

Introduction to web development with Symfony2

Notes and updates

Lecture 2: IDE and PHP version

I recommend you to download the 30-day demo version of PhpStorm: <http://www.jetbrains.com/phpstorm/download/index.html>

If you prefer a free IDE, a good choice is the PHP version of NetBeans: <https://netbeans.org/downloads/>

Another free alternative is Zend Eclipse PDT: <http://www.zend.com/en/company/community/pdt/downloads>

Lecture 7: A quick look at the project configuration

Important: You will have to configure PhpStorm preferences again after opening this project for the first time, because several parts are project-specific. To do so, please use the proper lecture from my Installing Symfony2 course again. You can also review the PDF documentation from that course if it is faster for you.

Lecture 13: Generating the Post and Author entities

Course update for Symfony 2.8:

The Doctrine's entities generator has been improved, take into account the following points:

- After entering the field length it will ask if you want it null and unique, accept the default values.
- It does not ask if you want an empty repository class anymore, it's created automatically.
- Some coding standards corrections appearing in the video are now solved by default.
- It does not generate a constructor automatically anymore for *Author*. Copy it from the video.

Lecture 14: Generating our first migration

Course update for Symfony 2.8:

- Set the 1.1 stable version for doctrine-migrations-bundle this way: "doctrine/doctrine-migrations-bundle": "~1.1" (check out the code here: <http://bit.ly/2jIGek2>).
- Composer has been improved since the recording of the video and it is capable to install dependencies automatically. So it is not necessary to add "doctrine/migrations" to install the migrations bundle anymore because it is added and installed automatically.
- If you have prepared your development environment with my *Installing Symfony2* free course you will not see the *timezone* error when creating the first migration. Ignore the video portion from 1:24 to 3:33

Lecture 15: Populating the database with fixtures

Course update for Symfony 2.8:

- Set the 2.3 stable version for *doctrine-fixtures-bundle* this way: "doctrine/doctrine-fixtures-bundle": "~2.3" (check out the code here: <http://bit.ly/2jMp0cZ>).

Lecture 18: Installing PHPUnit and running the suite test

Let's install PHPUnit with **Homebrew** for OS X.

In Linux you can use similar commands using the package manager of your distribution.

In Windows you depend on the tool that you installed in the previous chapter, and you have to use the PHP PEAR package manager. If you have problems at this point I'm sorry not to help, as I don't use Windows.

Lecture 24: Iterating over Twig posts, using partials and the Twig extensions

Course update for Symfony 2.8:

- In order to enable Twig Extensions you have to add the next line in *composer.json*, at the end of the "require" block
 - "twig/extensions": "~1.3" (code: <http://bit.ly/2jtWTOw>)
- Remember to run "composer update".

Lecture 25: Adding a sidebar for the latest posts

Course update for Symfony 2.8:

- Repository classes are now created automatically. You will only have to write methods.

Lecture 26: Installing Doctrine Extensions to add a slug to the post

Course update for Symfony 2.8:

- In order to enable *Doctrine Extensions* you have to add the next line in *composer.json*, just below "doctrine-migrations-bundle":
 - "stof/doctrine-extensions-bundle": "~1.2" (code: <http://bit.ly/2k7VBdU>)
- When running the *slug* migration, you will see an error. This is because it's set as NOT NULL, and since the table already has data, when adding the column it will be obviously *null* by default. Open the latest migration (in app/DoctrineMigrations) and locate this line:
 - ```
$this->addSql('CREATE TABLE post (id INTEGER NOT NULL, author_id INTEGER NOT NULL, createdAt DATETIME NOT NULL, title VARCHAR(150) NOT NULL COLLATE BINARY, body CLOB NOT NULL COLLATE BINARY, slug VARCHAR(255) NOT NULL, PRIMARY KEY(id), CONSTRAINT FK_5A8A6C8DF675F31B FOREIGN KEY (author_id) REFERENCES author (id) NOT DEFERRABLE INITIALLY IMMEDIATE)');
```
  - Notice slug VARCHAR(255) NOT NULL, add this: DEFAULT "" (code: <http://bit.ly/2k7RnmX>)
  - Run the migration again.

## Lecture 32: Setting the final routes in the list

Course update for Symfony 2.8:

- Since Symfony 2.7, the routes list is executed with the `debug:router` command, which is the same I use in the video but inverted:
  - `php app/console debug:router`
- Each time you want to see the application routes, use this new form, because although both work currently, the new one will be the only one available since Symfony 3.0

## Lecture 34: Creating the repository

Course update for Symfony 2.8:

- The *Author* repository was created automatically after the entity. You will only have to implement the methods.

## Lecture 35: Adding the Sluggable behavior

Course update for Symfony 2.8:

- As we experienced with the *Post* entity, we will have to add a DEFAULT value to the *Author's slug*.
  - Add `DEFAULT ''` after *slug's* NOT NULL in the `ALTER TABLE author` statement (code: <http://bit.ly/2kiqoH5>) and in the `CREATE TABLE post` statement (look into the code, line 25).
- Run the migration again.

## Lecture 39: Adapting links in the main and post pages

Course update for Symfony 2.8:

- Remember to use the new command to list the routes: `php app/console debug:router`

## Lecture 40: Controlling the case of an author with no posts and linking to the main page

Course update for Symfony 2.8:

- In the video I have repeated the *author* identifier when I defined the translation for `empty.posts`. This causes an error in the current version of Symfony, so you have to define it as `author.empty.posts` just below *author*.

## Lecture 44: Creating a partial to show comments

Course update for Symfony 2.8:

- Follow the recommendation in the video to fix the *author* translations.
- It's not necessary to re-install *assets* after modifying the CSS file.

## Lecture 47: Generating a Doctrine FormType

Course update for Symfony 2.8:

- When you edit the *CommentType.php* file you will see that the methods are different than the ones in the video. Don't worry, this changed recently but it will work the same way.

## Lecture 49: Adding a CSS for the form

Course update for Symfony 2.8:

- In *PostControllerTest.php* you will have to change `blog_modelbundle_comment[authorName]` with `comment[authorName]`, and the same for the *body* (code: <http://bit.ly/2k7Wgfo>).

# Lecture 52: Creating the security controller

Course update for Symfony 2.8:

- The login action has been simplified in the current version of Symfony. Do these modifications (code: <http://bit.ly/2kJdG5d>):
  - Remove all content from the *loginAction* method, excepting the *return*.
  - Just above the *return*, add this line to get a security *helper*:
    - `$helper = $this->get('security.authentication_utils');`
  - Adapt the *return* with these new values:
    - `'last_username' => $helper->getLastUsername(),`
    - `'error' => $helper->getLastAuthenticationError(),`

# Lecture 54: Adding the administrator and configuring the firewall

Course update for Symfony 2.8:

- In the current Symfony version some security options in *security.yml* do not appear by default anymore. You can leave it as you have it, but adding:
  - The plain text *encoder*.
  - The *users provider* as it appears in the video to define the password.
- Check out how this file should look like: <http://bit.ly/2jIUzGH>

# Lecture 55: Generating the CRUD

Course update for Symfony 2.8:

- It's possible that you cannot find the templates the generator creates in `ModelBundle/Resources/views`. In that case they are created in `app/resources/views`.
- Remove the *model\_author* route from `app/config/routing.yml`

# Lecture 56: Cleaning the generated controller by the CRUD

Course update for Symfony 2.8:

- Before the generator used to create the *newAction* and *createAction* actions. Now it only creates *createAction*, with unified functionalities.
- Same case for *editAction*.
- It's not necessary to create the PHPDoc (comment) for the *\$entity* variable, because it's not used anymore.
- *@Template* statements in PHPDoc are not generated, don't worry about this.
- Notice that some methods are not setting the *\$id* parameter anymore, but they have an *\$author*.
- Please check out how the final code looks like: <http://bit.ly/2kKuhSo>

# Lecture 60: Cleaning the generated templates by the CRUD

Course update for Symfony 2.8:

- Instead of *entity*, the generator uses the *author* object.
- In the *edit* template change the button value from *Edit* to *Update*.

# Lecture 61: Adapting the AuthorType and the controller

Course update for Symfony 2.8:

- In *AuthorController*:
  - When prefixing *blog\_admin\_* do it for *generateUrl* and *redirectToRoute*.
  - Add `use Sensio\Bundle\FrameworkExtraBundle\Configuration\Template;`
  - For each action returning an *array* to the template, remove `$this->render` and return only the array. Then add *@Template()* in the PHPDoc. You can check out how the final file looks like: <http://bit.ly/2kifP7a>
- Before running the tests, open `AdminBundle/Tests/Controller/AuthorControllerTest.php` and replace *blog\_modelbundle\_author* with *author*.

## Lecture 62: Testing the authors administration

- If you see the *login* form without styles applied, re-install *assets*:
  - `php app/console assets:install --symlink`

## Lecture 63: Generating the CRUD

Course update for Symfony 2.8:

- Remember to move the templates from `app/Resources/views`
- Remember also to remove the *model\_post* route.

## Lecture 64: Adapting the test

Course update for Symfony 2.8:

- Do not use *blog\_modelbundle\_post* when interacting with the *form*, use *post* instead.
- When creating the *\$authorValue* variable, use the *#post\_author* id instead of *#blog\_modelbundle\_post\_author*

## Lecture 66: Cleaning the controller

Course update for Symfony 2.8:

- When prefixing *blog\_admin\_* do it for *generateUrl* and *redirectToRoute*.
- Add `use Sensio\Bundle\FrameworkExtraBundle\Configuration\Template;`
- For each action returning an *array* to the template, remove *\$this->render* and return only the *array*. Then add *@Template()* in the PHPDoc. You can check out how the final file looks like: <http://bit.ly/2kigoy2>



## Lecture 68: Cleaning templates and testing the posts administration

Course update for Symfony 2.8:

- Instead of *entity*, the generator uses the *post* object.
- In the *edit* template change the button value from *Edit* to *Update*.

## Lecture 75: Creating our first service for the author

Course update for Symfony 2.8:

- The `@doctrine.orm.entity_manager` argument must be defined using quotes (code: <http://bit.ly/2k7K2DJ>)
- The *services* file is now in YAML format by default, so it's not necessary to configure its extension in the *DependencyInjection* directory anymore (in fact this directory does not exist now).
- `container:debug` is deprecated and is removed from Symfony3. You must use the new syntax `debug:container`.

## Lecture 78: Refactoring the post controller

Course update for Symfony 2.8:

- As we did earlier, the `@doctrine.orm.entity_manager` argument must be defined using quotes.