ADIOS framework

SQL vs. ADIOS data types

(& Tables, Forms and Inputs)

June 28, 2023

ADIOS Data Types

- ADIOS contains its own set of “data types”, e.g. int, lookup, date, file, color, varchar, image, …
- The complete list of available data types can be found in *src/Core/DB/DataTypes* folder.
- ADIOS data type is used to:
  - Install the database structure;
  - Render the SQL commands for inserting and updating records;
  - Provide automatic joins for the foreign keys (the “lookup”) when selecting data;
  - Customize the user interface, mostly:
    - Define the way how the data is visualized in tables
    - Define the way how the input for the data looks like

ADIOS Data Types, contd.

- Each column in the model is an ADIOS data type, not the SQL column.
- During the **installation of the database**, the column is translated to the SQL syntax (see get_sql_create_string() method in each data type)
  - Selected data types automatically get indexes: int, lookup, boolean, date, datetime.
- When rendering the **tables**, some parameters are used to customize the look (e.g., “title” or “unit”)
- When rendering the **inputs**, some parameters are used to customize the input (e.g., “title”, “description”, “inputStyle”, “unit”)

Example – lookup column

ADIOS column
“id_customer_category”

- type: lookup
- title: “Customer category”
- model: “Widgets/Customers/Models/CustomerCategory”
- required: true
- inputStyle: “select:
- order: “name asc”

SQL notation for this column (extract)

```
create table `customers`
  `id_customer_category` int(8)
  …
  index (`id_customer_category`)
;
```

Example – lookup column, contd.

The “id_customer_category”:

- will be created as an int(8) with an SQL index
- In a table, the name of the category will be displayed (instead of the numeric value in the SQL database). This is configured in a “lookupSqlValue” property of the model “Widgets/Customers/Models/CustomerCategory”
- In a form, the input will be always rendered as a <select> (see the “inputStyle” parameter)
  - Note: Default implementation of the input for the lookup is using the autocomplete in case there will be more than 35 items to select.

Why all this?

Centralized definition of both SQL notation of the column and its interpretation in the user interface (tables, forms, inputs).

In a columns() method of a model.

Available parameters of the columns

Parameter name	Description
type	The most important. The ADIOS type of the column.
title	Title in the tables, label for the inputs.
description	Additional description of the column, used in the input.
unit	E.g. “%”, “EUR”, “kg”. Used in the tables and inputs.
input_style	Used for lookups, available options: select, autocomplete. Used also for ints, available option: slider. If this is used, additional parameters can be provided: max, min and step. Note: shall be changed to inputStyle.
input	Name of the class for the completely custom input.
value	Initial value of the input (used in editing).
enum_values	If an “int” contains “enum_values”, it is rendered as a <select>.
placeholder	Used for <input> HTML tags (e.g. for varchar or int inputs).
interface	Used in “text” data type. Available options: plain_text, formatted_text, json_editor.
readonly	If set to true, the input is rendered as readonly.
decimals	Number of decimal characters for the float type.
onchange	Javascript code to run for onchange. Note: This shall not be parametrized in the columns() but in the formParams() method.
subdir	Used for file data type. The relative path to the sub folder (starting from ROOT_DIR/upload) where the files should be uploaded.

This table does not list all available parameters. Only the most common used.

The parameters are still in an ongoing development. They should be:

- Understood
- Reviewed
- Optimized

I strongly recommend to study the Input view:

src/Core/Views/Input.php

Search for ‘$this->params’.

Some additional comments

- There is no data type transated to SQL's ENUM. For enums, the "int" is used, with an additional "enum_values" parameter".
- Refactoring from underscore_notation to camelCase is needed.
- The parametrization of inputs is a "living development" – any comments or ideas are welcome.

Rendering forms (full of inputs)

It is important to understand the **templating mechanism** of the Form view.

The Form view renders the forms for adding and editing records in the database. **By default, it iterates the columns defined in the columns()** method and for each column it renders the appropriate input. The input can be parametrized by the parameters explained in the previous slides.

The template can be customized by the “template” parameter of the Form view. How it works – this is for the separate presentation.

Note: Using templates in the Form is not the same as using the Grid view !

Any questions?