

# Capítulo 4

## JAX-RS – Java API for RESTful Web Services

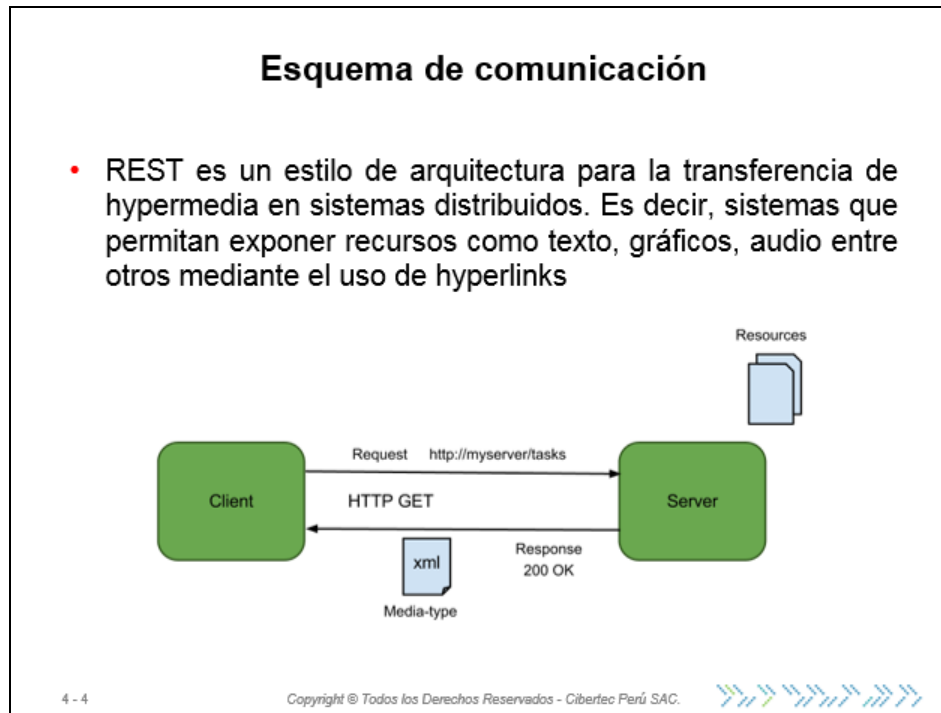
**Al finalizar el capítulo, el alumno podrá**

- Conocer las principales características de la especificación JAX-RS para el desarrollo de Web Services.
- Implementar aplicaciones que utilicen JAX-RS para la publicación y consumo de Web Services

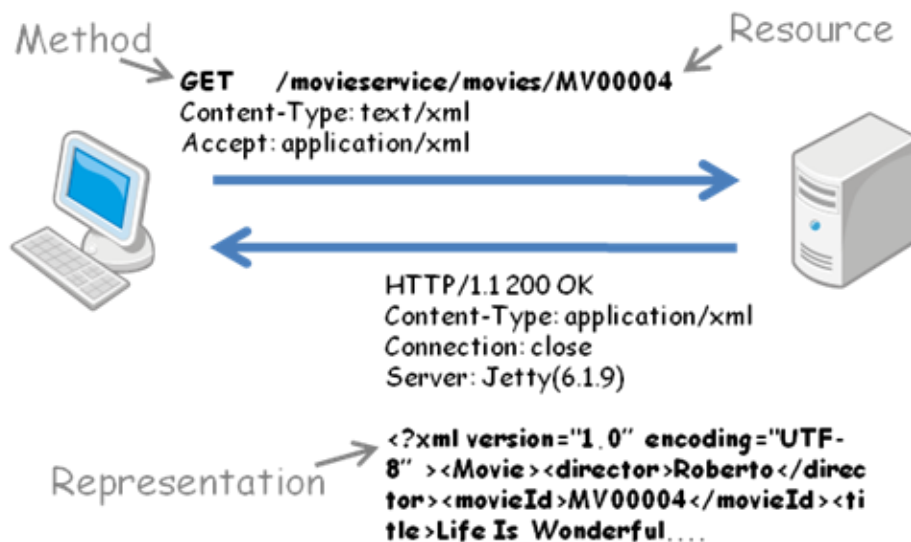
### **Temas**

1. Esquema de comunicación
2. Principios de arquitectura REST
3. JSR 311 – Java API for RESTful web services
4. JAX-RS Jersey Reference Implementation
5. JSR 339 – JAX-RS 2.0

## 1. Esquema de comunicación



REST es un estilo de arquitectura de software para la transferencia de hypermedia en sistemas distribuidos. Es decir, sistemas que permitan exponer recursos como texto, gráficos, audio entre otros mediante el uso de hyperlinks.



Tal como se explicó en capítulos previos, REST hace uso de los siguientes componentes dentro de su esquema de comunicación:

- HTTP (HyperText Transfer Protocol)
- Methods
- Resources
- Representations

## 2. Principios de arquitectura REST

### Arquitectura REST

- REST utiliza URL como un identificador uniforme o dirección para acceder a la representación de un recurso
  - <http://www.iana.org/domains/reserved>
  - <http://www.example.com/bugs/by-state/open>
- REST se basa en los MIME Types para describir el formato de los recursos en internet
  - text/html
  - text/xml
  - application/xml
  - application/json



4 - 8
Copyright © Todos los Derechos Reservados - Cibertec Perú SAC.


### 2.1. Identificación de recursos

La unidad mínima de abstracción para los datos en REST es el llamado “recurso”, cada recurso debe ser identificado y ubicado mediante una URI (Uniform Resource Identifier).

La URI ha sido una tecnología fundamental para el desarrollo de la Web ya que permite identificar la representación de recursos de una manera descriptiva.

No existen reglas absolutas para la definición de URIs pero en general deben tener una estructura clara y describir su propósito.

A continuación se muestran algunos ejemplos de URIs recomendadas por la W3C:

- <http://www.example.com/software/releases/1.0.3.tar.gz>
- <http://www.example.com/software/releases/latest.tar.gz>
- <http://www.example.com/weblog/2006/10/24/0>
- <http://www.example.com/sales/2004/Q4>
- <http://www.example.com/bugs/by-state/open>

## 2.2. Representación de los recursos

La interacción con los servicios se realiza mediante representaciones de recursos. Es decir, un recurso es referenciado por una URI y puede tener múltiples formatos. Diferentes plataformas necesitan diferentes formatos. Por ejemplo, los navegadores necesitan HTML, Javascript necesita JSON y Java puede necesitar XML o algún otro tipo de formato.

La forma de acceder a un recurso en la Web es a través de representaciones del mismo que pueden ser especificados mediante los MIME Types.

El formato MIME (Multi-purpose Internet Mail Extension) se compone principalmente de dos partes y contempla atributos opcionales.

**tipo/subtipo; atributo=valor; atributo=valor**

El “**tipo**” pertenece a la familia del formato y el “**subtipo**” es una categoría de la respectiva familia. Opcionalmente, pueden contener una serie de atributos definidos por nombre/valor y delimitados por el carácter (;).

Ejemplos

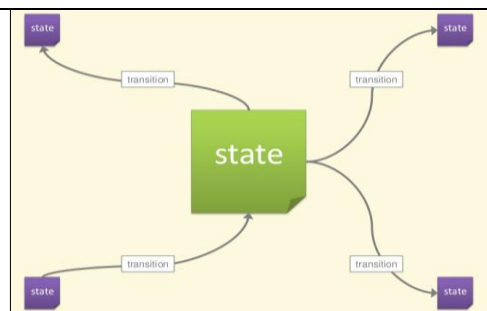
- text/plain
- text/html
- application/xml
- text/html; charset=iso8859-1
- application/vnd+xml; versión=1.1

## 2.3. HATEOAS (Hypermedia as a Engine of Application State)

El estado actual de una aplicación web debería ser capturada en uno o más documentos de hipertexto, y residir tanto en el cliente como en el servidor. El servidor conoce sobre el estado de sus recursos, aunque no intenta seguirle la pista las sesiones individuales de los clientes. Esta es la misión del navegador, el sabe como navegar de recurso en recurso y recoger información que él necesita o cambiar el estado que necesita cambiar.

### **HATEOAS (Hypermedia como mecanismo de estados de aplicación):**

Es el principio sobre el cual se basa la navegación en Internet a través de hyperlinks.



La aplicación de HATEOAS en REST indica que el acceso a los recursos debe ser manejado a través de estados y las transiciones entre éstos deben realizarse mediante hyperlinks que a su vez pueden estar enlazados hacia otros recursos con sus respectivos hyperlinks.

Este principio fundamental de Internet es conocido como HATEOAS (Hypermedia As The Engine Of Application State) y ha sido adoptado por los servicios basados en REST y en el cual se resalta el uso de Hypermedia que es una tecnología orientada al enlace de recursos mediante hyperlinks.

En el ejemplo mostrado a continuación, se visualiza cómo partir de un recurso en la Web es posible acceder a la representación de otros recursos asociados al mismo a través del principio HATEOAS.

GET <http://orders.myintranet.com/orders/111>

```
<order id="111">
  <customer>
    http://customers.myintranet.com/customers/32133
  </customer>
  <order-entries>
  <order-entry>
    <quantity>5</quantity>
    <product>
      http://products.myintranet.com/productos/111
    </product>
  ...
```

### 3. JSR 311 – Java API for RESTful web services

#### **JSR 311 Java API for RESTful Web Services**

- Especificación estándar que propone una estructura basada en anotaciones para desarrollo de RESTful Web Services dentro del entorno de ejecución Java
- Simplifica desarrollo Web Services basados en REST-Style
- Es independiente a la plataforma
- Utiliza las anotaciones para definir los recursos `@PATH` y sus operaciones `@GET`, `@POST`, `@PUT` entre otros

4 - 8

Copyright © Todos los Derechos Reservados - Cibertec Perú SAC.



---

La especificación JSR 311 – JAX-RS propone un modelo de programación estándar que simplifica el desarrollo de web services basados en REST-style.

Es independiente de plataforma. Los servicios desarrollados en una plataforma deben ser fácilmente portables a otras plataformas.

Utiliza anotaciones para definir los recursos `@Path` y sus respectivas operaciones `@GET`, `@POST`, `@PUT` entre otras.

Se basa en el concepto de POJO (Plain Old Java Object) para definir sus componentes.

Provee soporte para múltiples formatos de comunicación como XML, JSON, XHTML, HTML entre otros.

Anotaciones JAX-RS	Significado	Ejemplo
@Path	Especifica una referencia relativa de una URL.	@Path("/productos")
@Produces	Especifica el MIME type de la representación de un recurso al ser enviado hacia el cliente.	@Produces("text/html")
@Consumes	Especifica el MIME type de la representación de un recurso al ser enviado desde el cliente.	@Consumes("text/plain")

Métodos HTTP	Significado	Ejemplo
GET	Utilizado para obtener la representación de un recurso	GET /productos/311
POST	Utilizado para crear un nuevo recurso	POST /productos
PUT(*)	Utilizado para actualizar la información de un recurso	PUT /productos/311
DELETE(*)	Utilizado para eliminar lógicamente un recurso	DELETE /productos/311
HEAD(**)	Similar al GET pero sólo obtiene un código de respuesta y cabeceras asociadas a la petición mas no el recurso en sí.	HEAD /productos/311
OPTIONS(**)	Utilizado para determinar las capacidades del servidor y sus recursos	OPTIONS * HTTP/1.1

(\*) No soportados en la mayoría de web browsers

(\*\*) Manejados internamente por JAX-RS

Anotaciones JAX-RS	Significado	Ejemplo
@PathParam	Extrae parámetros de la URL	@Path("productos/{id}") @PathParam("id")
@QueryParam	Extrae parámetros del queryString de la URL	@QueryParam("descripcion")
@FormParam	Extrae parámetros de un formulario HTML	@FormParam("precio")
@HeaderParam	Extrae parámetros enviados en la cabecera de la petición	@HeaderParam("User-Agent")
@CookieParam	Extrae parámetros enviados en los cookies de la cabecera	@CookieParam("last-visit")
@MatrixParam	Extrae parámetros compuestos de la petición	@MatrixParam("fabricante")



## 4. JAX-RS Jersey Reference Implementation

### JAX-RS Jersey Reference Implementation

- Jersey es una implementación de referencia de JAX-RS
- Forma parte de la comunidad Glassfish
- Puede ser utilizada de manera integrada en su servidor de aplicaciones o en modo stand-alone con otros servidores
- Posee alto rendimiento, extensibilidad y simplicidad de uso en su implementación

4 - 13

Copyright © Todos los Derechos Reservados - Cibertec Perú SAC.

