

Difference between spring @Controller and @RestController annotation

[Ask Question](#)

Difference between spring @Controller and @RestController annotation.

Can @Controller annotation be used for both Web MVC and REST applications?

If yes, how can we differentiate if it is Web MVC or REST application.

[java](#) [spring](#) [spring-mvc](#)

edited Nov 21 '15 at 19:36



naXa

11.6k 7 77 123

asked Aug 11 '14 at 11:34



Srikanth

1,374 2 7 7

12 Answers

- @Controller is used to mark classes as Spring MVC Controller.
- @RestController is a convenience annotation that does nothing

This site uses cookies to deliver our services and to show you relevant ads and job listings. By using our site, you acknowledge that you have read and understand our Cookie Policy, Privacy Policy, and our Terms of Service. Your use of Stack Overflow's Products and Services, including the Stack Overflow Network, is subject to these policies and terms.

```
@Controller  
@ResponseBody  
public class MyController { }  
  
@RestController  
public class MyRestController { }
```

edited Aug 11 '14 at 15:18

answered Aug 11 '14 at 11:41



micha

30.7k 8 54 73

[stackoverflow.com/questions/35407390/...](https://stackoverflow.com/questions/35407390/) – gstackoverflow Feb 15 '16 at 10:58

- 21 I think `@RestController` also converts the response to JSON/XML automatically.
– [arnabkaycee](#) Oct 21 '16 at 13:05

- 1 Just sharing a link to a Spring tutorial explaining the difference
spring.io/guides/gs/rest-service – [Mina Samy](#) Dec 25 '16 at 14:49
- 4 Also if you use template engine like `Thymeleaf` it will not work with `@RestController` because of `@ResponseBody` which included in this annotation. – [Sonique](#) Jun 26 '17 at 7:44
- 2 `@ResponseBody` makes the returned objects to something that could be in the body, e.g. JSON or XML ([source](#)) – [Martin Thoma](#) Jun 29 '17 at 9:49

Home

PUBLIC

Stack Overflow

Tags

Users

Jobs

Teams
Q&A for work



[Learn More](#)

In the code below I'll show you the difference between `@controller`

```
@Controller  
public class restClassName{  
  
    @RequestMapping(value={"/uri"})  
    @ResponseBody  
    public ObjectResponse functionRestName(){  
        //...  
        return instance
```

```
}
```

and `@RestController`

```
@RestController
public class restClassName{

    @RequestMapping(value={"/uri"})
    public ObjectResponse functionRestName(){
        //...
        return instance
    }
}
```

the `@ResponseBody` is activated by default. You don't need to add it above the function signature.

edited Aug 23 at 8:00



Md. Abu Nafee Ibna
Zahid

401 1 4 14

answered Dec 22 '15 at 9:54



BERGUIGA Mohamed
Amine

3,354 1 24 25

`@RestController` annotated classes are the same as `@Controller` but the `@ResponseBody` on the handler methods are implied.

answered Aug 11 '14 at 11:41



Bart

13.9k 3 41 70

Excellent Article genuinec.com/spring-frameworkrestcontroller-vs-controller/ –
Karan Kaw Nov 14 '16 at 13:23

Actually, be careful - they are not exactly the same.

If you define any interceptors within your application, they will not apply to Controllers annotated as `@RestController`, however they do work with `@Controller` annotated controllers.

ie. configuration for the interceptor:

```
@Configuration
public class WebMvcConfiguration extends WebMvcConfigurerAdapter {

    @Override
    public void addInterceptors(InterceptorRegistry registry) {
        registry.addInterceptor(new
TemplateMappingInterceptor()).addPathPatterns("/**", "/admin-
functions**").excludePathPatterns("/login**");
    }

}
```

and in the declaration of a Spring controller:

```
@Controller
public class AdminServiceController {...
```

Will work, however

```
@RestController
public class AdminServiceController {...
```

does not end up having the interceptor being associated with it.

answered Feb 4 '15 at 4:42



Using Spring 4x this is working fine for me. – [Ben Dol](#) Aug 31 '15 at 3:06

² `@RestController` was introduced in Spring 4x. This annotation is also

annotated itself by `@Controller` so if it not working like an `@Controller` then report this as a bug. – [gaoagong](#) Oct 1 '15 at 17:01 

@bertybro, that's not quite right. You can associate an `Interceptor` to a `@RestController`. – [Ravi-A-Doer](#) Feb 5 '17 at 9:52

I've certainly successfully attached an `Interceptor` to a `@RestController`. – [Ben Barden](#) May 3 at 13:57 

If you use `@RestController` you cannot return a view (By using `ViewResolver` in Spring/springboot) and yes `@ResponseBody` is not needed in this case.

If you use `@controller` you can return a view in Spring webMVC.

answered Jun 27 '17 at 11:54



As you can see in Spring documentation ([Spring RestController Documentation](#)) Rest Controller annotation is the same as Controller annotation, but assuming that `@ResponseBody` is active by default, so all the json are parsed to java objects.

answered Aug 11 '14 at 11:41



THE new `@RestController` annotation in Spring4+, which marks the class as a controller where every method returns a domain object instead of a view. It's shorthand for `@Controller` and `@ResponseBody` rolled together.

answered Jan 18 '17 at 8:59



@RestController was provided since Spring 4.0.1. [These controllers](#) indicate that here @RequestMapping methods assume @ResponseBody semantics by default.

In earlier versions the similar functionality could be achieved by using below:

1. @RequestMapping coupled with @ResponseBody like

```
@RequestMapping(value = "/abc", method = RequestMethod.GET,
produces ="application/xml")
public @ResponseBody MyBean fetch(){
    return new MyBean("hi")
}
```
2. <mvc:annotation-driven/> may be used as one of the ways for using JSON with Jackson or xml.
3. MyBean can be defined like

```
@XmlRootElement(name = "MyBean")
@XmlType(propOrder = {"field2", "field1"})
public class MyBean{
    field1
    field2 ..
    //getter, setter
}
```

4. @ResponseBody is treated as the view here among MVC and it is dispatched directly instead being dispatched from Dispatcher Servlet and the respective converters convert the response in the related format like text/html, application/xml, application/json .

However, the Restcontroller is already coupled with ResponseBody and the respective converters. Secondly, here, since instead of converting the responsebody, it is automatically converted to http response.

answered Jun 16 '17 at 11:21



hi.nitish

1,067 8 16

The class is flagged as a `@RestController`, meaning it's ready for use by Spring MVC to handle web requests. `@RequestMapping` maps / to the `index()` method. When invoked from a browser or using curl on the command line, the method returns pure text. That's because `@RestController` combines `@Controller` and `@ResponseBody`, two annotations that results in web requests returning data rather than a view.

answered Mar 7 at 8:28



Vajahat Ali

34 6

-
- `@Controller` : This annotation is just a specialized version of `@Component` and it allows the controller classes to be auto-detected based on classpath scanning.
 - `@RestController` : This annotation is a specialized version of `@Controller` which adds `@Controller` and `@ResponseBody` annotation automatically so we do not have to add `@ResponseBody` to our mapping methods.

edited Aug 23 at 10:19



Jayendran

1,847 2 11 30

answered Aug 23 at 9:48



Prashant Goswami

60 5

`@RestController` is composition of `@Controller` and `@ResponseBody` , if

we are not using the `@ResponseBody` in Method signature then we need to use the `@RestController`.

edited Nov 2 '16 at 16:27



D0dger

1,182 1 9 23

answered Jan 28 '15 at 14:12



sambhu

105 1 5

Instead of using `@Controller` and `@ResponseBody`, `@RestController` let's you expose Rest APIs in Spring 4.0 and above.

answered Mar 9 '17 at 2:44



CoffeeBeanie

1

You want to say *I think* `@RestController` also converts the response to JSON/XML automatically. ? you used an abstract sentence instead of explain it clearly, I won't suggest that. – [cinqS](#) Mar 9 '17 at 3:07

Come to think of it, `@Controller` does the same as it take hte produces and consumes mime types JSON/XML or otherwise...`@ResponseBody` tells Controller to behave like REST endpoint without producing a View.
RestController implicitly does that. – [CoffeeBeanie](#) Mar 13 '17 at 6:00

protected by [Makoto](#) Jan 21 at 4:16

Thank you for your interest in this question. Because it has attracted low-quality or spam answers that had to be removed, posting an answer now requires 10 [reputation](#) on this site (the [association bonus does not count](#)).

Would you like to answer one of these [unanswered questions](#) instead?

