

# Tema 9

Creando Node.js REST API

# ¿Qué es REST Api?

- Hablamos de REST Api o librería de funciones a una serie de acciones dentro de nuestra web a las que podemos acceder a través del protocolo HTTP.
- Por lo tanto, realizaremos diferentes peticiones sobre urls definidas en nuestro proyecto y las respuestas que obtenemos vendrán dadas en formatos muy sencillos para ser parseados desde otro sistema diferente.
- Es una manera muy sencilla de crear puntos de acceso a nuestra web para que se puedan acceder a los datos que tenemos almacenados desde cualquier sistema, como por ejemplo, podría ser un dispositivo móvil.

# Protocolo HTTP

- Una transacción Http está formada por un encabezado en el cual se pueden especificar cosas como la acción requerida del servidor, el tipo de dato retornado o el código de estado.
- El uso de este tipo de encabezados nos permite enviar información descriptiva en la transacción, permitiendo así la autenticación, cifrado e identificación del usuario.
- El servidor envía al cliente un código de estado que indica si la petición fue correcta o no.
- Aparte recibimos los datos propiamente dichos de la transacción. Se puede encapsular cualquier tipo de archivo y todo tipo de formato.
- También, en una petición Http podemos encontrar información sobre el objeto que se retorna.

# Métodos de Petición

- Http define 8 métodos que indica la acción que desea que se efectúe sobre el recurso identificado.
- **HEAD.** Pide una respuesta idéntica a la que conseguiríamos con la petición GET pero sin el cuerpo de la respuesta, sólo recibiríamos la cabecera con la información de la petición.
- **GET.** Pide una representación del recurso especificado. Conseguiremos la información completa del recurso, incluyendo la cabecera con la información del mismo.
- **POST.** Envía los datos para que sean procesados por el recurso identificado. Es la petición más común cuando estamos, por ejemplo, enviando la información desde un formulario web.

# Métodos de Petición

- **PUT.** Sube, carga o realiza un upload de un archivo especificado. Es el camino más eficiente para subir archivos a un servidor.
- **DELETE.** Borra el recurso especificado.
- **TRACE.** Pedimos al servidor que nos envíe un mensaje de confirmación con los mismos datos enviados en la petición. Podemos usarlo para realizar comprobaciones sobre el estado del servidor con el que estamos trabajando.
- **OPTIONS.** Nos devuelve los métodos HTTP que el servidor soporta para una URL específica.
- **CONNECT.** Se utiliza para saber si se tiene acceso a un host.

# Respuestas HTTP

- Dependiendo de la respuesta obtenida en la petición nos devolverá diferentes respuestas informando de lo sucedido en el proceso.
- Podemos encontrar las diferentes respuestas en este enlace

[http://es.wikipedia.org/wiki/Anexo:Códigos\\_de\\_estado\\_HTTP](http://es.wikipedia.org/wiki/Anexo:Códigos_de_estado_HTTP)

# Rest API en Node

- Para poder implementar un servicio REST en Node.js, como veremos en los ejercicios, simplemente tenemos que tratar las diferentes peticiones, como hemos hecho en los temas anteriores y procesando el contenido de la respuesta, teniendo en cuenta la url sobre la cual hacemos la petición y el tipo de petición.
- En los ejercicios veremos cómo implementar un servicio de este tipo utilizando Express como base de nuestro proyecto y como ayuda para mapear las diferentes urls.