

Spring framework 5 - HTTP

HTTP

HTTP (*Hyper Text Transfer Protocol*) es un protocolo de la capa de aplicación que define un conjunto de reglas para transferir archivos, texto, imágenes, sonidos, videos, etc.

Métodos HTTP

El protocolo HTTP define múltiples métodos listados a continuación:

- GET** : Es utilizado para obtener información del servidor
- HEAD**: Es igual que GET pero solo devuelve el *status* y los *headers*
- POST**: Es utilizado para enviar información al servidor
- PUT**: Reemplaza el contenido del recurso con el nuevo
- PATCH**: Aplica modificaciones parciales al recurso especificado
- DELETE**: Borra la información del recurso especificado
- TRACE**: Es utilizado para *debugging* enviando un *echo* de vuelta al usuario
- OPTIONS**: Describe las opciones de comunicación con el servidor
- CONNECT**: Establece un tunel con el servidor basado en la URL

URI

URI (*Uniform Resource Identifier*) Es el nombre que se utiliza para identificar un recurso de forma única. Contiene 3 partes:

- Schema**: Protocolo a utilizar para acceder al recurso
- Host**: Ubicación del servidor (DNS, hostname o IP)
- Path**: Ruta al recurso en el servidor

Petición HTTP

Una petición HTTP se conforma de lo siguiente:

- Método HTTP**: Método HTTP, ejemplo GET
- URI**: URI del recurso solicitado, ejemplo: `http://www.twitter.com/search?q=devs4j`
- Versión**: Versión del protocolo utilizado, ejemplo 1.1
- Headers**: Información adicional de la petición, ejemplo `Accept:text/html`.
- Body**: Cuerpo de la petición HTTP

Respuesta HTTP

Una respuesta HTTP se conforma de lo siguiente:

- Versión**: Versión del protocolo utilizado, ejemplo 1.1
- Status code**: Estatus de la respuesta
- Headers**: Información adicional de la respuesta, ejemplo `Transfer-Encoding: chunked`.
- Body**: Cuerpo de la respuesta HTTP

Connectionless

HTTP es un protocolo al que se le conoce como *connectionless*. Esto significa que no se debe mantener una conexión viva hacia el servidor, de este modo el cliente y el servidor se conocen solo durante la petición.

Stateless

Al ser un protocolo *connectionless*, también es considerado como *stateless*, esto significa que cada petición es independiente a la anterior. Dado este comportamiento, ni el cliente ni el servidor retienen información durante diferentes peticiones.

Status HTTP

Un *status* HTTP es un número que representa el resultado de una petición. Se clasifica en rangos:

- 1xx (100-199)**: Informacional
- 2xx (200-299)**: Petición exitosa
- 3xx (300-399)**: Redirecciones
- 4xx (400-499)**: Error del lado del cliente
- 5xx (500-599)**: Error del lado del servidor

REST

REST (*Representational State Transfer*) Es un estilo de arquitectura que define un conjunto de reglas para comunicar aplicaciones, se basa en el protocolo HTTP.

Resource

La principal abstracción de información en REST es un recurso. Cualquier información que puede ser nombrada puede ser un recurso

Definición de recursos

Una mala práctica común al definir recursos es utilizar verbos en su definición. A continuación un ejemplo:

- Método http**: GET
- Endpoint**: `/getUsers`
- Descripción**: Devuelve una lista de usuarios

Esto es una mala práctica por los siguiente:

- Es redundante decir que el recurso es `/getUsers` y se utilizará el método HTTP GET
- El recurso `/getUsers` debería soportar otros métodos HTTP, al hacerlo se volvería contradictorio dado que el *endpoint* sería POST `/getUsers`

Peticiones exitosas

A continuación se presentan los *status* HTTP que se utilizan para representar **peticiones exitosas** de acuerdo a cada método HTTP:

- 200 OK**: La petición fue exitosa. Se puede aplicar a métodos como GET, HEAD, PUT y TRACE
- 201 CREATED**: La petición fue exitosa y se creó un nuevo recurso. Se genera en métodos POST.
- 204 NOT CONTENT**: No hay contenido que devolver. Se utiliza como resultado de un método DELETE.

Errores del lado del cliente

Algunos errores comunes del lado del cliente:

- 400 BAD REQUEST**: Error genérico que indica que existe un error del lado del cliente
- 401 UNAUTHORIZED**: Cuando el cliente no tiene suficientes privilegios para acceder a un recurso
- 404 NOT FOUND**: El recurso solicitado no existe
- 405 METHOD NOT ALLOWED**: El método HTTP no esta soportado en el recurso solicitado
- 429 TO MANY REQUESTS**: Demasiadas peticiones. Se utiliza cuando se tiene una cantidad límite de peticiones por cliente.
- 409 CONFLICT**: Existe un conflicto



Errores del lado del servidor

Algunos errores del lado del servidor:

- 500 INTERNAL SERVER ERROR**: Error genérico que significa que existe un error del lado del servidor
- 501 NOT IMPLEMENTED**: Indica que el servidor no soporta la funcionalidad solicitada
- 502 BAD GATEWAY**: Error que indica que el servidor recibió un error al invocar a otro servidor
- 503 SERVICE UNAVAILABLE**: El servidor no puede procesar la petición.

Versionamiento de API

Existen múltiples formas de versionar un API y existen discusiones activas sobre cual es la mejor opción:

-**Versionamiento en la URL**: La versión del API se define en la URL, ejemplo:

`HTTP GET: /application/api/v1/users`

-**Versionamiento en los headers**: La versión del API se define en los *headers*:

`HTTP GET: /application/api/users`
`Accept: application/raidentrance.api.v2+json`

-**Versionamiento en el dominio**: La versión del API se define en el dominio, ejemplo:

`apiv1.devs4j.com/users`

-**Versionamiento en un request param**: La versión del API se define en un *request param*, ejemplo:

`HTTP GET: /application/api/v1/users?version=1`

Paginación, filtros y ordenamiento

Cuando se desarrollan servicios web REST, una buena práctica es proveer paginación, filtros y ordenamiento a través de **query params**, a continuación se muestran algunos ejemplos:

Filtros

-GET `/cars?color=red` : Devuelve una lista de autos color rojo

-GET `/cars?seats<=2` : Devuelve una lista de autos con 2 asientos o menos

Ordenamiento

-GET `/cars?sort=-color&+model` : Devuelve una lista de autos ordenada de forma descendente por color y ascendente por modelo

Paginación

-GET `/cars?page=1&size=10` : Devuelve la página 1 de autos de un tamaño de 10

