

# Spring MVC

## Цель работы

- Обучиться приемам работы с Spring MVC

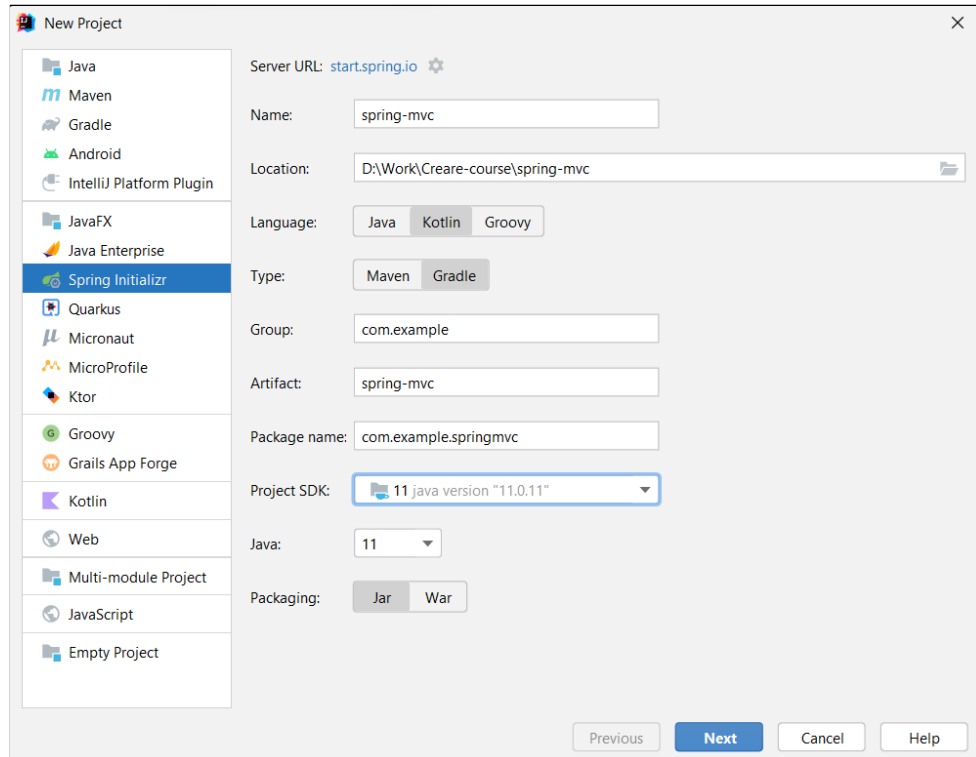
## Дополнительные материалы

## Теоретический материал

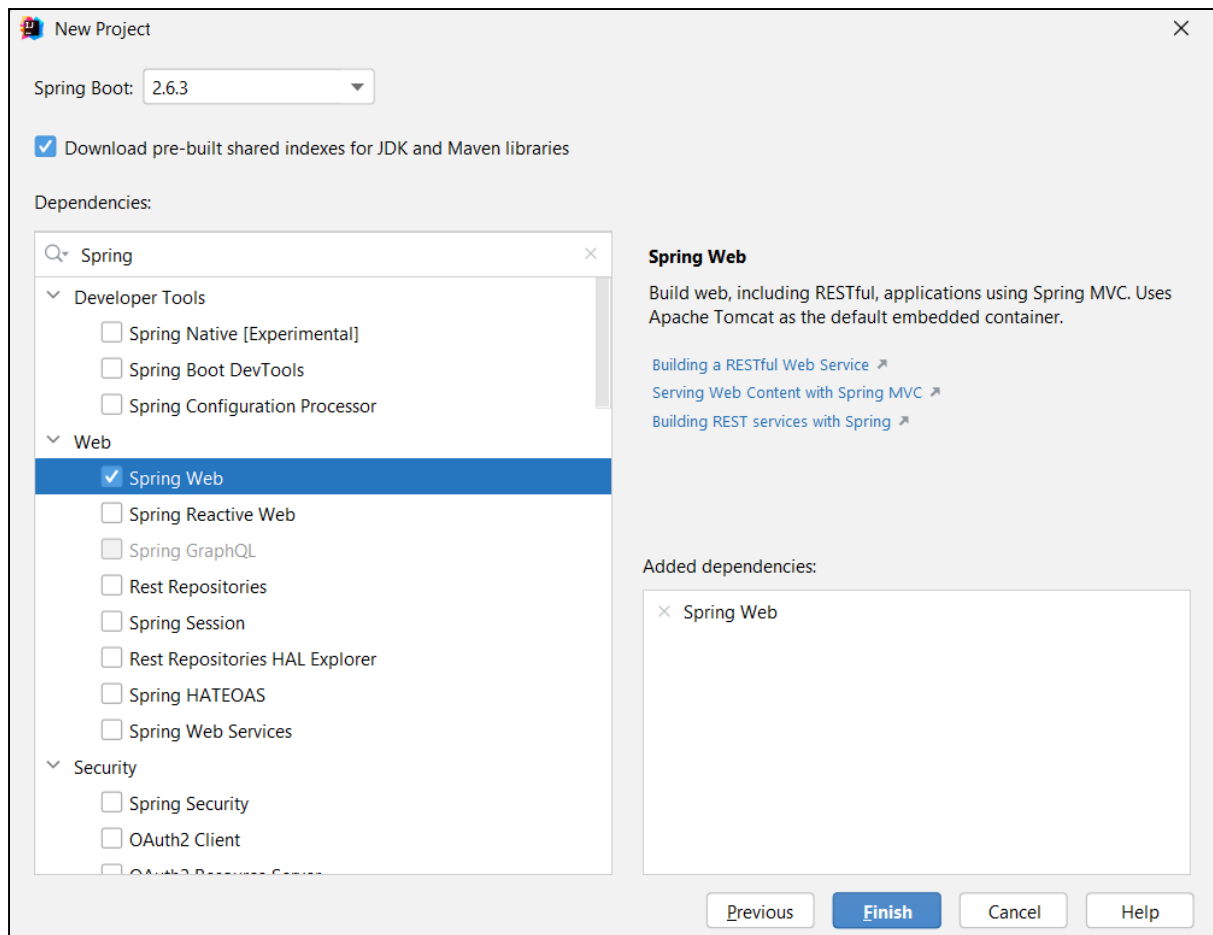
## Практическая работа

### Работа с Spring MVC

1. Открываем IDEA и проект типа Spring Initializer.



2. Затем в следующем окне выбираем пакет spring web.



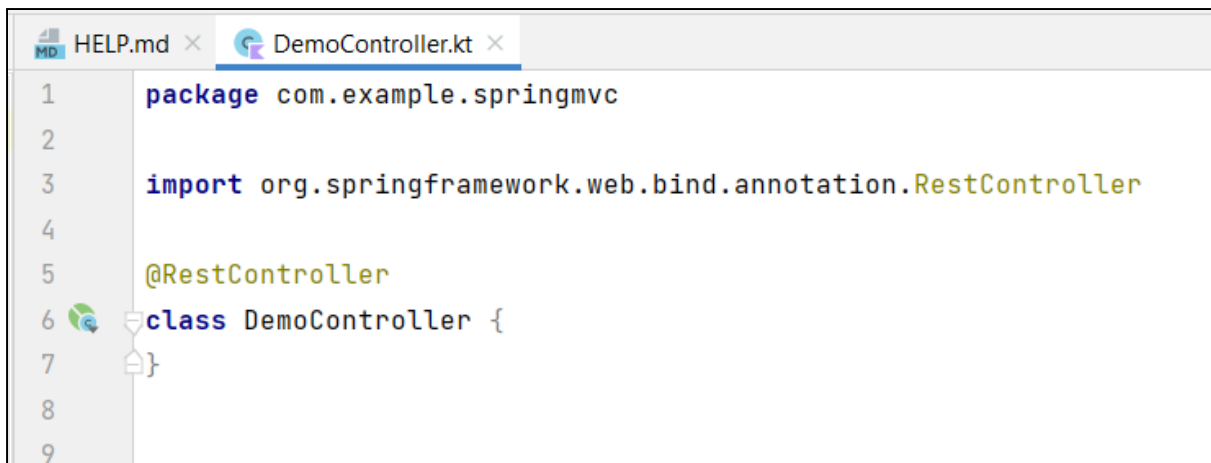
3. Если в IDEA нет типа проекта spring initializr, то все те же самые действия можно сделать на сайт spring.

<https://start.spring.io/index.html>

4. На сайте выбираем нужные значения, потом добавляем зависимость spring web.


5. Можно также создать обычный гредловый проект и уже потом добавить нужные пакеты spring.

6. Создаем класс контроллер. Затем добавляем к нему аннотацию RestController.



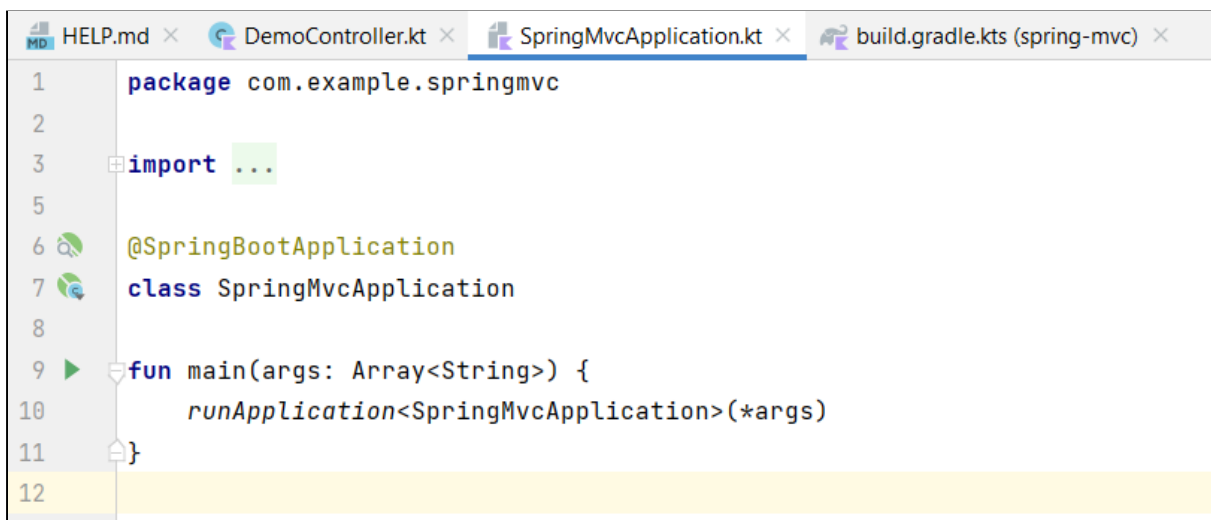
```
1 package com.example.springmvc
2
3 import org.springframework.web.bind.annotation.RestController
4
5 @RestController
6 class DemoController {
7 }
8
9
```

7. Далее пишем RequestMapping для обработки какого-то запросы и какой-то результат на этот запрос.



```
@RestController
class DemoController {
    @RequestMapping("/home", method = [RequestMethod.GET])
    fun home(): String? {
        return "HOME"
    }
}
```

8. Если при создании проекта было использовано spring initializr, то spring уже создал точку входа в программу самостоятельно, но если был использован обычный gradle проект, то нужно создать точку входа.

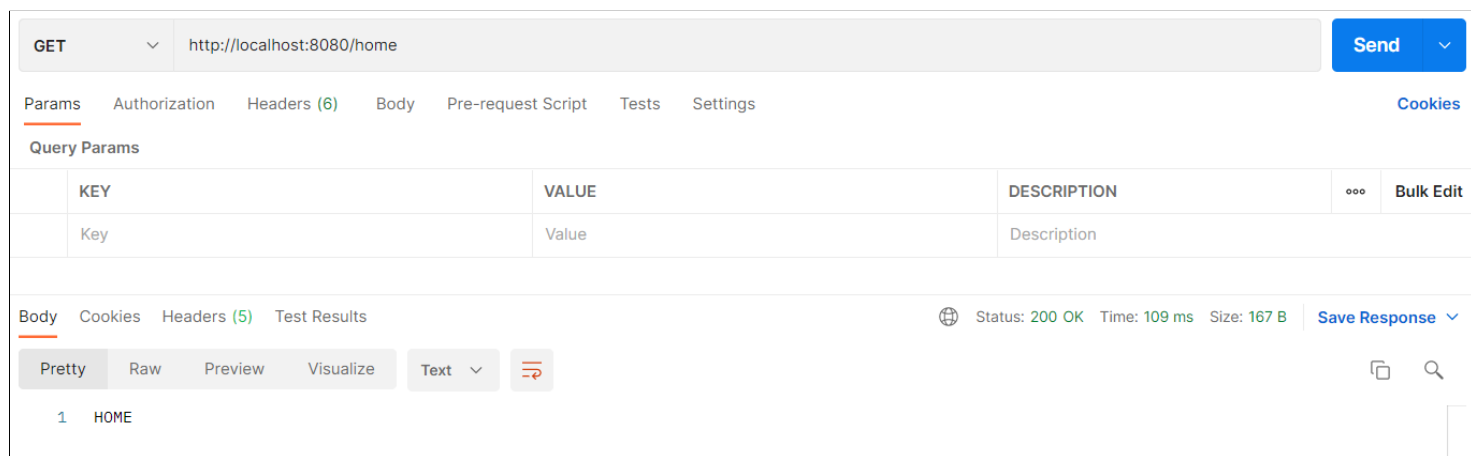


```
1 package com.example.springmvc
2
3 import ...
4
5
6 @SpringBootApplication
7 class SpringMvcApplication
8
9 fun main(args: Array<String>) {
10     runApplication<SpringMvcApplication>(*args)
11 }
12
```

9. Теперь остается запустить программу. Если все успешно, то в консоли мы увидим:

[illegible]

10. Переходим в postman и создаем наш запрос.



11. Как мы видим все прошло успешно.

## Практические задания

1. Спроектировать и реализовать REST API для редактирования TODO. В этот раз уже со всеми функциями - списками, подзадачами и аттачами.
2. Написать тесты на методы REST API с использованием @AutoConfigureMockMvc: <https://spring.io/guides/gs/testing-web/>

## Теоретические вопросы

1. Что такое MVC ?
2. Жизненный цикл запроса.
3. Что означает аннотация @Controller ?
4. Что означает аннотация @RequestMapping ?
5. Что означает аннотация @PathVariable ?
6. Что означает аннотация @ResponseBody ?
7. Что означает аннотация @RestController ?
8. Что означает аннотация @RequestBody ?
9. Что означает аннотация @RequestParam ?
10. Что означает аннотация @GetMapping ?
11. С помощью каких инструментов происходит обработка ошибок ?