symfony-si / **symfony-cheatsheet**

Symfony cheat sheet - useful practical commands, code and snippets you will frequently need for working with Symfony framework

🔗 **symfony.si/cheatsheet**

⚖ View license

☆ **23** stars    ⑂ **11** forks

| ☆ Star | ⊙ Watch ▾ |
|--------|-----------|

Code    Issues **1**    Pull requests    Actions    Security    Insights

⑂ master ▾                                    ⋯

petk   …                           on 19 Feb 2015   🕘

View code

README.md

# Symfony Cheat Sheet

Useful practical commands, code and things you will frequently need for working with Symfony framework.

## Table of contents

- Symfony Installer
- Composer
- Console
  - General
  - Cache
  - Bundle

# Symfony Installer

```
# Install Symfony Installer
curl -LsS http://symfony.com/installer > symfony.phar
sudo mv symfony.phar /usr/local/bin/symfony
chmod a+x /usr/local/bin/symfony
```

```
# Create a new project
symfony new my_project
```

```
# Update the Symfony installer
symfony self-update
```

```
# List help
symfony about
```

# Composer

```
# Create new project with Composer
composer create-project symfony/framework-standard-edition ~2.6
```

```
# Add package to project
composer require vendor/library
```

```
# Update all dependencies of current project
composer update
```

```
# Install dependencies for current project with versions defined in composer.lock
composer install
```

# Console

```
# Display help for given command
php app/console help [command]
```

```
# Display all configured public services
php app/console container:debug [--show-private] [service_name]
```

```
# Dump all assets to the filesystem
php app/console assetic:dump [--watch] [--force] [--period=...] [write_to]
```

```
# Dump the default configuration for an extension/bundle
php app/console config:dump-reference <bundle_or_extension_alias>
```

```
# Send messages from spool
php app/console swiftmailer:spool:send [--message-limit=...] [--time-limit=...] [--re
```

```
# Extract translation strings from templates of a given bundle.
# It can display them or merge the new ones into the translation files
php app/console translation:update [--prefix=...] [--update-format=...] [--dump-messa
```

```
# Lint a template and outputs to stdout the first encountered syntax error
php app/console twig:lint <filename>`
```

## Cache

```
# Clear the cached information
php app/console cache:clear [--no-warmup] [--no-optional-warmers]
```

```
# Warm up an empty cache
php app/console cache:warmup  [--no-optional-warmers]`
```

## Bundle

```
# Generate a bundle
php app/console generate:bundle [--namespace=...] [--dir=...] [--bundle-name=...] [--
```

## Routing

```
# Display current routes for application
php app/console router:debug [route_name]
```

```
# Dump all routes as Apache rewrite rules
php app/console router:dump-apache [--base-uri=...] [script_name]
```

```
# Debug routes by simulating a path info match
php app/console router:match <path_info>
```

```
# Display current exposed routes for an application
php app/console fos:js-routing:debug <name>
```

## Doctrine

```
# Clear all metadata cache for an entity manager
php app/console doctrine:cache:clear-metadata [--em=...] [--flush]
```

```
# Clear all query cache for an entity manager
php app/console doctrine:cache:clear-query [--em=...] [--flush]
```

```
# Clear all result cache for an entity manager
php app/console doctrine:cache:clear-result [--em=...] [--flush]
```

```
# Create the configured databases
php app/console doctrine:database:create [--connection=...]
```

```
# Import mapping information from an existing database
php app/console doctrine:mapping:import [--em=...] [--filter=...] [--force] <bundle>
```

```
# Show basic information about all mapped entities
php app/console doctrine:mapping:info [--em=...]
```

```
# Generate a new Doctrine entity inside a bundle
php app/console doctrine:generate:entity [--entity=...] [--fields=...] [--format=...]
```

```
# Generate entity classes and method stubs from your mapping information
php app/console doctrine:generate:entities [--path=...] [--no-backup] <name>
```

```
# Generate a form class based on a Doctrine entity
php app/console doctrine:generate:form <entity>
```

```
# Generate a CRUD based on a Doctrine entity
php app/console doctrine:generate:crud [--entity=...] [--route-prefix=...] [--with-wr
```

```
# Execute arbitrary DQL directly from the command line
php app/console doctrine:query:dql [--hydrate=...] [--first-result=...] [--max-result
```

```
# Execute arbitrary SQL directly from the command line
php app/console doctrine:query:sql [--depth=...] [--connection=...] <sql_to_execute>
```

```
# Executes (or dumps) the SQL needed to generate the database schema
php app/console doctrine:schema:create [--dump-sql] [--em=...]
```

```
# Executes (or dumps) the SQL needed to update the database schema to match the curre
php app/console doctrine:schema:update [--complete] [--dump-sql] [--force] [--em=...]
```

```
# Validate the Doctrine mapping files
php app/console doctrine:schema:validate [--em=...]
```

```
# Ensure that Doctrine is properly configured for a production environment
php app/console doctrine:ensure-production-settings [--complete] [--em=...]
```

```
# Load data fixtures to your database
php app/console doctrine:fixtures:load [--fixtures=...] [--append] [--em=...] [--purg
```

# Twig

```
{# Echo variable with html escaping #}
{{ content }}
```

```
{# Echo raw variable and disable html escaping #}
{{ content|raw }}
```

```
{# Extend template #}
{% extends '::base.html.twig' %}
```

# Controller

```
use Symfony\Component\HttpFoundation\Response;

public function helloAction()
{
    return new Response('Hello world!');
}
```

## Packages

No packages published

We use **optional** third-party analytics cookies to understand how you use GitHub.com so we can build better products. Learn more.

| Accept | Reject |
|---|---|