Projet **FW**

TP1 : Création d'un Bundle

1 - Exemple traité

Nous allons prendre comme application un logiciel de gestion de projets d'entreprise :

• chaque département (RH, Comptabilité, Informatique, …) gère des projets spécifiques (« développement des compétences managériales » pour le département RH, « norme Bâle III » pour le département Comptabilité, « site E-commerce » et « Intranet mobile » pour le département Informatique…).

• Les employés travaillent sur des projets.

2 - Création d'un bundle avec le générateur

Configuration SSH dans Mobaxterm pour accéder à WebDev

- Hote : webdev.iut-blagnac.fr
- User name : votre login habituel en sa
- lle TP (de la forme : pmartin)
- Port : 22

Vous allez créer un bundle appelé **Projet** en utilisant le générateur de Bundle de Symfony (création automatique de plusieurs dossiers et fichiers par défaut).

1. Connectez-vous au serveur WebDev avec l'outil MobaXterm.
   Votre mot de passe est celui habituel dans les salles TP (**Attention à ne pas autoriser Mobaxterm à réaliser la sauvegarde automatique de votre mot de passe sur les PC des salles TP !**)
2. Allez dans le répertoire **public_html\symfony** en utilisant la commande **cd**.
3. Taper la commande : **php app/console**.

Apparaît alors une liste de commandes associées à Symfony2. C'est grâce à l'une de ces commandes que nous allons générer notre premier bundle.
Entrez la commande : **php app/console generate:bundle**

Le générateur vous demande alors de renseigner plusieurs options (suivez le modèle de la copie d'écran ci-dessous) :

```
lnonne@webdev public_html/symfony$ php app/console generate:bundle

  Welcome to the Symfony2 bundle generator

Your application code must be written in bundles. This command helps
you generate them easily.

Each bundle is hosted under a namespace (like Acme/Bundle/BlogBundle).
The namespace should begin with a "vendor" name like your company name, your
project name, or your client name, followed by one or more optional category
sub-namespaces, and it should end with the bundle name itself
(which must have Bundle as a suffix).

See http://symfony.com/doc/current/cookbook/bundles/best_practices.html#index-1
for more
details on bundle naming conventions.

Use / instead of \  for the namespace delimiter to avoid any problem.

Bundle namespace: ProjetBundle

The namespace sometimes contain a vendor namespace (e.g. VendorName/BlogBundle i
nstead of simply ProjetBundle).
If you've *did* type a vendor namespace, try using a forward slash / (Acme/BlogB
undle)?

Keep ProjetBundle as the bundle namespace (choose no to try again)? [yes]:

In your code, a bundle is often referenced by its name. It can be the
concatenation of all namespace parts but it's really up to you to come
up with a unique name (a good practice is to start with the vendor name).
Based on the namespace, we suggest ProjetBundle.

Bundle name [ProjetBundle]:
```

```
The bundle can be generated anywhere. The suggested default directory uses
the standard conventions.

Target directory [/home/lnonne/public_html/symfony/src]:

Determine the format to use for the generated configuration.

Configuration format (yml, xml, php, or annotation): yml

To help you get started faster, the command can generate some
code snippets for you.

Do you want to generate the whole directory structure [no]? yes

  Summary before generation

You are going to generate a "ProjetBundle\ProjetBundle" bundle
in "/home/lnonne/public_html/symfony/src/" using the "yml" format.

Do you confirm generation [yes]?

  Bundle generation

Generating the bundle code: OK
Checking that the bundle is autoloaded: OK
Confirm automatic update of your Kernel [yes]?
Enabling the bundle inside the Kernel: OK
Confirm automatic update of the Routing [yes]?
Importing the bundle routing resource: OK

  You can now start using the generated code!
```

Une fois la génération terminée, vous devriez trouver un répertoire **ProjetBundle** dans le dossier **/home/VotreCompte/public_html/symfony/src/**

Le bundle **Projet** a donc été créé. Vous noterez qu'il est nécessaire d'avoir le suffixe **Bundle** à la fin du nom de notre bundle **ProjetBundle**.

3 - Lien entre le Bundle Projet et Symfony2

Grace à cette commande, Symfony2 a été informé de la création de ce nouveau bundle au cours de sa génération. Le générateur a mis à jour le noyau (kernel) de Symfony2.

• Ouvrez le fichier **symfony/app/AppKernel.php**.
• Vous devriez voir le code :

```
…
new Sensio\Bundle\FrameworkExtraBundle\.....,
…
new ProjetBundle\ProjetBundle(),
```

Cette ligne vous permettra d'utiliser au sein de votre application toutes les fonctionnalités de Symfony2.

Pour plus de commodité, nous allons effectuer une modification dans l'un des fichiers de Symfony2. Elle nous permettra d'utiliser le préfixe **/projet** dans les URL de notre application.

• Ouvrez le fichier **symfony/app/config/routing.yml**.
• Compléter son contenu par le code suivant :

```
projet:
        resource: "@ProjetBundle/Resources/config/routing.yml"
        prefix:   /projet
```

• Sauvegardez.

Symfony2 sait désormais que vous avez créé un nouveau bundle et notre application est prête à fonctionner. C'est ce que nous allons vérifier, en essayant d'afficher un premier message.

4 - Afficher un premier message

Ouvrez le fichier
symfony/src/ProjetBundle/Controller/DefaultController.php

• Dans la fonction appelée **indexAction()**, on voit le code existant :

```
class DefaultController extends Controller {

   public function indexAction($name)    {

      return $this->render('ProjetBundle:Default:index.html.twig', array('name' => $name) );

   }

}
```

• Puis ouvrez le fichier **symfony/src/ProjetBundle/Resources/views/Default/index.html.twig**
• Modifier le code existant par :

```
Bonjour {{ name }} !
```

• Ouvrez le fichier **symfony/src/ProjetBundle/Resources/config/routing.yml**
• Dans ce fichier, voir le code :

```
projet_homepage:

    path:    /hello/{name}

    defaults: { _controller: ProjetBundle:Default:index }
```

• Dans votre navigateur web, tapez l'URL :
https://webdev.iut-blagnac.fr/~VotreCompte/symfony/web/projet/hello/laurent

• Vous devriez voir le texte "Bonjour Laurent !" s'afficher.

Si le texte ne s'affiche pas, il faut vider le cache comme suit :

- Aller dans le répertoire symfony sous Mobaxterm
- Taper : **php app/console cache:clear -e prod**
    - Ou **php app/console cache:clear** si on veut vider le cache de l'environnement de développement

- Tester le mode Développement en tapant les 2 URL suivantes :
    - https://webdev.iut-blagnac.fr/~VotreCompte/symfony/web/**app_dev.php**/projet/hello/laurent

    - https://webdev.iut-blagnac.fr/~VotreCompte/symfony/web/**app_dev.php/XXXX**/hello/laurent

Dossier à rendre sous Moodle (en **PDF UNIQUEMENT**) :

- Page de garde avec nom-prénom, groupe, date, matière, numéro et intitulé du TP
- Sommaire
- Introduction (objectif / thème du TP)
- **Diagramme de séquence** explicitant le fonctionnement de Symfony lors de l'appel de l'URL ci-dessus.
- Pour chaque nouvelle page :
    - Code « intéressant » commenté
    - Copie d'écran pour la trace d'exécution
- Conclusion (notions importantes abordées dans ce TP)