

Introducción a los Servicios REST

DESARROLLO DE SERVICIOS REST JERSEY Y CLIENTES JQUERY



Guadalupe Ortiz Bellot
Departamento de Ingeniería Informática

INTRODUCCIÓN A LOS SERVICIOS REST

Contenido

- Introducción
- Conceptos básicos de REST
- Formatos de transferencia de datos
- Anexo 1. Libros y referencias de consulta

INTRODUCCIÓN A LOS SERVICIOS REST

Contenido

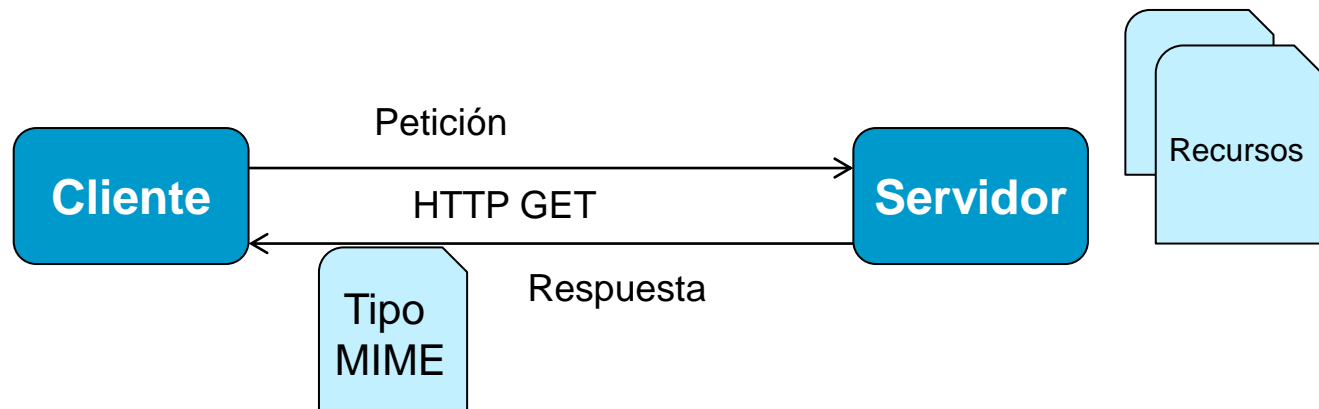
- **Introducción**
 - REST a vista de pájaro
 - REST versus SOA
- Conceptos básicos de REST
- Formatos de transferencia de datos
- Anexo 1. Libros y referencias de consulta

INTRODUCCIÓN

REST a vista de pájaro

REST: Transferencia de estado representacional.

1. Un usuario hace una petición (por ejemplo GET) a una dirección html de una aplicación, por ejemplo a través del navegador.
2. El navegador envía una solicitud HTTP al servidor.
3. El servidor responde con un documento HTML con un tipo MIME.



INTRODUCCIÓN

REST versus SOAP (i)

- Los servicios construidos con el estilo de la arquitectura REST (denominado servicios RESTful) **encapsulan los datos en un formato XML simple** y los transportan a través de HTTP como una petición de una página web a un servidor web.
- Los servicios web RESTful son particularmente útiles cuando sólo es necesario **transmitir y recibir mensajes simples**.
- SOAP se utiliza sobre todo para las aplicaciones empresariales para integrar **tipos y aplicaciones más complejos**, así como sistemas legacy.

INTRODUCCIÓN

REST versus SOAP (ii)

- REST:
 - Ligerero
 - Legible por un humano
 - Fácil de construir



- SOAP
 - Fácil de consumir - a veces
 - Fuerte tipado - comprobación de tipos
 - Herramientas de desarrollo
 - Más seguridad



INTRODUCCIÓN A LOS SERVICIOS REST

Contenido

- Introducción
- **Conceptos básicos de REST**
 - Principios
 - JAX-RS
 - Ejemplo
- Formatos de transferencia de datos
- Familiarízate con las etiquetas de Rest y Pruébalas
- Anexo 1. Libros y referencias de consulta

CONCEPTOS BÁSICOS DE REST

Principios (i)

REST es un estilo arquitectónico para servicios que utiliza estándares web. Sus principios fundamentales son:

- Todo puede ser identificado como un recurso y cada recurso puede ser identificada por un URI.
- Un recurso puede ser representado en múltiples formatos, definido por un *media type*.
- Se usan métodos HTTP estándar para interactuar con el recurso : principalmente GET, POST, PUT y DELETE.
- La comunicación entre el cliente y el endpoint es sin estado.

CONCEPTOS BÁSICOS DE REST

Principios (ii)

- La World Wide Web es un ejemplo clásico de la arquitectura de estilo REST: los URI identifican los recursos y HTTP es el protocolo por el cual se accede a estos.
- HTTP proporciona una interfaz uniforme y un conjunto de métodos para manipular el recurso
- Un programa cliente, como un navegador web, puede acceder, actualizar, agregar o quitar un recurso web a través del URI utilizando diversos métodos HTTP.

CONCEPTOS BÁSICOS DE REST

JAX-RS (i)

- API estándar basada en anotaciones para crear un servicio web RESTful en Java y un cliente para su invocación.
- La especificación JAX -RS cumple los siguientes objetivos:
 - **Centrada en POJO:** La API JAX-RS proporciona un conjunto anotaciones y clases/interfaces asociados que se pueden utilizar en los POJOs con el fin de exponerlos como recursos RESTful .
 - **Centrada en HTTP:** Dado que los recursos REST están expuestos a través de HTTP, la especificación proporciona un mapeado claro entre el protocolo HTTP y las clases y métodos correspondientes de la API JAX-RS.

CONCEPTOS BÁSICOS DE REST

JAX-RS (ii)

- Mediante el uso de esta API: se puede marcar un POJO a través de anotaciones que permiten identificar:
 - un recurso como una URI
 - un conjunto de métodos bien definidos para acceder a los recursos (GET, POST, etc)
 - múltiples formatos de representación de un recurso

```
@GET
@Path("/saludo")
@Produces(MediaType.TEXT_PLAIN)
public String hola( ) { //... }
```

CONCEPTOS BÁSICOS DE REST

JAX-RS (iii)

- En tiempo de ejecución, el entorno que implementa la especificación JAX - RS es responsable de la invocación de la aplicación Java correspondiente mediante el **mapeo de la solicitud HTTP** con uno de los **métodos Java** que satisface la petición.
- Hay que determinar la **clase y métodos** Java que representan el recurso, el **tipo de contenido** y el **método HTTP** de la invocación.

`http://applicationName/saludo`

```
GET
Texto Plano
public String hola( )
```

CONCEPTOS BÁSICOS DE REST

JAX-RS (iv)

- Ofrece soporte para el uso de los métodos estándar HTTP GET, POST, PUT, DELETE, HEAD y OPTIONS
 - **GET**: Recuperar un recurso
 - **POST**: Crear un recurso
 - **PUT**: Actualizar un recurso
 - **DELETE**: Borrar un recurso
 - **HEAD**: Como GET, pero no devuelve el cuerpo. Se utiliza para obtener metainformación acerca del recurso. Si no hay método marcado con HEAD, se hace un GET y se descarta el cuerpo.
 - **OPTIONS**: Pide a las opciones de comunicación disponibles. Si no hay ningún método marcado con @OPTIONS, se genera una respuesta automática.

CONCEPTOS BÁSICOS DE REST

Ejemplo (i)

```
@Path("orders")
```

```
public class OrderResource {
```

```
@GET
```

```
public List<Order> getAll() {...}
```

```
@GET
```

```
@Path("{oid}")
```

```
public Order getOrder(@PathParam("oid") int id) {...}
```

```
@DELETE
```

```
@Path("{id}")
```

```
public void deleteXml(@PathParam("id") int id) {...}
```

```
@XmlRootElement
```

```
public class Order {...}
```

<http://localhost:8080/store/webresources/orders>
<http://localhost:8080/store/webresources/orders/1>

CONCEPTOS BÁSICOS DE REST

Ejemplo (ii)

```
@GET
```

```
@Path("{oid}")
```

```
@Produces({"application/xml", "application/json"})
```

```
public Order getOrder(@PathParam("oid") int id) { ...}
```

```
@POST
```

```
@Path("create")
```

```
@Consumes("application/x-www-form-urlencoded")
```

```
public Order createOrder(@FormParam("id") int id,  
                        @FormParam("name") String name) { ...}
```

```
@PUT
```

```
@Path("{id}")
```

```
@Consumes("*/xml")
```

```
public Order putXml(@PathParam("id") int id, String  
content) { ...}
```

INTRODUCCIÓN A LOS SERVICIOS REST

Contenido

- Introducción
- Conceptos básicos de REST
- **Formatos de transferencia de datos**
 - **Conceptos básicos**
 - **Especificación en el servicio**
 - **Especificación en el cliente**
- Anexo 1. Libros y referencias de consulta

FORMATOS DE TRANSFERENCIA DE DATOS

Conceptos Básicos

- El cliente consulta o actualiza los recursos a través de la URI mediante el intercambio de representaciones del recurso.
- Dichas representaciones contienen información en formatos tales como HTML, XML o JavaScript Object Notation (JSON).
- El cliente debe conocer el tipo devuelto por el servicio.
- Por lo general, el cliente especifica la representación que quiere (Accept), y el servidor devuelve los recursos deseados en ese formato.
- Toda la información necesaria para procesar una petición de un recurso está contenida dentro de la propia solicitud, con lo que la interacción es sin estado.

FORMATOS DE TRANSFERENCIA DE DATOS

Especificación en el servicio

- De forma predeterminada, un recurso REST se publica o se consume con el `* / *` tipo MIME.
- Un recurso REST puede restringir los tipos de medios admitidos por la petición y la respuesta con las anotaciones `@Consumes` and `@Produces`, respectivamente.
- Estas anotaciones se pueden especificar en la clase o en los métodos. Si la anotación se especifica en el método anula la de la clase.

FORMATOS DE TRANSFERENCIA DE DATOS

Especificación en el cliente

- Content-Type: indica el tipo enviado (por ejemplo "text/plain", "text/xml", "text/html", "application/json", "application/atom+xml")
- Accept: indica el tipo de recurso que se espera recibir
- Con jquery:
 - contentType (envío)
 - dataType (espero recibir)

Anexo 1

Libros y referencias de consulta

Java EE 7 Essentials

By: Arun Gupta

Publisher: O'Reilly Media, Inc.

Pub. Date: August 23, 2013

Print ISBN-13: 978-1-4493-7017-6

Apache CXF Web Service Development

By: Naveen Balani ; Rajeev Hathi

Publisher: Packt Publishing

Pub. Date: December 17, 2009

Print ISBN-13: 978-1-847195-40-1

RESTful Java Web Services

By: Jose Sandoval

Publisher: Packt Publishing

Pub. Date: November 11, 2009

Print ISBN-13: 978-1-847196-46-0

Developing RESTful Services with JAX-RS 2.0, WebSockets, and JSON

By: Masoud Kalali; Bhakti Mehta

Publisher: Packt Publishing

Pub. Date: October 15, 2013

Print ISBN-13: 978-1-78217-812-5

REST with Java (JAX-RS) using Jersey - Tutorial

Lars Vogel

<http://www.vogella.com/articles/REST/article.html>

Creación de un servicio Web REST y su despliegue en Tomcat

Montes Cumbreira, Javier; Carmona Román, Salvador

<http://rodin.uca.es/xmlui/handle/10498/17553>

Tutorial de implementación y despliegue de un servicio web REST

Herrera Vaca, Alberto

<http://rodin.uca.es/xmlui/handle/10498/17551>