

# Práctica Uno

## Objetivo

Crear servicios REST que devuelvan texto plano, xml y json

## Notas

Introducir en el “pom.xml” del programa servidor, bajo la etiqueta “dependencies” esta dependencia

```
<dependency>  
  <groupId>org.glassfish.jersey.media</groupId>  
  <artifactId>jersey-media-json-binding</artifactId>  
  <version>2.27</version>  
</dependency>
```

y ésta otra para no tener que escribir código Java repetitivo

```
<dependency>  
  <groupId>org.projectlombok</groupId>  
  <artifactId>lombok</artifactId>  
  <version>1.16.20</version>  
  <scope>provided</scope>  
</dependency>
```

## Instrucciones

Crear una aplicación Web con Maven llamada “**PracticaUno**”

Crear una clase llamada “**Persona**” con un único atributo, **nombre**, de tipo **String**

El “ApplicationPath” que vamos a usar se llamará “rest”:

```
@ApplicationPath("rest")
```

1. Crear un servicio web (Restful Web Services from Patterns -> Simple Root Resource) con la siguientes características:

- a. path: textoplano
  - b. Class Name: TextoPlanoResource
  - c. MIME type: text/plain
  - d. Representation Class: java.lang.String
  - e. Adaptar el método "getText" para que devuelva una String
  - f. Colocar una traza en el método "putText" que nos muestre el texto enviado por el cliente
  - g. Ejecutar el programa
  - h. Abrir una ventana de comandos y probar que ambos métodos funcionan:
  - i. curl <http://localhost:8080/PracticaUno/rest/textoplano> (GET -> método "getText")
  - j. curl -X PUT -H "Content-Type: text/plain" -d "Segundo mensaje" <http://localhost:8080/PracticaUno/rest/textoplano> (PUT -> método "putText")
2. Crear un servicio web (Restful Web Services from Patterns -> Simple Root Resource) con la siguientes características:
- a. path: textojson
  - b. Class Name: TextoJsonResource
  - c. MIME type: application/json
  - d. Representation Class: Persona (hay que indicar el camino de paquete, por ejemplo com.curso.practicauno.Persona)
  - e. Adaptar el método "getJson" para que devuelva una instancia de la clase Persona
  - f. Colocar una traza en el método "putJson" que nos muestre la Persona enviada por el cliente
  - g. Ejecutar el programa

- h. Abrir una ventana de comandos y probar que ambos métodos funcionan:
  - i. curl <http://localhost:8080/PracticaUno/rest/textojson> (GET -> método "getJson")
  - j. curl -X PUT -H "Content-Type: application/json" -d {"nombre":"abc"} <http://localhost:8080/PracticaUno/rest/textojson> (PUT -> método "putJson")
3. Crear un servicio web (Restful Web Services from Patterns -> Simple Root Resource) con la siguientes características:
- a. path: textoxml
  - b. Class Name: TextoXmlResource
  - c. MIME type: application/xml
  - d. Representation Class: Persona (hay que indicar el camino de paquete, por ejemplo com.curso.practicauno.Persona)
  - e. Adaptar el método "getXml" para que devuelva una instancia de la clase Persona
  - f. Colocar una traza en el método "putXml" que nos muestre la Persona enviada por el cliente
  - g. Ejecutar el programa
  - h. Abrir una ventana de comandos y probar que ambos métodos funcionan:
  - i. curl <http://localhost:8080/PracticaUno/rest/textoxml> (GET -> método "getXml")
  - j. curl -X PUT -H "Content-Type: application/xml" -d "<?xml version='1.0' encoding='UTF-8' standalone='yes'><persona><nombre>Persona

XML</nombre></persona>”

<http://localhost:8080/PracticaUno/rest/textoxml> (PUT -> método  
“putJson”)