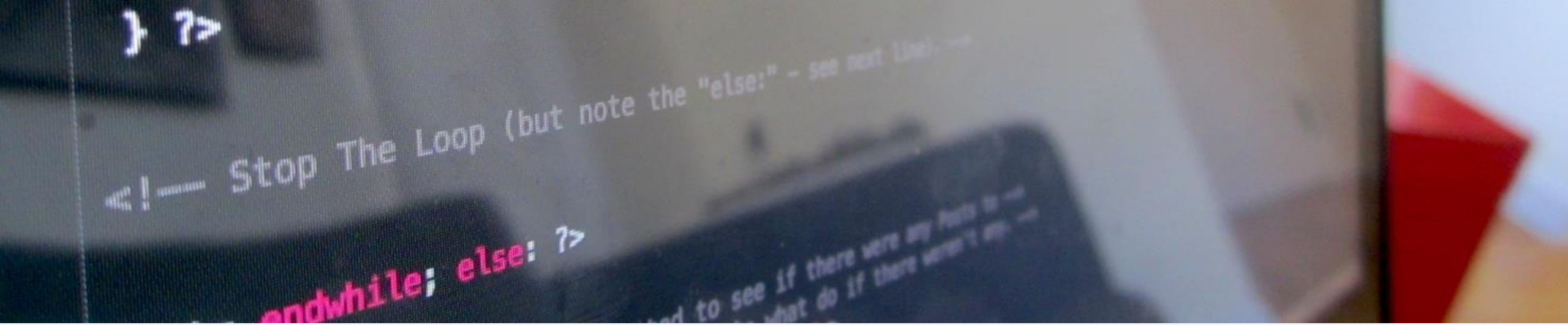
NetBeans Day 2017

Busting the myths about PHP
Up and running with PHP MVC Applications using NetBeans

Adamopoulos Panagiotis

Senior Software Engineer @ UniSystems



Panagiotis Adamopoulos
Senior Software Engineer @ UniSystems

Web Developer since 1999, beginning from Microsoft and ASP and turned to Linux World in 2003

Active in open source projects like Drupal, Symfony, Linux Foundation etc

panagiotis.adamopoulos@gmail.com
https://gr.linkedin.com/in/padamopoulos

Backend Development

- Develop in PHP, Codeigniter, Symfony
- Develop in Python, Web2PY
- Develop in Java, Spring and Vaadin Framework

Frontend Development

- Develop in Twig, AngularJS and React

Books

- Developing Enterprise Application with Symfony Framework, 2016, Desmos Publications
- Drupal 8 Security Cookbook, 2017, Under Publication by Desmos Publications

TOPICS

PHP Myth Busting in Business Application Development

General Data - History

Preview some milestones of PHP and Java History

PHP for the Enterprise – Symfony Framework

Using PHP Symfony Framework for building enterprise web applications

PHP Programming

Object and Aspect Oriented Programming in PHP

Java vs PHP

There are not so many differences as we think!

PHP MVC Frameworks

The MVC model advantages for PHP development

NetBeans Advantages for developers

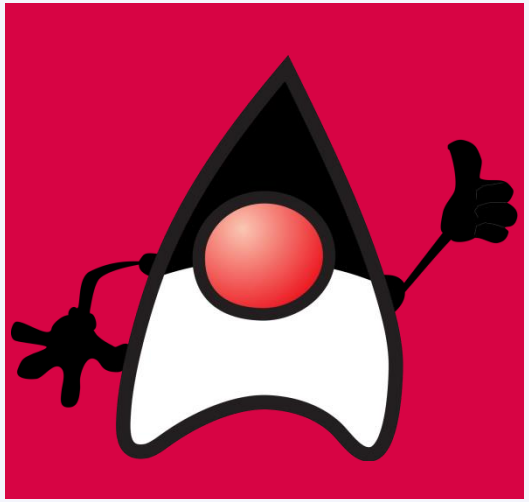
What makes NetBeans our favorite IDE.

NETBEANS

Not a duel but a matching game.

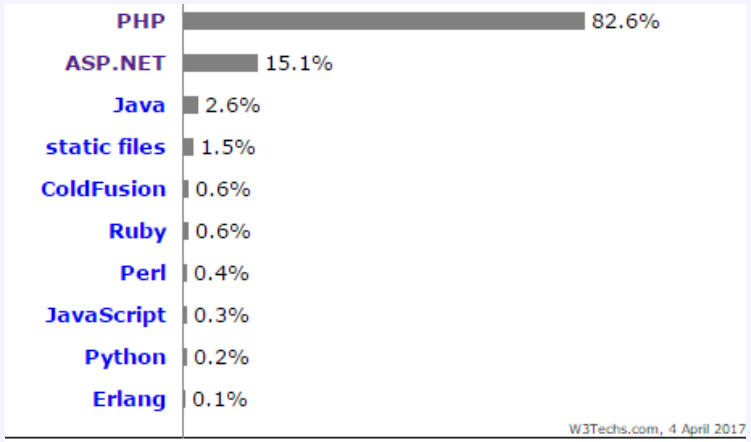
Often developers, especially ones who work primarily with PHP or Java, will argue about the overarching superiority of one language to the other. This can get fairly charged, and commentators write in high-level terms.

Either you use PHP or Java, NetBeans is an ideal IDE for developing great apps.



JAVA

A few facts about JAVA



Percentages of websites using various server-side programming languages
Note: a website may use more than one server-side programming language

2,6%
Server Side
Programming
Languages
Market Penetration

1996

The first version was released on January 23, 1996 and called Oak. The first stable version, JDK 1.0.2, is called Java 1.

2004

This is the first release in a long time containing major enhancements to the core language itself. In this release the numbering scheme has also changed.

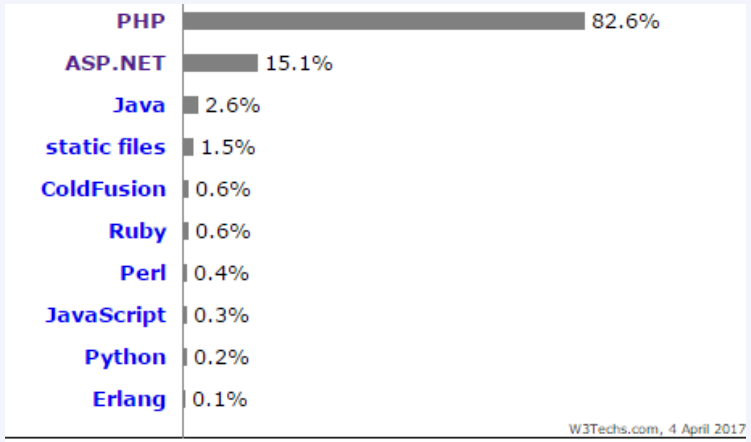
2006

Although Java 6 is not nearly as massive an upgrade as Java 5 was, Sun says that Java Platform, Standard Edition 6 is a major feature release.

PHP is one of the top Job Promising Programming Languages

PHP

A few facts about PHP



Percentages of websites using various server-side programming languages
Note: a website may use more than one server-side programming language

82,6%
Server Side
Programming
Languages
Market Penetration

1994

Rasmus Lerdorf created PHP. It was just a set of CGI scripts written in C.

2004

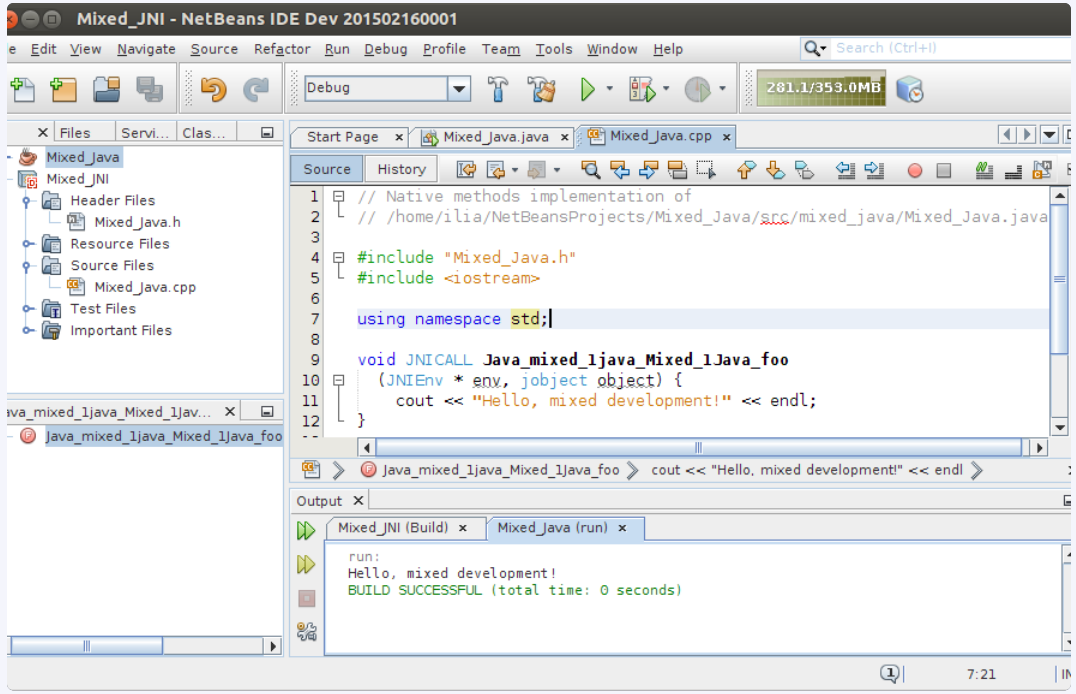
PHP 3.0 was the first version that closely resembles PHP as it exists today.

2006

PHP became Object Oriented. It is mainly driven by its core, the Zend Engine 2.0 with a new object model and dozens of other new features.

NETBEANS

Necessary Tools for PHP Application Development



Begin developing PHP Applications needs no more than 10 minutes.

What is needed:

1. A LAMP / WAMP / MAMP Web server

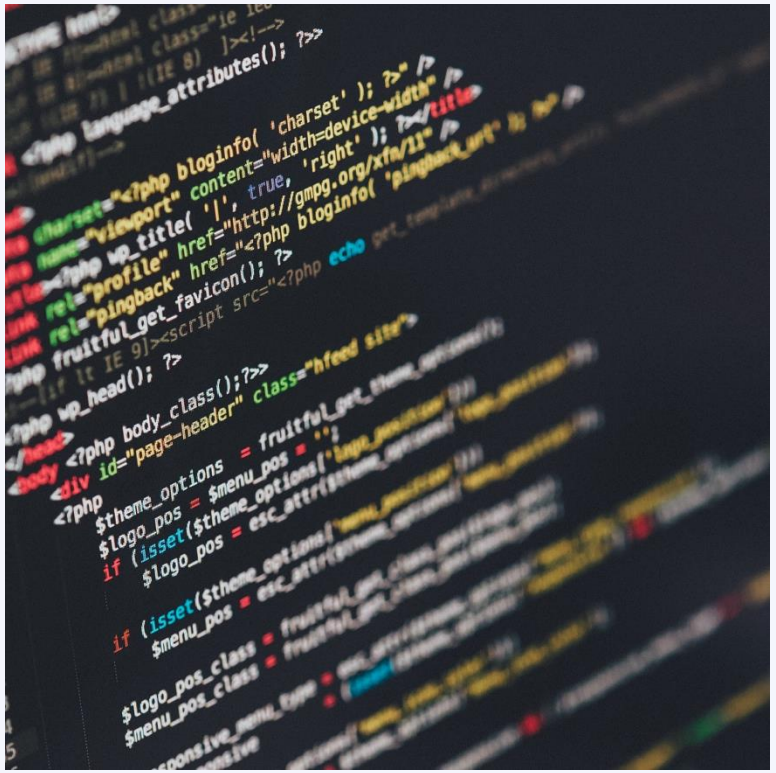
2. An Integrated Development Environment

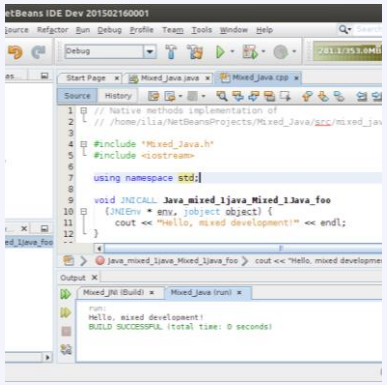
3. Some extra tools, like composer, which is a php dependency manager.

MYTHS

Common myths about PHP

1. It's not a real language--you can't do proper object-oriented designs etc
2. You can't compile PHP, so it will always be slow
3. PHP is crappy because it's easy to hijack with all those global variables
4. You can't develop in PHP as fast as other languages like Ruby on Rails
5. PHP is only good for web sites - it's no good for anything else





PHP Programming

Object Oriented Programming in PHP

PHP began as a set of CGI scripts and continued to be a scripting language until version 4.

On July 13, 2004, PHP 5 was released, powered by the new Zend Engine II and included full support for Object Oriented Programming.

Current version 7 is considered to be ready for a successful migration to a just-in-time (JIT) compiler (compilation done during execution of a program – at run time – rather than prior to execution).

PHP Programming

PHP features that remind us of Java charms

Multiple Inheritance with PHP Traits.
Traits are a mechanism for code reuse in single inheritance languages. A Trait is similar to a class, but only intended to group functionality in a fine-grained and consistent way.

Object Relational Mapping
The Doctrine Project is the home to several PHP libraries primarily focused on database storage and object mapping. The core projects are a Object Relational Mapper (ORM) and the Database Abstraction Layer (DBAL) it is built upon. Doctrine has greatly benefited from concepts of the Hibernate ORM and has adapted them to fit the PHP language.

PHP Annotations
Although annotations are not natively built, php-annotations library implements a complete, “industrial strength” annotation engine for PHP

MYTHS

Common myths about PHP

1. It's not a real language - you can't do proper object-oriented designs etc

BUSTED

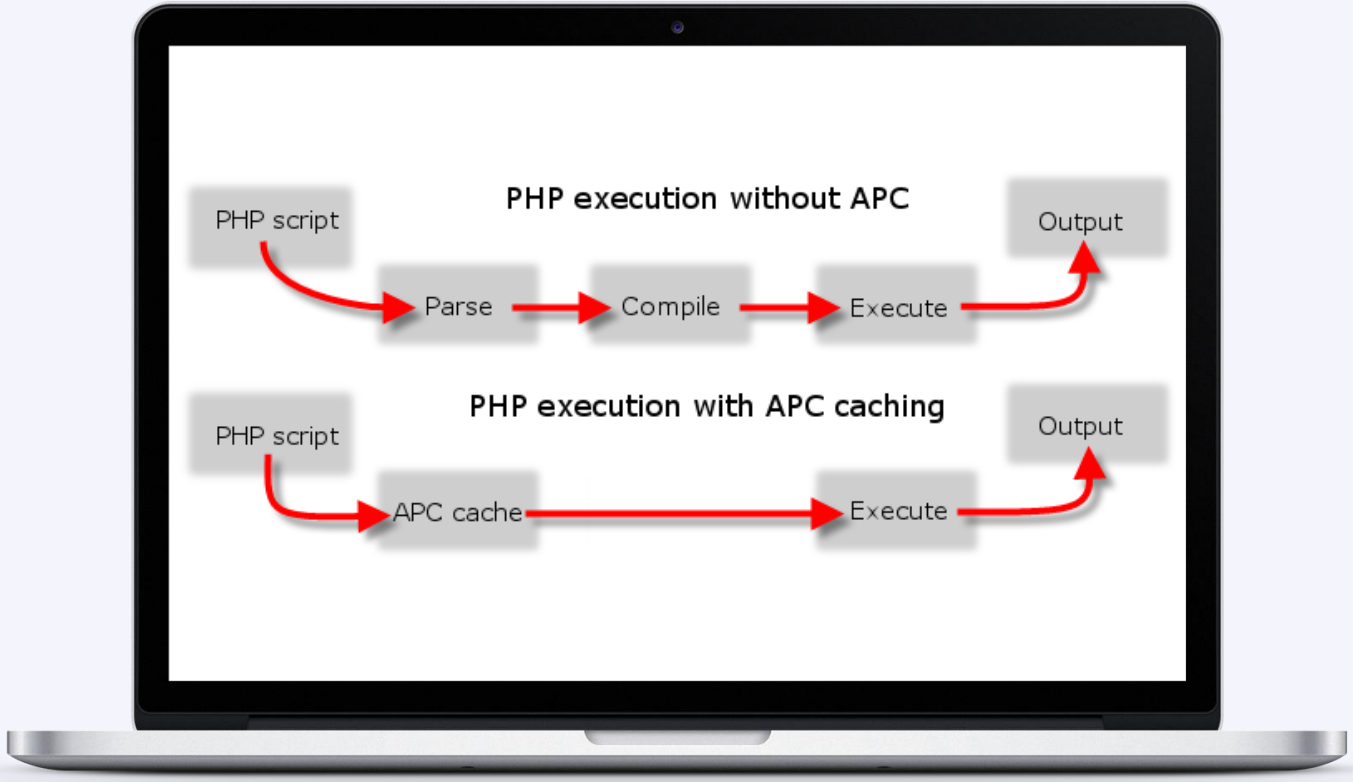
2. You can't compile PHP, so it will always be slow

3. PHP is crappy because it's easy to hijack with all those global variables

4. You can't develop in PHP as fast as other languages like Ruby on Rails

5. PHP is only good for web sites - it's no good for anything else

You can't compile PHP, so it will always be slow



PHP is an interpreted language, and it doesn't have a built-in compiler.

Python has a built-in runtime compiling system, so you get compiled byte-code without having to do anything.
But you can accelerate PHP quite similar to Python with **opcode cache.**

When PHP is executed, the interpreter makes two passes: first a conversion to native bytecode, and then execution of the bytecode. An opcode cache stores that first pass to disk, so subsequent calls can use what is essentially the same as compiled code.

MYTHS

Common myths about PHP

1. It's not a real language--you can't do proper object-oriented designs etc

BUSTED

2. You can't compile PHP so it will always be slow

BUSTED

3. PHP is crappy because it's easy to hijack with all those global variables

4. You can't develop in PHP as fast as other languages like Ruby on Rails

5. PHP is only good for web sites - it's no good for anything else

MYTHS

It's easy to hijack with all those global variables

This point tends to be funny…

PHP has two particularly annoying "features" that have turned out to be security nightmares, originally there to make it simple to program:
register_globals and magic_quotes_gpc.

These vulnerabilities are widely known, and PHP has been set with register_globals turned off for several years now.

For any experienced PHP programmer, these are solved problems.

MYTHS

Common myths about PHP

1. It's not a real language--you can't do proper object-oriented designs etc

BUSTED

2. You can't compile PHP so it will always be slow

BUSTED

3. PHP is crappy because it's easy to hijack with all those global variables

BUSTED

4. You can't develop in PHP as fast as other languages like Ruby on Rails

5. PHP is only good for web sites - it's no good for anything else

MYTHS

You can't develop as fast as other languages like Ruby on Rails

Ok. Now we're getting to the ridiculous one.

First off, Rails isn't a language, it's a framework. And by many accounts, it's a good one, providing a lot of really powerful features right out of the box.

Symfony or Laravel are the rails of PHP.

While Ruby may be a nice language, there's a lot more support for PHP right now, in available talent, web servers, scaling experience, and breadth of libraries available.

MYTHS

Common myths about PHP

1. It's not a real language - you can't do proper object-oriented designs etc

BUSTED

2. You can't compile PHP so it will always be slow

BUSTED

3. PHP is crappy because it's easy to hijack with all those global variables

BUSTED

4. You can't develop in PHP as fast as other languages like Ruby on Rails

BUSTED

5. PHP is only good for web sites - it's no good for anything else

MYTHS

PHP is only good for web sites

Let's think of some well known web applications:

- Wikipedia
- Facebook
- Parts of Yahoo
- Flickr
- The White House

If these are not enterprise enough, then YES PHP is not enough for enterprise applications.

We will talk again later about Enterprise PHP applications

MYTHS

Common myths about PHP

1. It's not a real language - you can't do proper object-oriented designs etc

BUSTED

2. You can't compile PHP so it will always be slow

BUSTED

3. PHP is crappy because it's easy to hijack with all those global variables

BUSTED

4. You can't develop in PHP as fast as other languages like Ruby on Rails

BUSTED

5. PHP is only good for web sites - it's no good for anything else

BUSTED

MVC

PHP MVC Frameworks

Advantages of using them

- Frameworks support almost all database connections like SQL, My SQL, Oracle and ODBC.
- PHP frameworks follow MVC (Model View Controller) architecture which is a great combination of database application (model), HTML coding (view) and input/ output instructions (controller).
- Make your code lightweight, sharp and secure and they speed up development
- As MVC pattern is used, the code gets optimized hence runs faster.
- They have several creative functions that are prepared to have an edge in coding.
- There are functions for handling date formats, database connections, handling emails, editing strings, etc.
- Provide Unit Testing and Debugging methodologies.

Further Read : http://symfony.com/why-use-a-framework

SYMFONY

MVC Architecture

Handle data and business logic → **Model**

Present data to the user in any supported format and layout → **View**

Receive user requests and call appropriate resources to carry them out → **Controller**

SYMFONY

Installation

✓ **Wamp/ Lamp Server :**

http://www.wampserver.com/en/
https://www.digitalocean.com/community/tutorials/how-to-install-linux-apache-mysql-php-lamp-stack-on-centos-6

✓ **Netbeans PHP or Full version**

https://netbeans.org/downloads/

✓ **Symfony:**

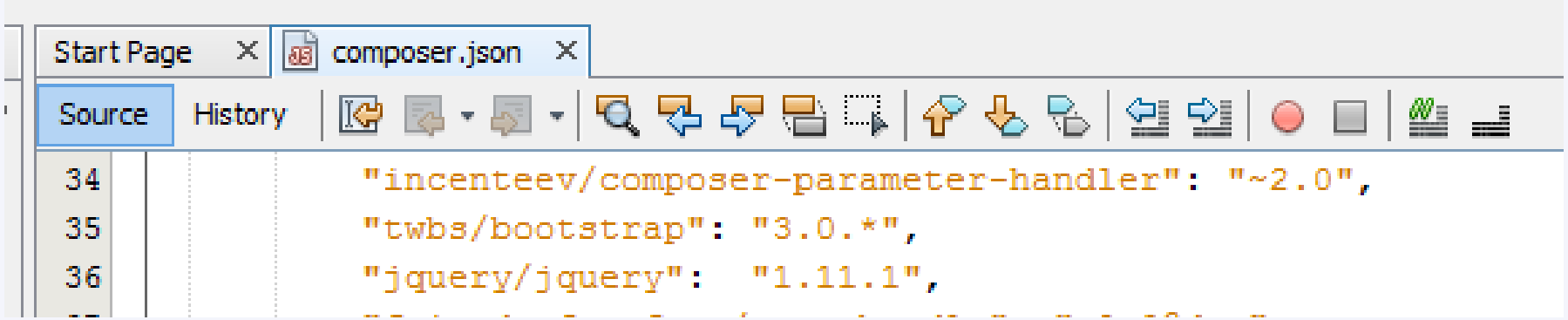
http://symfony.com/download

SYMFONY

Composer

Composer is a tool for dependency management in PHP. It allows you to declare the libraries your project depends on and it will manage (install/update) them for you.

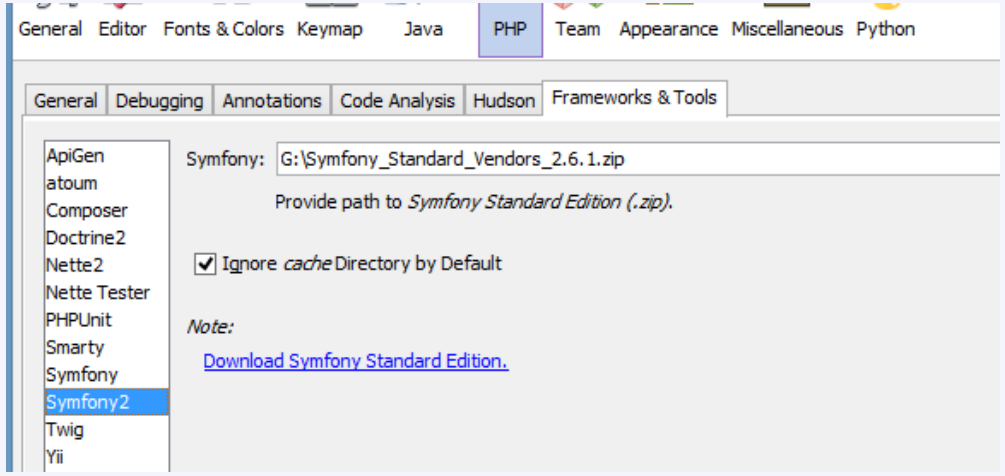
Reminds of Maven?



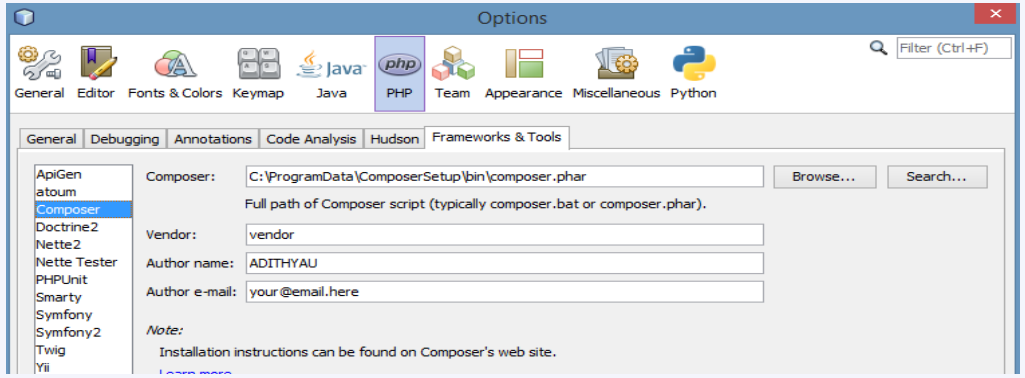
SYMFONY

Netbeans IDE

Point the Symfony ZIP file location in
Tools >>Option >> PHP >> Frameworks and Tools

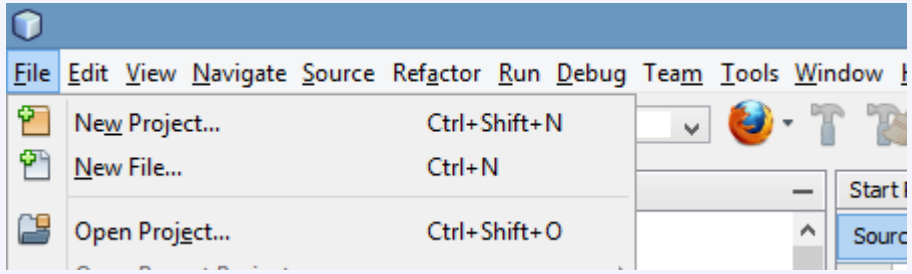


Point the Composer.Phar file location in
Tools >>Option >> PHP >> Frameworks and Tools

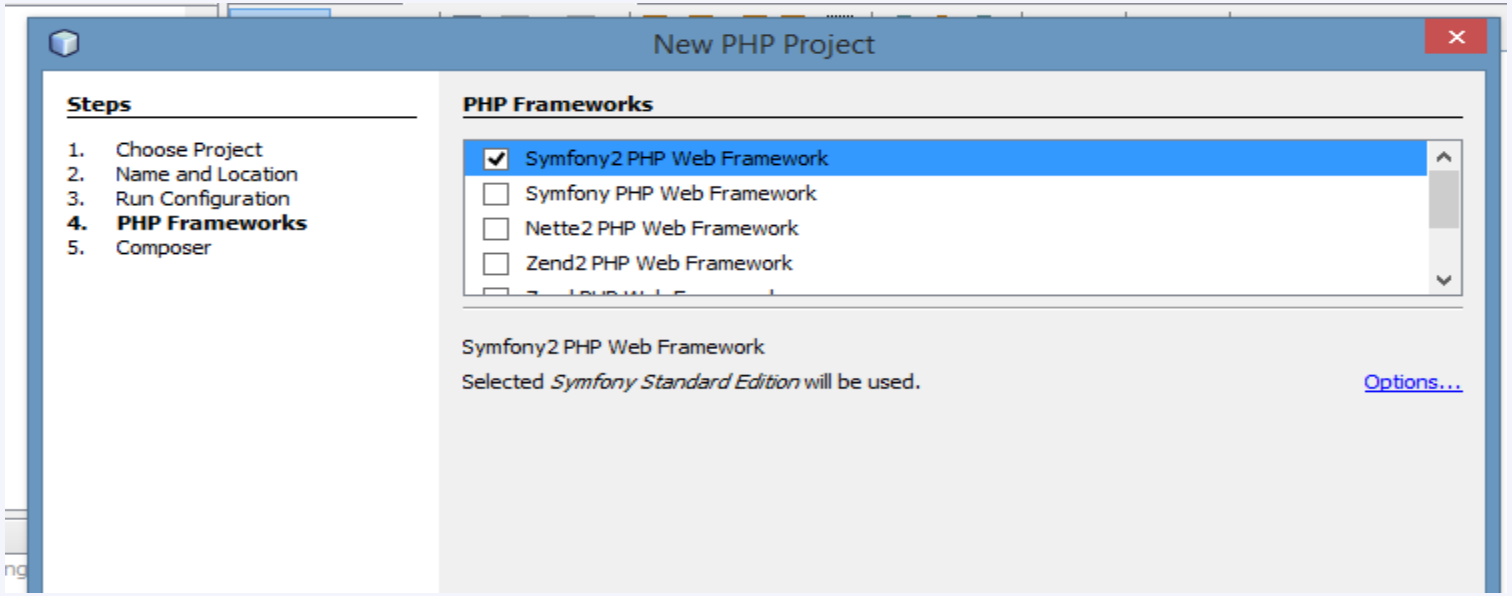


SYMFONY

Create a new Symfony Project in NetBeans



Installation and usage:
https://www.youtube.com/watch?v=gmtMjzrjlLY



SYMFONY

Create a new Symfony Project in NetBeans

✓ Install XDebug using Wizard: http://xdebug.org/wizard.php

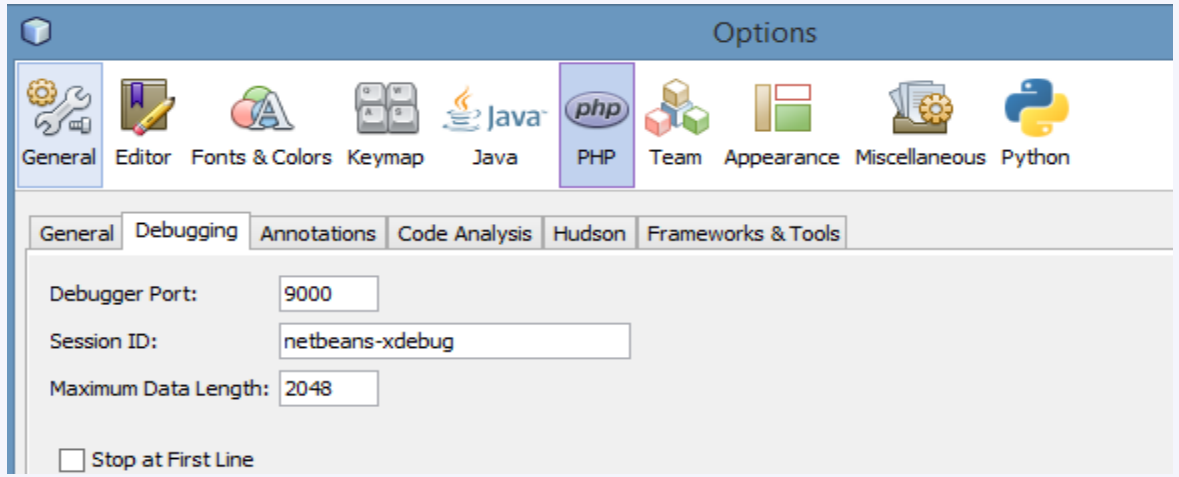
✓ Configure the XDebug : Change the PHP.ini to add extension and configure

```
[xdebug]
zend_extension = "xxxx"
xdebug.remote_enable=1
xdebug.remote_host=127.0.0.1
xdebug.remote_mode=req
xdebug.remote_port=9000
xdebug.remote_handler=dbgp
xdebug.idekey=netbeans-xdebug
```

SYMFONY

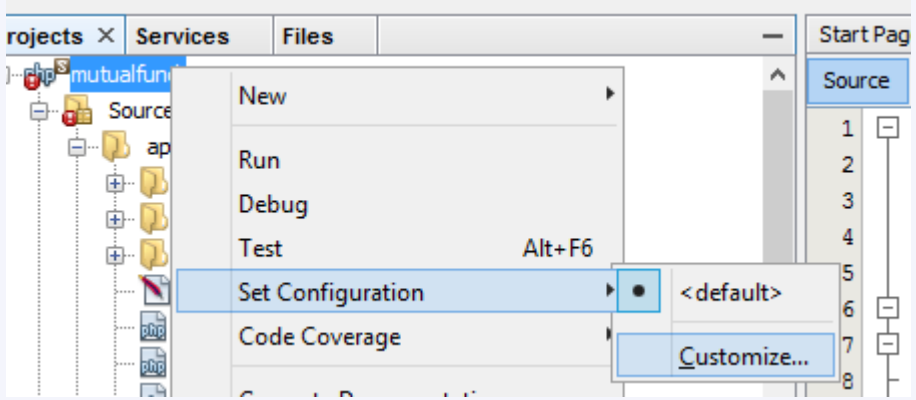
Debugging in NetBeans

✓Configure the Netbeans



✓Create a run configuration for the web link

✓Put breakpoint and run debugging



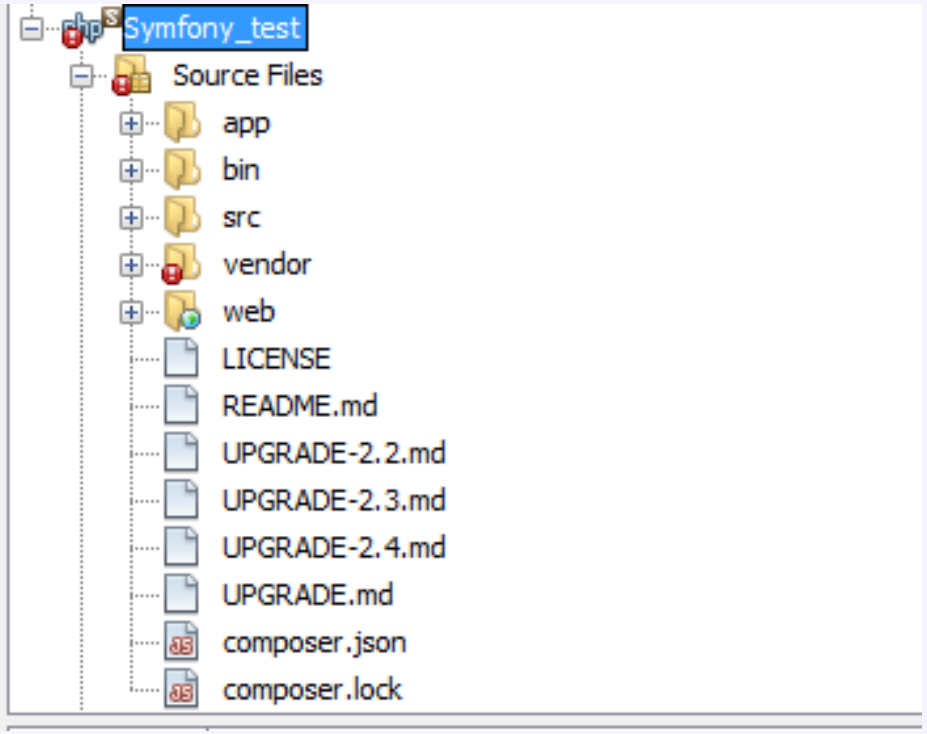
SYMFONY

Hello World

Navigate to yoursitepath/web/app_dev.php/demo

SYMFONY

Application Structure



app:
The application configuration, templates and translations.

src:
The project's PHP code.

vendor:
The third-party dependencies.(Read Only)

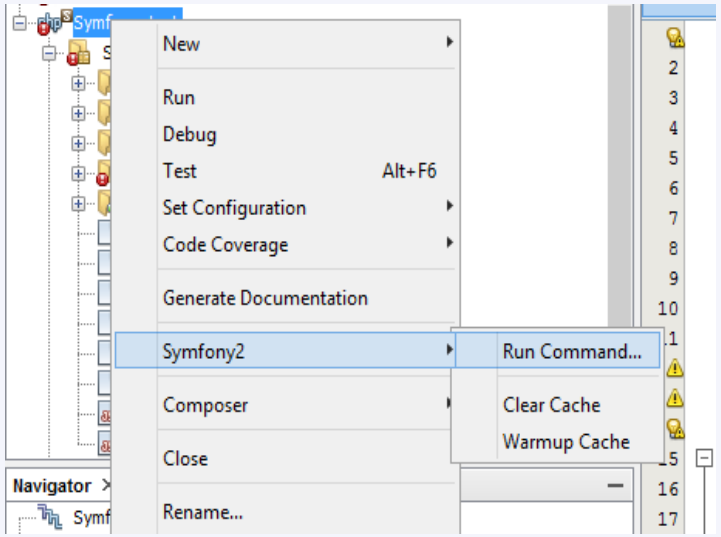
web:
The web root directory.

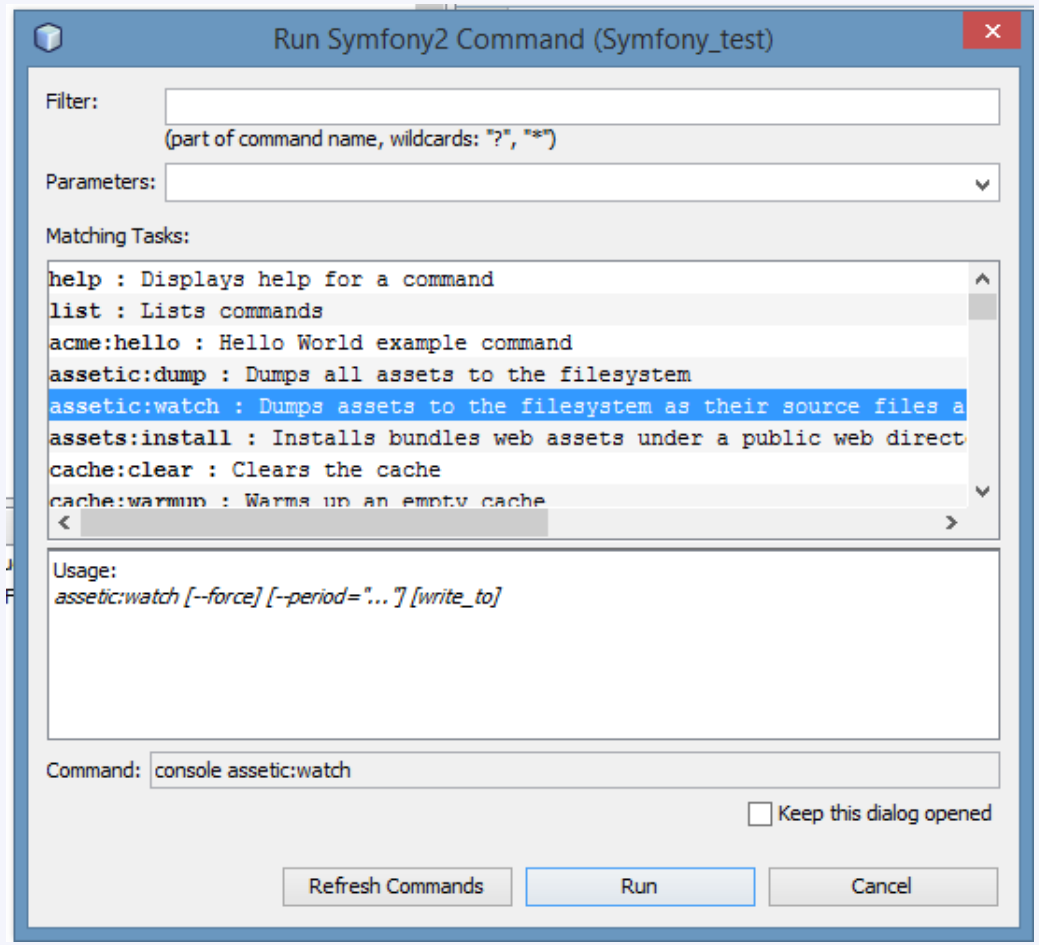
composer.json: Composer file which indicates additional packages/libraries required

SYMFONY

Symfony Commands

Its a CLI for symfony.
Symfony commands can be used to run commonly performed actions in Symfony.

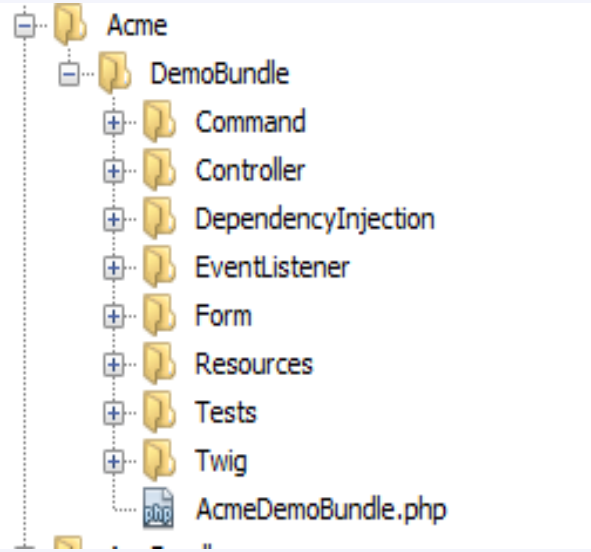


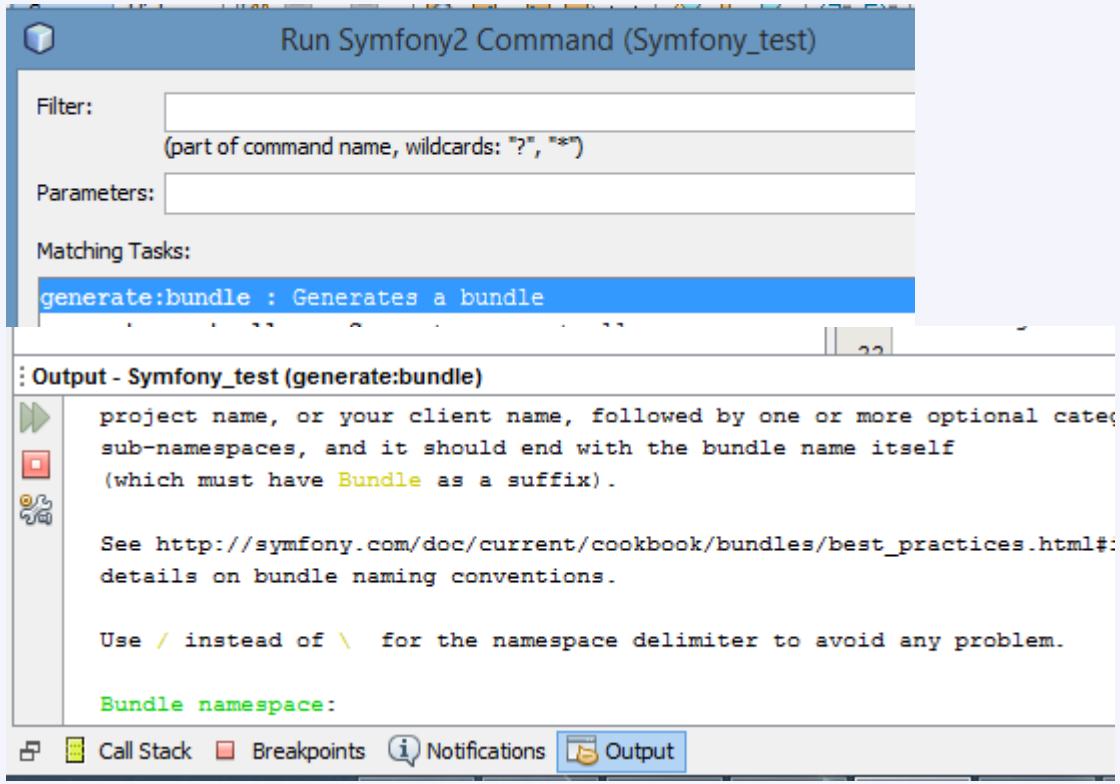


SYMFONY

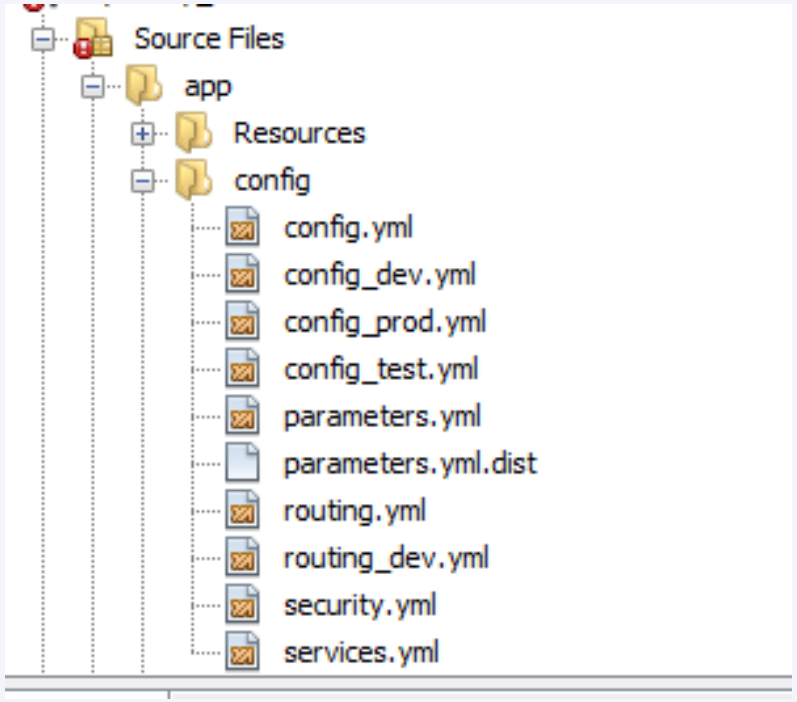
Bundles

A bundle is nothing more than a directory that houses everything related to a specific feature, including PHP classes, configuration, stylesheets, JS files etc,. Creation of Bundle can be done using generate:bundle command





Config Files



config.yml:
Main config file for doctrine,twig and other components

parameters.yml:
holds the values for the literals/parameters used in config

routing.yml:
Main router of the symfony

security.yml:
Controls access levels permissions etc

SYMFONY

Type of config files

Annotations

```
15      /**
16       * @var integer
17       *
18       * @ORM\Column(name="id", type="integer")
19       * @ORM\Id
20       * @ORM\GeneratedValue(strategy="AUTO")
21       */
22      private $id;
```

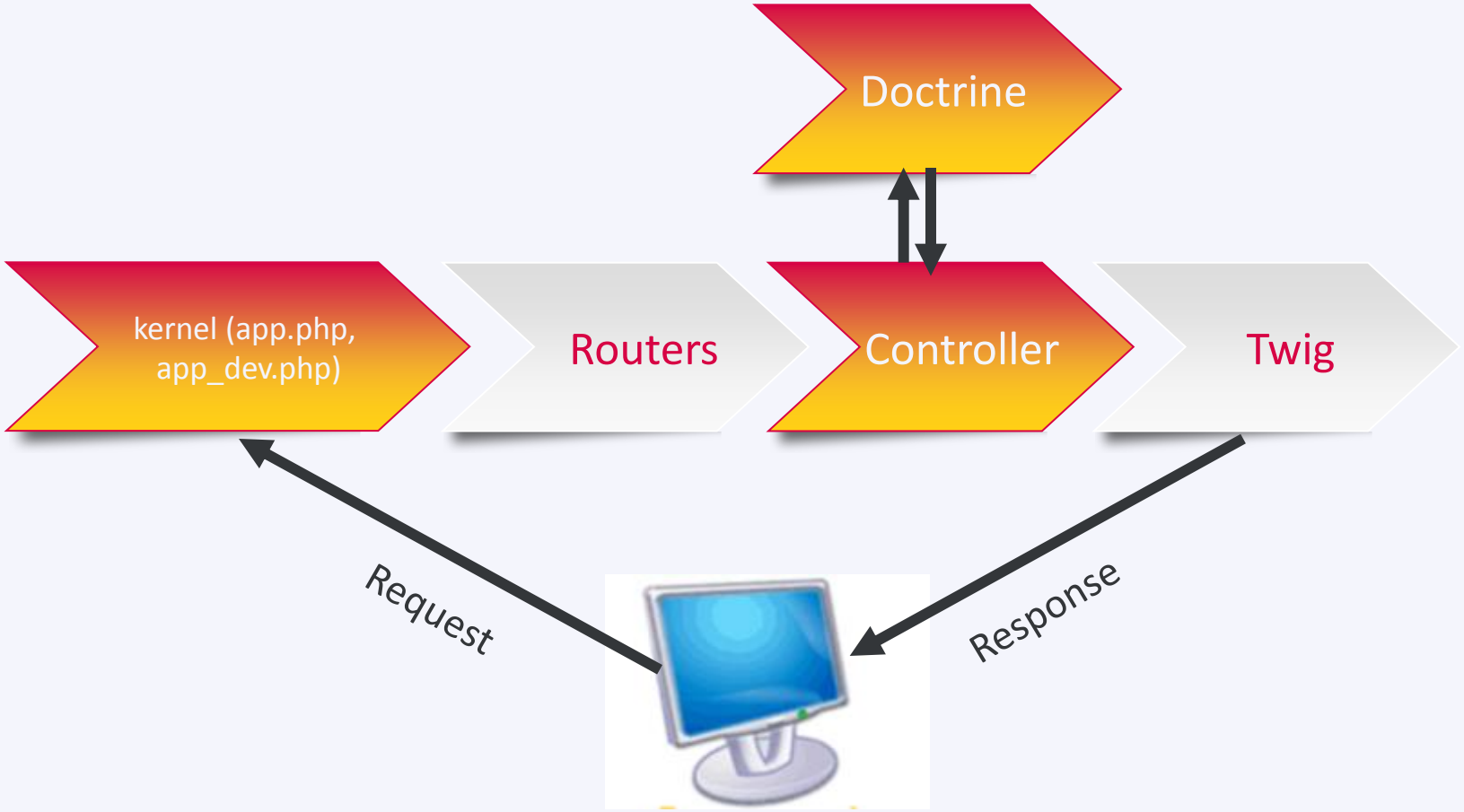
YML file

```
1  mftrack_homepage:
2      path:     /hello/{name}
3      defaults: { _controller: mftrackBundle:Default:index }
4  mftrackBundle_mutualfunds:
5          resource: "@mftrackBundle/Resources/config/routing/mdata.yml"
6          prefix:   /mutualfunds
7  mftrackBundle_nav:
8          resource: "@mftrackBundle/Resources/config/routing/navdata.yml"
9          prefix:   /nav
10 mftrackBundle_transact:
11         resource: "@mftrackBundle/Resources/config/routing/tdata.yml"
12         prefix:   /transact
13 fundbyname:
```

My recommendations:

Annotations for Entity
Rest all YML file

Symfony Architecture



SYMFONY

Routers

Decides the controller method to be called based on the URL path. Can take the parameter values from URL and pass it on to Controller.

```
13  fundbyname:
14      path:     /funds/{mfcode}
15      defaults: { _controller: "mftrackBundle:fundsdata:mfdetails" }
16  fundsindex:
17      path:     /funds
18      defaults: { _controller: "mftrackBundle:fundsdata:index" }
```

Powerful enough to differenciate between GET and POST requests

SYMFONY

Controller

Acts like a linker between business object and the presentation layer.
Router will pass the control to specific function in the controller class along with parameters from URL.

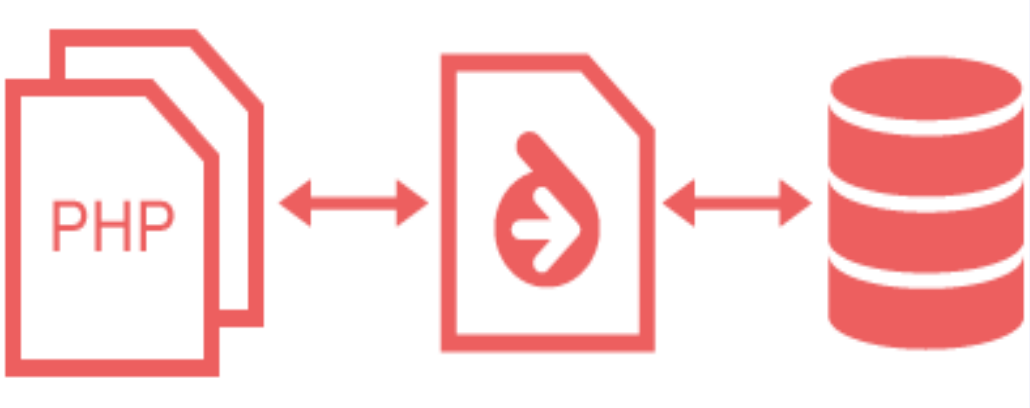
```
25      /**
26       * @Route("/hello/{name}", name="_demo_hello")
27       * @Template()
28       */
29      public function helloAction($name)
30      {
31          return array('name' => $name);
32      }
```

Main job is to get the data from business object, process it and prepare the data to be displayed/sent back to user.

Doctrine: ORM

Model part in MVC is handled by Doctrine in Sympony.

It acts like an ORM and Database Abstraction Layer.



Also provides persistence services to the application layer reducing the load to the database.

Uses of ORM

- Perfect database abstraction: Can migrate to another DB in one simple shot

- Knows the relationships between the database.

- Handles all validations, before commit operations etc., there by reducing significant code.

- Provides a persistent layer for intermediate operations and large I/O database operations.

SYMFONY

Entity

```php
<?php

namespace mf\mftrackBundle\Entity;

use Doctrine\ORM\Mapping as ORM;

/**
 * mdata
 *
 * @ORM\Table(name="mdata")
 * @ORM\Entity
 */
class mdata
{
    /**
    * @var integer
    *
    * @ORM\Column(name="mfcode", type="integer")
    * @ORM\Id
    */
    private $mfcode;

```

Entity files are used to describe the table properties and schema. This can be used to then create the same in underlying database.

Has following information:

- Table name
- Field properties
- Setter/Setter functions
- Associations/Relationships
- Events

SYMFONY

Creation of Entity Files

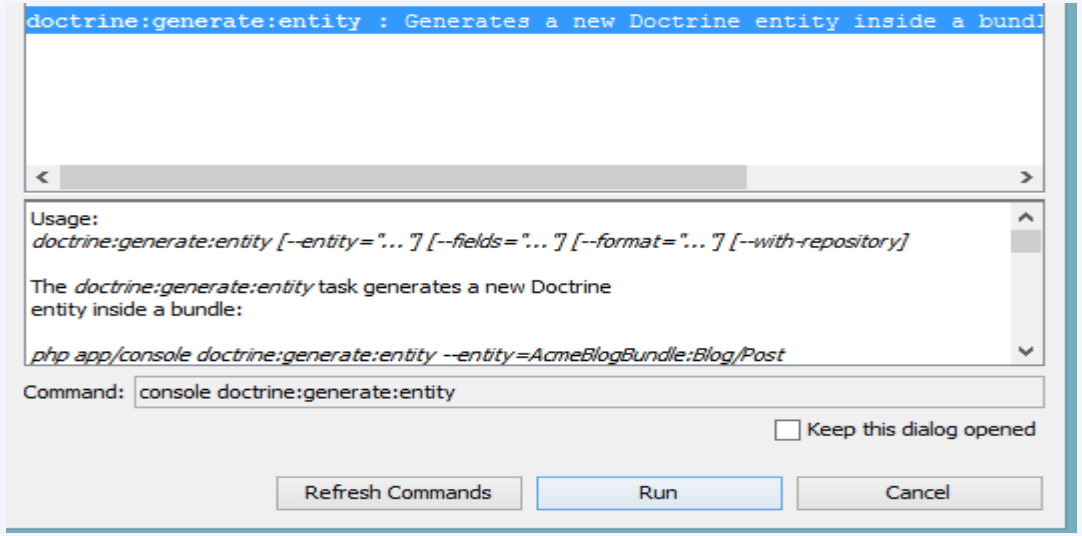
We can use Sympony commands to automate most of the tasks related to Entity

doctrine:generate:entity => Will create all the required entity files

doctrine:database:create =>Save the schema and table in database

doctrine:schema:create=> Create the tables with the new schema

doctrine:schema:update=> Update the DB with the new schema



SYMFONY

Doctrine in action

Will take some time to master the Doctrine commands alternatives for 'SELECT' and other DB operations command.

```
        $em = $this->getDoctrine()->getManager();

//        $repository = $em
//            ->getRepository('mftrackBundle:tdata');
//
//        $tdata = $repository->findAll();
//
        $repository = $em
            ->getRepository('mftrackBundle:mdata');

        $mdata = $repository->findAll();
```

A pure OOP way of getting the data. The result of doctrine query is an OBJECT not ARRAY.

```
        $query = $repository->createQueryBuilder('tdata')
         ->where('tdata.mfcode = :mfcode')
         ->andWhere('tdata.transaction = 1')
         ->setParameters(array('mfcode'=> $mutualfund->getMfcode()))
         ->orderBy('tdata.date', 'ASC')
         ->getQuery();
        $tdata = $query->getResult();
```

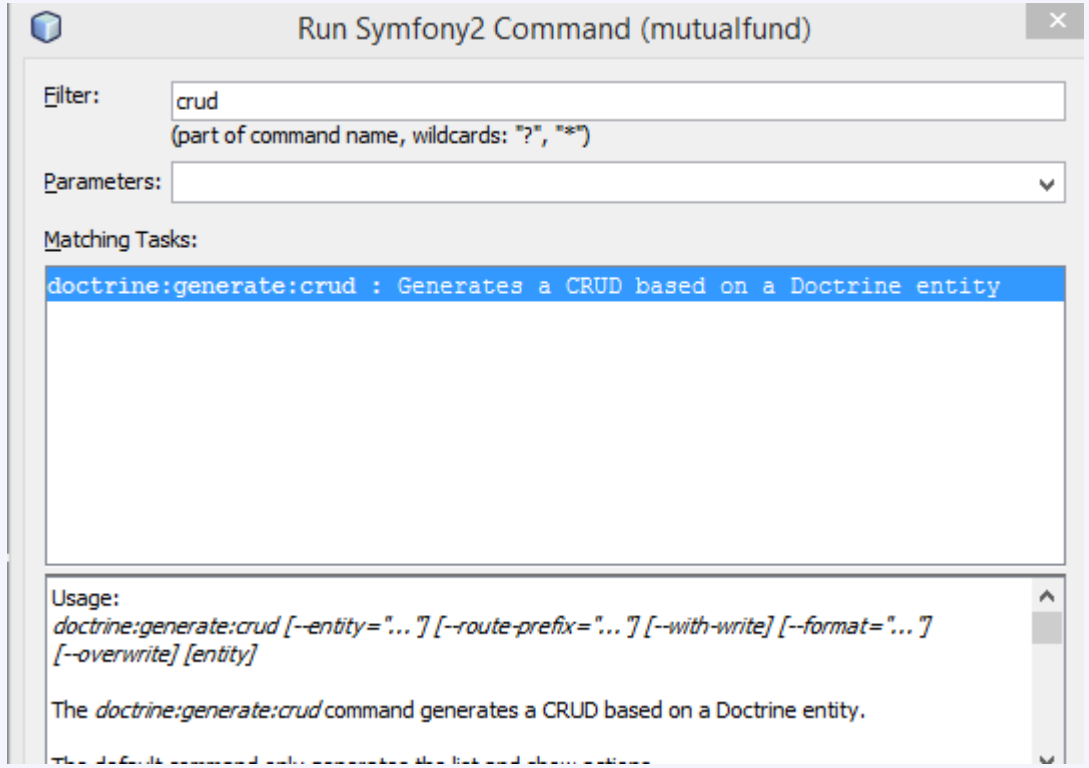
SYMFONY

CRUD Generator

Most of the business objects/ tables have Create , Read , Update ,Delete operations.

We can use the Sympony's inbuilt CRUD generator to reduce our significant effort. It
creates appropriate controllers, form controllers, views etc,.

doctrine:generate:crud



SYMFONY

Twig

A template engine for Sympony.

Main task is to get the data(Array) from controller and render it according to the template.

```
25  /**
26   * @Route("/hello/{name}", name="_demo_hello")
27   * @Template()
28   */
29  public function helloAction($name)
30  {
31      return array('name' => $name);
32  }
33
```

```
1   {% extends "AcmeDemoBundle::layout.html.twig" %}
2
3   {% block title "Hello " ~ name %}
4
5   {% block content %}
6       <h1>Hello {{ name }}!</h1>
7   {% endblock %}
8
9   {% set code = code(_self) %}
10
```

SYMFONY

Twig Features

Supports Inheritance : We can override or extend a base template.

```
{% extends "AcmeDemoBundle::layout.html.twig" %}
```

Supports simple programming commands like FOR,IF etc

```
{% for fund in funds %}
    <tr>

        <td><a href="{{ path('fundbyname', { 'mfcode': fund.mfcode }) }}">{{ fund.mfname }}</td>
        <td>{{ fund.units }}</td>
        <td>{{ fund.avg_nav }}</td>
        <td>{{ fund.total_rprofit }}</td>
        <td>{{ fund.total_urprofit }}</td>
    </tr>
{% endfor %}
```

Can create URL links by calling routes.

SYMFONY

Twig Features

Advanced feature: Supports Asset

Can be used to club the JS/CSS or external files into a minfied versions.

```
27  {% javascripts '@mftrackBundle/Resources/public/js/*' %}
28      <script type="text/javascript" src="{{ asset_url }}"></script>
29  {% endjavascripts %}
30
31  {% stylesheets '@mftrackBundle/Resources/public/css/*' %}
32      <link rel="stylesheet" href="{{ asset_url }}" />
33  {% endstylesheets %}
```

It can also run filters / optimizer before minifiing the versions

```
{% block stylesheets %}
    {% stylesheets 'bundles/app/css/*' filter='cssrewrite' %}
        <link rel="stylesheet" href="{{ asset_url }}" />
    {% endstylesheets %}
{% endblock %}
```

More info: http://symfony.com/doc/current/cookbook/assetic/asset_management.html

Quiz

I am the one who assigns the work to the proper controller and method depending upon the URL parameters. Who Am I ?

Quiz

I am a Persistent layer and also act like ORM / Database Abstraction. People sometimes call me as Model ;) Who Am I ?

Quiz

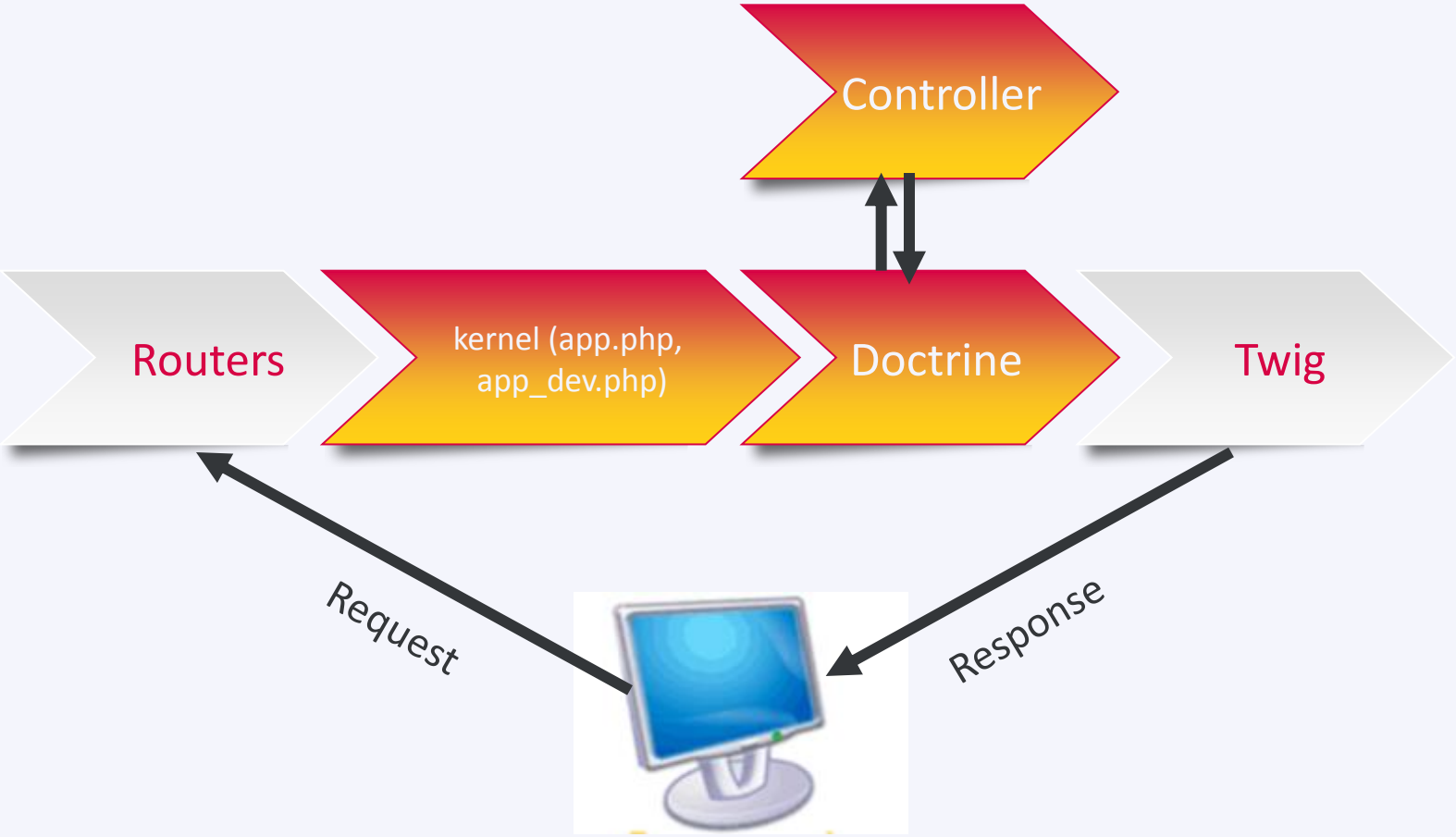
By using _____ in Twig, we can minify or run the optimizer for static files like JS/CSS etc

Quiz

I am an installer which can be used to download and install many popular PHP libraries. Some call me Google Play for PHP. Who Am I ?

Quiz

Something is wrong here



Quiz

Can you find similarities between Symfony Framework and Spring Framework?

Resources

Symfony Getting Started Book:

http://symfony.com/doc/current/book/index.html

Symfony Best Practices:

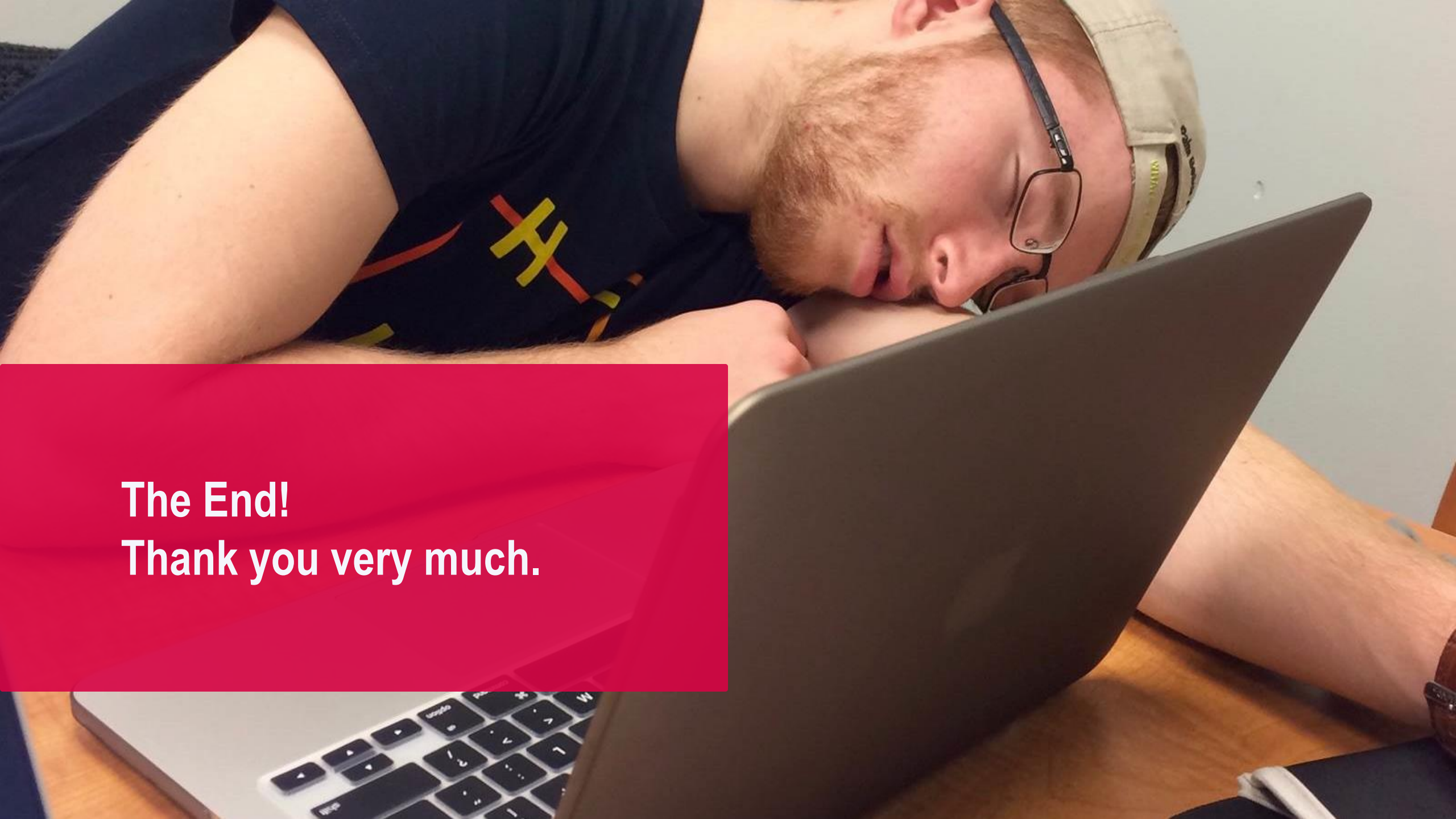
http://symfony.com/doc/current/best_practices/index.html

Screencasts:

knpuniversity.com

Jobeet Tutorial with Symfony2:

www.ens.ro/2012/03/21/jobeet-tutorial-with-symfony2/



The End!
Thank you very much.