

HTTP

¿Qué es?

Hypertext Transfer Protocol

¿Qué función cumple?

Define la sintaxis y la semantica que usan los elementos del software de la arquitectura web para comunicarse

Versiones

HTTP/1.1, HTTP/1.2, HTTP/2 o HTTP/3

¿Cómo funciona?

Mediante una estructura basada en petición-respuesta, el cliente realiza una petición con un formato medido y que puede o no albergar información, el servidor le enviara una respuesta.

Estructura de los mensajes

Está organizada en 3 líneas:

1ª línea:

- Para peticiones: acción que se le requiere al servidor, URL del recurso y versión HTTP soportada.
- Para respuestas: versión HTTP usada, código de respuesta y URL del recurso.

2ª línea: compuesta por metadatos que aportan flexibilidad al protocolo.

3ª línea: depende de la segunda línea y de la acción que se le requiera al servidor. Suele contener información que se le envía al servidor en la petición, mientras que en la respuesta puede traer los datos solicitados

Verbos

- GET: solicita un recurso
- POST: envía información para crear un recurso
- PUT: actualiza un recurso quitando lo que había antes
- DELETE: borra un recurso

Códigos de respuesta

- 1xx: respuestas informativas
- 2xx: respuestas correctas
- 3xx: respuestas de redirección
- 4xx: errores debido al cliente
- 5xx: errores debido al servidor

JSON

¿Qué es?

Es un formato basado en texto con la sintaxis de JavaScript. Es muy útil para transmitir datos en red.

Almacena con una extensión .json y un MIME (son una serie de convenciones dirigidas al intercambio de archivos a través de internet).

Estructura

Es posible incluir los mismos tipos básicos que en JavaScript (Cadenas, números, booleanos, arrays, objetos...). Los arrays se representan entre corchetes [] mientras que los objetos son entre llaves { }.

Ventajas

- Es autodescriptivo y fácil de entender.
- Se parsea más rápido.
- Velocidad de procesamiento alta.

JSON en Java

Para trabajar con JSON en Java encontramos que podemos parsear un objeto a JSON (para enviar datos a través de HTTP) o podemos pasar de JSON a Java (útil para recibir información). Para hacer este cambio de JSON a Java encontramos herramientas como la librería Jackson, que hará automáticamente el cambio si definimos los objetos que vayamos a utilizar.

Ejemplo

```
{
  "alumnos": ["Antonio Montero","Daniel Fernández"],
  "curso": "2DAM",
  "asignaturas": ["Acceso a datos","Programación de servicios y procesos"]
}
```

REST

¿Qué es?

Es un estilo de arquitectura para realizar una circulación de mensajes entre cliente y servidor.

Niveles

Nivel 0

No tiene ningún tipo de organización y se accede a la información directamente, utilizando por ejemplo el protocolo HTTP se envía un mensaje al servidor en algún formato como JSON o XML lo leerá y nos devolverá una respuesta.

Nivel 1: recursos

En el encontramos organización gracias a la utilización de recursos (suelen ser conceptos de negocio) permitiendo así realizar las peticiones de forma independiente a cada uno de estos, favorece la reutilización, permite el borrado, búsquedas, etc ...

Nivel 2: verbos

Las peticiones se definen de forma más concreta y dependiendo de la operación se utilizará un tipo diferente de envío. Algunos tipos son:

- GET (solicita consultas)
- POST (inserta recursos)
- PUT (actualiza recursos)
- DELETE (borra recursos).

Nivel 3: Hateoas

En el podemos acceder a un recurso cuya URL este contenida en otro recurso al cuál hemos hecho la petición, de esta forma se reduce el acoplamiento y aumenta la flexibilidad.