Create a full Todo Application with Symfony from Scratch

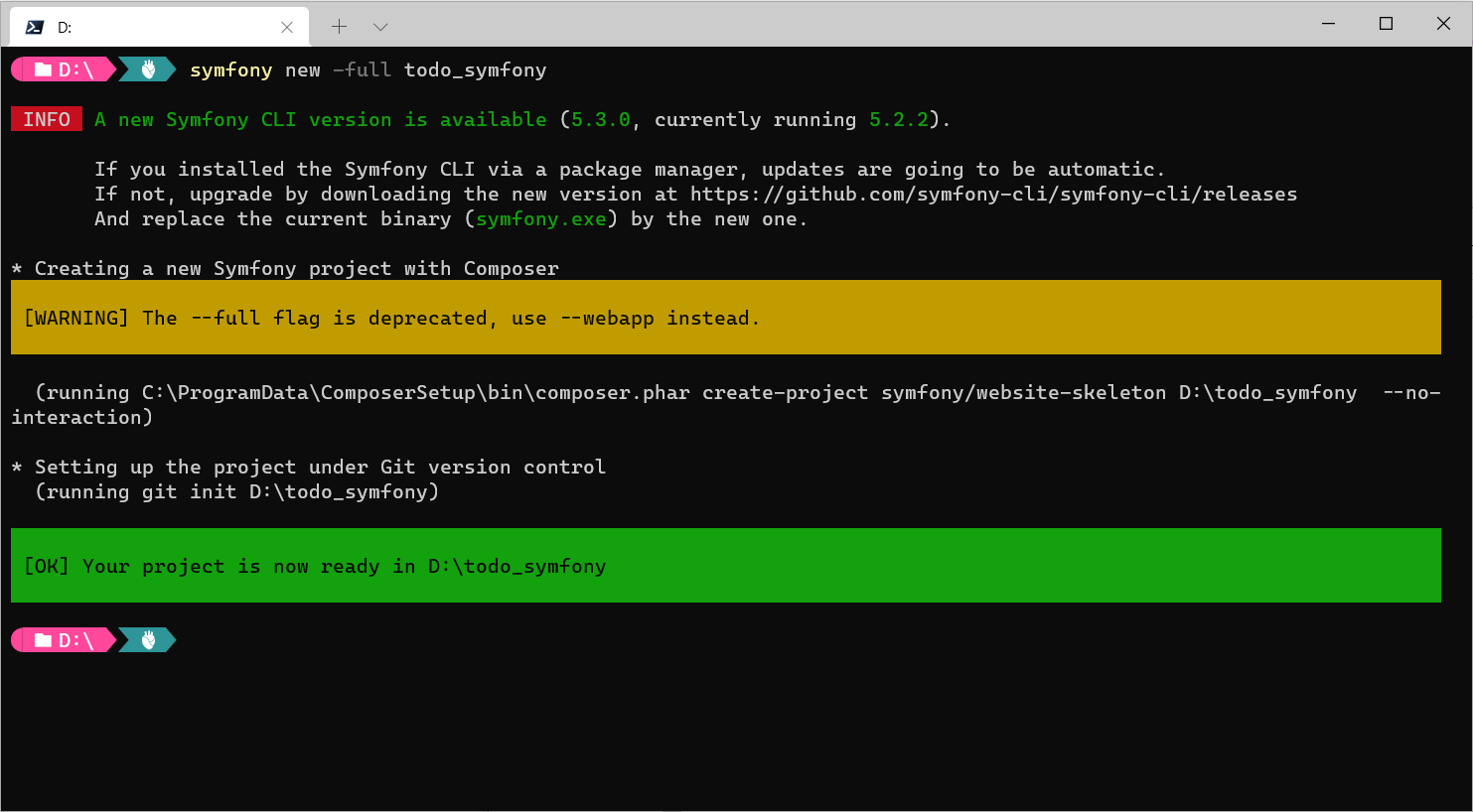
# Setup the Environment Development

First download symfony CLI from <https://symfony.com/download>

After that you can use the “**symfony**” command to create project

Create new Symfony project:

|  |
| --- |
| symfony new –full todo\_symfony |

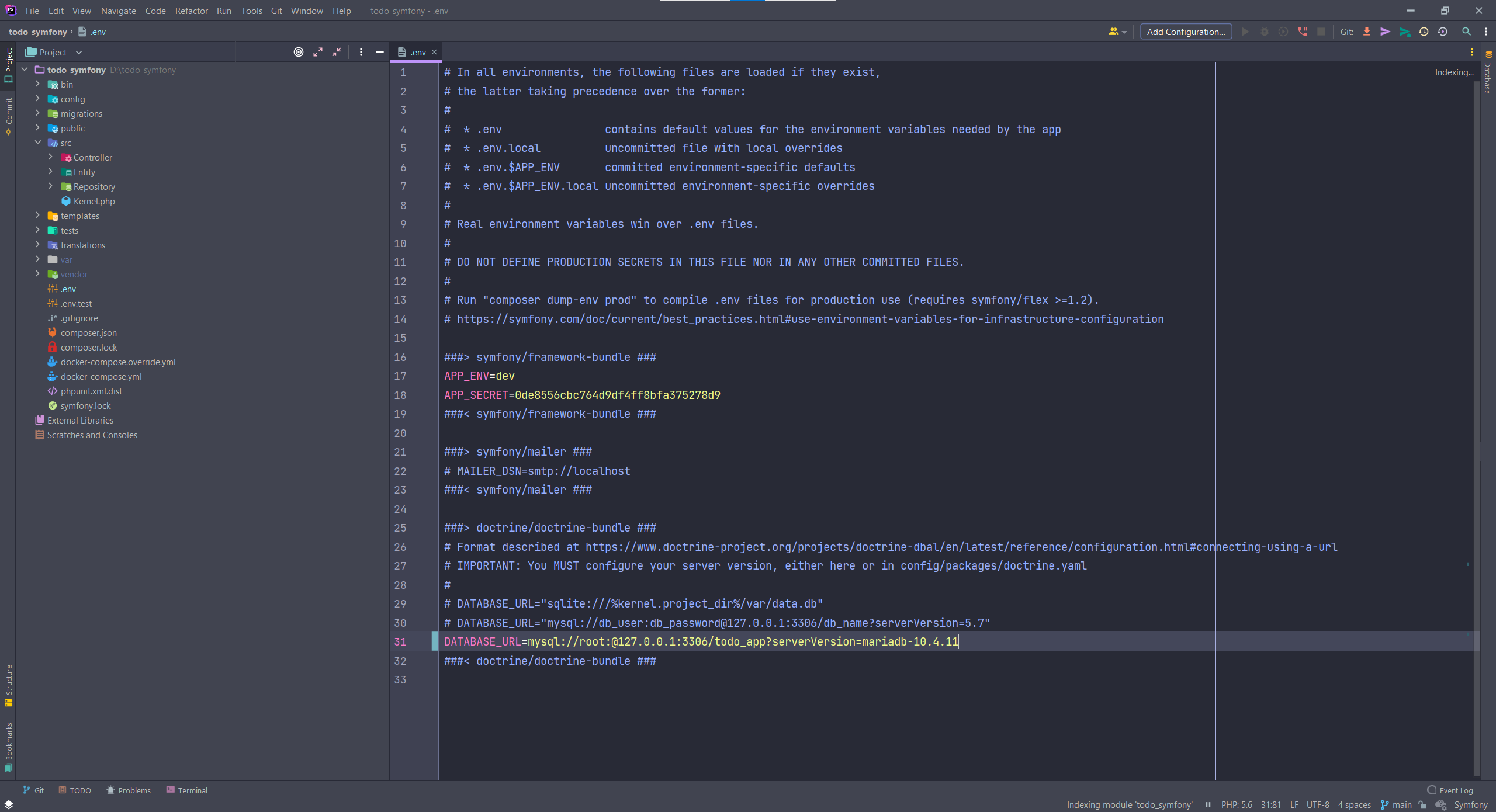


Go to the project folder

|  |
| --- |
| cd todo\_symfony |

Modify the **DATABASE\_URL** environment variable in the **.env** file connect with your database, for example if you use localhost mysql with username **root**, password **empty**, database name **basic\_crud\_db**

|  |
| --- |
| DATABASE\_URL=mysql://root:@127.0.0.1:3306/todo\_app?serverVersion=mariadb-10.4.11 |

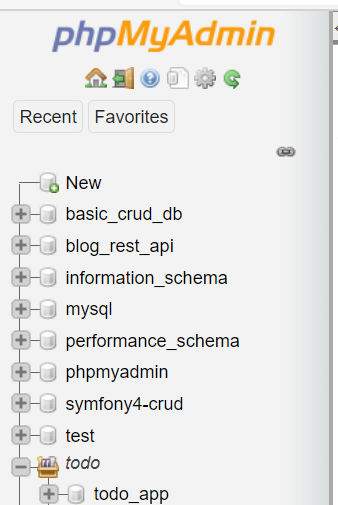


You can create the database directly in your database client interface, or create by Symfony command line:

|  |
| --- |
| php bin/console doctrine:database:create |

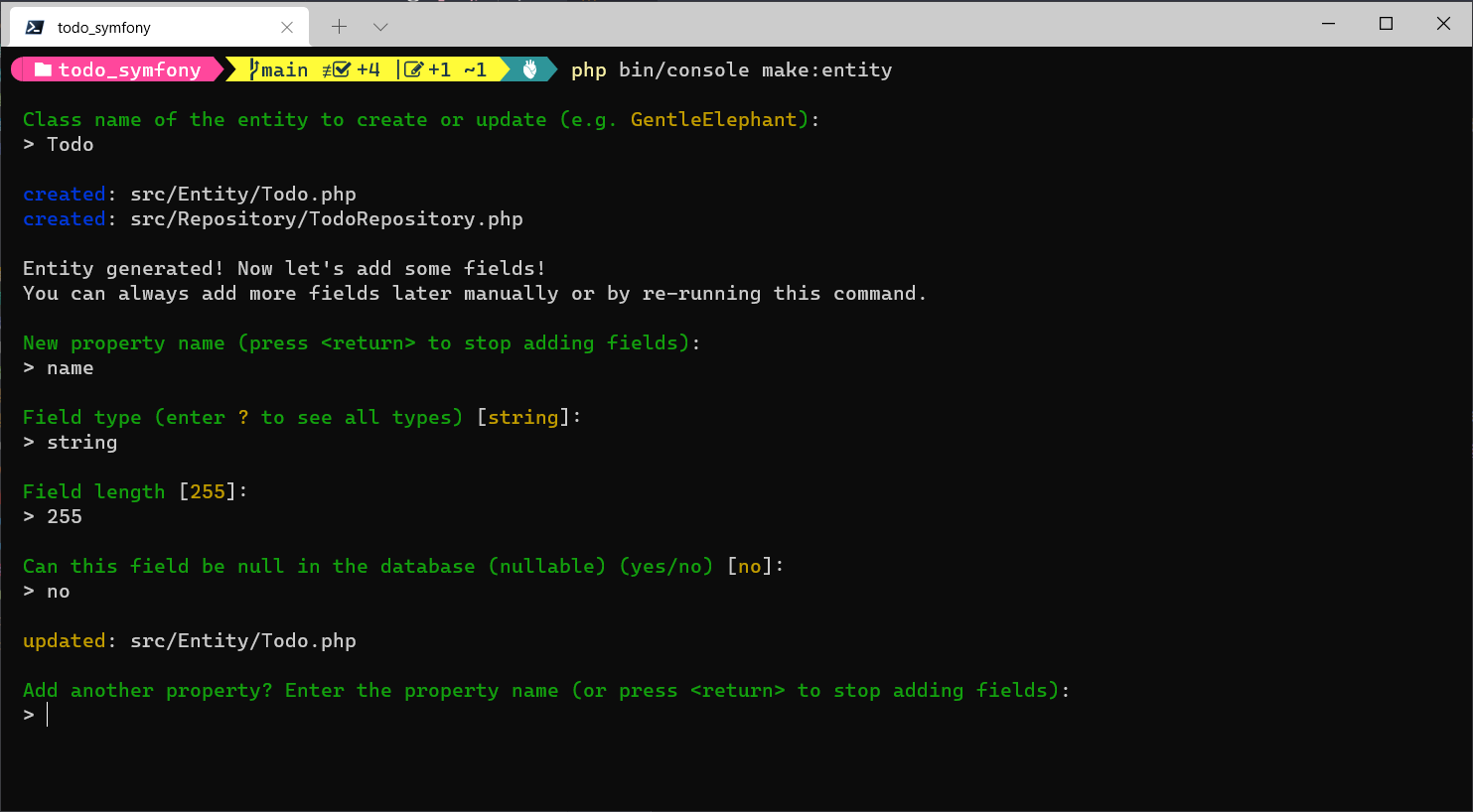


You can see database todo\_app is created in phpMyAdmin

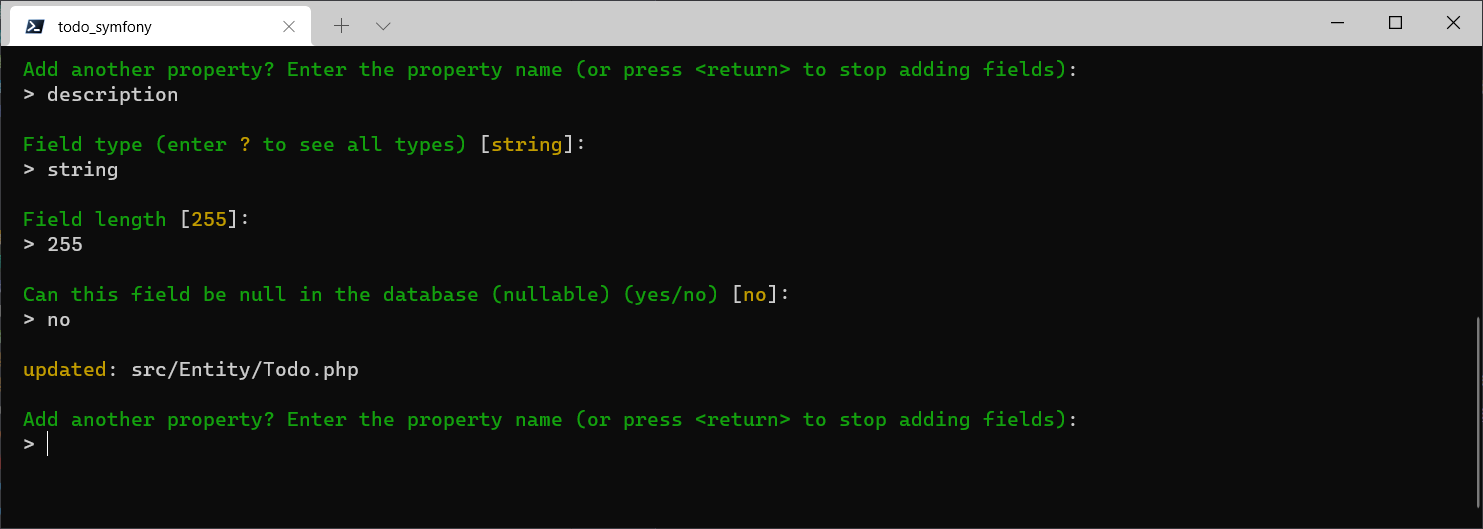


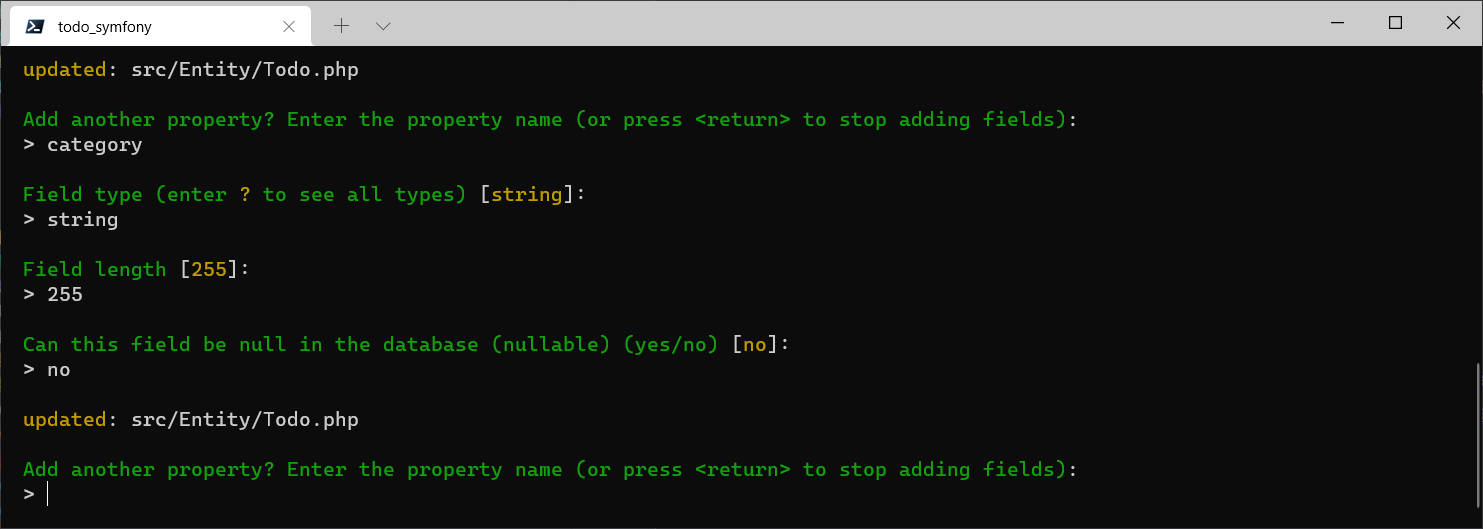
Now let’s make simple Todo entity

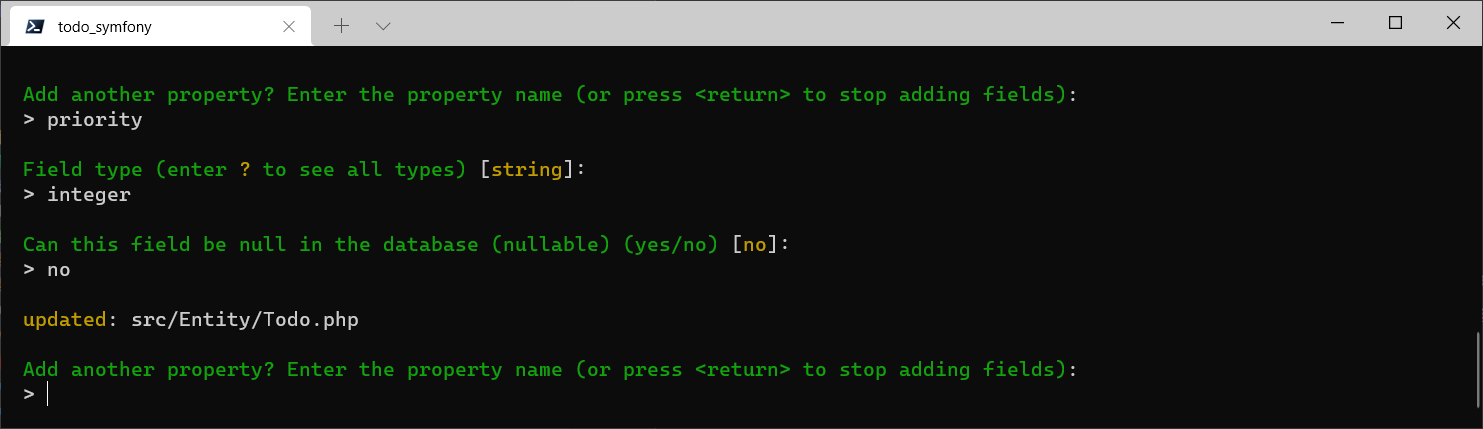
|  |
| --- |
| php bin/console make:entity |

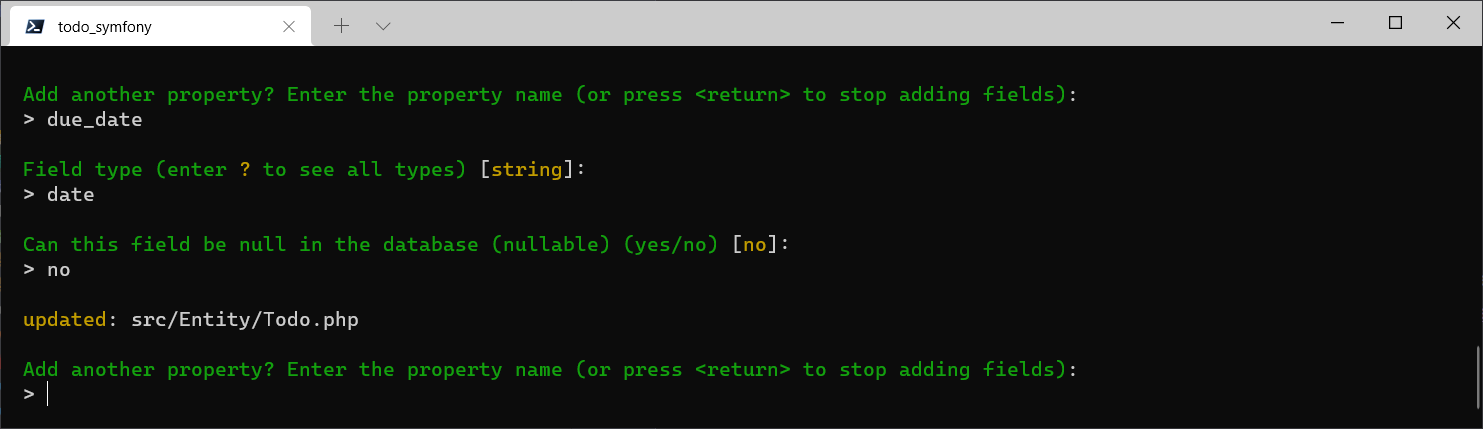


Then, you will add more properties as full (category, description, priority, due\_date)



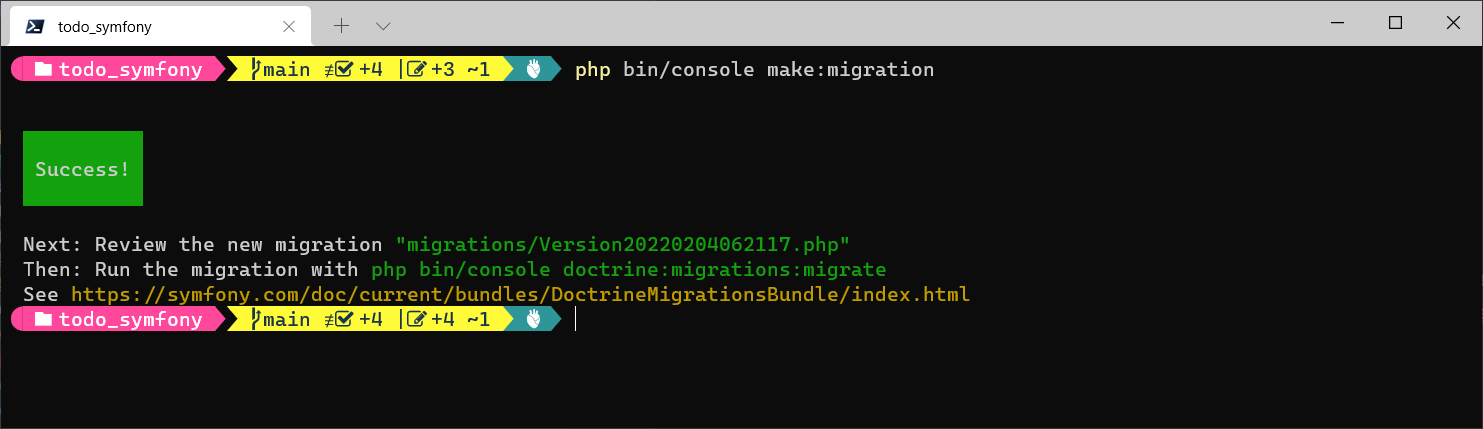






Then, you will Create a migration file by using following command:

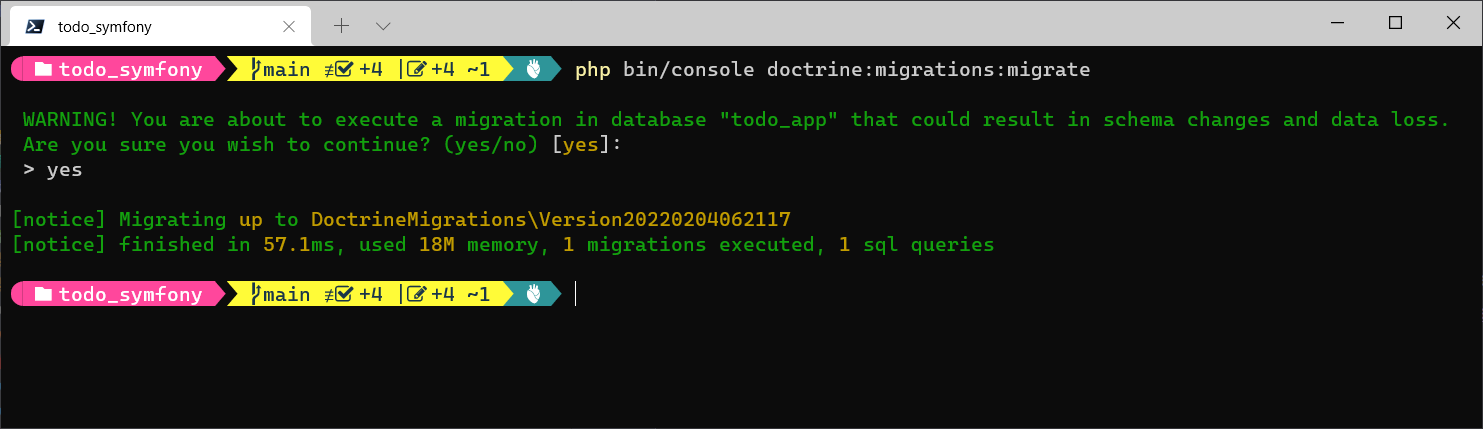
|  |
| --- |
| php bin/console make:migration |



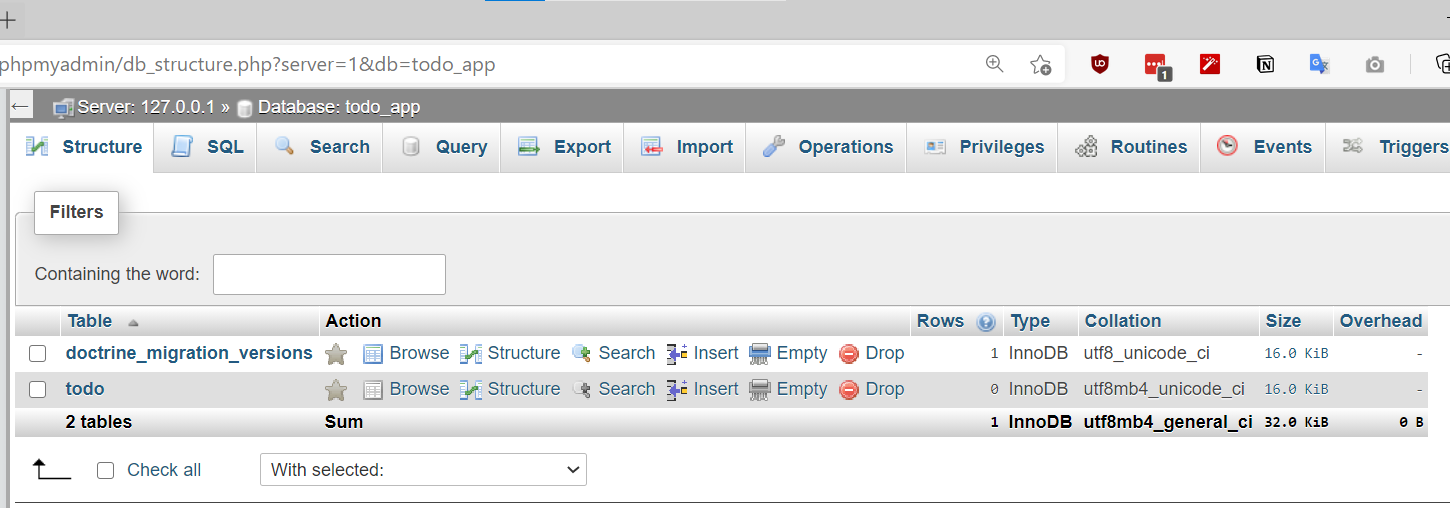
The make:migration will create an instruction file contain SQL query

In order to create the table, you have to run following command

|  |
| --- |
| php bin/console doctrine:migrations:migrate |



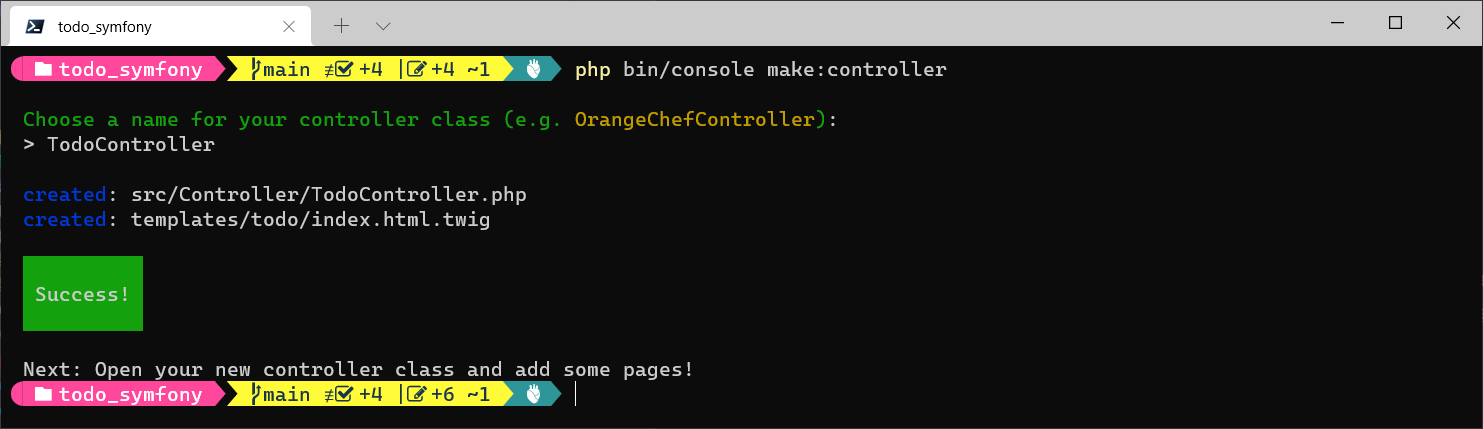
Now new table “todo” already been created in our todo\_app database



# Controller

Then, it’s time to make the first controller. To create the controller, you have to run the following command

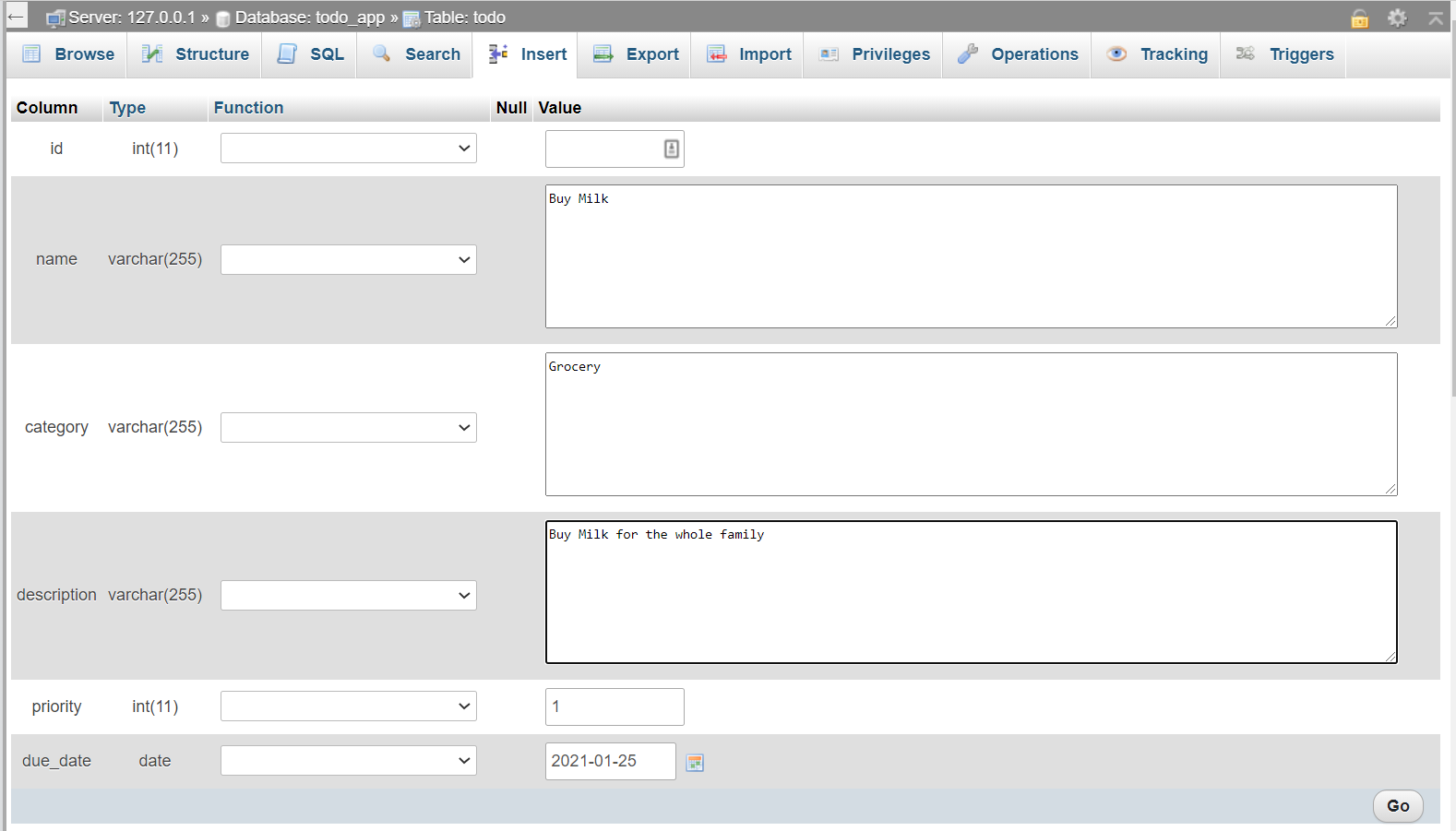
|  |
| --- |
| php bin/console make:controller |



## listAction

|  |
| --- |
| /\*\*  \* @Route("/todo", name="todo\_list")  \*/ public function listAction**() {** $todos = $this->getDoctrine**()** ->getRepository**(**'App:Todo'**)** ->findAll**()**;  return $this->render**(**'todo/index.html.twig', **[** 'todos' => $todos  **])**; **}** |

This action will dislay all Todos of the table in the database. Before that, you have to create some Todos in Phpmyadmin



# View

## Base.html.twig

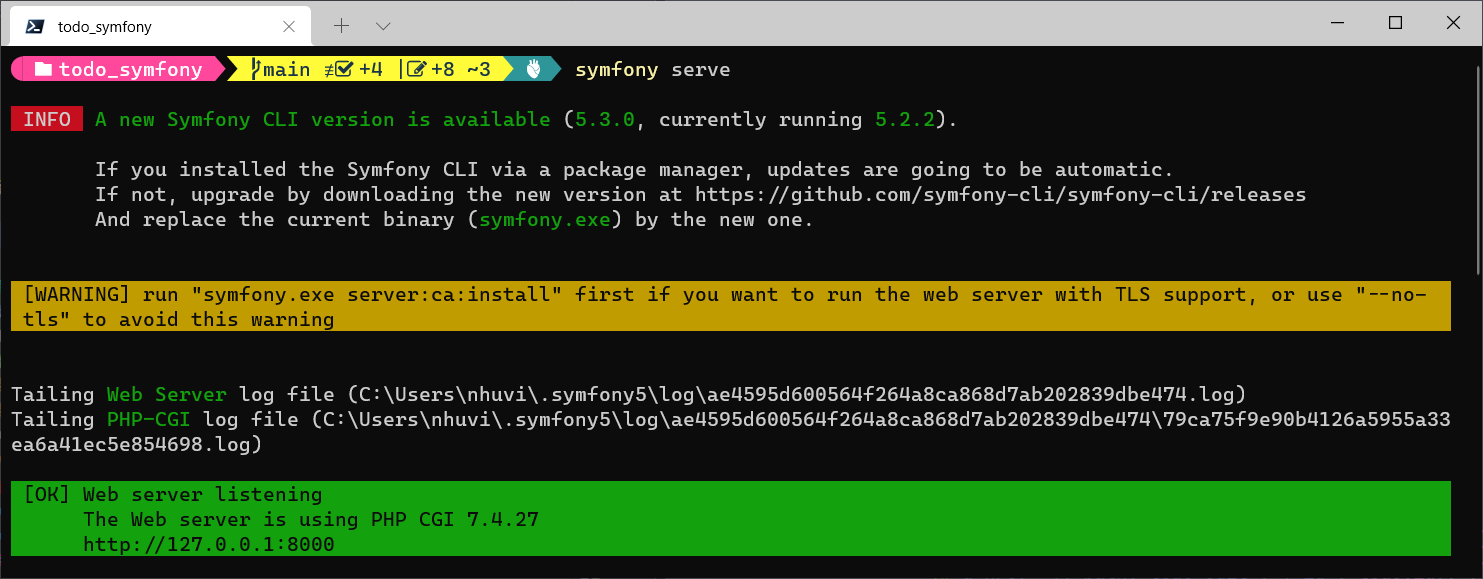
The base.html.twig must be set in order to reuse for other views

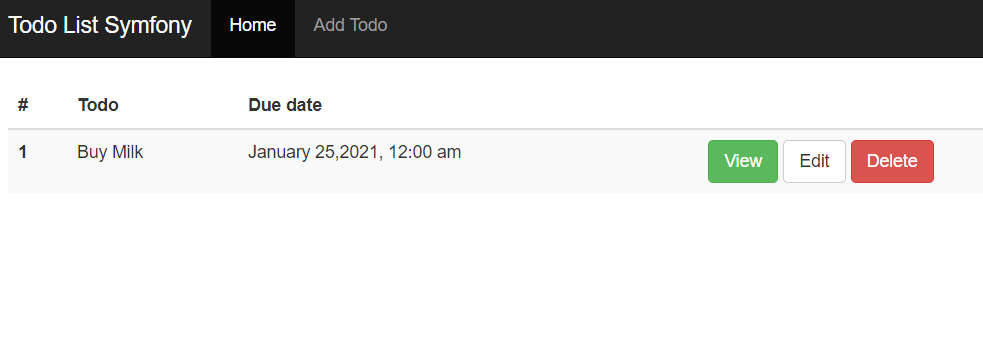
|  |
| --- |
| **<!doctype** html**> <**html lang="en"**> <**head**>  <**meta charset="utf-8"**>  <**meta http-equiv="X-UA-Compatible" content="IE=edge"**>  <**meta name="viewport" content="width=device-width, initial-scale=1"**>** <!-- The above 3 meta tags \*must\* come first in the head; any other head content must come \*after\* these tags -->  **<**meta name="description" content=""**>  <**meta name="author" content=""**>  <**link rel="icon" href="../../favicon.ico"**>  <**link rel="canonical" href="https://getbootstrap.com/docs/3.4/examples/starter-template/"**>  <**title**>{%** block title **%}**Welcome!**{%** endblock **%}</**title**>** <!-- Bootstrap core CSS -->  **<**link href="https://stackpath.bootstrapcdn.com/bootstrap/3.4.1/css/bootstrap.min.css" rel="stylesheet"**>  {%** block stylesheets **%}{%** endblock **%} </**head**> <**body**> <**nav class="navbar navbar-inverse"**>  <**div class="container"**>  <**div class="navbar-header"**>  <**button type="button" class="navbar-toggle collapsed" data-toggle="collapse" data-target="#navbar"  aria-expanded="false" aria-controls="navbar"**>  <**span class="sr-only"**>**Toggle navigation**</**span**>  <**span class="icon-bar"**></**span**>  <**span class="icon-bar"**></**span**>  <**span class="icon-bar"**></**span**>  </**button**>  <**a class="navbar-brand" href="#"**>**Todo List Symfony**</**a**>  </**div**>  <**div id="navbar" class="collapse navbar-collapse"**>  <**ul class="nav navbar-nav"**>  <**li class="active"**><**a href="#"**>**Home**</**a**></**li**>  {#**<li><a href="{{ path('todo\_create') }}">Add Todo</a></li>**#}  <**li**><**a href="#"**>**Add Todo**</**a**></**li**>  </**ul**>  </**div**>**<!--/.nav-collapse -->  **</**div**> </**nav**>  <**div class="container"**>  <**div class="row"**>  <**div class="col-md-12"**>  {%** for flash\_message in app.session.flashBag.get**(**'notice'**) %}  <**div class="alert alert-success"**>{{** flash\_message **}}</**div**>  {%** endfor **%}  {%** for flash\_message in app.session.flashBag.get**(**'error'**) %}  <**div class="alert alert-danger"**>{{** flash\_message **}}</**div**>  {%** endfor **%}  {%** block body **%}{%** endblock **%}  </**div**>  </**div**> </**div**> </**div**>**<!-- /.container -->  **{%** block javascripts **%}{%** endblock **%}  </**body**> </**html**>** |

Then, you will have to display all Todos in the *Todo/Index.html.twig*

|  |
| --- |
| **{%** extends 'base.html.twig' **%}  {%** block body **%}  <**div class="bs-example" data-example-id="striped-table"**>  <**table class="table table-striped"**>  <**thead**>  <**tr**>  <**th**>**#**</**th**>  <**th**>**Todo**</**th**>  <**th**>**Due date**</**th**>  <**th**></**th**>  </**tr**>  </**thead**>  <**tbody**>  {%** for todo in todos **%}  <**tr**>  <**th scope="row"**>{{** todo.id **}}</**th**>  <**td**>{{** todo.name **}}</**td**>  <**td**>{{** todo.dueDate|date**(**'F j,Y, g:i a'**) }}</**td**>  <**td**>  {#**<a href="{{ path('todo\_details', {'id': todo.id}) }} " class='btn btn-success'>View</a>**#}  {#**<a href="{{ path('todo\_edit', {'id': todo.id}) }}" class='btn btn-default'>Edit</a>**#}  {#**<a href="{{ path('todo\_delete', {'id': todo.id}) }}" class='btn btn-danger'>Delete</a>**#}  <**a href="#" class='btn btn-success'**>**View**</**a**>  <**a href="#" class='btn btn-default'**>**Edit**</**a**>  <**a href="#" class='btn btn-danger'**>**Delete**</**a**>  </**td**>  </**tr**>  {%** endfor **%}  </**tbody**>  </**table**>  </**div**> {%** endblock **%}** |

Then, run the Server with symfony serve to see the *Todo/Index.html.twig* page





## detailsAction

The code below will help to perform the detailsAction of the Controller. It will help to retrieve the information of a specific Todo based on the $id then display its information.

|  |
| --- |
| /\*\*  \* @Route("/todo/details/{id}", name="todo\_details")  \*/ public function detailsAction**(***$id***) {** $todos = $this->getDoctrine**()** ->getRepository**(**'App:Todo'**)** ->find**(***$id***)**;   return $this->render**(**'todo/details.html.twig', **[** 'todos' => $todos  **])**; **}** |

Then, you have to create the view *details.html.twig* to display information of an specific Todo

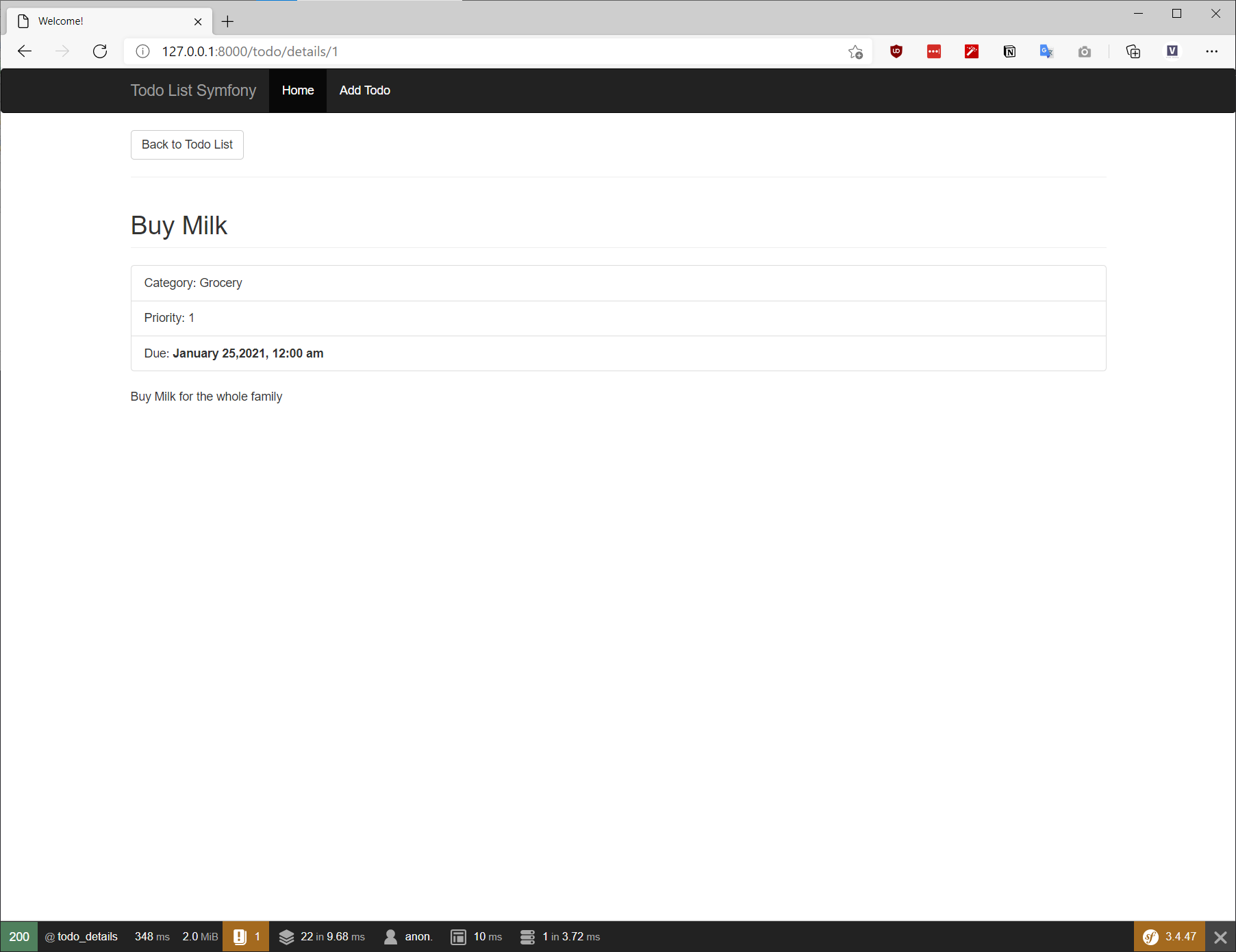
details.html.twig

|  |
| --- |
| **{%** extends 'base.html.twig' **%}  {%** block body **%}  <**a href="**{{** path**(**'todo\_list'**) }}**" class="btn btn-default"**>**Back to Todo List**</**a**>  <**hr**/>  <**h2 class="page-header"**>{{** todos.name **}}</**h2**>  <**ul class="list-group"**>  <**li class="list-group-item"**>**Category: **{{** todos.category **}}</**li**>  <**li class="list-group-item"**>**Priority: **{{** todos.priority **}}</**li**>  <**li class="list-group-item"**>**Due: **<**strong**>{{** todos.dueDate|date**(**'F j,Y, g:i a'**) }}</**strong**></**li**>  </**ul**>  <**p**>{{** todos.description **}}</**p**> {%** endblock **%}** |

In the *index.html.twig,* you will have to remove and replace following lines

<a href="{{ path('todo\_details', {'id': todo.id}) }} " class='btn btn-success'>View</a> (add this line)  
  
 ~~<a href="#" class='btn btn-success'>View</a> (Remove this line)~~

Then, you can see the details of the an Todo



## deleteAction

To delete a Todo, you will implement *deleteAction($id)*

|  |
| --- |
| /\*\*  \* @Route("/todo/delete/{id}", name="todo\_delete")  \*/ public function deleteAction**(***$id***) {** $em = $this->getDoctrine**()**->getManager**()**;  $todo = $em->getRepository**(**'App:Todo'**)**->find**(***$id***)**;  $em->remove**(**$todo**)**;  $em->flush**()**;    $this->addFlash**(** 'error',  'Todo deleted'  **)**;    return $this->redirectToRoute**(**'todo\_list'**)**; **}** |

Then in the *index.html.twig,* you will remove and replace following lines to use **Delete** button

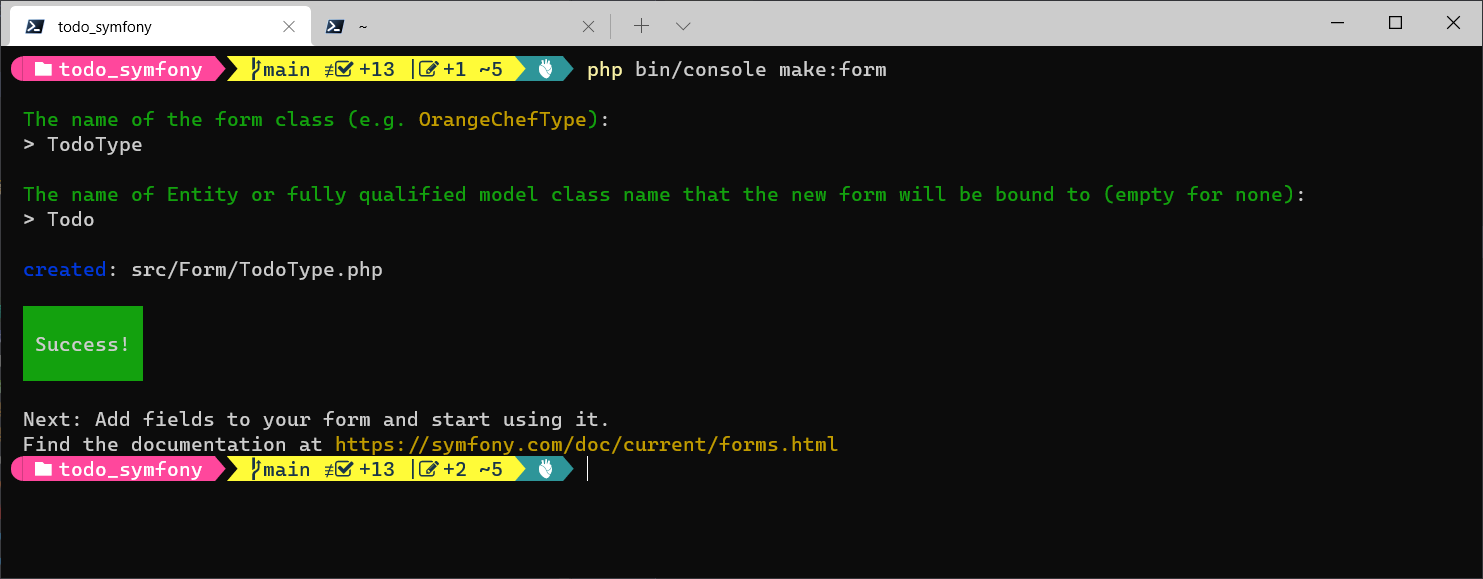
|  |
| --- |
| <a href="{{ path('todo\_delete', {'id': todo.id}) }}" **class**='btn btn-danger'>**Delete**</a>  ~~<a href="#"~~ **~~class~~**~~='btn btn-danger'>~~**~~Delete~~**~~</a> (Remove)~~ |

## createAction

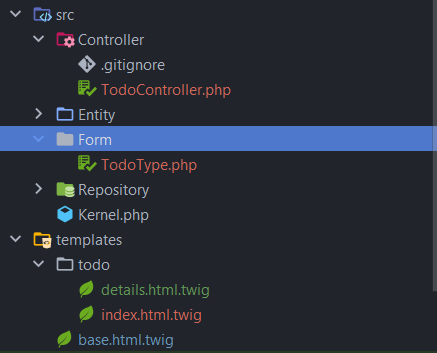
Before create a new Todo, you will have to create a FormType by using

php bin/console make:form

In the terminal, you will config as below:



You can see a file named *TodoType.php* is created in Form folder



You will config the class *TodoType* as below

TodoType.php

|  |
| --- |
| class TodoType extends AbstractType **{** public function buildForm**(**FormBuilderInterface *$builder*, array *$options***)  {** *$builder* ->add**(**'name', TextType::class**)** ->add**(**'category', TextType::class**)** ->add**(**'description', TextareaType::class**)** ->add**(**'priority', ChoiceType::class, **[** 'choices' => **[** '1' => "1",  '2' => "2",  '3' => "3",  **]**,  **])** ->add**(**'due\_date', DateType::class, **[**'widget' => 'single\_text'**])**;  **}** public function configureOptions**(**OptionsResolver *$resolver***)**: void  **{** *$resolver*->setDefaults**([** 'data\_class' => Todo::class,  **])**;  **} }** |

To create a new Todo, you will have to implement *createAction(Request $request)*

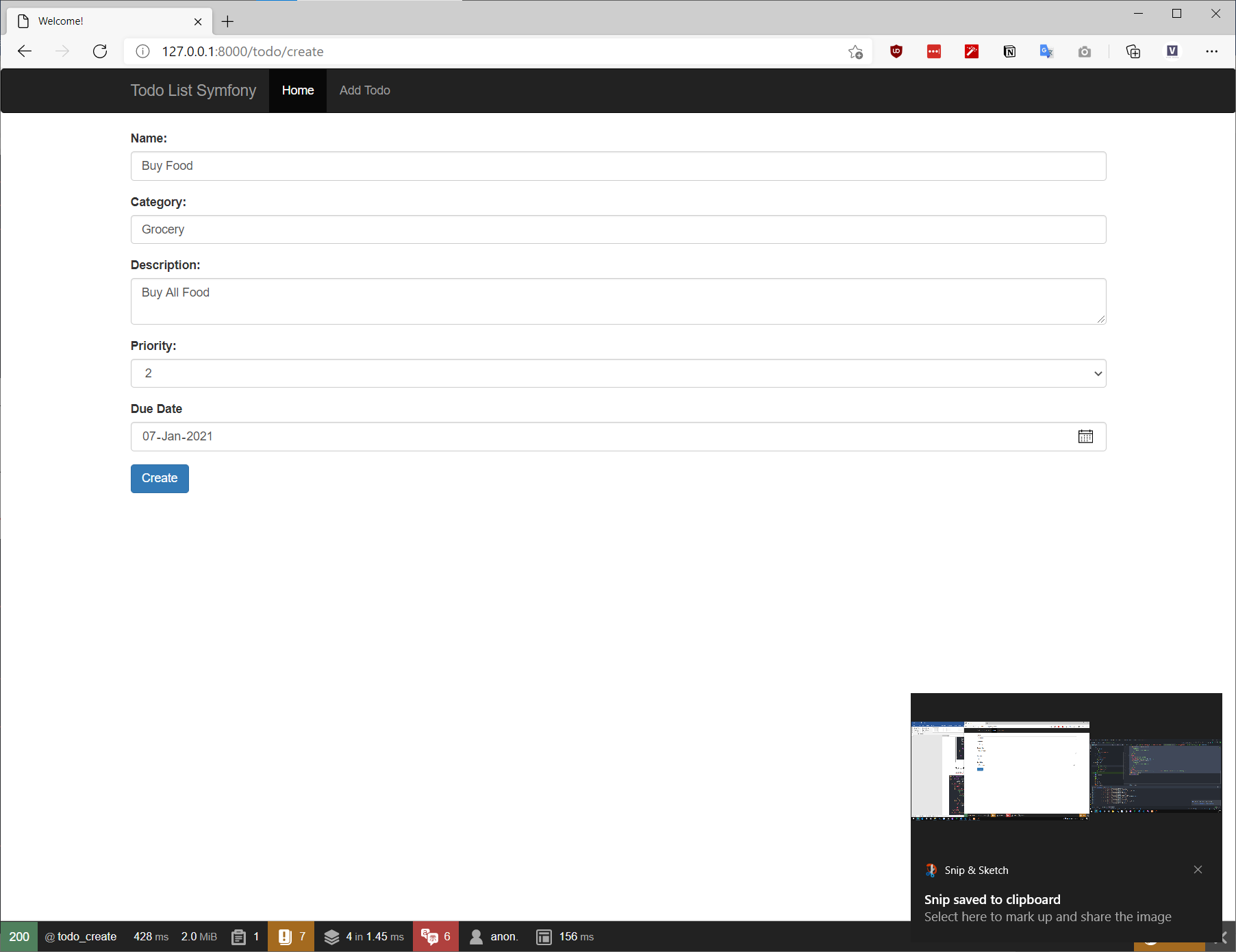
|  |
| --- |
| /\*\*  \* @Route("/todo/create", name="todo\_create", methods={"GET","POST"})  \*/ public function createAction**(**Request *$request***) {** $todo = new Todo**()**;  $form = $this->createForm**(**TodoType::class, $todo**)**;    if **(**$this->saveChanges**(**$form, *$request*, $todo**)) {** $this->addFlash**(** 'notice',  'Todo Added'  **)**;    return $this->redirectToRoute**(**'todo\_list'**)**;  **}** return $this->render**(**'todo/create.html.twig', **[** 'form' => $form->createView**()  ])**; **}** public function saveChanges**(***$form*, *$request*, *$todo***) {** *$form*->handleRequest**(***$request***)**;    if **(***$form*->isSubmitted**()** && *$form*->isValid**()) {** *$todo*->setName**(***$request*->request->get**(**'todo'**)[**'name'**])**;  *$todo*->setCategory**(***$request*->request->get**(**'todo'**)[**'category'**])**;  *$todo*->setDescription**(***$request*->request->get**(**'todo'**)[**'description'**])**;  *$todo*->setPriority**(***$request*->request->get**(**'todo'**)[**'priority'**])**;  *$todo*->setDueDate**(**\DateTime::**createFromFormat(**'Y-m-d', *$request*->request->get**(**'todo'**)[**'due\_date'**]))**;  $em = $this->getDoctrine**()**->getManager**()**;  $em->persist**(***$todo***)**;  $em->flush**()**;    return true;  **}** return false; **}** |

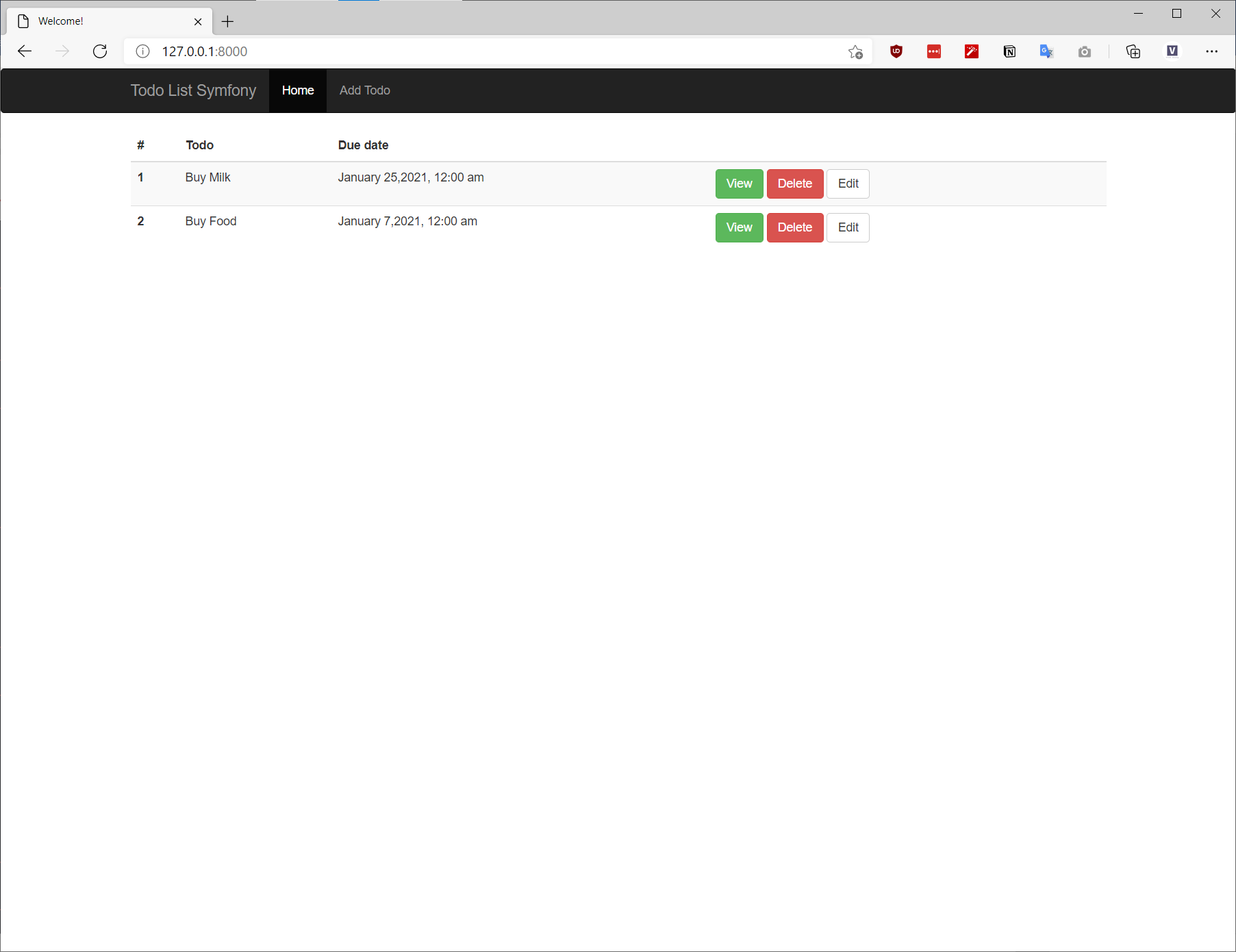
The final step is to create the view for create Action

create.html.twig

|  |
| --- |
| **{%** extends 'base.html.twig' **%} {%** block body **%}  {{** form\_start**(**form**) }}  <**div class="form-group"**>  <**label for="name"**>**Name:**</**label**>  {{** form\_widget**(**form.name,**{** 'attr':**{** 'class':'form-control',  'placeholder': 'Task name'  **}  }) }}  <**div class="invalid-feedback"**>  {{** form\_errors**(**form.name**) }}  </**div**>  </**div**>  <**div class="form-group"**>  <**label for="category"**>**Category:**</**label**>  {{** form\_widget**(**form.category,**{** 'attr':**{** 'class':'form-control',  'placeholder': 'Category name'  **}  }) }}  </**div**>  <**div class="form-group"**>  <**label for="description"**>**Description:**</**label**>  {{** form\_widget**(**form.description,**{** 'attr':**{** 'class':'form-control',  'placeholder': 'Description'  **}  }) }}  </**div**>  <**div class="form-group"**>  <**label for="priority"**>**Priority:**</**label**>  {{** form\_widget**(**form.priority,**{** 'attr':**{** 'class':'form-control'  **}  }) }}  </**div**>  <**div class="form-group"**>  <**label for="due\_date"**>**Due Date**</**label**>  {{** form\_widget**(**form.due\_date,**{** 'attr':**{** 'class':'form-control'  **}  }) }}  </**div**>  <**input type="submit" name="create\_todo" value="Create" class="btn btn-primary"**/>  {{** form\_end**(**form**) }} {%** endblock **%}** |

Then you can start to create a new Todo





## editAction

In the controller, you will start to implement the *editAction($id, Request $request)*

|  |
| --- |
| /\*\*  \* @Route("/todo/edit/{id}", name="todo\_edit")  \*/ public function editAction**(***$id*, Request *$request***) {** $em = $this->getDoctrine**()**->getManager**()**;  $todo = $em->getRepository**(**'App:Todo'**)**->find**(***$id***)**;    $form = $this->createForm**(**TodoType::class, $todo**)**;    if **(**$this->saveChanges**(**$form, *$request*, $todo**)) {** $this->addFlash**(** 'notice',  'Todo Edited'  **)**;  return $this->redirectToRoute**(**'todo\_list'**)**;  **}** return $this->render**(**'todo/edit.html.twig', **[** 'form' => $form->createView**()  ])**; **}** |

Then, you will have to create the view *edit.html.twig* to display the form and submit it to the controller

edit.html.twig

|  |
| --- |
| **{%** extends 'base.html.twig' **%}  {%** block body **%}  {{** form\_start**(**form**) }}  <**div class="form-group"**>  <**label for="name"**>**Name:**</**label**>  {{** form\_widget**(**form.name,**{** 'attr':**{** 'class':'form-control',  'placeholder': 'Task name'  **}  }) }}  </**div**>  <**div class="form-group"**>  <**label for="category"**>**Category:**</**label**>  {{** form\_widget**(**form.category,**{** 'attr':**{** 'class':'form-control',  'placeholder': 'Category name'  **}  }) }}  </**div**>  <**div class="form-group"**>  <**label for="description"**>**Description:**</**label**>  {{** form\_widget**(**form.description,**{** 'attr':**{** 'class':'form-control',  'placeholder': 'Description'  **}  }) }}  </**div**>  <**div class="form-group"**>  <**label for="priority"**>**Priority:**</**label**>  {{** form\_widget**(**form.priority,**{** 'attr':**{** 'class':'form-control'  **}  }) }}  </**div**>  <**div class="form-group"**>  <**label for="due\_date"**>**Due Date**</**label**>  {{** form\_widget**(**form.due\_date,**{** 'attr':**{** 'class':'form-control'  **}  }) }}  </**div**>  <**input type="submit" name="edit\_todo" value="Edit" class="btn btn-primary"**/>  {{** form\_end**(**form**) }} {%** endblock **%}** |

Then, you will remove and replace following lines in the *index.html.twig* to use the Edit button

<a href=**"#"** class=**'btn btn-default'**>**Edit**</a> **(Remove this line)**

<a href=**"{{ path('todo\_edit', {'id': todo.id}) }}"** class=**'btn btn-default'**>**Edit**</a> **(Add this line)**

Then, you can update a Todo

