Retour D'experience sur Symfony

Djambazian Nicolas

QU'EST-CE QUE SYMFONY ?

C'est du PHP

C'est du PHP

Mois
- Vieux
- Pas de deamon
- Verbeux

Plus
- Vieux
- Orienté Objet
- Non typé
- Reflection
- Namespace + Autoload

Symfony, c'est des tas de trucs

Symfony, c'est des tas de trucs

- MVC
- Services avec injection de dépendances
- Event Listener
- Un gestionnaire de packet (composer)
- Une tonne de Bundle

- ORM puissant
- Système de routing
- Oganisation en Bundles
- Des tests fonctionnels (et unitaires)
- Des tas de Design pattern

Tout ce qui faut pour faire du code propre

Sans obligation bien sur

```php
foreach ($session->getVid4() as $vidCours)
{
    if (!empty($vidCours) && is_numeric($vidCours->getId()) && $vidCours->
        getName() != '')
    {
        $vid = $em->getRepository('CD\ConfigBundle\Entity\Video')->findOneById(
            $vidCours->getId());
        $vid->setName($vidCours->getName());
        $vid->setDescription($vidCours->getDescription());
        $em->flush();
    }
}
/**
 * Ah ah lol! You don't know why?? Really? Just look around you
 * and see the number of line in this action.. See all this
 * copy-pasted code.. Everything is wrong here. In fact it's a
 * miracle that everything else works properly >_<
 */
// a decommenter pour pouvoir modifier les informations de la video de sessions,
//     MAIS ce code bug, et je ne sais pas pourquoi
// _____
//     _____
/*if (!empty($session->getVidSess()) && is_numeric($session->getVidSess()->getId
    ()) && $session->getVidSess()->getName()!='') {
    $vid=$em->getRepository('CD\ConfigBundle\Entity\Video')->findOneById($session
        ->getVidSess()->getId());
    $vid->setName($session->getVidSess()->getName());
    $vid->setDescription($session->getVidSess()->getDescription());
    $em->flush();
}*/
```

Sans obligation bien sur

```php
foreach ($session->getVid4() as $vidCours)
{
    if (!empty($vidCours) && is_numeric($vidCours->getId()) && $vidCours->
        getName() != '')
    {
        $vid = $em->getRepository('CD\ConfigBundle\Entity\Video')->findOneById(
            $vidCours->getId());
        $vid->setName($vidCours->getName());
        $vid->setDescription($vidCours->getDescription());
        $em->flush();
    }
}
/**
 * Ah ah lol! You do                              you
 * and see the numbe
 * copy-pasted code.                              a
 * miracle that ever
 */
// a decommenter pou                              video de sessions,
//       MAIS ce code bu
// ─────────────────────────────────────────────────────────

/*if (!empty($session                            getVidSess()->getId
    ()) && $session                              video de sessions
  $vid=$em->getRepos                             ndOneById($session
        ->getVidSess(
  $vid->setName($ses
  $vid->setDescription($session->getVidSess()->getDescription());
  $em->flush();
}*/
```

UNE API REST EN SYMFONY

Pas (encore ?) de génération automatique

Des Bundles, encore des bundles

- FosRestBundle
- ApiDocBundle
- JMSSerializerBudle
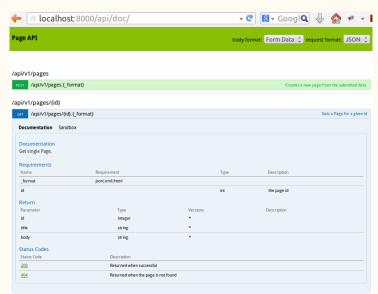
FosRestBundle

```php
class UsersController
{
    public function getUsersAction()
    {} // "get_users"        [GET] /users

    public function postUsersAction()
    {} // "post_users"       [POST] /users

    public function getUserAction($slug)
    {} // "get_user"         [GET] /users/{slug}

    public function putUserAction($slug)
    {} // "put_user"         [PUT] /users/{slug}

    public function patchUserAction($slug)
    {} // "patch_user"       [PATCH] /users/{slug}

     public function removeUserAction($slug)
    {} // "remove_user"      [GET] /users/{slug}/remove

    public function deleteUserAction($slug)
    {} // "delete_user"      [DELETE] /users/{slug}

    public function getUserCommentAction($slug, $id)
    {} // "get_user_comment"       [GET] /users/{slug}/comments/{id}

    public function editUserCommentAction($slug, $id)
    {} // "edit_user_comment"      [GET] /users/{slug}/comments/{id}/edit
}
```

ApiDocBundle

```
/**
 * @ApiDoc(
 *   resource = true,
 *   description = "Creates a new page from the submitted data.",
 *   input = "Acme\BlogBundle\Form\PageType",
 *   statusCodes = {
 *     200 = "Returned when successful",
 *     400 = "Returned when the form has errors"
 *   }
 * )
 */
public function postPageAction
```

ApiDocBundle

JMSSerializerBudle

```php
/**
 * @Serializer\ExclusionPolicy("all")
 */
class Order
{
    /**
     * @var int
     * @Serializer\Type("integer")
     * @Serializer\Expose
     */
    private $id;

    /**
     * @var Product[]
     * @Serializer\Type("array<Product>")
     * @Serializer\Expose
     */
    private $products;

    /**
     * @var float
     * @Serializer\Type("float")
     * @Serializer\Expose
     */
    private $total;

    // ...
}
```

Et enfin, les formulaires Symfony

```php
class UserAccountType extends AbstractType
{
    public function buildForm(FormBuilderInterface $builder, array $options)
    {
        $builder
            ->add('firstName')
            ->add('lastName')
            ->add('email')
            ->add('phone')
        ;
    }
}

class UserController extends FOSRestController
{
    public function postAction(Request $request, User $user)
    {
        $form = $this->createForm(new UserAccountType(), $user);
        $form->handleRequest($request);
        if ($form->isValid()) {
            // save $user
            return $this->view($user, Response::HTTP_OK);
        }
        return $this->view($form, Response::HTTP_BAD_REQUEST);
    }
}
```

Magique !

Magique !

Mais pas trop fait pour de l'API

BILAN

Lourd ! Mais puissant, bien architecturé, scalable, complet, …

Ma Reco : Pour les gros projet