

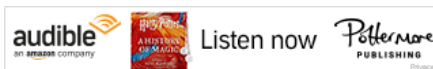
## Top 20 Spring REST Interview Questions Answers for Java Programmers to Prepare Well in 2018

Hello guys, I have been sharing some REST with Spring tutorials from last a couple of weeks and today, I am going to share some of the frequently asked Spring and REST interview questions to Java developers applying for Web developer roles. Since [Spring Framework](#) is the most popular and the standard framework for developing Java web application and RESTful Web Services, a good knowledge of Spring core and Spring MVC is expected from any senior Java developer, but if the job description mention about REST and Web Services, you also need to be aware of how to develop RESTful Web Services using Spring Framework. From Spring 3.1, the framework has been enhanced a lot to support many features needed for RESTful API out-of-the-box e.g. `HTTPMessageConverter` can convert your HTTP response to [JSON](#) or [XML](#) by just detecting relevant library in classpath e.g. Jackson and JAXB.

Spring also provides customized annotations for RESTful Web Services e.g. `@RestController` which can make your Controller REST aware, so that you don't need to do common stuff required by every single REST API e.g. converting the response to JSON.

A good knowledge of Spring Security is also mandatory for developing and security RESTful Web Services in the real world. Since you cannot make life a non-trivial REST API without security, a good knowledge of security basics e.g. [HTTP basic authentication](#), digest authentication, OAuth, and JWT is very important.

Btw, If you are not familiar with that concept, I suggest you go through [REST with Spring](#) and [Learn Spring Security](#) Course to gain some experience before your Interview. These two courses are specially designed to provide you some real-world experience to boost both your knowledge and experience with Spring MVC, REST, and Spring Security.



## 20 Spring REST Web Service Interview Questions

Here are a couple of frequently asked questions about using REST Web Services in Spring Framework.

**What does REST stand for? (answer)**

REST stands for REpresentational State Transfer, which uses HTTP protocol to send data from client to server e.g. a book in the server can be delivered to the client using JSON or XML.

### **What is a resource?** (answer)

A resource is how data is represented in REST architecture. By exposing entities as the resource it allows a client to read, write, modify, and create resources using HTTP methods e.g. [GET](#), [POST](#), [PUT](#), DELETE etc.

### **What are safe REST operations?** (answer)

REST API uses HTTP methods to perform operations. Some of the HTTP operations which doesn't modify the resource at the server is known as safe operations e.g. GET and HEAD. On the other hand, [PUT](#), POST, and DELETE are unsafe because they modify the resource on the server.

### **What are idempotent operations? Why is idempotency important?** (answer)

There are some HTTP methods e.g. GET which produce same response no matter how many times you use them e.g. sending multiple GET request to the same URI will result in same response without any side-effect hence it is known as idempotent.

On the other hand, the POST is not idempotent because if you send multiple POST request, it will result in multiple resource creation on the server, but again, PUT is idempotent if you are using it to update the resource.

Even, multiple PUT request to update a resource on a server will give same end result. You can further take [HTTP Fundamentals](#) course by Pluralsight to learn more about idempotent methods of HTTP protocol and HTTP in general.

### **Is REST scalable and/or interoperable?** (answer)

Yes, REST is Scalable and interoperable. It doesn't mandate a specific choice of technology either at client or server end. You can use Java, C++, Python or JavaScript to create RESTful Web Services and Consume them at the client end. I suggest you read a good book on REST API e.g. [RESTful Web Services](#) to learn more about REST.

### **What are the advantages of the RestTemplate?** (answer)

The `RestTemplate` class is an implementation of Template method pattern in Spring framework. Similar to other popular template classes e.g. `JdbcTemplate` or `JmsTemplate`, it also simplifies the interaction with RESTful Web Services on the client side. You can use it to consume a RESTful Web Service very easily as shown in this example.

### **Which HTTP methods does REST use?** (answer)

REST can use any HTTP methods but the most popular ones are GET for retrieving a resource, POST for creating a resource, PUT for updating resource and DELETE for removing a resource from the server.

### What is an `HttpMessageConverter` in Spring REST? (answer)

An `HttpMessageConverter` is a [Strategy interface](#) that specifies a converter that can convert from and to HTTP requests and responses. Spring REST uses this interface to convert HTTP response to various formats e.g. JSON or XML.

Each `HttpMessageConverter` implementation has one or several MIME Types associated with it. Spring uses the "Accept" header to determine the content type client is expecting.

It will then try to find a registered `HttpMessageConverter` that is capable of handling that specific content-type and use it to convert the response into that format before sending to the client.

### How to create a custom implementation of `HttpMessageConverter` to support a new type of request/responses? (answer)

You just need to create an implementation of `AbstractHttpMessageConverter` and register it using the `WebMvcConfigurerAdapter#extendMessageConverters()` method with the classes which generate a new type of request/response.

### Is REST normally stateless? (answer)

Yes, REST API should be stateless because it is based on HTTP which is also stateless. A Request in REST API should contain all the details required to process i.e. it should not rely on previous or next request or some data maintained at the server end e.g. Sessions. REST specification put a constraint to make it stateless and you should keep that in mind while designing your REST API.

### What does `@RequestMapping` annotation do? (answer)

The `@RequestMapping` annotation is used to map web requests to Spring Controller methods. You can map request based upon HTTP methods e.g. GET and POST and various other parameters. For examples, if you are developing RESTful Web Service using Spring then you can use `produces` and `consumes` property along with media type annotation to indicate that this method is only used to produce or consumers JSON as shown below:

```
@RequestMapping (method = RequestMethod.POST, consumes="application/json")
public Book save(@RequestBody Book aBook) {
    return bookRepository.save(aBook);
}
```

You can similarly create other handler methods to produce JSON or XML. If you are not familiar with these annotations then I suggest you join [Spring MVC For Beginners](#) course on Udemy to learn from scratch.

### Is `@Controller` a stereotype? Is `@RestController` a stereotype? (answer)

Yes, both `@Controller` and `@RestController` are stereotypes. The `@Controller` is

actually a specialization of Spring's `@Component` stereotype annotation. This means that class annotated with `@Controller` will also be automatically be detected by Spring container as part of container's component scanning process.

And, `@RestController` is a specialization of `@Controller` for RESTful web service. It not only combines `@ResponseBody` and `@Controller` annotation but also gives more meaning to your controller class to clearly indicate that it deals with RESTful requests.

Spring Framework may also use this annotation to provide some more useful features related to REST API development in future.

### What is the difference between `@Controller` and `@RestController`? ([answer](#))

There are many differences between `@Controller` and `@RestController` as discussed in my earlier article (see the answer) but the most important one is that with `@RestController` you get the `@ResponseBody` annotation automatically, which means you don't need to separately annotate your handler methods with `@ResponseBody` annotation. This makes the development of RESTful web service easier using Spring. You can see here to learn

```
@RestController
@EnableAutoConfiguration
public class Example {

    public static void main(String[] args) {
        SpringApplication.run(Example.class, args);
    }

    @RequestMapping("/")
    public String home() {
        return "Hello World!";
    }
}
```

### When do you need `@ResponseBody` annotation in Spring MVC? ([answer](#))

The `@ResponseBody` annotation can be put on a method to indicates that the return type should be written directly to the HTTP response body (and not placed in a Model, or interpreted as a view name).

For example:

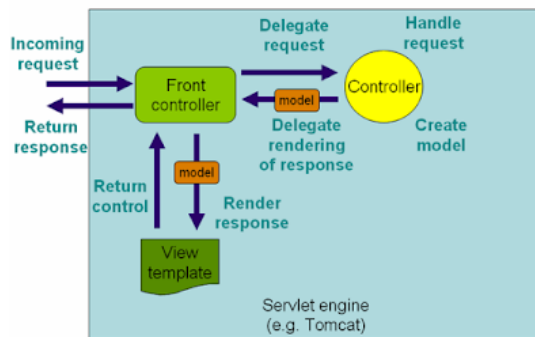
```
@RequestMapping(path = "/hello", method = RequestMethod.PUT)
@ResponseBody
public String helloWorld() {
    return "Hello World";
}
```

Alternatively, you can also use `@RestController` annotation instead of `@Controller` annotation. This will remove the need for using `@ResponseBody` because as discussed in the previous answer, it comes automatically with `@RestController` annotation.

### What does `@PathVariable` do in Spring MVC? Why it's useful in REST with Spring? ([answer](#))

It's one of the useful annotations from Spring MVC which allows you to read values from URI like query parameter. It's particularly useful in case of creating RESTful web service using Spring because in REST resource identifiers are part of URI. This question is normally asked to experienced Spring MVC developers e.g. 4 to 6 years of experience.

For example, in the URL `http://myapp.com/books/101` if you want to extract 101 the id, then you can use `@PathVariable` annotation of Spring MVC. If you are not familiar with Spring MVC annotations then [Spring MVC For Beginners: Build Java Web App in 25 Steps](#) is a good place to start with.



**What is the HTTP status return code for a successful DELETE statement? (answer)**

There is no strict rule with respect to what status code your REST API should return after a successful DELETE i.e it can return 200 Ok or 204 No Content. In general, if the DELETE operation is successful and the response body is empty return 204. If the DELETE request is successful and the response body is NOT empty, return 200

**What does CRUD mean? (answer)**

CRUD is a short form of Create, Read, Update and Delete. In REST API, the POST is used to create a resource, GET is used to read a resource, [PUT](#) is used to updated a resource and DELETE is used to remove a resource from the server.This one is another beginner level Spring MVC questions for 1 to 3 years experienced programmers

**Where do you need @EnableWebMVC? (answer)**

The `@EnableWebMvc` annotation is required to enable Spring MVC when Java configuration is used to configure Spring MVC instead of XML. It is equivalent to `<mvc: annotation-driven>` in XML configuration.

It enables support for `@Controller`-annotated classes that use `@RequestMapping` to map incoming requests to handler methods not already familiar with Spring's support for Java configuration, [Spring Master Class](#) on Udemy is a good place to start.

**When do you need @ResponseStatus annotation in Spring MVC? (answer)**

A good questions for 3 to 5 years experienced spring developers. The `@ResponseStatus` annotation is required during error handling in Spring MVC and REST. Normally when an error or exception is thrown at server side, web server return a blanket HTTP status code 500 - Internal server error.

This may work for a human user but not for REST clients. You need to send them proper status code e.g. 404 if the resource is not found. That's where you can use `@ResponseStatus` annotation, which allows you to send custom HTTP status code along with proper error message in case of Exception.

In order to use it, you can create custom exceptions and annotated them using `@ResponseStatus` annotation and proper HTTP status code and reason.

When such exceptions are thrown from controller's handler methods and not handled anywhere else, then appropriate HTTP response with the proper HTTP status code, which you have set is

sent to the client.

For example, if you are writing a RESTful Web Service for a library which provides book information then you can use `@ResponseStatus` to create Exception which returns HTTP response code 404 when a book is not found instead of Internal Server Error (500), as shown below:

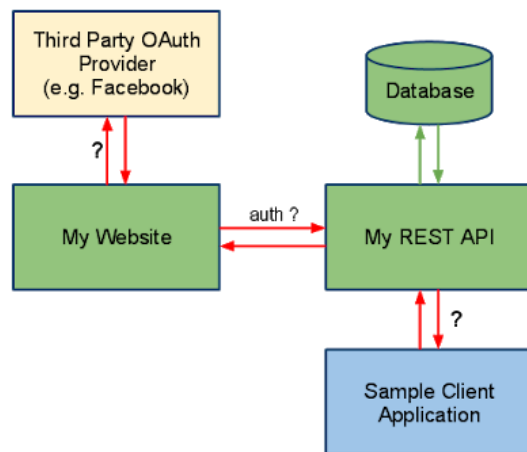
```
@ResponseStatus(value=HttpStatus.NOT_FOUND, reason="No such Book")
// 404
public class BookNotFoundException extends RuntimeException {
    // ...
}
```

If this Exception is thrown from any handler method then HTTP error code 404 with reason "No such Book" will be returned to the client.

### Is REST secure? What can you do to secure it? ([answer](#))

This question is mostly asked with experienced Java programmers e.g. 2 to 5 years experience with both REST and Spring. Security is a broad term, it could mean security of message which is provided by encryption or access restriction which is provided using authentication and authorization. REST is normally not secure but you can secure it by using Spring security.

At the very least you can enable HTTP basic authentication by using HTTP in your Spring security configuration file. Similarly, you can expose your REST API using [HTTPS](#) if the underlying server supports HTTPS.



### Does REST work with transport layer security (TLS)? ([answer](#))

TLS or Transport Layer Security is used for secure communication between client and server. It is the successor of SSL (Secure Socket Layer). Since HTTPS can work with both SSL and TLS, REST can also work with TLS.

Actually, REST says anything about Security, it's up to the server which implements that. Same RESTful Web Service can be accessed using HTTP and HTTPS if the server supports [SSL](#).

If you are using Tomcat, you can see [here](#) to learn more about how to enable SSL in Tomcat.

### Do you need Spring MVC in your classpath for developing RESTful Web Service? ([answer](#))

This question is often asked to Java programmers with 1 to 2 years of experience in Spring.


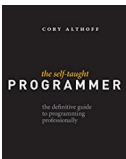

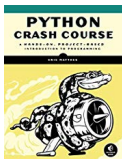



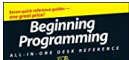
Short answer is Yes, you need Spring MVC in your Java application's classpath to develop RESTful web services using Spring framework. It's actually Spring MVC which provides all useful annotations e.g. `@RestController`, `@ResponseCode`, `@ResponseBody`, `@RequestBody`, and `@PathVariable`, hence you must spring-mvc.jar or appropriate Maven entry in your pom.xml

That's all about some **frequently asked Spring REST Interview questions** for beginners and experienced Java JEE developers. These questions are also very useful to brush up your knowledge about Spring REST if you are going to take Spring Certification. If you need more questions from Spring certification perspective, you will find a lot of question on this topic on David Mayer's [Core Spring Simulator](#), one of the best simulator to pass Spring certification at the moment.

Other **Spring tutorials and Resources** you may like

- [Spring Master Class - Beginner to Expert](#)
- [Spring Framework 5: Beginner to Guru](#)
- [5 Free Courses to learn Core Spring and Spring Boot](#)
- [How Spring MVC Frameworks works?](#)
- [How Basic Authentication works in Spring Security?](#)
- [What is the Role of DispatcherServlet in Spring?](#)
- [What is difference between @Service, @Controller, and @Component in Spring](#)
- [REST With Spring Certification Course by Baeldung](#)

Thanks for reading this article, if you like this article then please share with your friends and colleagues. If you have any question which is not answered on this list then please drop a comment and I'll try my best to find an answer for you.

			
Code: The Hidden Language of Computer Hardware an...	The Self-Taught Programmer: The Defini...	Beginning C++ Through Game Programming	Python Crash Course: A Hands-On, Project-Bas...
\$25.47 <del>\$29.99</del>	\$4.99	\$17.84 <del>\$39.95</del>	\$27.16 <del>\$39.95</del>
(409)	(181)	(224)	(339)
			

Digital Marketing Strategy Course. Lead future Digital Teams.

Amity Online | Sponsored

Changed your mind about your order? Return it for free!

PayPal | Sponsored

Choose a plane and play this Game for 1 Minute

Delta Wars | Sponsored

Hair Transplantation: Search Options Nearby

Hair Transplantation | Search Ads | Sponsored