**RESTful** Web Services are basically REST Architecture based Web Services. In REST Architecture everything is a resource. RESTful web services are light weight, highly scalable and maintainable and are very commonly used to create APIs for web-based applications. This tutorial will teach you the basics of RESTful Web Services and contains chapters discussing all the basic components of RESTful Web Services with suitable examples.

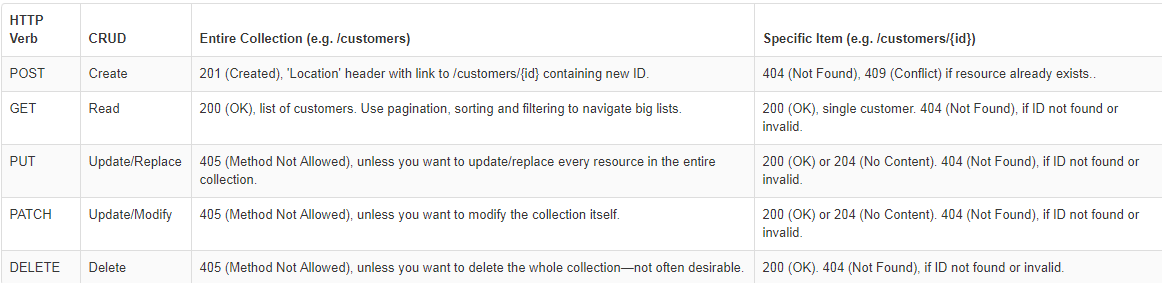
## **What is REST architecture?**

REST stands for **REpresentational State Transfer**. REST is web standards based architecture and uses **HTTP Protocol**. 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 modifies the resources. Here **each resource is identified by URIs**. REST uses various representation to represent a resource like text, JSON, XML. JSON is the most popular one.

### **HTTP methods**

Following four HTTP methods are commonly used in REST based architecture.



**Introduction to RESTFul web services**

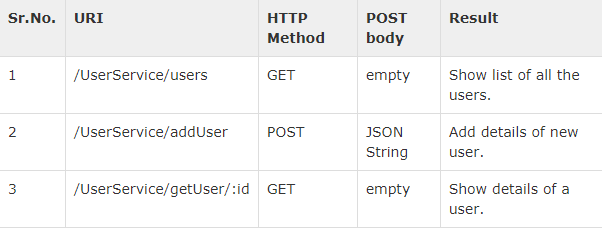
A web service is a collection of open protocols and standards used for exchanging data between applications or systems. Software applications written in various programming languages and running on various platforms can use web services to exchange data over computer networks like the Internet in a manner similar to inter-process communication on a single computer. This interoperability (e.g., between Java and Python, or Windows and Linux applications) is due to the use of open standards.

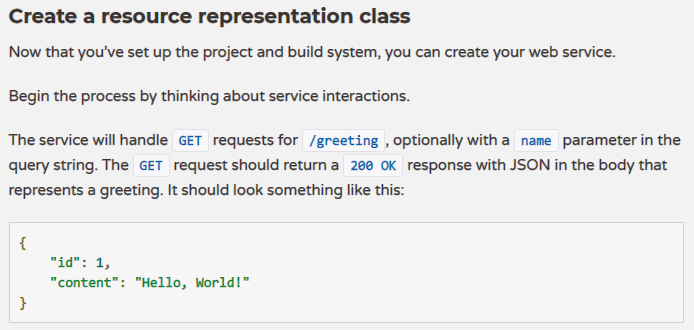
Web services based on REST Architecture are known as RESTful web services. These web services uses HTTP methods to implement the concept of REST architecture. A RESTful web service usually defines

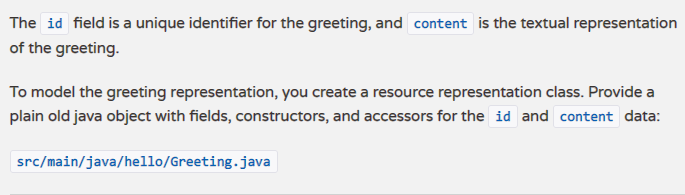
1. a URI, Uniform Resource Identifier
2. a service, provides resource representation such as JSON and
3. Set of HTTP Methods.

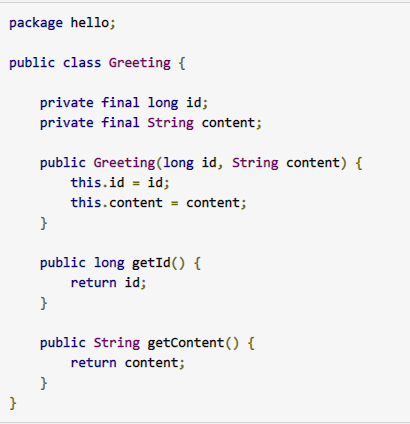
## **Creating Restful Web service**

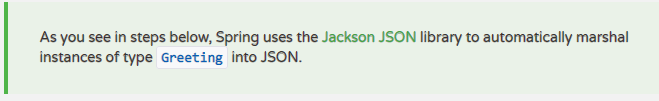
In next chapters, we'll create a web service say user management with following functionalities –

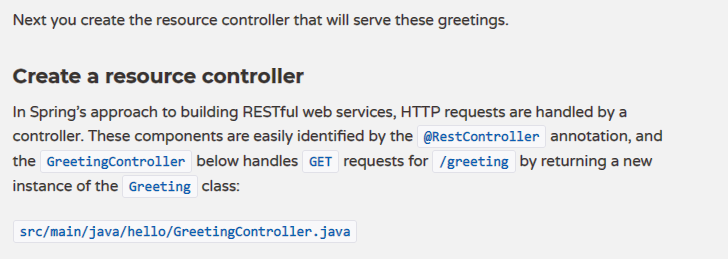


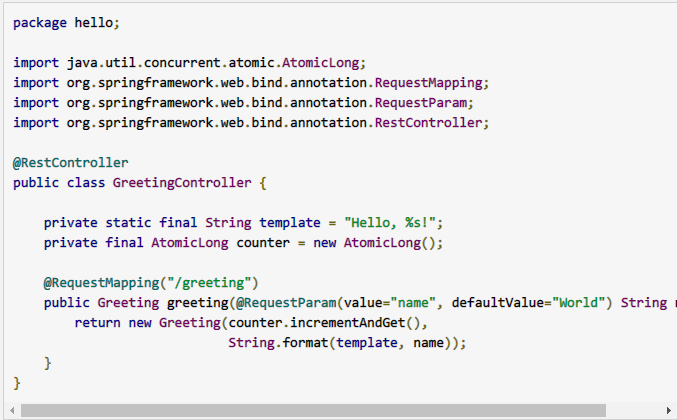


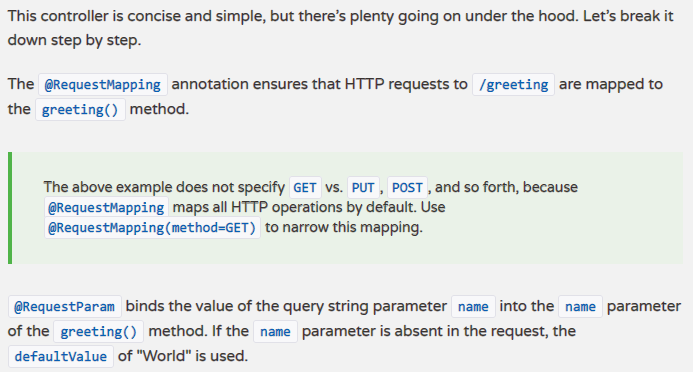


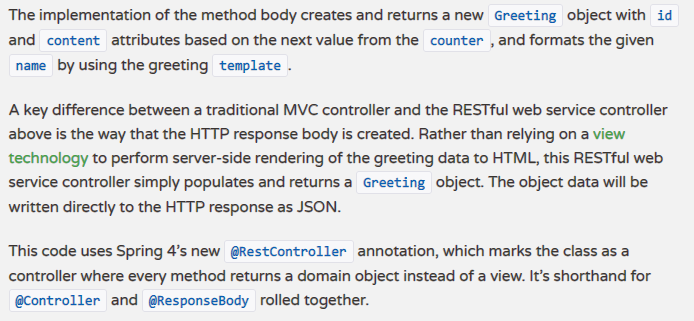


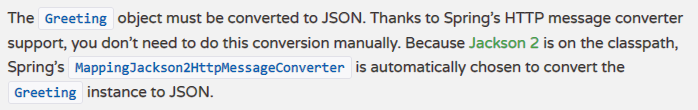




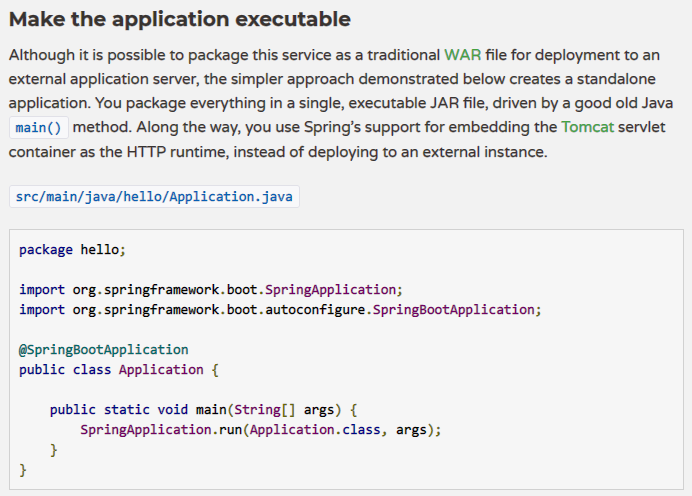


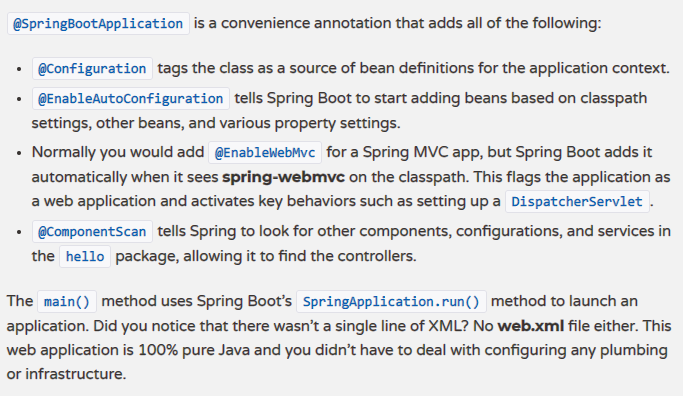


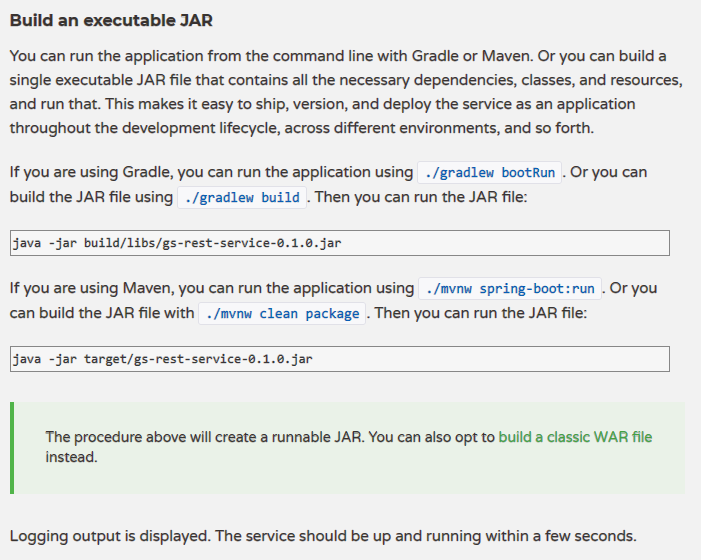


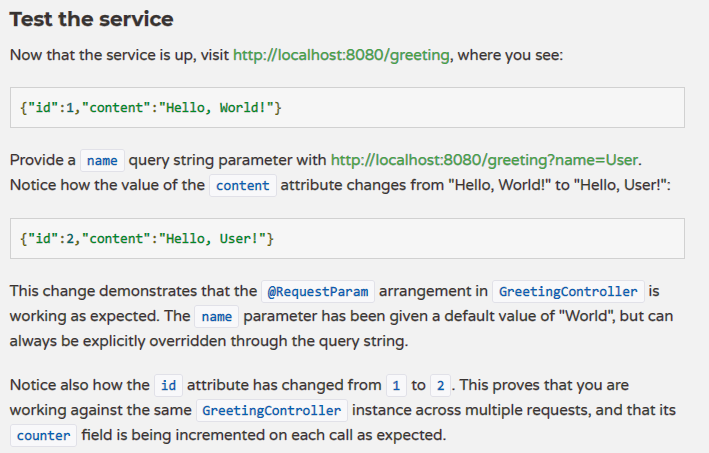


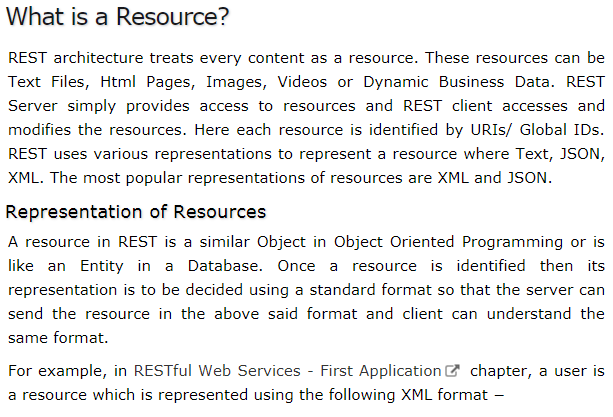
Otherwise you need to use @XmlRootElement(name = "user") and @XmlElement annotations from javax.xml.bind.annotation package.

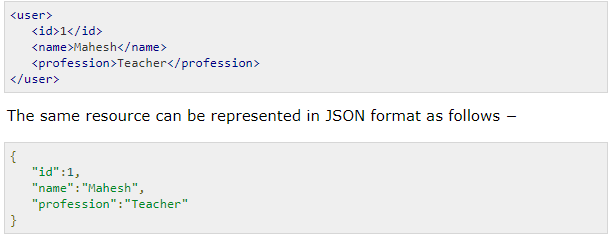












Note:



# RESTful Web Services – Messages

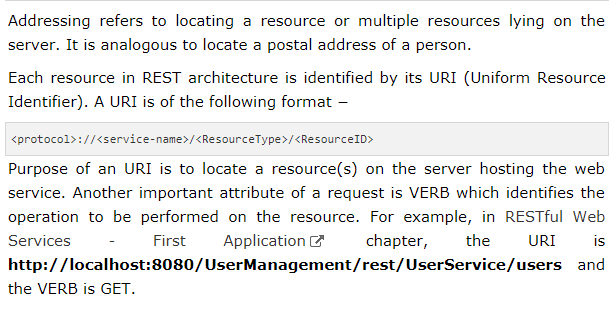
# 

# 

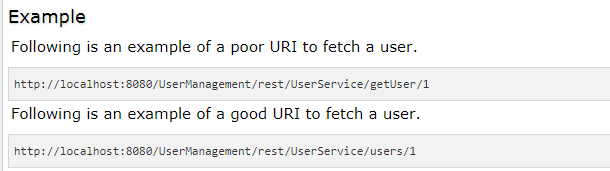
# 

# 

# RESTful Web Services - Addressing



Note: @Path annotation is for <ResourceType> and @RequestMapping annotation is for <ResourceID>



# Idempotent REST API:

# 

# 

# 

# 

# 

# 

# 

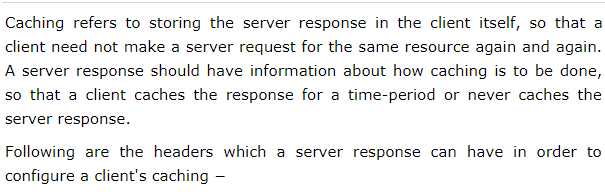
# 

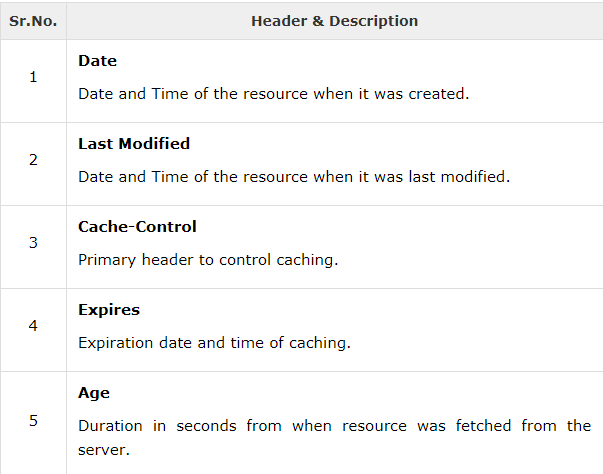
# RESTful Web Services - Statelessness

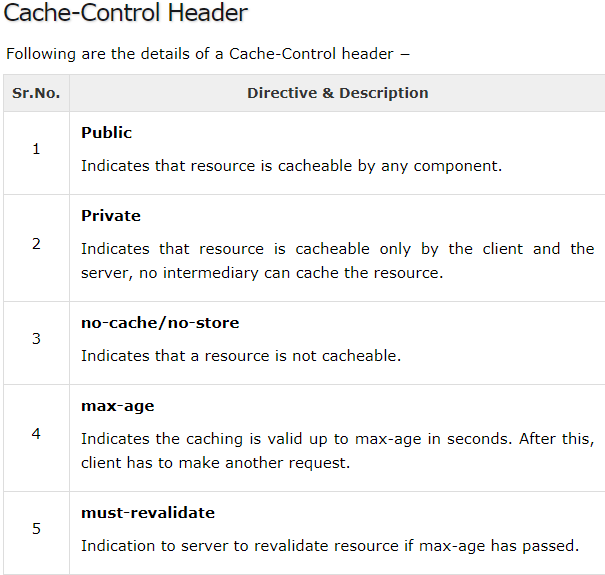
# 

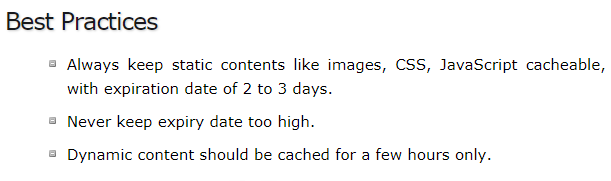
# 

RESTful Web Services – Caching

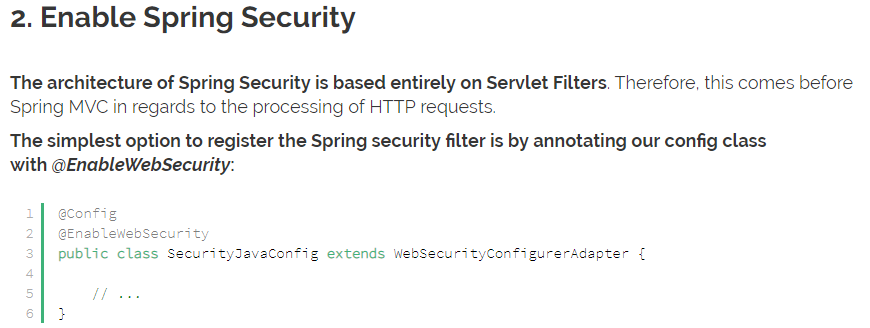


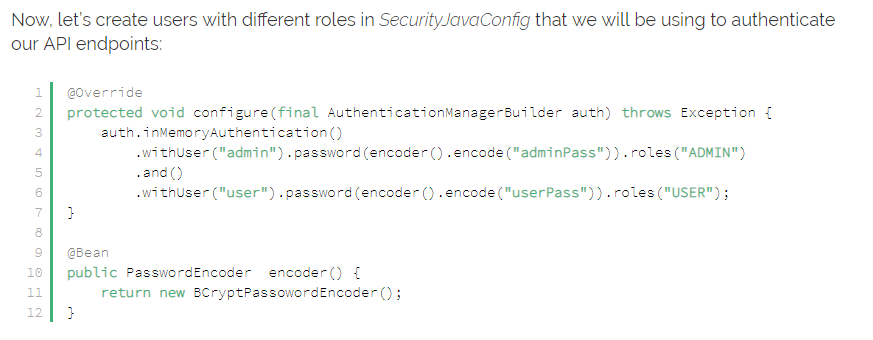


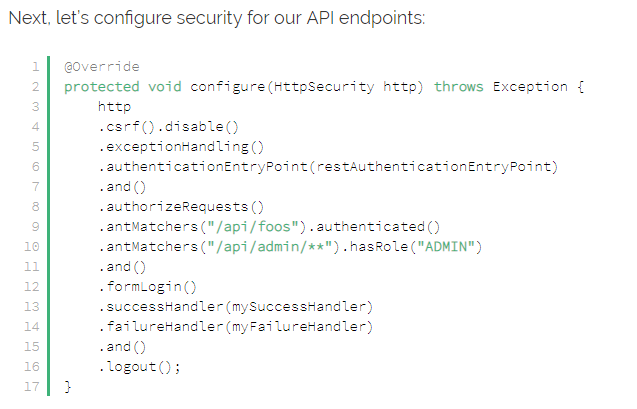




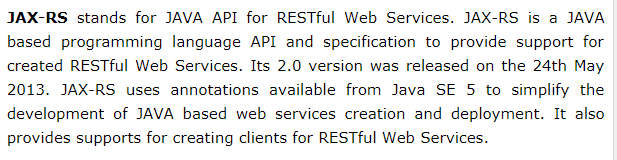
# RESTful Web Services - Security







# RESTful Web Services - Java (JAX-RS)



# 

