

Symfony 1.4 on Git HOWTO

Version 1.1

by Alec Joseph Rivera <agirivera@gmail.com>

1. Introduction

This HOWTO is only meant to supplement the Getting Started guide from the Symfony Documentation, not as a standalone guide. The guide can viewed at: http://www.symfony-project.org/getting-started/1_4/en, or downloaded as a PDF at: <http://www.symfony-project.org/get/pdf/getting-started-1.4-en.pdf>.

2. What is Symfony?

Quoting for Wikipedia:

"Symfony is a web application framework written in PHP which follows the model-view-controller (MVC) paradigm."

"Symfony aims to speed up the creation and maintenance of web applications and to replace repetitive coding tasks... Symfony has low performance overheads when dealing with an environment that supports a PHP accelerator."

"Symfony is aimed at building robust applications in an enterprise context, and aims to give developers full control over the configuration: from the directory structure to the foreign libraries, almost everything can be customized. To match enterprise development guidelines, Symfony is bundled with additional tools to help developers test, debug and document projects."

3. Prerequisites

This HOWTO assumes you already have the LAMP stack up and running. Refer to LAMP HOWTO for lamp installation if not yet done. Also since Symfony uses the Subversion SCM, we will have to the make sure git svn is working properly. Please refer to the Git HOWTO for the corresponding procedure. Since Symfony uses the Subversion SCM, we will have to the make sure git svn is working properly. Please refer to the Git HOWTO for the corresponding procedure.

4. Installation

1. Create your project root directory in /srv/http together with lib/vendor structure to hold the framework.

```
$ cd /srv/http
$ sudo mkdir -p {project_name}/lib/vendor
$ sudo chown -R {your_username}:http {project_name}
```

Afterwards, you can make a convenient link to your project anywhere on your home directory.

```
$ ln -s /srv/http/{project_name}
```

2. Clone the symfony-1.4 svn repository.

```
$ git svn clone http://svn.symfony-project.com/branches/1.4 \
  lib/vendor/symfony-1.4
```

When a 1.4 update comes along, go to the lib/vendor/symfony-1.4 directory and execute the following command to update your copy.

```
$ git svn rebase
```

3. Now take care of the svn externals by first retrieving a script that extends git's power (based on a work by Andre Pang) and running git-svn-update-externals inside the framework directory lib/vendor/symfony-1.4.

```
$ git clone git://git-sue.git.sourceforge.net/gitroot/git-sue/git-sue \
  lib/vendor/git-sue
```

```
$ cd lib/vendor/symfony-1.4
$ ../git-sue/git-svn-update-externals
```

4. Assuming you are back on your project root directory, verify your Symfony installation.

```
$ lib/vendor/symfony-1.4/data/bin/symfony -V
```

If Symfony has been installed correctly, the version should be displayed.

5. Initialize your Symfony project.

```
$ lib/vendor/symfony-1.4/data/bin/symfony generate:project {project_name}
```

Note that Symfony also creates a symfony script at your project root directory. You can

use that instead of referring back to lib/vendor/..., which can be really repetitive.

6. Modify `./symfony` to add directory tracking support for git. Before the `chdir()` call in `symfony`, add the following line:

```
register_shutdown_function(create_function('', \
    'touch("cache/.ignore"); touch("log/.cache");'));
```

7. Initialize a project application. Repeat for other applications if project consists of more than one application. You can skip this step if your project is still on the drawing board, but pay attention to step 9.

```
$ ./symfony generate:app {app_name}
```

8. Configure git scm on project root and add user information. Signing key and remote repository are optional (team/public project), although they are recommended for obvious reasons.

```
$ git init
$ git config user.name {real_name}
$ git config user.email {email_address}

$ # git config user.signingkey {gpg_key}
$ # git remote add origin {repo_url}
```

9. Add git exclude information to `.git/info/exclude`. Files to be excluded are normally editing backup copies and vendor files (which are already gitted). Special care is also needed for cache and log directory because git don't track empty directory but can introduce symfony errors later on.

```
~
cache/*
!cache/.ignore
log/*
!log/.ignore
```

You can create the `.ignore` files now, especially if you skipped step 7.

```
$ touch cache/.ignore log/.ignore
```

10. Perform your initial commit. Optionally, you can also pushed your newly initialized repository to the remote repo if you'll be working with a team or just making it available publicly.

```
$ git add .  
$ git commit -a -m 'Initial commit.'  
  
$ # git push origin master
```

11. Configure your httpd configuration with virtual hosting as appropriate, and don't forget to alias /sf to lib/vendor/symfony-1.4/data/web/sf. Refer to the LAMP HOWTO for the procedures. Upon (re)starting your httpd, you should be able now to see the Symfony Congratulations page.

5. Get Involved

In closing, if you have passing interest or you will heavily doing web development with Symfony, I recommend you to subscribe to:

Symfony Project Blog	http://www.symfony-project.org/blog/
Symfony Users Group	http://groups.google.com/group/symfony-users
Symfony Blogs Feed	http://feeds.feedburner.com/symfony/planet

6. Comments and Suggestions

You can contact me at my email address <agirivera@gmail.com> for corrections, typos, comments and suggestions.

7. Changelog

12/04/2009 - Git miscellaneous added
12/03/2009 - Initial release

8. Copyrights and License

"symfony" is a trademark of Fabien Potencier. All rights reserved.

"git" is a little a bit hazy but it was started out by Linus Torvalds and nurtured by many other hackers around the net.

This HOWTO is licensed under Creative Commons Attribution-Noncommercial-Share Alike 3.0 Philippines License. For information, visit <http://creativecommons.org>.