

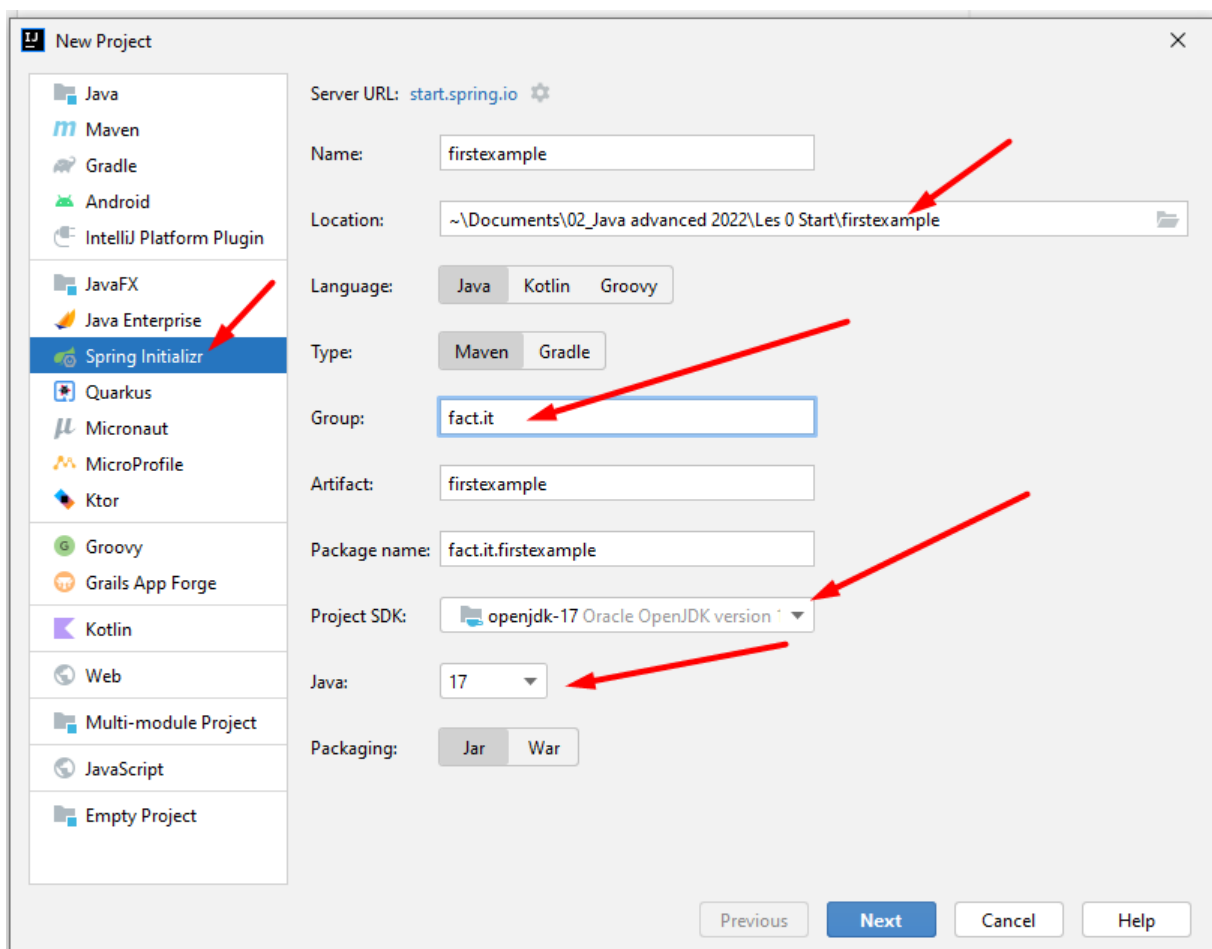
# Creating a project in IntelliJ

## Contents

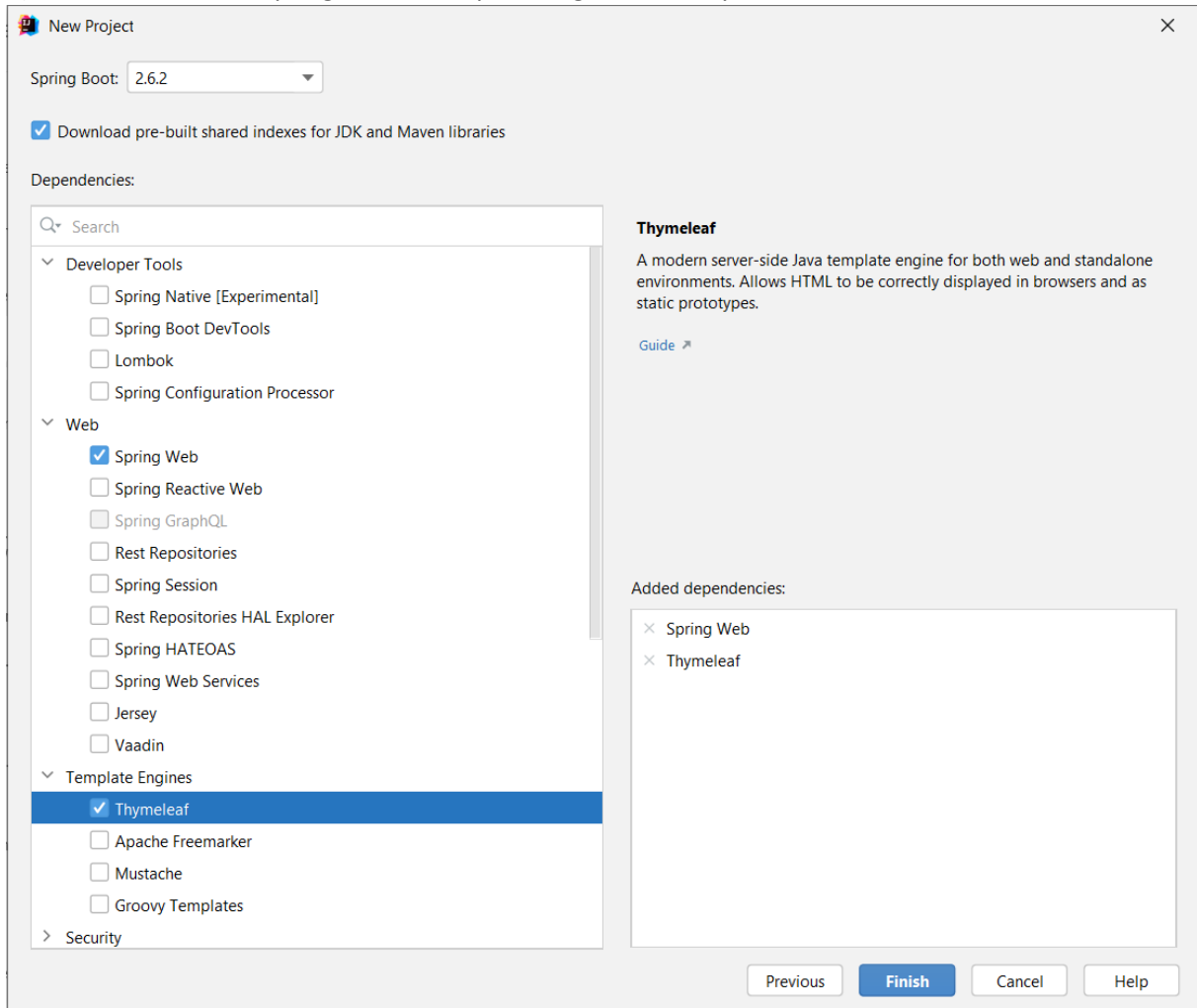
|  |   |
|--|---|
| A. Web application with Thymeleaf .....        | 1 |
| Create template for Thymeleaf .....            | 2 |
| Structure of MVC.....                          | 3 |
| B. Web application with Thymeleaf and JPA..... | 5 |

## A. Web application with Thymeleaf

- 1) File > New > Project...
- 2) Choose Spring Initializr and choose a name for your project (e.g. *firstexample*)  
Choose the location where you want your project to be saved. Note that if you choose a new Location (by browsing) you have to repeat the name of your project at the Location (at the end).  
For "Group" always fill in *fact.it*. Make sure you mark the *Project SDK 17* (and Java 17).

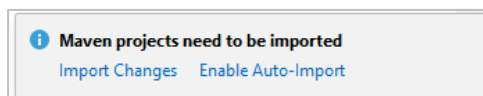


### 3) Choose Web and Spring Web + Template Engines and Thymeleaf




### 4) Click Finish

### 5) You may get this window. Then click **Enable Auto-Import**.



### Create template for Thymeleaf

You probably already have this from last academic year 1ITF Java. But if you don't have it (yet), do the following:

- 1) From the menu, select File > Settings > Editor > File and Code Templates.
- 2) Click on and  enter 'Thymeleaf HTML file', as extension 'html' and enter the following code

```
<DOCTYPE html>
<html lang="en" xmlns:th="http://www.thymeleaf.org">
<head>
  <meta http-equiv="Content-Type" content="text/html; charset=UTF-8">.
```

```

        <title>Title</title>
    </head>.
    <body>.
        Body content
    </body>
</html>.

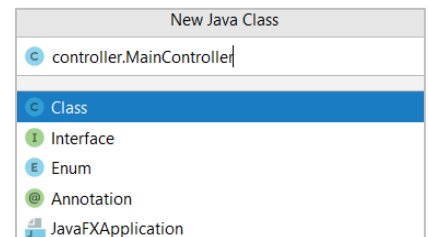
```

3) Click on ok

## Structure of MVC

### 1) Controller

- In the project overview, go to src > main > java > fact.it.yourproject
- Right-click and choose New > Java Class
- Name that class controller.MainController and press Enter



- Add the annotation @Controller at the top of MainController: **@Controller** and make the mappings

```
package fact.it.vbthymeleaf.controller;
```

```
import org.springframework.stereotype. Controller;
```

```
import org.springframework.ui.Model;
```

```
import org.springframework.web.bind.annotation. RequestMapping;
```

```
@Controller
```

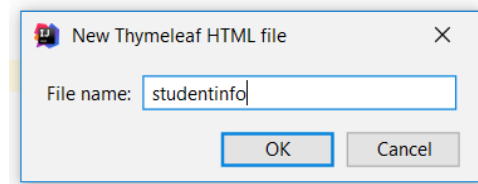
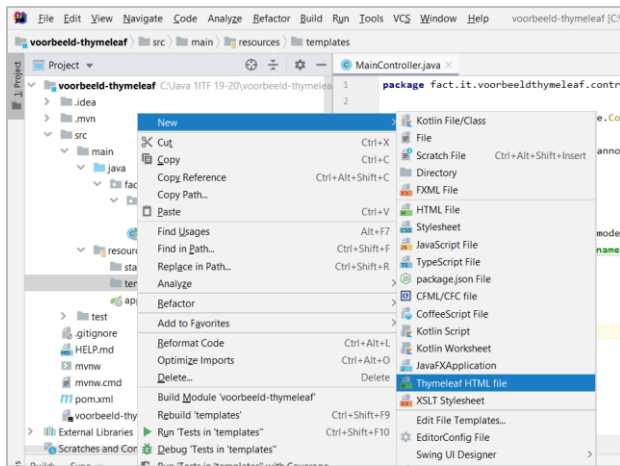
```
public class MainController { @RequestMapping("/") public String studentInfo(Model model){
model.addAttribute("firstname", "Kristine"); return "studentinfo";
}}
```

### 2) Model

- You create the classes in the package *model*, which is located under fact.it.yourproject, as before.

### 3) View layer

- In the Project Overview, go to src > main > resources > templates
- Right click and choose New > Thymeleaf HTML file> choose a name



- The Thymeleaf template appears. Complete.

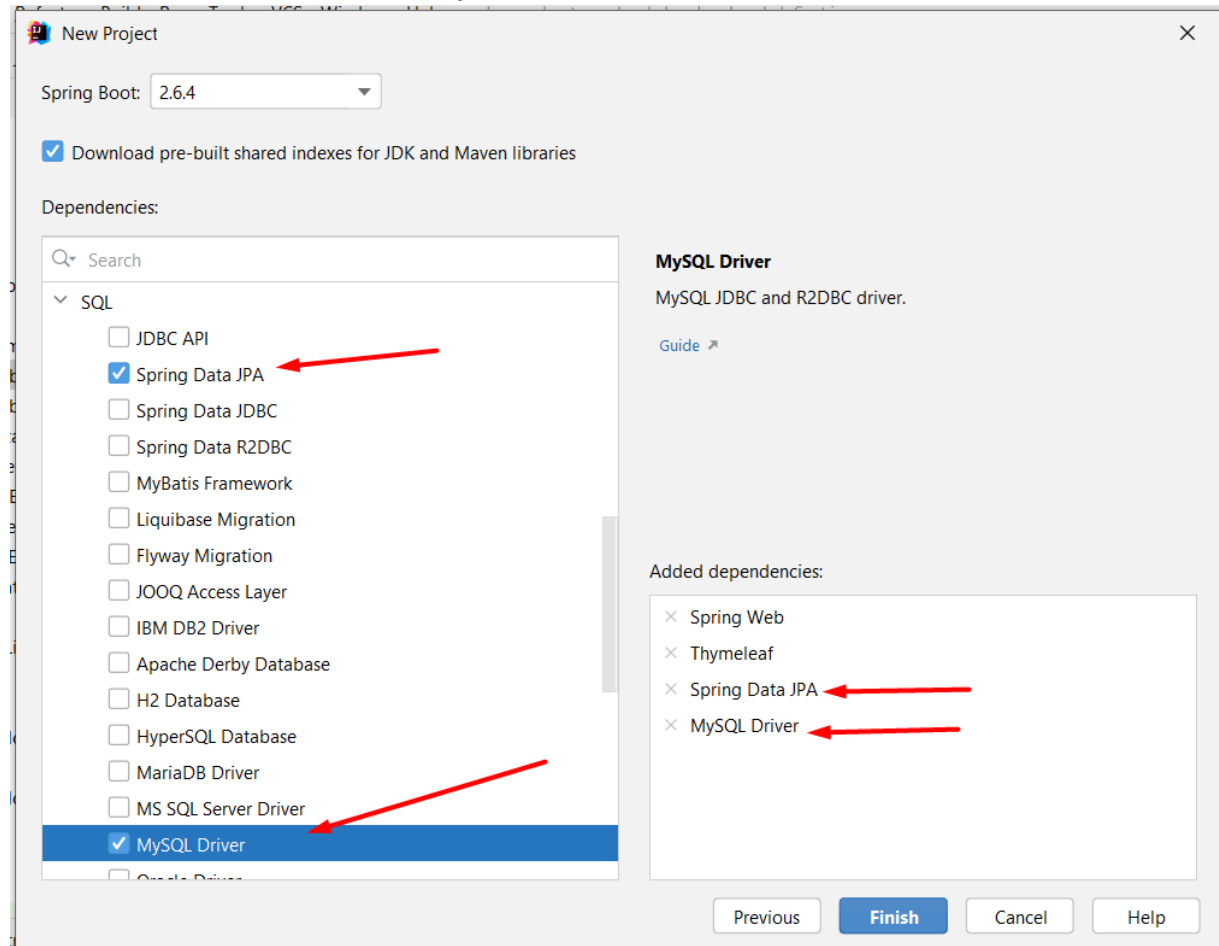


- Click on run
- Go to browser and enter "http://localhost:8080" as URL.

## B. Web application with Thymeleaf and JPA (from lesson 3!)

Follow steps 1 to 6 of A. Creating a web application with Thymeleaf (A).

7. Then click on SQL in the left column and in the middle column choose **Spring Data JPA**
8. And lower in the same list, click on **MySQL Driver**.



9. Finally, click next and finish...

10. Put the following code in your application.properties file

```
spring.datasource.url=jdbc:mysql://localhost:3306/lesson?serverTimezone=UTC
spring.datasource.username=admin
spring.datasource.password=sql
spring.jpa.hibernate.use-new-id-generator-mappings= false
```

11. Add the following code in your application.properties file depending on the situation
  - a. If you want to start from a "clean slate" and delete the existing tables with the content from the start of your application, add the following code

**`spring.jpa.hibernate.ddl-auto=create-drop`**

- b. However, if you want to continue working on the tables of content already in your database, replace the above code with:

**`spring.jpa.hibernate.ddl-auto=update`**

- c. If you are running your application with entities for the first time, it is best to leave out this code (because of the ddl car)