(/)

The Spring @Controller and @RestController Annotations

Last updated: February 1, 2024

> Written by: baeldung (https://www.baeldung.com/author/baeldung)

> Reviewed by: Michal Aibin (https://www.baeldung.com/editor/michal-author)

REST (https://www.baeldung.com/category/rest) (/)

Spring MVC (https://www.baeldung.com/category/spring/spring-web/spring-mvc)

Spring Annotations (https://www.baeldung.com/tag/spring-annotations)

Spring MVC Basics (https://www.baeldung.com/tag/spring-mvc-basics)

Get started with Spring and Spring Boot, through the reference *Learn Spring* course:

>> CHECK OUT THE COURSE (/ls-course-start)

1. Overview

In this brief tutorial, we'll discuss the difference between *@Controller* and *@RestController* annotations in Spring MVC.

We can use the first annotation for traditional Spring controllers, and it has been part of the framework for a very long time.

Spring 4.0 introduced the *@RestController* annotation in order to simplify the creation of RESTful web services. **It's a convenient annotation that combines *@Controller* and *@ResponseBody***, which eliminates the need to annotate every request handling method of the controller class with the *@ResponseBody* annotation.

Further reading:

Spring RequestMapping (/spring-requestmapping)

Spring @RequestMapping - Basic Example, @RequestParam, @PathVariable, Header mapping

Read more (/spring-requestmapping) →
(/)

Spring @RequestParam Annotation (/spring-request-param)

A detailed guide to Spring's @RequestParam annotation

Read more (/spring-request-param) →

2. Spring MVC *@Controller*

We can annotate classic controllers with the *@Controller* annotation. This is simply a specialization of the *@Component* class, which allows us to auto-detect implementation classes through the classpath scanning.

We typically use *@Controller* in combination with a *@RequestMapping* annotation for request handling methods.

Let's see a quick example of the Spring MVC controller:

```java
@Controller
@RequestMapping("books")
public class SimpleBookController {

    @GetMapping("/{id}", produces = "application/json")
    public @ResponseBody Book getBook(@PathVariable int id) {
        return findBookById(id);
    }

    private Book findBookById(int id) {
        // ...
    }
}
```

We annotated the request handling method with *@ResponseBody*. This annotation enables automatic serialization of the return object into the *HttpResponse*.

3. Spring MVC @RestController

@RestController is a specialized version of the controller. It includes the *@Controller* and *@ResponseBody* annotations, and as a result, simplifies the controller implementation:

```java
@RestController
@RequestMapping("books-rest")
public class SimpleBookRestController {

    @GetMapping("/{id}", produces = "application/json")
    public Book getBook(@PathVariable int id) {
        return findBookById(id);
    }

    private Book findBookById(int id) {
        // ...
    }
}
```

The controller is annotated with the *@RestController* annotation; therefore, the *@ResponseBody* isn't required.

Every request handling method of the controller class automatically serializes return objects into *HttpResponse*.

4. Conclusion

In this article, we examined the classic and specialized REST controllers available in the Spring Framework.

The complete source code for the examples is available in the GitHub project (https://github.com/eugenp/tutorials/tree/master/spring-web-modules/spring-mvc-basics). This is a Maven project, so it can be imported and used as is.

(/)

Get started with Spring and Spring Boot, through the *Learn Spring* course :

>> CHECK OUT THE COURSE (/ls-course-end)

Learning to build your API
with Spring?

Download the E-book (/rest-api-spring-guide)

Comments are closed on this article!

COURSES

ALL COURSES (/ALL-COURSES)

ALL BULK COURSES (/ALL-BULK-COURSES)

ALL BULK TEAM COURSES (/ALL-BULK-TEAM-COURSES) (/)

THE COURSES PLATFORM (HTTPS://COURSES.BAELDUNG.COM)

SERIES

JAVA "BACK TO BASICS" TUTORIAL (/JAVA-TUTORIAL)

JACKSON JSON TUTORIAL (/JACKSON)

APACHE HTTPCLIENT TUTORIAL (/HTTPCLIENT-GUIDE)

REST WITH SPRING TUTORIAL (/REST-WITH-SPRING-SERIES)

SPRING PERSISTENCE TUTORIAL (/PERSISTENCE-WITH-SPRING-SERIES)

SECURITY WITH SPRING (/SECURITY-SPRING)

SPRING REACTIVE TUTORIALS (/SPRING-REACTIVE-GUIDE)

ABOUT

ABOUT BAELDUNG (/ABOUT)

THE FULL ARCHIVE (/FULL_ARCHIVE)

EDITORS (/EDITORS)

JOBS (/TAG/ACTIVE-JOB/)

OUR PARTNERS (/PARTNERS)

PARTNER WITH BAELDUNG (/ADVERTISE)

TERMS OF SERVICE (/TERMS-OF-SERVICE)

PRIVACY POLICY (/PRIVACY-POLICY)

COMPANY INFO (/BAELDUNG-COMPANY-INFO)

CONTACT (/CONTACT)