



**DRUPALCON
SYDNEY**

CODING & DEVELOPMENT | PIERS WARMERS | FEBRUARY 7 2013

Symfony Components

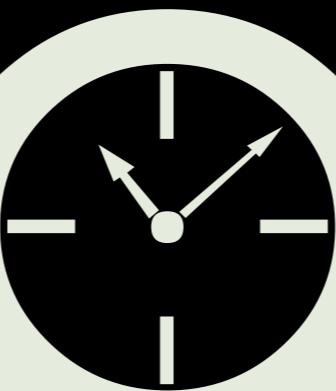
A Travel Guide

Piers Warmers



@warmers

*Part 1:
a step back in time...*



PHP since ~1996...

We were bad

```
<?php

$link = mysql_connect('localhost', 'root', 'root') or die('Connection Failed: ' . mysql_error());
mysql_select_db('drupalCon') or die('Could not select database');
$result = mysql_query('SELECT * FROM people') or die('Query failed: ' . mysql_error());

echo "<table>";

while ($line = mysql_fetch_array($result, MYSQL_ASSOC)) {
    echo "<tr>";
    foreach ($line as $col_value) {
        echo "<td>$col_value</td>";
    }
    echo "</tr>";
}

echo "</table>";

mysql_free_result($result);
mysql_close($link);
```

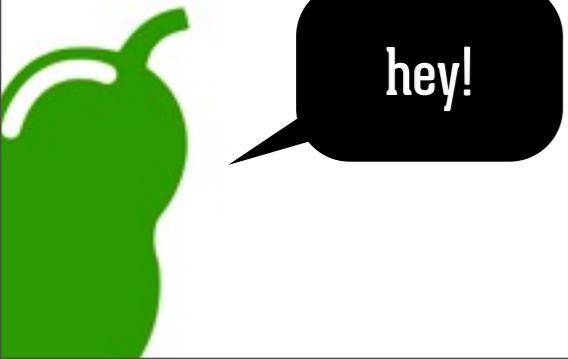
...and despite having strong open source community...

...and despite having strong open source community...

Sharing PHP = copy+paste

...and despite having strong open source community...

Sharing PHP = copy+paste
...and PEAR





is broken!

...but in 2009 something changed

...but in 2009 something changed

- Here comes PHP 5.3**

...but in 2009 something changed

- Here comes PHP 5.3**
- Lots of new features [including Namespaces, sweet!]**

...but in 2009 something changed

- Here comes PHP 5.3
- Lots of new features [including Namespaces, sweet!]
- Soon after FIG publishes PSR-0, 1, and 2

PSR-0: Getting organised

```
<?php

namespace Acme\Toolkit\Exception;

class FileNotFoundException extends \RuntimeException
{
    /**
     * Constructor.
     *
     * @param string      $message
     * @param int         $code
     * @param \Exception $previous
     */
    public function __construct($message = "Not Found", $code = 0, \Exception $previous = null)
    {
        parent::__construct($message, $code, $previous);
    }
}
```

PSR-0: Getting organised

```
<?php

namespace Acme\Toolkit\Exception;

class FileNotFoundException extends \RuntimeException
{
    /**
     * Constructor.
     *
     * @param string      $message
     * @param int         $code
     * @param \Exception $previous
     */
    public function __construct($message = "Not Found", $code = 0, \Exception $previous = null)
    {
        parent::__construct($message, $code, $previous);
    }
}
```



The class:

\Acme\Toolkit\Exception\FileNotFoundException

PSR-0: Getting organised

```
<?php

namespace Acme\Toolkit\Exception;

class FileNotFoundException extends \RuntimeException
{
    /**
     * Constructor.
     *
     * @param string      $message
     * @param int         $code
     * @param \Exception $previous
     */
    public function __construct($message = "Not Found", $code = 0, \Exception $previous = null)
    {
        parent::__construct($message, $code, $previous);
    }
}
```

The class:
\\Acme\\Toolkit\\Exception\\FileNotFoundException

Would be located in:
/Acme/Toolkit/Exception/FileNotFoundException.php

PSR-1: Getting standardised

```
<?php

namespace Acme\Entity\Message;

class Message {

    const STATUS_SENT = 'SENT';

    var $recipientList;

    function __construct() {
        $this->recipientList = array();
    }

    function addRecipient($recipient) {
        $this->recipientList[$recipient->getId()] = $recipient;
        return $this;
    }
}
```

PSR-1: Getting standardised

```
<?php

namespace Acme\Entity\Message;

class Message {
    const STATUS_SENT = 'SENT';

    var $recipientList;

    function __construct() {
        $this->recipientList = array();
    }

    function addRecipient($recipient) {
        $this->recipientList[$recipient->getId()] = $recipient;
        return $this;
    }
}
```

PSR-1: Getting standardised

```
<?php

namespace Acme\Entity\Message;

class Message {

    const STATUS_SENT = 'SENT';

    var $recipientList;

    function __construct() {
        $this->recipientList = array();
    }

    function addRecipient($recipient) {
        $this->recipientList[$recipient->getId()] = $recipient;
        return $this;
    }
}
```

PSR-1: Getting standardised

```
<?php

namespace Acme\Entity\Message;

class Message {

    const STATUS_SENT = 'SENT';

    var $recipientList;

    function __construct() {
        $this->recipientList = array();
    }

    function addRecipient($recipient) {
        $this->recipientList[$recipient->getId()] = $recipient;
        return $this;
    }
}
```

PSR-1: Getting standardised

```
<?php

namespace Acme\Entity\Message;

class Message {

    const STATUS_SENT = 'SENT';

    var $recipientList;

    function __construct() {
        $this->recipientList = array();
    }

    function addRecipient($recipient) {
        $this->recipientList[$recipient->getId()] = $recipient;
        return $this;
    }
}
```

PSR-2: Getting stylish

```
<?php
namespace Acme\Entity\Message;

use Acme\Entity\Recipient;

class Message
{
    const STATUS_SENT = 'SENT';

    private $recipientList;

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

    public function addRecipient(Recipient $recipient)
    {
        $this->recipientList[$recipient->getId()] = $recipient;
        return $this;
    }
}
```

PSR-2: Getting stylish

```
<?php
namespace Acme\Entity\Message;

use Acme\Entity\Recipient;

class Message
{
    const STATUS_SENT = 'SENT';

    private $recipientList;

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

    public function addRecipient(Recipient $recipient)
    {
        $this->recipientList[$recipient->getId()] = $recipient;
        return $this;
    }
}
```

PSR-2: Getting stylish

```
<?php
namespace Acme\Entity\Message;

use Acme\Entity\Recipient;

class Message
{
    const STATUS_SENT = 'SENT';

    private $recipientList;

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

    public function addRecipient(Recipient $recipient)
    {
        $this->recipientList[$recipient->getId()] = $recipient;
        return $this;
    }
}
```

PSR-2: Getting stylish

```
<?php
namespace Acme\Entity\Message;

use Acme\Entity\Recipient;

class Message
{
    const STATUS_SENT = 'SENT';

    private $recipientList;

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

    public function addRecipient(Recipient $recipient)
    {
        $this->recipientList[$recipient->getId()] = $recipient;
        return $this;
    }
}
```

PSR-2: Getting stylish

```
<?php
namespace Acme\Entity\Message;

use Acme\Entity\Recipient;

class Message
{
    const STATUS_SENT = 'SENT';

    private $recipientList;

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

    public function addRecipient(Recipient $recipient)
    {
        $this->recipientList[$recipient->getId()] = $recipient;
        return $this;
    }
}
```

PSR-2: Getting stylish

```
<?php
namespace Acme\Entity\Message;

use Acme\Entity\Recipient;

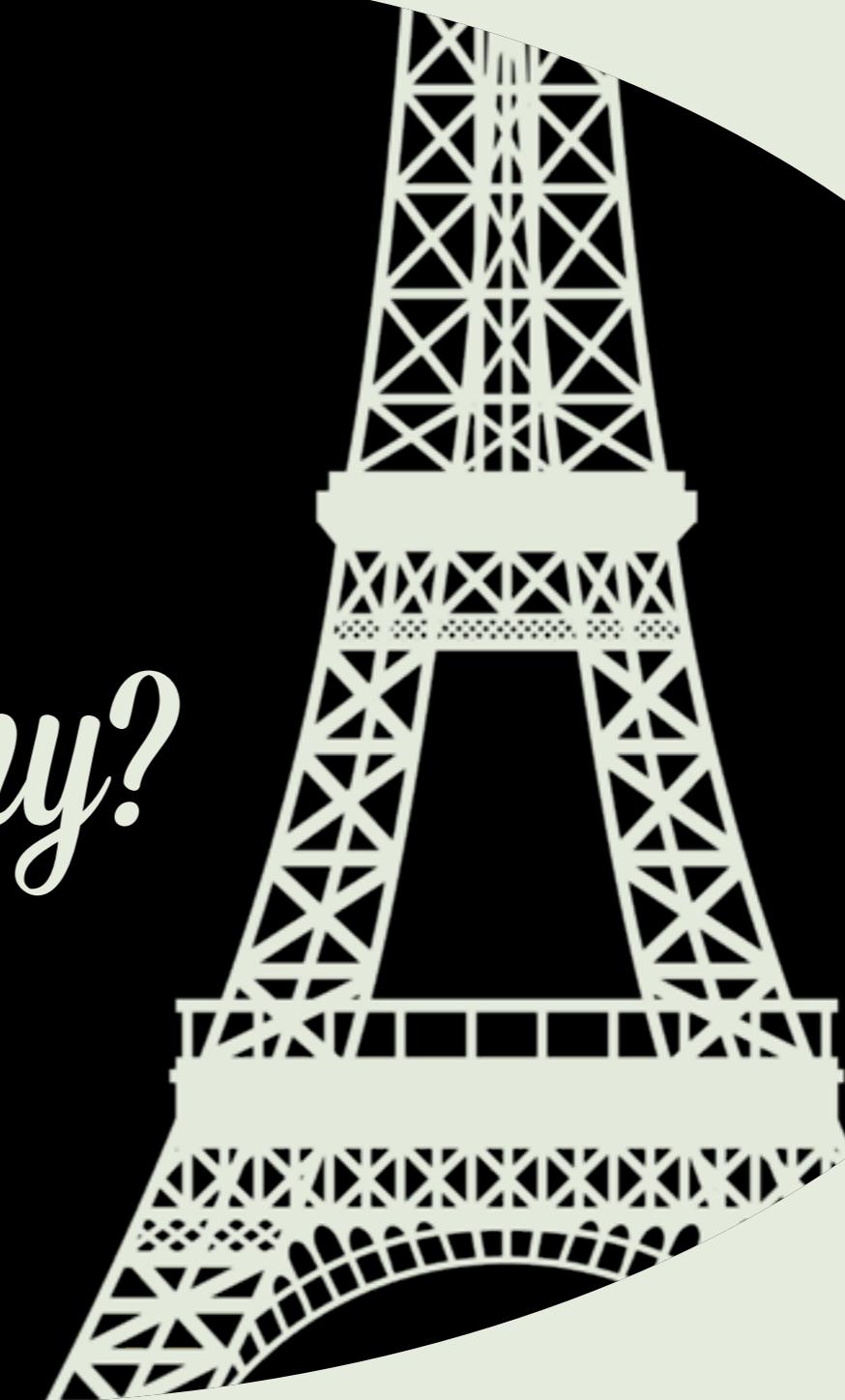
class Message
{
    const STATUS_SENT = 'SENT';

    private $recipientList;

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

    public function addRecipient(Recipient $recipient)
    {
        $this->recipientList[$recipient->getId()] = $recipient;
        return $this;
    }
}
```

Part 2:
what is Symphony?



Symfony is...

Symfony is...

- A collection of Object-Oriented PHP 5.3+ classes

Symfony is...

- A collection of Object-Oriented PHP 5.3+ classes
- That all conform to PSR-0, 1, 2 and 3

Symfony is...

- A collection of Object-Oriented PHP 5.3+ classes
- That all conform to PSR-0, 1, 2 and 3
- These classes are grouped into components

Symfony Framework is...

Symfony Framework is...

- Simply an implementation of Symfony Components

Symfony Framework is...

- Simply an implementation of Symfony Components
- This implies something interesting...

“SYMFONY IS A REUSABLE SET OF
STANDALONE, DECOUPLED AND COHESIVE
PHP COMPONENTS THAT SOLVE COMMON
WEB DEVELOPMENT PROBLEMS”

DRIES BUYTAERT

Symfony Components



includes

BrowserKit	HttpKernel
ClassLoader	Locale
Console	Process
Config	Routing
CssSelector	Security
DependencyInjection	Serializer
DomCrawler	Templating
EventDispatcher	Translation
Finder	Validator
Form	Yaml
HttpFoundation	

Decoupled Components

Decoupled Components

- Can be moved and used in many contexts

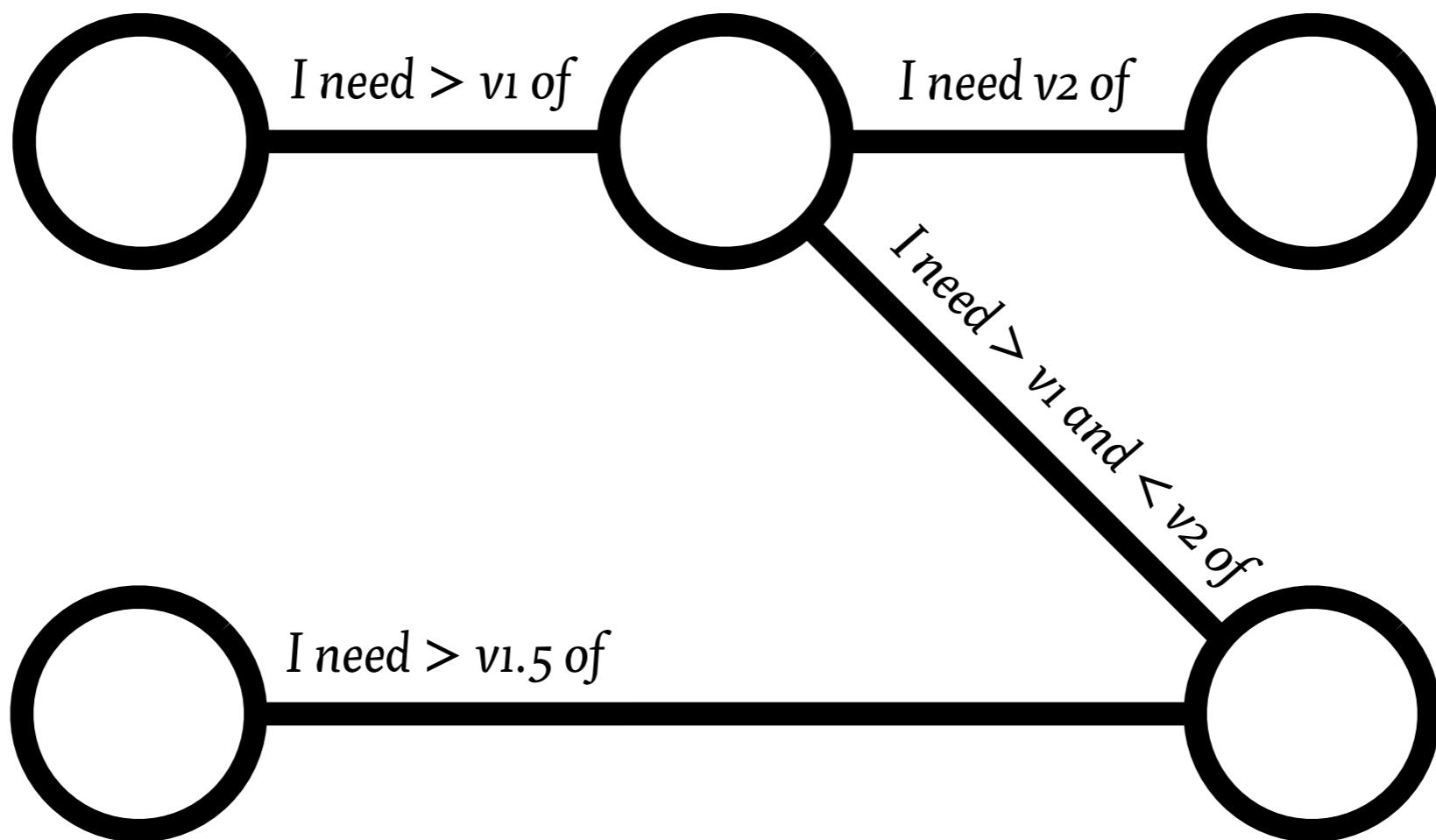
Decoupled Components

- Can be moved and used in many contexts
- This has been a core development philosophy

Decoupled Components

- Can be moved and used in many contexts
- This has been a core development philosophy
- What about cross dependancies?

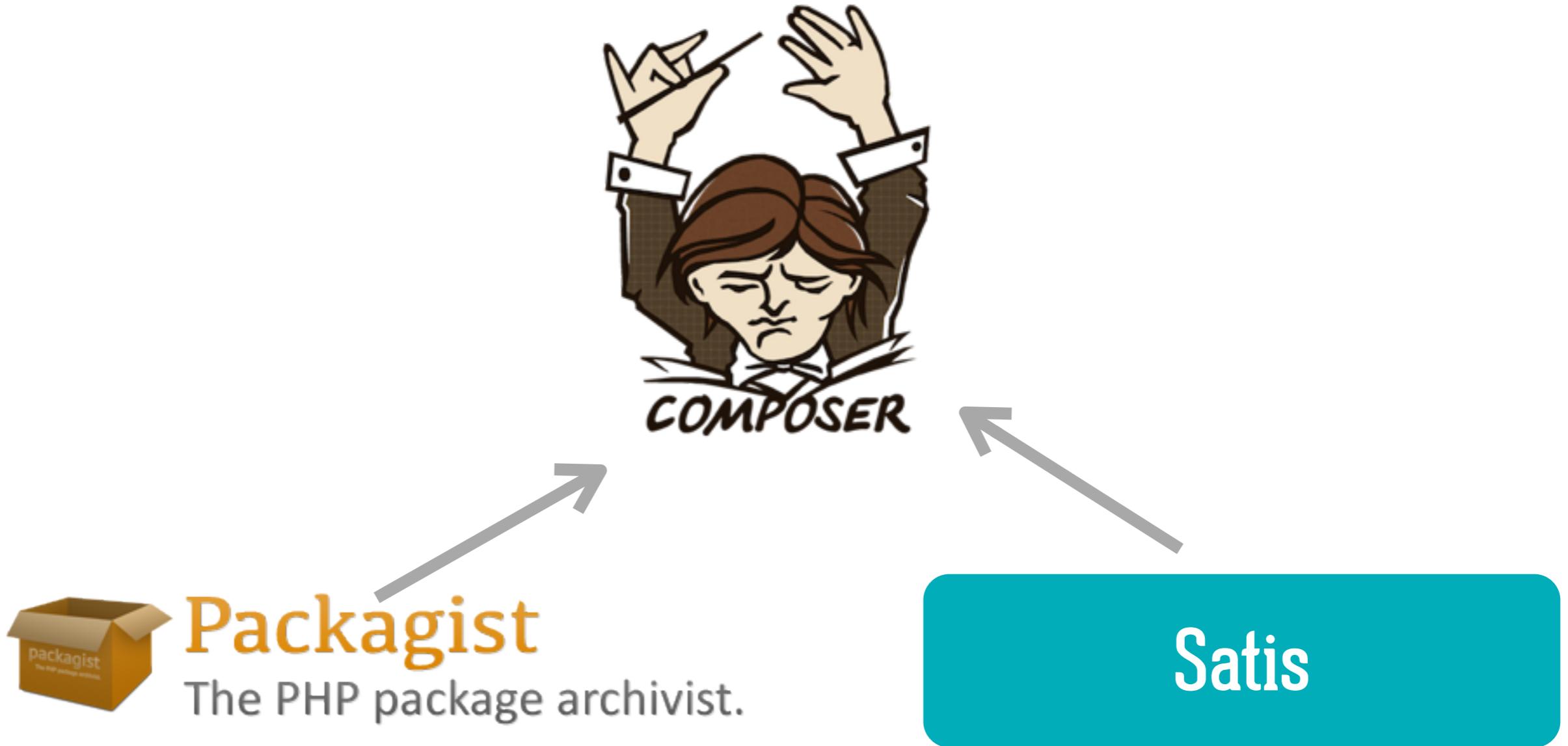
Dependancy management is complicated





Dependency Manager

PSR-0 + SPDX + SemVer



See: <https://packagist.org/> and <https://github.com/composer/satis>

Packagist

https://packagist.org/search/?q=aws

Create a new account | Login

Packagist

The PHP package archivist.

Submit Package

Packagist is the main **Composer** repository. It aggregates all sorts of PHP packages that are installable with Composer.
Browse packages or submit your own.

aws

Packages

aws/aws-sdk-php ④ 2 424 ★ 1
AWS SDK for PHP

amazonwebservices/aws-sdk-for-php ④ 14 269 ★ 3
AWS SDK for PHP

jmhobbs/swiftmailer-transport-aws-ses ④ 305 ★ 0
Add AWS SES support to Swiftmailer

cybernox/amazon-webservices-bundle ④ 2 255 ★ 0
A Symfony2 Bundle for interfacing with Amazon Web Services (AWS)

99designs/facade ④ 117 ★ 0
A PHP5 library for accessing various services via a stream based request/response pattern

aw/zeta-cache-bundle ④ 4 ★ 0

Composer and Packagist

Composer and Packagist

- Map dependancies [composer.json]

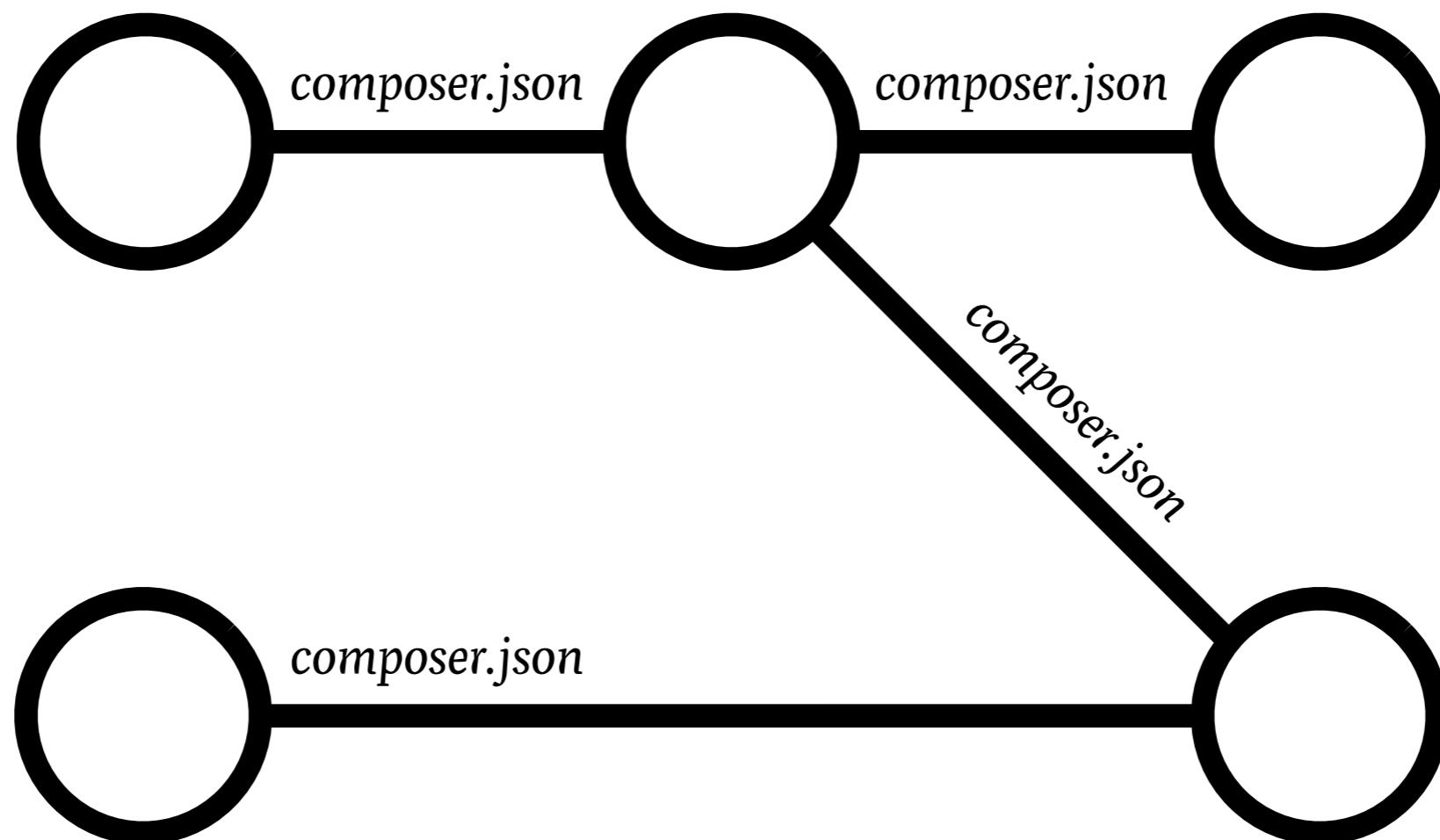
Composer and Packagist

- Map dependancies [[composer.json](#)]
- Handle installs and updates [[composer.phar](#)]

Composer and Packagist

- Map dependancies [[composer.json](#)]
- Handle installs and updates [[composer.phar](#)]
- Share dependency states [[composer.lock](#)]

Dependancies defined in: **composer.json**



composer.json [<https://drupal.org/project/git>]

```
{  
  "name": "drupal/git",  
  "description": "Wrapper for the Gitter library for Drupal.",  
  "type": "drupal-module",  
  "license": "GPL-2.0+",  
  "authors": [  
    {  
      "name": "Dave Hall",  
      "email": "skwashd@gmail.com"  
    },  
    {  
      "name": "James Harvey",  
      "email": "james@webmarkdesigns.com.au"  
    }  
],  
  "require": {  
    "klaussilveira/gitter": "dev-master"  
  },  
  "repositories": [  

```

composer.json [<https://drupal.org/project/git>]

```
{  
  "name": "drupal/git",  
  "description": "Wrapper for the Gitter library for Drupal.",  
  "type": "drupal-module",  
  "license": "GPL-2.0+",  
  "authors": [  
    {  
      "name": "Dave Hall",  
      "email": "skwashd@gmail.com"  
    },  
    {  
      "name": "James Harvey",  
      "email": "james@webmarkdesigns.com.au"  
    }  
,  
    "require": {  
      "klaussilveira/gitter": "dev-master"  
    },  
    "repositories": [  
      {  
        "type": "vcs",  
        "url": "https://github.com/skwashd/gitter"  
      }  
    ]  
  }  
}
```

Terminal:

composer.json [<https://drupal.org/project/git>]

```
{  
  "name": "drupal/git",  
  "description": "Wrapper for the Gitter library for Drupal.",  
  "type": "drupal-module",  
  "license": "GPL-2.0+",  
  "authors": [  
    {  
      "name": "Dave Hall",  
      "email": "skwashd@gmail.com"  
    },  
    {  
      "name": "James Harvey",  
      "email": "james@webmarkdesigns.com.au"  
    }  
,  
    "require": {  
      "klaussilveira/gitter": "dev-master"  
    },  
    "repositories": [  
      {  
        "type": "vcs",  
        "url": "https://github.com/skwashd/gitter"  
      }  
    ]  
  }  
}
```

Terminal:

```
$ curl -s https://getcomposer.org/installer | php
```

composer.json [<https://drupal.org/project/git>]

```
{  
  "name": "drupal/git",  
  "description": "Wrapper for the Gitter library for Drupal.",  
  "type": "drupal-module",  
  "license": "GPL-2.0+",  
  "authors": [  
    {  
      "name": "Dave Hall",  
      "email": "skwashd@gmail.com"  
    },  
    {  
      "name": "James Harvey",  
      "email": "james@webmarkdesigns.com.au"  
    }  
,  
    "require": {  
      "klaussilveira/gitter": "dev-master"  
    },  
    "repositories": [  
      {  
        "type": "vcs",  
        "url": "https://github.com/skwashd/gitter"  
      }  
    ]  
  }  
}
```

Terminal:

```
$ curl -s https://getcomposer.org/installer | php  
$ php composer.phar install
```

composer.json [<https://drupal.org/project/git>]

```
{  
  "name": "drupal/git",  
  "description": "Wrapper for the Gitter library for Drupal.",  
  "type": "drupal-module",  
  "license": "GPL-2.0+",  
  "authors": [  
    {  
      "name": "Dave Hall",  
      "email": "skwashd@gmail.com"  
    },  
    {  
      "name": "James Harvey",  
      "email": "james@webmarkdesigns.com.au"  
    }  
,  
    "require": {  
      "klaussilveira/gitter": "dev-master"  
    },  
    "repositories": [  
      {  
        "type": "vcs",  
        "url": "https://github.com/skwashd/gitter"  
      }  
    ]  
  }
```

Terminal:

```
$ curl -s https://getcomposer.org/installer | php  
$ php composer.phar install
```

Loading composer repositories with package information
Installing dependencies

- Installing `symfony/process` [v2.1.7]
 Downloading: 100%

- Installing `klaussilveira/gitter` [dev-master 7df3001]
 Cloning 7df3001e1fb40cc114aeeb8e8fda887f329def78

Writing lock file
Generating autoload files

./vendor/autoload.php

```
<?php  
// autoload.php generated by Composer  
require_once __DIR__ . '/composer' . '/autoload_real.php';  
return ComposerAutoloaderInit4c8f50218055a97a43b4b03f25f7a345::getLoader();
```

./vendor/autoload.php

```
<?php  
// autoload.php generated by Composer  
require_once __DIR__ . '/composer' . '/autoload_real.php';  
return ComposerAutoloaderInit4c8f50218055a97a43b4b03f25f7a345::getLoader();
```

Include in your own code for easy loading of classes:

```
<?php  
require_once __DIR__ . '/vendor/autoload.php';  
// your code goes here...
```

This stuff is really cool!

This stuff is really cool!

- Lots of libraries now managed: 7,200+

This stuff is really cool!

- Lots of libraries now managed: 7,200+
- Simple end to end solution for package distribution

This stuff is really cool!

- Lots of libraries now managed: 7,200+
- Simple end to end solution for package distribution
- Helps break the "not invented here" cycle

This stuff is really cool!

- Lots of libraries now managed: 7,200+
- Simple end to end solution for package distribution
- Helps break the "not invented here" cycle
- Removes the barriers associated with previous distribution tools

This stuff is really cool!

- Lots of libraries now managed: 7,200+
- Simple end to end solution for package distribution
- Helps break the "not invented here" cycle
- Removes the barriers associated with previous distribution tools
- Allowing the PHP community to...

take isolated projects...

Application 1

S B G D ? ? ?

Application 2

W G C T ? ? ?

Framework 1

I R E D ? ? ?

Framework 2

A H N D ? ? ?

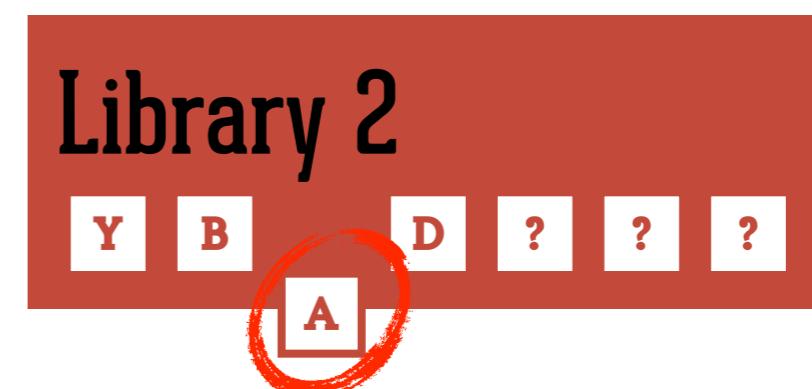
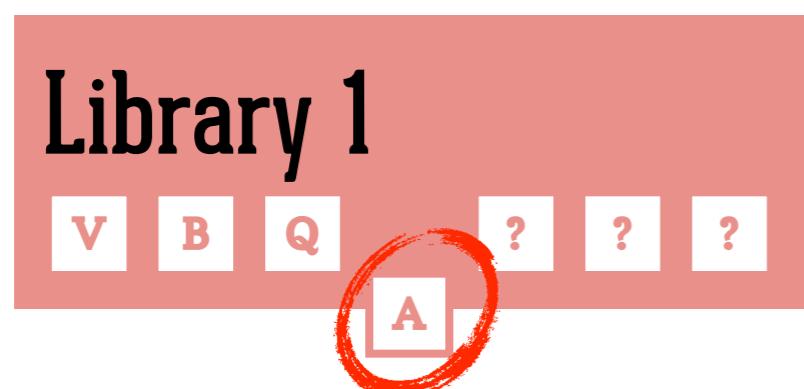
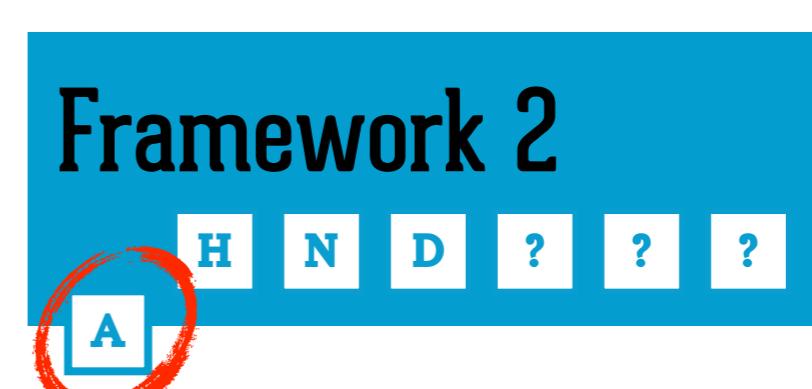
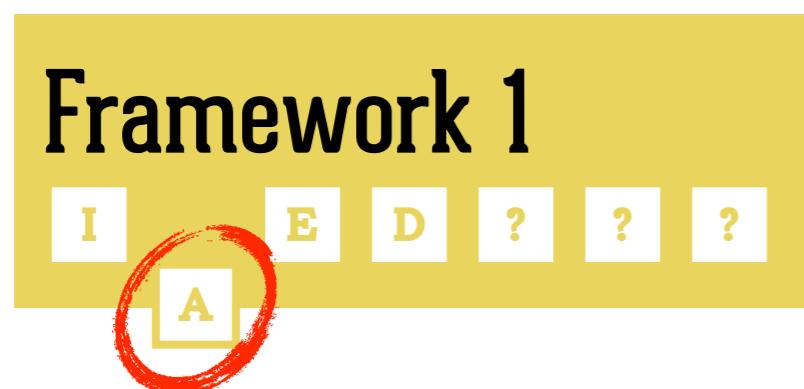
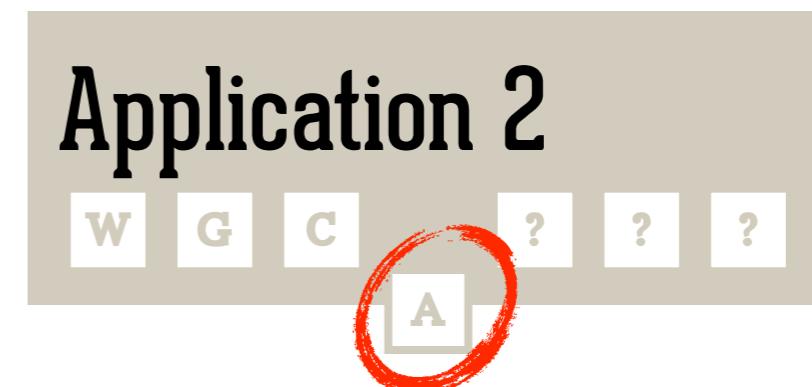
Library 1

V B Q A ? ? ?

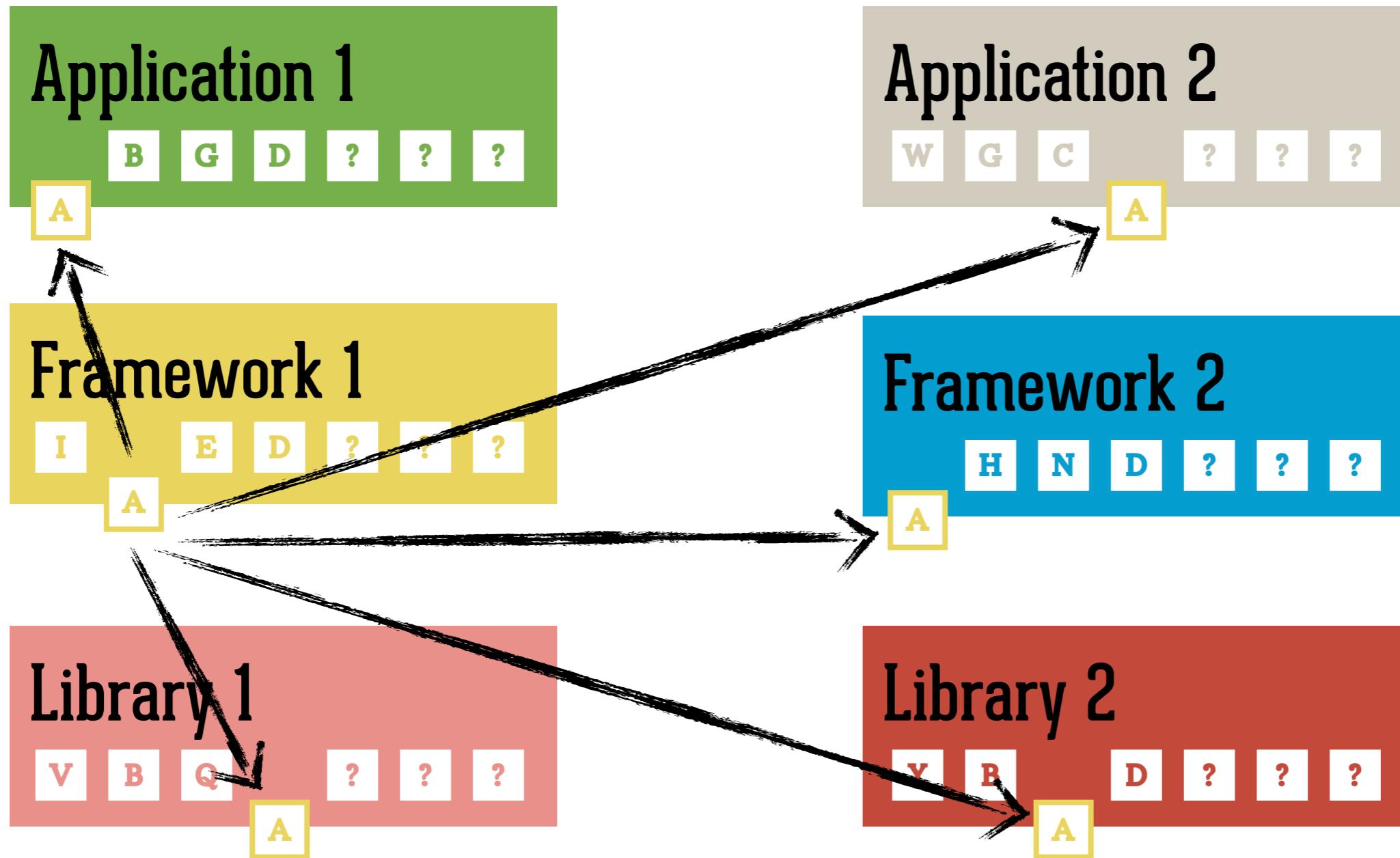
Library 2

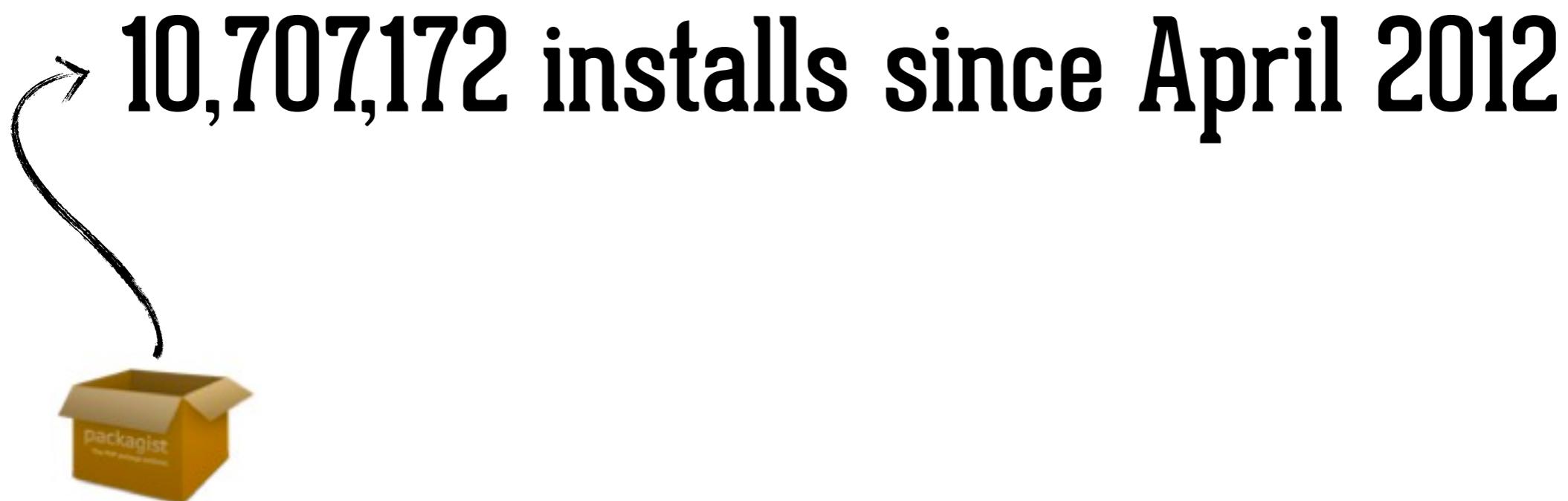
Y B A D ? ? ?

identifying common logic...



and share properly!





10,707,172 installs since April 2012

Part 3:

`<?= “lets see the code!”;`



Recieve a request

composer.json

```
{  
    "require": {  
        "symfony/http-foundation": "2.1.*"  
    }  
}
```

Recieve a request

`composer.json`

```
{  
    "require": {  
        "symfony/http-foundation": "2.1.*"  
    }  
}
```

`request.php`

```
<?php  
  
require_once __DIR__ . '/vendor/autoload.php';  
  
use Symfony\Component\HttpFoundation\Request;  
  
$request = Request::createFromGlobals();  
  
$foo = $request->get('foo');  
  
// do something with $foo...
```

Recieve a request

composer.json

```
{  
    "require": {  
        "symfony/http-foundation": "2.1.*"  
    }  
}
```

request.php

```
<?php  
require_once __DIR__ . '/vendor/autoload.php';  
  
use Symfony\Component\HttpFoundation\Request;  
  
$request = Request::createFromGlobals();  
  
$foo = $request->get('foo');  
  
// do something with $foo...
```

Recieve a request

composer.json

```
{  
    "require": {  
        "symfony/http-foundation": "2.1.*"  
    }  
}
```

request.php

```
<?php  
  
require_once __DIR__ . '/vendor/autoload.php';  
  
use Symfony\Component\HttpFoundation\Request;  
  
$request = Request::createFromGlobals();  
  
$foo = $request->get('foo');  
  
// do something with $foo...
```

Recieve a request

composer.json

```
{  
    "require": {  
        "symfony/http-foundation": "2.1.*"  
    }  
}
```

request.php

```
<?php  
  
require_once __DIR__ . '/vendor/autoload.php';  
  
use Symfony\Component\HttpFoundation\Request;  
  
$request = Request::createFromGlobals();  
  
$foo = $request->get('foo');  
  
// do something with $foo...
```

Recieve a request

composer.json

```
{  
    "require": {  
        "symfony/http-foundation": "2.1.*"  
    }  
}
```

request.php

```
<?php  
  
require_once __DIR__ . '/vendor/autoload.php';  
  
use Symfony\Component\HttpFoundation\Request;  
  
$request = Request::createFromGlobals();  
  
$foo = $request->get('foo');  
  
// do something with $foo...
```

Recieve a request

composer.json

```
{  
    "require": {  
        "symfony/http-foundation": "2.1.*"  
    }  
}
```

request.php

```
<?php  
  
require_once __DIR__ . '/vendor/autoload.php';  
  
use Symfony\Component\HttpFoundation\Request;  
  
$request = Request::createFromGlobals();  
  
$foo = $request->get('foo');  
  
// do something with $foo...
```

Send a response

composer.json

```
{  
    "require": {  
        "symfony/http-foundation": "2.1.*"  
    }  
}
```

Send a response

composer.json

```
{  
    "require": {  
        "symfony/http-foundation": "2.1.*"  
    }  
}
```

response.php

```
<?php  
  
require_once __DIR__ . '/vendor/autoload.php';  
  
use Symfony\Component\HttpFoundation\Response;  
  
$response = new Response();  
  
$response  
    ->setContent('Hello DrupalCon!')  
    ->setStatusCode(200)  
    ->setTtl(3600)  
    ->send()  
;
```

Send a response

composer.json

```
{  
    "require": {  
        "symfony/http-foundation": "2.1.*"  
    }  
}
```

response.php

```
<?php  
  
require_once __DIR__ . '/vendor/autoload.php';  
  
use Symfony\Component\HttpFoundation\Response;  
  
$response = new Response();  
  
$response  
    ->setContent('Hello DrupalCon!')  
    ->setStatusCode(200)  
    ->setTtl(3600)  
    ->send()  
;  
;
```

Send a response

composer.json

```
{  
    "require": {  
        "symfony/http-foundation": "2.1.*"  
    }  
}
```

response.php

```
<?php  
  
require_once __DIR__ . '/vendor/autoload.php';  
  
use Symfony\Component\HttpFoundation\Response;  
  
$response = new Response();  
  
$response  
    ->setContent('Hello DrupalCon!')  
    ->setStatusCode(200)  
    ->setTtl(3600)  
    ->send()  
;
```

Create a route

composer.json

```
{  
    "require": {  
        "symfony/routing": "2.1.*"  
    }  
}
```

Create a route

composer.json

```
{  
    "require": {  
        "symfony/routing": "2.1.*"  
    }  
}
```

route.php

```
<?php  
  
require_once __DIR__().'/vendor/autoload.php';  
  
use ...;  
  
$routes = new RouteCollection();  
  
$routes->add(  
    'say',  
    new Route(  
        '/say/{message}/{name}',  
        array('message' => 'hello', 'name' => 'world')  
    )  
);  
  
...
```

Match a route

route.php [continued]

```
<?php  
...  
$context = new RequestContext();  
$request = Request::createFromGlobals();  
$context->fromRequest($request);  
  
$response = new Response();  
  
try {  
    $matcher = new UrlMatcher($routes, $context);  
    $attributes = $matcher->match($request->getPathInfo());  
  
    if($attributes['_route'] == 'say') {  
        $response->setContent(sprintf('%s %s', $attributes['message'], $attributes['name']));  
    }  
} catch (MethodNotAllowedException $e) {  
    $response->setContent('Method Not Allowed')->setStatusCode(405);  
} catch (ResourceNotFoundException $e) {  
    $response->setContent('Not Found')->setStatusCode(404);  
}  
  
$response->send();
```

Match a route

route.php [continued]

```
<?php  
...  
  
$context = new RequestContext();  
$request = Request::createFromGlobals();  
$context->fromRequest($request);  
  
$response = new Response();  
  
try {  
    $matcher = new UrlMatcher($routes, $context);  
    $attributes = $matcher->match($request->getPathInfo());  
  
    if($attributes['_route'] == 'say') {  
        $response->setContent(sprintf('%s %s', $attributes['message'], $attributes['name']));  
    }  
} catch (MethodNotAllowedException $e) {  
    $response->setContent('Method Not Allowed')->setStatusCode(405);  
} catch (ResourceNotFoundException $e) {  
    $response->setContent('Not Found')->setStatusCode(404);  
}  
  
$response->send();
```

Match a route

route.php [continued]

```
<?php  
...  
  
$context = new RequestContext();  
$request = Request::createFromGlobals();  
$context->fromRequest($request);  
  
$response = new Response();  
  
try {  
    $matcher = new UrlMatcher($routes, $context);  
    $attributes = $matcher->match($request->getPathInfo());  
  
    if($attributes['_route'] == 'say') {  
        $response->setContent(sprintf('%s %s', $attributes['message'], $attributes['name']));  
    }  
} catch (MethodNotAllowedException $e) {  
    $response->setContent('Method Not Allowed')->setStatusCode(405);  
} catch (ResourceNotFoundException $e) {  
    $response->setContent('Not Found')->setStatusCode(404);  
}  
  
$response->send();
```

Match a route

route.php [continued]

```
<?php  
...  
  
$context = new RequestContext();  
$request = Request::createFromGlobals();  
$context->fromRequest($request);  
  
$response = new Response();  
  
try {  
    $matcher = new UrlMatcher($routes, $context);  
    $attributes = $matcher->match($request->getPathInfo());  
  
    if($attributes['_route'] == 'say') {  
        $response->setContent(sprintf('%s %s', $attributes['message'], $attributes['name']));  
    }  
} catch (MethodNotAllowedException $e) {  
    $response->setContent('Method Not Allowed')->setStatusCode(405);  
} catch (ResourceNotFoundException $e) {  
    $response->setContent('Not Found')->setStatusCode(404);  
}  
  
$response->send();
```

Match a route

route.php [continued]

```
<?php  
...  
  
$context = new RequestContext();  
$request = Request::createFromGlobals();  
$context->fromRequest($request);  
  
$response = new Response();  
  
try {  
    $matcher = new UrlMatcher($routes, $context);  
    $attributes = $matcher->match($request->getPathInfo());  
  
    if($attributes['_route'] == 'say') {  
        $response->setContent(sprintf('%s %s', $attributes['message'], $attributes['name']));  
    }  
} catch (MethodNotAllowedException $e) {  
    $response->setContent('Method Not Allowed')->setStatusCode(405);  
} catch (ResourceNotFoundException $e) {  
    $response->setContent('Not Found')->setStatusCode(404);  
}  
  
$response->send();
```

Use a templating engine

`composer.json`

```
{  
    "require": {  
        "twig/twig": "1.*",  
        "symfony/http-foundation": "2.1.*"  
    }  
}
```

Use a templating engine

composer.json

```
{  
    "require": {  
        "twig/twig": "1.*",  
        "symfony/http-foundation": "2.1.*"  
    }  
}
```

route.php

```
<?php  
  
require_once __DIR__ . '/vendor/autoload.php';  
  
use Symfony\Component\HttpFoundation\Response;  
  
$loader = new Twig_Loader_Filesystem(__DIR__);  
$twig = new Twig_Environment($loader);  
  
$response = new Response(  
    $twig->render(  
        'index.html.twig',  
        array(  
            'message' => 'Hi there DrupalCon!',  
        )  
    )  
);  
  
$response->send();
```

Use a templating engine

composer.json

```
{  
    "require": {  
        "twig/twig": "1.*",  
        "symfony/http-foundation": "2.1.*"  
    }  
}
```

route.php

```
<?php  
  
require_once __DIR__ . '/vendor/autoload.php';  
  
use Symfony\Component\HttpFoundation\Response;  
  
$loader = new Twig_Loader_Filesystem(__DIR__);  
$twig = new Twig_Environment($loader);  
  
$response = new Response(  
    $twig->render(  
        'index.html.twig',  
        array(  
            'message' => 'Hi there DrupalCon!',  
        )  
    )  
);  
  
$response->send();
```

Use a templating engine

composer.json

```
{  
    "require": {  
        "twig/twig": "1.*",  
        "symfony/http-foundation": "2.1.*"  
    }  
}
```

route.php

```
<?php  
  
require_once __DIR__ . '/vendor/autoload.php';  
  
use Symfony\Component\HttpFoundation\Response;  
  
$loader = new Twig_Loader_Filesystem(__DIR__);  
$twig = new Twig_Environment($loader);  
  
$response = new Response(  
    $twig->render(  
        'index.html.twig',  
        array(  
            'message' => 'Hi there DrupalCon!',  
        )  
    )  
);  
  
$response->send();
```

Use a templating engine

composer.json

```
{  
    "require": {  
        "twig/twig": "1.*",  
        "symfony/http-foundation": "2.1.*"  
    }  
}
```

route.php

```
<?php  
  
require_once __DIR__ . '/vendor/autoload.php';  
  
use Symfony\Component\HttpFoundation\Response;  
  
$loader = new Twig_Loader_Filesystem(__DIR__);  
$twig = new Twig_Environment($loader);  
  
$response = new Response(  
    $twig->render(  
        'index.html.twig',  
        array(  
            'message' => 'Hi there DrupalCon!',  
        )  
    )  
);  
  
$response->send();
```

index.html.twig

```
<html>  
  <head>  
    <title>...</title>  
  </head>  
  <body>  
    <h1>{{ message }}</h1>  
  </body>  
</html>
```

Use a templating engine

composer.json

```
{  
    "require": {  
        "twig/twig": "1.*",  
        "symfony/http-foundation": "2.1.*"  
    }  
}
```

route.php

```
<?php  
  
require_once __DIR__ . '/vendor/autoload.php';  
  
use Symfony\Component\HttpFoundation\Response;  
  
$loader = new Twig_Loader_Filesystem(__DIR__);  
$twig = new Twig_Environment($loader);  
  
$response = new Response(  
    $twig->render(  
        'index.html.twig',  
        array(  
            'message' => 'Hi there DrupalCon!',  
        )  
    )  
);  
  
$response->send();
```

index.html.twig

```
<html>  
  <head>  
    <title>...</title>  
  </head>  
  <body>  
    <h1>{{ message }}</h1>  
  </body>  
</html>
```

Use a templating engine

`composer.json`

```
{  
    "require": {  
        "twig/twig": "1.*",  
        "symfony/http-foundation": "2.1.*"  
    }  
}
```

`route.php`

```
<?php  
  
require_once __DIR__ . '/vendor/autoload.php';  
  
use Symfony\Component\HttpFoundation\Response;  
  
$loader = new Twig_Loader_Filesystem(__DIR__);  
$twig = new Twig_Environment($loader);  
  
$response = new Response(  
    $twig->render(  
        'index.html.twig',  
        array(  
            'message' => 'Hi there DrupalCon!',  
        )  
    )  
);  
  
$response->send();
```

`index.html.twig`

```
<html>  
  <head>  
    <title>...</title>  
  </head>  
  <body>  
    <h1>{{ message }}</h1>  
  </body>  
</html>
```

Use a templating engine

composer.json

```
{  
    "require": {  
        "twig/twig": "1.*",  
        "symfony/http-foundation": "2.1.*"  
    }  
}
```

route.php

```
<?php  
  
require_once __DIR__ . '/vendor/autoload.php';  
  
use Symfony\Component\HttpFoundation\Response;  
  
$loader = new Twig_Loader_Filesystem(__DIR__);  
$twig = new Twig_Environment($loader);  
  
$response = new Response(  
    $twig->render(  
        'index.html.twig',  
        array(  
            'message' => 'Hi there DrupalCon!',  
        )  
    )  
);  
  
$response->send();
```

index.html.twig

```
<html>  
  <head>  
    <title>...</title>  
  </head>  
  <body>  
    <h1>{{ message }}</h1>  
  </body>  
</html>
```

Logging to a file [PSR-3]

composer.json

```
{  
    "require": {  
        "monolog/monolog": "1.*"  
    }  
}
```

Logging to a file [PSR-3]

composer.json

```
{  
    "require": {  
        "monolog/monolog": "1.*"  
    }  
}
```

route.php

```
<?php  
  
require_once __DIR__ . '/vendor/autoload.php';  
  
use Monolog\Logger;  
use Monolog\Handler\StreamHandler;  
  
$log = new Logger('drupalCon');  
$log->pushHandler(new StreamHandler(__DIR__ . '/access.log', Logger::DEBUG));  
  
$log->addDebug('Hey, I just processed a request!');  
  
// From access.log:  
// [2013-02-05 22:59:42] drupalCon.DEBUG: Hey, I just processed a request! [] []
```

Logging to a file [PSR-3]

composer.json

```
{  
    "require": {  
        "monolog/monolog": "1.*"  
    }  
}
```

route.php

```
<?php  
  
require_once __DIR__ . '/vendor/autoload.php';  
  
use Monolog\Logger;  
use Monolog\Handler\StreamHandler;  
  
$log = new Logger('drupalCon');  
$log->pushHandler(new StreamHandler(__DIR__ . '/access.log', Logger::DEBUG));  
  
$log->addDebug('Hey, I just processed a request!');  
  
// From access.log:  
// [2013-02-05 22:59:42] drupalCon.DEBUG: Hey, I just processed a request! [] []
```

Logging to a file [PSR-3]

composer.json

```
{  
    "require": {  
        "monolog/monolog": "1.*"  
    }  
}
```

route.php

```
<?php  
  
require_once __DIR__ . '/vendor/autoload.php';  
  
use Monolog\Logger;  
use Monolog\Handler\StreamHandler;  
  
$log = new Logger('drupalCon');  
$log->pushHandler(new StreamHandler(__DIR__ . '/access.log', Logger::DEBUG));  
  
$log->addDebug('Hey, I just processed a request!');  
  
// From access.log:  
// [2013-02-05 22:59:42] drupalCon.DEBUG: Hey, I just processed a request! [] []
```

Logging to a file [PSR-3]

composer.json

```
{  
    "require": {  
        "monolog/monolog": "1.*"  
    }  
}
```

route.php

```
<?php  
  
require_once __DIR__ . '/vendor/autoload.php';  
  
use Monolog\Logger;  
use Monolog\Handler\StreamHandler;  
  
$log = new Logger('drupalCon');  
$log->pushHandler(new StreamHandler(__DIR__ . '/access.log', Logger::DEBUG));  
  
$log->addDebug('Hey, I just processed a request!');  
  
// From access.log:  
// [2013-02-05 22:59:42] drupalCon.DEBUG: Hey, I just processed a request! [] []
```

Part 4:

Drupal & Symfony?



Symfony Components



includes

BrowserKit	HttpKernel
ClassLoader	Locale
Console	Process
Config	Routing
CssSelector	Security
DependencyInjection	Serializer
DomCrawler	Templating
EventDispatcher	Translation
Finder	Validator
Form	Yaml
HttpFoundation	

Drupal & Symfony



Version 8+

using

BrowserKit	HttpKernel
ClassLoader	Locale
Console	Process
Config	Routing
CssSelector	Security
DependencyInjection	Serializer
DomCrawler	Templating
EventDispatcher [→ D9]	Translation
Finder	Validator
Form	Yaml
HttpFoundation	

What are these used for?

What are these used for?

- Class Loading utilities [yay PSR-0]

What are these used for?

- Class Loading utilities [yay PSR-0]
- Dependancy Injection [manage construction of objects]

What are these used for?

- Class Loading utilities [yay PSR-0]
- Dependancy Injection [manage construction of objects]
- HTTP Foundation and HTTP Kernel [request, response goodies]

What are these used for?

- Class Loading utilities [yay PSR-0]
- Dependancy Injection [manage construction of objects]
- HTTP Foundation and HTTP Kernel [request, response goodies]
- Routing [match requests with specific code]

What are these used for?

- Class Loading utilities [yay PSR-0]
- Dependancy Injection [manage construction of objects]
- HTTP Foundation and HTTP Kernel [request, response goodies]
- Routing [match requests with specific code]
- YAML [part of the new configuration changes]

What are these used for?

- Class Loading utilities [yay PSR-0]
- Dependancy Injection [manage construction of objects]
- HTTP Foundation and HTTP Kernel [request, response goodies]
- Routing [match requests with specific code]
- YAML [part of the new configuration changes]
- Some utility libraries [process handling, serializing]

This isn't limited to Symfony...

Other Components



Version 8+

using

Doctrine/Common

Utility libs used by Doctrine DBAL and ORM

Guzzle/[Common, HTTP, Parser, Stream]

HTTP client & framework for accessing RESTful services

EasyRdf/EasyRdf

Consume and produce RDF.

Symfony-CMF/Routing

Toolkit for chain route handling

plus...

KrisWallsmith/Assetic

Asset management framework

See: <https://github.com/kriswallsmith/assetic>

KrisWallsmith/Assetic

Asset management framework

- **CSS Filters** [LESS, SASS, Compass, STYL, image data, minify, rewrite]

KrisWallsmith/Assetic

Asset management framework

- **CSS Filters** [LESS, SASS, Compass, STYL, image data, minify, rewrite]
- **JS Filters** [Google Closure, YUI compressor, Packer, Handlebars, Sprockets]

KrisWallsmith/Assetic

Asset management framework

- **CSS Filters** [LESS, SASS, Compass, STYL, image data, minify, rewrite]
- **JS Filters** [Google Closure, YUI compressor, Packer, Handlebars, Sprockets]
- **Image Filters** [optimise and compress JPG's and PNG's]

and...

Twig/Twig

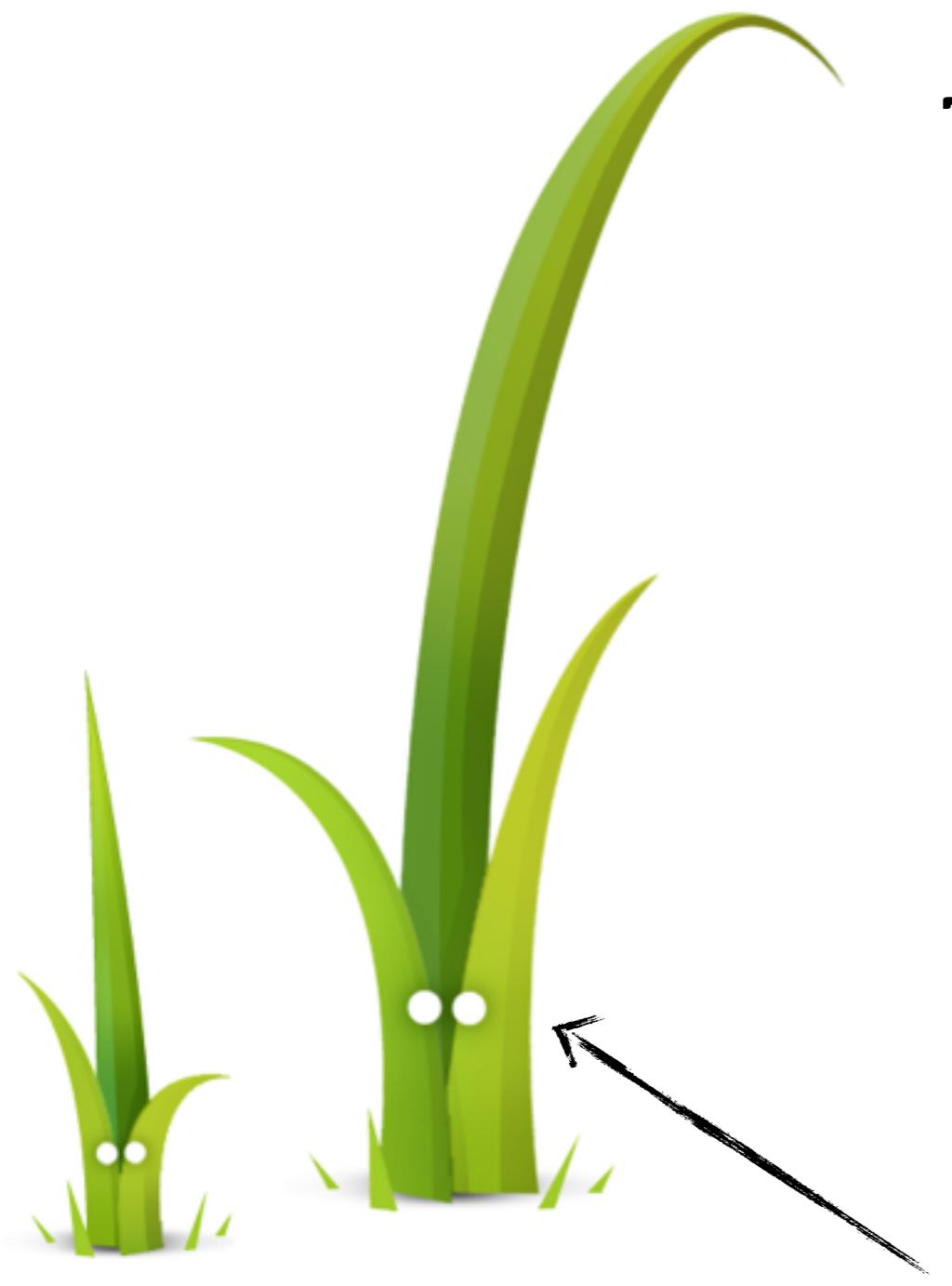
The infamous new template engine.



See: <http://twig.sensiolabs.org/>

Twig/Twig

The infamous new template engine.



cold dead eyes?

See: <http://twig.sensiolabs.org/>

Twig/Twig

The infamous new template engine.



See: <http://twig.sensiolabs.org/>

Twig/Twig

The infamous new template engine.

- Fast [compiled to optimised PHP]



See: <http://twig.sensiolabs.org/>

Twig/Twig

The infamous new template engine.

- **Fast** [compiled to optimised PHP]
- **Concise** [easy to learn, clean, designers will love you]

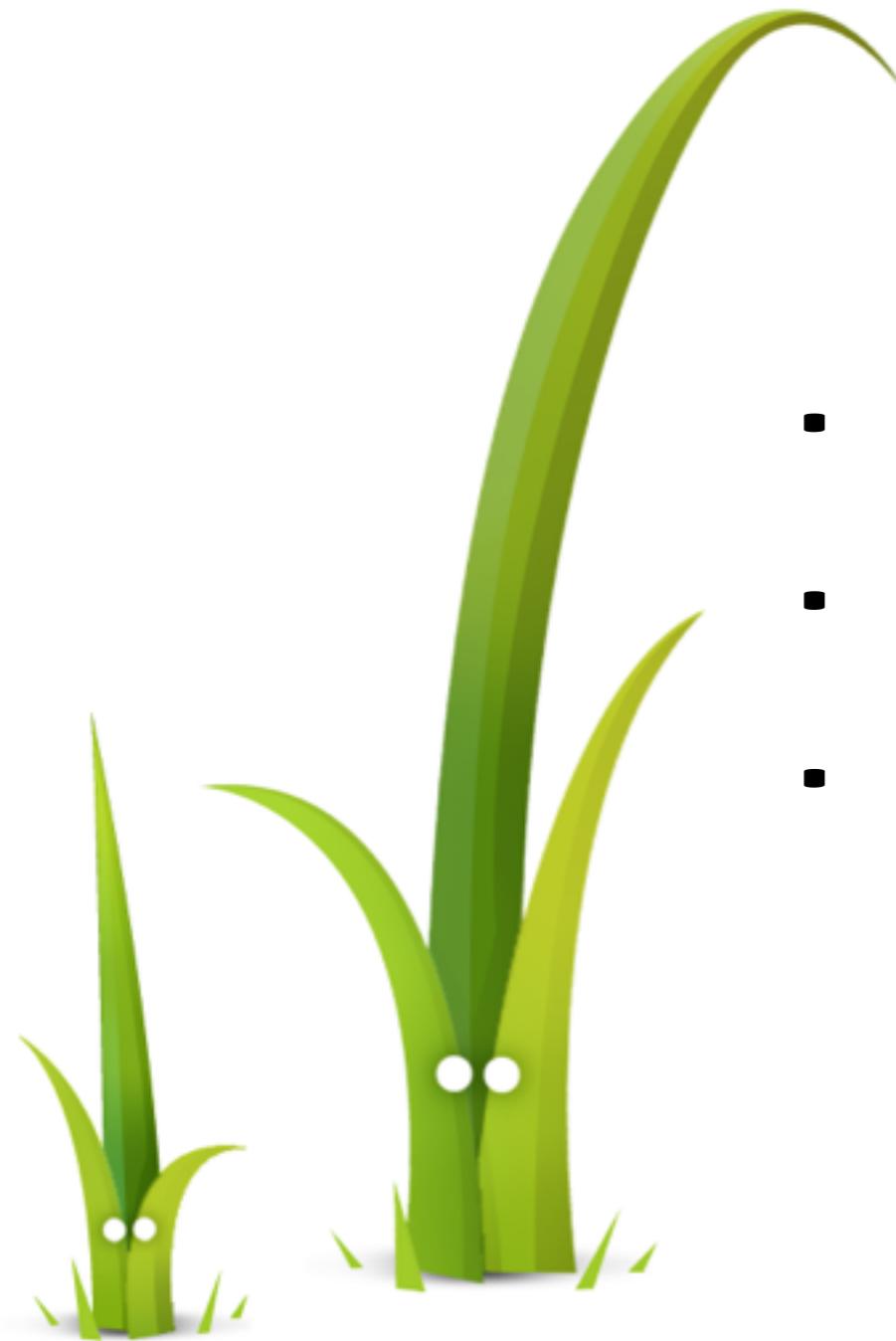


See: <http://twig.sensiolabs.org/>

Twig/Twig

The infamous new template engine.

- **Fast** [compiled to optimised PHP]
- **Concise** [easy to learn, clean, designers will love you]
- **Secure** [escaped output, sandboxed]





Twig/Twig

The infamous new template engine.

- **Fast** [compiled to optimised PHP]
- **Concise** [easy to learn, clean, designers will love you]
- **Secure** [escaped output, sandboxed]
- **Extendable** [custom tags and filters]



Twig/Twig

The infamous new template engine.

- **Fast** [compiled to optimised PHP]
- **Concise** [easy to learn, clean, designers will love you]
- **Secure** [escaped output, sandboxed]
- **Extendable** [custom tags and filters]
- **Helps enforce SoC** [helps keep templates for display only]

Side note:

Twig Panel

Friday, February 8
10:45am-11:45am

Why oh why!

Why oh why!

- Modernisation is imperative - latest and best

Why oh why!

- Modernisation is imperative - latest and best
- Requires some work - of course

Why oh why!

- Modernisation is imperative - latest and best
- Requires some work - of course
- Huge gains can be made...

Gains?

Gains?

- Move towards more standard architecture

Gains?

- Move towards more standard architecture
- Reduce the barrier to entry for new Drupal developers

Gains?

- Move towards more standard architecture
- Reduce the barrier to entry for new Drupal developers
- More focus on Drupal core

Gains?

- Move towards more standard architecture
- Reduce the barrier to entry for new Drupal developers
- More focus on Drupal core
- Allows for better integration

“A MOST IMPORTANT, BUT ALSO MOST
ELUSIVE, ASPECT OF ANY TOOL IS ITS
INFLUENCE ON THE HABITS OF THOSE WHO
TRAIN THEMSELVES IN ITS USE.”

EDSGER DIJKSTRA (COMPUTER SCIENTIST)

*Part 5:
yep, homework!*



A sample app.php

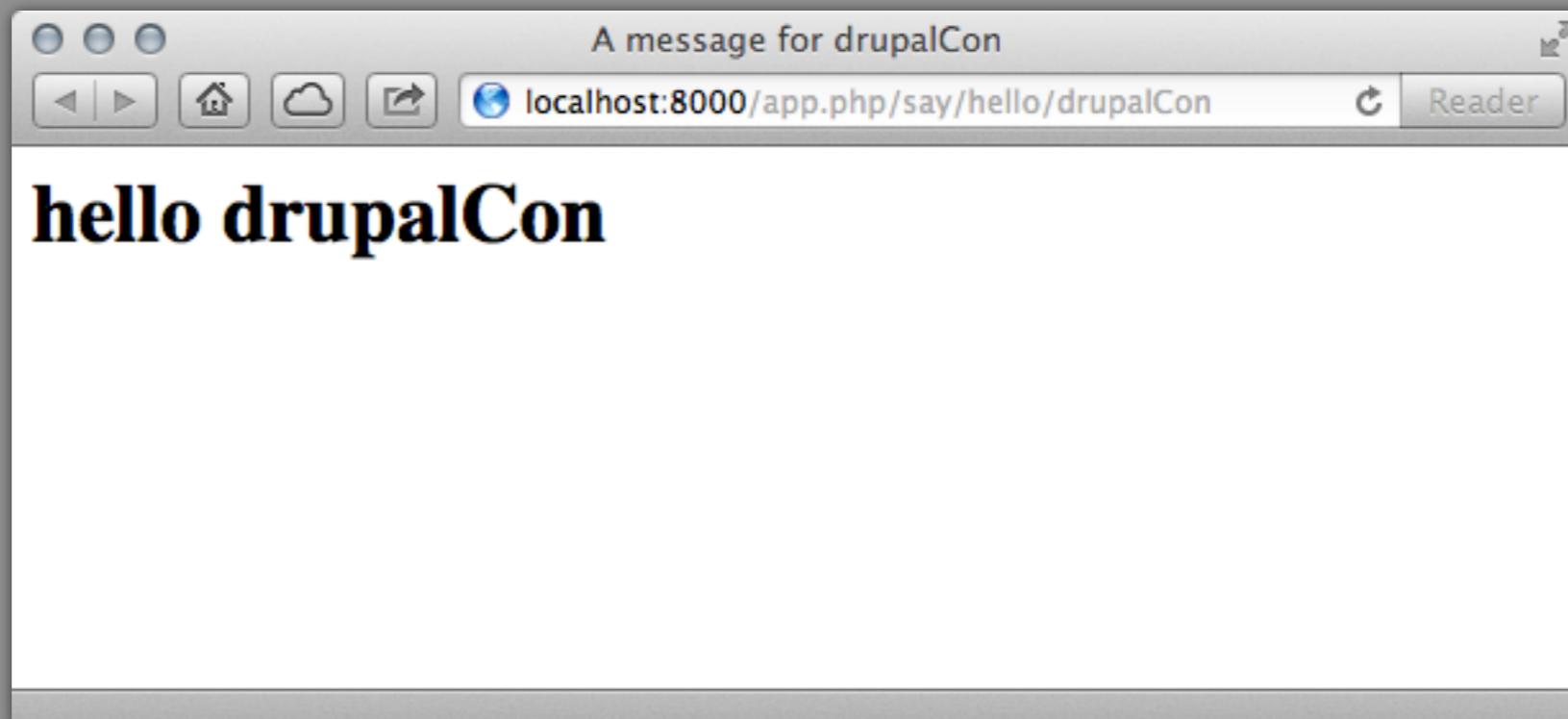
Terminal:

```
$ curl https://gist.github.com/pierswarmers/4712678/download > app.tar.gz | tar -zx  
$ cd gist4712678-*  
$ curl -s https://getcomposer.org/installer | php  
$ php ./composer.phar install  
$ php -S localhost:8000
```

A sample app.php

Terminal:

```
$ curl https://gist.github.com/pierswarmers/4712678/download > app.tar.gz | tar -zx  
$ cd gist4712678-*  
$ curl -s https://getcomposer.org/installer | php  
$ php ./composer.phar install  
$ php -S localhost:8000
```



See: <https://gist.github.com/pierswarmers/4712678> or here: <http://pierswarmers.com/2013/02/07/symfony-components/>

What next?

What next?

- Some Drush support for composer exists:
<https://drupal.org/project/composer>

What next?

- Some Drush support for composer exists:
<https://drupal.org/project/composer>
- Push for moving /core/vendor to /vendor:
<http://drupal.org/node/1805316> & <http://drupal.org/node/1475510>

What next?

- Some Drush support for composer exists:
<https://drupal.org/project/composer>
- Push for moving /core/vendor to /vendor:
<http://drupal.org/node/1805316> & <http://drupal.org/node/1475510>
- Unleash Drupal code to the non-Drupal world:
<http://drupal.org/node/1826054>

What next?

- Some Drush support for composer exists:
<https://drupal.org/project/composer>
- Push for moving /core/vendor to /vendor:
<http://drupal.org/node/1805316> & <http://drupal.org/node/1475510>
- Unleash Drupal code to the non-Drupal world:
<http://drupal.org/node/1826054>
- Start playing, write decoupled code, share like crazy
 - and turbocharge PHP!



Out of time...

CODING & DEVELOPMENT | PIERS WARMERS | FEBRUARY 7 2013

SESSION ID: 653

<http://sydney2013.drupal.org/symfony-components-travel-guide>