

ADIOS framework

Models and Callbacks

May 31, 2023

Model (\ADIOS\Core\Model)

- Implemented in src/Core/Model.php
- Definition of columns, indexes and DB upgrades
- DB-related methods, either ADIOS native or Eloquent's
- Support for 1:N and M:N relations
- Standardized CRUD operations
- 13 callbacks currently implemented

Implemented callbacks

```
onTableBeforeInit($tableObject) : void
onTableAfterInit($tableObject) : void
onTableAfterDataLoaded($tableObject) : void
onFormBeforeInit($formObject) : void
onFormAfterInit($formObject) : void
onBeforeInsert($data)
onModelBeforeUpdate($data)
onBeforeSave($data)
onAfterInsert($data)
onModelAfterUpdate($data)
onAfterSave($data, $returnValue)
onBeforeDelete(int $id)
onAfterDelete(int $id)
```

3 groups of callbacks:

- Callbacks for CRUD
 - Insert, Update, Save, Delete
- Callbacks for UI/Table:
 - Before/After init
 - After data loaded
 - These are the replacement for the `tableParams()`
- Callbacks for UI/Form:
 - Before/After init
 - These are the replacement for the `formParams()`

Legacy “callbacks” (copy&paste from old ADIOS)

```
tableRowCSSFormatter($data)
tableCellCSSFormatter($data) : void
tableCellHTMLFormatter($data) : void
tableCellCSVFormatter($data) : void
```

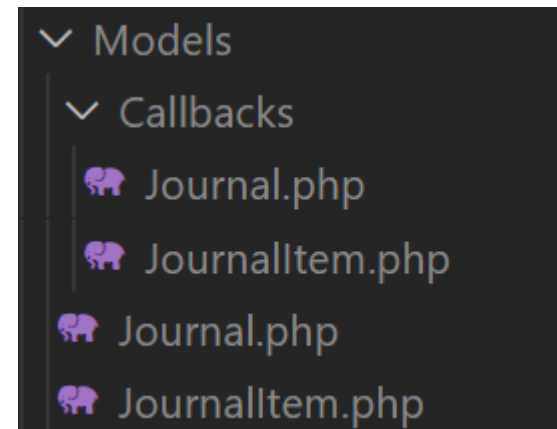
These will be replaced by “new” callbacks:

```
onTableRowCss
onTableCellCss
onTableCellHtml
onTableCellCsv
```

How to work with callbacks

1. Create a folder “Callbacks” in the folder “Models”

In the example on the right, we have two models “Journal” and “JournalItem” with two separate files for callbacks.



How to work with callbacks

1. Create a folder “Callbacks” in the folder “Models”
- 2. Create a trait in the “Callbacks” folder**

```
<?php

namespace ADIOS\Widgets\Finance>MainBook\Journal\Models\Callbacks;

trait Journal { ...
}
```

How to work with callbacks

1. Create a folder “Callbacks” in the folder “Models”
2. Create a trait in the “Callbacks” folder
3. **Implement the functionality of the callback**

```
<?php

namespace ADIOS\Widgets\Finance>MainBook\Journal\Models\Callbacks;

trait Journal {
    public function onBeforeInsert($data) {
        return $data;
    }
}
```

How to work with callbacks

1. Create a folder “Callbacks” in the folder “Models”
2. Create a trait in the “Callbacks” folder
3. **Implement the functionality of the callback**

```
<?php

namespace ADIOS\Widgets\Finance>MainBook\Journal\Models\Callbacks;

trait Journal {
    public function onBeforeInsert($data) {
        return $data;
    }
}
```


How to work with callbacks

1. Create a folder “Callbacks” in the folder “Models”
2. Create a trait in the “Callbacks” folder
3. Implement the functionality of the callback
4. **Use the trait in the model**

```
<?php

namespace ADIOS\Widgets\Finance>MainBook\Journal\Models;

class Journal extends \ADIOS\Core\Model {
    use \ADIOS\Widgets\Finance>MainBook\Journal\Models\Callbacks\Journal;

    // ...
}
```

Final words

- Implementation of the callbacks is separated from the definition of the model
- **Prototype builder generates the empty trait for callbacks automatically**

```
<?php
namespace ADIOS\Widgets\Finance\MainBook\Journal\Models\Callbacks;

trait Journal {
    /**
     * This trait has been generated by the prototype builder.
     * No callback is implemented.
     *
     * Available callbacks are listed and commented below.
     * If you will need to implement a functionality of any
     * callback, uncomment it.
     */
}
```