Spring MVC Web framework: Multiple row

1

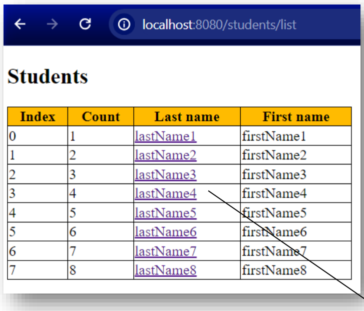
1

2

2

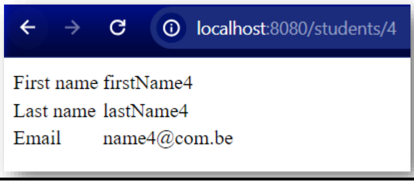
1. Multiple Row

Example


localhost:8080/students/list

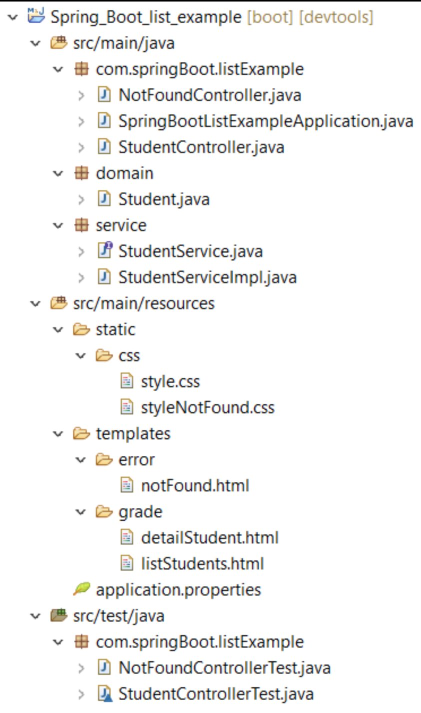
Students

Index	Count	Last name	First name
0	1	lastName1	firstName1
1	2	lastName2	firstName2
2	3	lastName3	firstName3
3	4	lastName4	firstName4
4	5	lastName5	firstName5
5	6	lastName6	firstName6
6	7	lastName7	firstName7
7	8	lastName8	firstName8


localhost:8080/students/4

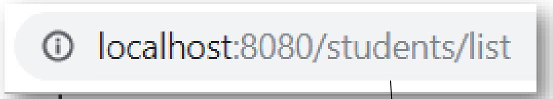
First name	firstName4
Last name	lastName4
Email	name4@com.be

3

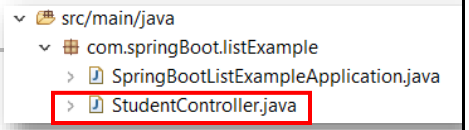


```
Spring_Boot_list_example [boot] [devtools]
  src/main/java
    com.springBoot.listExample
      NotFoundController.java
      SpringBootListExampleApplication.java
      StudentController.java
    domain
      Student.java
    service
      StudentService.java
      StudentServiceImpl.java
  src/main/resources
    static
      css
        style.css
        styleNotFound.css
    templates
      error
        notFound.html
      grade
        detailStudent.html
        listStudents.html
    application.properties
  src/test/java
    com.springBoot.listExample
      NotFoundControllerTest.java
      StudentControllerTest.java
```

4



localhost:8080/students/list



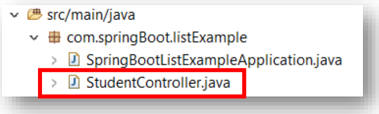
```
src/main/java
  com.springBoot.listExample
    SpringBootListExampleApplication.java
    StudentController.java
```

```
@Controller
@RequestMapping("/students")
public class StudentController {

    @Autowired
    private StudentService studentService;

    @GetMapping(value = "/list")
    public String listStudents(Model model) {
        model.addAttribute("studentList", studentService.findAll());
        return "grade/listStudents";

    }
```

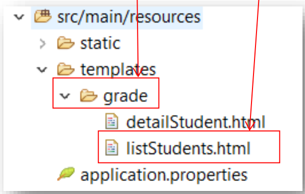
5

5

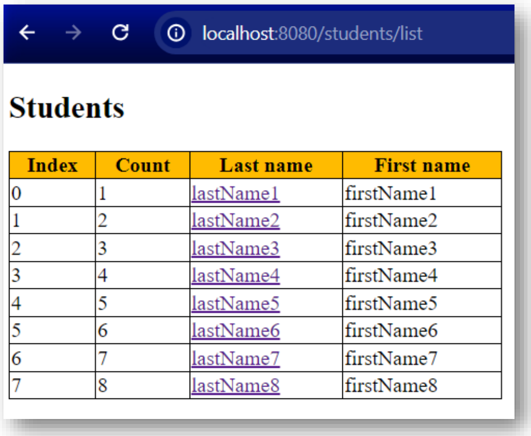
```
@GetMapping(value = "/list")
public String listStudents(Model model) {
    ...
    return "grade/listStudents";
}
```



```
src/main/java
  com.springBoot.listExample
    SpringBootListExampleApplication.java
    StudentController.java
```



```
src/main/resources
  static
  templates
    grade
      detailStudent.html
      listStudents.html
  application.properties
```



localhost:8080/students/list

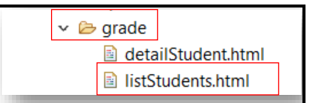
Students

Index	Count	Last name	First name
0	1	lastName1	firstName1
1	2	lastName2	firstName2
2	3	lastName3	firstName3
3	4	lastName4	firstName4
4	5	lastName5	firstName5
5	6	lastName6	firstName6
6	7	lastName7	firstName7
7	8	lastName8	firstName8

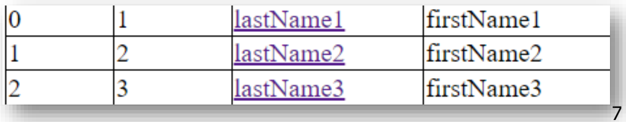
6

6

```
…
<html xmlns:th="http://www.thymeleaf.org">
…
<link rel="stylesheet" th:href="@{/css/style.css}" />
…

      <th:block th:each="student,iter: ${studentList}">
        <tr>
        <td th:text="${iter.index}"></td>
        <td th:text="${iter.count}"></td>

        <td>
            <a th:href="|/students/${student.id}|"
               th:text="${student.lastname}"></a>
        </td>

        <td th:text="${student.firstname}"></td>
        </tr>

       </th:block>
…
```



grade
detailStudent.html
listStudents.html



0	1	lastName1	firstName1
1	2	lastName2	firstName2
2	3	lastName3	firstName3

7

7

`<th:block th:each="student,iter: ${studentList}">`

th:block is a Thymeleaf-specific element that allows you to group other Thymeleaf attributes and elements together **without adding any additional markup to the HTML output**.

`<td th:text="${iter.index}"></td>`

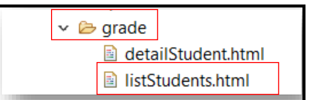
provides the index of the current iteration, starting from **0**.

`<td th:text="${iter.count}"></td>`

provides the current iteration count, starting from **1**.

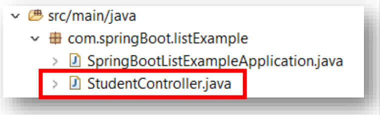
8

8



```
grade
    detailStudent.html
    listStudents.html
```

```
<a th:href="|/students/${student.id}|"
        th:text="${student.lastname}"></a>
```

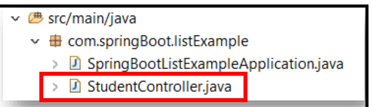


```
src/main/java
  com.springBoot.listExample
    SpringBootListExampleApplication.java
    StudentController.java
```

```
@Controller
@RequestMapping("/students")
public class StudentController {

@GetMapping(value = "/{id}")
public String show(@PathVariable Integer id, Model model) {

        Student student = studentService.findById(id);
        if (student == null) {
                return "redirect:/students/list";
        }
        model.addAttribute("student", student);
        return "grade/detailStudent";
}
```

9

9



```
src/main/java
  com.springBoot.listExample
    SpringBootListExampleApplication.java
    StudentController.java
```

```
@GetMapping(value = "/{id}")
public String show(@PathVariable Integer id, Model model) {
      Student student = studentService.findById(id);
```

OR

```
@GetMapping(value = "/{id}")
public String show(@PathVariable("id") Integer studentId,
                        Model model) {

        Student student = studentService.findById(studentId);
```

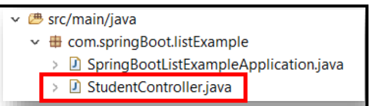
OR

int instead of **Integer**

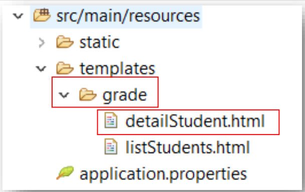
10

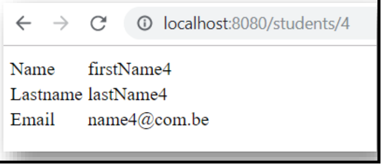
10

```
@RequestMapping(value = "/{id}",
                method = RequestMethod.GET)
public String show(@PathVariable Integer id, Model model) {
       ...
       return "grade/detailStudent";
}
```


```
src/main/java
  com.springBoot.listExample
    SpringBootListExampleApplication.java
    StudentController.java
```

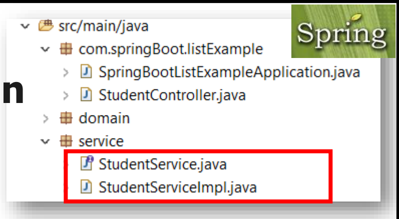
```
 ...
 <tr>
        <td>Name</td>
        <td th:text="${student.firstname}"></td>
 </tr>
 <tr>
        <td>Lastname</td>
        <td th:text="${student.lastname}"></td>
</tr>
<tr>
        <td>Email</td>
        <td th:text="${student.email}"></td>
</tr>
...
```

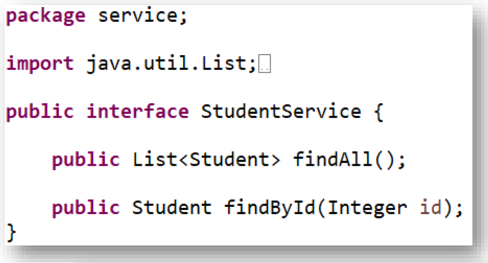

```
src/main/resources
  static
  templates
    grade
      detailStudent.html
      listStudents.html
  application.properties
```

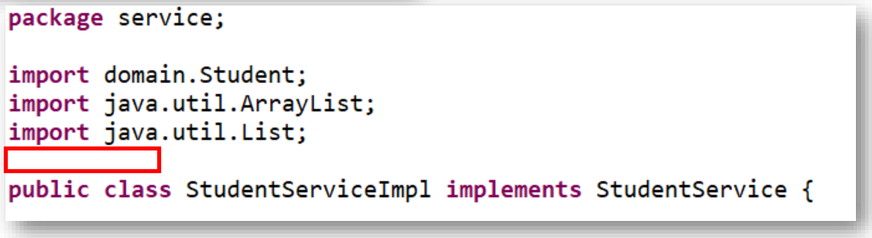

```
localhost:8080/students/4

Name     firstName4
Lastname lastName4
Email    name4@com.be
```

11

2.1 Package service SpringBootApplication: @Bean


```
src/main/java
  com.springBoot.listExample
    SpringBootListExampleApplication.java
    StudentController.java
  domain
  service
    StudentService.java
    StudentServiceImpl.java
```

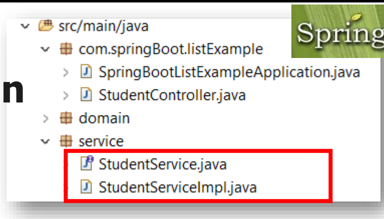

```
package service;

import java.util.List;

public interface StudentService {

    public List<Student> findAll();

    public Student findById(Integer id);
}
```


```
package service;

import domain.Student;
import java.util.ArrayList;
import java.util.List;

public class StudentServiceImpl implements StudentService {
```

12

12

2.1 Package service SpringBootApplication: @Bean



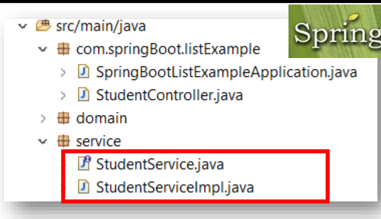
```
src/main/java
  com.springBoot.listExample
    SpringBootListExampleApplication.java
    StudentController.java
  domain
  service
    StudentService.java
    StudentServiceImpl.java
```

```
package com.springBoot.listExample;

import org.springframework.context.annotation.Bean;
import service.StudentService;
import service.StudentServiceImpl;
…

@SpringBootApplication
public class SpringBootListExampleApplication implements WebMvcConfigurer{

        …

        @Bean
        StudentService studentService() {
                return new StudentServiceImpl();
        }
}
```

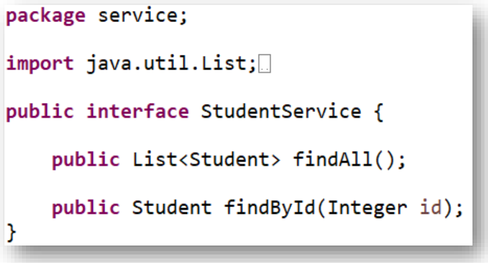
13

13

2.2 Package service SpringBootApplication: @ComponentScan



```
src/main/java
  com.springBoot.listExample
    SpringBootListExampleApplication.java
    StudentController.java
  domain
  service
    StudentService.java
    StudentServiceImpl.java
```



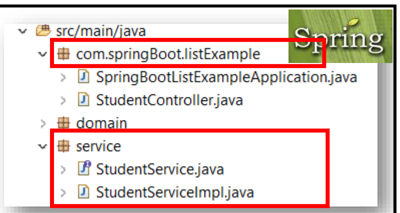
```
package service;

import java.util.List;

public interface StudentService {

    public List<Student> findAll();

    public Student findById(Integer id);
}
```

```
package service;
import org.springframework.stereotype.Service;
…

@Service
public class StudentServiceImpl implements StudentService {
…
```

14

14

2.2 Package service SpringBootApplication: @ComponentScan



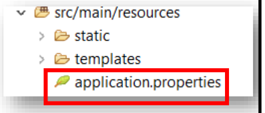
```
src/main/java
  com.springBoot.listExample
    SpringBootListExampleApplication.java
    StudentController.java
  domain
  service
    StudentService.java
    StudentServiceImpl.java
```

```
package com.springBoot.listExample;

import org.springframework.context.annotation.ComponentScan;
…
@SpringBootApplication
@ComponentScan({"service", "com.springBoot.listExample"})
public class SpringBootListExampleApplication
                                    implements WebMvcConfigurer{

 /*@ComponentScan OR define a @Bean for each service class
 @Bean
 StudentService studentService() {
        return new StudentServiceImpl();
 }*/
}
```

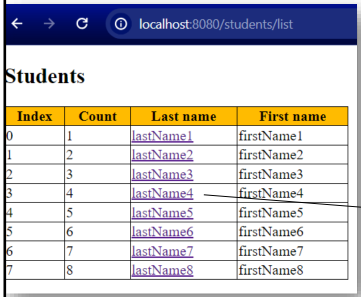
15

15

3. To remove the jsessionid from the url



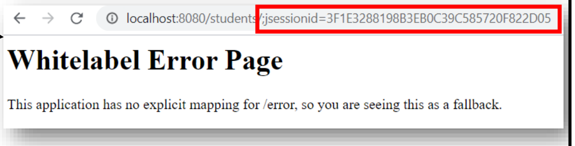
```
src/main/resources
  static
  templates
  application.properties
```



localhost:8080/students/list

Students

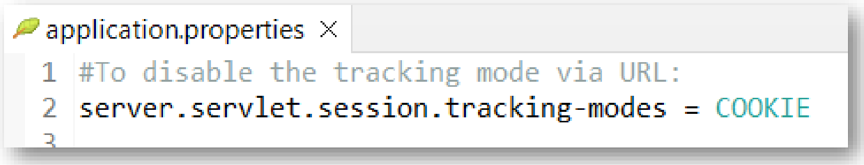
Index	Count	Last name	First name
0	1	lastName1	firstName1
1	2	lastName2	firstName2
2	3	lastName3	firstName3
3	4	lastName4	firstName4
4	5	lastName5	firstName5
5	6	lastName6	firstName6
6	7	lastName7	firstName7
7	8	lastName8	firstName8



localhost:8080/students;jsessionid=3F1E3288198B3EB0C39C585720F822D05

Whitelabel Error Page

This application has no explicit mapping for /error, so you are seeing this as a fallback.

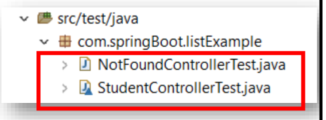


application.properties

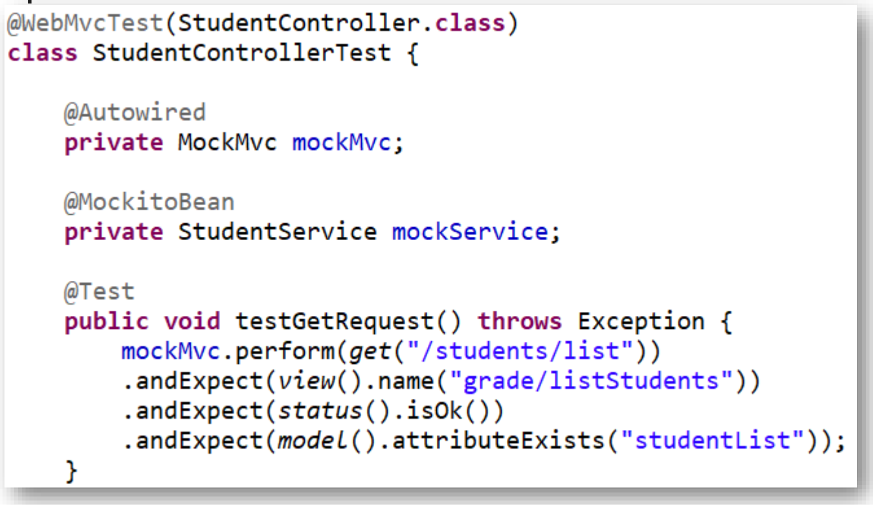
```
#To disable the tracking mode via URL:
server.servlet.session.tracking-modes = COOKIE
```

16

4. JUnit & Mockito



```
src/test/java
  com.springBoot.listExample
    NotFoundControllerTest.java
    StudentControllerTest.java
```



```java
@WebMvcTest(StudentController.class)
class StudentControllerTest {

    @Autowired
    private MockMvc mockMvc;

    @MockitoBean
    private StudentService mockService;

    @Test
    public void testGetRequest() throws Exception {
        mockMvc.perform(get("/students/list"))
        .andExpect(view().name("grade/listStudents"))
        .andExpect(status().isOk())
        .andExpect(model().attributeExists("studentList"));
    }
```

17

```
  @Test
  public void testGetDetailStudent() throws Exception {
          Student expectedStudent = new Student(1, "firstName1", "lastName1",
                                                                 "name1@com.be");
          when(mockService.findById(1)).thenReturn(expectedStudent);
          mockMvc.perform(get("/students/1"))
                      .andExpect(status().isOk())
                      .andExpect(view().name("grade/detailStudent"))
                      .andExpect(model().attributeExists("student"))
                      .andExpect(model().attribute("student", expectedStudent));
          }

  @Test
  public void testGetNoStudentFound() throws Exception {
          when(mockService.findById(1)).thenReturn(null);

          mockMvc.perform(get("/students/1"))
                      .andExpect(status().is3xxRedirection())
                      .andExpect(redirectedUrl("/students/list"));
  }
}
```

18