

Diseño de APIs: Protocolo HTTP



Mensajes HTTP

En el protocolo HTTP existen 2 tipos de mensajes:

- **Peticiones HTTP (Requests):** Son los mensajes que envía el cliente al servidor para solicitar información o realizar una operación
- **Respuestas HTTP (Responses):** Son las respuestas que el servidor envía al cliente para responder a sus peticiones

Petición HTTP

Toda petición HTTP está compuesta de:

- **Método HTTP** [https://datos.codeandcoke.com/apuntes:http#metodos_http]: El tipo de operación que se quiere realizar (registro, consulta, modificación, . . .)
- **Ruta o path** [https://datos.codeandcoke.com/apuntes:http#modelado_de_recursos_uris]: La ruta completa para identificar el recurso sobre el que se quiere operar
- **Cabeceras:** Información adicional para el servidor que procesa la petición
- **Cuerpo:** La información que se quiere enviar junto con la petición (sin contenido en operaciones GET). En una operación de registro, sería la información que se quiere registrar

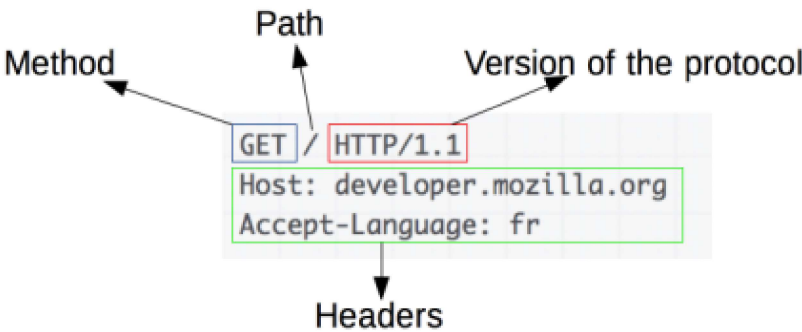


Figure 1: Estructura petición HTTP (fuente: <https://mozilla.org> [<https://mozilla.org>])

Respuesta HTTP

Toda petición HTTP conlleva una respuesta, ya sea por haberse ejecutado ésta correctamente o bien para informar de un error. Estará compuesta de:

- **Código de estado** [https://datos.codeandcoke.com/apuntes:http#codigos_de_estado]: Indica si la petición tuvo o no éxito y por qué.
- **Mensaje de estado:** Una descripción breve sobre el código de estado
- **Cabeceras:** Información adicional para el cliente que realizó la petición
- **Cuerpo:** La información que se quiere enviar al cliente. En una operación de consulta serían los datos que se solicitaron en la petición

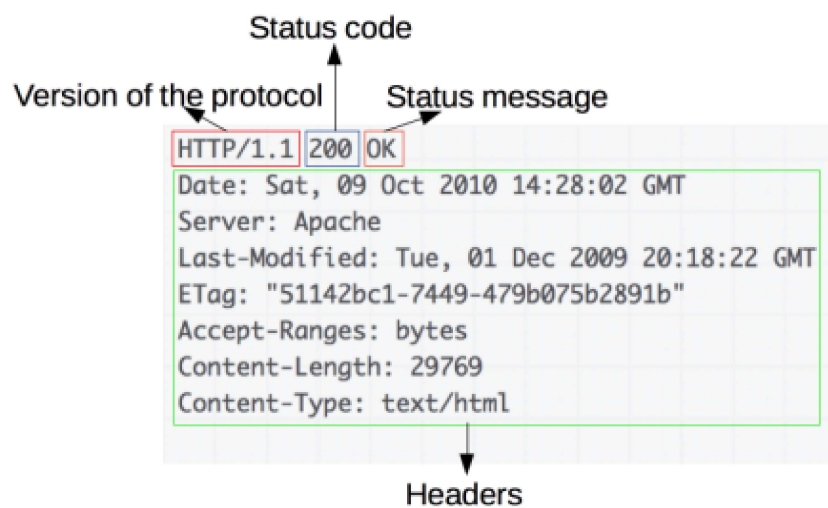


Figure 2: Estructura respuesta HTTP (fuente: <https://mozilla.org> [<https://mozilla.org>])

Métodos HTTP

El protocolo HTTP define una serie de métodos que permiten indicar el tipo de operación que se quiere realizar en toda petición HTTP.

La fundación Mozilla mantiene una documentación muy buena sobre todos los métodos que existen en el protocolo.

Documentación métodos HTTP en mozilla.org [<https://developer.mozilla.org/en-US/docs/Web/HTTP/Methods>]

Como durante el curso no veremos todos ellos, dejaremos en esta Wiki un resumen de lo más importante acerca de los métodos con los que vamos a trabajar.

Métodos HTTP más utilizados

En el protocolo HTTP, el método define el tipo de operación que se va a realizar. De entre los métodos existentes en el protocolo, los más utilizados son los que se corresponden con las operaciones CRUD:

Método	Cuándo usar	Ejemplo
GET	Operación de consulta	GET /books
POST	Operación de registro o acciones	POST /books ó POST /loan/12/cancel
DELETE	Operación de borrado	DELETE /book/12
PUT	Operación de modificación completa	PUT /book/12
PATCH	Operación de modificación parcial	PATCH /book/12

Cabe destacar el uso del método POST para todas aquellas operaciones que no se pueden clasificar como una operación CRUD de alta, baja, modificación o consulta de recurso. Por ejemplo, cancelar un préstamo, devolver un pedido, desactivar una tarjeta de crédito, . . .

Modelado de Recursos/URIs

Las URIs de cada recurso serán cada uno de los métodos que desarrollamos en el proyecto de nuestra API y mapeamos usando las anotaciones `@GetMapping`, `@PostMapping`, `@DeleteMapping`, `@PatchMapping` o `@PutMapping`.

Con el objetivo de definir una colección uniforme de operaciones o endpoints, es muy importante perder algo de tiempo para pensar y diseñar cómo deben de ser estas URLs.

El primer paso es tener en cuenta que el método HTTP seleccionado actuará como verbo y que la URL deberá definir al recurso sobre el que ese 'verbo' actúa (luego veremos que no todas las operaciones encajan en este modelo).

A continuación se muestran algunas URLs a modo de ejemplo:

Operaciones de consulta

- GET /products
- GET /products?category=food
- GET /products?category=food&price=100

- GET /product/{productId}
- GET /user/{userId}/orders
- GET /user/{userId}/order/{orderId}

Operaciones de registro

- POST /products
- POST /user/{userId}/orders

Operaciones de borrado

- DELETE /product/{productId}
- DELETE /user/{userId}/order/{orderId}

Operaciones de modificación (registro completo)

- PUT /product/{productId}
- PUT /user/{userId}/order/{orderId}

Operaciones de modificación parcial

- PATCH /product/{productId}
- PATCH /user/{userId}/order/{orderId}

Por ejemplo:

Task	Method	Path
Create a new task	POST	/tasks
Delete an existing task	DELETE	/tasks/{id}
Get a specific task	GET	/tasks/{id}
Search for tasks	GET	/tasks
Update an existing task	PUT	/tasks/{id}

Figure 3: Rutas para operaciones en una API

Códigos de estado

El protocolo HTTP define una serie de códigos de estado que permiten indicar el resultado de toda petición HTTP. Estos códigos proporcionan al cliente que realizó la petición información global sobre el resultado que se devuelve en la respuesta.

La fundación Mozilla mantiene una documentación muy buena sobre todos estos códigos de estado.

Códigos de estado HTTP [https://developer.mozilla.org/en-US/docs/Web/HTTP/Status]

Como durante el curso no veremos todos ellos, dejaremos en esta Wiki un resumen de lo más importante acerca de los códigos de estado con los que vamos a trabajar.

Existen 5 categorías:

- **1XX Información** Son el grupo de códigos que proporcionan información a nivel de protocolo
- **2XX Éxito** Se utilizan para indicar al cliente que la petición se ha realizado con éxito
- **3XX Redirección** Sirven para indicar al cliente que debe realizar alguna acción adicional para completar la petición
- **4XX Error en el cliente** Se utilizan para indicarle al cliente que ha cometido algún tipo de error al realizar la petición
- **5XX Error en el servidor** Indican que se ha producido algún tipo de error en el servidor mientras se ejecutaba la petición

Códigos de estado HTTP más utilizados

Código	Estado	Cuándo usar
200	OK	Ok en GET,PUT,PATCH
201	Created	Ok en POST (registro)
204	Not Content	Ok sin devolver datos (normalmente en DELETE)
400	Bad Request	Error de invocación del cliente
401	Unauthorized	Usuario/Contraseña incorrectos
403	Forbidden	Sin privilegios
404	Not Found	Recurso no encontrado (especificado en la entrada)
500	Internal Server Error	Error en backend

Ejercicios

1. ¿Qué código de estado HTTP devolverá una operación que devuelve la lista de usuarios conectados cuando no haya ninguno? ¿Y qué información devolverá en la respuesta?
 2. ¿Qué código de estado HTTP debe devolver una operación que registra un nuevo producto en la base de datos? ¿Debe devolver algo como respuesta?
 3. ¿Qué código de estado HTTP y respuesta devolverá una operación que utiliza el método HTTP DELETE?
 4. ¿Qué código de respuesta HTTP devolverá una operación que sirve para conocer la información de un producto determinado? ¿Y si no lo encuentra?
-