

Szász Zoltán

@rkon



Symfony 2

@webkonf 2012

zoltan@szasz.me
@szaszz





Symfony at a Glance

Symfony is a PHP Web Development Framework.

That doesn't answer your question?

Ok, let's try that again.

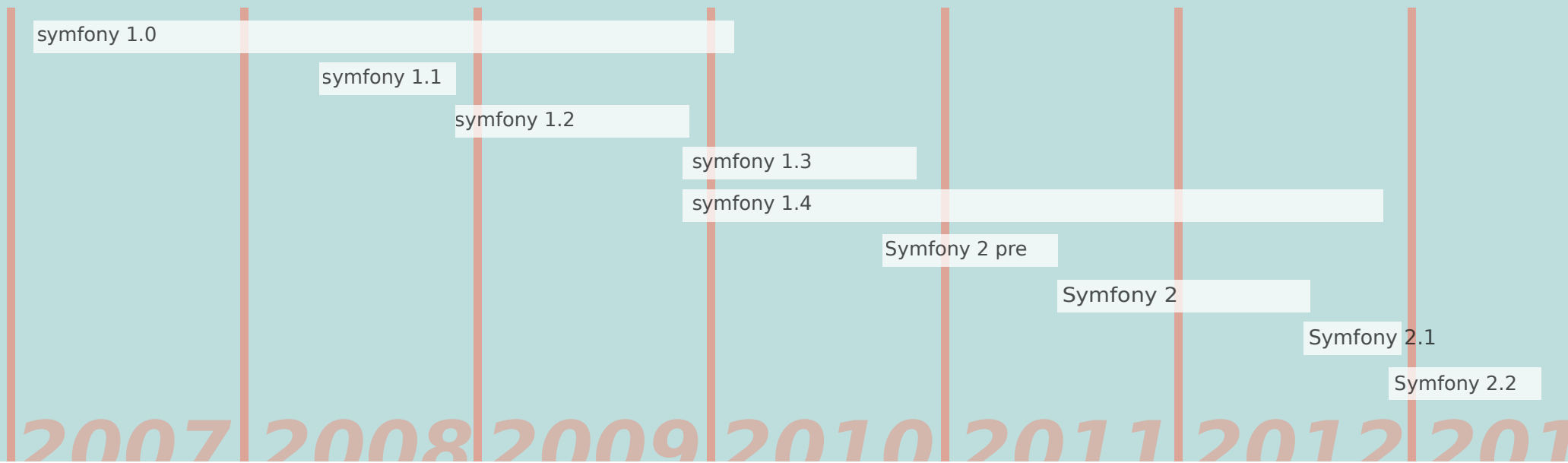
Symfony is a PHP Framework, a Philosophy, and a Community - all working together in harmony.



Symfony 2 - múlt



- symfony 1.4 - 2009. november
LTS - 3 év támogatás (v1.4 2012 novemberéig)
- Symfony 2.0 PR3 - 2010. szeptember
2.0 BETA 1 - 2011. április
2.0 RC 1 - 2011. június
2.0 - 2011. július
2.0.15 - 2012. május
2.1 - 2012. szeptember 11 (!)
2.1.2 - 2012. szeptember 20



Symfony 2 - jövő



- aktívan fejlesztik
- jó a dokumentáció, és egyre jobb lesz.
- törekvés a lehetséges legjobb kompatibilitásra, de lesznek visszafelé nem kompatibilis részek
- átláthatóság: pontos changelog + git log
- könnyen telepíthető (csomagból, githubról, vagy akár PHP Composerrel is).
- Várható megjelenések a "roadmap" szerint:

Symfony ~~2.1 (2012 szeptember)~~

2.2 (2013 február)

2.3 (2013 június) **Az első LTS release!**

Symfony 2 - mint framework



- Nem MVC framework, de lehet azt is csinálni belőle
- Nem olyan értelemben framework, mint az 1.x volt
(monolitikus VS moduláris)
- Nem csupán HTTP keretrendszer
- Legjobb módszerek a PHP világon kívülről (inspiráció)
Ruby on Rails, Django, Java, Spring
- A kód-újrafelhasználhatóság a legkiemelkedőbb erény
- DI
- Bundle
- Minden Bundle

Symfony 2 - mint framework



- Rugalmas konfiguráció (PHP || XML || YAML)
 - Konfig alapján statikus kód build
 - Annotációk használata (bár nem nyelvi elem)
 - 21 független, önállóan is használható komponens (git submodule, PHP composer, Pear)
-
- | | | |
|-----------------------|-------------------|---------------|
| • BrowserKit | • EventDispatcher | • Routing |
| • ClassLoader | • Finder | • Security |
| • Config | • Form | • Serializer |
| • Console | • HttpFoundation | • Templating |
| • CssSelector | • HttpKernel | • Translation |
| • DependencyInjection | • Locale | • Validator |
| • DomCrawler | • Process | • Yaml |

Symfony 2 - komponensek



- **Dependency Injection:** Az sf2 legmeghatározóbb része. Az alkalmazás a DI konténeren keresztül kap meg minden szükséges függőséget. A konfiguráció itt manifesztálódik.
- **Console:** CLI alkalmazások készítéséhez (taskok, batchek, background jobok, daemonok, stb). CLI paraméterek kezelése, validálása, stdout írása, dialogusablakok, stb)
- **EventDispatcher:** OO Observer pattern (fenttartható kódok :)). Események eldobásával értesíthetünk az adott csatornára feliratkozott más komponenseket. Ekkor a kód futtatása a megfigyelőben folytatódik. Így lehet pl: asszinkron műveleteket is végrehajtani.

Symfony 2 - komponensek



- **HttpFoundation & HttpKernel:** A teljes HTTP életciklus OOP kezelése a requesttől a responseig. Felelős még a session managementért is.
- **Finder:** OOP interfészen keresztül lehet fileokat és könyvtárakat listázni, keresni és filterezni.
- **YAML:** Nyelv független emberbarát leírónyelv. Yml fileokból csinál PHP tömböt és vice versa.
- **Locale & Translator:** i18n és l10n támogatás (ha nem elérhető a PHP intl modul. (inkább legyen))
- **Validator:** A neve mindent elmond

Symfony 2 - ORM



- Nincs dedikált ORM réteg, választható: Doctrine 2 (a kezdetektől) vagy Propel 1.6 (2012. április)
- Doctrine 2 nem pusztán ORM, hanem DBAL & ODM is
- MySQL, PostgreSQL, SQLite, MongoDB, CouchDB, PHPCR (Jackrabbit)
- Nem Active Record szemlélet, az ORM osztályok "sima" PHP osztályok
- Nincs "gyári" CRUD / scaffolding
- A megszokott Doctrine lehetőségek elérhetők: entity generálás (getter/setter), SQL stb.
- A jól bevált DQL megmaradt egy kis "extra" namespace-ekkel

Symfony 2 - ORM



```
1 <?php
2 namespace DiBlo\BookmarkBundle\Entity;
3
4 use Doctrine\ORM\Mapping as ORM;
5
6 /**
7  * @ORM\Entity
8  * @ORM\Table(name="bookmarks")
9  */
10 class Bookmark
11 {
12     /**
13      * @ORM\Id
14      * @ORM\Column(type="integer")
15      * @ORM\GeneratedValue(strategy="AUTO")
16      */
17     protected $id;
18
19     /**
20      * @ORM\Column(type="string", length=255)
21      */
22     protected $url;
23
24     /**
25      * @ORM\Column(type="string", length=255)
26      */
27     protected $title;
28 }
```

```
1 DiBlo\BookmarkBundle\Entity:
2   type: entity
3   table: bookmarks
4   id:
5     id:
6       type: integer
7       generator: { strategy: AUTO }
8   fields:
9     url:
10      type: string
11      length: 255
12     title:
13      type: string
14      length: 255
```

```
$query = $em->createQuery('SELECT url FROM DiBlo\BookmarkBundle\Entity\Bookmark u WHERE id = 20');
$bookmark = $query->getResult();
```

Symfony 2 - Controller



- Semmi extra, de azt nagyon jól csinálja
Request » Feldolgozás » **Response**
- Átírányítások, forwardok, template renderelés
- Paraméterek a routingból

```
# /src/DiBlo/BookmarkBundle/Resources/config/routing.yml
```

```
edit:
```

```
  pattern:      /konyvjeRlzo/szerkesztes/{id}
  defaults:    { _controller: DiBloBookmarkBundle:BookmarkAdmin:edit }
  requirements: { id:'[0-9]{1,6}' }
```

- A routingok beágyazhatók egymásba:

```
# /app/config/routing.yml
```

```
UserBundle:
```

```
  resource: "@DiBlo/UserBundle/Resources/config/routing.yml"
  prefix:   /user
```

```
# /src/DiBlo/UserBundle/Resources/config/routing.yml
```

```
fos_user_profile:
```

```
  resource: "@FOSUserBundle/Resources/config/routing/profile.xml"
  prefix:   /profile
```

Symfony 2 - Controller



```
1 <?php
2 namespace DiBlo\BookmarkBundle\Controller;
3 use Symfony\Bundle\FrameworkBundle\Controller\Controller;
4 use Symfony\Component\HttpFoundation\Request;
5 use DiBlo\BookmarkBundle\Entity\Bookmark;
6
7 class DefaultController extends Controller
8 {
9     public function editAction(Request $request, $id)
10    {
11        $bookmark = $this->getDoctrine()->getRepository('DiBloBookmarkBundle:Bookmark')->find($id);
12
13        $form = $this->createFormBuilder($bookmark)
14            ->add('url', 'url')
15            ->add('title', 'text')
16            ->getForm();
17
18        if ($request->getMethod() == 'POST')
19        {
20            $form->bindRequest($request);
21
22            if ($form->isValid())
23            {
24                $em = $this->getDoctrine()->getEntityManager();
25                $em->persist($bookmark);
26                $em->flush();
27
28                return $this->redirect($this->generateUrl('bookmark_save_success'));
29            }
30        }
31
32        return $this->render('DiBloBundle:Default:edit.html.twig', array(
33            'form' => $form->createView(),
34        ));
35    }
```

Symfony 2 - View



- Twig és PHP Engine. A Twig kód is PHP-vá fordul
- Emiatt nincs számottevő teljesítménykülönbség
A lefordított kód cache-elődik.
- Twig jellemzői:
 - OOP szemléletű sablonok
 - Symfony 2 alapértelmezett
 - jól olvasható, tömör, a sablon és a HTML jól elválik
 - könnyen tanulható
 - aktívan fejlesztik
 - automatikus kimenet escape-elés
 - vendorok is Twiget használnak, de facto szabvány
 - újrafelhasználhatóság
 - nincs PHP » nem kerülhet üzleti logika a sablonba

Symfony 2 - View



- A sablonban: `<div>{{item.value}}</div>`
Lehetséges adatszerkezetek:

```
$item['value']  
$item->value  
$item->value()  
$item->getValue()
```

- A Twiget ezeket mind helyesen tudja megjeleníteni.
A cache-elt PHP kód már a megfelelő formát fogja tartalmazni.
- Helperek: `<div>`
 `{{ datetime | formatDateFullTimeShort }}`
`</div>`

Symfony 2 - View



```
<form action="{{ path('bookmark_edit') }}" method="post">
    {{ form_errors(form) }}

    {{ form_row(form.url) }}
    {{ form_row(form.title) }}

    {{ form_rest(form) }}
    <input type="submit" />
</form>
```

Symfony 2 - Form



- Különálló komponens - egy a huszonegyből
- Form builderrel rakatjuk össze az "ORM" entitásból
- Transformerek: form » entitás && entitás » form átalakítás
- Rengeteg készen kapott és tetszőleges form elem
 - text
 - textarea
 - email
 - integer
 - money
 - url
 - number
 - password
 - percent
 - search
 - choice
 - entity
 - country
 - language
 - locale
 - timezone
 - date
 - birthday
 - datetime
 - time
 - checkbox
 - file
 - radio
 - csrf
 - repeated
 - collection
 - hidden

Symfony 2 - Validáció



- A szemantikai, üzleti logikai ellenőrzést ugyanaz a közös komponens tudja végezni
- Egyéni Constraint és Validator
- A Constraint "felparaméterezi" a validátort, ami elvégzi az ellenőrzést. Hiba esetén jelez és hozzáadja pl. a form megfelelő mezőjéhez a hibát
 - NotBlank
 - Blank
 - NotNull
 - Null
 - True
 - False
 - Type
 - Email
 - MinLength
 - MaxLength
 - Url
 - Regex
 - Ip
 - Max
 - Min
 - Date
 - DateTime
 - Time
 - Choice
 - Collection
 - UniqueEntity
 - Language
 - Locale
 - Country
 - File
 - Image
 - Callback

Symfony 2 - Tesztelés



- PHPUnit (min: 3.6.4): unit és integrációs tesztek, a megszokott módon, de extra namespace-ekkel

```
1 <?php
2 namespace DiBlo\BookmarkBundle\Tests\Bookmark;
3 use DiBlo\BookmarkBundle\Manager\Bookmark;
4
5 class ManagerTests extends \PHPUnit_Framework_TestCase
6 {
7     public function testGetEntity()
8     {
9         $bookmark = $this->getEntityById(1);
10
11         $this->assertEquals('1', $bookmark->getId());
12         $this->assertEquals('http://example.com', $bookmark->getUrl());
13     }
14 }
```

- Funkcionális tesztek (WebTestCase): Mintha a böngészőt nyomogatnánk. Lefedi az alkalmazás több rétegét egészen a routingtól, managereken keresztül a controlleren át a viewig.
request » test response » fill form » test response » ...

Symfony 2 - Tesztelés



- BDD (Behavior Driven Development): Behat.org
 - Behat: BDD framework
 - Mink: funkcionális (acceptance) test framework
 - A tesztek több eszközzel (driverrel) futtathatók: (GoutteDriver, SeleniumDriver, stb)
 - Gerkhin: a nyelv, a nyelvtan, amiben a sztorik és ezáltal a tesztek íródnak

Symfony 2 - Bundle



- **Assetic:** Hatékony Javascript és CSS kiszolgálásához
- **SonataAdminBundle:** A Symfony 1-ben megszokott admin generátoros "jóságokat" biztosítja az sf2-ben. Jelenleg csak Doctrine alapú generálás a stabil, de pre-alfa állapotban van a MongoDB és a PHP CR is.
- **SonataMediaBundle:** Különböző médiák kezeléséhez (képek, fileok, videók).
- **StofDoctrineExtensionsBundle:** Doctrine bővítmények:
 - Timestampable
 - Loggable
 - Sortable
 - Softdeletable
 - Tree processing (nested set)
 - Translatable & Translator
 - Sluggable

Symfony 2 - Bundle



- **FOSUserBundle:** Alapszintű felhasználó kezelés. Regisztráció (email megerősítéssel is), password reset. Jelenleg Doctrine, Propel, MongoDB, CouchDB támogatás.
- **FOSFacebookBundle:** Facebook PHP és Js SDK integráció. A felhasználókat az FB profiljukkal azonosíthatjuk.
- **FOSTwitterBundle:** A felhasználókat a Twitter profiljukkal tudjuk autentikálni.
- **FOSOAuthServerBundle:** OAuth2 autentikációs szerver.
- **JMSSerializerBundle:** Objektumok szerializálása (és deszerializálása) JSON, XML, vagy YAML formátumba.
- **JMSI18nRoutingBundle:** I18n alapú routing.

Symfony 2 - Bundle



- **KnplMenuBundle:** OOP menü interfész.
- **KnpsnappyBundle:** PDF, vagy kép generálása az általunk használt Twig/HTML templatekből.
- **LiipImageBundle:** Egyszerű kép manipulálás, thumbnail, átméretezés, kivágás készítés.
Backendek: GD, Imagick, Gmagick.
- **LiipThemeBundle:** Egyedileg konfigurálható layoutok
- **GoogleBundle:** Google Analytics, Adwords, Statikus térkép támogatás alkalmazásainkban.
Jó GA támogatás: Async beillesztés, multi-tracker, custom page view , vásárlás követés.

Symfony 2 - Linkek



- rövid bevezető: http://symfony.com/doc/current/quick_tour/
- teljes dokumentáció: <http://symfony.com/doc/current/book/>
- tippek és trükkök: <http://symfony.com/doc/current/cookbook/>
- letöltés: <http://symfony.com/download>
- a legnagyobb bundle gyűjtemény: <http://knpbundles.com/>
- Symfony2 roadmap: <http://symfony.com/blog/towards-symfony-2-1>
- Twig sablon motor: <http://twig.sensiolabs.org/>
- BDD framework: <http://behat.org/>
- BDD: <http://www.slideshare.net/everzet/bdd-in-symfony2>
- demók: <http://ipsum.knplabs.org/>

Szász Zoltán

@rkon



Fin

kérdések?

zoltan@szasz.me
@szaszz