

[JAVA TUTORIAL](#)[#INDEX POSTS](#)[#INTERVIEW QUESTIONS](#)[RESOURCES](#)[HIRE ME](#)[DOWNLOAD ANDROID APP](#)[CONTRIBUTE](#)

Subscribe to Download Java Design Patterns eBook

DOWNLOAD NOW

[HOME](#) » [SPRING](#) » SPRING RESTFUL WEB SERVICES EXAMPLE WITH JSON, JACKSON AND CLIENT PROGRAM

Spring Restful Web Services Example with JSON, Jackson and Client Program

APRIL 2, 2018 BY [PANKAJ](#) — [164 COMMENTS](#)

Spring is one of the most widely used Java EE framework. We have earlier seen how to use [Spring MVC](#) to create Java based web applications. Today we will learn to create **Spring Restful Web Services** using Spring MVC and then test it out with the Rest client. At the end, we will also look into how to invoke Spring Restful web service using **Spring RestTemplate API**.



Table of Contents [\[hide\]](#)

[1 Spring REST](#)

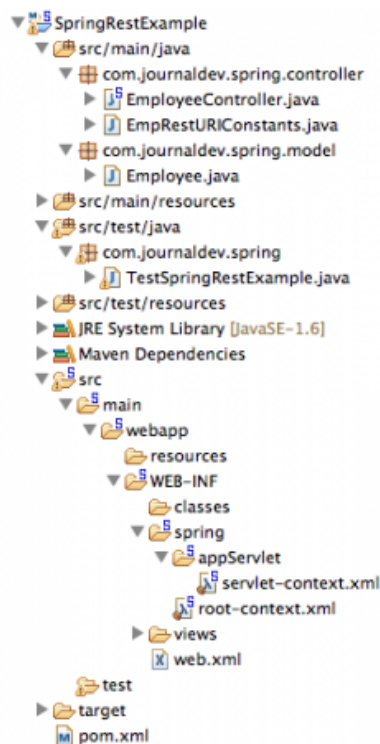
[1.1 Spring REST Configuration XML Files](#)

- 1.2 Spring REST Model Classes
- 1.3 Spring Restful web service End Points
- 1.4 Spring Restful web service Controller class
- 1.5 Spring Rest Client Program

Spring REST

We will use Spring latest version **4.0.0.RELEASE** and utilize Spring **Jackson JSON** integration to send JSON response in the rest call response. The tutorial is developed in Spring STS IDE for creating Spring MVC skeleton code easily and then extended to implement Restful architecture.

Create a new Spring MVC Project in the STS, our final project will look like below image. We will look into each of the components one by one.



Spring REST Configuration XML Files

Our pom.xml file looks like below.

```
<?xml version="1.0" encoding="UTF-8"?>
<project xmlns="http://maven.apache.org/POM/4.0.0"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
    xsi:schemaLocation="http://maven.apache.org/POM/4.0.0
  http://maven.apache.org/maven-v4_0_0.xsd">
  <modelVersion>4.0.0</modelVersion>
  <groupId>com.journaldev</groupId>
  <artifactId>SpringRestExample</artifactId>
  <name>SpringRestExample</name>
```

```

<packaging>war</packaging>
<version>1.0.0-BUILD-SNAPSHOT</version>
<properties>
    <java-version>1.6</java-version>
    <org.springframework-version>4.0.0.RELEASE</org.springframework-
version>
    <org.aspectj-version>1.7.4</org.aspectj-version>
    <org.slf4j-version>1.7.5</org.slf4j-version>
    <jackson.databind-version>2.2.3</jackson.databind-version>
</properties>
<dependencies>
    <!-- Jackson -->
    <dependency>

```

STS tool generates the pom.xml file for us, however I have updated the [Spring Framework](#), AspectJ, SLF4J and Jackson version to the latest one as of today. Most of the part is common and generated automatically, the important point to note is that I have added Jackson JSON libraries in the dependency because we will use that to convert Objects to JSON and vice versa.

```

<?xml version="1.0" encoding="UTF-8"?>
<web-app version="2.5" xmlns="http://java.sun.com/xml/ns/javaee"
    xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
    xsi:schemaLocation="http://java.sun.com/xml/ns/javaee
http://java.sun.com/xml/ns/javaee/web-app_2_5.xsd">

    <!-- The definition of the Root Spring Container shared by all Servlets and
Filters -->
    <context-param>
        <param-name>contextConfigLocation</param-name>
        <param-value>/WEB-INF/spring/root-context.xml</param-value>
    </context-param>

    <!-- Creates the Spring Container shared by all Servlets and Filters -->
    <listener>
        <listener-
class>org.springframework.web.context.ContextLoaderListener</listener-class>
    </listener>

    <!-- Processes application requests -->
    <servlet>
        <servlet-name>appServlet</servlet-name>

```

This file is generated automatically and I haven't changed anything in that. However if you want to change context configuration files and their location, you can do it in the web.xml file.

```
<?xml version="1.0" encoding="UTF-8"?>
<beans xmlns="http://www.springframework.org/schema/beans"
       xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
       xsi:schemaLocation="http://www.springframework.org/schema/beans
http://www.springframework.org/schema/beans/spring-beans.xsd">

    <!-- Root Context: defines shared resources visible to all other web
components -->

</beans>
```

This file contains the shared resources that will be visible to all the web components, we will be developing a simple rest service and that's why I haven't changed anything here.

```
<?xml version="1.0" encoding="UTF-8"?>
<beans:beans xmlns="http://www.springframework.org/schema/mvc"
            xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
            xmlns:beans="http://www.springframework.org/schema/beans"
            xmlns:context="http://www.springframework.org/schema/context"
            xsi:schemaLocation="http://www.springframework.org/schema/mvc
http://www.springframework.org/schema/mvc/spring-mvc.xsd
http://www.springframework.org/schema/beans
http://www.springframework.org/schema/beans/spring-beans.xsd
http://www.springframework.org/schema/context
http://www.springframework.org/schema/context/spring-context.xsd">

    <!-- DispatcherServlet Context: defines this servlet's request-processing
infrastructure -->

    <!-- Enables the Spring MVC @Controller programming model -->
    <annotation-driven />

    <!-- Handles HTTP GET requests for /resources/** by efficiently serving up
static resources in the ${webappRoot}/resources directory -->
    <resources mapping="/resources/**" location="/resources/" />
```

Most of the part is auto generated and contains boiler-plate configurations. However important points to note are **annotation-driven** element to support annotations based configuration and plugging in `MappingJackson2HttpMessageConverter` to the `RequestMappingHandlerAdapter` `messageConverters` so that Jackson API kicks in and converts JSON to Java Beans and vice versa. By having this configuration, we will be using JSON in request body and we will receive JSON data in the response.

Spring REST Model Classes

Let's write a simple POJO class that will serve as input and output to our Restful web service methods.

```
package com.journaldev.spring.model;

import java.io.Serializable;
import java.util.Date;

import com.fasterxml.jackson.databind.annotation.JsonSerialize;
import com.fasterxml.jackson.databind.ser.std.DateSerializer;

public class Employee implements Serializable{

    private static final long serialVersionUID = -7788619177798333712L;

    private int id;
    private String name;
    private Date createdAt;

    public int getId() {
        return id;
    }
    public void setId(int id) {
        this.id = id;
    }
}
```

The only important point to note is the use of `@JsonSerialize` annotation to use `DateSerializer` class for Date conversion from Java type to JSON format and vice versa.

Spring Restful web service End Points

We will have following rest web services end points.

SL. NO	URI	HTTP METHOD	DETAILS
1	/rest/emp/dummy	GET	Health Check service, to insert a dummy data in the Employees data storage
2	/rest/emp/{id}	GET	To get the Employee object based on the id
3	/rest/emps	GET	To get the list of all the Employees in the data store

4	/rest/emp/create	POST	To create the Employee object and store it
5	/rest/emp/delete/{id}	PUT	To delete the Employee object from the data storage based on the id

We have a class defining all these URI as String constants.

```
package com.journaldev.spring.controller;

public class EmpRestURIConstants {

    public static final String DUMMY_EMP = "/rest/emp/dummy";
    public static final String GET_EMP = "/rest/emp/{id}";
    public static final String GET_ALL_EMP = "/rest/emp";
    public static final String CREATE_EMP = "/rest/emp/create";
    public static final String DELETE_EMP = "/rest/emp/delete/{id}";
}
```

Spring Restful web service Controller class

Our EmployeeController class will publish all the web service end points mentioned above, let's look at the code of the class and then we will learn about each of the methods in detail.

```
package com.journaldev.spring.controller;

import java.util.ArrayList;
import java.util.Date;
import java.util.HashMap;
import java.util.List;
import java.util.Map;
import java.util.Set;

import org.slf4j.Logger;
import org.slf4j.LoggerFactory;
import org.springframework.stereotype.Controller;
import org.springframework.web.bind.annotation.PathVariable;
import org.springframework.web.bind.annotation.RequestBody;
import org.springframework.web.bind.annotation.RequestMapping;
import org.springframework.web.bind.annotation.RequestMethod;
import org.springframework.web.bind.annotation.ResponseBody;

import com.journaldev.spring.model.Employee;
```

```
/**  
 * Handles requests for the Employee service
```

For simplicity, I am storing all the employees data in the HashMap empData. @RequestMapping annotation is used to map the request URI to the handler method. We can also specify the HTTP method that should be used by client application to invoke the rest method.

@ResponseBody annotation is used to map the response object in the response body. Once the response object is returned by the handler method, MappingJackson2HttpMessageConverter kicks in and convert it to JSON response.

@PathVariable annotation is the easy way to extract the data from the rest URI and map it to the method argument.

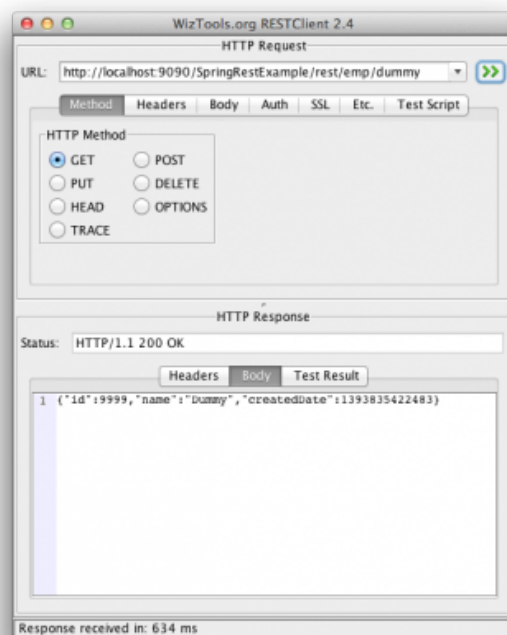
@RequestBody annotation is used to map the request body JSON data into the Employee object, again this is done by the MappingJackson2HttpMessageConverter mapping.

Rest of the code is simple and self understood, our application is ready for deployment and testing. Just export as WAR file and copy it in the servlet container web app directory. If you have server configured in the STS, you can simply run it on the server to get it deployed.

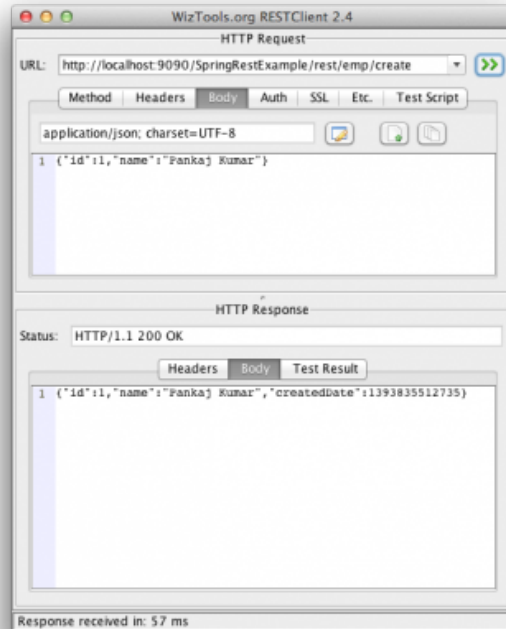
I am using WizTools RestClient to invoke the rest calls but you can also use Chrome extension Postman.

Below screenshots shows the different invocations of the rest apis exposed by our application and it's output.

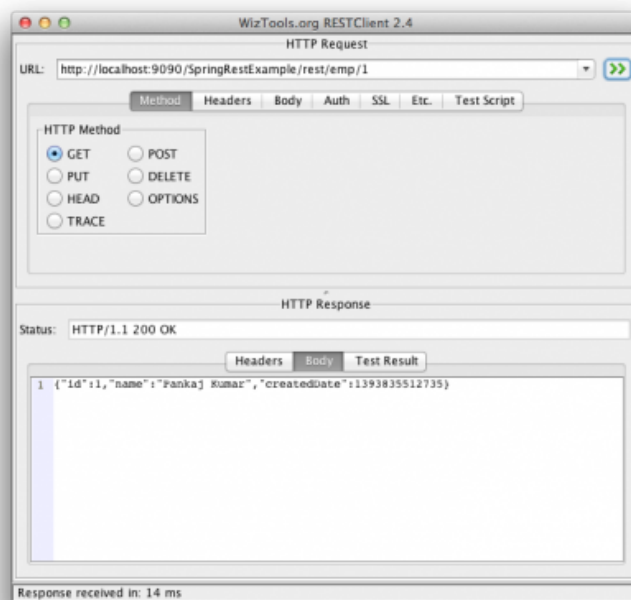
Health Check – Get Dummy Employee Rest Call



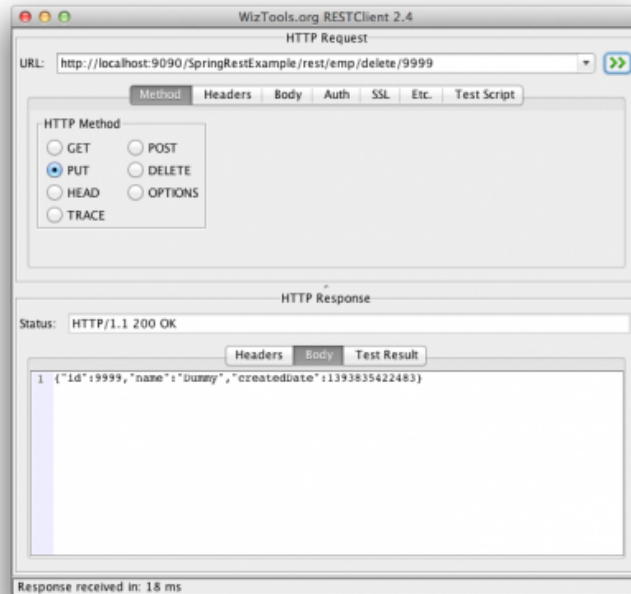
Create Employee POST Rest Call: Make sure request Content-Type is set to "application/json" otherwise you will get HTTP Error Code 415.



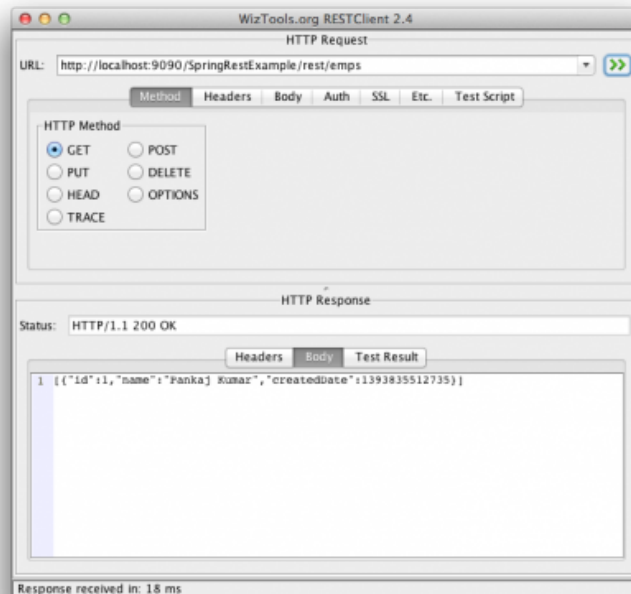
Get Employee Rest Call



Delete Employee Rest Call



Get All Employees Rest Call



Spring Rest Client Program

Rest Clients are good to test our rest web service but most of the times, we need to invoke rest services through our program. We can use Spring RestTemplate to invoke these methods easily. Below is a simple program invoking our application rest methods using RestTemplate API.

```
package com.journaldev.spring;
```

```
import java.util.LinkedHashMap;
```

```
import java.util.List;

import org.springframework.web.client.RestTemplate;

import com.journaldev.spring.controller.EmpRestURIConstants;
import com.journaldev.spring.model.Employee;

public class TestSpringRestExample {

    public static final String SERVER_URI =
"http://localhost:9090/SpringRestExample";

    public static void main(String args[]){

        testGetDummyEmployee();
        System.out.println("*****");
        testCreateEmployee();
        System.out.println("*****");
```

Most of the program is simple to understand, however when invoking rest method returning a Collection, we need to use `LinkedHashMap` because JSON to object conversion doesn't know about the `Employee` object and converts it to the collection of `LinkedHashMap`. We can write a utility method to convert from `LinkedHashMap` to our Java Bean object.

When we run above program, we get following output in the console.

```
ID=9999,Name=Dummy,CreateDate=Tue Mar 04 21:02:41 PST 2014
*****
ID=1,Name=Pankaj Kumar,CreateDate=Tue Mar 04 21:02:41 PST 2014
*****
ID=1,Name=Pankaj Kumar,CreateDate=Tue Mar 04 21:02:41 PST 2014
*****
2
ID=1,Name=Pankaj Kumar,CreateDate=1393995761654
ID=9999,Name=Dummy,CreateDate=1393995761381
```

Another point is that `RestTemplate` put methods doesn't have option to set response object because PUT method should be used to store something on the server and a simple HTTP 200 status code should be sufficient.

[Download Spring Restful Webservice Project](#)

That's all for the Spring Restful web application tutorial. Download the sample project from above link and play around with it to learn more.

UPDATE: Because of so many requests to provide similar example with XML as well as supporting both XML and JSON, I have extended this application in [Spring REST XML JSON Example](#) to support both xml and JSON requests and response. I strongly suggest you to go through that to see the beauty of spring framework and how easy it is to achieve this.

« PREVIOUS

[Spring IoC, Spring Bean Example Tutorial](#)

NEXT »

[Spring MVC File Upload Example Tutorial – Single and Multiple Files](#)

About Pankaj

If you have come this far, it means that you liked what you are reading. Why not reach little more and connect with me directly on [Google Plus](#), [Facebook](#) or [Twitter](#). I would love to hear your thoughts and opinions on my articles directly.

Recently I started creating video tutorials too, so do check out my videos on [Youtube](#).

FILED UNDER: [SPRING](#)

Comments

Kadapa Kishore Kumar Reddy says

MAY 30, 2018 AT 5:04 AM

Hi Pankaj,

actually i have doubt , if i use @RestController then i can write my response directly to HTTP Response object. But, i want to serve that response to JSP. how to serve this response to JSP? please , help me on this, i searched and read so many blogs and websites still i didnt get any clarification.

#STACKOVERFLOW Click on this link <https://stackoverflow.com/questions/30987062/how-to-use-spring-4-rest-controller-to-serve-for-jsp-view>

same question asked by some one but here also i didnt get any clarification. if you know then give a replay as soon as possible b'cz i need it . thanks

[Reply](#)

Pankaj says

MAY 30, 2018 AT 8:22 AM

1. You can reuse the business logic of your code by providing a Spring MVC wrapper that will serve JSP page and call the already defined methods to generate the response data. This should work easily if your Controller class and business logic are in separate methods/class.
2. Another option is to define JSP pages and use Ajax to call the REST web service and parse JSON response to generate HTML page.

[Reply](#)

prasad says

MAY 24, 2018 AT 10:24 AM

I am a beginner to spring, Not understanding 1 %. waste of my time.

[Reply](#)

Pankaj says

MAY 24, 2018 AT 10:42 AM

If you have no idea about Spring framework, please go through the basic tutorials first on Spring Dependency Injection, Spring Beans, IoC Container, Spring MVC and then try to build Spring REST web service. You should also have basic understanding of Restful web services.

[Reply](#)

Raj Baral says

MAY 18, 2018 AT 3:34 AM

Hi, Pankaj.

I'm trying to deploying this WAR to a JBoss application server.

The WAR is deployed but <http://localhost:8080/SpringRestExample-1.0.0-BUILD-SNAPSHOT/> returns "The requested resource is not available." page. The same is returned for other API end points too.

In the server log I can see "No Spring WebApplicationInitializer types detected on classpath".

[Reply](#)**J Manas Kumar says**

JANUARY 11, 2018 AT 7:04 AM

How can I sent xml data and also json data both and get whatever response I want to get in xml or json???

Can anyone please help me??

with complete crud application using hibernate or what is possible...

Regards

Manas

[Reply](#)**Pankaj says**

JANUARY 11, 2018 AT 8:32 AM

check this tutorial: <https://www.journaldev.com/8934/spring-rest-xml-and-json-example>

[Reply](#)**J Manas Kumar says**

JANUARY 11, 2018 AT 8:57 PM

I have tried that but it showing this below error.

WARN : org.springframework.web.servlet.PageNotFound – No mapping found for HTTP request with URI [/SpringRestExample/] in DispatcherServlet with name 'appServlet'

How to resolve that?? i have tried a little but unable to do that.

i have changed the url pattern to " /* " but still no effect.

Can u suggest any solution for this.

[Reply](#)**Shweta Chaudhari says**

DECEMBER 27, 2017 AT 6:01 AM

Hi Pankaj,
I am getting following error after deploying the application on Tomcat Server.
SEVERE: Invalid message received with signature 18245
Please help.
Thanks,
Shweta
[Reply](#)

Manoranjan Varsney says

APRIL 12, 2018 AT 3:57 AM

Dont think to much Baby

[Reply](#)

Santhosh K S says

SEPTEMBER 28, 2017 AT 3:04 AM

Thank you very much for giving very much useful post.

[Reply](#)

raaz says

SEPTEMBER 19, 2017 AT 3:00 AM

I am trying to run this project am getting this warn message

WARN : org.springframework.web.servlet.PageNotFound – No mapping found for HTTP request with URI [/spring/] in DispatcherServlet with name 'appServlet'

Could you please help me.

[Reply](#)

kd says

SEPTEMBER 1, 2017 AT 10:03 PM

How can. I secure these web services?

[Reply](#)

Sumit says

APRIL 13, 2017 AT 3:50 AM

In the testGetAllEmployees method we can use Employee[] in place of LinkedHashMap. It will give us the employee objects directly. Modified code is below:-

```
private static void testGetAllEmployee() {  
    RestTemplate restTemplate = new RestTemplate();  
    Employee[] emps = restTemplate.getForObject(SERVER_URI+EmpRestURIConstants.GET_ALL_EMP,  
    Employee[].class);  
    System.out.println(emps.length);  
    for(Employee emp : emps){  
        System.out.println("ID="+emp.getId()+"Name="+emp.getName()+"CreatedDate="+emp.getCreatedDate());  
    }  
}
```

[Reply](#)

Jaz says

APRIL 5, 2017 AT 12:24 PM

Very helpfulThanks

[Reply](#)

Shaswat Talati says

MARCH 13, 2017 AT 3:28 PM

So HttpResponseMessage are already there by default

Why we need to add HttpResponseMessage and RequestMappingHandlerAdapter explicitly ?

[Reply](#)

aayush says

FEBRUARY 13, 2017 AT 1:42 AM

hi

I am learning rest web services.I go through this example ,its easy to understand thanks for it.

Can you please tell me where actually use web services? it returns output in form like above

e.eg

```
{"employees": [  
  { "firstName": "John", "lastName": "Doe" },  
  { "firstName": "Anna", "lastName": "Smith" },  
  { "firstName": "Peter", "lastName": "Jones" }  
]}
```

what its use?why it requires in such form

please clarify this i am confused.

thanks

[Reply](#)

Yogesh says

JUNE 3, 2018 AT 10:17 PM

its very easy. there is nothing in this

[Reply](#)**Alejandro Barrero says**

SEPTEMBER 2, 2016 AT 12:05 PM

To clarify. the error is "GET request for "http://localhost:8080/FIRST_REST/rest/emp/dummy" although I have in the pom.xml

`FIRST_REST``FIRST_REST`[Reply](#)**Pankaj says**

SEPTEMBER 2, 2016 AT 10:00 PM

What is your WAR file name? That becomes the servlet context of the WAR. It should be FIRST_REST.war to work.

[Reply](#)**Uttam K Nanda says**

JANUARY 12, 2017 AT 3:22 AM

Hi Pankaj,

your tutorials are very good and helpful. keep up the good work.

I have one small query –

why the date in Rest client displayed in Long format but not in proper date format ?

Even i face the same problem.

I appreciate your help on this. thank you.

[Reply](#)**Alejandro Barrero says**

SEPTEMBER 2, 2016 AT 11:41 AM

I am getting a 404 error on the client

```
private static void testGetDummyEmployee() {
```

```
    RestTemplate restTemplate = new RestTemplate();
```



```
Employee emp = restTemplate.getForObject(SERVER_URI+EmpRestURIConstants.DUMMY_EMP,
Employee.class);
printEmpData(emp);
}
```

Running on Tomcat

[Reply](#)

Murali says

SEPTEMBER 16, 2016 AT 1:57 AM

Please verify SERVER_URI is not correct.

[Reply](#)

nikhil says

AUGUST 5, 2016 AT 7:33 AM

I am getting below error can anyone help. Thanks

Inside Create Employee

WARN : org.springframework.web.client.RestTemplate – POST request for

"http://localhost:9990/SpringRestExample/rest/emp/create" resulted in 405 (Method Not Allowed);
invoking error handler

Exception in thread "main" org.springframework.web.client.HttpClientErrorException: 405 Method Not
Allowed

at

org.springframework.web.client.DefaultResponseErrorHandler.handleError(DefaultResponseErrorHandler.java:91)

at org.springframework.web.client.RestTemplate.handleResponseError(RestTemplate.java:576)

at org.springframework.web.client.RestTemplate.doExecute(RestTemplate.java:532)

at org.springframework.web.client.RestTemplate.execute(RestTemplate.java:489)

at org.springframework.web.client.RestTemplate.postForObject(RestTemplate.java:318)

at com.journaldev.spring.TestSpringRestExample.testCreateEmployee(TestSpringRestExample.java:43)

at com.journaldev.spring.TestSpringRestExample.main(TestSpringRestExample.java:19)

[Reply](#)

Dmytro says

JULY 13, 2016 AT 5:47 AM

Dear mr.Kumar !

Thank you a lot for helpful tutorial and detaled notes.

[Reply](#)

eli says

JUNE 17, 2016 AT 11:28 PM

thanks alot for your useful tutorial.really thanks.

[Reply](#)**nagaraj says**

MAY 16, 2016 AT 6:35 AM

I am trying the above project the getting some error please known any one reply me

Downloading:

`https://repo.maven.apache.org/maven2/org/codehaus/groovy/groovy/1.8.3/groovy-1.8.3.jar``May 17, 2016 6:49:53 PM org.apache.maven.wagon.providers.http.httpclient.impl.execchain.RetryExec execute``INFO: I/O exception (org.apache.maven.wagon.providers.http.httpclient.NoHttpResponseException) caught when processing request to {``tls}->http://hdc-proxy.wipro.com:8080->https://repo.maven.apache.org:443: The target server failed to respond``May 17, 2016 6:49:53 PM org.apache.maven.wagon.providers.http.httpclient.impl.execchain.RetryExec execute``INFO: Retrying request to {tls}->http://hdc-proxy.wipro.com:8080-``>https://repo.maven.apache.org:443``May 17, 2016 6:50:03 PM org.apache.maven.wagon.providers.http.httpclient.impl.execchain.RetryExec execute``INFO: I/O exception (org.apache.maven.wagon.providers.http.httpclient.NoHttpResponseException) caught when processing request to {``tls}->http://hdc-proxy.wipro.com:8080->https://repo.maven.apache.org:443: The target server failed to respond``May 17, 2016 6:50:03 PM org.apache.maven.wagon.providers.http.httpclient.impl.execchain.RetryExec execute``INFO: Retrying request to {tls}->http://hdc-proxy.wipro.com:8080-``>https://repo.maven.apache.org:443``May 17, 2016 6:50:05 PM org.apache.maven.wagon.providers.http.httpclient.impl.execchain.RetryExec execute``INFO: I/O exception (org.apache.maven.wagon.providers.http.httpclient.NoHttpResponseException) caught when processing request to {``tls}->http://hdc-proxy.wipro.com:8080->https://repo.maven.apache.org:443: The target server failed to respond``May 17, 2016 6:50:05 PM org.apache.maven.wagon.providers.http.httpclient.impl.execchain.RetryExec execute``INFO: Retrying request to {tls}->http://hdc-proxy.wipro.com:8080-``>https://repo.maven.apache.org:443`

Downloaded:

`https://repo.maven.apache.org/maven2/org/codehaus/groovy/groovy/1.8.3/groovy-1.8.3.jar (5394 KB at 106.1 KB/sec)`

[INFO] Generating project in Batch mode

Downloading: <https://repo.maven.apache.org/maven2/org/apache/maven/archetypes/Maven-archetype-quickstart/1.0/Maven-archetype-quickstart-1.0.jar>

[INFO] -----

[INFO] BUILD FAILURE

[INFO] -----

[INFO] Total time: 01:14 min

[INFO] Finished at: 2016-05-17T18:50:56+05:30

[INFO] Final Memory: 10M/40M

[INFO] -----

[ERROR] Failed to execute goal org.apache.maven.plugins:maven-archetype-plugin:2.4:generate (default-cli) on project SpringRestExa

mple: The desired archetype does not exist (org.apache.maven.archetypes:Maven-archetype-quickstart:1.0) -> [Help 1]

[ERROR]

[ERROR] To see the full stack trace of the errors, re-run Maven with the -e switch.

[ERROR] Re-run Maven using the -X switch to enable full debug logging.

[ERROR]

[ERROR] For more information about the errors and possible solutions, please read the following articles:

[ERROR] [Help 1] <http://cwiki.apache.org/confluence/display/MAVEN/MojoFailureException>

C:\nagaraj\SpringRestExample>.49:53 PM

org.apache.maven.wagon.providers.http.httpclient.impl.execchain.RetryExec execute

C:\nagaraj\SpringRestExample> request to [tls]-><http://hdc-proxy.wipro.com:8080->

><https://repo.maven.apache.org:443>

The filename, directory name, or volume label syntax is incorrect.

[Reply](#)

Millan says

AUGUST 26, 2016 AT 9:32 AM

check ur internet connection

clear ur local repository(.m2)

clean ur project, import again

[Reply](#)

LINSEY TIRADO says

MAY 10, 2016 AT 10:52 PM

Nice comments – BTW , if someone is interested in merging of two PDF files , my employees came upon article here <https://disqus.com/by/resto4plate/>.

[Reply](#)

vishal says

MAY 7, 2016 AT 12:38 AM

whenever am running am getting this error with 404 status
org.springframework.web.servlet.PageNotFound – No mapping found for HTTP request with URI [/spring/] in DispatcherServlet with name 'appServlet'

[Reply](#)

Vipul says

APRIL 26, 2016 AT 4:28 AM

Hi,

I want to pass image also in REST API how can i do it ?

Can you please help me ?

Thanks

[Reply](#)

Wu says

FEBRUARY 28, 2016 AT 2:50 PM

Thanks for sharing Pankaj. Could you tell me what the url to get home.jsp page?

[Reply](#)

Mrunmay says

JUNE 25, 2016 AT 2:55 AM

just put the JSP page name in double quotes while returning ModelAndView Object.

For e.g.

```
return new ModelAndView("Home", "msg", "Hello this is Home page");
```

[Reply](#)

Debasish says

FEBRUARY 12, 2016 AT 11:26 AM

Thank you for sharing your knowledge! That message converter part was missing and service was not working for me – this resolved the issue.

[Reply](#)

srinivas says

DECEMBER 15, 2015 AT 4:37 AM

The example is good one.

It would be helpful if you show with hibernate

[Reply](#)**Mamuka says**

NOVEMBER 15, 2015 AT 7:11 PM

Hi All

to produce xml

I added in pom.xml :

org.codehaus.jackson

jackson-mapper-asl

1.9.13

and in requestmapping :

```
@RequestMapping(value = EmpRestURIConstants.DUMMY_EMP, method = RequestMethod.GET  
    ,produces={"application/xml", "application/json"})
```

now depending on headers : accepts – application/xml , or application/json

I am getting xml as well as json

Regards

[Reply](#)**sunil says**

OCTOBER 14, 2015 AT 11:11 PM

Can you please let me know why put used for delete operation.Is there any specific reason?

```
@RequestMapping(value = EmpRestURIConstants.DELETE_EMP, method = RequestMethod.PUT)
```

```
public @ResponseBody Employee deleteEmployee(@PathVariable("id") int empId) {
```

```
    logger.info("Start deleteEmployee.");
```

```
    Employee emp = empData.get(empId);
```

```
    empData.remove(empId);
```

```
    return emp;
```

```
}
```

[Reply](#)**Pankaj says**

OCTOBER 15, 2015 AT 4:56 AM

No, we could have used POST or DELETE also.

[Reply](#)

Abimael says

SEPTEMBER 28, 2015 AT 5:00 PM

Great job!

[Reply](#)**James says**

SEPTEMBER 19, 2015 AT 1:40 PM

Thank you, this helped me very much in learning rest with Spring.

[Reply](#)**Abdul Manaf C J says**

SEPTEMBER 8, 2015 AT 11:38 PM

Hi pankaj,

iam a buginner in spring framework. i need to create a restful webservices using spring mvc. the response in diffrent format like xml, json, plaintext. iam download your souce code. it working fine. but how to add xml serialization to this project. please reply

[Reply](#)**Vibhav says**

AUGUST 29, 2015 AT 9:57 AM

I am new to spring .Following your tutorials are a real joy and I am really confident that all Spring mistry will be solved with this. Really you are doing a great work . Hats off to you and keep up the good work.

[Reply](#)**Mamuka says**

AUGUST 10, 2015 AT 8:02 PM

what do you mean don't copy it's bad karma ?

[Reply](#)**kasun silva says**

JULY 22, 2015 AT 5:13 AM

Hi Pankaj,

It's great example. Is there any related maven archetype for your example?

[Reply](#)

Shailesh says

JULY 1, 2015 AT 1:37 AM

If we want to process data before passing to one of the method of controller then how we can do this in spring framework

[Reply](#)

Ricardo Tannure says

JUNE 18, 2015 AT 6:25 AM

Hi Pankaj,

I have the problem below:

The @RequestBody object is null when I use POST Service.

The client is over IIS with .Net.

The RestService is over JBoss 7.0 and the code is the same yours.

Could you help me with that?

Thanks,

Ricardo

[Reply](#)

Patil says

MAY 27, 2015 AT 10:41 PM

when i post rest service, it is coming as string instead of object in parameter. i have copied the same code. do i need to add any configuration ?

[Reply](#)

Patil says

MAY 27, 2015 AT 10:39 PM

Hi All,

when i post service, it is coming as string instead of object. I have copied the same example.

Thanks,

-Patil

[Reply](#)

sujit says

MAY 21, 2015 AT 9:58 PM

Thank's a lot,It is very helpful.

[Reply](#)**BrahmaReddy says**

APRIL 27, 2015 AT 1:16 AM

When i try to fetch the lazy load entites i am getting this error,

HTTP Status 500 – Could not write JSON: could not initialize proxy – no Session (through reference chain: java.util.ArrayList[0]->org.commerce.hibernate.Tblcategory["tblcreation"]->org.commerce.hibernate.Tblcreation_\$\$_jvstcdg_5["createdByUserid"]); nested exception is com.fasterxml.jackson.databind.JsonMappingException: could not initialize proxy – no Session (through reference chain:

[Reply](#)**Dxtr says**

MAY 12, 2016 AT 4:22 AM

any solution for this?

[Reply](#)**Sankardoss says**

APRIL 26, 2015 AT 11:28 PM

Thanks for the post. I have tried the Employee example mentioned above. I get org.springframework.web.client.HttpClientErrorException: 405 Method Not Allowed exception while executing the test program. Please help. Thanks

[Reply](#)**Riley says**

APRIL 12, 2015 AT 9:18 AM

first-rate tutorial. thank you sir.

[Reply](#)**Krishanu says**

APRIL 9, 2015 AT 2:50 AM

How to run the jar file....

i got error while running the jar file creating by Spring tools suite maven.

Error message is "no main manifest attribute "..

plz help me out... or tell me other process to run.

[Reply](#)

Pankaj says

APRIL 9, 2015 AT 5:02 AM

Check out this article <https://www.journaldev.com/1344/how-to-run-jar-file-in-java>

[Reply](#)

Ivan says

APRIL 6, 2015 AT 5:11 AM

For createEmployee() method after setting Content-Type to "application/json" I always get HTTP Status 400 "The request sent by the client was syntactically incorrect." How to fix this?

[Reply](#)

Navaneet says

APRIL 5, 2015 AT 1:04 AM

Thanks...Great way of explanation..Very useful.

[Reply](#)

Navaneet says

APRIL 5, 2015 AT 1:01 AM

Thanks...great way of explanation. very useful.

[Reply](#)

Rosyid says

MARCH 24, 2015 AT 7:41 PM

Hi Pankaj i have deployed successfully this app in Tomcat, using java client and rest client application, for both i am getting 404 Error. My context path is same as SpringRestExample. Can you pls let me know what can be the issue?

[Reply](#)

abhilash says

MARCH 22, 2015 AT 9:12 AM

Hi Pankaj, for posting an xml file, I have written a function as

```
@RequestMapping(value = EmpRestURIConstants.UPLOAD_FILE, method = RequestMethod.POST,
consumes=MediaType.MULTIPART_FORM_DATA_VALUE)
public @ResponseBody byte[] testPost(@RequestParam("file") MultipartFile file, HttpServletResponse
response){
}
```

and Client program is written as,

```
MultiValueMap parts = new LinkedMultiValueMap();
parts.add("Content-Type", MediaType.APPLICATION_XML);
parts.add("file", resource);
ResponseEntity re = template.exchange(url, HttpMethod.POST,
new HttpEntity<MultiValueMap>(parts),
String.class);
```

But I am getting org.springframework.web.client.HttpServerErrorException: 500 Internal Server Error
Whats wrong with Client Code or Server Side Program?

[Reply](#)**abhilash says**

MARCH 22, 2015 AT 12:10 AM

Hi Pankaj i have deployed successfully this app in Tomcat, using java client and rest client application, for both i am getting 404 Error. My context path is same as SpringRestExample. Can you pls let me know what can be the issue?

[Reply](#)**abhilash says**

MARCH 22, 2015 AT 12:50 AM

Its Solved and working fine. now

[Reply](#)**srvjena7@gmail.com says**

JULY 28, 2015 AT 2:34 AM

how u sloved the issuei am also geting same error can u please help me

[Reply](#)**mohan says**

APRIL 12, 2017 AT 4:10 PM

check your scan controller package.

change if you change the according to your controller package

[Reply](#)

David Brown says

MARCH 5, 2015 AT 2:46 PM

Howdy Pankaj, very well done, good clean presentation. Only 1 oddity. The TestSpringRestExample client runs correctly. Using postman all the GETs work but not the POST for creating a new employee. Return status: 400 The request sent by the client was syntactically incorrect. I suppose I could try to capture the actual POST request but Postman seems to not have this feature.

POST /rest/emp/create HTTP/1.1

Host: localhost:8080

Content-Type: application/json

Cache-Control: no-cache

[Reply](#)

Pankaj says

MARCH 6, 2015 AT 12:01 AM

If test java program is working, then your project is having no issues. Just check if you are missing payload in the POST?

[Reply](#)

Mario amado says

FEBRUARY 24, 2015 AT 9:14 AM

I read carefully your posts i i found them very good!

So i would like to know how can i write a json with this format:

```
{
  "people": [
    {
      "data": {
        "name": "José Sócrates",
        "genre": "male",
        "birthdate": "1930/8/25",
        "profession": "Político",
        "image": "image1.png",
```

```
"wikipediaUrl": "http://wikipedia.com",
"desc": "José Sócrates Carvalho Pinto de Sousa é um político português. Foi secretário- geral do Partido Socialista, de Setembro de 2004 a Julho de 2011 e Primeiro-ministro de Portugal de 12 de março de 2005 a 21 de junho de 2011.",
"Known_as": "José Sócrates ",
"Equivalente_webpage": "http://wikipedia.com",
"Oficial_website": "http://wikipedia.com",
"social_media_presence": "JealHouse ",
"Description": "Description ",
"Place_of_birth": "Portugal",
"Nationality": "Portuguese",
"Ethnicity": " Unknow",
"Parents": "Motherr",
"Children": " 2",
"Sibling_s": " other pages"
}
}
}
}
```

[Reply](#)**Vivek says**

JULY 14, 2016 AT 8:56 PM

It should be simple. create a model class,

```
public class Person implements Serializable {
    List data;
}
```

create another class called as Data which has all attributes like name, birthdate,profession.

Use this model class to map data from Dao, and assign it to the person.setData(AnotherModelClass);
return it from controller. Thats it.

[Reply](#)**ashok says**

FEBRUARY 18, 2015 AT 11:51 AM

hi,

how can pass the uri to the rest template dynamically

thanks,

ashok

[Reply](#)

Ashwin says

FEBRUARY 18, 2015 AT 3:57 AM

Nice Article

suppose if i want to send messages in xml format?? Is it possible

[Reply](#)**karthick says**

FEBRUARY 16, 2015 AT 10:38 PM

Thanks for this article, it really helps. But i am not able to build this project, i haven't used maven before. Can you please explain the steps on how to build this project using maven ?

[Reply](#)**Ashwin says**

FEBRUARY 18, 2015 AT 4:00 AM

If you are working on eclipse, download Maven, and Spring tool suite. from Eclipse market place. then click on new project, Spring mvc.

specify the top level package and click finish.. You will see every bit of code which is shown above.

[Reply](#)**Anmol says**

FEBRUARY 16, 2015 AT 7:43 PM

Thanks a lot Pankaj!!! your example was really easy to follow unlike many other examples on web which are hard to understand.

[Reply](#)**thxill says**

FEBRUARY 7, 2015 AT 8:44 PM

Where is the download link? I can't see any.

[Reply](#)**Naveen says**

FEBRUARY 5, 2015 AT 10:23 AM

Hi PK,

You are making our life simple. Thanks alot for your great effort.

[Reply](#)

Umang Rathod says

FEBRUARY 4, 2015 AT 11:38 PM

can we do this example with different two model class?like employee and Student ?

[Reply](#)

Ashwin says

FEBRUARY 18, 2015 AT 4:04 AM

yes you can..

you can write a different controller class and create new urls to access your other model class.

you can then send your new model object in @requestbody and @responsebody.

[Reply](#)

rajan says

JANUARY 31, 2015 AT 3:16 AM

Hello sir,

This tutorial very helpful, I learned a lot. I want to display json data into jsp page. how do I do that. please

can you alter this tutorial and consume rest json to display from jsp

page.

expecting your help sir,

thanks

rajan

[Reply](#)

hairale says

JANUARY 13, 2015 AT 2:20 AM

hi. Can i help me create project with sub project use maven . you can for me sample .thank you .

[Reply](#)

ritesh says

JANUARY 8, 2015 AT 9:00 PM

can anybody plz tell me step by step. because i dont have any idea. ow to create a spring mvc rest service....plzz

[Reply](#)**Pankaj says**

JANUARY 9, 2015 AT 1:41 AM

Every step is given in the tutorial, even project is available for download. What else you are looking for?

[Reply](#)**Ashwin says**

FEBRUARY 18, 2015 AT 4:09 AM

Check my reply to Karthick. Once downloaded, you can proceed with all the steps Pankaj has mentioned.

[Reply](#)**Chandan says**

DECEMBER 19, 2014 AT 11:24 PM

i am getting this warning

WARN : org.springframework.web.servlet.PageNotFound – No mapping found for HTTP request with URI [/SpringRestExample/] in DispatcherServlet with name 'appServlet'

how to solve it

[Reply](#)**Ashwin says**

FEBRUARY 18, 2015 AT 4:12 AM

Check if you have configured the @RequestMapping properly.

Also check the base scan package in your dispatcher servlet is indeed scanning your controller class.

[Reply](#)**Hemant Mali says**

DECEMBER 8, 2014 AT 9:22 PM

Hi Pankaj,

Thanks for the tutorial. I have one question: I am not using STS, then what change I have to make in my project setup to run this application. I am using Eclipse Juno and Maven.

[Reply](#)**Pankaj says**

DECEMBER 9, 2014 AT 2:09 AM

STS provide option to create skeletal project, you can install the Spring Framework plugin in Eclipse too, search for that in marketplace. If you don't want to use these plugins, you will need to create a maven project yourself and add all the dependencies manually. That's all.

[Reply](#)**pushpendra kumar says**

NOVEMBER 30, 2014 AT 11:42 PM

Can you tell me how I can make a simple signup sample with this approach with this approach(data validations are also there)?

[Reply](#)**pushpendra kumar says**

DECEMBER 2, 2014 AT 4:18 AM

or how we can make android client for this .

[Reply](#)**nitin says**

NOVEMBER 27, 2014 AT 9:00 AM

Hello Pankaj,

Thanks for the tutorial , very helpful to understand.

I am seeing 404 , the only warning i see is –

INFO [http-apr-8080-exec-10] org.apache.catalina.core.ApplicationContext.log No Spring WebApplicationInitializer types detected on classpath

any idea why this could be?

i tried project from scratch and also with your files...

I searched on net, and tried all possible steps to eliminate it. But no luck

i tried with jre 6,7; created new workspace etc....

Please help.

If any one else has seen this issue , please advise...

Thanks

Nitin

[Reply](#)

jun says

NOVEMBER 17, 2014 AT 12:33 AM

Thanks ojiisan!

[Reply](#)**Shiva says**

NOVEMBER 10, 2014 AT 12:47 AM

Can you please provide the difference between PUT and POST with respect to the article?

[Reply](#)**Dakota says**

NOVEMBER 9, 2014 AT 7:29 PM

Great help in this tutorial. It's really useful for people who aren't using Spring Boot- thanks!

[Reply](#)**Mohammad says**

NOVEMBER 6, 2014 AT 12:36 AM

Hi Pankaj,

Thanks for your clear example, I really enjoy to read tutorials you've provided, it is awesome.

I'm trying to provide an API for my project and I've used Jackson convertor, it works fine however I need to add xml convertor to my API as well (e.g. /api/rest/xml besides /api/rest/json and so on). I realized a message convertor like MappingJackson2HttpMessageConverter should be added but I'm not sure how can I distinguish XML requested not JSON? it seems this convertors will convert all requests to JSON and Objects vice versa. Could you please let me know how to solve it, thanks a lot.

[Reply](#)**Guillermo says**

FEBRUARY 18, 2017 AT 3:55 AM

If somebody have this error again.

I solve it writting manually the class for jsonMessageConverter.

Probably a crazy error for format in copy/paste ☐

[Reply](#)

Demetrio says

OCTOBER 24, 2014 AT 7:40 AM

WARN : org.springframework.web.client.RestTemplate – GET request for "http://localhost:8080/SpringRestExample/rest/emp/dummy" resulted in 404 (Not Found); invoking error handler

I am trying to debug but I am getting the warinning above. I placed a break in TestSpringRestExample and it is breaking in RestTemplate in response.close();

protected T doExecute(Uri url, HttpMethod method, RequestCallback requestCallback, ResponseExtractor responseExtractor) throws RestClientException {

...

```
finally {  
    if (response != null) {  
        response.close();  
    }  
}
```

[Reply](#)**Mario M. says**

OCTOBER 23, 2014 AT 1:25 AM

first of all good work and thanks for the effort. I have a rather silly question, if my client is a web browser like mozilla, to create an employee: What would be the proper url string for this service using JSON notation?

ie "http: // localhost:8080/SpringRestExample/rest/emp/create ????"

Thanks in advance

[Reply](#)**CH says**

OCTOBER 18, 2014 AT 7:51 AM

When I run the example the URL I must use is `http://localhost:8080/spring/` and not `http://localhost:8080/SpringRestExample/`. What is it that is determining "spring" vs.

"SpringRestExample" in the URL? I though this was based on the application name.

I ran your Spring MVC example and the URL is `localhost:8080/SpringMVCExample/` as expected. I have compared the two projects and I cannot see a difference that would map the url differently.

[Reply](#)**Pankaj says**

OCTOBER 18, 2014 AT 10:00 AM

Its because of the application context file automatically created by STS. Best option is to export as war file into separate container.

[Reply](#)

simon says

SEPTEMBER 26, 2014 AT 1:24 AM

could you pls send me the sample project.

I do not have a google account.

tk's for much.

mail address: 8923040@qq.com

[Reply](#)**chan says**

SEPTEMBER 10, 2014 AT 3:10 AM

Thanks. I have one question. suppose if i use browser as my client to access – testGetAllEmployee(). i.e in browser if i call url for testGetAllEmployee() then i want show all details in html/css format instead of json format. how can i do it?

[Reply](#)**chan says**

SEPTEMBER 10, 2014 AT 3:15 AM

Adding some doubts.

suppose if we access the url then it will return json format response. so anybody can call this url , they can get response as json and they can use it their website/any where. how can we provide security?

[Reply](#)**Pankaj says**

SEPTEMBER 10, 2014 AT 5:23 AM

1. If you want to present HTML, then you can forward the JSON to JSP page and render it as HTML response.
2. There are different ways to make your application secure, easiest is to ask user to send UserID and Password as header data in HTTP request and validate it against your user database.

[Reply](#)**chan says**

SEPTEMBER 15, 2014 AT 5:55 AM

Thanks.

my website don't have any login page at all. anybody can come and see required information. suppose i have spring controller which will return json object based on user

URI. On jsp page i will render this json object and display content using front end technologies.

My question is if anybody trying to retrieve URI then it will json object, so how can i secure URI?

example – <http://localhost:9090/SpringRestExample/rest/emp/1> – this will return emp record 1 in json format. so anybody can call this URI in their website and can get information and display into their website. how can i secure this one no body cant access other than my website.

Thanks

[Reply](#)

Sapna says

OCTOBER 28, 2014 AT 1:29 PM

hi Pankaj,

do you have any thoughts on the above question.

I too have the same thoughts in mind..

Thanks in advance!

Rohit More says

SEPTEMBER 10, 2014 AT 2:03 AM

Hello Pankaj,

I appreciate your efforts for us like beginner. I went through your blog and followed instruction its working perfectly alright. I have small doubt.

If I have controller with 1 function

```
@RequestMapping(value="/getvalue", method=RequestMethod.GET)
```

```
public String getStatastics(String string1, String string2, String string3, EnumDemo demo, Model model){
```

```
String resultedString = string1 + string2 + string3;
```

```
System.out.println(demo.enumValue1);
```

```
return resultedString;
```

```
}
```

Now I want to create client with calling this above function.

I would be really happy if you provide me some direction.

Thanks.

[Reply](#)

Vinod says

AUGUST 26, 2014 AT 2:36 PM

I tried downloading the example, and importing to eclipse. When I tried running the application on server. (after doing the mvn-clean and install – both complete successfully) I get a 404 with a warning message on the console :

WARN : org.springframework.web.servlet.PageNotFound – No mapping found for HTTP request with URI [/SpringRestExample/] in DispatcherServlet with name 'appServlet'

Any idea what I'm doing wrong here?

[Reply](#)

Pankaj says

AUGUST 27, 2014 AT 1:21 AM

Check your war file name, if you are deploying on tomcat, your servlet context will be the file name (excluding .war extension)

[Reply](#)

Vinod says

AUGUST 27, 2014 AT 9:36 AM

I'm trying to run the project through eclipse. Project (Right Click) ->Run As ->Run On Server. And I get the 404 error. Is there something different I should be doing?

[Reply](#)

Jayaprakash says

SEPTEMBER 2, 2014 AT 3:59 AM

I am also getting the same error.I changed the servlet-context.xml file name to SpringRestExample.xml and change the path in web.xml to /WEB-INF/spring/appServlet/SpringRestExample.xml.But getting same 404 error.Please help

[Reply](#)

Gagz says

SEPTEMBER 27, 2014 AT 8:33 PM

i am also getting same error. please let me know how to resolve.

No mapping found for HTTP request with URI [/spring/] in DispatcherServlet with name 'appServlet'

[Reply](#)

Arvind Singh says

DECEMBER 16, 2014 AT 3:30 AM

Hi Vinod,

How you resolve this issue, i am facing same issue.

[Reply](#)**gurjinder singh says**

AUGUST 19, 2014 AT 12:31 AM

thanks your code helps me a lot

[Reply](#)**devloper says**

AUGUST 8, 2014 AT 8:19 AM

Great article! I am able to create web service using this article, thank you so much!

[Reply](#)**lsm says**

AUGUST 5, 2014 AT 5:02 PM

Hi!, firstly thank you for this great work.

And secondly, could you post something about how to serialize a Hibernate model class (make a JSON), please.

I tried to convert the Employee class model to a Hibernate class model, and then get data directly from a database. But a get this error : "Could not write JSON: No serializer found for class org.hibernate.proxy.pojo.javassist.JavassistLazyInitializer and no properties discovered to create BeanSerializer (to avoid exception, disable SerializationFeature.FAIL_ON_EMPTY_BEANS)) (through reference chain: com.journaldev.spring.model.Person_\$\$jvstf07_0["handler"]); nested exception is com.fasterxml.jackson.databind.JsonMappingException: No serializer found for class org.hibernate.proxy.pojo.javassist.JavassistLazyInitializer and no properties discovered to create BeanSerializer (to avoid exception, disable SerializationFeature.FAIL_ON_EMPTY_BEANS)) (through reference chain: com.journaldev.spring.model.Person_\$\$jvstf07_0["handler"])"

All works great if I make a data bind between a new POJO Employee model and his Hibernate model counterpart.

Thanks!

[Reply](#)

devloper says

AUGUST 8, 2014 AT 8:21 AM

implement Serializable interface

[Reply](#)**Jorel says**

JULY 28, 2014 AT 7:31 AM

Hello! Really good job! it's the first project which work without editing somethings.

I have a question how to create an user without the Java program, I tryed this request but without success.... Have you got an issue?

```
curl -X POST -H "Content-type:application/json" -H "Accept:application/json" -d '{"id":2,"name":"JLabel"}'
http://localhost:8080/SpringRestExample/rest/emp/create
```

[Reply](#)**Lars says**

JULY 31, 2014 AT 3:21 PM

I got the same problem here on a Win 7 System. I read that there could be some problems with curl for windows (if you use that). In that case the syntax :

```
curl http://localhost:8080/SpringRestExample/rest/emp/create -X POST -H "Content-
type:application/json" -d @json.txt
```

worked for me with the json inside the txt file.

Seems to be an issue with the single quotes in win command line. Cygwin should also do the job.

Best regards,

Lars

[Reply](#)**Shiv says**

JULY 27, 2014 AT 12:15 AM

How could we have XML as Response. I have a requirement to have my GET call response back with Json or XML as per client request. Please advise.

[Reply](#)**Rohit says**

JULY 22, 2014 AT 5:19 AM

Thank you for this article.

This is really helpful.

Thanks a ton.

[Reply](#)

Volodia says

JULY 21, 2014 AT 3:36 AM

This line can be changed

```
restTemplate.getForObject(SERVER_URI+EmpRestURIConstants.GET_ALL_EMP, List.class
```

to

```
restTemplate.getForObject(SERVER_URI+EmpRestURIConstants.GET_ALL_EMP, Employee[].class
```

Now you have array of mapped Employees and can easily create any collection.

[Reply](#)

Rahul says

JUNE 21, 2014 AT 7:15 PM

Extremely helpful tutorial ☐

I have a small doubt :

In pom.xml I ad to comment this :

```
<!--
```

```
org.springframework
```

```
spring-context
```

```
${org.springframework-version}
```

Exclude Commons Logging in favor of SLF4j

```
commons-logging
```

```
commons-logging
```

```
-->
```

and instead have to add this :

```
org.springframework
```

```
spring-tx
```

```
4.0.0.RELEASE
```

And then why pom.xml reports "ArtifactTransferException: Failure to transfer org.springframework:spring-context:jar:4.0.0.RELEASE from <http://repo1.maven.org/maven2> was cached in the local repository,....." ?

I have cleared the local repository after that, but the problem still persists.

Any suggestions please ☐

[Reply](#)

Pankaj says

JUNE 22, 2014 AT 2:37 AM

You need to force update the dependencies.
you need to have constant defined for the versions.

[Reply](#)

Wendy says

JUNE 20, 2014 AT 12:19 PM

Hi Pankaj,

Thank you very much for this nice and thorough article. I followed it through and was able to successfully test it without using WizTools RestClient.

I just have one question: I found your article by search "Spring Rest service eclipse json" on google. I was expecting the response format is in Json. When you get a chance, could you explain how to get the response in Json?

Thanks,

Wendy

[Reply](#)

smartboy says

JUNE 19, 2014 AT 2:38 AM

Good explanation

but I don't find any URL to download your project after following you on twitter

[Reply](#)

Víctor Hernández says

JUNE 16, 2014 AT 8:44 PM

First of all thanks for the tutorial, it has helped me a lot, try to test the createEmployee method, however I did not succeed, try it through a jsp with ajax and then try using postman, I think the problem is trying to send objects and hope you can help me, greetings from Mexico.

[Reply](#)

Sam says

JUNE 15, 2014 AT 9:15 AM

Your download link still has a problem. After downloading, the file size is only 9K and nothing in the zip file except ".settings" directory. Could you please verify? Thank you.

[Reply](#)

Pankaj says

JUNE 15, 2014 AT 6:56 PM

Sorry about that, I have fixed it and verified myself, it should be working fine now.

[Reply](#)**Sudha says**

JUNE 12, 2014 AT 10:16 PM

Hi ,

I want to write a jersey client for file upload (which is a spring rest)

```
@RequestMapping(method = RequestMethod.POST, consumes = "multipart/form-data")
```

```
@ResponseBody
```

```
public String upload(@RequestParam("file") MultipartFile file)
```

this is how my code will look like.

Please provide me some idea. I tried with some example but i am getting this error

org.glassfish.jersey.message.internal.MessageBodyProviderNotFoundException: MessageBodyWriter not found for media type=multipart/form-data, type=class com.sun.jersey.multipart.MultiPart, genericType=class com.sun.jersey.multipart.MultiPart.

[Reply](#)**Nagaraju says**

JUNE 10, 2014 AT 10:54 PM

nothing is there in u r downloaded file yaar.....

[Reply](#)**Pankaj says**

JUNE 11, 2014 AT 10:33 AM

it was an issue with the plugin I use for managing downloads, solved it. Please try now.

[Reply](#)**mallikarjuna rao says**

JUNE 10, 2014 AT 10:41 PM

Hi Pankaj,

Its really great article. there are no files in the zip file when I open that. could you please share the complete project.

Best Regards

Mallik

[Reply](#)

Nagaraj says

JUNE 5, 2014 AT 5:31 AM

Today changed jre settings and deployed. I could see some errors in the console. Please let me know if you are getting any idea based on the error trace below.

SEVERE: ContainerBase.addChild: start:

org.apache.catalina.LifecycleException: Failed to start component
[StandardEngine[Catalina].StandardHost[localhost].StandardContext[/spring]]

Caused by: java.lang.NoClassDefFoundError:

org.springframework.web.context.WebApplicationContext

SEVERE: Error deploying configuration descriptor

C:\MarketPlace\DevTools\SpecialProject\springsource\vfabric-tc-server-developer-
2.9.3.RELEASE\base-instance\conf\Catalina\localhost\spring.xml

java.lang.IllegalStateException: ContainerBase.addChild: start: org.apache.catalina.LifecycleException:
Failed to start component [StandardEngine[Catalina].StandardHost[localhost].StandardContext[/spring]]

[Reply](#)

Anand says

OCTOBER 14, 2014 AT 3:04 AM

Nagaraj , did u get any soln reg this i tried many soln which mentioned in hte net.
like

1. remove all servers and clear the servers folder in eclipse workspace folder
2. clean, publish, etc,
3. adding common-loggins jar to dependency.

None works out, could u please help me with your soln..

[Reply](#)

Nagaraj says

JUNE 4, 2014 AT 5:16 AM

It's a good example to begin with. Easy to understand than other tutorials related to Spring RESTful Services. I used STS and created this project as Spring MVC. I didn't download your project. But created the structure and contents as mentioned here.

I deployed with VMware vFabric tc Server Developer Edition v2.9 .. It looks to be deployed properly but on invocation getting 404. Tried with as well as in Tomcat 7 also same issue. Getting 404 in browser as well as in WizTools RestClient 3.2.2

I created a Sample Dynamic Web project, that got deployed and its rendering the required output.

There is only one warning in the server console and i searched in Web, its mentioned nothing to worry about this.

WARNING: [SetContextPropertiesRule][Context] Setting property 'source' to 'org.eclipse.jst.jee.server:SpringRestExample' did not find a matching property.
Your advice would be helpful. I can attach the server log if needed.

[Reply](#)

Nagaraj says

JUNE 4, 2014 AT 5:44 AM

I downloaded the Project and tried. Even then getting the same issue.

[Reply](#)

Michal Skultety says

JUNE 4, 2014 AT 12:52 AM

Thanks, the example works (which is not so common with many EE examples) and is reduced to a minimum to demonstrate just the technology without any useless patterns show-off, I really like it...

[Reply](#)

Pradeep says

JUNE 3, 2014 AT 12:47 PM

Hi, I have deployed the above code in WAS 7. getting below error.

Permission:

getenv.* : Access denied (java.lang.RuntimePermission getenv.*)

Code:

org.springframework.core.env.AbstractEnvironment in {file:/opt/httpd/root/apps/sample/ibm7/dev-ear/sample.war/WEB-INF/lib/spring-core-4.0.0.RELEASE.jar}

Could you please suggest me how can i override this error.

I have tried the below options, but no luck

1) added below two lines in root-context.xml

2) added below line in spring.properties copied in classes folder

spring.getenv.ignore=true

Thanks,

Pradeep.

[Reply](#)

Pradeep says

JUNE 3, 2014 AT 12:49 PM

added below code in root-context.xml but no luck

[Reply](#)**Pankaj says**

JUNE 3, 2014 AT 8:01 PM

From the error it looks like permission issue, I have never faced this issue but one pointer I can give you is to check any security manager configuration or policy files are present that is preventing environment variables access?

[Reply](#)**ranjith says**

MAY 28, 2014 AT 1:25 AM

I am new to spring, i have imported your project but getting error in servlet-context.xml error like,

Class 'org.springframework.http.converter.json.MappingJacksonHttpMessageConverter' not found [config set: SpringRestExample/web-context]

[Reply](#)**Pankaj says**

MAY 28, 2014 AT 9:11 AM

Check your spring version and jars for this class.

[Reply](#)**Mohan says**

MAY 27, 2014 AT 5:28 AM

How can I implement spring validators in the example? I saw that you have already explained spring validators in [Post](#).

[Reply](#)**Thomas says**

MAY 16, 2014 AT 6:19 AM

You should really overthink your "RESTful" URL design in this example.

RESTful URLs do not have any verbs in them, just nouns. So the right URLs for your example would be:

GET /employees (get a list of all employees)

POST /employees (create a new employee)

GET /employees/id (get a specific employee from id)

DELETE /employees/id (delete a specific employee from id)

Although I appreciate your motivation to provide examples for other people on the internet, you should ensure that you do not teach wrong things.

[Reply](#)

Pankaj says

MAY 17, 2014 AT 10:43 AM

Yes you are right, but it's for example only to show how it works. I am sure if you are working on a real project, you will have your own Rest URIs and won't go for this structure.

The takeaway of this tutorial is to learn how to implement Restful web service in Spring.

[Reply](#)

Mortier says

MAY 13, 2014 AT 11:40 PM

Perfect!, first example i see to extract the paths with enums, now take it to the next level and implement Hateoas? ☐

[Reply](#)

Bhabadyuti says

MAY 12, 2014 AT 8:34 PM

Sir at which you have used restful web services i mean how to know that this application using restful web service. ??

please make me clear..

Thanks

[Reply](#)

Pankaj says

MAY 12, 2014 AT 11:19 PM

I think you should read about Restful on wikipedia.

[Reply](#)

Anil says

MAY 6, 2014 AT 10:22 PM

I want to expose rest service to receive byte array, so in my service i used

@RequestBody final byte[] byteArray

This is resulting me below exception

org.codehaus.jackson.map.JsonMappingException: Can not deserialize instance of byte[] out of START_OBJECT token

Could you please help me out in this.

[Reply](#)**Pankaj says**

MAY 7, 2014 AT 12:08 AM

you can't sent byte[] because as error says, it should be a serialized object.

[Reply](#)**spring788@hotmail.com says**

MAY 6, 2014 AT 2:13 AM

How did you come up with the name "spring Jackson" for your program?..spring Jackson, oregon

[Reply](#)**Pankaj says**

MAY 6, 2014 AT 9:51 AM

Thats a coincidence, Spring and Jackson both are Java frameworks. ☐

[Reply](#)**Narayana says**

APRIL 8, 2014 AT 11:26 AM

Can we make this application to support any kind of format(json,xml, etc) to generate and also to receive ..?

[Reply](#)**Ana says**

APRIL 5, 2014 AT 12:15 PM

I am trying to do an API that will receive a list of Customers for example an this list is processed by the service, but I have some questions:

- If I receive the list of Customers inside the Body as @RequestBody then how should I process it.
- My controller must process all in a once because we have to process all the customers or none.
- I am trying to deserialize the Jason content to a llist of customers, but after this I should process customer by customer invoking the service n times but my transactions are at service level not at controller level. . Could you please give me any idea about how to do it?

I am doing my final degree work and I am really stuff by this.

Many thanks,

Ana

[Reply](#)

Pankaj says

APRIL 5, 2014 AT 7:23 PM

Once you have received the list in controller request handler method, its upto you how to process it. I am assuming there will be some DB calls, third party calls etc. You can iterate over the list and do it one by one or in batch.

[Reply](#)

ashok says

APRIL 1, 2014 AT 9:43 AM

Thanks a lot boss, quick reference.

Small doubt, in controller class, while creating employee you are purring created emp in to a map (empData) , but you are returning pojo class object (emp). Why so ?

Any ways Thanks !

[Reply](#)

Pankaj says

APRIL 1, 2014 AT 3:41 PM

Here Map is our data storage system that holds all the Employee objects. Based on the rest call, we are returning one employee or list of employees.

[Reply](#)

ashok says

APRIL 4, 2014 AT 6:45 AM

Thank you Pankaj.

I am new to spring, just trying to understand your example and implement it in my app.

One more doubt, i am using spring 3.0.5, would like to use same service in webapp and mobile client.

depends up on request, i've to send response.

if request from webapp, have to send ModelAndViewObject which will go through ViewResolver.

if it is from Mobile, have to send JOSN object

Could it possible ? if so please help on configurations.

Thanks in advance.

Ashok

[Reply](#)

Scott says

MARCH 23, 2014 AT 4:03 AM

I cannot tell you how thankful I am for this. After banging my head against the wall for half the day you have given me the "magic" configuration that will correctly return json from my REST controllers. Thank you, thank you, thank you!

Note to anyone using Spring 4 and getting a 406 HTTP response when trying to make a call to your controllers: just copy the spring-context.xml file on this page and be done with it.

[Reply](#)

Pankaj says

MARCH 23, 2014 AT 2:07 PM

Thanks for the kind words, I really appreciate it.

[Reply](#)

mallikarjuna rao says

JUNE 9, 2014 AT 4:27 PM

this example is not in readable when I extract it. please provide complete example with source.

Thanks

Mallik

[Reply](#)

mallikarjuna rao says

JUNE 9, 2014 AT 4:34 PM

its asking user name and password while extracting. zip file does not contain any files with out username and password.

[Reply](#)

sreeni says

MARCH 7, 2014 AT 7:20 PM

Great work, very helpful.

[Reply](#)**mallikarjuna Rao says**

JUNE 10, 2014 AT 10:44 PM

please share the complete project code.

[Reply](#)**Pankaj says**

JUNE 11, 2014 AT 10:35 AM

it was an issue with the plugin I use for managing downloads, solved it. Please try now.

[Reply](#)

Leave a Reply

Your email address will not be published. Required fields are marked *

Comment

Name *

Email *

☐

Save my name, email, and website in this browser for the next time I comment.

[DOWNLOAD ANDROID APP](#)



[SPRING FRAMEWORK](#)

Spring Tutorial

Spring Core

- > [Spring Framework](#)
- > [Spring Dependency Injection](#)
- > [Spring IoC and Bean](#)
- > [Spring Bean Life Cycle](#)
- > [Spring REST](#)
- > [Spring REST XML](#)
- > [Spring RestTemplate](#)
- > [Spring AOP](#)
- > [Spring AOP Method Profiling](#)
- > [Spring Annotations](#)
- > [Spring @Autowired](#)
- > [Spring @RequestMapping](#)

Spring MVC

- > [Spring MVC Example](#)
- > [Spring MVC Tutorial](#)
- > [Spring MVC Exception Handling](#)
- > [Spring MVC Validator](#)
- > [Spring MVC Interceptor](#)
- > [Spring MVC File Upload](#)
- > [Spring MVC i18n](#)
- > [Spring MVC Hibernate MqSQL](#)

Spring ORM

- > [Spring ORM](#)
- > [Spring ORM JPA](#)
- > [Spring Data JPA](#)
- > [Spring Transaction](#)

- > [Spring JdbcTemplate](#)

Spring Security

- > [Spring Security Overview](#)
- > [Spring Security Example Tutorial](#)
- > [Spring Security UserDetailsService](#)
- > [Spring MVC Login Logout](#)
- > [Spring Security Roles](#)

Spring Boot

- > [Spring Boot Tutorial](#)
- > [Spring Boot Components](#)
- > [Spring Boot CLI Hello World](#)
- > [Spring Boot Initilizr Web](#)
- > [Spring Boot Initilizr IDE](#)
- > [Spring Boot Initilizr CLI](#)
- > [Spring Boot Initilizr Tools](#)
- > [Spring Boot MongoDB](#)
- > [Spring Boot Redis Cache](#)
- > [Spring Boot Interview Questions](#)

Spring Batch

- > [Spring Batch](#)
- > [Spring Batch Example](#)

Spring AMQP

- > [Spring AMQP](#)
- > [Spring RabbitMQ](#)
- > [Spring AMQP RabbitMQ](#)
- > [Apache ActiveMQ](#)
- > [Spring ActiveMQ Tutorial](#)
- > [Spring ActiveMQ Example](#)

Spring Integrations

- > [Spring JDBC](#)
- > [Spring DataSource JNDI](#)
- > [Spring Hibernate](#)
- > [Spring Primefaces JPA](#)
- > [Spring Primefaces MongoDB](#)
- > [Spring Primefaces Hibernate](#)
- > [SpringRoo Primefaces Hibernate](#)
- > [Spring JSF](#)
- > [Spring JDF Hibernate](#)

Miscellaneous

- > [Spring Data MongoDB](#)
- > [Spring Interview Questions](#)

RECOMMENDED TUTORIALS

Java Tutorials

- > [Java IO](#)
- > [Java Regular Expressions](#)
- > [Multithreading in Java](#)
- > [Java Logging](#)
- > [Java Annotations](#)
- > [Java XML](#)
- > [Collections in Java](#)
- > [Java Generics](#)
- > [Exception Handling in Java](#)
- > [Java Reflection](#)
- > [Java Design Patterns](#)
- > [JDBC Tutorial](#)

Java EE Tutorials

- > [Servlet JSP Tutorial](#)
- > [Struts2 Tutorial](#)
- > [Spring Tutorial](#)
- > [Hibernate Tutorial](#)
- > [Primefaces Tutorial](#)
- > [Apache Axis 2](#)
- > [JAX-RS](#)
- > [Memcached Tutorial](#)

Web Services Testing Tool

Learn Web Programming

Learn Web Development

Oracle Database Tools

Java Programming Courses

Learn Programming Online

Employer Interview Questions

Web Design Tutorials



