

# Retour d'experience sur Symfony

Djambazian Nicolas



QU'EST-CE QUE SYMFONY ?

# C'est du PHP



# C'est du PHP

## Mois

- Vieux
- Pas de daemon
- Verbeux

## Plus

- Vieux
- Orienté Objet
- Non typé
- Reflection
- Namespace et Autoload

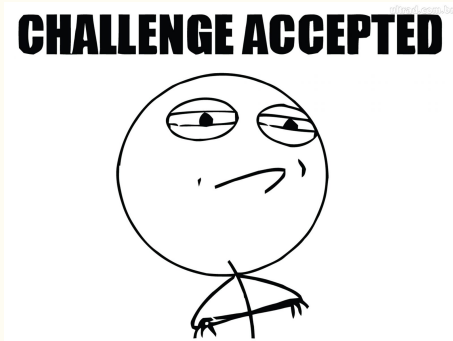
# Symfony, c'est des tas de trucs



# Symfony, c'est des tas de trucs

- MVC
- Services avec injection de dépendances
- Event Listener
- Un gestionnaire de packet (composer)
- Une tonne de Bundles
- ORM puissant
- Système de routing
- Organisation en Bundles
- Des tests fonctionnels (et unitaires)
- Des tas de Design patterns

# Tout ce qu'il faut pour faire du code propre



# Sans obligation bien sûr

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





# Sans obligation bien sûr

```
foreach ($session->getVid4() as $vidCours)
{
    if (!empty($vidCours) && is_numeric($vidCours->getId()) && $vidCours->
        getName() != '')
    {
        $vid = $sem->getRepository('CD\ConfigBundle\Entity\Video')->findOneById(
            $vidCours->getId());
        $vid->setName($vidCours->getName());
        $vid->setDescription($vidCours->getDescription());
        $sem->flush();
    }
}

/**
 * Ah ah lol! You do
 * and see the number
 * copy-pasted code.
 * miracle that ever
 */
// a commenter pour
// MAIS ce code b

/* if (!empty($session
    )) && $session-
    $vid=$sem->getRepos
    ->getVidSess(
    $vid->setName($ses
    $vid->setDescription($session->getVidSess()->getDescription());
    $sem->flush();
*/
```



UNE API REST EN SYMFONY

# Pas (encore ?) de génération automatique



# Des Bundles, encore des bundles

- FosRestBundle
- ApiDocBundle
- JMSSerializerBuddle

# FosRestBundle

```
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
}
```



```
/**
 * @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
```

localhost:8000/api/doc/

Page API body format: Form Data request format: JSON

**/api/v1/pages**  
POST /api/v1/pages.{\_format} Creates a new page from the submitted data.

**/api/v1/pages/{id}**  
GET /api/v1/pages/{id}.{\_format} Gets a Page for a given id

**Documentation** Sandbox

Documentation  
Get single Page.

**Requirements**

Name	Requirement	Type	Description
_format	json xml html		
id		int	the page id

**Return**

Parameter	Type	Versions	Description
id	integer	*	
title	string	*	
body	string	*	

**Status Codes**

Status Code	Description
200	Returned when successful
404	Returned when the page is not found

# JMSSerializerBuddle

```
/**
 * @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

```
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 !



Mais pas trop fait pour de l'API



BILAN

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



# Ma Reco : Pour les gros projets

