Rest APIs

Table of Contents

[What is REST ? 2](#_Toc80207792)

[HTTP Methods 2](#_Toc80207793)

[RESTFul Web Services 2](#_Toc80207794)

[Background 3](#_Toc80207795)

[Restful Methods 5](#_Toc80207796)

[Exercise 6](#_Toc80207797)

# What is REST?

* REST stands for **Re**presentational **S**tate **T**ransfer.
* REST is web standards based architecture and uses **HTTP Protocol** for data communication.
* It revolves around resource where every component is a resource and a resource is accessed by a common interface using HTTP standard methods.
* REST was first introduced by **Roy Fielding in 2000.**
* In REST architecture, a **REST Server** simply provides access to resources and **REST client** accesses and presents the resources.
* Here each resource is identified by **URIs**
* REST uses various **representations** to represent a resource like **text, JSON and XML**.
* Most popular light weight data exchange format used in web services is **JSON**

# HTTP Methods

Following well known HTTP methods are commonly used in REST based architecture.

1. GET - Provides a read only access to a resource.
2. POST - Used to create a new resource.
3. DELETE - Used to remove a resource.
4. PUT - Used to update an existing resource or create a new resource.

# RESTful Web Services

A web service is a collection of open protocols and standards used for exchanging data between applications or systems.

It is platform & technology independent solution.

Software applications written in various programming languages and running on various platforms can use web services to **exchange data** over computer networks like the Internet in a manner similar to inter-process communication on a single computer.

This interoperability (e.g., between Java and Python, or Windows and Linux applications or java & dotNet ) is due to the use of open standards.

Web services based on REST Architecture are known as RESTful web services.

These web services use HTTP methods to implement the concept of REST architecture. A RESTful web service usually defines a URI, Uniform Resource Identifier a service, provides resource representation such as JSON and set of HTTP Methods.

## Background

Web services have really come a long way since its inception. In 2002, the World Wide Web consortium (W3C) had released the definition of WSDL (web service definition language) and SOAP web services. This formed the standard of how web services are implemented.

In 2004, the web consortium also released the definition of an additional standard called RESTful. Over the past couple of years, this standard has become quite popular and is being used by many of the popular websites around the world which include Facebook and Twitter.

REST is a way to access resources which lie in a particular environment. For example, you could have a server that could be hosting important documents or pictures or videos. All of these are an example of resources. If a client, say a web browser needs any of these resources they have to send a request to the server to access these resources. Now REST defines a way on how these resources can be accessed.

Example of a web application can be an application which has a requirement to talk to other applications such Facebook, Twitter, and Google.

Now if a client application had to work with sites such as Facebook, Twitter, etc. they would probably have to know the language Facebook, Google and Twitter are built on, and also on what platform they are built on.

Based on this, we can write the interfacing code for our web application, but this could prove to be a nightmare.

So instead, Facebook, Twitter, and Google expose their functionality in the form of Restful web services. This allows any client application to call these web services via REST.

What is Restful Web Service?

REST is used to build Web services that are lightweight, maintainable, and scalable in nature. A service which is built on the REST architecture is called a RESTful service. The underlying protocol for REST is HTTP, which is the basic web protocol. REST stands for REpresentational State Transfer

The key elements of a RESTful implementation are as follows:

1. Resources The first key element is the resource itself. Let assume that a web application on a server has records of several employees. Let's assume the URL of the web application is http://www.server.com. Now in order to access an employee record resource via REST, one can issue the command http://www.server.com/employee/1 - This command tells the web server to please provide the details of the employee whose employee number is 1.

2. Request Verbs - These describe what you want to do with the resource. A browser issues a GET verb to instruct the endpoint it wants to get data. However, there are many other verbs available including things like POST, PUT, and DELETE. So in the case of the example http://www.server.com/employee/1 , the web browser is actually issuing a GET Verb because it wants to get the details of the employee record.

3. Request Headers These are additional instructions sent with the request. These might define the type of response required or the authorization details.

4. Request Body - Data is sent with the request. Data is normally sent in the request when a POST request is made to the REST web service. In a POST call, the client actually tells the web service that it wants to add a resource to the server. Hence, the request body would have the details of the resource which is required to be added to the server.

5. Response Body This is the main body of the response. So in our example, if we were to query the web server via the request http://www.server.com/employee/1 , the web server might return an XML document with all the details of the employee in the Response Body.

6. Response Status codes These codes are the general codes which are returned along with the response from the web server. An example is the code 200 which is normally returned if there is no error when returning a response to the client.

## Restful Methods

The below diagram shows mostly all the verbs (POST, GET, PUT, and DELETE) and an example of what they would mean.

Let's assume that we have a RESTful web service is defined at the location. http://www.server.com/employee . When the client makes any request to this web service, it can specify any of the normal HTTP verbs of GET, POST, DELETE and PUT. Below is what would happen If the respective verbs were sent by the client.

POST This would be used to create a new employee using the RESTful web service

GET - This would be used to get a list of all employee using the RESTful web service

PUT - This would be used to update all employee using the RESTful web service

DELETE - This would be used to delete all employee using the RESTful web service

# Exercise

|  |  |  |  |
| --- | --- | --- | --- |
| Action | Request Method | URL | What it does |
| Get all Students | Get | /students | Get all students data from server |
| Get 1 student | Get | /students/2 | Get 1 students data with id 2 from server |
| Add Student | Post | /students  Student data | Sever will store this data on server side |
| Update Student | Put | /students  Student data | Server updates the data |
| Delete Student | Delete | /students  studentId | Server will delete the student data |

Creating Rest API for book resources via Spring Boot

|  |  |  |  |
| --- | --- | --- | --- |
| URI | Give data | HTTP Methods | Functionality |
| localhost:9192/BookRestCrud/books/1 | - | GET | Get a single book |
| localhost:9192/BookRestCrud/books | - | GET | Get a list of books |
| localhost:9192/BookRestCrud/books | 1 books data | POST | Add a new book |
| localhost:9192/BookRestCrud/books/1 | Updated book data | PUT | Update a book |
| localhost:9192/BookRestCrud/books/1 |  | DELETE | Delete a book |