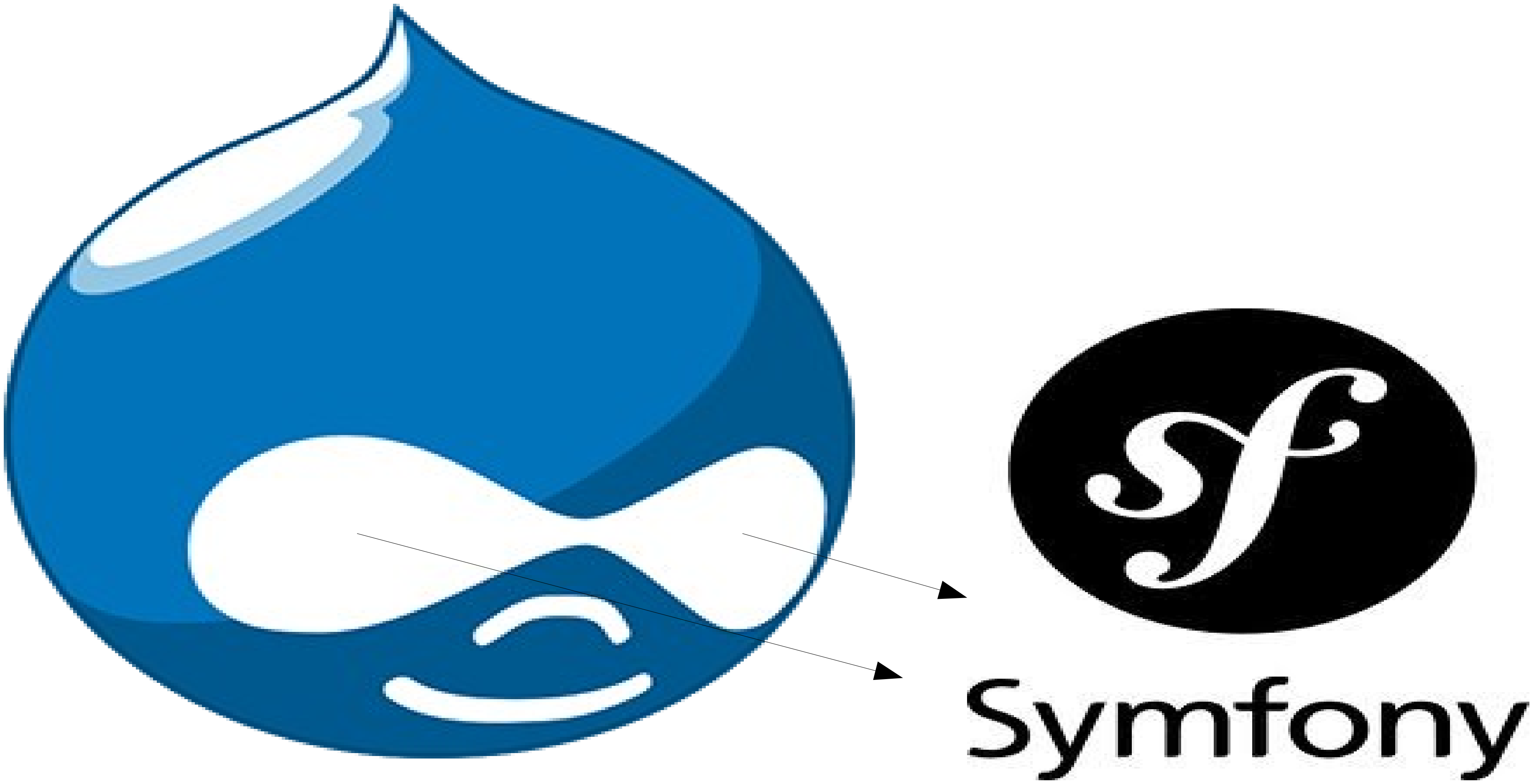
Symfony2 and Drupal



Why to talk about Symfony2 framework?

Me and why Symfony2?

- Timo-Tuomas ”Tipi / TipiT” Koivisto, M.Sc.
- Drupal experience ~6 months
- Symfony2 ~40h
- Coming from the (framework) Java world
- My boss did not let me learn Drupal ;)
- And I come from Finland...

Contents

- Situation right now
- Frameworks in general
- Symfony2 and Drupal 8 connection
- Background, general, project, terminology
- Symfony2's features
- Development demo(someone who knows)
- Summary

Symfony2 web application PHP framework

- What is a framework and why to use them?
- Examples: ZEND, PEAR,
- General features ja principles
- What frameworks can do?
- What is full stack web application PHP framework?
- Is Drupal a framework?

Frameworks

- ”A software framework is an abstraction in which software providing generic functionality can be selectively changed by user code, thus providing application specific software. It is a collection of software libraries providing a defined application programming interface (API).”
- ”A web application framework is a software framework that is designed to support the development of dynamic websites, web applications and web services. The framework aims to alleviate the overhead associated with common activities performed in Web development.”

-Wikipedia

Symfony2 & Drupal history together

- The Web Services and Context Core Initiative (WSCCI, “Whiskey”, former ”Butler”): Aims to transform Drupal to a REST server, with a CMS on top of it.
- Changes simplified:

  1. Performance

  2. Every service can communicate which each other

  3. Make developing easier
- Technical level: HTML, JSON, XML per URL etc communication, build HTTP handling object, rebuild menu/routing system, performance towards panels etc...
- Complicated? Yes. What next?
- ”Symfony2 realizes 90% what we need.” -crell

Symfony2 & Drupal future together

- How to leverage Symfony2?
- Symfony2 components will be in core:

  HttpFoundation: OO layer for handling HTTP requests, responses, etc.

  ClassLoader: For autoloading classes (PHP 5.3, PSR-0 standard)

  HttpKernel: Symfony2 core port to Drupal?
- Big changes coming... Do I need to understand all this?
- Bonus: Symfony2 community support available!

Symfony2 framework

- ”Symfony is a full-stack framework, a library of cohesive classes written in PHP.” It is full-stack MVC framework implementing REST.
- Basic RAD with MVC-architecture
- Project since 2005. Newest stable 2.0.8, 07/2011 2.0. MIT licence.
- Behind company named: Sensio (France) → Permanence
- Separation of concerns philosophy and ”Brick by brick” prinsible
- Standalone components and libraries decoupled
- Support through 'Vendors' and community (1K members, 353 contributors all together, 74 in last two monts (3.1.2011))

Symfony glossary/jargon (for Drupal Devs)

- Distribution = Release = Package of confs, bundles, enables
- Bundle = Plugin = Module = Implements a single feature. (in Drupal: ”Subtype of an entity that can be configured separately. E.g. Node types.”)
- Vendor = supplier of PHP libraries and bundles. Not usually commercial.
- Service container = (aka dependency injection container) PHP object that performs a "global" task. SOA arch. Compare Drupal ”Utilities”.
- Router = Dispatcher = Menu system = A route is a map from a URL pattern to a controller/resource
- A route = A map from a URL pattern to a controller (registered path)
- Controller = Actions: PHP function you create that takes information from the HTTP request and constructs and returns an HTTP response (as a Symfony2 Response object).
- Resources = Service configuration files
- Asset === Asset

What do we get?

- Managetable. Extensible. Fast. Controllable. Modular system.
- Preconfigured system with bundles, PHP-libraries and build in functions
- ”Tools” (commands, scripts), Eclipse/Netbeans plugin
- Bundles: DoctrineBundle, SecurityBundle, SwiftmailerBundle, WebProfilerBundle etc...
- Template engine (Twig), Doctrine ORM and other technologies
- Cache storages: Memcache, SQLite, File, EAcceleration, APC
- Testing: Lime, PHPUnit (possible)
- Context handler, Examples, ”Web Debug toolbar”, Internationalization, Netbeans & GIT support etc...

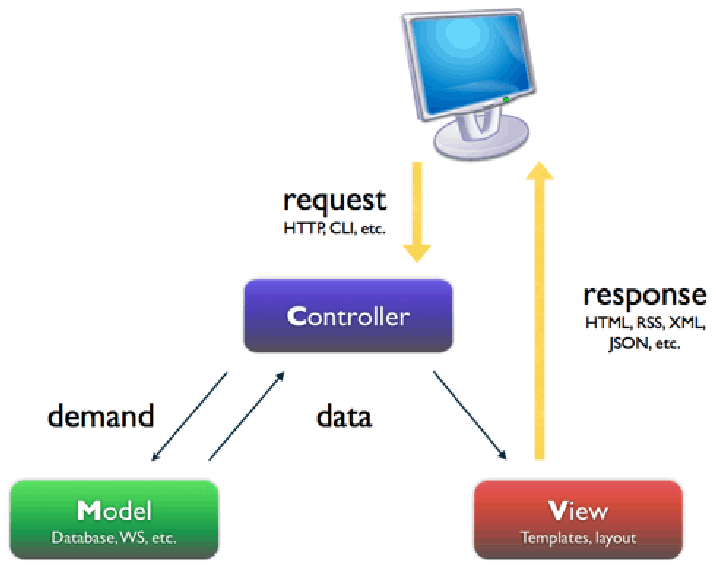
Installation

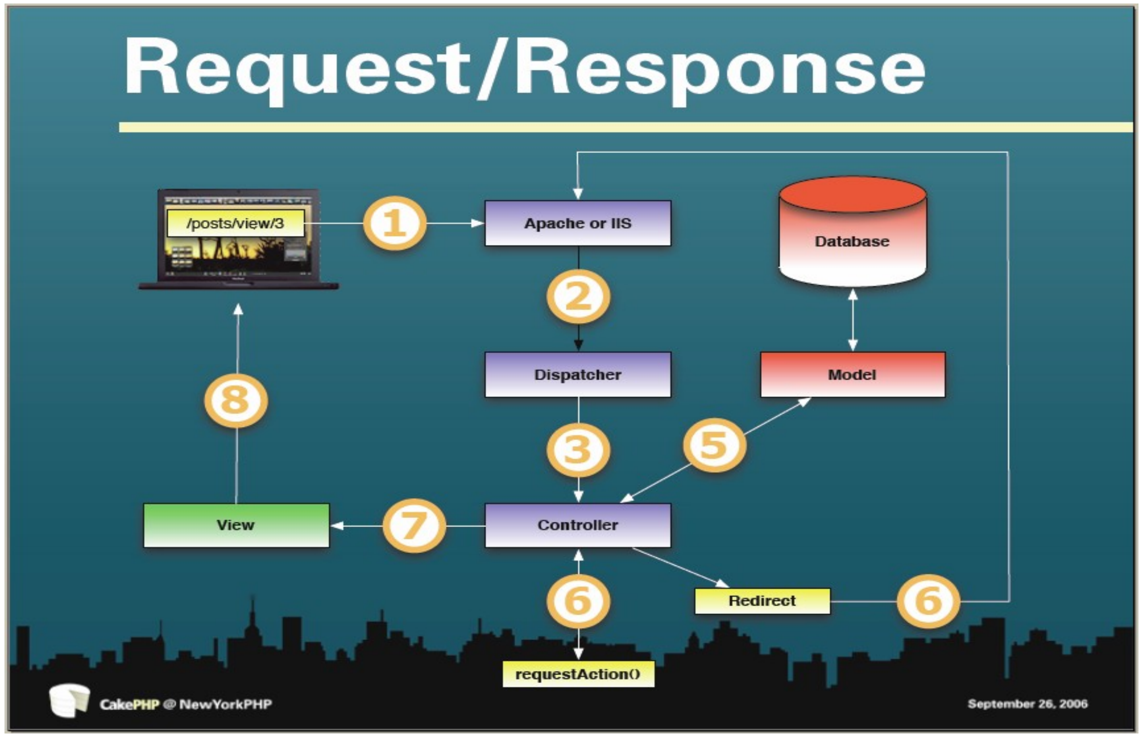
- http://symfony.com/download 2.8MB .tgz /13.3MB or 'git clone http://github.com/symfony/symfony-standard.git'
- set vHost und basic environment
- Exec: 'php bin/vendors install'
- Exec: '../symfony/ php app/check.php'
- Navigate to http://symfony.local/web/config.php

Netbeans with Symfony

- Support through plugin ver. 1.0: NetBeans 7.1 RC1 or newer
- Lacks TWIG and CLI (command-line interface) support
- Unstable Twig plugin for writing templates: http://blogsh.de/twig-for-netbeans/
- Tools → Options → PHP → Symfony script for setting up new Symfony projects
- Tools for database configuration, project initialization, etc
- Netbeans addons and shotcuts
- Eclipse users have their (better?) plugin :)

Architecture





- Controller: PHP function that takes request and returns response.
- View: Front End = UI = Twig + JS + HTML-pages
- Model: Doctrine + Database

(Default) directory structure

- app/: The application configuration
- src/: The project's PHP code
- vendor/: The third-party dependencies
- web/: The web root directory
- Flexible, use some components, integrate to other techniques...

Developing with Symfony2

- Bundles: Structure set of files that implements one feature. → Module
- Rounting: app/config/routing_dev.yml → Map from user request to virtual path / resource / what is executed.
- We write Controllers = Actions: usually a single method inside a controller object.
- Service container (”dependency injection”) → SOA arch. Speed and extensibility. No need to create an object, just use the service.
- Pages, Template engine and database handling...

Developing with Symfony2

- Dev/prod/etc. environments: (un)cached, configuration differs -> easy import to production environment
- "Web Debug Toolbar": Developer's best friend which include debugging, profiler etc.
- Command line tool: 'php app/console list'
- Good documentation about concepts: http://symfony.com/doc/current/book
- Not so good API: http://api.symfony.com/2.0/
- Demo time...

Controller example

- /**
- *Implements simple controller. The goal of a controller is always the same: create and return a Response object.
- */
-
- namespace Acme\HelloBundle\Controller;
- use Symfony\Component\HttpFoundation\Response; //use Response service (service object) class / library
-
- class HelloController {
-     public function indexAction($name) {
    - //Here use services and do what ever.
    - $router = $this->get('router'); //easy service access. Or $mailer or your own defined.
-         return new Response('<html><body>Hello '.$name.'!</body></html>');
-     }
- }

Bundles

```
class AppKernel extends Kernel {

public function registerBundles()  {

$bundles = array(

  new Symfony\Bundle\FrameworkBundle\FrameworkBundle(),

  new Symfony\Bundle\TwigBundle\TwigBundle(),

  new Symfony\Bundle\SwiftmailerBundle\SwiftmailerBundle(),

  new JMS\SecurityExtraBundle\JMSSecurityExtraBundle(),

  );
```

-
  ```
  if (in_array($this->getEnvironment(), array('dev', 'test'))) {
  ```

-
  ```
    $bundles[] = new Acme\DemoBundle\AcmeDemoBundle(); //our bundle, program, module, application, plugin, service
  ```

-
  ```
    $bundles[] = new Symfony\Bundle\WebProfilerBundle\WebProfilerBundle();
  ```

-
  ```
    $bundles[] = new Sensio\Bundle\GeneratorBundle\SensioGeneratorBundle();
  ```

-
  ```
    }
  ```

-
  ```
    return $bundles; //can be configured via YAML, XML, PHP -files in app/config/config.yml
  ```

-
  ```
    }
  ```

-
  ```
  }
  ```

Drupal 8 and Symfony2

- Drupal 8 includes: ClassLoader and HttpFoundation (maybe HttpKernel) libraries to core
- These makes integration between two ”frameworks” easy → extend Drupal faster and better
- MVC vs. Procedural event/listener driven collision?
- Twig: Layout template defined as 'blocks', inheritance. → Pages build more like Panels

Where to start

- Get the big picture: http://symfony.com/doc/current/quick_tour/the_big_picture.htr
- Netbeans & Symfony: http://netbeans.org/kb/docs/php/symfony-screencast.html
- Book from core team: http://symfony.com/doc/current/book/index.html

Summary

- Symfony2: Full stack web application framework written in PHP
- Why to reinvent the wheel?
- Drupal 8 core: Includes atleast two components from Symfony2.
- Symfony2 is a similar modular framework as Drupal. Flexible, managetable and extensible.
- Concept sililarities: front controller, mapping, templates. ”Easy” to learn for a Drupal Dev.
- Differents: terminology, architecture, use of annotations.
- How to leverage Symfony2 with Drupal in the future? Ask some smarter guy. :)

Got interested?

- suns' blog post: http://www.unleashedmind.com/en/blog/sun/drupal-8-the-path-forward
- 'crell' road map:
- http://www.garfieldtech.com/blog/refocusing-wscci
- 'crell' about Drupal 8 and sscci project: http://modulesunraveled.com/podcast/009-larry-garfield-and-the-future-o
- Other blog posts will be released soon... → drupal.org/planet
- Symfony project: http://symfony.com/
- Get the big picture: http://symfony.com/doc/current/quick_tour/the_big_picture.htr

P.S. Project needs SKILLFUL people...

Thank you!

- Questions?

Tipi Koivisto, @TipiT

tipi@koivisto.eu