Flowpack Expose Documentation

Release 0.0.1

Marc Neuhaus

August 12, 2014

Contents

CHAPTER 1

What is Expose ?

Expose provides various components to quickly build tables, forms and whole cruds for Doctrine Entities. The general aim and focus is to reduce the boilerplate code you currently have to write for things like forms, lists, etc. to as little as possible while still retaining as much flexibility as possible.

1.1 Installation

1.1.1 Composer

add this repository to your composer.json:

```
{
  "type": "vcs",
  "url": "git@github.com:mneuhaus/Flowpack.Expose.git"
}
```

```
composer require flowpack/expose
```

1.1.2 Required pending ChangeSets

create an `beard.json` file with this content in your project root:

```
{
  "defaults": {
    "gerrit_api_endpoint": "https://review.typo3.org/",
    "gerrit_git": "git.typo3.org"
  },
  "changes": [
    {
      "name": "[WIP][FEATURE] Add a way to clear caches by Path + FilePattern",
      "type": "gerrit",
      "path": "Packages/Framework/TYPO3.Flow",
      "change_id": "25078"
    },
    {
      "name": "!!![WIP][FEATURE] ControllerInheritence Fallbacks",
      "type": "gerrit",
      "path": "Packages/Framework/TYPO3.Fluid",
      "change_id": "31939"
    }
```

```
  ]
}
```

Then run the following command in your root directory:

```
beard patch
```

`beard` is a little helper to automatically patch based on gerrit changes specified in gerrit.json. (https://github.com/mneuhaus/Beard)

1.2 CrudController

The `CrudController` enables you to quickly create a controller with a complete CRUD workflow for your entities. Given you have an entity called 'MyPackageDomainModelCompany'. Then you can simply create a `CompanyController` in your package that extends the `CrudController` like this:

```
<?php
namespace My\Package\Controller;

use Flowpack\Expose\Controller\CrudController;
use TYPO3\Flow\Annotations as Flow;

/**
 */
class CompanyController extends CrudController {
    /**
     * @var string
     */
    protected $entity = '\My\Package\Domain\Model\Company';
}
```

That's it, now you can go to `/my.package/company/index` to test the controller. You should see an empty listview and a button in the top right corner to create a new company. You don't need to copy any templates, layouts or partials by default. Expose contains sensible default templates for list, new, edit, etc.

1.2.1 Change a template

If you want to change a template all you need to do is to copy the template that you want to use over to your CompanyController.

For example, if you want to change the edit template you need to copy the file `Flowpack.Expose/Private/Resources/Templates/Crud/Edit.html` to `My.Package/Private/Resources/Templates/Company/Edit.html` the view will automatically use your edit template instead of the default expose template.

Templates

Template	Description
Templates/Crud/Index.html	This template contains the list view for the specified entity
Templates/Crud/New.html Templates/Crud/Edit.html	These templates contain a simple form which display a form field for each of the properties in the specified model.
Templates/Crud/Show.html	You can use this template to create a page displays details about a specific Entity.

Partials

Template	Description
Partials/Pagination.html	This partial contains the limit and pagination that is display under the list view.
Partials/SortField.html	This partial is used by the SortBehavior to wrap sortable properties with some useful markup
Partials/Search.html	This partial is used to render the search form above the list view
Partials/Table/Actions.html	This partial renders the local actions for each row in the list view
Partials/Table/Body.html	This partial renders the rows itself in the list view
Partials/Table/Header.html	This partial renders the header of the list view
Partials/Table/Layout.html	This partial defines the general layout of the list view
Partials/Form/Wrap/Default.html	This partial is used to wrap the rendered form control with label, validation errors, infotext, etc.
Partials/Form/Field/*.html	These partials are used to render the different controls

1.2.2 Change existing Actions

since you inherit from the CrudController you can of course override any existing action to suit your needs. This might be useful to set some properties of the model before persisting it inside the createAction.

1.2.3 Add new Actions

Aside from altering existing actions you can create as many new actions as you like. If you want to include your new action at some location in the default expose templates you can use the `Flowpack\Expose\Annotations\Action` annotation.

```
<?php
namespace My\Package\Controller;

use Flowpack\Expose\Controller\CrudController;
use TYPO3\Flow\Annotations as Flow;
use Flowpack\Expose\Annotations as Expose;

/**
 */
class CompanyController extends CrudController {
    /**
     * @var string
     */
    protected $entity = '\My\Package\Domain\Model\Company';

    /**
     * @Expose\Action(type="local", label="My Custom Action")
     * @param '\My\Package\Domain\Model\Company' $entity
     * @return void
     */
    public function myCustomAction($entity) {
      // ...
    }
}
```

The `Flowpack\Expose\Annotations\Action` annotation has 3 options that you can set:

Option	Description
label	Contains the label that will be used to render the action
type	Contains the type of the action. This can be either `global`, `local` or `batch`. See the section below for details
class	Contains a class that will be added to the action link tag for styling

Action types

There are 3 different types of actions that you can define to be used by Expose:

global:

A global action is an action that can act without a specific entity. Expose includes the action `new` as a global action to create a new entity of the specified type. These will be displayed in the top right corner of the list view by default.

local:

A local action is an action that receives a specific entity as argument and does something with that. Expose includes the actions `edit`, `delete` and `show` as local actions. These will be displayed by default inside each row of entities in the list view

batch:

A batch action is an action that receives an array of entities to perform actions on them. Expose includes the action `deleteBatch` by default. These will be display in a select box above the list view and will be executed for each marked row, when you click on the execute button right beside the select box.

1.3 Configuration

By default Expose will parse the Class you're working with by itself and use defaults to give you the best default experience it can. You can additionally create a Configuration File in your Configuration Folder called `Expose.ModelName.yaml`.

The General Syntax of this File looks like this

```
'\My\Package\Domain\Model\MyModel':
    listProperties: someProperty
    defaultSortBy: someProperty
    properties:
        someProperty:
            label: Some cool Property
            control: Textarea
```

1.3.1 Class Configurations

Name	Description
listProperties	Contains properties that should be display as columns in the list view
searchProperties	List of properties used by the SearchBehavior
filterProperties	List of properties used by the FilterBehavior
defaultSortBy	Default property to sort by
defaultOrder	Default order to sort the property
defaultWrap	Default wrap to use for the form controls
layout	Layout used by the Crud Controller
listBehaviors	Array of Behaviors that are used by the list view.

1.3.2 Property Configurations

Name	Description
label	Description that will be places under the form control
control	Contains properties that should be display as columns in the list view
infotext	Description that will be places under the form control
optionsProvider	Description that will be places under the form control
wrap	Description that will be places under the form control

1.4 OptionsProvider

1.4.1 What is an OptionsProvider

This OptionsProvider is very useful to add options to a select for a simple string property

OptionsProvider are used to provide Options for FormFields like SingleSelectionMixin, MultiSelectionMixin and similar. The interface consists of only 1 method called "getOptions" which should return an associative array of key, value pairs.

1.4.2 Array

This OptionsProvider is very useful to add options to a select for a simple string property

```
'\TYPO3\Party\Domain\Model\ElectronicAddress':
    properties:
       type:
           control: 'SingleSelect'
           optionsProvider:
               Name: Array
               Options:
                   new: 'New'
                   done: 'Done'
                   rejected: 'Rejected'
```

Settings

Name	Required	Description
Options	yes	Contains the options that will be provided

1.4.3 Constant

This OptionsProvider is used to load options from an Entities class by using a regular expression to match existing constants

```
TYPO3\Party\Domain\Model\ElectronicAddress:
    Properties:
        type:
            Element: TYPO3.Form:SingleSelectDropdown
            OptionsProvider:
                Name: ConstOptionsProvider
                Regex: TYPE_.+
```

Settings

Name	Required	Description
Regex	yes	Contains a Regular Expression to filter the class constants
EmptyOption	no	Set this setting to add an emtpy option to the beginning of the options

1.4.4 Country

This OptionsProvider is provides a localized list of countries

Settings

Name	Required	Description
EmptyOption	no	Set this setting to add an emtpy option to the beginning of the options

1.4.5 Relation

This OptionsProvider is used to fetch entities based on the orm relation of a property.

Settings

Name	Required	Description
QueryMethod	no	Method to call on the Repository to create a query
EmptyOption	no	Set this setting to add an emtpy option to the beginning of the options

1.4.6 Role

OptionsProvider for Policy Roles

1.4.7 Custom OptionsProvider

You can easily create an optionsProvider for your special usecase.

```
<?php
class MyOptionsProvider extends \Flowpack\Expose\Core\OptionsProvider\AbstractOptionsProvider {
    /**
     * @return array $options
     */
    public function getOptions() {
        $myOptions = array();
        // some logic to actually fill $myOptions
        return $myOptions;
    }
}
?>
```

1.5 ViewHelpers

1.5.1 Form/Field

You can use this ViewHelper to create complete form fields for your form. By default you have to take care of quite a lot of things yourself to render a form field, like

- form control
- label
- relation between label and form control to enable focusing by clicking on the label
- wrapper around the label + control for better styling
- showing validation errors next to the form control
- add a class to the wrapper around the label + control to indicate an validation error
- maybe add an infotext

To make this easier and reduce the fluid code needed you can use this viewhelper like this:

Basic usage

```
<e:form.field name="foo" control="Textfield" wrap="Default" value="bar" />
```

This will render a `Textfield` with the name `foo` inside the default wrapper based on Bootstrap 3 and a value of ‘‘bar‘

```
<div class="form-group">
  <label for="foo" class="col-sm-3 control-label">Foo</label>
  <div class="col-sm-9">
    <input class="form-control" id="foo" type="text" name="foo" value="bar">
  </div>
</div>
```

Output of the same field when validation failed

```
<div class="form-group has-error">
  <label for="foo" class="col-sm-3 control-label">Foo</label>
  <div class="col-sm-9">
    <input class="form-control" id="foo" type="text" name="foo" value="bar">
    <span class="help-block">This property is required.</span>
  </div>
</div>
```

Usage with an object bound form

To make things even easier you can use it in combinatin with the binding of objects to you form like this:

```
<f:form action="create" object="myObject" name="myObject">
  <e:form.field property="someString" />
  <e:form.field property="someRelation" />
  <e:form.field property="someBoolean" />
  ...
</f:form>
```

This will automatically resolve the control that should be used based on the property type and use the default wrap.

```
<form action="...">
   <div class="form-group">
     <label for="someString" class="col-sm-3 control-label">Some String</label>
     <div class="col-sm-9">
       <input class="form-control" id="someString" type="text" name="someString">
     </div>
   </div>
```

```
  <div class="form-group">
    <label for="someRelation" class="col-sm-3 control-label">Some String</label>
    <div class="col-sm-9">
      <input class="form-control" id="someRelation" type="text" name="someRelation">
      <select class="form-control" id="someRelation" name="someRelation">
        <!-- Options provided by the RelationOptionsProvider -->
      </select>
    </div>
  </div>
  <div class="form-group">
    <label for="someString" class="col-sm-3 control-label">Some String</label>
    <div class="col-sm-9">
      <input class="form-control" id="someString" type="text" name="someString">
    </div>
  </div>
</form>
```

Arguments

Name	Type	Required	Description
additionalAttributes	array	no	Additional tag attributes. They will be added directly to the resulting HTML tag.
data	array	no	Additional data-* attributes. They will each be added with a "data-" prefix.
name	string	no	Name of input tag
value	mixed	no	Value of input tag
property	string	no	Name of Object Property. If used in conjunction with <f:form object="...">, "name" and "value" properties will be ignored.
control	string	no	Specifies the control to use to render this field
wrap	string	no	Specifies the wrap used to render the field
class	string	no	CSS class(es) for this element
dir	string	no	Text direction for this HTML element. Allowed strings: "ltr" (left to right), "rtl" (right to left)
id	string	no	Unique (in this file) identifier for this HTML element.
lang	string	no	Language for this element. Use short names specified in RFC 1766
style	string	no	Individual CSS styles for this element
title	string	no	Tooltip text of element
accesskey	string	no	Keyboard shortcut to access this element
tabindex	integer	no	Specifies the tab order of this element
onclick	string	no	JavaScript evaluated for the onclick event

1.5.2 Form/PropertyHasResults

You can use this viewhelper to check if a property has validation errors.

Examples

```
<div class="form-group {e:form.propertyHasResults(property: someProperty, then: 'has-error')}">
  ...
</div>

<e:form.propertyHasResults property="someProperty">
  This property has some errors!
</e:form.propertyHasResults>
```

Arguments

Name	Type	Required	Description
then	mixed	no	Value to be returned if the condition if met.
else	mixed	no	Value to be returned if the condition if not met.
property	string	yes	Name of the property to check for Validation errors

1.5.3 Form/PropertyResults

Checks if the specified property has errors and adds them as a variable to the view.

Example

```
<e:form.propertyResults property="someProperty">
  <f:for each="{errors}" as="error">
    <p class="help-block">{error.message}</p>
  </f:for>
</e:form.propertyResults>
```

Arguments

Name	Type	Required	Description
property	string	yes	Name of the propert to check for Validation errors
as	string	no	Name of the variable the errors will be assigned into

1.5.4 Format/Ucfirst

returns the string with the first character capitalized

given a variable named `foo` contains a string `fooBar`

```
{foo -> e:format.ucfirst()}
```

Will result in:

```
FooBar
```

1.5.5 Format/Uuid

Arguments

Name	Type	Required	Description
object	object	yes	

1.5.6 Link/Action

A view helper for creating links to actions.

= Examples =

<code title="Defaults"> <f:link.action>some link</f:link.action> </code> <output> some link (depending on routing setup and current package/controller/action) </output>

<code title="Additional arguments"> <f:link.action action="myAction" controller="MyController" package="YourCompanyName.MyPackage" subpackage="YourCompanyName.MySubpackage" arguments="{key1: 'value1', key2: 'value2'}">some link</f:link.action> </code> <output> some link (depending on routing setup) </output>

Arguments

Name	Type	Required	Description
additionalAttributes	array	no	Additional tag attributes. They will be added directly to the resulting HTML tag.
data	array	no	Additional data-* attributes. They will each be added with a "data-" prefix.
action	string	yes	Target action
arguments	array	no	Arguments
controller	string	no	Target controller. If NULL current controllerName is used
package	string	no	Target package. if NULL current package is used
subpackage	string	no	Target subpackage. if NULL current subpackage is used
section	string	no	The anchor to be added to the URI
format	string	no	The requested format, e.g. ".html
additionalParams	array	no	additional query parameters that won't be prefixed like $arguments (overrule $arguments)
addQueryString	boolean	no	If set, the current query parameters will be kept in the URI
argumentsToBeExcluded-FromQueryString	array	no	arguments to be removed from the URI. Only active if $addQueryString = TRUE
useParentRequest	boolean	no	If set, the parent Request will be used instead of the current one
absolute	boolean	no	By default this ViewHelper renders links with absolute URIs. If this is FALSE, a relative URI is created instead
class	string	no	CSS class(es) for this element
dir	string	no	Text direction for this HTML element. Allowed strings: "ltr" (left to right), "rtl" (right to left)
id	string	no	Unique (in this file) identifier for this HTML element.
lang	string	no	Language for this element. Use short names specified in RFC 1766
style	string	no	Individual CSS styles for this element
title	string	no	Tooltip text of element
accesskey	string	no	Keyboard shortcut to access this element
tabindex	integer	no	Specifies the tab order of this element
onclick	string	no	JavaScript evaluated for the onclick event
name	string	no	Specifies the name of an anchor
rel	string	no	Specifies the relationship between the current document and the linked document
rev	string	no	Specifies the relationship between the linked document and the current document
target	string	no	Specifies where to open the linked document

1.5.7 Link/Filter

Arguments

Name	Type	Required	Description
additionalAttributes	array	no	Additional tag attributes. They will be added directly to the resulting HTML tag.
data	array	no	Additional data-* attributes. They will each be added with a "data-" prefix.
property	string	yes	Property to sort by
value	string	no	value to sort by
action	string	no	Target action
arguments	array	no	Arguments
controller	string	no	Target controller. If NULL current controllerName is used
package	string	no	Target package. if NULL current package is used
subpackage	string	no	Target subpackage. if NULL current subpackage is used
section	string	no	The anchor to be added to the URI
format	string	no	The requested format, e.g. ".html
additionalParams	array	no	additional query parameters that won't be prefixed like $arguments (overrule $arguments)
addQueryString	boolean	no	If set, the current query parameters will be kept in the URI
argumentsToBeExcluded-FromQueryString	array	no	arguments to be removed from the URI. Only active if $addQueryString = TRUE
useParentRequest	boolean	no	If set, the parent Request will be used instead of the current one
absolute	boolean	no	By default this ViewHelper renders links with absolute URIs. If this is FALSE, a relative URI is created instead
class	string	no	CSS class(es) for this element
dir	string	no	Text direction for this HTML element. Allowed strings: "ltr" (left to right), "rtl" (right to left)
id	string	no	Unique (in this file) identifier for this HTML element.
lang	string	no	Language for this element. Use short names specified in RFC 1766
style	string	no	Individual CSS styles for this element
title	string	no	Tooltip text of element
accesskey	string	no	Keyboard shortcut to access this element
tabindex	integer	no	Specifies the tab order of this element
onclick	string	no	JavaScript evaluated for the onclick event
name	string	no	Specifies the name of an anchor
rel	string	no	Specifies the relationship between the current document and the linked document
rev	string	no	Specifies the relationship between the linked document and the current document
target	string	no	Specifies where to open the linked document

1.5.8 Link/Sort

This viewhelper only works in conjunction with the SortProcessor.

Example

```
<e:link.sort property="someProperty">Some Property</e:link.sort>
```

Arguments

Name	Type	Required	Description
additionalAttributes	array	no	Additional tag attributes. They will be added directly to the resulting HTML tag.
data	array	no	Additional data-* attributes. They will each be added with a "data-" prefix.
property	string	yes	Property to sort by
action	string	no	Target action
arguments	array	no	Arguments
controller	string	no	Target controller. If NULL current controllerName is used
package	string	no	Target package. if NULL current package is used
subpackage	string	no	Target subpackage. if NULL current subpackage is used
section	string	no	The anchor to be added to the URI
format	string	no	The requested format, e.g. ".html
additionalParams	array	no	additional query parameters that won't be prefixed like $arguments (overrule $arguments)
addQueryString	boolean	no	If set, the current query parameters will be kept in the URI
argumentsToBeExcluded-FromQueryString	array	no	arguments to be removed from the URI. Only active if $addQueryString = TRUE
useParentRequest	boolean	no	If set, the parent Request will be used instead of the current one
absolute	boolean	no	By default this ViewHelper renders links with absolute URIs. If this is FALSE, a relative URI is created instead
class	string	no	CSS class(es) for this element
dir	string	no	Text direction for this HTML element. Allowed strings: "ltr" (left to right), "rtl" (right to left)
id	string	no	Unique (in this file) identifier for this HTML element.
lang	string	no	Language for this element. Use short names specified in RFC 1766
style	string	no	Individual CSS styles for this element
title	string	no	Tooltip text of element
accesskey	string	no	Keyboard shortcut to access this element
tabindex	integer	no	Specifies the tab order of this element
onclick	string	no	JavaScript evaluated for the onclick event
name	string	no	Specifies the name of an anchor
rel	string	no	Specifies the relationship between the current document and the linked document
rev	string	no	Specifies the relationship between the linked document and the current document
target	string	no	Specifies where to open the linked document

1.5.9 Navigation/Menu

Arguments

Name	Type	Required	Description
as	string	no	

1.5.10 Navigation/MenuItem

Arguments

Name	Type	Required	Description
additionalAttributes	array	no	Additional tag attributes. They will be added directly to the resulting HTML tag.
data	array	no	Additional data-* attributes. They will each be added with a "data-" prefix.
action	string	no	Target action
arguments	array	no	Arguments
controller	string	no	Target controller. If NULL current controllerName is used
package	string	no	Target package. if NULL current package is used
subpackage	string	no	Target subpackage. if NULL current subpackage is used
section	string	no	The anchor to be added to the URI
format	string	no	The requested format, e.g. ".html
additionalParams	array	no	additional query parameters that won't be prefixed like $arguments (overrule $arguments)
addQueryString	boolean	no	If set, the current query parameters will be kept in the URI
argumentsToBeExcluded-FromQueryString	array	no	arguments to be removed from the URI. Only active if $addQueryString = TRUE
useParentRequest	boolean	no	If set, the parent Request will be used instead of the current one
absolute	boolean	no	By default this ViewHelper renders links with absolute URIs. If this is FALSE, a relative URI is created instead
item	array	no	
class	string	no	CSS class(es) for this element
dir	string	no	Text direction for this HTML element. Allowed strings: "ltr" (left to right), "rtl" (right to left)
id	string	no	Unique (in this file) identifier for this HTML element.
lang	string	no	Language for this element. Use short names specified in RFC 1766
style	string	no	Individual CSS styles for this element
title	string	no	Tooltip text of element
accesskey	string	no	Keyboard shortcut to access this element
tabindex	integer	no	Specifies the tab order of this element
onclick	string	no	JavaScript evaluated for the onclick event
name	string	no	Specifies the name of an anchor
rel	string	no	Specifies the relationship between the current document and the linked document
rev	string	no	Specifies the relationship between the linked document and the current document
target	string	no	Specifies where to open the linked document

1.5.11 Actions

This viewhelper looks for actions annotated with the `\Flowpack\Expose\Annotations\Action` annotation and filter them by the type of action specified (local, global, batch)

Example

```
<e:actions type="global">
  <f:for each="{actions}" key="action" as="actionAnnotation">
    <e:link.action action="{action}" class="{actionAnnotation.class}">
      {actionAnnotation.label}
    </e:link.action>
  </f:for>
</e:actions>
```

Arguments

Name	Type	Required	Description
type	string	yes	Type of actions to return [locallglobalbatch]
as	string	no	Variable to assign the actions into the view with

1.5.12 Behavior

Arguments

Name	Type	Required	Description
objects	objects	yes	
behaviors	array	no	

1.5.13 Block

Arguments

Name	Type	Required	Description
name	string	yes	

1.5.14 Process

Arguments

Name	Type	Required	Description
objects	objects	yes	
processors	array	no	

1.5.15 Property

You can use this viewHelper to retrieve a property from ab object based on the name of the property stored in a variabl

Example

```
<f:for each="{properties}" as="property">
  <e:property object="{object}" name="{property}" />
</f:for>
```

Arguments

Name	Type	Required	Description
object	object	yes	Object to get the property or propertyPath from
name	string	yes	Name of the property or propertyPath

1.5.16 Render

ViewHelper that renders a section or a specified partial

== Examples ==

<code title="Rendering partials"> <f:render partial="SomePartial" arguments="{foo: someVariable}" /> </code> <output> the content of the partial "SomePartial". The content of the variable {someVariable} will be available in the partial as {foo} </output>

<code title="Rendering sections"> <f:section name="someSection">This is a section. {foo}</f:section> <f:render section="someSection" arguments="{foo: someVariable}" /> </code> <output> the content of the section "someSection". The content of the variable {someVariable} will be available in the partial as {foo} </output>

<code title="Rendering recursive sections"> <f:section name="mySection">

<ul>

<f:for each="{myMenu}" as="menuItem">

<li> {menuItem.text} <f:if condition="{menuItem.subItems}">

<f:render section="mySection" arguments="{myMenu: menuItem.subItems}" />

</f:if>

</f:for>

</f:section> <f:render section="mySection" arguments="{myMenu: menu}" /> </code> <output>

<li>menu1

**
** menu1a menu1b

[...] (depending on the value of {menu}) </output>

<code title="Passing all variables to a partial"> <f:render partial="somePartial" arguments="{_all}" /> </code> <output> the content of the partial "somePartial". Using the reserved keyword "_all", all available variables will be passed along to the partial </output>

Arguments

Name	Type	Required	Description
section	string	no	Name of section to render. If used in a layout, renders a section of the main content file. If used inside a standard template, renders a section of the same file.
partial	string	no	Reference to a partial.
arguments	array	no	Arguments to pass to the partial.
optional	boolean	no	Set to TRUE, to ignore unknown sections, so the definition of a section inside a template can be optional for a layout

1.5.17 Schema

Arguments

Name	Type	Required	Description
className	string	yes	
as	string	no	

1.5.18 User

This ViewHelper helps you to use the currently logged in party in your fluid template without littering every controller the securityContext and assigning the current party to the view.

Arguments

Name	Type	Required	Description
as	string	no	

1.5.19 Wrap

Arguments

Name	Type	Required	Description
name	string	yes	Name of the Wrapper
arguments	array	no	Arguments supplied to the callback applying this wrapper

This version of the documentation has been rendered at: August 12, 2014