

安装和配置 Symfony

本章的目标是让你兴奋一下，我们会运行一个用 Symfony 构建的应用程序。

幸运的是，Symfony 提供了一个发行版，这是一个你下载后可以直接开始开发的功能齐全的 Symfony 上手项目。



如果你想知道最好的创建项目和使之加入源代码管理的方法，请浏览文章最后的部分。

安装 Symfony2 发行版



首先，检查一下你已经安装和配置好了一个支持 php 的 web 服务器，比如 apache。想了解更多 Symfony2 运行要求的内容，请查阅 [requirements reference](#)

Symfony2 发行版是一个功能齐全的应用，它包括了 Symfony2 的核心类库、一些有用的 bundle、一个清晰的目录结构和一些默认配置。你下载的发行版，就是一个可以立即开始开发的功能性应用骨架。

先浏览 Symfony2 的下载页面：<http://symfony.com/download>。在这个页面，你可以看到 Symfony 标准版，这是 Symfony2 最重要的版本。有两种方式开始着手项目：

一、Composer

[Composer](#) 是 php 的一个依赖管理库，你可以使用它下载 Symfony2 的标准版。

首先，你要在自己的电脑上[下载 Composer](#)。如果你安装了 curl 会非常简单：

```
1 $ curl -s https://getcomposer.org/installer | php
```



如果你的电脑不满足使用 composer 的条件，运行命令的时候你会看到一些建议。按照建议做以使 composer 可以很好的工作

Composer 是一个可执行的 PHAR 文件，你可以使用它下载标准发行版：

```
1 $ php composer.phar create-project symfony/framework-standard-edition /path/to/webroot/Symfony 2.4.*
```



可以在任何 Composer 命令后面增加--prefer-dist 选项，以使 vendor 文件可以下载的更快

这个命令会花费几分钟去下载标准发行版和它所需的 vendor 库。当命令结束时，你应该有一个像下面这样的目录结构：

```
1 path/to/webroot/ <- your web server directory (sometimes named httdocs or public)
2   Symfony/ <- the new directory
3     app/
4       cache/
5       config/
6       logs/
7       src/
8       ...
9       vendor/
10      ...
11     web/
12       app.php
13      ...
```

二、下载文档包

你也可以下载标准发行版的文档包。这种方式有两个选择：

- 下载.tgz 或.zip 文档，两个文档一样，想用哪个就下那个
- 下载有或无 vendor 的版本。如果你打算使用更多的第三方库或者包，而且想使用 Composer 管理他们，就下载无 vendor 版。

下载一个文档到你 web 服务器的跟目录，然后解压。在 unix 命令行，可以按照下面的命令做（用真实的名字替换###）：

```
1 # for .tgz file
2 $ tar zxvf Symfony_Standard_Vendors_2.4.###.tgz
3
4 # for a .zip file
5 $ unzip Symfony_Standard_Vendors_2.4.###.zip
```

如果你下载的是“without vendor”版，你需要阅读下面的章节。



你可以简单的推翻默认的结构。阅读[如何覆盖 symfony 默认的目录结构](#)

所有公共资源和处理进来的请求的前端控制器文件都在 `Symfony/web` 目录。所以，建设你把文档解压到你的 web 服务器的根目录下或者虚拟主机的文档目录，你的应用的 url 应该以 `http://localhost/Symfony/web/` 开头。



下面的例子假设你没有修改文档根目录，所以所有的 url 都以 `http://localhost/Symfony/web/` 开头

更新 vendors

现在你已经下载了一个功能齐全的 Symfony 项目，用它你可以开始开发你的应用了。Symfony 项目依赖很多的外部库。他们通过一个叫 [Composer](#) 的库被下载到 vendor 目录。根据你如何下载的 Symfony，你可能需要立即更新你的 vendor 目录。但是，更新你的 vendor 是非常安全的，它能保证你拥有你所需的所有库。

第一步，获取 Composer

```
1 $ curl -s http://getcomposer.org/installer | php
```

保证你下载的 `composer.phar` 文件和 `composer.json` 文件在同一个目录 (默认是 Symfony 项目的跟目录)。

第二步，安装 vendors

```
1 $ php composer.phar install
```

这个命令会下载所有需要的 vendor 库 (包括 Symfony 自身) 到 vendor 目录。



如果你没有安装 curl，你可以在 <http://getcomposer.org/installer> 手动下载 installer。文件把这个文件放到你的项目中，然后运行：

```
1 $ php installer
2 $ php composer.phar install
```



当运行 `php composer.phar install` 或者 `php composer.phar update`, Composer 在执行 `install/update` 命令之后会清除缓存和安装资源。默认情况下这些资源会被复制到 `web` 目录。

如果你的操作系统支持, 你可以创建符号链接而不是复制。为了创建符号链接, 在你的 `composer.json` 文件中增加 `extra` 节点, 使用 `symfony-assets-install` 作为 key, `symlink` 为值:

```
"extra": {
    "symfony-app-dir": "app",
    "symfony-web-dir": "web",
    "symfony-assets-install": "symlink"
}
```

当给 `symfony-assets-install` 赋 `relation` 而不是 `symlink` 的值时, 这个命令会创建相对的符号链接

配置和安装

现在, 所有所需的第三方库被安装到 `vendor` 目录。你也会有一个默认的安装的程序在 `app/` 目录中, 还有一些 demo 代码在 `src/` 目录。

Symfony2 自带了一个服务的测试工具, 帮助你确认你的 web 服务器和 php 符合 Symfony 的要求。使用这个地址检查配置:

```
1 http://localhost/config.php
```

如果出现任何问题, 在继续之前先解决他们。



配置权限

一个常见问题是 `app/cache` 和 `app/logs` 目录必须对 web 服务器和命令行用户都是可写的。在 UNIX 系统中, 如果 web 服务器的用户和命令行用户不是同一个用户, 你可以运行一次下面的命令保证权限被正确的配置。

1. 在支持 `chmod +a` 的系统上使用 `acl`

很多系统支持你使用 `chmod +a` 命令。先尝试这个命令, 如果出错了, 尝试下面的方法。这个方法让找到你的 web 服务器用户, 然后把它设置为 `APACHEUSER`:

```
1 $ rm -rf app/cache/*
2 $ rm -rf app/logs/*
3
4 $ APACHEUSER=$(ps aux | grep -E '[a]pache|[h]ttpd|[_]www|[w]wv-data' | grep -v root | head -1 | cut -d\ -f1)
5 $ sudo chmod +a "$APACHEUSER allow delete,write,append,file_inherit,directory_inherit" app/cache app/logs
6 $ sudo chmod +a "$(whoami) allow delete,write,append,file_inherit,directory_inherit" app/cache app/logs
```

2. 在不支持 `chmod +a` 的系统上使用 `acl`

一些系统不支持 `chmod +a`，但是支持其他的工具叫做 `setfacl`。你需要[支持 ACL](#)，以及在使用前安装它。这个方法用命令获取你的 web 服务器用户，然后设置为 `APACHEUSER`：

```
1 $ APACHEUSER=$(ps aux | grep -E '[a]pache|[h]ttpd|[l]www|[w]ww-data' | grep -v root | head -1 | cut -d\ -f1)
2 $ sudo setfacl -R -m u:$APACHEUSER:rwX -m u:whoami:rwX app/cache app/logs
3 $ sudo setfacl -dR -m u:$APACHEUSER:rwX -m u:whoami:rwX app/cache app/logs
```

3. 不使用 ACL

如果你没有权限改变目录的 ACL，你需要改变 `umask` 以使缓存和日志目录是组可写的。为了实现这点，把下面的命令放在 `app/console`，`web/app.php` 和 `web/app_dev.php` 前面：

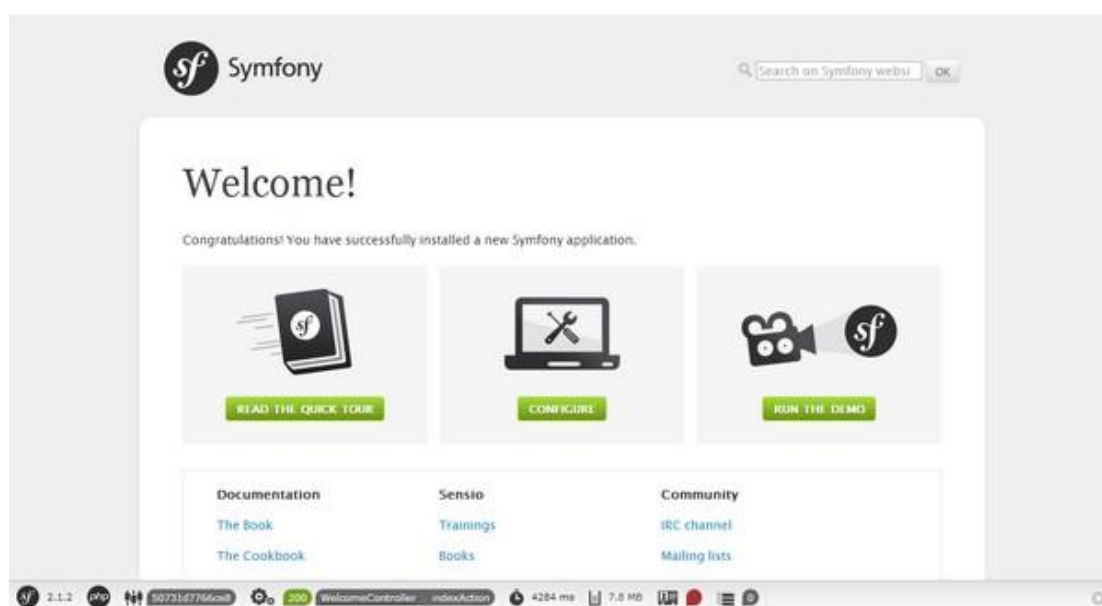
```
1 umask(0002); // This will let the permissions be 0775
2
3 // or
4
5 umask(0000); // This will let the permissions be 0777
```

当所有事情都好了，点击 “Go to the Welcome page” 去请求你第一个真的 Symfony2

页面：

```
1 http://localhost/app_dev.php/
```

Symfony2 会欢迎和祝贺你到目前为止的努力。





想要获取精简的 url，你要更改 web 服务器或虚拟主机的根目录到 `symfony/web` 目录。

尽管在开发环境这不是必须的，但是推荐在你的应用部署到生成环境时这么做。想获取配置 web 服务器根目录的信息，阅读[配置 web 服务器](#)或查阅你的 web 服务器的官方文档：

[Apache](#)，[Nginx](#)。

开始开发

现在，你已经有了一个功能丰富的 Symfony2 应用，你可以开始开发了。

你的版本可能包含一些实例的代码 -- 检查 `README.md` 文件，了解你的版本中包含了哪些实例的代码。

如果你是刚接触 Symfony，看看[创建页面](#)，你会学到如何创建页面、改变配置和在你的新应用中的所有事情。

也要保证查看 [CookBook](#)，他介绍了 Symfony 中一些特定问题的解决方法。



如果你想移除 demo 代码，看看 cookbook 中的文章[如何移除 AcmeDemoBundle](#)

使用代码控制工具

如果你在使用像 git 或 svn 这样的版本控制工具，你可以像往常一样配置你的版本控制、提交你的项目。Symfony 标准发行版是你的新项目的第一个版本。

忽略 vendor 目录

如果你下载的是“without vendor”版，你可以安全的忽略 vendor 目录，不把它提交到版本库。如果使用 git，创建一个 `.gitignore` 文件，然后添加：

```
1 /vendor/
```

现在 vendor 目录不会被提交到版本库。这很好（事实上很非常好）。因为当其他人 clone

或者检出这个项目 他们只要运行 `php composer.phar install` 脚本就能安装所有依赖的库。