



usage

```
$ php app/console [options] command [arguments]
```

<> required
[] optional

options

<code>--help</code>	<code>-h</code>	Display help message	<code>--ansi</code>	Force ANSI output	<code>--shell</code>	<code>-s</code>	Launch the shell
<code>--quiet</code>	<code>-q</code>	Do not output any message	<code>--no-ansi</code>	Disable ANSI output	<code>--env</code>	<code>-e</code>	The Environment name
<code>--verbose</code>	<code>-v</code>	Increase verbosity of messages	<code>--no-interaction</code>	<code>-n</code>	<code>--no-debug</code>		Switches off debug mode
<code>--version</code>	<code>-V</code>	Display application version		Do not ask any interactive question			

commands

GENERAL

<code>\$ php app/console [list]</code>	List available commands and show the Symfony version	FrameworkBundle
<code>\$ php app/console help [command]</code>	Displays help for a command	FrameworkBundle
<code>\$ php app/console container:debug</code> <code> [--show-private] [service_name]</code>	Displays all configured public services	FrameworkBundle
<code>\$ php app/console assetic:dump</code> <code> [--watch] [--force] [--period=...] [write_to]</code>	Dumps all assets to the filesystem	AsseticBundle
<code>\$ php app/console config:dump-reference</code> <code> <bundle_or_extension_alias></code>	Dumps the default configuration for an extension/bundle	FrameworkBundle
<code>\$ php app/console swiftmailer:spool:send</code> <code> [--message-limit=...] [--time-limit=...]</code> <code> [--recover-timeout=...]</code>	Sends e-mails from the spool (queue)	SwiftmailerBundle
<code>\$ php app/console translation:update</code> <code> [--prefix=...] [--update-format=...]</code> <code> [--dump-messages] [--force]</code> <code> <locale> <bundle></code>	Extract translation strings from templates of a given bundle. It can display them or merge the new ones into the translation files	FrameworkBundle
<code>\$ php app/console twig:lint <filename></code>	Lints a template and outputs to stdout the first encountered syntax error	TwigBundle

CACHE

<code>\$ php app/console cache:clear</code> <code> [--no-warmup] [--no-optional-warmers]</code>	Clear the cached information	FrameworkBundle
<code>\$ php app/console cache:warmup</code> <code> [--no-optional-warmers]</code>	Warms up an empty cache	FrameworkBundle

ROUTING

<code>\$ php app/console router:debug [route_name]</code>	Displays current routes for an application	FrameworkBundle
<code>\$ php app/console router:dump-apache</code> <code> [--base-uri=...] [script_name]</code>	Dumps all routes as Apache rewrite rules	FrameworkBundle
<code>\$ php app/console router:match <path_info></code>	Debug routes by simulating a path info match	FrameworkBundle
<code>\$ php app/console fos:js-routing:debug <name></code>	Displays current exposed routes for an application	FOSJsRoutingBundle

BUNDLE

<code>\$ php app/console assets:install <target_dir></code> <code> [--symlink] [--relative]</code>	Installs bundles web assets under a public web directory	FrameworkBundle
<code>\$ php app/console generate:bundle</code> <code> [--namespace=...] [--dir=...]</code> <code> [--bundle-name=...] [--format=...]</code> <code> [--structure]</code>	Generates a bundle	FrameworkBundle

USER

<code>\$ php app/console fos:user:activate <username></code>	Activate a user	FOSUserBundle
<code>\$ php app/console fos:user:deactivate <username></code>	Deactivate a user	FOSUserBundle
<code>\$ php app/console fos:user:promote</code> <code> [--super] <username> [role]</code>	Promote a user by adding a rule	FOSUserBundle
<code>\$ php app/console fos:user:demote</code> <code> [--super] <username> [role]</code>	Demote a user by removing a rule	FOSUserBundle
<code>\$ php app/console fos:user:create</code> <code> [--super-admin] [--inactive] <username> [email] [password]</code>	Create a user	FOSUserBundle
<code>\$ php app/console fos:user:change-password</code> <code> <username> <password></code>	Change the password of a user	FOSUserBundle



DATABASE / DOCTRINE

<code>\$ php app/console doctrine:cache:clear-metadata [--em=...] [--flush]</code>	Clears all metadata cache for an entity manager	DoctrineBundle
<code>\$ php app/console doctrine:cache:clear-query [--em=...] [--flush]</code>	Clears all query cache for an entity manager	DoctrineBundle
<code>\$ php app/console doctrine:cache:clear-result [--em=...] [--flush]</code>	Clears all result cache for an entity manager	DoctrineBundle
<code>\$ php app/console doctrine:database:create [--connection=...]</code>	Creates the configured databases	DoctrineBundle
<code>\$ php app/console doctrine:database:drop [--connection=...] [--force]</code>	Drops the configured databases	DoctrineBundle
<code>\$ php app/console doctrine:mapping:convert [--filter=...] [--force] [--from-database] [--extend=...] [--num-spaces=...] [--namespace=...] [--em=...] <to-type> <dest-path></code>	Converts mapping information between supported formats	DoctrineBundle
<code>\$ php app/console doctrine:mapping:import [--em=...] [--filter=...] [--force] <bundle> <mapping-type></code>	Imports mapping information from an existing database	DoctrineBundle
<code>\$ php app/console doctrine:mapping:info [--em=...]</code>	Shows basic information about all mapped entities	DoctrineBundle
<code>\$ php app/console doctrine:generate:entity [--entity=...] [--fields=...] [--format=...] [--with-repository]</code>	Generates a new Doctrine entity inside a bundle	DoctrineBundle
<code>\$ php app/console doctrine:generate:entities [--path=...] [--no-backup] <name></code>	Generates entity classes and method stubs from your mapping information	DoctrineBundle
<code>\$ php app/console doctrine:generate:form <entity></code>	Generates a form class based on a Doctrine entity	DoctrineBundle
<code>\$ php app/console doctrine:generate:crud [--entity=...] [--route-prefix=...] [--with-write] [--format=...]</code>	Generates a CRUD based on a Doctrine entity	DoctrineBundle
<code>\$ php app/console doctrine:query:dql [--hydrate=...] [--first-result=...] [--max-result=...] [--depth=...] [--em=...] <dql_to_execute></code>	Executes arbitrary DQL directly from the command line	DoctrineBundle
<code>\$ php app/console doctrine:query:sql [--depth=...] [--connection=...] <sql_to_execute></code>	Executes arbitrary SQL directly from the command line	DoctrineBundle
<code>\$ php app/console doctrine:schema:create [--dump-sql] [--em=...]</code>	Executes (or dumps) the SQL needed to generate the database schema	DoctrineBundle
<code>\$ php app/console doctrine:schema:drop [--dump-sql] [--force] [--full-database] [--em=...]</code>	Executes (or dumps) the SQL needed to drop the current database schema	DoctrineBundle
<code>\$ php app/console doctrine:schema:update [--complete] [--dump-sql] [--force] [--em=...]</code>	Executes (or dumps) the SQL needed to update the database schema to match the current mapping metadata	DoctrineBundle
<code>\$ php app/console doctrine:schema:validate [--em=...]</code>	Validates the Doctrine mapping files	DoctrineBundle
<code>\$ php app/console doctrine:ensure-production-settings [--complete] [--em=...]</code>	Ensures that Doctrine is properly configured for a production environment	DoctrineBundle
<code>\$ php app/console doctrine:fixtures:load [--fixtures=...] [--append] [--em=...] [--purge-with-truncate]</code>	Load data fixtures to your database	DoctrineFixturesBundle
<code>\$ php app/console generate:doctrine:crud</code>	or <code>\$ php app/console doctrine:generate:crud</code>	alias
<code>\$ php app/console generate:doctrine:entities</code>	or <code>\$ php app/console doctrine:generate:entities</code>	
<code>\$ php app/console generate:doctrine:entity</code>	or <code>\$ php app/console doctrine:generate:entity</code>	
<code>\$ php app/console generate:doctrine:form</code>	or <code>\$ php app/console doctrine:generate:form</code>	
<code>\$ php app/console init:acl</code>	Mounts ACL tables in the database	SecurityBundle