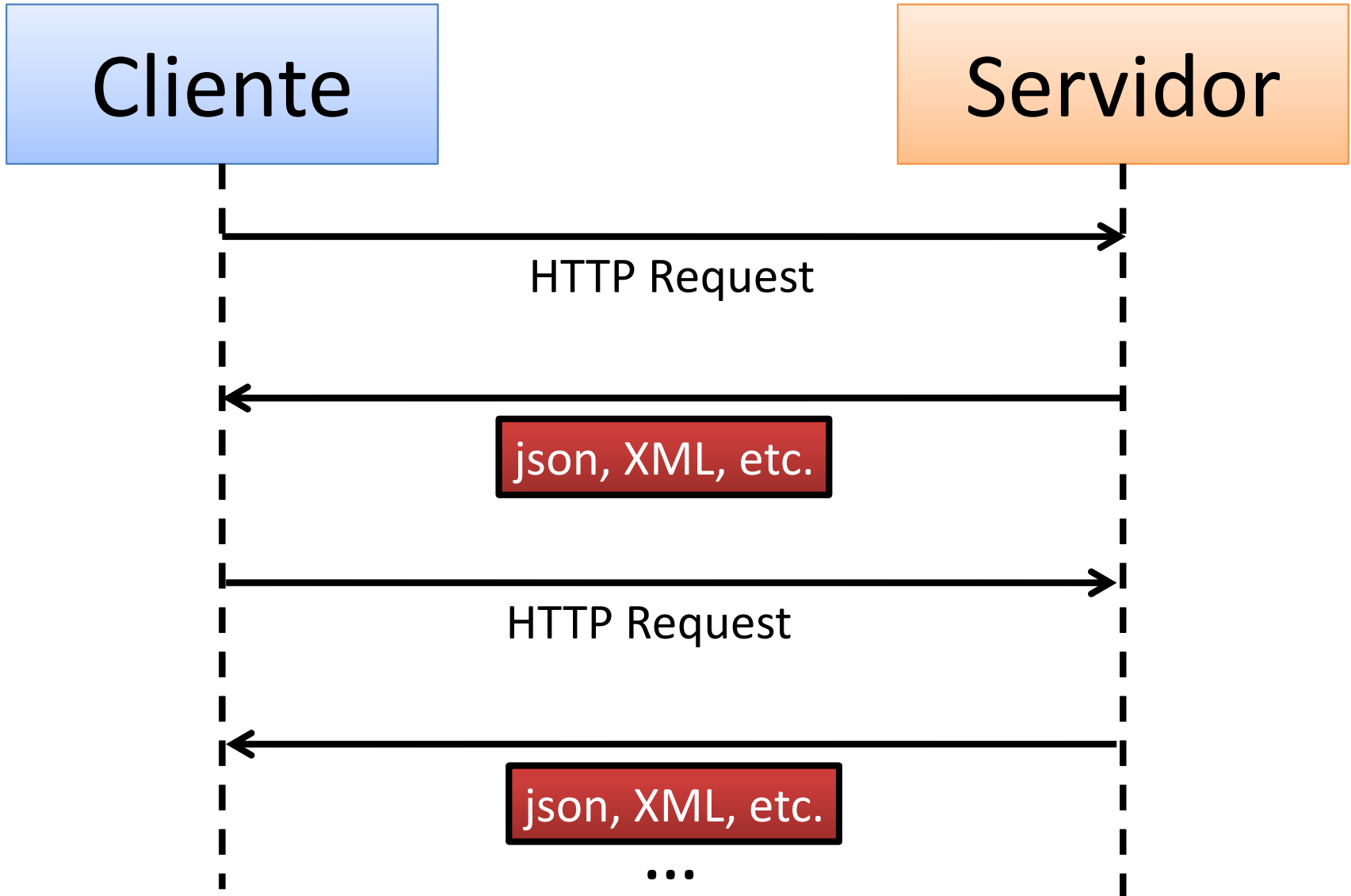


REST

# REST



# REST

- Protocolo para intercambiar información entre cliente y servidor web
- Utiliza HTTP
- Petición HTTP: llamado a un método

# Ejercicio

- Abrir las siguientes URL en el navegador:

<http://jsonplaceholder.typicode.com/albums>

<http://jsonplaceholder.typicode.com/albums/10>

<http://jsonplaceholder.typicode.com/albums/?userId=2>

# Ejercicio 2: Usar REST sin navegador

- Desde una línea de comandos
  - Ejecutar

```
telnet jsonplaceholder.typicode.com 80
```

- Copiar y pegar lo siguiente:

```
GET /albums/1 HTTP/1.1
```

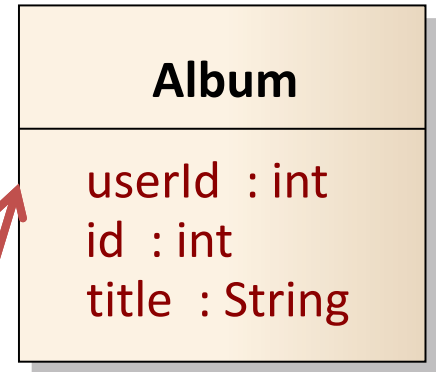
```
Host: jsonplaceholder.typicode.com
```

- Oprimir dos veces **Enter**

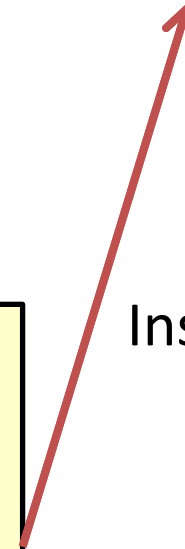
# JavaScript Object Notation (JSON)

- Subconjunto de Javascript
- Representa objetos
- “Fácil” de leer

```
{  
  "userId": 2,  
  "id": 12,  
  "title": "consequatur autem  
    doloribus natus consectetur"  
}
```



Instancia de



# Peticiones REST en Angular

- GET
- POST
- PUT
- DELETE

```
git clone https://bitbucket.org/jpavlich/pw.git
```

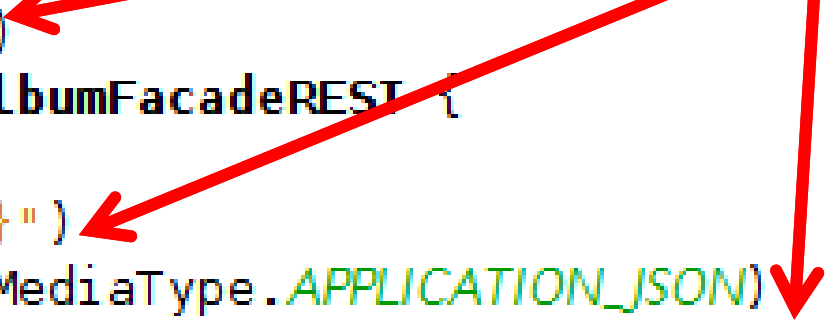
<https://angular.io/guide/http>

# Creación de servicios REST en Java EE



# Servicios REST en Java

- El servicio se llama usando la siguiente URL:  
`http://servidor.com/albums/1`



```
@Path("albums")
public class AlbumFacadeREST {
    @GET
    @Path("{id}")
    @Produces(MediaType.APPLICATION_JSON)
    public Album find(@PathParam("id") Long id) {

        // returns the album from the database
    }
}
```

# Servicios REST en Spring

```
git clone https://bitbucket.org/jpavlich/pw.git
```

<https://spring.io/guides/gs/rest-service/>

# Ejercicio

- Crear un servicio REST que recupere una lista de entidades del proyecto
  - Con persistencia (usando JPA)
- Crear un cliente en Angular que recupere dicha información y la muestre en una tabla