



Symfony Cheat Sheet

Course	Vide
Confidence	<div><div></div> Confident</div>
Next Review	Vide
Materials	Vide
Last Edited	9 novembre 2023 20:42
Person	Jihed Bouazizi

```
//Created by: Jihed Bouazizi //Creating a project symfony new FirstProject
--version=5.4 --webapp composer require doctrine/annotations //Starting &
Shutting down a server symfony server:start symfony server:stop //Creating
a controller symfony console make:controller ServiceController /*Doctrine
& Database To create a database first you need to modify .env with the root
and the database name you chose. Example: Library 5 is the database name
*/ DATABASE_URL="mysql://root@localhost/library5" //Then creating the
database in xampp symfony console doctrine:database:create //First
migration command symfony console make:migration //Updating the database
command symfony console doctrine:migrations:migrate //Canceling the last
update symfony console doctrine:migrations:migrate prev //Entity
//Creating & Updating Entity symfony console make:entity /*The entity
Repository will be generated automatically in src/Repository*/ //Form
//Creating a form symfony console make:form FormName //More about the form
down bellow //CRUD //Your Controller MUST contain these Uses: use
Doctrine\Persistence\ManagerRegistry; use
Symfony\Component\HttpFoundation\Request; use App\Form\FORM_NAME; use
App\Entity\ENTITY_NAME; use App\Repository\REPOSITORY_NAME; /*For this
example we're using the author workshop Link:
https://drive.google.com/file/d/1cYB47k98BxEydHVTlXbdjQUd3e7_lfCl/view */
//Creating /*Parameters @ Route path: /author/form < @ Route name:
author_add @ Class/Entity name: Author @ Reading page: author_show @ Form
path: templates/author/form.html.twig @ Form Name: formA */ #
[Route('/author/form', name: 'author_add')] public function
AddAuthor(ManagerRegistry $doctrine, Request $request): Response { $author
=new Author(); $form=$this->createForm(AuthorFormType::class,$author);
$form->handleRequest($request); if($form->isSubmitted()){ $em= $doctrine-
>getManager(); $em->persist($author); $em->flush(); return $this->
redirectToRoute('author_show'); } return $this-
>render('author/form.html.twig',[ 'formA'=>$form->createView(), []]; }
/*Taking the formA variable and using it to make the form Form path:
templates/author/form.html.twig*/ {{form_start(formA)}}
{{form_label(formA.email,"Your e-mail:")}} {{form_widget(formA.email)}}
{{form_label(formA.Username,"Your Username:")}}
{{form_widget(formA.Username)}} {{form_label(formA.age,"Your Age:")}}
{{form_widget(formA.age)}} {{form_end(formA)}}
```

Your E-mail:
 Your Username:
 Your Age:

```
//-----// /*Reading Parameters @
Route path: /author/show @ Route name: author_show @ Class/Entity name:
Author @ Reading page: author_show @ List path:
templates/author/authorlist.html.twig */ #[Route('/author/show', name:
'author_show')] public function show(AuthorRepository $rep): Response {
$authors = $rep->findAll(); return $this-
>render('author/authorlist.html.twig', ['authors'=>$authors]); } /*List
path: templates/author/authorlist.html.twig The list will use a For loop
to list the authors plus showing the update & delete button following
these 2 lines: <a href="{{path('author_update',
{'id':author.id})}}">Update</a> <a href="{{path('author_delete',
{'id':author.id})}}">Delete</a> NOTE: Use the path names and not the
routes (author_delete/author_update) which will redirect us to either
deleting or updating the author. An add button will also be added with the
following line in case you wanted to insert another author. <a href="{{
path('author_add')}}">Add</a> */ {% block body %} <h1>Liste des
Auteurs</h1> <table> <thead> <tr> <th>Nom d utilisateur</th>
<th>Email</th> <th>Actions</th> </tr> </thead> <tbody> {% for author in
authors %} <tr> <td>{{ author.username }}</td> <td>{{ author.email }}</td>
<td> <a href="{{path('author_update',{'id':author.id})}}">Update</a> | <a
href="{{path('author_delete',{'id':author.id})}}">Delete</a> </td> </tr>
{% endfor %} </tbody> </table> <a href="{{path('author_add')}}">Add</a>
{% endblock %}
```

127.0.0.1:8000/author/show

Liste des Auteurs

Nom d'utilisateur	Email	Actions
Jihed Bouazizi	Jihed.bouazizi@esprit.tn	Update Delete
Add		



```
//-----// /*Updating Parameters @
Route path: /author/update/{id} @ Route name: author_update @ Class/Entity
name: Author @ Reading page: author_show @ Form path:
templates/author/form.html.twig @ Form Name: formA */ /* Updating needs an
ID, putting the one you need to update in {id} will automatically Create
the form and fill it with the retrieved data, all you need to do is
changing whatever you want and pressing the update button*/ #
[Route('/author/update/{id}', name: 'author_update')] public function
UpdateAuthor(ManagerRegistry $doctrine, Request $request, AuthorRepository
$rep, $id): Response { $author = $rep->find($id); $form=$this-
>createForm(AuthorFormType::class,$author); $form-
>handleRequest($request); if($form->isSubmitted()){ $em= $doctrine-
>getManager(); $em->persist($author); $em->flush(); return $this->
redirectToRoute('author_show'); } return $this-
>render('author/form.html.twig',[ 'formA'=>$form->createView(), ]); }
/*Update path: templates/author/form.html.twig Same form as adding */ //--
-----// /*Deleting Parameters @
Route path: /author/delete/{id} @ Route name: author_delete @ Class/Entity
```