table and tiles types , where the plural data set is, the parameters are taken from the first line.

Work with the configurator:

Configurator fields and their application:

1. column - the name of the table field
2. title - field alias that will be shown on the interface
3. type - field type (detailed description in the section “Field types”)
4. roles - user roles from the framework table . roles that show the field
5. visible - field visibility
6. default value - the default value that will go to the sql condition of the query
7. width - field width
8. visible_condition - visibility conditions
9. select _ condition - conditions for obtaining dependencies, for types select , multiselect , typehead , multitypehead from a related table
10. join - use JOIN instead of LEFT JOIN in a SQL query
11. updatable - update data after editing the field
12. class name - the name of the CSS field class
13. relation - related table (by foreign key), also allows configuring deeper links
14. relation config - fields from a linked table, when selecting fields, they will be added to the SQL query and to the configurator, the selected fields will also be used for select types ,typehead in the fields of the form full component and for the select filter in tiles , table components

Types of fields:

Form full:

1. text - text input
2. number - numeric input
3. date - date type input
4. checkbox - input type checkbox
5. select - a component of type Select . Gets the input value and label . It is configured in the relation and relation config where it is necessary to select the value (for example, id) and label (for example, name) from the auxiliary table. You can customize the query condition in select condition
6. typehead - performs a search, the logic is the same as the type of select
7. array -
8. label - non-editable field
9. password - input type password, password is hashed when added to database
10. autocomplete - autocomplete, search is performed on the specified field
11. multiselect - multiple choice, writes an array to the table from the identifiers of the selected table, any table can be selected, the relationship is not important
12. multitypehead - does not work in this version
13. file - file type input , writes to the json database with file data.
[{"thumbnail": "", "original": "", "src": "", "uri": "," filename ":" "," content_type ":" "," size ": 0}]]
14. files , - input type file with the possibility of multiple loading
15. image - input file type, saves only pictures, writes json to the database with information

[{" original ": " http ://jelata . framework/files/355 f 6 af 3- db 46-4666- bd 18-
0 b 982 dd 9 c 720 Terminator _ Hauptseite . jpg ", " src ":
" http ://jelata . framework/files/355 f 6 af 3- db 46-4666- bd 18-
0 b 982 dd9 c 720 Terminator _ Hauptseite . jpg "," thumbnail ":" http ://jelata . framework/f
iles/355 f 6 af 3- db 46-4666- bd 18-
0 b 982 dd 9 c 720 Terminator _ Hauptseite . jpg "," size ":
23993," content _ type ":" image/jpeg "," filename":" Terminator _ Hauptseite . jpg "," uri ":"
/files/355 f 6 af 3- db 46-4666- bd 18-0 b 982 dd 9 c 720 Terminator _ Hauptseite . jpg "}]

16. images - input file type with the possibility of multiple images loading
17. gallery - Gallery component, used to display images saved using image types , images
18. filelist - Non-editable list of files saved using file , files types
19. datetime - component to fill in the date and time
20. time - to fill the time
21. color - shows color depending on the value (i.e. a green square will be drawn for the green value)
22. colorpicker - color selection, the name of the color will be written into the base

Table & Tiles :

Only some types are used for these components, all others are displayed as plain text.

1. color - shows the color depending on the value (i.e. a green square will be drawn for the green value)
2. colorpicker - shows color depending on the value (i.e. a green square will be drawn for the green value)
3. file - Non-editable list of files saved using file , files types
4. files - Non-editable list of files saved using file , files types
5. image - Gallery component, used to display images saved using image types , images images - Gallery component, used to display images saved using image types , images
6. gallery - Gallery component, used to display images saved using image types , images
7. filelist - Non-editable list of files saved using file , files types
8. images - Gallery component, used to display images saved using image types , images
9. array - shows a table inside a table, one to many for bundles