

cjw NETWORK

Developers united in eZ Publish



How we started with eZ Publish 5

A real use case

A cookbook for successful migration from eZ 4 to the Symfony stack

Ekkehard Dörre - Coolscreen
Donat Fritschy - Webmanufaktur

cjw NETWORK
Developers united in eZ Publish

Presenters

- **Ekke** is a consultant with deep knowledge in eZ Publish 4 and 5, eZ Find / Apache Solr and with a faible for coming cutting edge web technologies. He is one of the organizers of the PHP Unconference since seven years.
- **Donat** is owner of Webmanufaktur, a full service web agency in Switzerland. He works as projects manager, software architect and developer and likes thinking outside of the box.
- Members of CJW Network

Why a Cookbook?

- eZ Publish 5 allows for a smooth migration of eZ legacy projects to the Symfony stack, permitting them to profit from the exiting new possibilities
- However, it is a completely new beast
- This workshop presents some basic recipes for beginners

Agenda

Things we'll cover:

- eZ 5 Installation
- Building our first Bundle
- Overriding the Page Layout Template
- Integrating old Templates
- Overriding Content Type (formerly Class) Templates
- Overriding Field Type (formerly Attribute) Templates
- One more thing...



Installation

Prepare the ingredients...

eZ 5 Installation

Recipe #1: Use the installation package

- Use the installation packages from share.ez.no
- These are consistent and tested
- Everybody knows about what you speak
- Forking from [GitHub](https://github.com) is great, if you want and are able to contribute

eZ 5 Installation

Recipe #2: Read the Installation notes

- eZ 5 is a complex install and different from what you know
- Actually, it combines two environments
 - Symfony
 - eZ Publish legacy (eZ 4.7)
- <https://confluence.ez.no/display/EZP/Requirements>
- <https://confluence.ez.no/display/EZP/Normal+installation>

Common Pitfalls:

- Linking the assets
- Directory and file permissions

eZ 5 Installation

Recipe #3: Get directory and file permissions right

Strategy 1 (quick and dirty)

- Same user/group for web server and console user

Strategy 2

- Separate users for web server and console user
- Both members of `www` group
- Usually requires `umask(0007)`
- <https://confluence.ez.no/x/9YBx>
- <http://symfony.com/doc/current/book/installation.html#configuration-and-setup>

eZ 5 Installation

Recipe #4: Use the setup wizard

- This will give you a testable environment...
- ... which will immediately show you all problems ;-)
- <http://ezpublish.ezsc/>
- http://ezpublish.ezsc/ezdemo_site_admin
- Login: admin / Password: ezsc

eZ 5 Installation

Recipe #5: The console is you friend

Check out the `console` command! First, log into the virtual machine using SSH

```
$ ssh ezsc@vm.ezsc
ezsc@vm.ezsc''s password: ezsc
$ cd /var/www/ezpublish
```

BASH

To list all available commands use

```
$ php ezpublish/console
```

BASH

The most important commands:

```
$ php ezpublish/console cache:clear

$ php ezpublish/console assets:install
$ php ezpublish/console assetic:dump

$ php ezpublish/console twig:lint
```

BASH



Creating Bundles

bring to the boil...

Creating a Bundle

Recipe #6: Use bundles for your sites

A Bundle is similar to an eZ extension and module. We suggest you create separate 'site' bundles for all sites and 'functional' bundles for common components.

Creation of a bundle is easy:

```
$ php ezpublish/console generate:bundle
```

BASH

Follow suggested Namespace conventions: `YourCompany/YourCustomer/ComponentBundle`
(`CjwNetwork/Summertime2013/CookBookBundle`)

You may define a shorted name for your bundle, as we have: `CjwCookBookBundle`

Create the Bundle in the `src` folder and answer `yes` to all questions.

Note: this will also change `ezpublish/EzPublishKernel.php` and `ezpublish/config/routing.yml` to reference the generated bundle.

Testing Your Bundle

A generated bundle contains sample code that allows for easy testing:

<http://ezpublish.ezsc/hello/demo>

The magic is done through a controller which receives the request from the router and prepares a response with the help of a template renderer.

`src/CjwNetwork/SummerCamp2013/CookBookBundle/Controller/DefaultController.php`

```
class DefaultController extends Controller
{
    public function indexAction($name)
    {
        return $this->render('CjwCookBookBundle:Default:index.html.twig', array('name' => $name));
    }
}
```

PHP

Note: when implementing your own controllers, follow the code in eZDemoBundle as it includes additional classes for accessing the repository.

Inspecting the TWIG Template

src/CjwNetwork/SummerCamp2013/CookBookBundle/Resources/views/Default/index.html.twig

```
Hello {{ name }}
```

TWIG

We add some formatting and apply a TWIG filter:

```
<h1>Hello and good morning {{ name | upper }}!</h1>
```

TWIG

- <http://ezpublish.ezsc/hello/demo>
- TWIG Doc <http://twig.sensiolabs.org/doc/filters/upper.html>

Adding a Page Layout

Unlike eZ Publish legacy, TWIG templates work “bottom up” and support inheritance.

Therefore it's easy to show the output in the standard eZ Demo Layout:

`src/CjwNetwork/SummerCamp2013/CookBookBundle/Resources/views/Default/index.html.twig`

```
{# This template extends pagelayout.html.twig and just replaces the 'content' block #}  
{% extends "eZDemoBundle::pagelayout.html.twig" %}  
  
{% block content %}  
<h1>Hello and good morning {{ name | upper }}!</h1>  
{% endblock %}
```

TWIG



Overriding Standard Templates

Dish up...

Creating a TWIG Template for Article

src/CjwNetwork/SummerCamp2013/CookBookBundle/Resources/views/full/article.html.twig

TWIG

```
{% extends noLayout ? viewbaseLayout : "ezDemoBundle::pagelayout.html.twig" %}
{% block content %}
    {# render a simple field #}
    <h3>{{ ez_render_field( content, "title" ) }}</h3>
    {# add a class attribute #}
    {{ ez_render_field(
        content,
        "short_title",
        {
            'attr': { 'class': 'foobar' }
        }
    ) }}
    {# add an id to uniquely address this element #}
    {{ ez_render_field(
        content,
        "author",
        {
            'attr': { 'id': 'authors' }
        }
    ) }}
    {{ ez_render_field( content, "intro" ) }}
    {{ ez_render_field( content, "body" ) }}
{% endblock %}
```

Configuration Settings

Recipe #7: Define Settings in your Bundle using Prepend

Besides the global configuration settings in `ezpublish/config/ezpublish.yml` there are other possibilities to define settings:

<https://confluence.ez.no/display/EZP/Import+settings+from+a+bundle>

We prefer the one which allows the settings to be “prepended” to the normal settings, as no changes to the global settings are needed.

Note: when implementing your own controllers, follow the code in `eZDemoBundle` as it includes additional classes for accessing the repository.

Configuration Settings

Recipe #7: Define Settings in your Bundle using Prepend

src/CjwNetwork/SummerCamp2013/CookBookBundle/DependencyInjection/CjwCookBookExtension.php

PHP

```
use Symfony\Component\DependencyInjection\ContainerBuilder;
use Symfony\Component\Config\FileLocator;
use Symfony\Component\DependencyInjection\Extension\PrependExtensionInterface;
use Symfony\Component\HttpKernel\DependencyInjection\Extension;
use Symfony\Component\DependencyInjection\Loader;
use Symfony\Component\Yaml\Yaml;

class CjwCookBookExtension extends Extension implements PrependExtensionInterface

public function prepend( ContainerBuilder $container )
{
    // Loading our YAML file containing our template rules
    $config = Yaml::parse( __DIR__ . '/../Resources/config/override.yml' );
    // We explicitly prepend loaded configuration for "ezpublish" namespace.
    // So it will be placed under the "ezpublish" configuration key, like in ezpublish.yml.
    $container->prependExtensionConfig( 'ezpublish', $config );
}
```

Override Settings

src/CjwNetwork/SummerCamp2013/CookBookBundle/Resources/config/override.yml

```
# We explicitly prepend config for "ezpublish" namespace in service container extension,  
# so no need to repeat it here  
system:  
  ezdemo_site:  
    location_view:  
      full:  
        article_test:  
          template: "CjwCookBookBundle:full:article.html.twig"  
          match:  
            Identifier\ContentType: article
```

YML

Change the Field Type Template ezauthor ...

src/CjwNetwork/SummerCamp2013/CookBookBundle/Resources/views/fields/field_templates.html.twig

TWIG

```
{# you must inherit from this template in order to use the block() functions !#}  
{% extends "EzPublishCoreBundle::content_fields.html.twig" %}  
  
{% block ezauthor_field %}  
    {% spaceless %}  
        {% if field.value.authors|length() > 0 %}  
            <ul {{ block( 'field_attributes' ) }}>  
                {% for author in field.value.authors %}  
                    <li><a href="mailto:{{ author.email|escape( 'url' ) }}">xx {{ author.name }} xx</a></li>  
                {% endfor %}  
            </ul>  
        {% endif %}  
    {% endspaceless %}  
{% endblock %}
```

... and the Field Type Template ezdatetime

src/CjwNetwork/SummerCamp2013/CookBookBundle/Resources/views/fields/field_templates.html.twig

TWIG

```
{% block ezdatetime_field %}
    {% spaceless %}
        {% if field.value.value %}
            {% if fieldSettings.useSeconds %}
                {% set field_value = field.value.value|localizeddate( 'short', 'medium', parameters.locale ) %}
            {% else %}
                {% set field_value = field.value.value|localizeddate( 'short', 'short', parameters.locale ) %}
            {% endif %}
            xx {{ block( 'simple_block_field' ) }} xx
        {% endif %}
    {% endspaceless %}
{% endblock %}
```

Override Settings for Field Types

src/CjwNetwork/SummerCamp2013/CookBookBundle/Resources/config/override.yml

YML

```
# We explicitly prepend config for "ezpublish" namespace in service container extension,  
# so no need to repeat it here  
system:  
  ezdemo_site:  
    location_view:  
      full:  
        article_test:  
          template: "CjwCookBookBundle:full:article.html.twig"  
          match:  
            Identifier\ContentType: article  
  field_templates:  
    -  
      template: "CjwCookBookBundle:fields:field_templates.html.twig"  
      # Priority is optional (default is 0). The higher it is, the higher your template gets in the list.  
      priority: 10
```




A Brand New Page Layout

... and enjoy!

Adding a New Page Layout for Our TWIG Article

We have prepared a brand new page layout in our bundle. You can find it at

`src/CjwNetwork/SummerCamp2013/CookBookBundle/Resources/views/pagelayout.html.twig`

To use it, change

`src/CjwNetwork/SummerCamp2013/CookBookBundle/Resources/views/full/article.html.twig`

to reference it:

```
{% extends noLayout ? viewbaseLayout : "CjwCookBookBundle::pagelayout.html.twig" %}
```

TWIG

Now articles (only!) are shown using the new page layout.

Note: to set the page layout for all your site, adjust the settings in `ezpublish/config/parameters.yml`



Some More Goodies

any sweets?

Set the Missing HTML Title

In eZDemoBundle the page title is not correctly set. `{{ title|default('Home') }}` is empty, so we need to set it. We look for the legacy path, then for new eZ Publish content and the for our own bundle.

`src/CjwNetwork/SummerCamp2013/CookBookBundle/Resources/views/pagelayout.html.twig`

TWIG

```
{% if ezpublish.legacy.has( 'path' ) %}
    {% set path = ezpublish.legacy.get( 'path' ) %}
    {% set title %}
        CJW Network {% for pathitem in path|reverse %} / {{ pathitem.text }}{% endfor %}
    {% endset %}
{% elseif content is defined%}
    {% set title %}
        CJW Network / {{ content.contentInfo.name }} / (location ID is #{{ location.id }})
    {% endset %}
{% else %}
    {% set title %}
        CJW Network / {{ name | capitalize }}
    {% endset %}
{% endif %}
```

<https://confluence.ez.no/display/EZP/Twig+Helper>

Include an Old eZ Publish Template (.tpl)

To include an old eZ Template, e.g.

`ezpublish_legacy/extension/ezdemo/design/ezdemo/templates/footer/latest_news.tpl` into your TWIG page layout, use the following code:

`src/CjwNetwork/SummerCamp2013/CookBookBundle/Resources/views/pagelayout.html.twig`

```
{% block latest_news %}  
    {% include "design:footer/latest_news.tpl" %}  
{% endblock %}
```

TWIG

Include an Old eZ Publish Template (.tpl)

You can also pass parameters to an old template: change the fetch in

`ezpublish_legacy/extension/ezdemo/design/ezdemo/templates/footer/latest_news.tpl`

to ('content', 'tree', hash('parent_node_id', \$parent_node_id, ...)

and set the variable `{ $parent_node_id }` in your TWIG page layout

`src/CjwNetwork/SummerCamp2013/CookBookBundle/Resources/views/pagelayout.html.twig`

```
{% block latest_news %}
    {% include "design:footer/latest_news.tpl" with {"parent_node_id": 2} %}
{% endblock %}
```

TWIG

Override Several Blocks of the Page Layout

You can override several blocks of the TWIG page layout in a template that inherits from it.

`src/CjwNetwork/SummerCamp2013/CookBookBundle/Resources/views/full/article.html.twig`

```
{% block latest_news %}  
    We set the content here for the latest_news block in our page layout  
{% endblock %}  
  
{% block content %}  
    ...  
{% endblock %}
```

TWIG

<https://confluence.ez.no/display/EZP/Legacy+code+and+features#Legacycodeandfeatures-LegacyTemplateinclusion>



One More Thing...

Globally Overriding Resources!

Recipe #8: Override resources from bundles using the `ezpublish/Resource` folder

Symfony allows for globally overrides of resources. You can teach eZ Publish the same trick!

Suppose you want to override the eZDemo page layout:

- In the `ezpublish` folder, create a `Resources` folder
- Replicate the directory structure for the elements you want to override
- e.g. `Resources/eZDemoBundle/views` - use the correct bundle name!!!
- Place your files there

<Thank You!>

Ekkehard Dörre

<http://share.ez.no/community/profile/7431>

[@ekkeD](#)

<http://www.coolscreen.de>

Donat Fritschy

<http://share.ez.no/community/profile/10451>

[@webmanufaktur](#)

<http://www.webmanufaktur.ch>

cjw