

Symfony Cheat Sheet



Symfony CLI: Installation

```
curl -sS https://get.symfony.com/cli/installer | bash
```

Symfony CLI: Basic Commands

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

Creating a New Symfony Project

Command	Description
<code>symfony new <directory></code>	Create a new Symfony project in <directory>
<code>symfony new --webapp <directory></code>	Create a new web application
<code>symfony new --api <directory></code>	Create a new API application
<code>composer create-project symfony/skeleton <directory></code>	Create a new Symfony project using Composer

Flex Commands

Command	Description
<code>composer symfony:dump-env <env></code>	Compile .env files to .env.local.php
<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

Routing Attributes

use `Symfony\Component\Routing\Attribute\Route`;

Attribute	Description
<code>#[Route('/path')]</code>	Define a route for a controller method
<code>#[Route('/path', name: 'name')]</code>	Define a named route for a method
<code>#[Route('/path', methods: ['GET'])]</code>	Define allowed HTTP methods
<code>#[Route('/blog/{slug}']]</code>	Define a route with a parameter
<code>#[Route('/page/{page}', requirements: ['page' => '\d+'])]</code>	Define a route with a parameter constraint
<code>#[Route(path: ['en' => '/about-us', 'nl' => '/over-ons'], name: 'about')]</code>	Define localized routes

AbstractController Methods

Method	Description
<code>\$this->render('template.html')</code>	Returns a Response with the rendered template
<code>\$this->redirectToRoute('homepage')</code>	Returns a RedirectResponse to a named route
<code>\$this->redirectToRoute('name', ['param' => 'value'])]</code>	Returns a redirect to a route with parameters
<code>\$this->redirectToRoute('name', [], 301)</code>	Returns a RedirectResponse with status code
<code>\$this->redirect('http://example.com')</code>	Returns a RedirectResponse to an external URL
<code>\$this->createNotFoundException(\$msg)</code>	Returns a NotFoundHttpException
<code>\$this->createForm(FormType::class, \$data)</code>	Creates a new Form instance based on the type

Form Methods

Method	Description
<code>\$form->handleRequest(\$request)</code>	Handle a form submission
<code>\$form->isSubmitted()</code>	Check if the form was submitted
<code>\$form->isValid()</code>	Check if the form is valid

Form Fields: Text Fields

Type	Description
TextType	A basic text input field
TextareaType	A multi-line textarea field
EmailType	An email field (<input type="email">)
PasswordType	A password field
SearchType	A search field
UrlType	A URL field
TelType	A tel (phone number) input field
UuidType	A UUID field

Form Fields: Numeric Fields

Type	Description
IntegerType	A number field for integers
MoneyType	A money field, with a specifiable currency
NumberType	A number field
PercentType	A number field for percentages
RangeType	A range input field (slider)

Form Fields: Choice Fields

Type	Description
ChoiceType	Field for selecting one or more options
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

The expanded and multiple options decide whether a dropdown, radio buttons or checkboxes are rendered.

Form Fields: Symfony UX Fields

Type	Description
CropperType	A Cropper.js image cropper field
DropzoneType	A Dropzone file upload field

Form Fields: Date and Time Fields

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

Form Fields: Other Fields

Type	Description
CheckboxType	A single checkbox field
FileType	A file upload field
RadioType	A radio button field
HiddenType	A hidden field

Form Fields: Field Groups

Type	Description
CollectionType	A group of fields that can be added or removed
RepeatedType	A field that is repeated (e.g. password confirmation)

See <https://symfony.com/doc/current/reference/forms/types.html> for a full list of form types.

Validation: Basic Constraints

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

Attribute	Description
<code>#[Assert\NotBlank]</code>	Value is not an empty string/array, false or null
<code>#[Assert\Blank]</code>	Value is an empty string or null
<code>#[Assert\NotNull]</code>	Value is not strictly equal to null
<code>#[Assert\IsNull]</code>	Value is exactly equal to null
<code>#[Assert\Type('string')]</code>	Value is of a specific type

Validation: Comparison Constraints

Attribute	Description
<code>#[Assert\<Not>EqualTo('foo')]</code>	Value is (not) equal to another value
<code>#[Assert\<Not>IdenticalTo('foo')]</code>	Value is (not) identical to another value
<code>#[Assert\LessThan<OrEqual>(5)]</code>	Value is less than (or equal to) another value
<code>#[Assert\GreaterThan<OrEqual>(5)]</code>	Value is greater than (or equal to) another value
<code>#[Assert\Range(min: 2, max: 10)]</code>	Number or DateTime object is within a range

Validation: String Constraints

Attribute	Description
<code>#[Assert\Length(2, 10)]</code>	String is between <code><min></code> and <code><max></code> characters long
<code>#[Assert\Email]</code>	String is a valid email address
<code>#[Assert\Url]</code>	String is a valid URL
<code>#[Assert\Hostname]</code>	String is a valid hostname
<code>#[Assert\Ip]</code>	String is a valid IP address
<code>#[Assert\NoSuspiciousCharacters]</code>	String does not contain spoofing characters
<code>#[Assert\Uuid]</code>	String is a valid UUID
<code>#[Assert\CssColor]</code>	String is a valid CSS color

Validation: Password Constraints

Attribute	Description
<code>#[Assert\UserPassword]</code>	String is the authenticated user's password
<code>#[Assert\NotCompromisedPassword]</code>	Password is not found in any data breaches
<code>#[Assert>PasswordStrength]</code>	Password has a enough entropy

Validation: Number Constraints

Attribute	Description
<code>#[Assert\Positive<OrZero>]</code>	Value is positive (or zero)
<code>#[Assert\Negative<OrZero>]</code>	Value is negative (or zero)

Validation: Date and Time Constraints

Attribute	Description
<code>#[Assert\Date]</code>	Value is a valid date string in Y-m-d format
<code>#[Assert\DateTime]</code>	Value is a valid datetime in the specified format
<code>#[Assert\Time]</code>	Value is a valid time string in H:i:s format
<code>#[Assert\Timezone]</code>	Value is a valid timezone identifier

Validation: Choice Constraints

Attribute	Description
<code>#[Assert\Choice(['A', 'B', 'C'])]</code>	Value is one of the specified choices
<code>#[Assert\Country]</code>	Value is a valid ISO 3166-1 country code
<code>#[Assert\Language]</code>	Value is a valid two-letter language code
<code>#[Assert\Locale]</code>	Value is a valid locale identifier

Validation: File Constraints

Attribute	Description
<code>#[Assert\File]</code>	Value is a valid file path or File object
<code>#[Assert\Image]</code>	Same as File, but only accepts image MIME types

Validation: Financial and other Number Constraints

Attribute	Description
<code>#[Assert\Bic]</code>	Value is a valid Business Identifier Code (BIC)
<code>#[Assert\CardScheme]</code>	Value is a valid credit card number
<code>#[Assert\Currency]</code>	Value is a valid 3-letter ISO 4217 currency
<code>#[Assert\Luhn]</code>	Value passes the Luhn algorithm
<code>#[Assert\Iban]</code>	Value is a valid IBAN
<code>#[Assert\Isbn]</code>	Value is a valid ISBN

See <https://symfony.com/doc/current/reference/constraints.html> for a full list of validation constraints.

Symfony Twig Extensions

On top of the default Twig functions and filters, Symfony provides additional functions and filters. See https://symfony.com/doc/current/reference/twig_reference.html for a full reference.

Twig: Routing Functions

Function	Description
<code>path('home')</code>	Returns the relative URL for a route
<code>path('blog_post', {id: 3})</code>	Returns the relative URL for a route with parameters
<code>url('home')</code>	Returns the absolute URL for a route
<code>url('blog_post', {id: 3})</code>	Returns the absolute URL for a route with params
<code>url('home', [], true)</code>	Returns a scheme-relative URL for a route
<code>absolute_url('/blog/3')</code>	Returns the absolute URL for a path
<code>relative_path('/blog/3')</code>	Returns the relative URL for a path

Twig: Form Functions

Function	Description
<code>form(form)</code>	Renders the complete form
<code>form(form, {'method': 'GET'})</code>	Renders the form, overriding some options
<code>form_start(form)</code>	Renders the opening <code><form></code> tag
<code>form_end(form)</code>	Renders the closing <code></form></code> tag
<code>form_label(field)</code>	Renders the label for the given field
<code>form_help(field)</code>	Renders the help text for the given field
<code>form_errors(field)</code>	Renders any errors for the given field
<code>form_widget(field)</code>	Renders the HTML widget for the given field
<code>form_row(field)</code>	Renders the complete row for the given field, including the label, errors, help and widget
<code>form_rest(form)</code>	Renders all fields that have not been rendered yet

Twig: Form Tags

Tag	Description
<code>{% form_theme form 'bootstrap_5_layout.html.twig' %}</code>	Apply a form theme to a single form

Twig: Security Functions

Function	Description
<code>is_granted('ROLE_USER')</code>	Check if the current user has a role
<code>is_granted('edit', post)</code>	Check if the current user can edit a post
<code>csrf_token('delete-item')</code>	Generate a CSRF token for a specific intention
<code>logout_path('main')</code>	Generate a relative logout URL for the given firewall
<code>logout_url('main')</code>	Generate an absolute logout URL for the given firewall

Twig: Asset Functions

Function	Description
<code>asset('style.css')</code>	Returns the public path for an asset
<code>asset_version()</code>	Returns the asset version
<code>importmap()</code>	Outputs the importmap

Twig: Translation Filters

Filter	Description
<code>'foo' trans</code>	Translate a message
<code>'foo' trans({name: 'value'})</code>	Translate a message with parameters
<code>'foo' trans({name: 'value'}, 'app')</code>	Translate a message from a specific domain
<code>'foo' trans({}, null, 'nl')</code>	Translate a message to a specific locale

Twig: Translation Tags

Tag	Description
<code>{% trans %}message{% endtrans %}</code>	Translate a translation message in the default locale
<code>{% trans with {'name': 'value'} %}...</code>	Translate a message with parameters
<code>{% trans from 'app' %}...</code>	Translate a message from a specific domain
<code>{% trans into 'nl' %}...</code>	Translate a message to a specific locale
<code>{% trans_default_domain 'app' %}</code>	Set the default translation domain for this template

Twig: String Filters

Filter	Description
<code>'dateOfBirth' humanize</code>	Transform a string into a human-readable format
<code>':+1:' emojify</code>	Convert emoji shortcodes to Unicode emojis

Console: Basic Usage

Command	Description
<code>bin/console</code>	List all available commands
<code>bin/console <namespace></code>	List all available commands in a namespace / bundle
<code>bin/console <command></code>	Run a command
<code>bin/console help <command></code>	Display help for a command
<code>bin/console --env=<env></code>	Run the console in dev/test/prod
<code>bin/console -v vv vvv</code>	Increase output verbosity
<code>bin/console cache:clear</code>	Clear the cache
<code>bin/console cache:warmup</code>	Warm up an empty cache

Console: Debugging Commands

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

Console: Linting Commands

Command	Description
<code>bin/console lint:container</code>	Check service arguments for type mismatches
<code>bin/console lint:twig <directory></code>	Lint Twig templates for syntax errors
<code>bin/console lint:yaml <directory></code>	Lint YAML files for syntax errors
<code>bin/console lint:xliff <directory></code>	Lint XLIFF translation files for syntax errors

Console: AssetMap Commands

Command	Description
<code>bin/console asset-map:compile</code>	Compile assets for production
<code>bin/console importmap:require <package></code>	Add an NPM package to the import map
<code>bin/console importmap:install</code>	Download all importmap packages
<code>bin/console importmap:update</code>	Update all importmap packages
<code>bin/console importmap:remove <package></code>	Remove a package from the importmap
<code>bin/console importmap:outdated</code>	List outdated importmap packages
<code>bin/console importmap:audit</code>	Check dependencies for vulnerabilities

Console: Doctrine Commands

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

Console: Autocompletion

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