

Servicios REST

HTTP

HTTP es el protocolo que permite enviar documentos de un lado a otro en la web. Un protocolo es un conjunto de reglas que determina qué mensajes se pueden intercambiar y qué mensajes son respuestas apropiadas a otros. Otro protocolo común es POP3, que puede utilizar para buscar correo electrónico en su disco duro.

En HTTP, hay dos funciones diferentes: servidor y cliente. En general, el cliente siempre inicia la conversación; El servidor responde. HTTP está basado en texto; Es decir, los mensajes son esencialmente bits de texto, aunque el cuerpo del mensaje también puede contener otros medios. El uso del texto facilita el monitoreo de un intercambio HTTP.

VERBOS HTTP

Cada solicitud especifica un cierto verbo o método HTTP, en el encabezado de la solicitud. Esta es la primera palabra de todos las mayúsculas en el encabezado de la solicitud. Por ejemplo,

GET / HTTP/1.1

Significa que se está utilizando el método GET, mientras

DELETE /clients/anne HTTP/1.1

significa que se utiliza el método DELETE.

HTTP verbs tell the server what to do with the data identified by the URL. La solicitud puede contener opcionalmente información adicional en su cuerpo, que podría ser necesaria para realizar la operación, por ejemplo, los datos que desea almacenar con el recurso. Puede proporcionar estos datos en cURL con la opción -d.

GET

GET es el tipo más simple de método de solicitud HTTP; La que usan los navegadores cada vez que hace clic en un enlace o escribe una URL en la barra de direcciones. Indica al servidor que transmita los datos identificados por la URL al cliente. Los datos nunca deben ser modificados en el lado del servidor como resultado de una solicitud GET. En este sentido, una petición GET es de sólo lectura, pero por supuesto, una vez que el cliente recibe los datos, es libre de hacer cualquier operación con ella por su cuenta, por ejemplo, formatearla para su visualización.

PUT

Una petición PUT se utiliza cuando se desea crear o actualizar el recurso identificado por la URL. Por ejemplo,

PUT /clients/robin

Podría crear un cliente, llamado Robin en el servidor. Usted notará que REST es completamente agnóstico de servidor; No hay nada en la solicitud que informe al servidor cómo deben crearse los datos, sólo que debería. Esto le permite intercambiar fácilmente la tecnología del servidor si la necesidad surge. Las peticiones PUT contienen los datos que se utilizarán para actualizar o crear el recurso en el cuerpo. En cURL, puede agregar datos a la solicitud con el modificador -d.

```
curl -v -X PUT -d "some text"
```

DELETE

DELETE debe realizar lo contrario de PUT; Debe utilizarse cuando desee eliminar el recurso identificado por la URL de la solicitud.

```
curl -v -X DELETE /clients/anne
```

Esto eliminará todos los datos asociados con el recurso, identificados por /clients/anne.

POST

POST se utiliza cuando el procesamiento que desea que suceda en el servidor debe repetirse, si la solicitud POST se repite (es decir, no son idempotent, más de esto a continuación). Además, las solicitudes POST deben causar el procesamiento del cuerpo de la solicitud como un subordinado de la URL que está publicando.

En palabras claras:

```
POST /clients/
```

No debe causar que el recurso en /clients/, se modifique, sino un recurso cuya URL comience con /clients/. Por ejemplo, podría agregar un nuevo cliente a la lista, con un ID generado por el servidor.

```
/clients/some-unique-id
```

PUT solicitudes se utilizan fácilmente en lugar de solicitudes POST, y viceversa. Algunos sistemas utilizan sólo uno, algunos utilizan POST para crear operaciones y PUT para operaciones de actualización (ya que con una solicitud PUT siempre proporcionan la URL completa), algunos incluso utilizan POST para actualizaciones y PUT para crear.

A menudo, las solicitudes POST se utilizan para activar las operaciones en el servidor, que no encajan en el paradigma Crear / Actualizar / Eliminar; Pero esto, sin embargo, está más allá del alcance de REST.

Bibliografía

- <https://code.tutsplus.com/es/a-beginners-guide-to-http-and-rest--net-16340t>