

# Symfony Cheat Sheet



## Basic console usage

Command	Description
<code>bin/console</code>	List all available commands
<code>bin/console &lt;namespace&gt;</code>	List all available commands in a namespace / bundle
<code>bin/console &lt;command&gt;</code>	Run a command
<code>bin/console help &lt;cmd&gt;</code>	Display help for a command
<code>bin/console --env=&lt;env&gt;</code>	Run the console in a specific environment (dev/test/prod)
<code>bin/console -v vv vvv</code>	Increase output verbosity

## Debug commands

Command	Description
<code>debug:autowiring</code>	List all autowireable services
<code>debug:config &lt;bundle&gt;</code>	Dump bundle configuration
<code>debug:container</code>	List all public services
<code>debug:container &lt;service&gt;</code>	Get information about a service
<code>debug:container --env-vars</code>	Display container env vars
<code>debug:container --tags</code>	List tagged public services
<code>debug:container --tag=&lt;t&gt;</code>	List services with a specific tag
<code>debug:dotenv</code>	Display parsed dotenv values
<code>debug:dotenv &lt;search&gt;</code>	Display matching dotenv values
<code>debug:event-dispatcher</code>	List all registered event listeners
<code>debug:firewall &lt;name&gt;</code>	Display detailed information about a firewall
<code>debug:form</code>	Display form type information
<code>debug:messenger</code>	List dispatchable messages
<code>debug:router</code>	List all registered routes
<code>debug:router &lt;name&gt;</code>	Get information about a route
<code>debug:translation &lt;loc&gt;</code>	Display messages for locale
<code>router:match &lt;path&gt;</code>	Show matching route(s) for a path

## Doctrine commands

Command	Description
<code>doctrine:database:create</code>	Create the configured database
<code>doctrine:database:drop</code>	Drop the configured database
<code>doctrine:mapping:info</code>	List mapped entities
<code>doctrine:schema:update --dump-sql</code>	Generate SQL statements to sync the DB schema
<code>doctrine:schema:update --force</code>	Sync the DB schema with the mapping information
<code>doctrine:schema:validate</code>	Validate the mapping files
<code>doctrine:query:sql &lt;sql&gt;</code>	Run a SQL query
<code>doctrine:query:dql &lt;dql&gt;</code>	Run a DQL query
<code>doctrine:fixtures:load</code>	Load data fixtures to the DB

## Doctrine Migrations commands

Command	Description
<code>doc:migrations:generate</code>	Generate an empty migration class
<code>doc:migrations:diff</code>	Generate a migration to sync DB scheme with mapping files
<code>doc:migrations:migrate</code>	Migrate DB to the latest version
<code>doc:migrations:migrate &lt;alias&gt;</code>	Migrate DB to the (first/latest/prev/current/next) version
<code>doc:migrations:migrate &lt;FQCN&gt;</code>	Migrate DB to a specific version
<code>doc:migrations:up-to-date</code>	Check if the schema is up-to-date

## Other commands

Command	Description
<code>assets:install</code>	Install bundle assets
<code>cache:clear</code>	Clear the cache
<code>cache:warmup</code>	Warm up an empty cache

## MakerBundle commands

Command	Description
<code>make:auth</code>	Create a Guard authenticator
<code>make:command</code>	Create a new console command
<code>make:controller</code>	Create a new controller class
<code>make:crud</code>	Create CRUD for a Doctrine entity
<code>make:entity</code>	Create a new Doctrine entity class
<code>make:fixtures</code>	Create a Doctrine fixture class
<code>make:form</code>	Create a new form class
<code>make:functional-test</code>	Create a new functional test class
<code>make:message</code>	Create a new message & handler
<code>make:migration</code>	Create a new migration class
<code>make:registration-form</code>	Create a new registration form
<code>make:reset-password</code>	Create password reset
<code>make:security:form-login</code>	Generate a login form
<code>make:serializer:encoder</code>	Create a new serializer encoder
<code>make:serializer:normalizer</code>	Create a new serializer normalizer
<code>make:stimulus-controller</code>	Create a new Stimulus controller
<code>make:subscriber</code>	Create a new event subscriber
<code>make:twig-component</code>	Create a new Twig component
<code>make:twig-extension</code>	Create a new Twig extension
<code>make:unit-test</code>	Create a new unit test class
<code>make:user</code>	Create a new user class
<code>make:validator</code>	Create a new validator
<code>make:voter</code>	Create a new security voter

## Bash console autocompletion

```
bin/console completion bash | sudo tee /etc/bash_completion.d/console
```

## Form types

See <https://symfony.com/doc/current/reference/forms/types.html>.

Type	Description
TextType	A text (single line text) field
TextareaType	A multi-line text field
EmailType	An email field
IntegerType	A number field for integers
MoneyType	A money field, with a specifiable currency
NumberType	A number field
PasswordType	A password field
PercentType	A number field for percentages
SearchType	A search field
UrlType	A URL field
RangeType	A range input field (slider)
TelType	A tel (phone number) input field
CheckboxType	A single checkbox field
FileType	A file upload field
RadioType	A radio button field
HiddenType	A hidden field

## Choice form types

Type	Description
ChoiceType	Field for selecting one or more options (rendered as dropdown, radio buttons, or checkboxes)
EnumType	Choose from PHP enum cases
EntityType	Choose from a Doctrine entity
CountryType	Choose a country
LanguageType	Choose a language
LocaleType	Choose a locale
TimezoneType	Choose a timezone
CurrencyType	Choose a currency

## Date and time form types

Type	Description
DateType	A date field
DateIntervalType	A date interval field
DateTimeType	A date and time field
TimeType	A time field
BirthdaysType	A date field for birthdays
WeekType	Select a year and week

## Validation constraints

See <https://symfony.com/doc/current/reference/constraints.html>.

Usage:

use `Symfony\Component\Validator\Constraints` as `Assert`;

```
class User
{
    #[Assert\NotBlank]
    private string $name;
}
```

## Basic validation constraints

Constraint	Description
NotBlank	Value is not null, not an empty string, not 0, etc.
NotNull	Value is not strictly equal to null
Type(<type>)	Value is of a specific type
[Not]EqualTo	(Not) equal to another value
[Not]IdenticalTo	(Not) identical to another value
LessThan[0rEqual]	Less than (or equal to)
GreaterThan[0rEqual]	Greater than (or equal to)
Positive[0rZero]	Positive (or zero)
Negative[0rZero]	Negative (or zero)
Range	Between a minimum and maximum value

## String validation constraints

Constraint	Description
Length(<min>, <max>)	String is between <min> and <max> characters long
Email	String is a valid email address
Url	String is a valid URL
Hostname	String is a valid hostname
Ip	String is a valid IP address
UserPassword	String is the user's password
NotCompromisedPassword	String is not found in any data breaches
PasswordStrength	Password has a enough entropy
NoSuspiciousCharacters	String does not contain spoofing characters
Uuid	String is a valid UUID
CssColor	String is a valid CSS color

## Financial / Number constraints

Constraint	Description
Bic	Valid Business Identifier Code
CardScheme	Valid credit card number
Currency	Valid 3-letter ISO 4217 currency
Luhn	Passes the Luhn algorithm
Iban	Valid IBAN
Isbn	Valid ISBN

## Twig functions

Function	Description
asset(path)	Get the public path of an asset
is_granted(role, [object])	Check if access is allowed
path(route, [params])	Return relative URL for a route
url(route, [params])	Return absolute URL for a route
absolute_url(path)	Return absolute URL for a path

## Form related Twig functions

Function	Description
<code>form(form, options)</code>	Render a complete form
<code>form_start(form)</code>	Render the opening <code>&lt;form&gt;</code> tag
<code>form_end(form)</code>	Render the closing <code>&lt;/form&gt;</code> tag
<code>form_row(form)</code>	Render a form row
<code>form_widget(form)</code>	Render a form widget

## Symfony Attributes

Attribute	Description
<code>#[AsCommand]</code>	Register class as a console command
<code>#[AsController]</code>	Register class as a controller
<code>#[AsMessageHandler]</code>	Register class as a message handler
<code>#[AsEventListener]</code>	Register class as an event listener
<code>#[Route]</code>	Define a route for a method
<code>#[IsGranted]</code>	Require a role to access

## Symfony Flex commands

Command	Description
<code>composer symfony:dump-env &lt;env&gt;</code>	Compile <code>.env</code> files to <code>.env.local.php</code>
<code>composer symfony:recipes</code>	List available recipes
<code>symfony:recipes:install</code>	(Re)install recipes for packages
<code>symfony:recipes:update</code>	Update installed recipes

## Symfony CLI installation

```
wget https://get.symfony.com/cli/installer -O - | bash
```

## Symfony CLI basic commands

Command	Description
<code>symfony check:requirements</code>	Check requirements for running Symfony
<code>symfony version</code>	Display the Symfony CLI version
<code>symfony help</code>	Display all available commands
<code>symfony help &lt;command&gt;</code>	Display help for a specific command

## Creating a new Symfony project

Command	Description
<code>symfony new &lt;directory&gt;</code>	Create a new Symfony project
<code>symfony new --version=&lt;v&gt;</code>	Specify the skeleton version (lts/stable/next/previous)
<code>symfony new --webapp &lt;directory&gt;</code>	Create a new Symfony web application
<code>symfony new --demo &lt;dir&gt;</code>	Install the Symfony demo application

## Symfony Local Web Server

Command	Description
<code>symfony server:start</code>	Run a local web server
<code>symfony server:start -d</code>	Run the server in the background
<code>symfony server:status</code>	Display local web server status
<code>symfony open:local</code>	Open the local project in a browser
<code>symfony server:log</code>	Display local web server logs
<code>symfony server:stop</code>	Stop the local web server
<code>symfony server:ca:install</code>	Create a local Certificate Authority for serving HTTPS

## Local Proxy

Command	Description
<code>symfony proxy:start</code>	Run a local proxy server
<code>symfony proxy:domain:attach &lt;domain&gt;</code>	Attach a domain to the local proxy
<code>symfony proxy:domain:detach &lt;domain&gt;</code>	Detach domains from the proxy
<code>symfony proxy:status</code>	Get the status of the proxy
<code>symfony proxy:stop</code>	Stop the local proxy server

## PHP version switching & wrappers

Command	Description
<code>symfony local:php:list</code>	List locally available PHP versions
<code>symfony local:php:refresh</code>	Refresh the list of locally available PHP versions
<code>echo 8.2 &gt; .php-version</code>	Set project PHP version to 8.2
<code>symfony php -v</code>	Display the selected PHP version
<code>symfony php &lt;command&gt;</code>	Run a command with the selected PHP version
<code>symfony composer &lt;command&gt;</code>	Run a Composer command with the selected PHP version
<code>symfony console &lt;command&gt;</code>	Run a Symfony console command with the selected PHP version