

The Symfony Framework

YOUR FREE NEW TOOLKIT



KNPUniversity



Hallo!

- > Lead contributor to the Symfony documentation
- > KnpLabs US - Symfony consulting, training & kumbaya
- > Writer for KnpUniversity.com
awesome amazing PHP Tutorials!!!
- > Husband of the much more talented @leannapelham



KNPUniversity

knplabs.com
twitter.com/weaverryan

 **Knp** Labs

Act 1

Dancing on your own?

Drupal 7

```
/** Implements hook_menu() */  
function dinosaur_menu() {  
    $items['hello'] = array(  
        'title' => 'R0000000AR!',  
        'page callback' => 'favorite_dinosaur',  
    );  
  
    return $items;  
}  
  
function favorite_dinosaur() {  
    return 'Triceratops';  
}
```



Symfony, Silex, etc

routes & controllers

requests & responses

service container

And now

Symfony, Silex, D8

SILEX

The PHP micro-framework
based on the Symfony2 Components

a SENSIO LABS * product



```
require_once __DIR__ . '/vendor/autoload.php';

$app = new Silex\Application();

$app->get('/hello/{name}', function($name) use($app) {
    return 'Hello ' . $app->escape($name);
});

$app->run();
```

@weaverryan

SILEX

The PHP micro-framework
based on the Symfony2 Components

a SENSIO LABS * product



```
require_once __DIR__ . '/vendor/autoload.php';

$app = new Silex\Application();

$app->get('/hello/{name}', function($name) use($app) {
    return 'Hello ' . $app->escape($name);
});

$app->run();
```

An entire application
that says hallo!

Configure your web server

NGINX

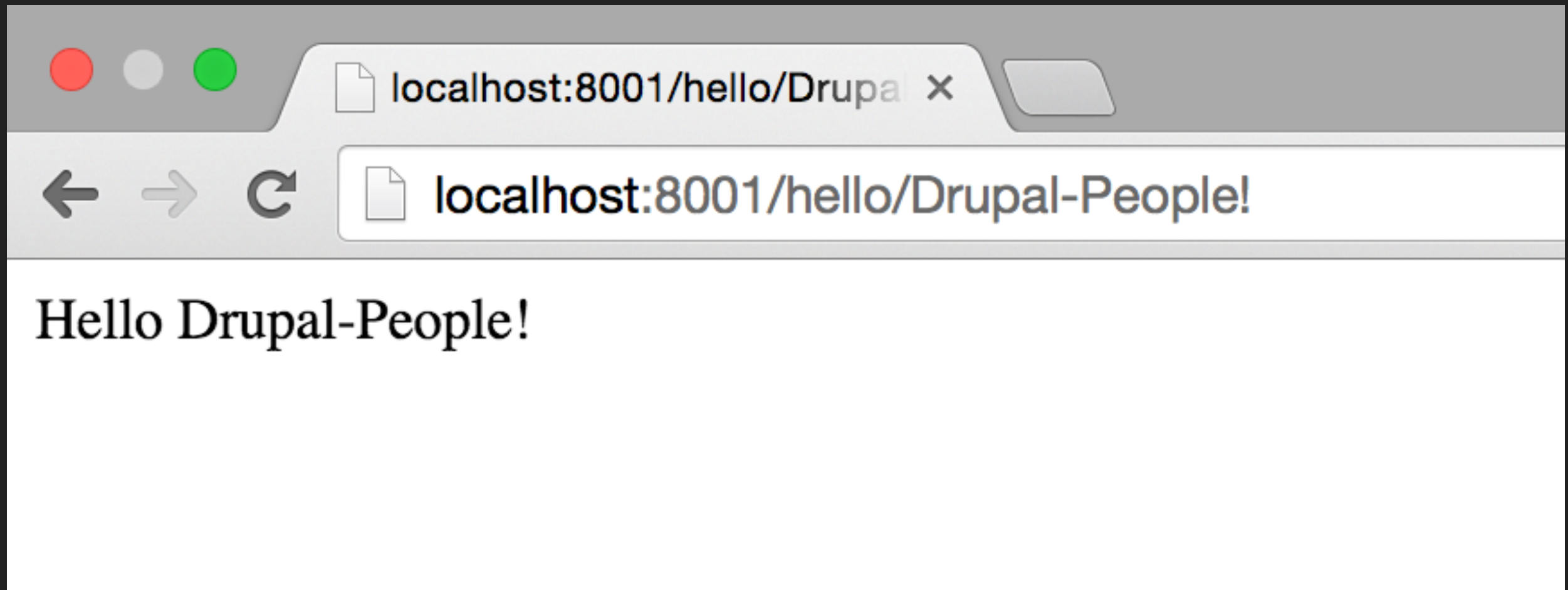


APACHE
HTTP SERVER

Or use the built-in PHP
web server `\o/`

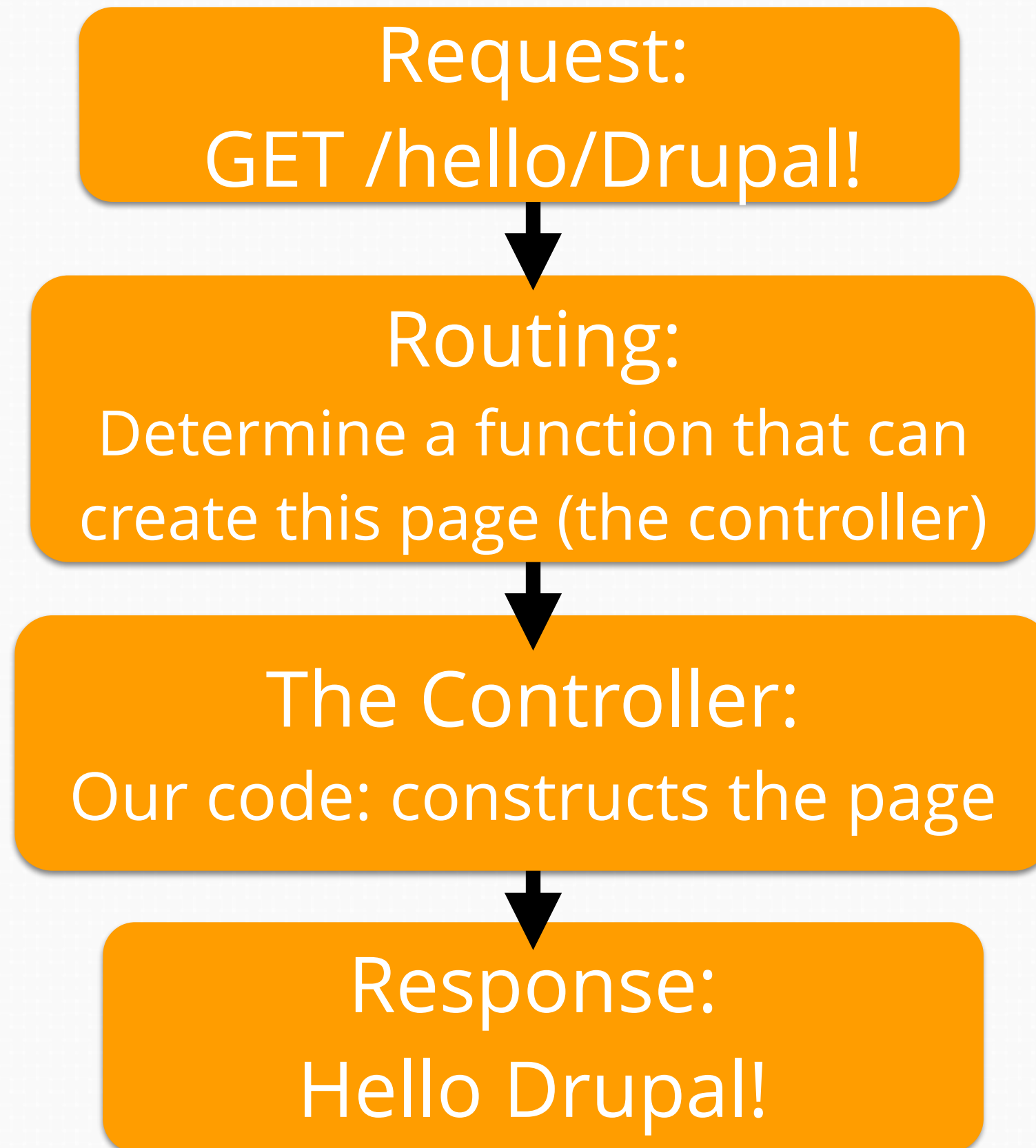
```
php -S localhost:8000
```

```
~/Sites/os/drupalcon/silex$ php -S localhost:8001  
PHP 5.5.22 Development Server started at Sun May 10 12:06:27 2015  
Listening on http://localhost:8001  
Document root is /Users/weaverryan/Sites/os/drupalcon/silex  
Press Ctrl-C to quit.
```



*** The built-in PHP web server can
be used with Drupal too!**

Request -> Response Framework



SILEX

The PHP micro-framework
based on the Symfony2 Components

a SENSIO LABS  product



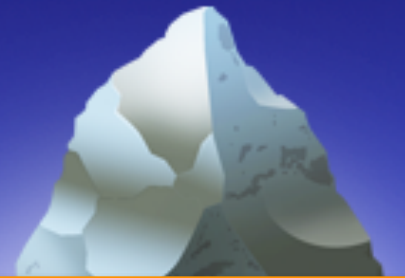
```
$app = new Silex\Application();  
$app->get('/hello/{name}', function($name) use($app) {  
    return 'Hello ' . $app->escape($name);  
});  
$app->run();
```

The route is matched
when the URI is
`/hello/*`

SILEX

The PHP micro-framework
based on the Symfony2 Components

a SENSIO LABS  product



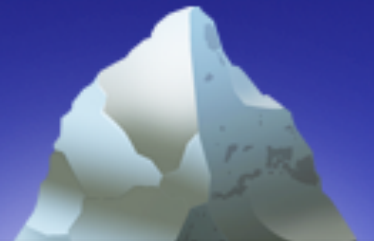
If the URI matches the route,
Silex executes this
function (the controller)

```
$app = new Silex\Application();  
  
$app->get('/hello/{name}', function($name) use($app) {  
    return 'Hello ' . $app->escape($name);  
});  
  
$app->run();
```

SILEX

The PHP micro-framework
based on the Symfony2 Components

a SENSIO LABS  product



The value of {name} is
passed as an argument
to the controller

```
$app = new Silex\Application();  
  
$app->get('/hello/{name}', function($name) use($app) {  
    return 'Hello ' . $app->escape($name);  
});  
  
$app->run();
```


SILEX

The PHP micro-framework
based on the Symfony2 Components

a SENSIO LABS  product



We construct the page
and celebrate!

```
$app = new Silex\Application;
```

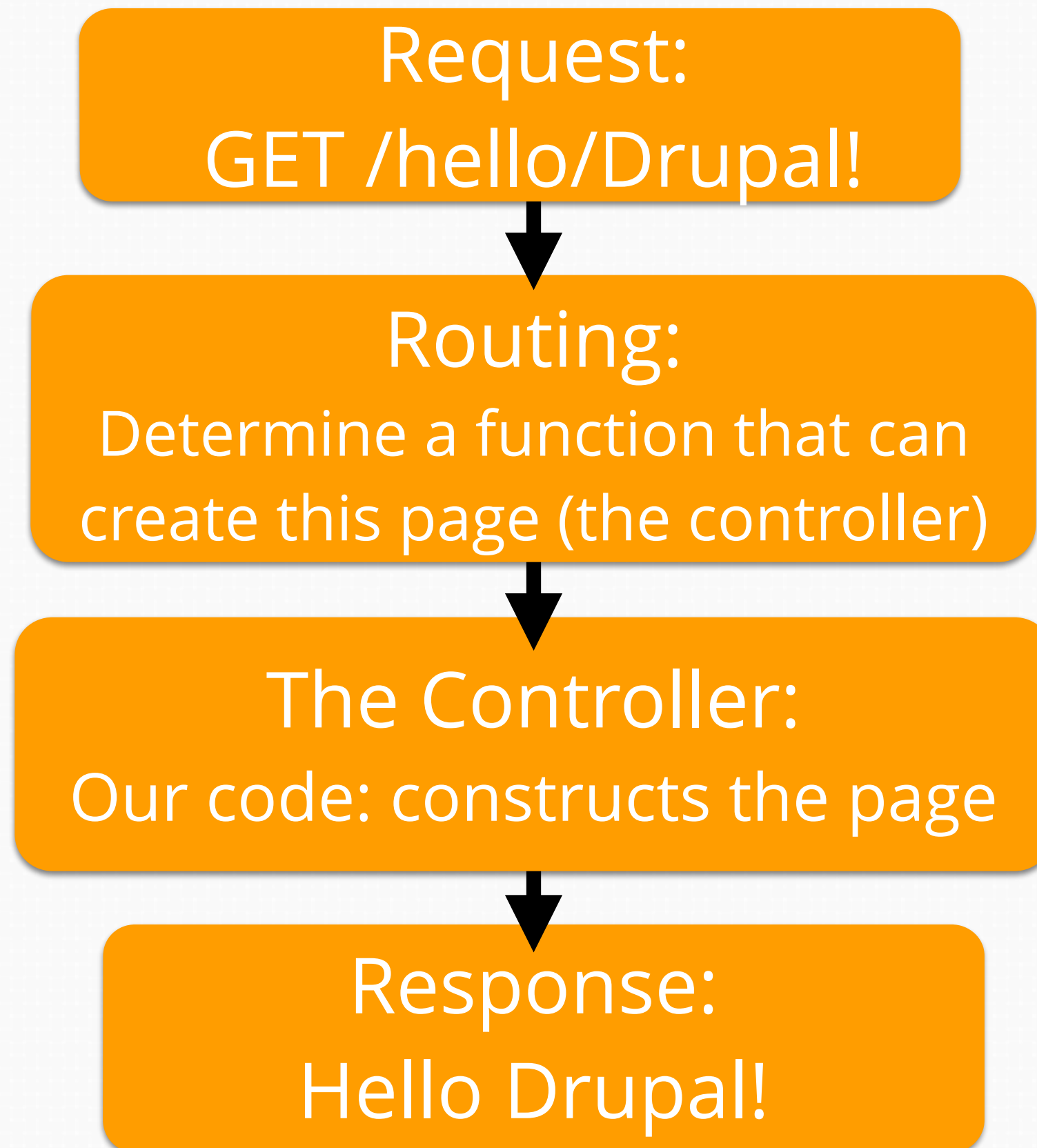
```
$app->get('/hello/{name}', function($name) {  
    return 'Hello ' . $app->escape($name);  
});
```

```
$app->run();
```



(or non-alcoholic beverage of your choice)

Request -> Response Framework



Act 2

Hello Symfony

Use **Symfony 2.6**
for the latest features

end of support
July 2015

Use **Symfony 2.3**
for long-term support

end of support
May 2016

✔ First, [install the Symfony Installer](#). Then, execute this command:

The **Symfony Installer** is a small PHP application that must be installed once in your computer. It greatly simplifies the creation of new projects based on the Symfony framework.

downloads the installer

Installation on Linux and Mac OS X

```
$ sudo curl -Ls https://symfony.com/installer -o /usr/local/bin/symfony  
$ sudo chmod a+x /usr/local/bin/symfony
```

Installation on Windows

```
c:\> php -r "readfile('https://symfony.com/installer');" > symfony
```

Move the downloaded file to your projects directory and execute it as **php symfony**.

```
~/Sites/os/drupalcon$ symfony new my_dir_name
```

Downloading Symfony...

4.91 MB/4.91 MB

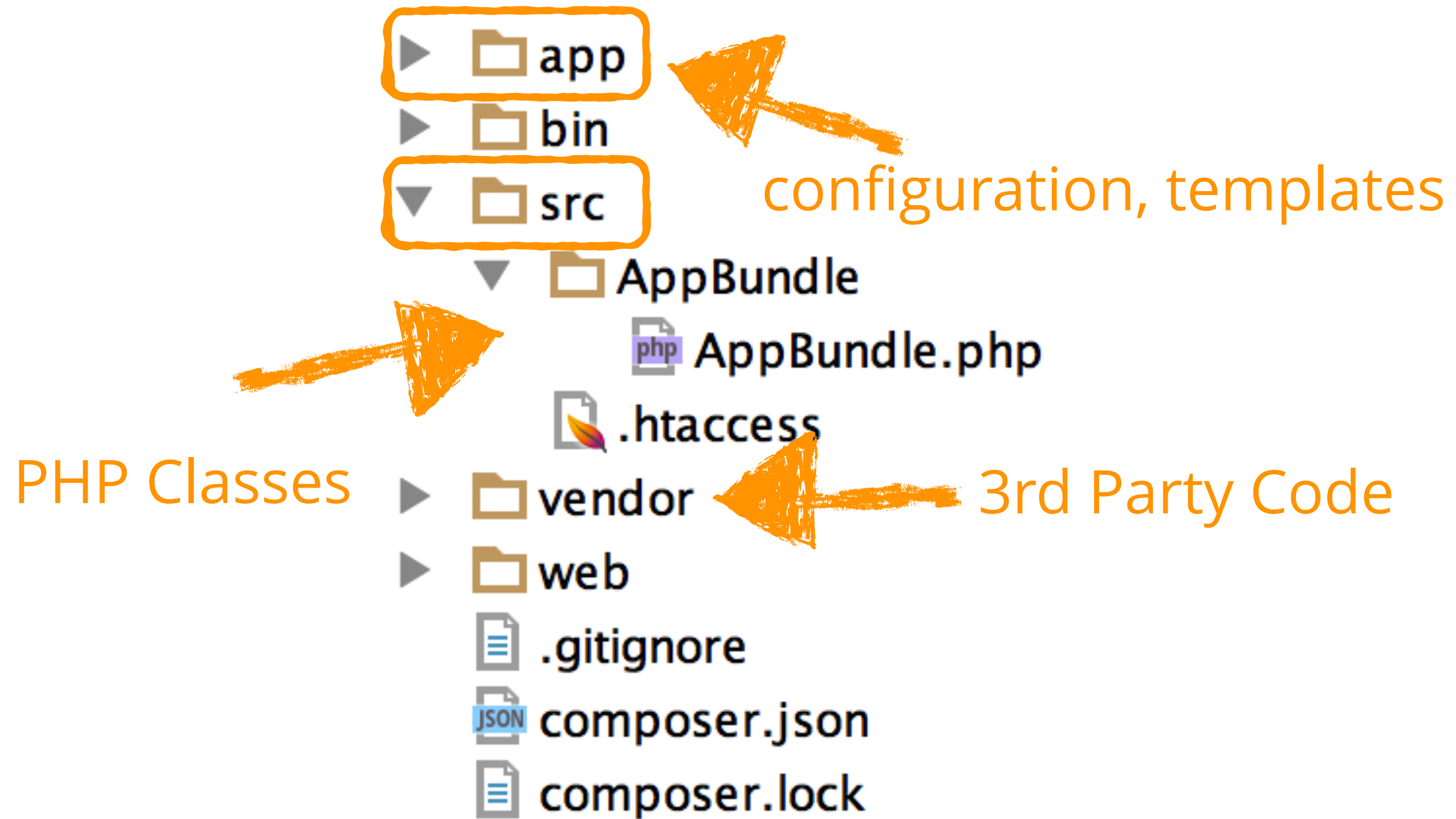
100%

Preparing project...

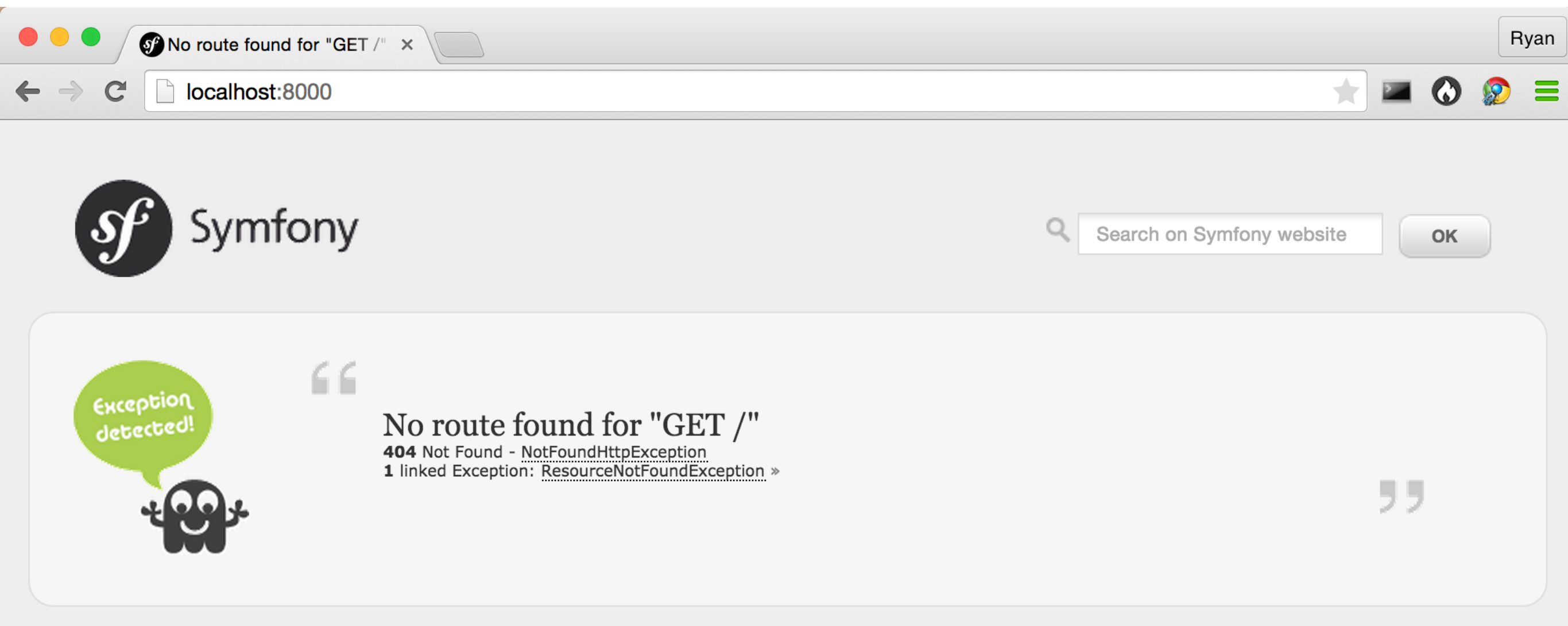
✓ Symfony 2.6.6 was **successfully installed**. Now you can:

- * Change your current directory to **/Users/weaverryan/Sites/os/drupalcon/symfony**
- * Configure your application in **app/config/parameters.yml** file.
- * Run your application:
 1. Execute the **php app/console server:run** command.
 2. Browse to the **http://localhost:8000** URL.
- * Read the documentation at **http://symfony.com/doc**

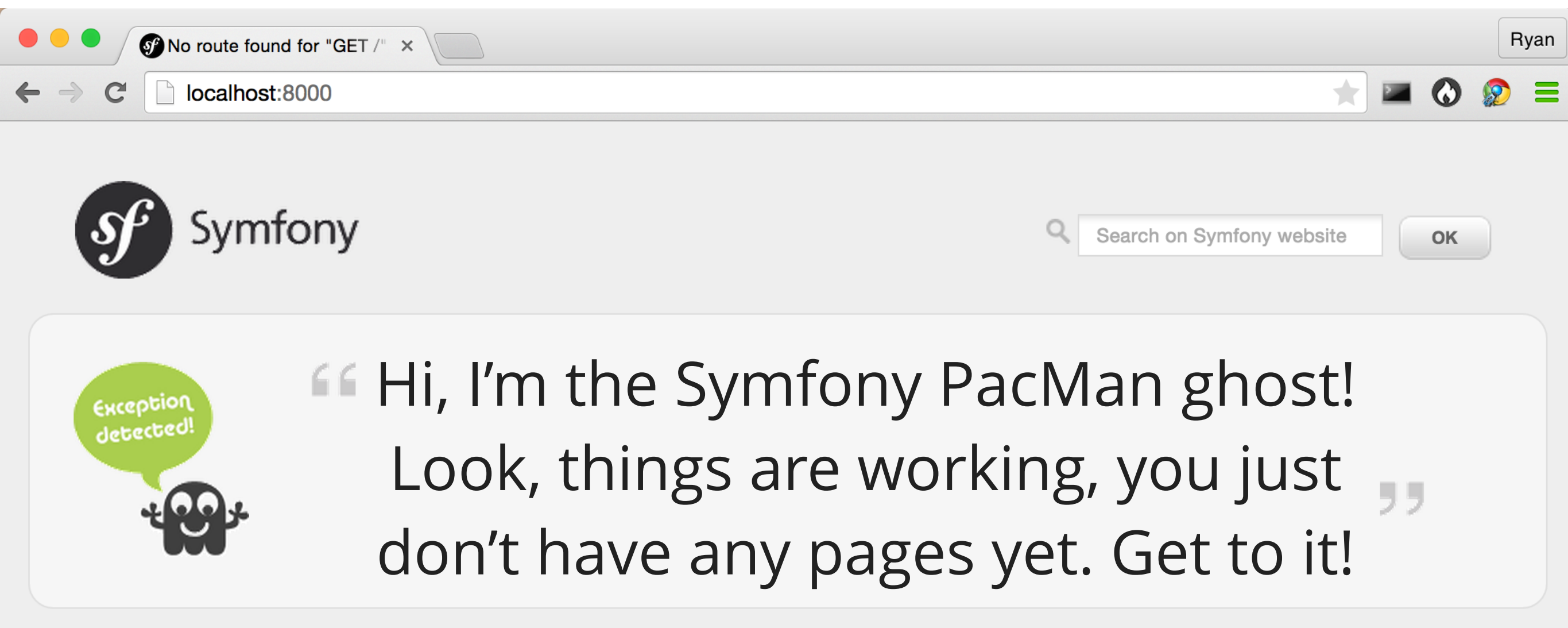
Symfony Project Structure



```
~/Sites/os/drupalcon/symfony$ ./app/console server:start  
Web server listening on http://127.0.0.1:8000  
~/Sites/os/drupalcon/symfony$ █
```

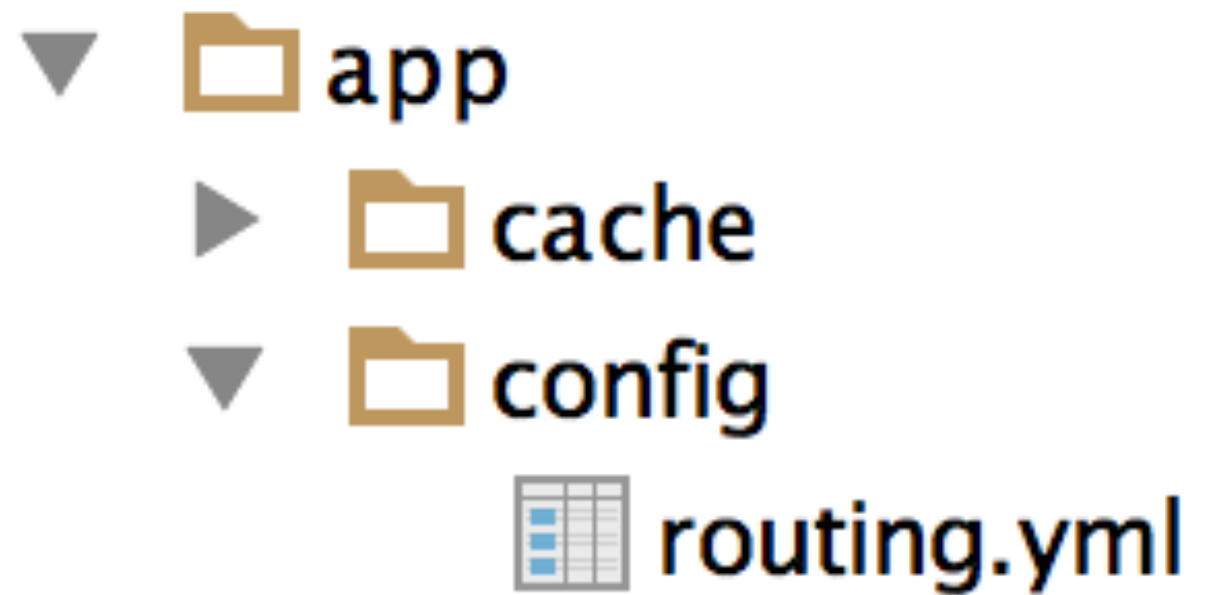
@weaverryan



@weaverryan

Install ☒

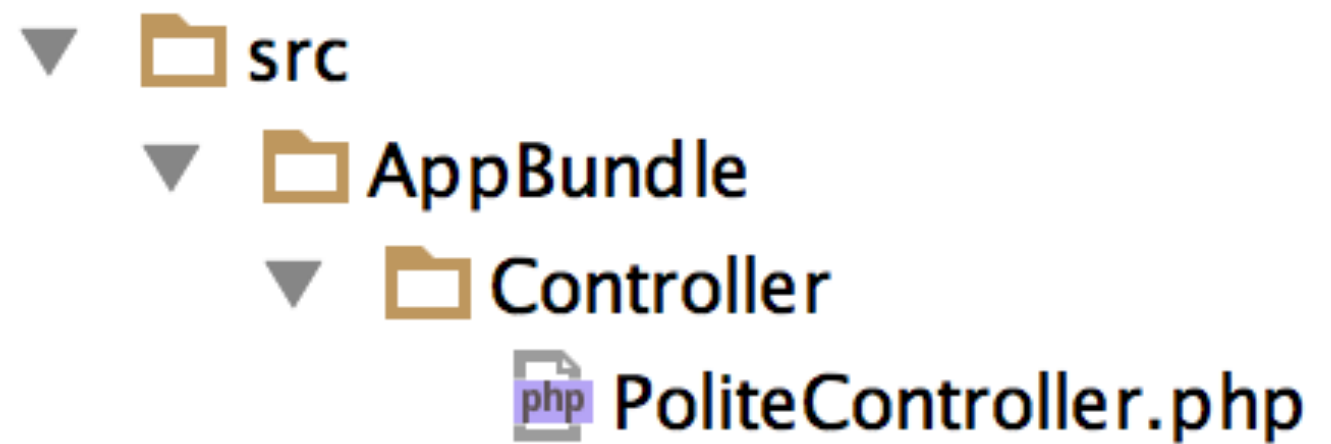
Build a page! ☐



```
hello_world:  
  path: /hello/{name}  
  defaults:  
    _controller: AppBundle\...sayHelloAction
```



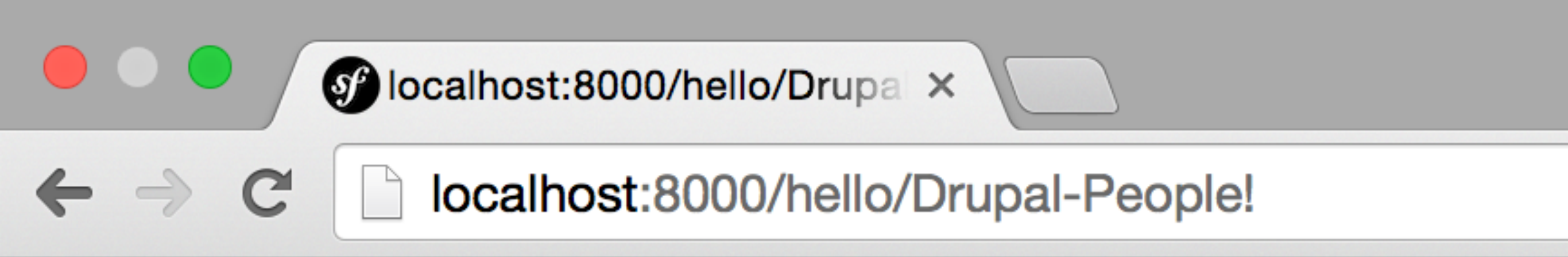
```
AppBundle\Controller\PoliteController::sayHelloAction
```



```
namespace AppBundle\Controller;

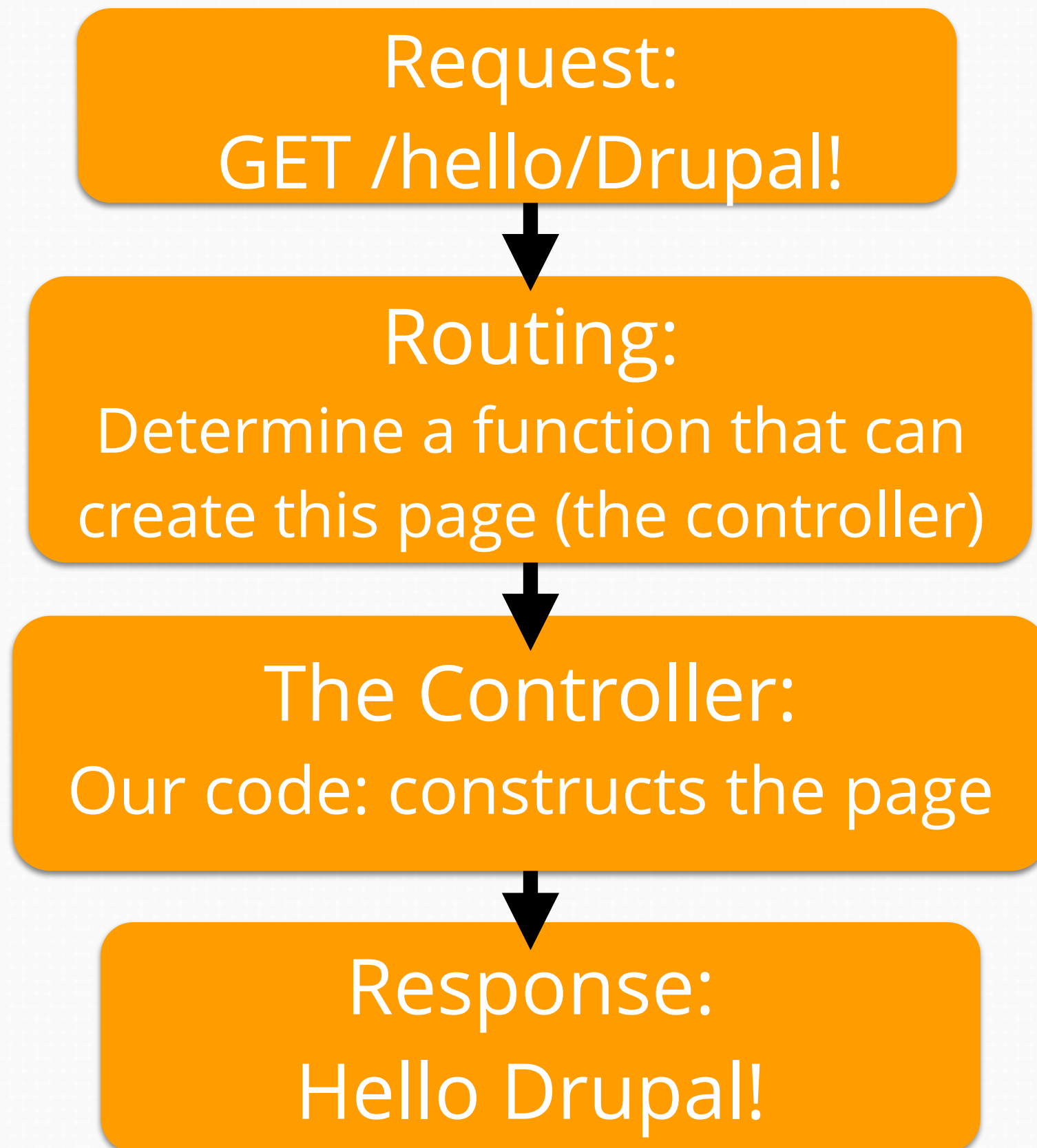
use Symfony\Component\HttpFoundation\Response;

class PoliteController
{
    public function sayHelloAction($name)
    {
        return new Response('Hello ' . $name);
    }
}
```



Hello Drupal-People!

Request -> Response Framework



Debugging?

```
~/Sites/os/drupalcon/symfony$ ./app/console debug:router
```

[router] Current routes

Name	Method	Scheme	Host	Path
_wdt	ANY	ANY	ANY	/_wdt/{token}
_profiler_home	ANY	ANY	ANY	/_profiler/
_profiler_search	ANY	ANY	ANY	/_profiler/search
_profiler_search_bar	ANY	ANY	ANY	/_profiler/search_bar
_profiler_purge	ANY	ANY	ANY	/_profiler/purge
_profiler_info	ANY	ANY	ANY	/_profiler/info/{about}
_profiler_phpinfo	ANY	ANY	ANY	/_profiler/phpinfo
_profiler_search_results	ANY	ANY	ANY	/_profiler/{token}/search
_profiler	ANY	ANY	ANY	/_profiler/{token}
_profiler_router	ANY	ANY	ANY	/_profiler/{token}/router
_profiler_exception	ANY	ANY	ANY	/_profiler/{token}/exception
_profiler_exception_css	ANY	ANY	ANY	/_profiler/{token}/exception.css
_twig_error_test	ANY	ANY	ANY	/_error/{code}.{_format}
hello_world	ANY	ANY	ANY	/hello/{name}

```
~/Sites/os/drupalcon/symfony$
```



@weaverryan

Welcome!

x

←

→

↻

localhost:8000/hello/Drupal-People!

Yo look! It's Drupal-People!!

Status	200 OK
Controller	PoliteController ::sayHelloAction
Route name	hello_world
Has session	no

2.6.6

php

200

0

355 ms

13.0 MB

0

0

sf Welcome! x

localhost:8000/hello/Drupal-People!

Yo look! It's Drupal-People!!

DB Queries	0
Query time	0.00 ms
Invalid entities	0

sf 2.6.6 php profiler/56e145?panel=db

0 355 ms 13.0 MB 0



CONFIG



REQUEST



EXCEPTION



EVENTS



LOGS



TIMELINE



ROUTING



FORMS



SECURITY



E-MAILS

0



DOCTRINE

0

0 MS



DUMP()

0

View last 10

Profile for: GET <http://localhost:8000/hello/Drupal-People!> by 127.0.0.1 at Sun, 10 May 20

Logs

Filter

Min. Priority

INFO (2)



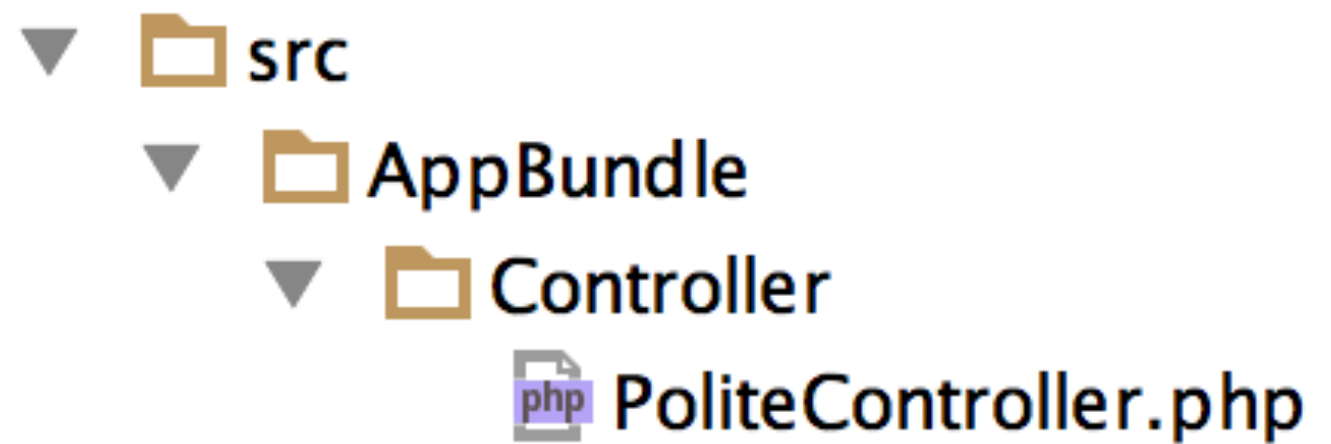
INFO - Matched route "hello_world" (parameters: "_controller": "AppBundle\Controller\PoliteC
People!", "_route": "hello_world")

INFO - Created greeting: Yo look! It's %s!

default section event_listener event_listener_loading template doctrine propel child_sections



Can we do even less
work?



```
// ...
```

```
class PoliteController
{
    /**
     * @Route("/hello/{name}", name="hello_world")
     */
    public function sayHelloAction($name)
    {
        return new Response('Hello '.$name);
    }
}
```

Act 3

Services and the “container”

Services == Useful Objects

The container == the object
that contains all the services

In **Silex**, **Symfony** & **Drupal 8**
there is a “container”.

If you have it, you can use any
service (useful object)

In **Symfony** and **Drupal 8**

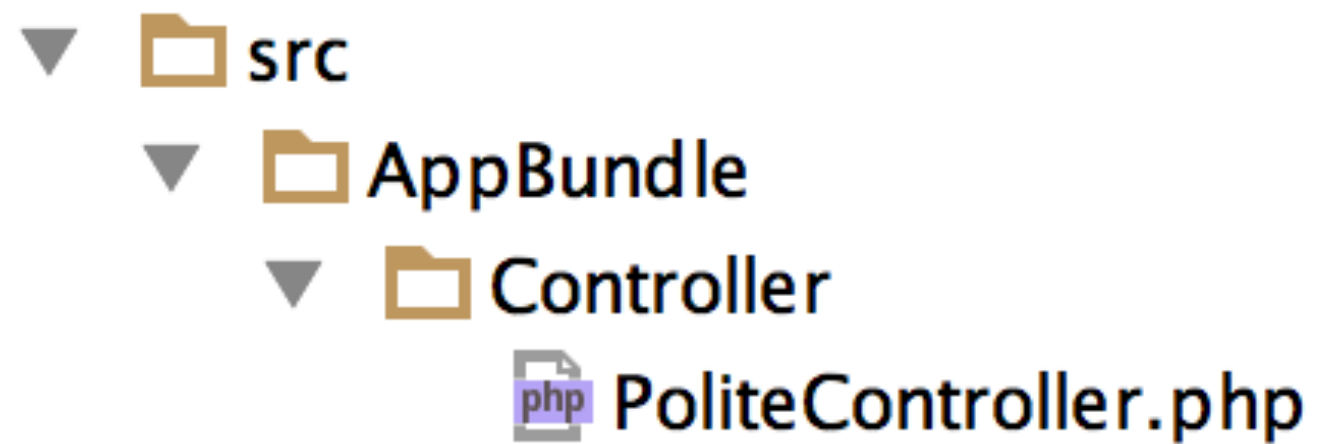
The container is pre-loaded
with many useful services
(objects)

```
~/Sites/drupalcon$ php app/console debug:container | wc -l  
224
```

That's 224 built-in services

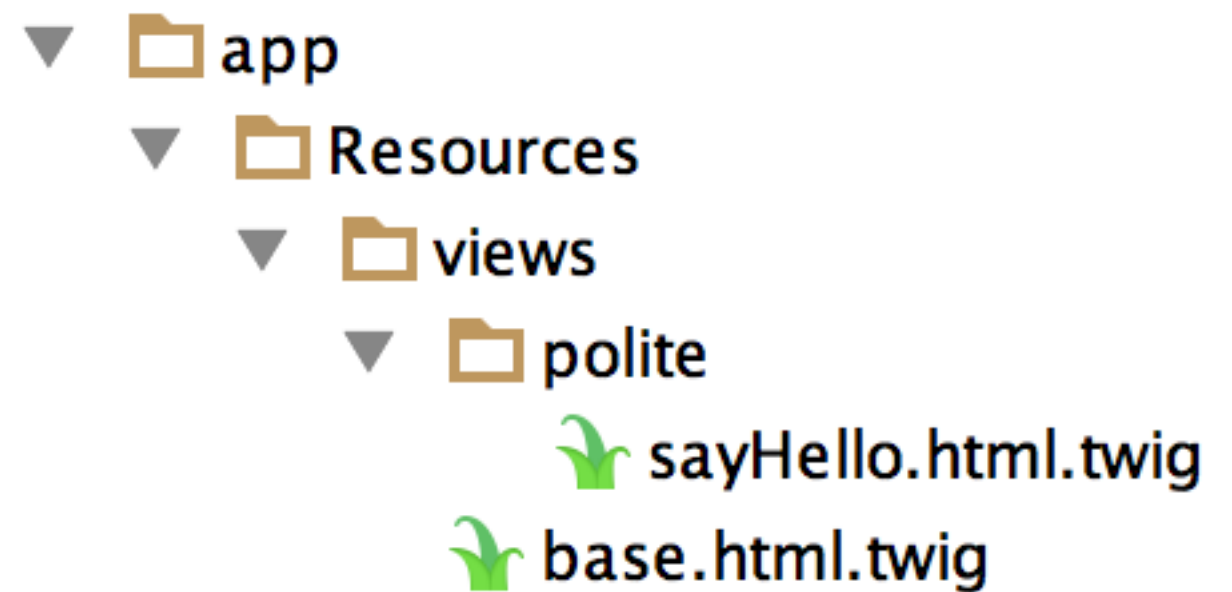

```
~/Sites/drupalcon$ php app/console debug:container templating
[container] Information for service templating
Service Id      templating
Class           Symfony\Bundle\TwigBundle\Debug\TimedTwigEngine
Tags            -
Scope           container
Public          yes
```


How do I get access to the
container inside a controller?



```
/**
 * @Route("/hello/{name}", name="hello_world")
 */
public function sayHelloAction($name)
{
    $html = $this->container->get('templating')->render(
        'polite/sayHello.html.twig',
        ['myName' => $name]
    );

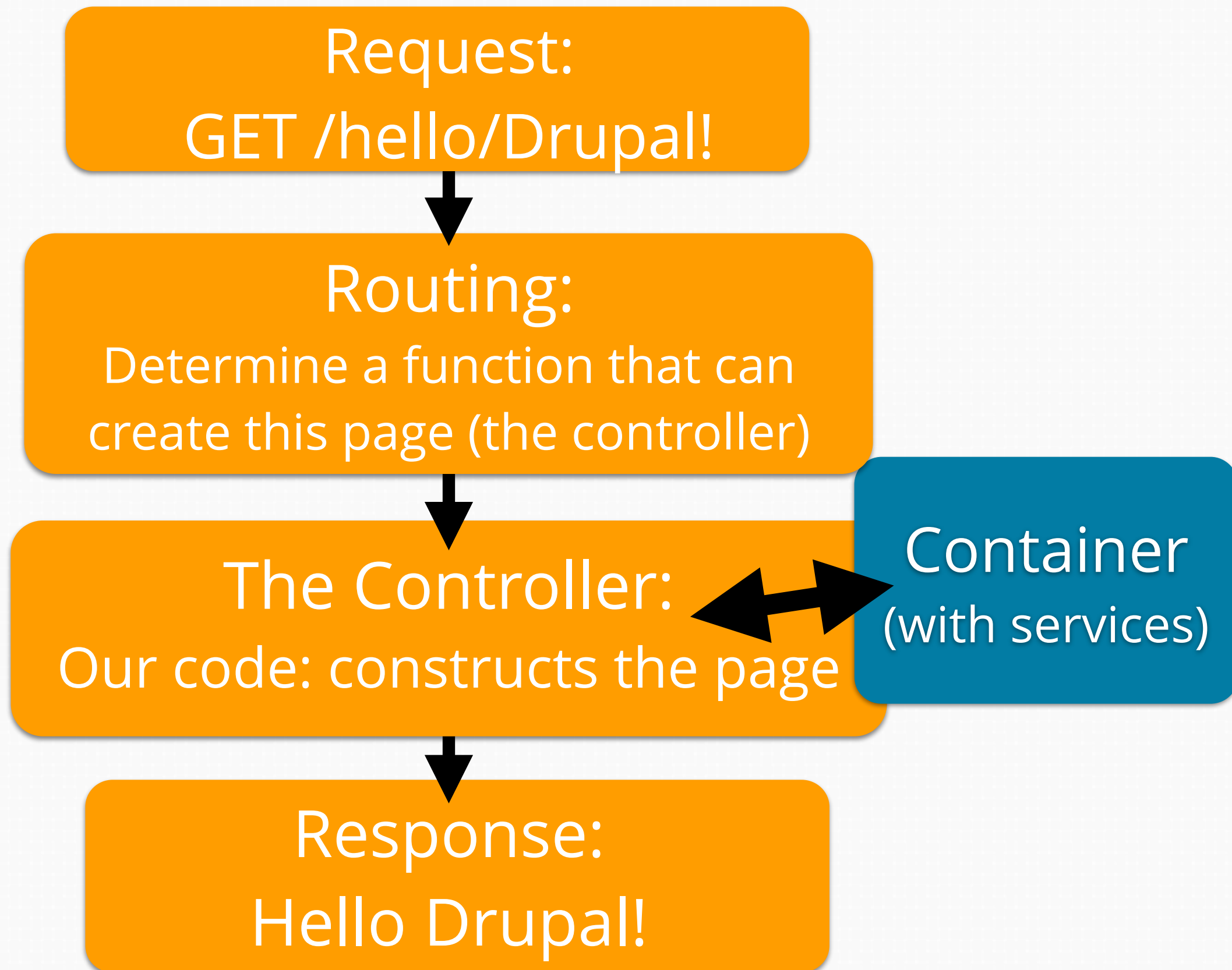
    return new Response($html);
}
```



```
{% extends 'base.html.twig' %}
```

```
{% block body %}  
    Hello {{ myName }}!  
{% endblock %}
```

Request -> Response Framework



What else does
Symfony do?

Doctrine ORM

```
$em = $this->container  
    ->get( 'doctrine.orm.entity_manager' );  
  
$post = $em->getRepository( 'AppBundle:Post' )  
    ->findOneBySlug( $slug );  
  
// ...  
  
$em->persist( $post );  
$em->flush();
```

or just use the DBAL or PDO

```
$conn = $this->container  
    ->get('database_connection');
```

```
$sql = 'SELECT id, name FROM post';  
$posts = $conn->fetchAll($sql);
```


Forms

```
$form = $this->container->get('form.factory')
    ->createBuilder()
    ->add('email', 'email')
    ->add('username', 'text')
    ->add('gender', 'choice', [
        'choices' => ['f' => 'Female', 'm' => 'Male']
    ])
    ->getForm();
```

```
$form->handleRequest($request);
if ($form->isValid()) {
    $data = $form->getData();
    // do some stuff
}
```

```
$html = $this->container->get('templating')->render(
    'user/register.html.twig',
    ['form' => $form->createView()]
);
```

```
return new Response($html);
```


Forms

```
{{ form_start(form) }}  
    {{ form_row(form.email) }}  
    {{ form_row(form.username) }}  
    {{ form_row(form.gender) }}  
  
    <button type="submit">Do it!</button>  
{{ form_end(form) }}
```

or just do it yourself

```
$email = $request->request->get('email');  
$username = $request->request->get('username');  
$gender = $request->request->get('gender');
```

... and infinitely more with
community bundles

Act 4

Creating your own Services

Now we want to
select a random
greeting each time

Put this logic in our
controller?

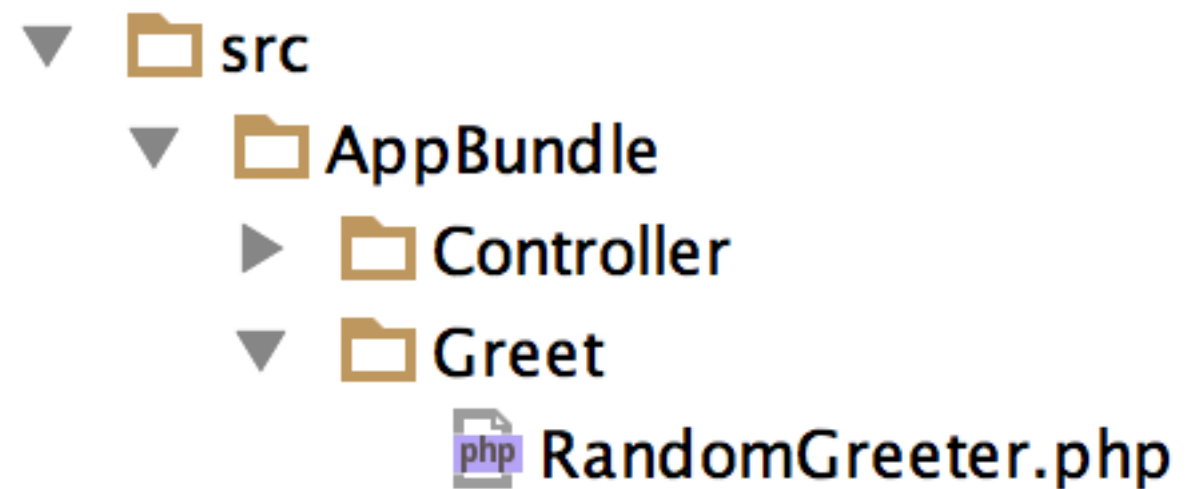
How about a flat
function
somewhere?

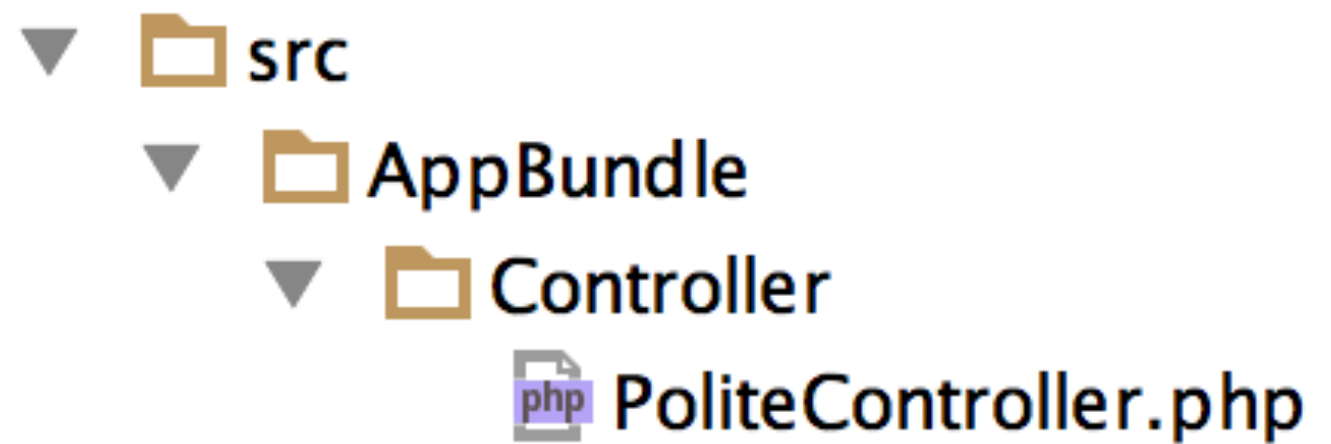
```
namespace AppBundle\Greet;
```

```
class RandomGreeter  
{
```

```
    private static $greetings = [  
        'Hello %s',  
        'Hola %s!',  
        'git blame. Ah, I knew it was %s!'  
    ];
```

```
    public function randomlyGreet($name)  
    {  
        $key = array_rand(self::$greetings);  
        $greeting = self::$greetings[$key];  
  
        return sprintf($greeting, $name);  
    }  
}
```





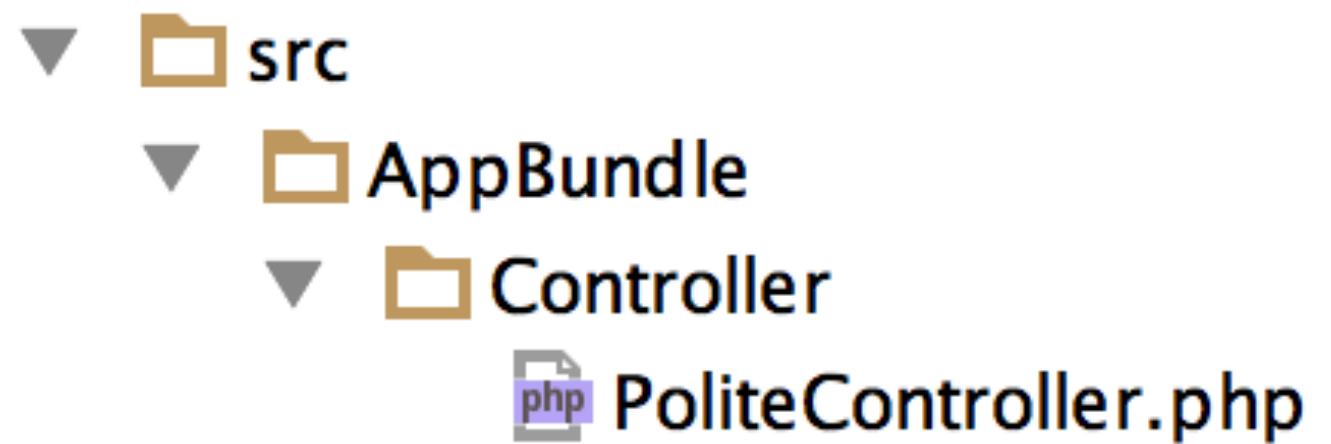
```
public function sayHelloAction($name)
{
    $greeter = new RandomGreeter();
    $greeting = $greeter->randomlyGreet($name);

    $html = $this->container->get('templating')->render(
        'polite/sayHello.html.twig',
        ['theGreeting' => $greeting]
    );

    return new Response($html);
}
```


Could we *log* which
greeting was chosen?

```
~/Sites/drupalcon$ php app/console debug:container logger
[container] Information for service logger
Service Id      logger
Class           Symfony\Bridge\Monolog\Logger
Tags            -
Scope           container
Public          yes
```

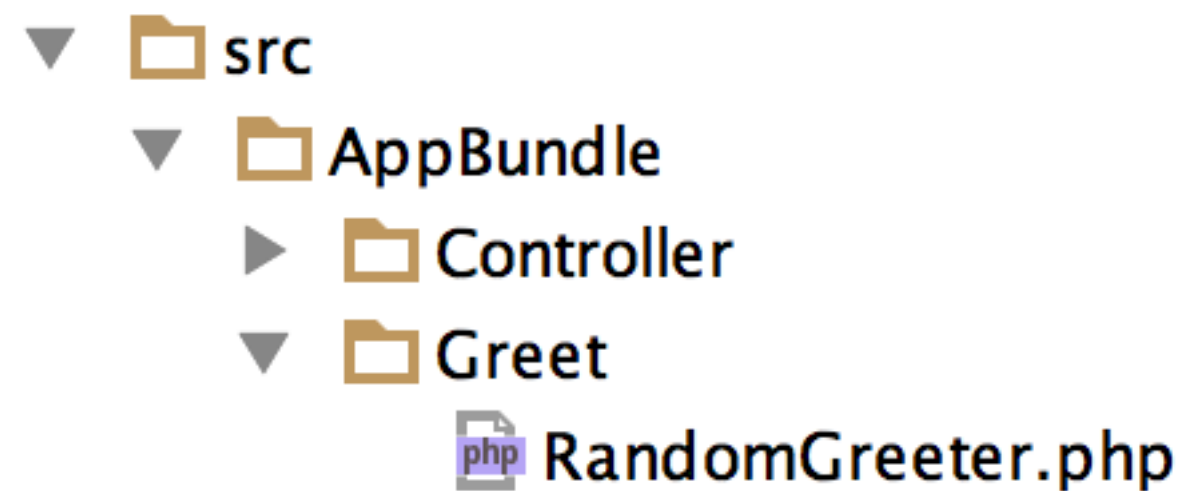


```
public function sayHelloAction($name)
{
    $greeter = new RandomGreeter();
    $greeting = $greeter->randomlyGreet($name);

    $this->container->get('logger')
        ->info('Created greeting: '.$greeting);

    // ...
}
```

Could we log from
inside *RandomGreeter*?



```
class RandomGreeter
{
    public function randomlyGreet($name)
    {
        $key = array_rand(self::$greetings);
        $greeting = self::$greetings[$key];

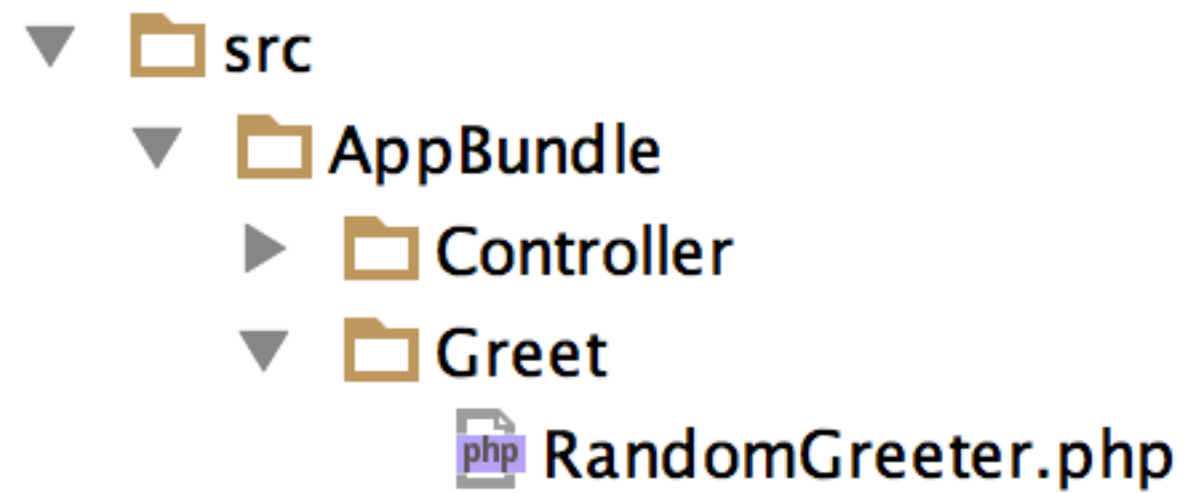
        $this->container->get('logger')
            ->info('Created greeting: '.$greeting);

        return printf($greeting, $name);
    }
}
```

There's no container property! That's magic only the controller has

omg

DEPENDENCY INJECTION



```
class RandomGreeter
{
    private $logger;

    public function __construct($logger)
    {
        $this->logger = $logger;
    }

    // ...
}
```




```
class RandomGreeter
{
    private $logger;

    public function __construct(LoggerInterface $logger)
    {
        $this->logger = $logger;
    }

    // ...
}
```



If you're feeling fancy
and/or awesome

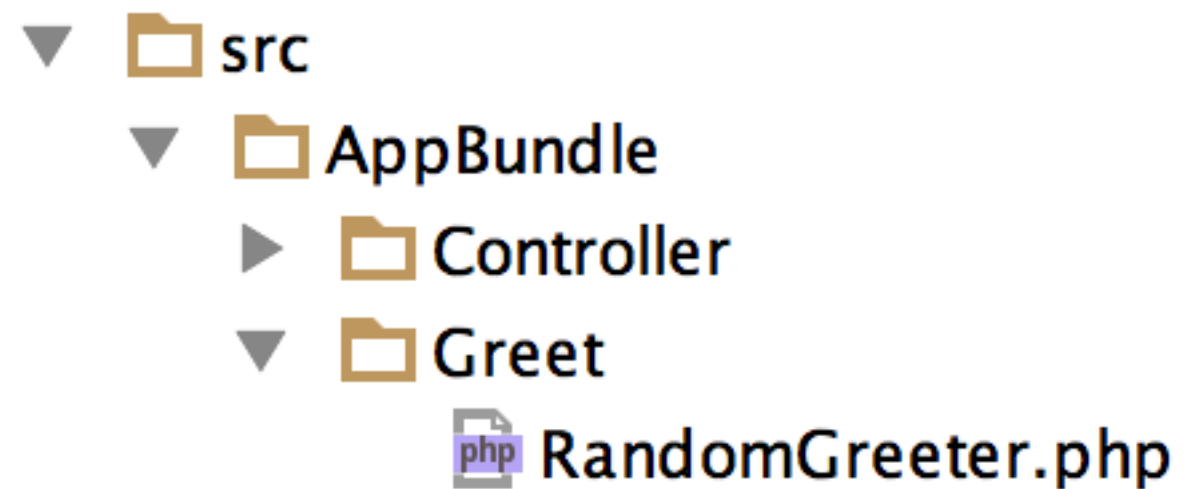

```
class RandomGreeter
{
    private $logger;

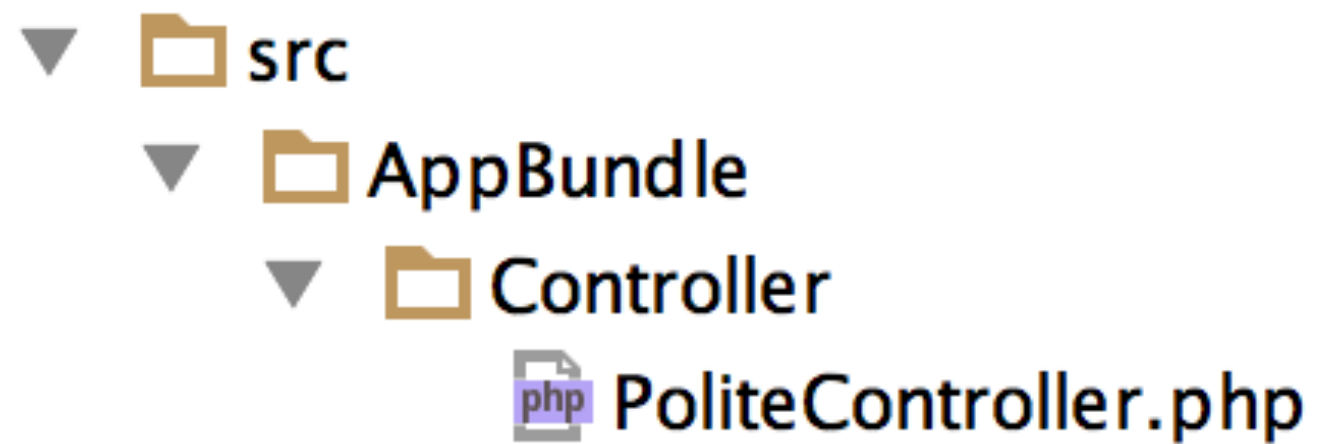
    // ...

    public function randomlyGreet($name)
    {
        $key = array_rand(self::$greetings);
        $greeting = self::$greetings[$key];

        $this->logger
            ->info('Created greeting: '.$greeting);

        return sprintf($greeting, $name);
    }
}
```





```
public function sayHelloAction($name)
{
    $greeter = new RandomGreeter(
        $this->container->get('logger')
    );
    $greeting = $greeter->randomlyGreet($name);

    // ...
}
```

Act 5

Teach Symfony how to
instantiate your services



```
services:
  my_random_greeter:
    class: AppBundle\Greet\RandomGreeter
    arguments:
      - "@logger"
```

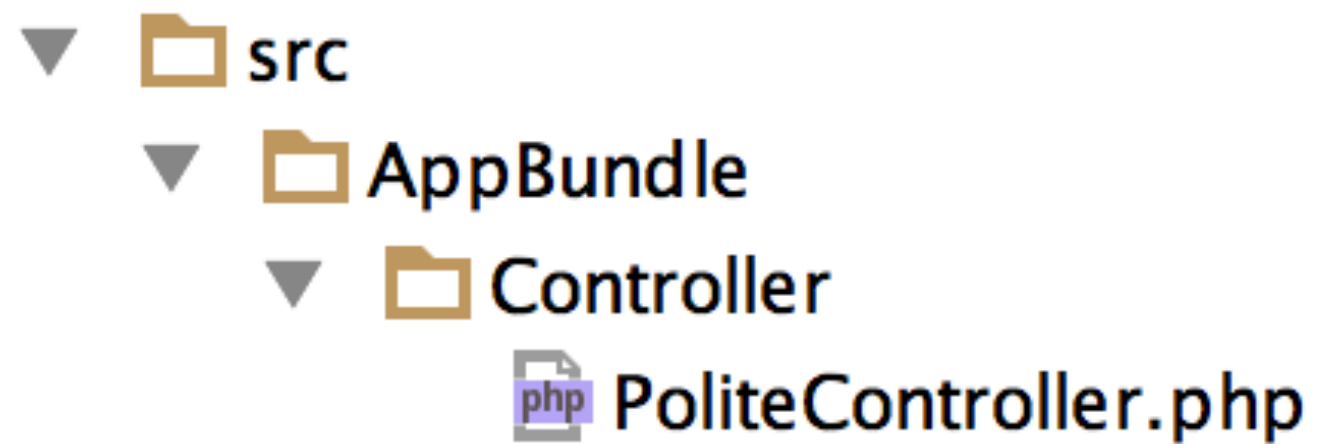


```
services:
  my_random_greeter:
    class: AppBundle\Greet\RandomGreeter
    arguments:
      - "@logger"
```

```
~/Sites/drupalcon$ php app/console debug:container my_random_greeter
```

```
[container] Information for service my_random_greeter
```

Service Id	my_random_greeter
Class	AppBundle\Greet\RandomGreeter
Tags	-
Scope	container
Public	yes



```
public function sayHelloAction($name)
{
    /* $greeter = new RandomGreeter(
        $this->container->get('logger')
    ); */

    $greeter = $this->container
        ->get('my_random_greeter');
    $greeting = $greeter->randomlyGreet($name);

    // ...
}
```

Act 6

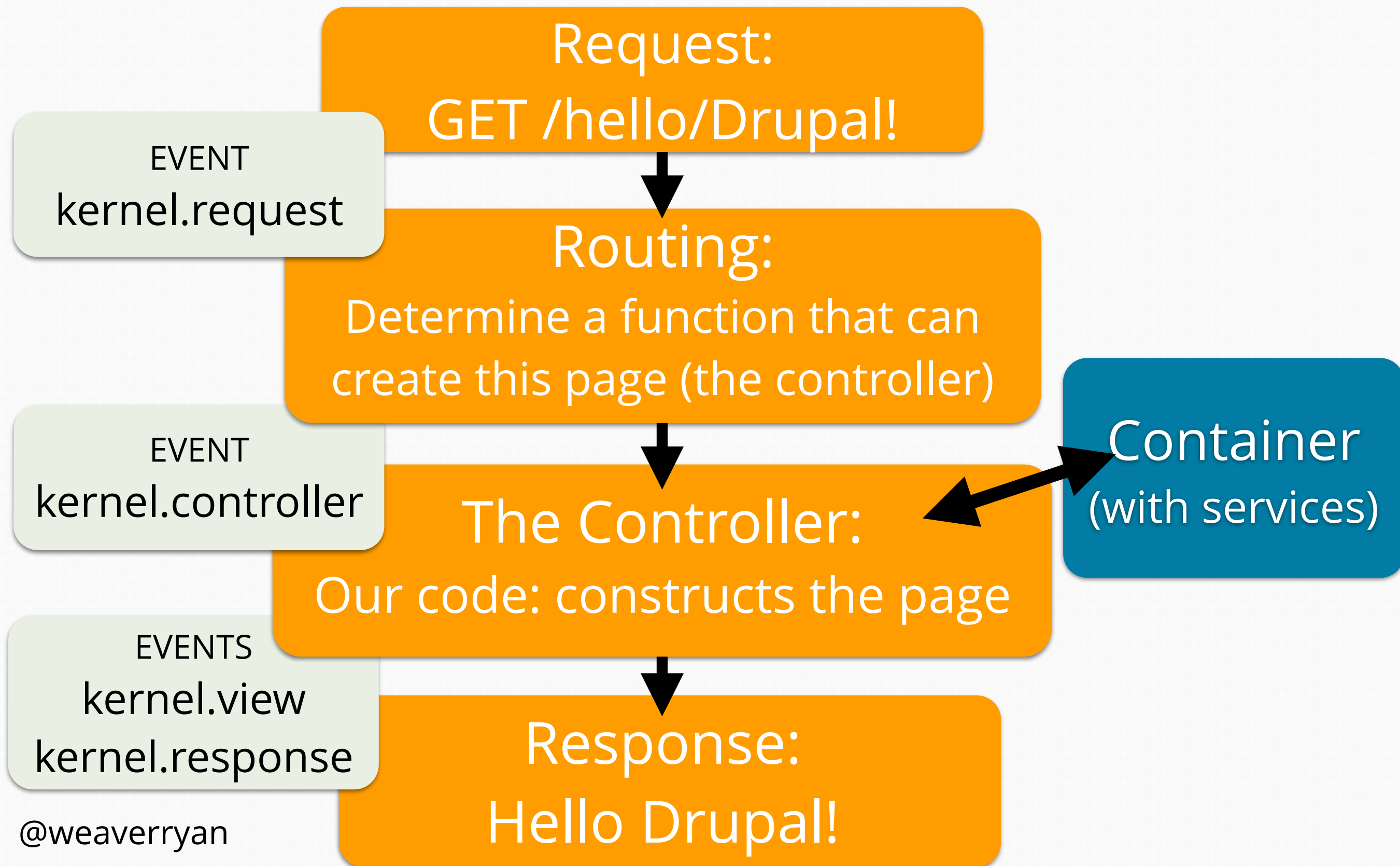
Events (extra credit)

Just like Drupal
“hooks”, Silex has
events

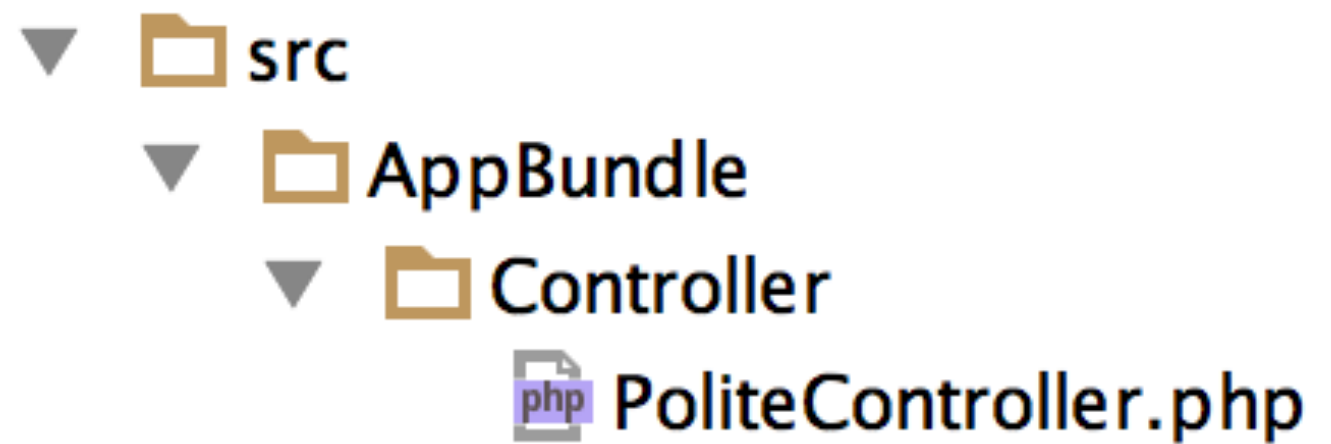
YOU CAN TELL SILEX

“Hi! When event XXXXX happens, execute this function. kthxbai”

Request -> Response Framework



What if we didn't
return a Response
from the controller?



```
public function sayHelloAction($name)
{
    $greeter = $this->container
        ->get('my_random_greeter');
    $greeting = $greeter->randomlyGreet($name);

    return [
        'template' => 'polite/sayHello.html.twig',
        'variables' => ['theGreeting' => $greeting]
    ];
}
```

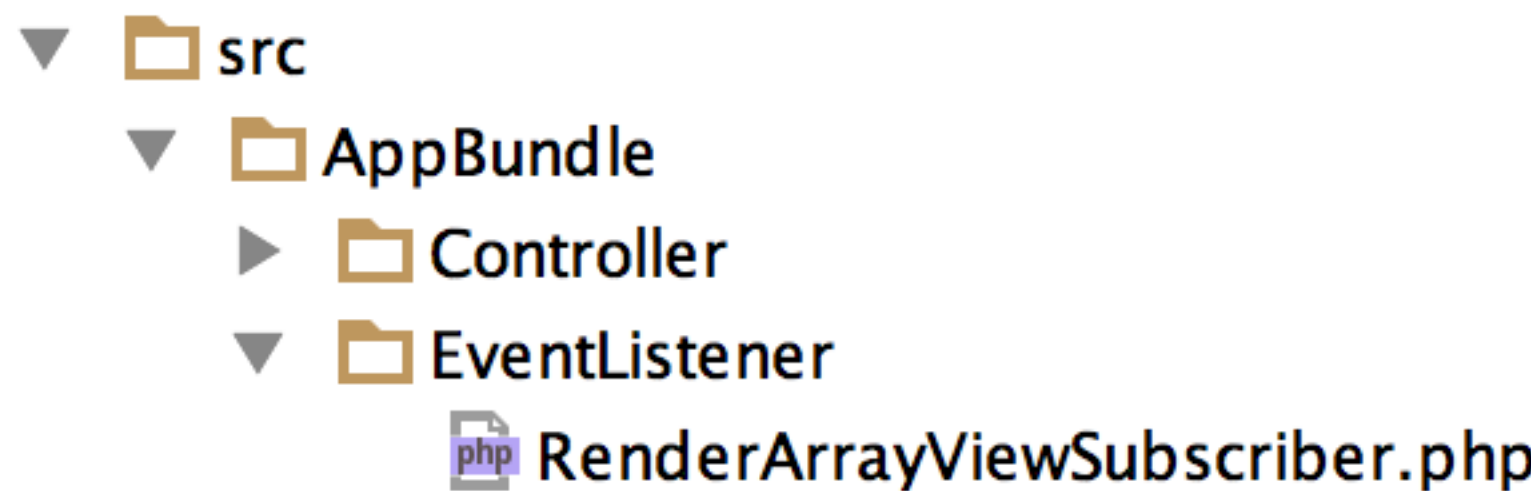


“

I'm so angry right now!!!!

500 Internal Server Error - [LogicException](#)

”



```
class RenderArrayViewSubscriber implements EventSubscriberInterface
{
    private $templating;

    public function __construct(EngineInterface $templating)
    {
        $this->templating = $templating;
    }

    public function onView()
    {
        // call me if the controller does not
        // return a Response
    }

    public static function getSubscribedEvents()
    {
        return ['kernel.view' => 'onView'];
    }
}
```



```
services:
```

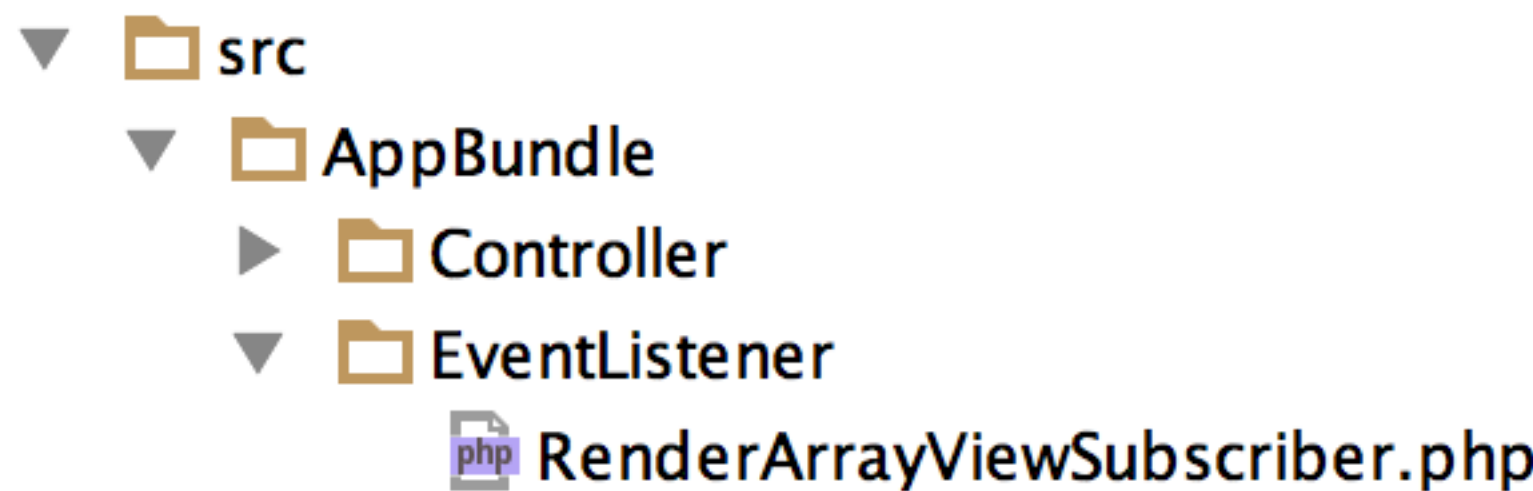
```
  # ...
```

```
  listener.render_array_view_listener:
```

```
    class: AppBundle\EventListener\RenderArrayViewSubscriber
```

```
    arguments:
```

```
      - "@templating"
```

```
public function onView(GetResponseForControllerResultEvent $event)
{
    $controllerResult = $event->getControllerResult();

    if (!is_array($controllerResult)) {
        return;
    }
    if (!isset($controllerResult['template'])) {
        return;
    }

    $template = $controllerResult['template'];
    $variables = $controllerResult['variables'];
    $html = $this->templating->render($template, $variables);

    $response = new Response($html);
    $event->setResponse($response);
}
```


Act 7

Build something amazing

“

It'd be nice to learn by
looking at a real, fully-
feature application

~/Sites symfony demo

Downloading Symfony Demo Application...

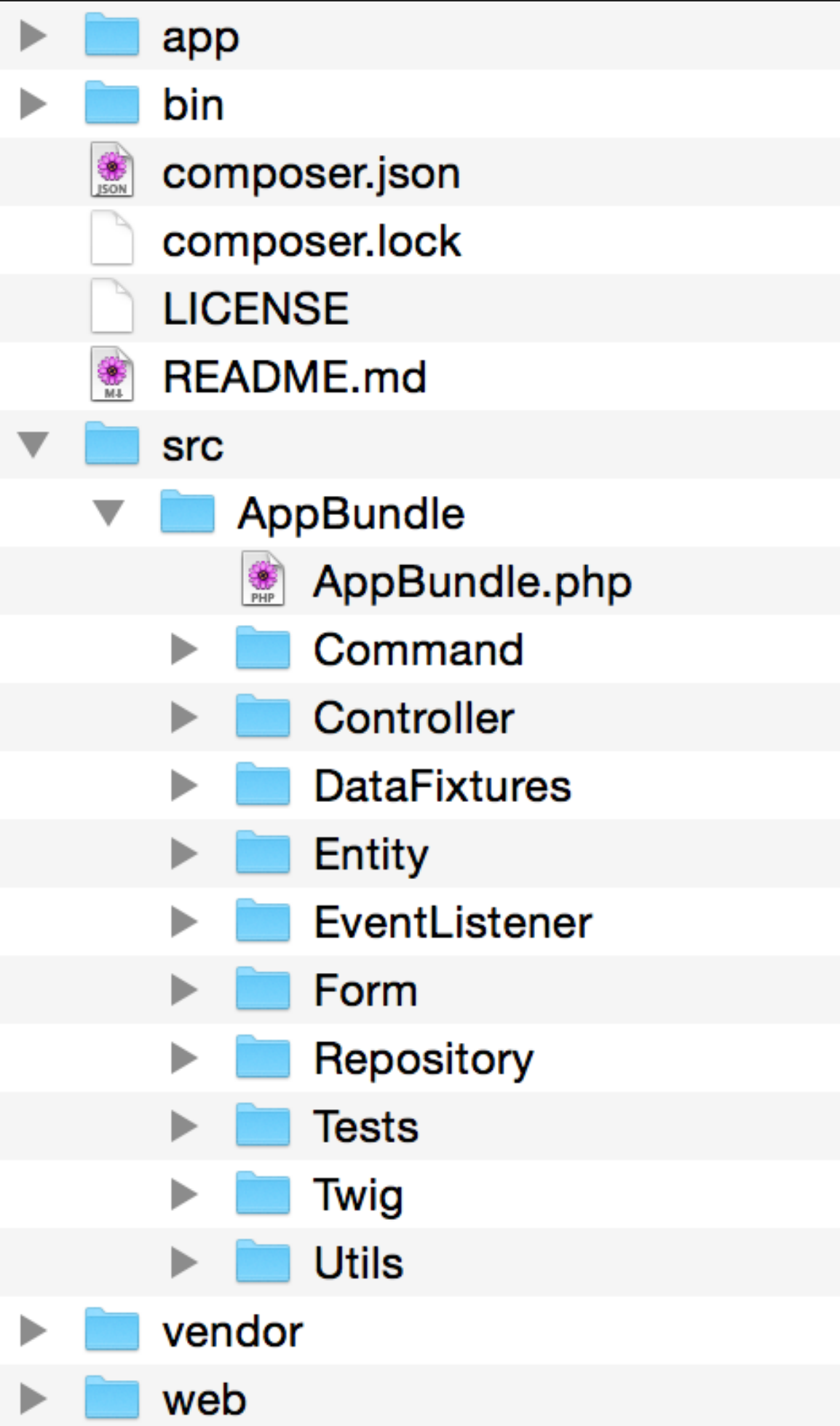
6.58 MB/6.58 MB

Preparing project...

✓ Symfony Demo Application was successfully installed

1. Change your current directory to `/Users/wea`
2. Execute the `php app/console server:run` command
3. Browse to the `http://localhost:8000` URL to

@weaverryan



 Secure Sign in

Username

Password:

 Sign in

← Try either of the following users

Username	Password	Role
john_user	kitten	ROLE_USER (regular user)
anna_admin	kitten	ROLE_ADMIN (administrator)

NOTE If these users don't work, reload application fixtures by running this command from the terminal:

```
$ php app/console doctrine:fixtures:load
```

TIP If you want to create new users, run this other command:

```
$ php app/console app:add-user
```

This is a **demo application** built in the Symfony Framework to illustrate the recommended way of developing Symfony applications.

For more information, check out the [Symfony Documentation](#).

Click on this button to show the source code of the **Controller** and **template** used to render this page.

 Show source code

Use PhpStorm

+ the Symfony plugin

<http://bit.ly/phpstorm-symfony>

PhpStorm EAP User

Expiration date: May 22, 2015



The Book

The Cookbook

The Components

Symfony Best Practices

Reference

Index

Contributing

Bundles

Symfony CMF

Docs for symfony 1.x

Learn Symfony

[Read the book →](#)

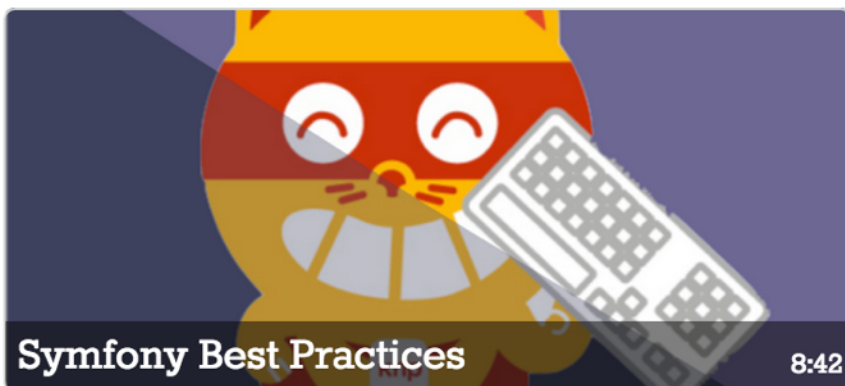
The Symfony Book

Prepared by the core team, this is the reference that covers the needs for every kind of Symfony developer.

[Controllers](#)[Routing](#)[Templating](#)[Doctrine](#)[Testing](#)[Forms](#)[Validation](#)[Security](#)[HTTP Cache](#)[Translation](#)[Services](#)[Performance](#)

<http://symfony.com/doc>

KnpuUniversity.com Screencasts



Use Silex!

```
require_once __DIR__.' /vendor/autoload.php';

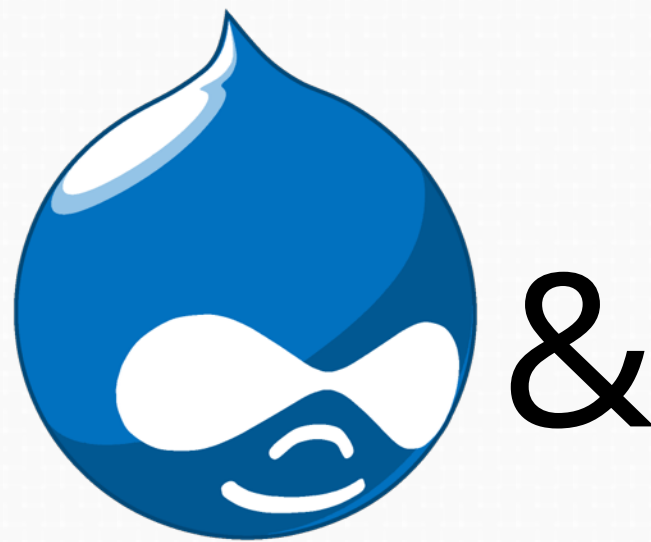
$app = new Silex\Application();

$app->get('/hello/{name}', function($name) use($app) {
    return 'Hello '.$app->escape($name);
});

$app->run();
```


Use D8

Act 8



PRINCIPAL THEMES

- Request/Response
- Routing/Controller
- PHP Namespaces/Autoloading
- Services/Container
- Events/Listeners
- Profiler

All are the same in **Silex, Drupal & Symfony**

You can use Silex to learn Drupal!



You can use Silex
to learn Symfony!



Symfony

You can use Symfony
to learn Drupal!



Finally, We have more
tools to solve problems



THANK YOU!

Ryan Weaver
@weaverryan



KNPUniversity

