

# How to use Spring RestTemplate

📅 May 1, 2017 (<https://mydevgeek.com/use-spring-resttemplate/>) 👤 Damith Ganegoda (<https://mydevgeek.com/author/ffft/>) ➡ Java (<https://mydevgeek.com/category/java/>), Spring (<https://mydevgeek.com/category/spring/>)



## 1) Overview

This is a tutorial about how to consume REST service using Spring RestTemplate. You can find here (<https://github.com/damithme/sprint-resttemplate/tree/master/user-service>) user service application that we are going to use as a REST service.

Using this, we can call 3rd party REST services or use for implementing microservices.

## 2) Dependencies

We need the following dependency to work with Spring RestTemplate.

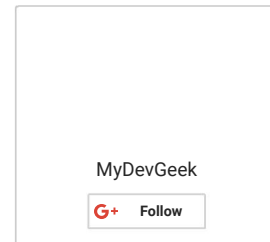
```
1 <dependency>
2   <groupId>org.springframework.boot</groupId>
3   <artifactId>spring-boot-starter-web</artifactId>
4 </dependency>
```

Other than that dependency, we will use the jackson-fasterxml dependency for convert JSON to/from Object (<http://mydevgeek.com/java-object-tofrom-json-jackson-2/>).



**MYDEVGEEK**  
([HTTPS://WWW.FACEBOOK.COM/MYDEVGEEK-391403891211607/](https://www.facebook.com/MYDEVGEEK-391403891211607/))

MyDevGeek  
(<https://www.facebook.cc/391403891211607/>)



### RECENT POSTS

Linked List – Insert a node at a specific position  
(<https://mydevgeek.com/linked-list-insert-a-node-at-a-specific-position/>)

Linked List – Insert a node at the head  
(<https://mydevgeek.com/linked-list-insert-a-node-at-the-head/>)

Angular 4 CRUD application with Spring Boot REST service – Part 3  
(<https://mydevgeek.com/angular-4-crud-application-with-spring-boot-rest-service-part-3/>)

Angular 4 CRUD application with Spring Boot REST service – Part 2  
(<https://mydevgeek.com/angular-4-crud-application-with-spring-boot-rest-service-part-2/>)

Java List of Objects to JSON – Jackson 2  
(<https://mydevgeek.com/java-list-json-jackson-2/>)

### POPULAR POSTS



Spring Boot REST web s  
Part 2 – CRUD operatio  
Service Layers, Assemb  
Utility classes  
(<https://mydevgeek.com/boot-rest-web-service-f-crud-operations-service-assembly-utility-class>)

```

1 <?xml version="1.0" encoding="UTF-8"?>
2 <project xmlns="http://maven.apache.org/POM/4.0.0" (http://maven.apa
3 xsi:schemaLocation="http://maven.apache.org/POM/4.0.0 (http
4 <modelVersion>4.0.0</modelVersion>
5
6 <groupId>com.mydevgeek</groupId>
7 <artifactId>rest-template-service</artifactId>
8 <version>0.0.1-SNAPSHOT</version>
9 <packaging>jar</packaging>
10
11 <name>rest-template-service</name>
12 <description>Demo Rest Template Service</description>
13
14 <parent>
15 <groupId>org.springframework.boot</groupId>
16 <artifactId>spring-boot-starter-parent</artifactId>
17 <version>1.5.3.RELEASE</version>
18 <relativePath/> <!-- lookup parent from repository
19 </parent>
20
21 <properties>
22 <project.build.sourceEncoding>UTF-8</project.build.
23 <project.reporting.outputEncoding>UTF-8</project.re
24 <java.version>1.8</java.version>
25 </properties>
26
27 <dependencies>
28 <dependency>
29 <groupId>org.springframework.boot</groupId>
30 <artifactId>spring-boot-starter-web</artifa
31 </dependency>
32
33 <dependency>
34 <groupId>org.springframework.boot</groupId>
35 <artifactId>spring-boot-starter-test</artif
36 <scope>test</scope>
37 </dependency>
38 <dependency>
39 <groupId>com.fasterxml.jackson.core</groupI
40 <artifactId>jackson-core</artifactId>
41 <version>2.8.7</version>
42 </dependency>
43 <dependency>
44 <groupId>com.fasterxml.jackson.core</groupI
45 <artifactId>jackson-databind</artifactId>
46 <version>2.8.7</version>
47 </dependency>
48 </dependencies>
49
50 <build>
51 <plugins>
52 <plugin>
53 <groupId>org.springframework.boot</
54 <artifactId>spring-boot-maven-plugi
55 </plugin>
56 </plugins>
57 </build>
58
59
60 </project>

```

## 3) Use GET

### 3.1) To retrieve a Java Object

Using `getForObject()`, directly convert to the Object.

```

1 String url = "http://localhost:8080/user/1" (http://localhost:8080/us
2
3 RestTemplate restTemplate = new RestTemplate();
4 User user = restTemplate.getForObject(url, User.class);
5
6 System.out.println("GET User : " + user.getFirstName() + " " + user.g

```

### 3.2) Pass parameters

Using `HashMap`, we can pass a parameters also.

```

1 String url = "http://localhost:8080/user/{id}" (http://localhost:8080/
2
3 Map<String, String> params = new HashMap<>();
4 params.put("id", "1");
5
6 RestTemplate restTemplate = new RestTemplate();
7 User user = restTemplate.getForObject(url, User.class, params);
8
9 System.out.println("GET User : " + user.getFirstName() + " " + user.g

```

### 3.2) Get response as a String and convert to the Java Object

06 Mar , 2017



Spring Boot REST web s  
Part 1 – Spring Data ani  
(https://mydevgeek.cor  
boot-rest-web-service-  
spring-data-mysql/)  
28 Feb , 2017

(https:// Angular 4 CRUD applica  
mydevge Spring Boot REST servic  
ek.com/ (https://mydevgeek.cor  
4-crud-application-with  
boot-rest-service-part-1  
18 Sep , 2017



Building a simple REST  
with Spring Boot  
(https://mydevgeek.cor  
a-simple-rest-service-w  
boot/)  
14 Feb , 2017



Deploying a Spring Boo  
Application on AWS usi  
RDS  
(https://mydevgeek.cor  
spring-boot-application  
ec2-rds/)  
12 Apr , 2017

#### CATEGORIES

Angular 4  
(https://mydevgeek.com/category/  
angular-4/) (3)

Apache Spark  
(https://mydevgeek.com/category/  
apache-spark/) (1)

AWS  
(https://mydevgeek.com/category/  
aws/) (1)

Data Structures  
(https://mydevgeek.com/category/  
data-structures/) (6)

DesignPattern  
(https://mydevgeek.com/category/  
designpattern/) (2)

Docker  
(https://mydevgeek.com/category/  
docker/) (1)

Dropwizard  
(https://mydevgeek.com/category/  
dropwizard/) (1)

IntelliJ  
(https://mydevgeek.com/category/i  
ntellij/) (4)

Java  
(https://mydevgeek.com/category/j  
ava/) (26)

```

1 String url = "http://localhost:8080/user/1 (http://localhost:8080/us
2
3 RestTemplate restTemplate = new RestTemplate();
4 HttpHeaders httpHeaders = restTemplate.headForHeaders(url);
5
6 ResponseEntity response = restTemplate.getForEntity(url, String.class);
7 System.out.println(response);

```

#### output

```

1 <200 OK,{"id":1,"firstName":"John","lastName":"Doe"},{Content-Type=[

```

convert String to Java Object.

```

1 ObjectMapper objectMapper = new ObjectMapper();
2 User user = objectMapper.readValue(response.getBody(), User.class);

```

## 4) Use POST

Normally, The POST method uses for creating a new resource. We'll create a new user using the POST method.

```

1 String url = "http://localhost:8080/user (http://localhost:8080/use
2
3 User user = new User();
4 user.setId(1L);
5 user.setFirstName("John");
6 user.setLastName("Doe");
7
8 RestTemplate restTemplate = new RestTemplate();
9 User result = restTemplate.postForObject(url, user, User.class);
10 System.out.println(result.getFirstName() + " " + result.getLastName

```

## 5) Use PUT

PUT method use for updating a exist resource.

```

1 String url = "http://localhost:8080/user/{id (http://localhost:8080
2
3 Map<String, String> params = new HashMap<>();
4 params.put("id", "1");
5
6 User user = new User();
7 user.setId(1L);
8 user.setFirstName("John");
9 user.setLastName("Smith");
10
11 RestTemplate restTemplate = new RestTemplate();
12 restTemplate.put(url, user, params);

```

## 6) Use DELETE

We can use the DELETE method for removing a resource.

```

1 String url = "http://localhost:8080/user/{id (http://localhost:8080/
2
3 Map<String, String> params = new HashMap<>();
4 params.put("id", "1");
5
6 RestTemplate restTemplate = new RestTemplate();
7 restTemplate.delete(url, params);

```

## 7) Using exchange()

We can use the `exchange()` with all HTTP methods above mentioned.

### 7.1) Using exchange send POST request

```

1 String url = "http://localhost:8080/user (http://localhost:8080/use
2 RestTemplate restTemplate = new RestTemplate();
3
4 User user = new User();
5 user.setId(1L);
6 user.setFirstName("John");
7 user.setLastName("Smith");
8
9 HttpEntity<User> request = new HttpEntity<>(user);
10 ResponseEntity<User> response = restTemplate
11 .exchange(url, HttpMethod.POST, request, User.class);
12
13 System.out.println(response.getStatusCode());
14 System.out.println(response.getBody());

```

Java 8  
(<https://mydevgeek.com/category/java-8/>) (4)

Java Core  
(<https://mydevgeek.com/category/java-core/>) (2)

Maven  
(<https://mydevgeek.com/category/maven/>) (3)

REST  
(<https://mydevgeek.com/category/rest/>) (10)

Spring  
(<https://mydevgeek.com/category/spring/>) (11)

SQL  
(<https://mydevgeek.com/category/sql/>) (1)

#### ARCHIVES

October 2017  
(<https://mydevgeek.com/2017/10/>) (2)

September 2017  
(<https://mydevgeek.com/2017/09/>) (4)

June 2017  
(<https://mydevgeek.com/2017/06/>) (1)

May 2017  
(<https://mydevgeek.com/2017/05/>) (2)

April 2017  
(<https://mydevgeek.com/2017/04/>) (3)

March 2017  
(<https://mydevgeek.com/2017/03/>) (8)

February 2017  
(<https://mydevgeek.com/2017/02/>) (12)

## 7.2) Using AsyncRestTemplate

Using `AsyncRestTemplate`, send REST calls asynchronously.

```
1 String url = "http://localhost:8080/user (http://localhost:8080/use
2 AsyncRestTemplate restTemplate = new AsyncRestTemplate();
3
4 User user = new User();
5 user.setId(1L);
6 user.setFirstName("John");
7 user.setLastName("Smith");
8
9 HttpEntity<User> request = new HttpEntity<>(user);
10 ListenableFuture<ResponseEntity<String>> response = restTemplate
11     .exchange(url, HttpMethod.POST, request, String.class);
12 response.addCallback(new ListenableFutureCallback<ResponseEntity<St
13     @Override
14     public void onFailure(Throwable throwable) {
15         //do something
16     }
17
18     @Override
19     public void onSuccess(ResponseEntity<String> response) {
20         System.out.println(response.getBody());
21     }
22 });
```

Complete examples find here (<https://github.com/damithme/sprint-resttemplate/tree/master/rest-template-service>).



(<http://www.facebook.com/sharer.php?u=https://mydevgeek.com/use-spring-resttemplate/&t=How+to+use+Spring+RestTemplate>)



(<http://twitter.com/share?text=How+to+use+Spring+RestTemplate-&url=https://mydevgeek.com/use-spring-resttemplate/&via=MyDevGee>)



([https://plusone.google.com/\\_/+1/confirm?hl=fr-FR&url=https://mydevgeek.com/use-spring-resttemplate/](https://plusone.google.com/_/+1/confirm?hl=fr-FR&url=https://mydevgeek.com/use-spring-resttemplate/))



(<http://www.tumblr.com/share/photo?source=https%3A%2F%2Fi2.wp.com%2Fmydevgeek.com%2Fwp-content%2Fuploads%2F2017%2F03%2Fspring.png%3Ffit%3D200%252C200%26ssl%3D1&caption=How+to+use+Spring+RestTemplate&clickthru-spring-resttemplate%2F>)

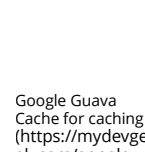
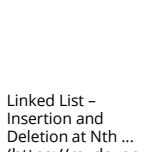
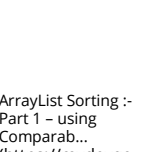
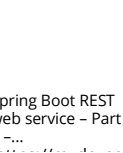


(<http://www.linkedin.com/shareArticle?mini=true&url=https://mydevgeek.com/use-spring-resttemplate/&title=How+to+use+Spring+RestTemplate>)



(<https://www.blogger.com/blog-this.g?u=https://mydevgeek.com/use-spring-resttemplate/&n=How+to+use+Spring+RestTemplate&t=MyDevGee>)

### Related Post

|   |   |   |  |
|---|---|---|--|
| <br>Google Guava<br>Cache for caching<br>( <a href="https://mydevgeek.com/tag/guava/">https://mydevgeek.com/tag/guava/</a> ) | <br>Linked List –<br>Insertion and<br>Deletion at Nth ...<br>( <a href="https://mydevgeek.com/tag/linked-list/">https://mydevgeek.com/tag/linked-list/</a> ) | <br>ArrayList Sorting :-<br>Part 1 – using<br>Comparab...<br>( <a href="https://mydevgeek.com/tag/arraylist/">https://mydevgeek.com/tag/arraylist/</a> ) | <br>Spring Boot REST<br>web service – Part<br>1 –...<br>( <a href="https://mydevgeek.com/tag/spring-boot/">https://mydevgeek.com/tag/spring-boot/</a> ) |
|---|---|---|--|



30979 total visits, 30 visits today

[Java \(https://mydevgeek.com/tag/java/\)](https://mydevgeek.com/tag/java/) [rest \(https://mydevgeek.com/tag/rest/\)](https://mydevgeek.com/tag/rest/)

[spring \(https://mydevgeek.com/tag/spring/\)](https://mydevgeek.com/tag/spring/)

0 Comments mydevgeek.com

Login ▾

Recommend Share

Sort by Best ▾



Start the discussion...

LOG IN WITH

OR SIGN UP WITH DISQUS ?

Name

Be the first to comment.

Subscribe Add Disqus to your siteAdd DisqusAdd Privacy

◀ CSV file reading and writing in Java using JCSV (<https://mydevgeek.com/csv-file-reading-writing-java-using-google-jcsv/>)

Strategy Design Pattern – Java 8 ▶ (<https://mydevgeek.com/strategy-design-pattern-java-8/>)

#### PAGES

[Contact \(https://mydevgeek.com/contact/\)](https://mydevgeek.com/contact/)

#### SUBSCRIBE TO BLOG VIA EMAIL

Enter your email address to subscribe to MyDevGeek blog and receive notifications of new posts by email.

Join 58 other subscribers

Email Address

SUBSCRIBE

MyDevGeek.com (<http://mydevgeek.com>) All rights reserved.