



UNJu | Universidad
Nacional de Jujuy



Argentina
programa
4.0

Unidad 3 - Clase 2

Otros métodos en Java



Unidad 3 | Programación WEB con Java



- Introducción a JAVA Web.
Arquitectura REST. Métodos HTTP. GET. POST.

- Otros métodos

- JSP (Java Server Pages)
- Presentación del TP Final



Unidad 3 | Métodos en JAVA



Un método es una colección de sentencias que realizan una tarea específica. Puede retornar un valor que podemos utilizar en nuestro programa o no retornar ninguno, esto es opcional. Empleando métodos ahorramos tiempo al no tener que reescribir un conjunto de sentencias varias veces.





UNJu | Universidad
Nacional de Jujuy



Argentina
programa
4.0

Métodos PUT, POST, DELETE



Métodos PUT, POST, DELETE



Los métodos PUT, POST y DELETE se pueden utilizar para modificar los recursos de objeto de negocio y los recursos de estructura de objeto. Sin embargo, las reglas empresariales de un objeto pueden impedir que se actualice mediante una solicitud de la API de REST.

Por ejemplo, una solicitud DELETE en un recurso de orden de trabajo puede fallar si las validaciones del objeto de negocio impiden las supresiones a causa del estado actual del objeto de negocio.



Métodos PUT, POST, DELETE - Sintaxis



método uri HTTP/1.1

parámetro=valor&parámetro=valor&...

método es PUT, POST o DELETE

uri es un URI y los parámetros de consulta asociados

parámetro es un parámetro del recurso que se actualiza

valor es el valor del parámetro



Método PUT



Utilice el método PUT para actualizar o insertar un recurso. Una solicitud de actualización debe proporcionar el ID exclusivo del recurso. Para actualizar un recurso de estructura de objeto, se requiere el ID del objeto principal.

Para actualizar o insertar un recurso de estructura de objeto, puede especificar el parámetro `_action` para identificar acciones proporcionadas por la infraestructura de integración, por ejemplo, las acciones `Change` o `AddChange`.



Método PUT



Ejemplo: actualizar un activo

El método siguiente actualiza el activo que tiene el ID 1234. La descripción del activo se cambia por `mi_nueva_descripción` y el tipo se cambia por `OPERATING`:

```
PUT maxrest/rest/mbo/asset/1234 HTTP/1.1
```

```
description=mi_nueva_descripción&type=OPERATING
```



Método PUT



Ejemplo: actualizar un objeto de nivel inferior

El método siguiente actualiza el objeto de negocio assetspec, que es un nivel inferior de asset. El valor se cambia por nuevo_valor y la descripción se cambia por mi_nueva_descripción:

```
PUT maxrest/rest/mbo/assetspec/5678 HTTP/1.1
```

```
value=mi nuevo valor&description=mi nueva descripción
```



Método PUT



Ejemplo: establecer el contexto de un objeto de nivel superior

Para insertar o actualizar un objeto de negocio de nivel inferior, es posible que tenga que establecer el contexto del objeto de negocio de nivel superior, como en el ejemplo siguiente, en el que se actualiza la descripción de la línea de OC y se proporciona el ID de la OC para el contexto:

```
PUT maxrest/rest/mbo/po/1234/poline/5678 HTTP/1.1  
description=mi nueva descripción
```



Método POST



Utilice el método POST para actualizar o insertar un recurso.

Para actualizar un recurso, debe especificar el ID del recurso. Para crear un recurso, debe especificar la clave primaria y todos los campos necesarios que no tienen un valor predeterminado, pero no es necesario ningún ID.

Para actualizar o insertar un recurso de estructura de objeto, puede especificar el parámetro `_action` para identificar acciones proporcionadas por la infraestructura de integración, por ejemplo, `Change` o `AddChange`.



Método POST



Cuando utilice el método POST para crear un recurso, especifique el parámetro de consulta `_ulcr` con el valor 1 para que la respuesta incluya un vínculo para que el cliente acceda al nuevo recurso. De lo contrario, el contenido del recurso se incluye en la respuesta. El vínculo se incluye en la propiedad de cabecera Ubicación y el código de respuesta HTTP es 201 para identificar que se proporciona un vínculo en lugar de los datos.



Método POST



Ejemplo: insertar un activo

El método siguiente inserta el activo 127 dentro de la planta BEDFORD utilizando un recurso de objeto de negocio. El activo y la planta forman la clave primaria.

```
POST maxrest/rest/mbo/asset HTTP/1.1
```

```
assetnum=127&siteid=BEDFORD&description=mi nueva descripción&type=OPERATING
```



Método POST



Ejemplo: actualizar un activo especificando su ID

El método siguiente actualiza un activo proporcionando el ID como parte del URI:

```
POST maxrest/rest/mbo/asset/1234 HTTP/1.1
```

```
description=mi nueva descripción&type=OPERATING
```



Método DELETE



El método DELETE requiere el ID exclusivo del recurso.

Para suprimir un recurso de estructura de objeto, se requiere el ID del objeto principal.



Método DELETE



Ejemplo: suprimir un activo

El método siguiente suprime el activo 1234:

```
DELETE maxrest/rest/mbo/asset/1234 HTTP/1.1
```



Método PATCH



El método HTTP PATCH se utiliza para realizar una actualización parcial de un recurso de OSLC. A diferencia del método PUT, el método PATCH no suprime las propiedades de recurso local que no están incluidas en la petición. El método PATCH se debe especificar utilizando la cabecera x-method-override.



Método PATCH



La implementación de PATCH tiene dos casos de uso:

Puede sustituir una propiedad de recurso local con el contenido de la petición. Este caso de uso es la implementación predeterminada.

Puede buscar y comparar los elementos de matriz de recurso contenidos en la petición con los elementos existentes en el servidor. Dependiendo de si se ha encontrado una coincidencia de búsqueda, los elementos de matriz de recurso se actualizan o insertan. Un elemento de matriz nunca se suprime de la propiedad del recurso local. Para utilizar este caso de uso, el cliente debe establecer la cabecera de petición HTTP PATCHTYPE en el valor `MERGE` (se distingue entre mayúsculas y minúsculas).



Método PATCH



Ejemplo: Actualización de una propiedad literal

El método siguiente actualiza la propiedad de título de la orden de trabajo:

```
POST /maximo/oslc/os/oslcwodetail/abc
```

```
x-method-override: PATCH
```

```
{
```

```
  "dcterms:title": "Check-out Leaking - Modified for Test"
```

```
}
```

A diferencia del método PUT, el método PATCH no actualiza otras propiedades de la orden de trabajo.

