

总览

今天,我将带着大家开始在10分钟内掌握Symfony框架！我在这里将引导你完成Symfony最重要的概念，并解释如何可以快速搭建一个简单的项目。

如果你以前使用过的web框架，对于Symfony你应该感到宾至如归的。如果不是，欢迎使用一种全新的方式来开发Web应用程序。

唯一的技术必要遵循本教程是有PHP 5.4或更高版本安装在你的计算机上。如果你使用的是打包的PHP解决方案，如WAMP，XAMP或MAMP，看看他们是否使用PHP 5.4或更高版本。你也可以在你的终端或命令控制台显示已安装的PHP版本，请执行以下命令：

```
php --version
```

安装的Symfony

在过去，Symfony的必须手动为每个新项目安装。现在，你可以使用Symfony的安装程序，它必须要在你第一次使用Symfony的计算机上安装。在Linux和Mac OS X系统，请执行以下控制台命令：

```
1. $ curl -Ls http://symfony.com/installer > symfony.phar
2. $ sudo mv symfony.phar /usr/local/bin/symfony
3. $ chmod a+x /usr/local/bin/symfony
```

如果你的系统没有安装cURL，执行以下命令来安装：

```
1. $ php -r "readfile('http://symfony.com/installer');" > symfony.phar
2. $ sudo mv symfony.phar /usr/local/bin/symfony
3. $ chmod a+x /usr/local/bin/symfony
```

安装Symfony的安装后，你可以打开一个新的控制台窗口，以便能够执行新的symfony命令：

```
symfony
```

在Windows系统中，执行以下控制台命令：

```
c:\> php -r "readfile('http://symfony.com/installer');" > symfony.phar
```

该命令会下载一个名为symfony.phar文件，其中包含的Symfony安装程序。保存或移动文件的目录中创建Symfony的项目，然后用这个命令执行的Symfony安装程序：

```
c:\> php symfony.phar
```

创建第一个symfony项目

一旦Symfony的安装设置，使用新的命令来创建新的Symfony项目。让我们创建一个新的称为MyProject的项目：

```
1. # Linux and Mac OS X
2. $ symfony new myproject
3.
4. # Windows
5. c:\> php symfony.phar new myproject
```

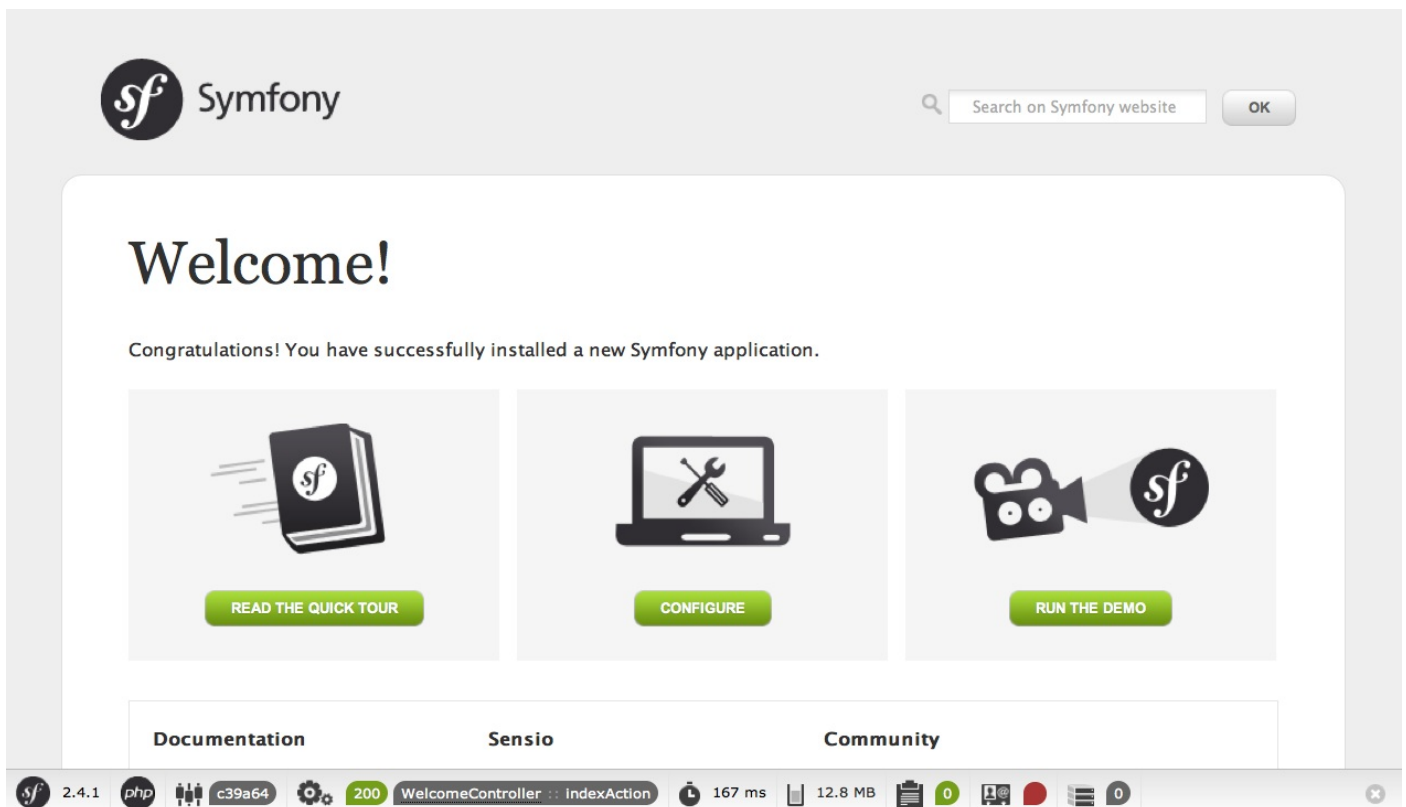
该命令会下载最新的Symfony的稳定版本，并在MyProject的/目录中创建一个空的项目，这样你就可以开始开发应用程序了。

运行的Symfony

本教程利用PHP内部的Web服务器来运行Symfony的应用程序。因此，运行Symfony的应用是浏览项目目录和执行该命令的问题：

```
$ cd myproject/$ php app/console server:run
```

打开浏览器，访问 <http://localhost:8000>，就可以看到的Symfony的欢迎页面：



恭喜！你的第一个symfony项目启动并运行成功了！

如果人品太差，在欢迎页面中，你可能会看到一个空白页或错误页。这是一个目录权限配置错误引起的。有根据不同的的操作系统几种可能的解决方案。但是这些我们在这里并不累述,读者自行百度或者google.

当你完成工作后你的Symfony的应用程序，你可以用：`server:stop` 命令 停止服务器:

```
$ php app/console server:stop
```

当然,如果你喜欢传统的Web服务器,如Apache或Nginx的,可以自行配置。

了解基础知识

一个框架的主要目标是维护你的代码架构,让你的应用程序变得简洁优雅,避免数据库频繁调用,以及相同的HTML标签在同一个脚本其他的PHP代码的混合。为了实现Symfony的这一目标,你需要先了解一些基本概念。

当开发一个Symfony的应用程序时,作为一个开发人员,你的责任就是编写响应用户请求的代码(例如 `http://localhost:8000/`),返回与它相关的资源(欢迎使用Symfony的HTML页面!)。

该代码执行的动作和控制器的定义。用户的请求和代码之间的映射是通过路由配置定义的。显示在浏览器的内容通常使用模板呈现。当你早些时候浏览 `http://localhost:8000/` 时, Symfony执行的控制器被定义在 `src/AppBundle/Controller/DefaultController.php` 中,并且用这个控制器来渲染 `app/Resources/views/default/index.html.twig` 模板,

在下面的部分,你将详细了解Symfony的控制器,路线和模板的内部运作。

Actions and Controllers

打开 `src/AppBundle/Controller/DefaultController.php` 文件,你会看到下面的代码(现在,不看 `@Route` 配置,因为这将在以后解释):

```
1. namespace AppBundle\Controller;
2.
3. use Sensio\Bundle\FrameworkExtraBundle\Configuration\Route;
4. use Symfony\Bundle\FrameworkBundle\Controller\Controller;
5.
6. class DefaultController extends Controller
7. {
8.     /**
9.      * @Route("/", name="homepage")
10.     */
11.     public function indexAction()
12.     {
13.         return $this->render('default/index.html.twig');
14.     }
15. }
```

在Symfony的应用中,控制器的名字通常为后缀为 `Controller` 的PHP类。在本实例中,控制器被称为 `Default` 和PHP类被称为 `DefaultController`。

在一个控制器(controller)中定义的方法被称为操作(actions),它们通常与应用程序中的一个URL相关联,其名称后缀为 `Action`。在这个例子中,在默认控制器 `Default` 只有称为 `index`,被定义在 `indexAction` 中。

操作(Action)通常很短 - 约10-15行代码 - 因为它们只是调用应用程序的其他部分来获得或产生所需要的信息,然后他们生成模板来向用户做出响应。

在本实例中,索引(index)实际上是空的,因为它不需要调用任何其它方法。`Action` 仅仅呈现的欢迎Symfony的模板! 内容。

路由 (Routing)

symfony的路由每个请求到通过匹配请求的URL,针对由构成的路径中的应用程序处理它的动作。再次打开 `src/AppBundle/Controller/DefaultController.php` 文件,并看看前的 `indexAction` 方法的代码:

```

1. // src/AppBundle/Controller/DefaultController.php
2. namespace AppBundle\Controller;
3.
4. use Sensio\Bundle\FrameworkExtraBundle\Configuration\Route;
5. use Symfony\Bundle\FrameworkBundle\Controller\Controller;
6.
7. class DefaultController extends Controller
8. {
9.     /**
10.      * @Route("/", name="homepage")
11.      */
12.     public function indexAction()
13.     {
14.         return $this->render('default/index.html.twig');
15.     }
16. }

```

这三行代码经由 `@Route()` 注释定义的路由配置。一个PHP的注释是一种方便的方式来配置方法，而无需编写定期PHP代码。要注意的是注释块是以 `/**` 开始,而普通的PHP注释以 `/*`。

`@Route()` 的第一个值定义了将触发操作执行的URL。当你没有应用程序的主机添加到URL（例如 `http://example.com`）时，这些URL总是相对的，它们通常被称为路径。在这种情况下，`/` 路径是指应用程序的主页。`@Route()` 的第二个值（）（例如 `name="homepage"`）是可选的，并且设置该路由的名称。现在这个名字是没有必要的，但后来这对用于链接的页面非常有用。

考虑到这一切，`Route("/", name="homepage")` 标注创建了一个新的路线被称为 `homepage`，这使得Symfony的执行 `Default` 控制器()的索引(`index`)操作，当用户浏览该应用程序的 `/` 路径。

除了PHP注释，路由也能用YAML，XML或PHP文件进行配置，这将在以后说到。这种灵活性是Symfony的一个主要特点，一个框架，它不会用一种特殊的配置去禁锢你。

模板(Templates)

该 `index` 的action的唯一内容是此PHP指令：

```
return $this->render('default/index.html.twig');
```

`$this->render()` 方法是一个方便的快捷方式来呈现一个模板。Symfony的提供了一些有用的快捷键，从控制器 (`Controller`) 类扩展任何控制器。

默认情况下，应用程序模板存储在 `app/Resources/views/` 目录，因此，`default/index.html.twig` 模板对应于 `app/Resources/views/default/index.html.twig`。打开该文件，你会看到下面的代码：

```

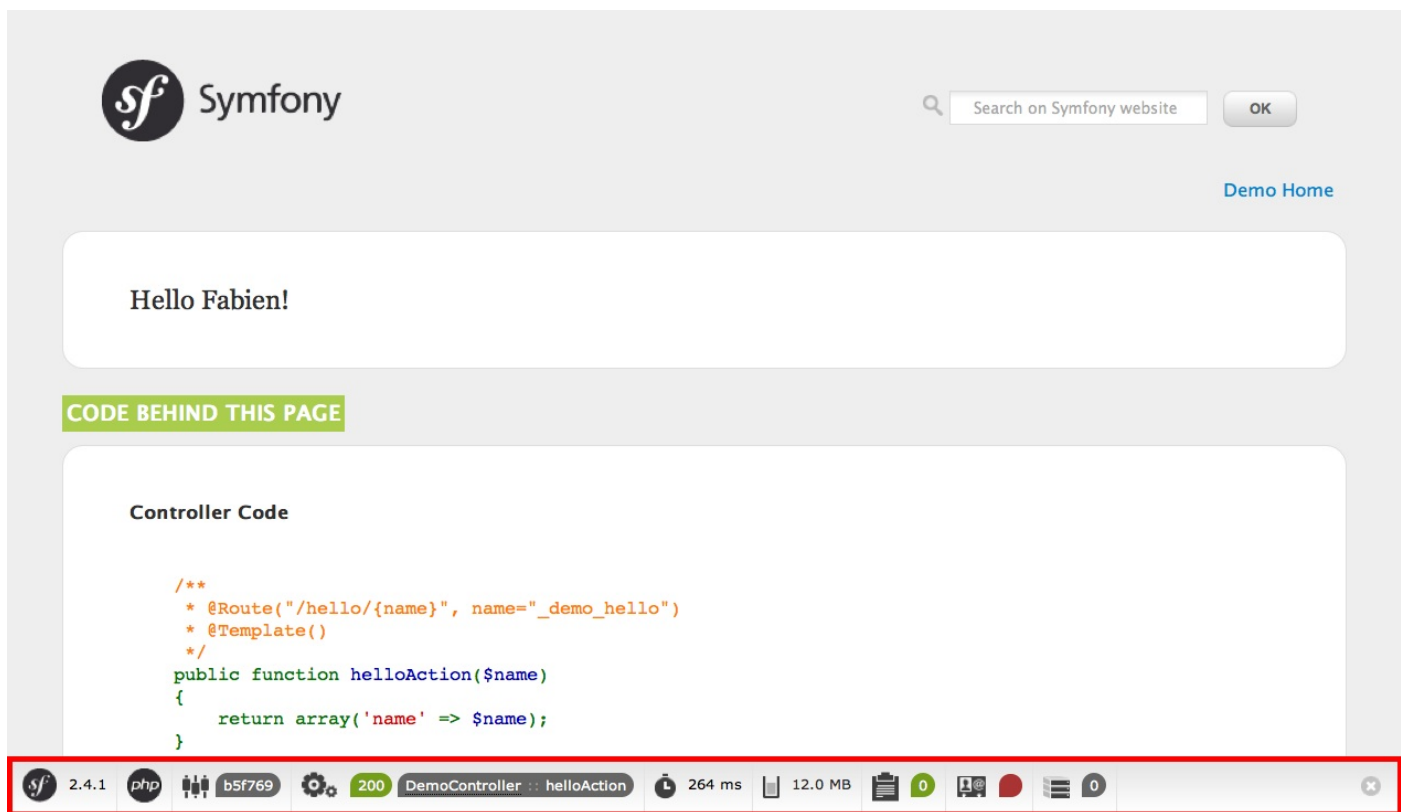
1. {# app/Resources/views/default/index.html.twig #}
2. {% extends 'base.html.twig' %}
3.
4. {% block body %}
5.     <h1>Welcome to Symfony!</h1>
6. {% endblock %}

```

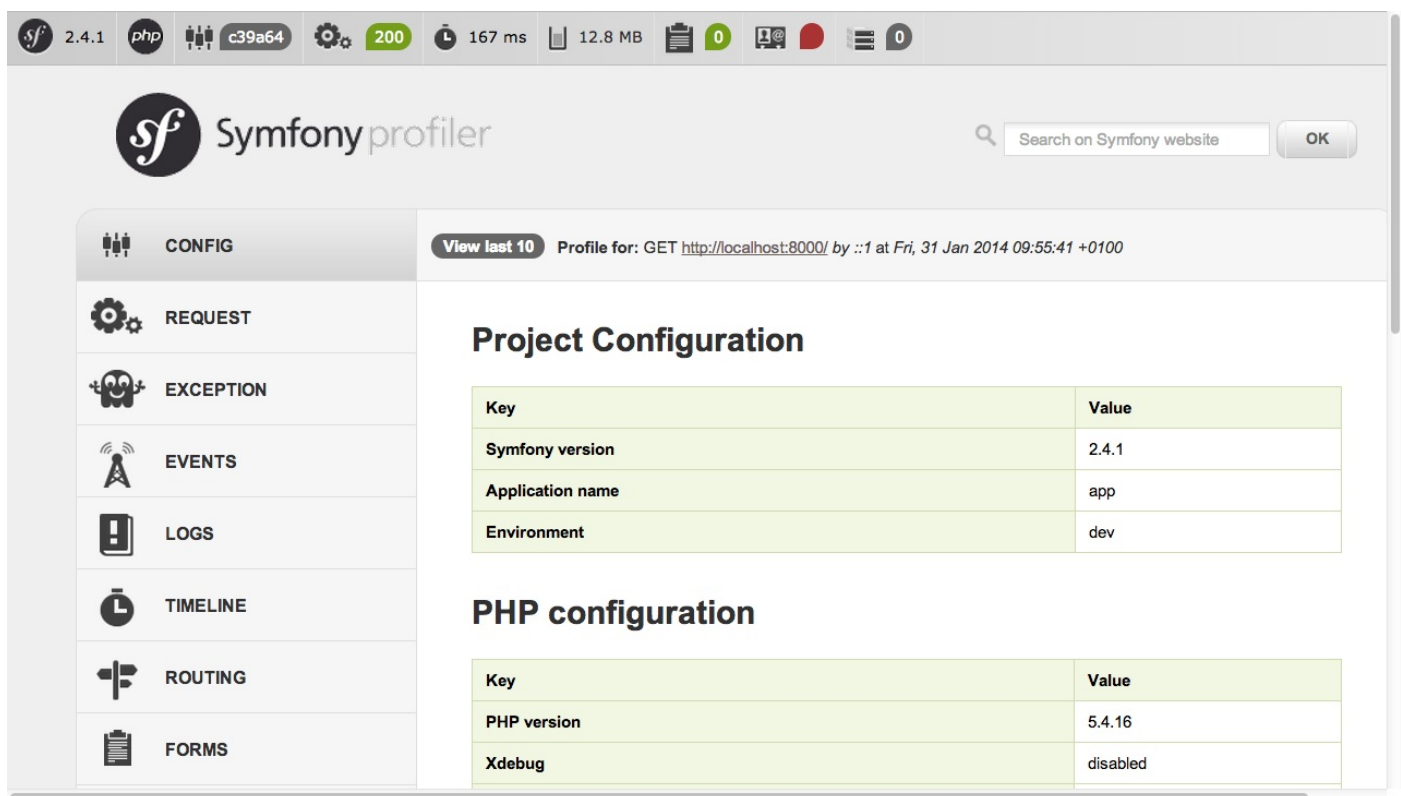
This template is created with [Twig](#), a new template engine created for modern PHP applications. The [second part of this tutorial](#) will introduce how templates work in Symfony.

工作环境(Working with Environments)

现在，你应该对Symfony是如何工作的有一个更好的了解，仔细看看，在Symfony的渲染页面的底部。你会发现与Symfony的标志的小酒吧。这是“网页调试工具”，它是一个Symfony的开发人员的最好的朋友！



但是你看到的只是冰山的一角;点击任何条部分，以打开该探查，并获得关于请求，查询参数，安全细节，和数据库查询更详细的信息：



此工具提供有关你的应用程序的相关信息，比如:你可能会担心你的访问者访问敏感信息，以及更多的内部信息。Symfony的是意识到这个问题，因为这个原因，它不会让你的应用程序在生产服务器上运行显示此栏。Symfony如何知道你的应用程序是在本地还是在生产服务器上运行？请继续阅读，发现执行环境的概念

什么是环境？

环境代表一组用于运行应用程序的配置。Symfony的定义了两种环境默认：`dev`（适合本地开发应用程序时进行），`prod`（对生产的应用程序进行优化）。

当你在浏览器中访问 `http://localhost:8000` 时，你执行的是 `dev` 环境的Symfony应用程序。要访问在 `prod` 环境的应

用程序，访问 `http://localhost:8000/app.php` 来代替。如果你希望始终显示开发环境中的URL，就可以访问 `http://localhost:8000/app_dev.php`。

环境之间的主要区别是，`dev` 的被优化，以提供大量的信息给开发人员，这意味着更坏的应用性能。同时，`prod` 进行了优化，以获得最佳的性能，这意味着调试信息以及调试工具栏被禁止。

环境之间的另一个不同是用于执行该应用程序的配置选项。当你访问的 `dev` 环境时，Symfony的加载 `app/config/config_dev.yml` 配置文件。当你访问 `prod` 环境，Symfony的加载 `app/config/config_prod.yml` 文件。

典型地，环境共享大量的配置选项。出于这个原因，你把你常用的配置在 `config.yml` 并覆盖每个环境中需要特定的配置文件：

```
1. # app/config/config_dev.yml
2. imports:
3.     - { resource: config.yml }
4.
5. web_profiler:
6.     toolbar: true
7.     intercept_redirects: false
```

在这个例子中，`config_dev.yml` 配置文件导入共同 `config.yml` 文件，然后将覆盖有其自己的选项任何现有的调试工具栏配置。

最后的思考

恭喜！你已经入门了，你先试试看。这是不是很难，是吗？还有很多更多的探索，但你应该已经看到Symfony的是如何使得它可以很容易实现 的网站更好更快。教程文档将持续更新翻译.欢迎大家访问我的主页.

- 我的主页: <http://aifei8.net>
- 我的博客: <http://blog.aifei8.net>
- 我的github:<http://github.com/Pengfei-Gao>



