

Web Service con Java

IEEE INTERNATIONAL SMARTCITIES WORKSHOP





Bienvenidos

José Luis Cobián Hermosillo
@cobymotion

github.com/cobymotion

Objetivo del taller

- Generar un servicio web para interactuar con una base de datos, usando la plataforma de java con uso de las tecnológicas JDBC, RESTFul, JAX-RS y EJB codificadas en formato JSON que puedan ser consumidas por diferentes clientes.

Tematica

- Crear una estructura sencilla de base de datos (1 entidad)
- Crear una aplicación Java estándar que permita controlar los aspectos de la conexión. (JDBC)
- Crear una aplicación web que permita hacer la implementación de la aplicación en un entorno web
- Crear sus RESTFul con formatos JSON para dos tareas, Lectura y escritura
- Probar con algún cliente REST comercial
- Probar una aplicación Java estándar (Java FX) para poder consumirlos

JDBC

- Java DataBase Connectivity es la API que permite ejecutar operaciones con la base de datos.
- JDBC lo utilizaremos para conectar con nuestra base de datos en Mysql.

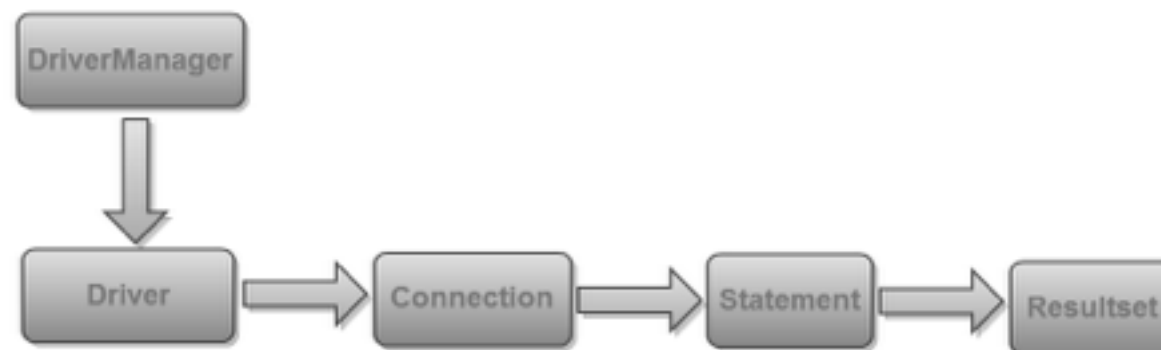
Para conectar necesitaremos los siguientes datos:

Usuario

Pass:

Url de la ubicación del servidor

JDBC Class Usage



Problema con JDBC

- Es lento y tiene problema con la concurrencia
- Una solución (No la mejor), es aplicarlo en un Singleton
- Singleton, es un patrón de diseño que consiste en evitar que se creen mas de un instancia del mismo objeto

Entidad

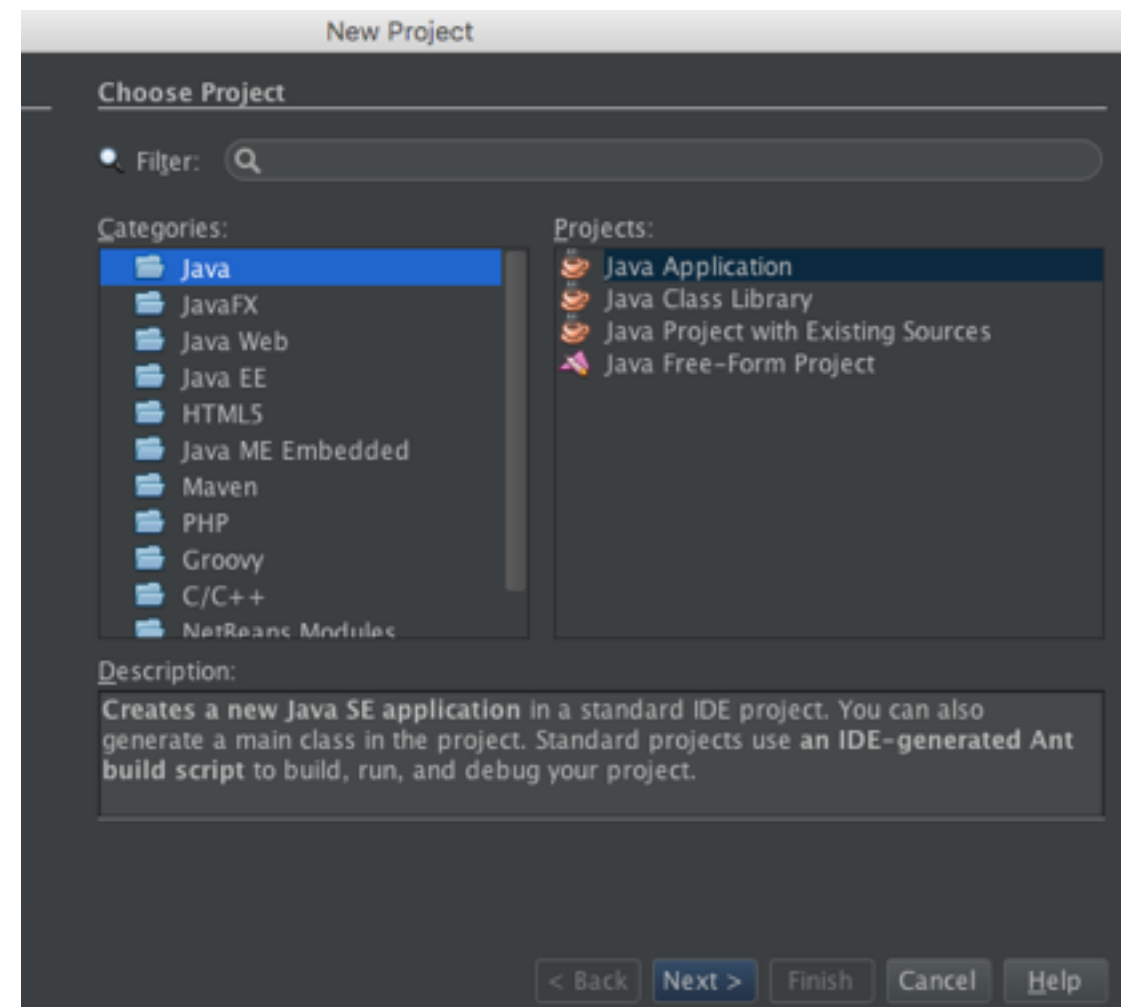
```
CREATE TABLE `lugares` (  
  `idlugares` int(11) NOT NULL  
  AUTO_INCREMENT,  
  `nombre` varchar(45) DEFAULT NULL,  
  `altitud` double DEFAULT NULL,  
  `longitud` double DEFAULT NULL,  
  PRIMARY KEY (`idlugares`)  
) ENGINE=InnoDB AUTO_INCREMENT=4 DEFAULT  
CHARSET=latin1;
```

lugares	
🔑	idlugares INT(11)
◇	nombre VARCHAR(45)
◇	altitud DOUBLE
◇	longitud DOUBLE
Indexes ▶	



Proyecto

- Crearemos un proyecto que sea JAVA Application, para esto usaremos NETBEANS
- ¿Porque Netbeans?



¿Que hacer ?

- Generar conexión con la base de datos
- Generar un archivo que tenga dos procesos que son agregar un lugar y recuperar los lugares

```
1 #Datos de la conexion
2 usr=root
3 pwd=
4 driver= com.mysql.jdbc.Driver
5 url= jdbc:mysql://localhost:3306/test_ieee
```



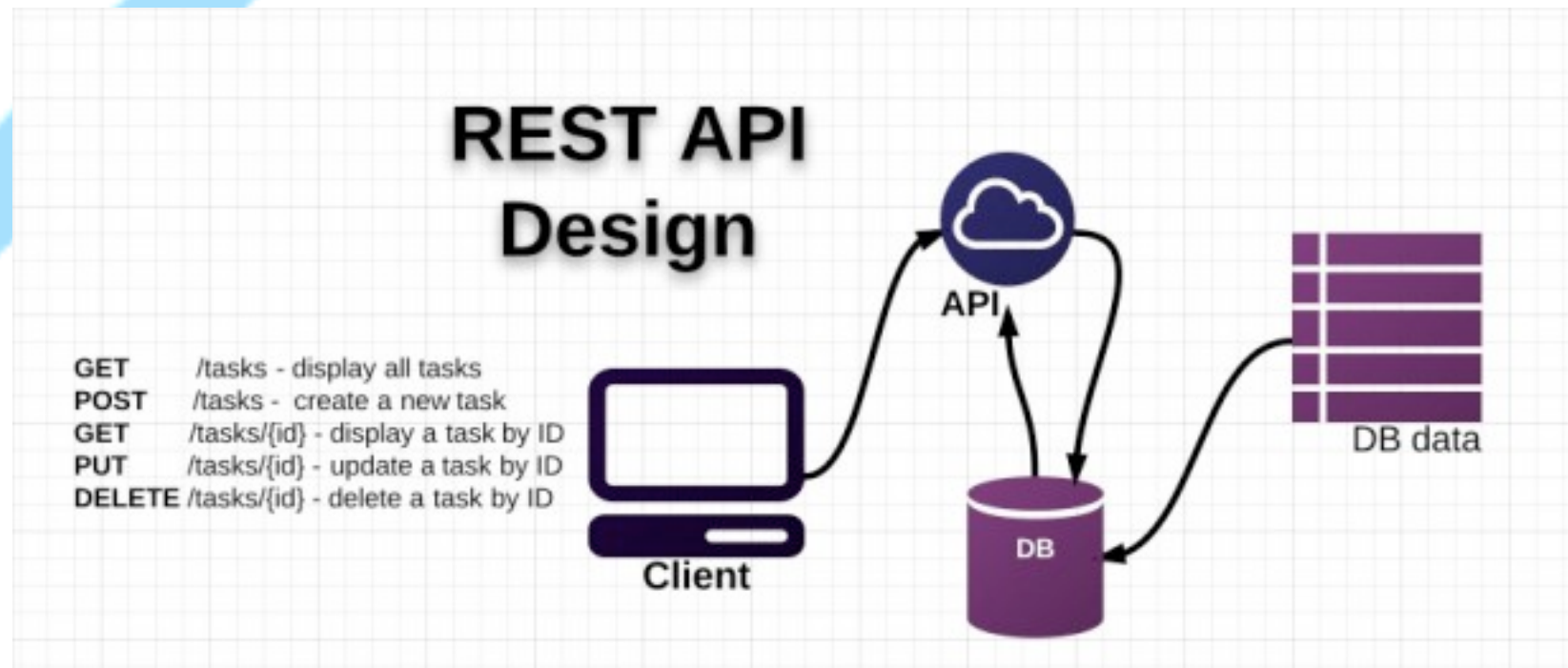
Probar la aplicación

JAVA WEB

- Para te la plataforma EE, tiene como principal característica que tiene clases referentes a los request dentro de un entorno web

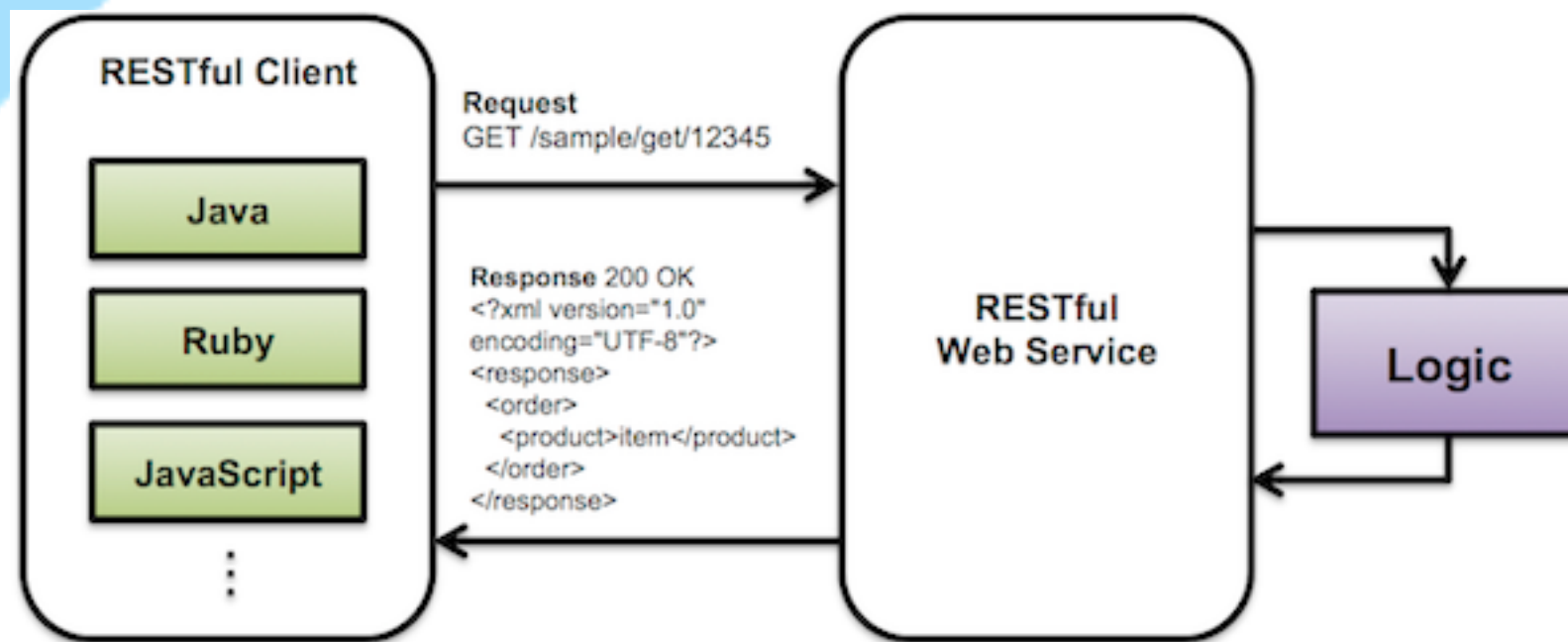
RESTFuI

- Representational State Transfer es una aplicación de servicios en la web simple que permite por medio de anotaciones evitar errores con los archivos de descripción.



JAX-RS

- Es un API que permite el trabajo con los RESTFul basado en anotaciones de Java
- Permite “**mapear**” los objetos mas sencillo.



Proyecto Web

- Importar la librería del proyecto anterior
- Crear una RestFul que permita invocar a los métodos creados en el anterior proyecto y ponerlos en formato de JSON

Probar la aplicación Client RestFul (Firefox)

Proyecto cliente

- Abriremos el proyecto de Java FX y probaremos que nuestro Servicio Web que trabaje correctamente.

¿Para considerar?

- Implementación de un RestFul
- Mapeo de los datos
- Conectividad y servicios



Consumirlo desde android

Dia 2
Android + RestFul

HttpURLConnection

- Al ser un RESTFul un servicio que acepta una serie de petición web, mediante sus encabezados, existen métodos muy simples para su uso.
- En la actualidad la sincronización es uno de los puntos a considerar por lo que la conexión puede generar que nuestra aplicación deje de responder.
- Hay algunos frameworks que permiten cuidar la seguridad de nuestros servicios:
 - Retrofit
 - Volley
- Para este ejemplo lo haremos genérico con HttpURLConnection que ya la trae Android.

Envio datos

- Para nuestro ejemplo en particular

```
URLConnection conn = (URLConnection) url.openConnection();  
conn.setDoOutput(true);  
conn.setRequestMethod("POST");  
conn.setRequestProperty("Content-Type", "application/json");
```

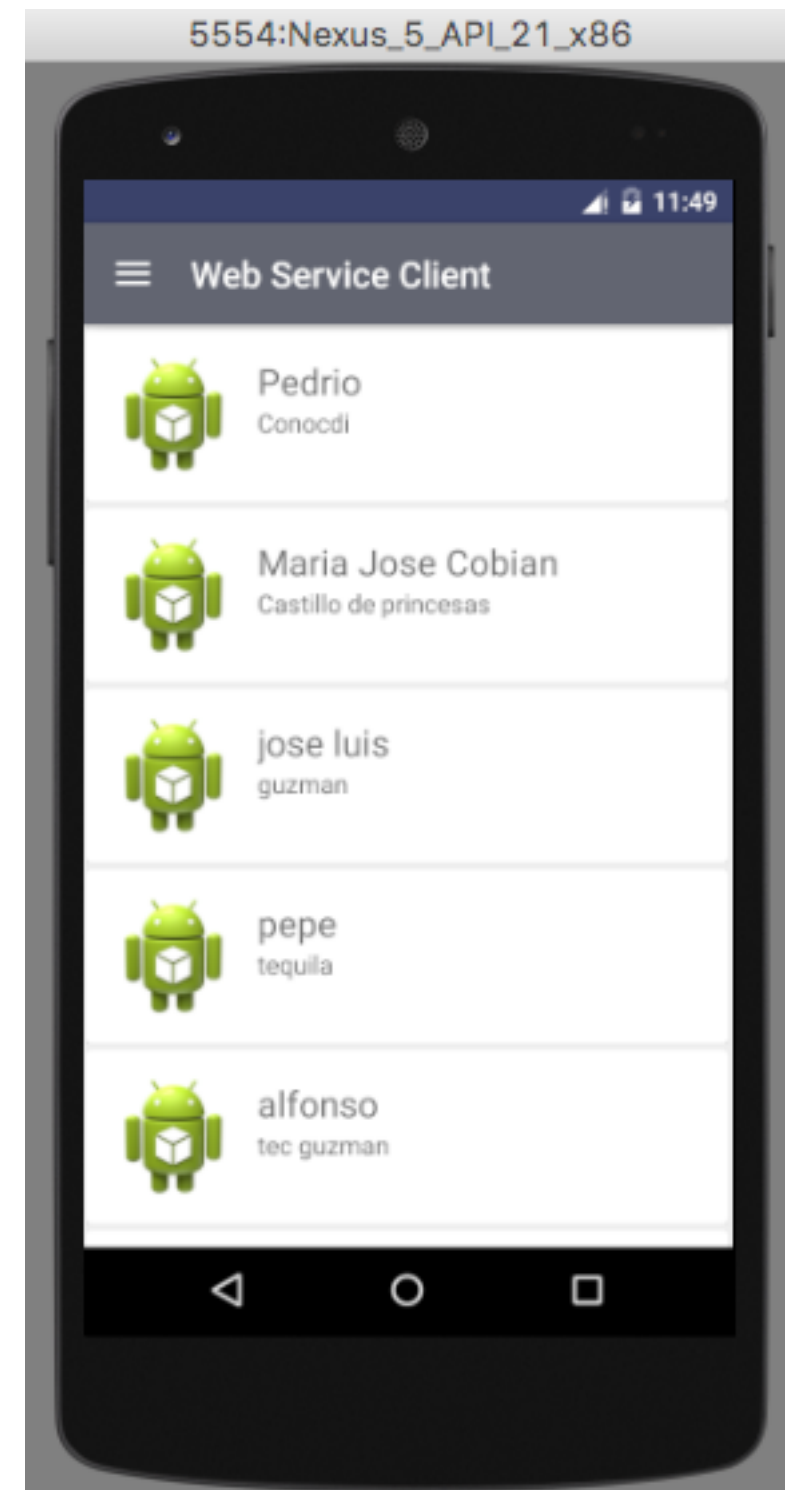
— Envio

```
OutputStream os = conn.getOutputStream();  
os.write(input.getBytes());  
os.flush();
```

Leer datos

```
URLConnection con =  
    (URLConnection) url.openConnection();  
BufferedReader rd = new BufferedReader  
    (new InputStreamReader(con.getInputStream()));  
String line = "";  
String resultado = "";  
while ((line = rd.readLine()) != null) {  
    resultado+=line;  
}
```

Aplicarlo a la aplicación





Gracias por la atención
@cobymotion