

ADIOS framework

Working with UI components

May 26, 2023

UI components - Examples

- **UI/Form** – rendering custom forms for inserting or editing records
- **UI/Table** – default list view of a data in a table (in a model)
- **UI/Input** – renders an input, based on its *DataType*
- **UI/Button** – renders a button

Rendering UI components

Option 1:

`$adios->renderAction()`

Automatically called by ADIOS core when an URL like <http://myproject/UI/Form?model=<MODEL>&id=<ID>> is requested.

ADIOS's routing routes the URL to the action which contains the `renderAction()` for the requested component.

Takeaway: You do not need to create an action in your widget, if you want to render a UI/Form or UI/Table. Just simply call the URL.

Option 2:

`$adios->ui->COMPONENT(PARAMS)`

Suitable when developing custom actions and you want to render any UI component.

Example: you want to render an input or a button inside your action's `render()`

```
public function render() {  
    $inputHtml = $adios->ui->Input([ ... ]);  
    $buttonHtml = $adios->ui->Button([ ... ]);  
    return $inputHtml.$buttonHtml;  
}
```

Rendering UI components

Option 3:

`(new \ADIOS\Core\UI\COMPONENT($adios, $inputId, $inputParams))->render()`

Equivalent to the **Option 2**. Example:

```
(new \ADIOS\Core\UI\Input\CheckboxField(  
    $this->adios,  
    "{$params['uid']}_ProductDomainAssignment",  
    [  
        "model" => "Widgets/Products/Models/ProductDomainAssignment",  
        "key_column" => "id_product",  
        "key_value" => $data['id'],  
        "value_column" => "domain",  
        "values" => $domainAssignmentValues,  
        "columns" => 2,  
    ]  
))->render()
```

View and Input

\ADIOS\Core\UI\View

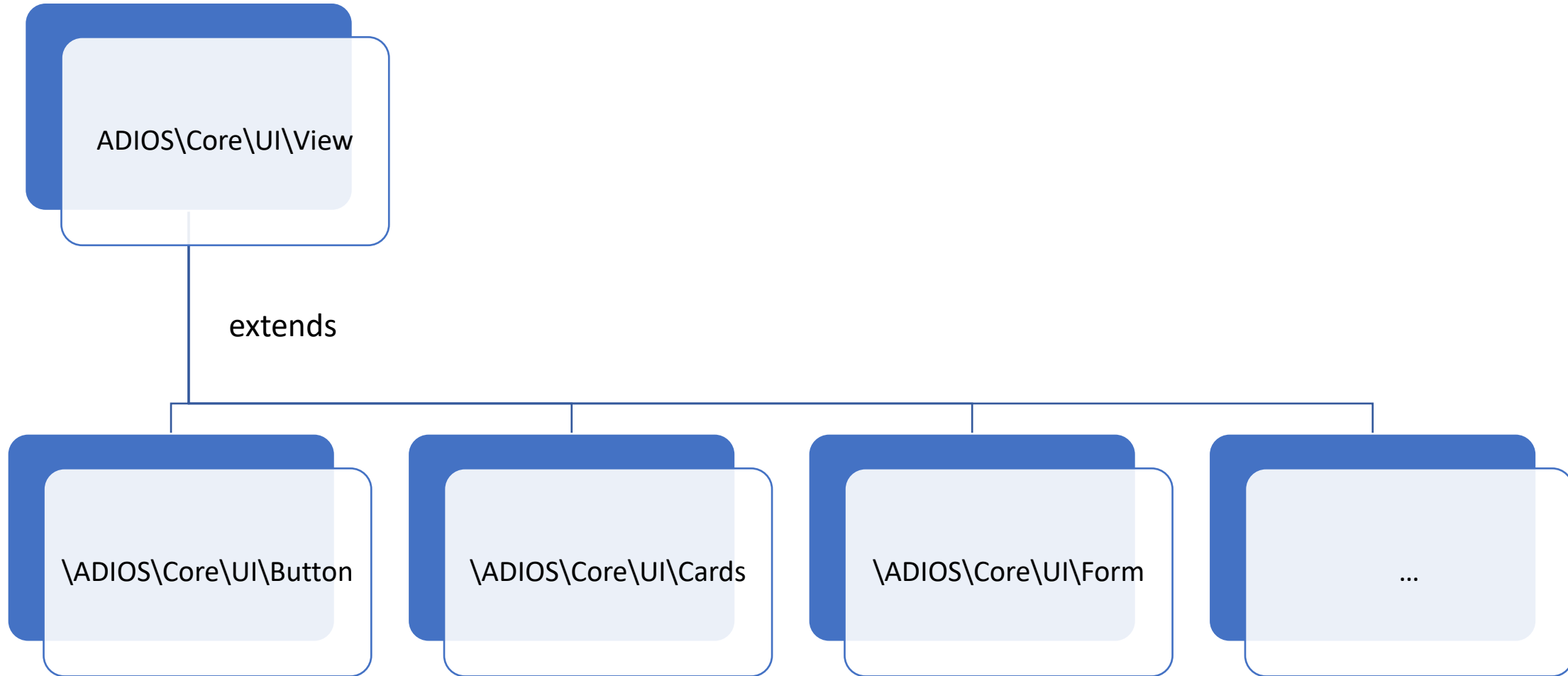
Is the base class providing low-level functionality for rendering any HTML.

\ADIOS\Core\Input

Is the base class providing low-level functionality for rendering inputs.

It implements inputs for the available *DataTypes* and can be extended to “complex” inputs.

UI components – class hierarchy



Available UI components (alphabetical order)

Button

Cards

DataTable

FileBrowser

Form

Grid

Html

Input

SettingsPanel

Table

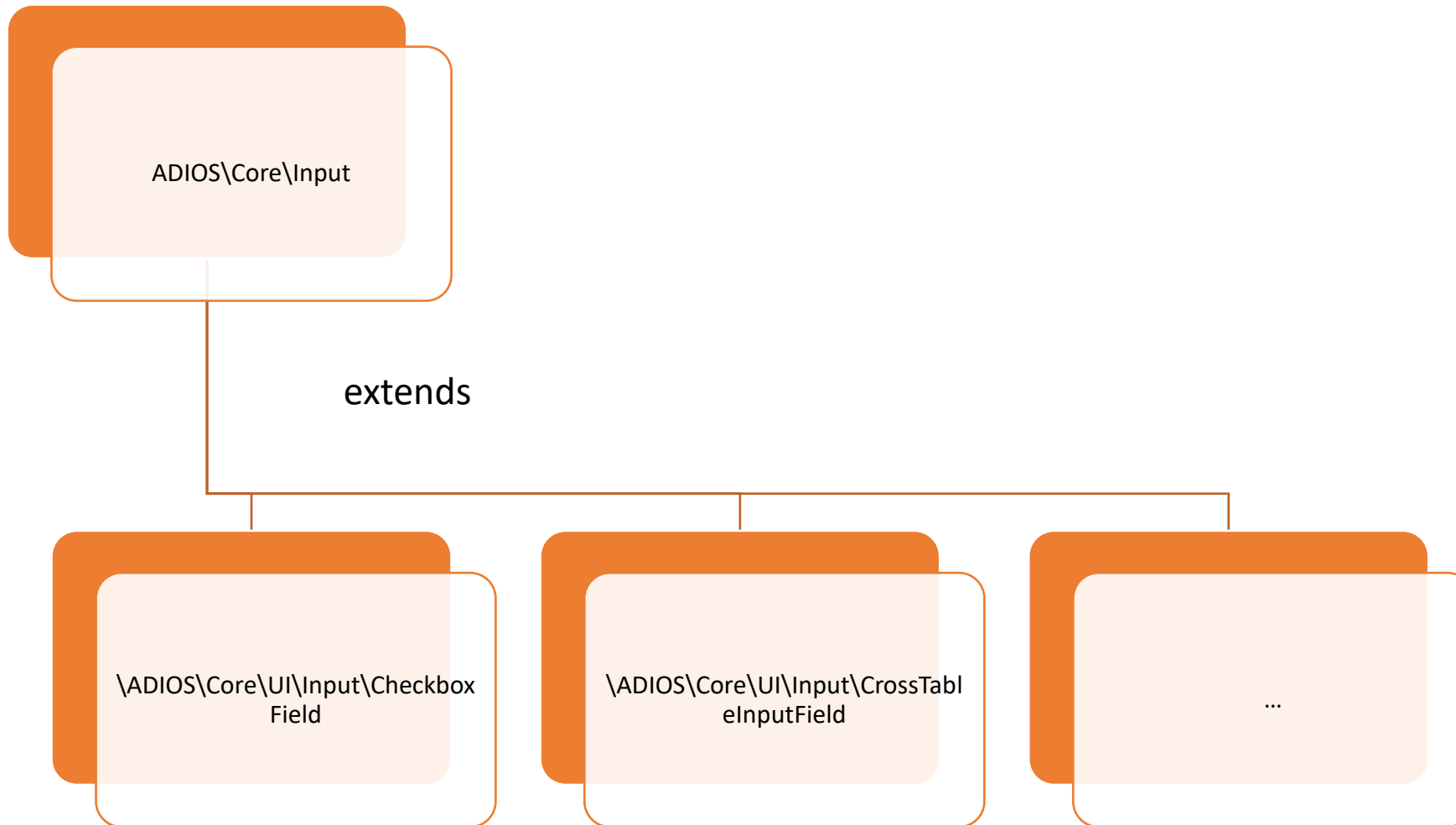
Tabs

Title

Tree

Window

“Complex” inputs – class hierarchy



Available “complex” input components

CheckboxField

CrossTableInputField

FileBrowser

ManyToMany

SettingsPanel

Tags

Tree

How to create own UI component (e.g. “MyComponent”)

- Create a file `src\Core\UI\MyComponent.php`
- Class `MyComponent` extends `\ADIOS\Core\UI\View`
 - `__construct($adios, $params)` pre-processes the parameters of the component
 - Public function `render()` returns the HTML of the component, based on its `$params`