

Hybrid approach with eZ Publish 5

*Ivo Lukač @ eZ Publish Summer Camp 2014
hands-on prepared by Ivan Herak*

**Hybrid means using both:
legacy and new stack in
optimal way**

What is optimal?

Legacy stack

- not improved any more
- requires traditional eZ developers
- lot of projects/features/extensions out there, would need years to refactor it all in eZ5/Symfony

New stack

- has lots of new features that are very interesting (e.g. http cache)
- requires developers who know Symfony
- lot of features exist as Symfony bundles

**Combining the two is difficult
but doable**

But its not for every situation

When not to use hybrid?

- if by doing refactoring you don't add any new value to the client / end user
- if the new project will be done completely (or almost) in the new stack
- if you don't know both stack well

When to use it then?

- if you have an existing legacy project or extension which needs to be improved but there is no budget/incentive to refactor everything
- you need to know both stacks well

**Build/compose/implement
new code with new stack**

**Inject new code into legacy in
such a way that its future
proof, and vice versa**

Injecting

- use closure for adding legacy code into new stack
- use ngsymfonytools and Symfony service containers for adding new stack code into legacy

Example #1

- we are going to call a feature from a Symfony bundle in a legacy eZ module
- Bundle: HTMLPDFAPI for converting HTML to PDF
- Legacy: ngpdf/download

Injecting service

- go to extension nghybridworkshop
- add to modules/ngpdf/download.php

```
$container = ezpKernel::instance()->getServiceContainer();  
$htmlpdfapi = $container->get( 'netgen_html_pdf_api' );  
$ezcontent = $container->get( 'ez_content' );
```

Use the service

- add to modules/ngpdf/download.php

```
$html = $sezcontent->viewLocation($nodeID, 'full')->getContent();  
  
$params = array('html' => $html);  
  
$pdfFile = $htmlpdfapi->generateFromHTML($params);  
  
header('Content-Type: application/pdf');  
  
header('Content-Disposition: attachment;filename='.$nodeID.'.pdf');  
  
echo $pdfFile->getBody(true);
```

```
git checkout Step1 --force
```


Example #2

- we are going to do a controller action which uses the same service
- Bundle: HTMLPDFAPI for getting the credit amount left
- Legacy: call the action from the legacy template with ngsymfonytools

New way of doing it

- go to HybridApproachBundle
- add to Controller/DefaultController.php getCreditsAction()

```
$htmlpdfapi = $this->container->get( 'netgen_html_pdf_api' );  
  
$credits = $htmlpdfapi->getCredits();  
  
$response = new Response();  
  
$response->setContent($credits->getBody(true));  
  
return $response;
```

ngsymfonytools magic

- go to nghybridworkshop
- add to design/standard/templates/ngpdf/view.tpl

<h2>

Available credits:

```
{symfony_render(  
symfony_controller( 'NetgenHybridApproachBundle:Default:getCredits' )  
)}
```

</h2>

```
git checkout Step2 --force
```

Example #3

- we are going to build a bundle that uses old fetches so they are replaceable
- Bundle: NetgenSearchAndFilter
- Legacy: content list fetch

Override full template

- go to HybridApproachBundle
- add to Resources/config/hybridapproach.yml (already added to ezpublish.yml)

```
ezpublish:
  system:
    ezdemo_site_group:
      location_view:
        full:
          netgen_search_and_filter_set:
            template: NetgenHybridApproachBundle:Default:filter_results.html.twig
            match:
              Id\Location: 75
```

```
git checkout Step3 --force
```

Building simple form

- go to HybridApproachBundle
- add to Form/Type/FilterSearchType.php buildForm()

```
$builder->add( 'price', 'choice', array(
    'label' => 'Filter by price:',
    'empty_value' => 'Choose an option',
    'choices' => array( '20' => '<20',
                        '40' => '<40',
                        '60' => '<60' )
) )->add( 'save', 'submit', array( 'label' => 'Filter!' ) );
```


Create a form template

- go to HybridApproachBundle
- create Resources/views/forms folder
- create Resources/views/forms/form.html.twig :

```
{{ form(form) }}
```

Use the search&filter

- go to HybridApproachBundle
- change the Resources/config/hybridapproach.yml:

```
ezpublish:
  system:
    ezdemo_site_group:
      location_view:
        full:
          netgen_search_and_filter_set:
            controller: NetgenSearchAndFilterBundle:Search:searchLocation
            match:
              Id\Location: 75
```

Add form to full view

- go to HybridApproachBundle
- add in Resources/views/Default/filter_results.html.twig

```
{% include form_template with  
    { form: form, form_action: path( "ez_urlalias", { locationId: location.id } ) }  
%}
```

```
git checkout Step4 --force
```

Overriding service

- go to HybridApproachBundle
- add in Resources/config/services.yml

```
netgen_search_and_filter.location_75:  
  class: Netgen\SearchAndFilterBundle\Components\SearchAdapter  
  arguments:  
    - @netgen_hybrid_approach.handler.legacy  
    - @netgen_hybrid_approach.form_type.filter_search_form_type  
    - @form.factory  
    - @netgen_hybrid_approach.criteria_builder.legacy_list_filter  
    - @netgen_hybrid_approach.result_converter.legacy  
    - NetgenHybridApproachBundle:Default:filter_results.html.twig  
    - NetgenHybridApproachBundle:forms:form.html.twig  
    - 10
```

Create legacy handler

- go to HybridApproachBundle
- add in Components/SearchHandler/LegacyListHandler.php

```
$legacyKernelClosure = $this->kernel;

$searchResults = $legacyKernelClosure()->runCallback(
    function () use ( $criteria )
    {
        return eZFunctionHandler::execute( 'content','list', $criteria );
    }
);
```

Create legacy criterion

- go to HybridApproachBundle
- add in Components/SearchCriteriaBuilder/LegacyFilterCriteriaBuilder.php

```
return array(  
    'parent_node_id' => 75,  
    'extended_attribute_filter' =>  
  
        array(  
            'id' => 'PriceAttributeFilter',  
            'params' => array( 'price' => $data['price'] )  
        ),  
  
    'sort_by' => array('attribute', true, 'product/price'));
```

Finnish up (1/2)

- go to HybridApproachBundle
- add in Resources/views/Default/filter_results.html.twig

```
{% if show_results %}

    <h2>{{ 'Search results'|trans }}</h2>


{% else %}


{% endif %}
```


Finnish up (2/2)

```
<p>{{ 'Found:'|trans }} {{ pager.getNbResults() }}</p>
<section class="content-view-full">
  <div class="class-folder row">
    <div class="span8">
      <section class="content-view-children">

        {% for hit in pager %}

          {{ render( controller( "ez_content:viewLocation",
                                {
                                  "locationId": hit.valueObject.contentInfo.mainLocationId,
                                  "viewType": "line"
                                }
                              )) }}

        {% endfor %}

      </section>

    </div>
  </div>
</section>
```

```
git checkout Step5 --force
```

Rate workshops

<http://vote.netgenlabs.com>