¿Que es una comunicación REST?

La transferencia de estado representacional (en inglés representational state transfer) o REST es un estilo de arquitectura software para sistemas hipermedia distribuidos como la World Wide Web.

Para que se usan? Ejemplifique.

REST es una interfaz para conectar varios sistemas basados en el protocolo HTTP (uno de los protocolos más antiguos) y nos sirve para obtener y generar datos y operaciones, devolviendo esos datos en formatos muy específicos, como XML y JSON.

- La comunicación REST (Representational State Transfer) es un estilo arquitectónico parasistemas distribuidos que permite la comunicación entre diferentes aplicaciones webutilizando el protocolo HTTP.
- En la comunicación REST, T, las aplicaciones web interactúan a través de una API (Interfaz dePrograMación de Aplicaciones) que utiliza un conjunto de verbos HTTP (GET, POST, PUT, DELETE, entre otros) para enviar y recibir datos en formato JSON (JavaScript ObjectNotation) o XML (Extensible Markup Language)
- En este modelo de comunicación, cada recurso (como un objeto o una colección de objetos) se representa con una URL (Uniform Resource Locator) única y es posible

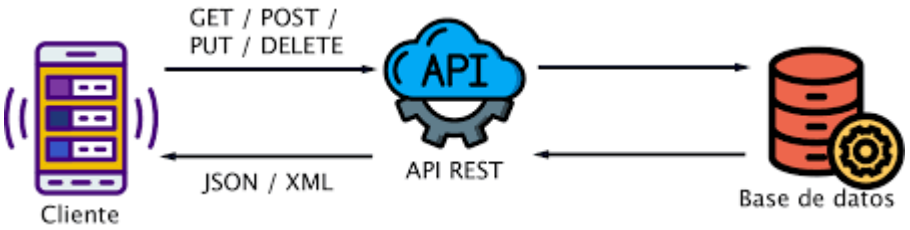
acceder a él mediante el uso de los verbos HTTP mencionados anteriormente. Esto permite a las aplicaciones web solicitar y recibir información específica, actualizarla o eliminarla según sea necesario.

- La comunicación REST se utiliza ampliamente en el desarrollo de aplicaciones web y móviles, ya que permite una comunicación eficiente y escalable entre sistemas distribuidos, independientemente del lenguaje de programación utilizado para desarrollarlos

Los servicios REST :

Te permiten acceder y/o modificar la información mediante los métodos HTTP, por lo cual puedes acceder a ellos mediante URLs. Por lo general regresan la información en formato JSON, aunque también pueden regresar archivos XML o csv.

Una parte importante del desarrollo de aplicaciones web, es entender que se trata de una constante **comunicación** entre dos equipos, el cliente y el servidor

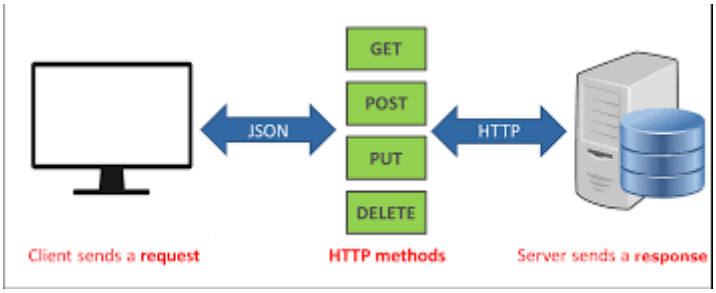


Solicitudes y comunicaciones

REST requiere que un cliente realice una solicitud al servidor para enviar o modificar datos.

Una solicitud consiste en:
● Un verbo o método HTTP que define qué tipo de operación realizará el servidor.
● Un encabezado, con el encabezado de solicitud que pasa información sobre la solicitud.
● Un camino o ruta al servidor, como por ej:
https://app.aluracursos.com/course/http-base-internet
● Información en el cuerpo de la solicitud, siendo esta información opcional

Métodos HTTP En la aplicación REST, los métodos más utilizados son:



● El método GET es el método más común, generalmente se usa para solicitar a un servidor que envíe un recurso;
● El método POST está diseñado para enviar datos de entrada al servidor. En la práctica, a menudo se usa para admitir formularios HTML;
● El método PUT edita y actualiza documentos en un servidor;
● El método DELETE que, como su nombre lo indica, elimina un determinado dato o colección del servidor

Códigos de respuesta

A cada respuesta que devuelve la aplicación REST se le envía un código que define el estado de la solicitud. Por ejemplo:
● 200 (OK) solicitud cumplida con éxito.
● 201 (CREADO) objeto o recurso creado con éxito.
● 204 (SIN CONTENIDO) objeto o recurso eliminado con éxito.
● 400 (MALA SOLICITUD) ocurrió un error en la solicitud (puede haber numerosas causas).
● 404 (NO ENCONTRADO) ruta o colección no encontrada.
● 500 (ERROR INTERNO DEL SERVIDOR), se ha producido algún error del servidor

¿Qué es REST en una imagen?

- Protocolo cliente/servidor "SIN ESTADO"
- Utiliza verbos http
  - GET (leer)
  - POST (crear)
  - PUT (editar)
  - DELETE (borrar)
- Devuelve
  - JSON
  - XML

HTTP STATUS
200 – OK
201 – CREATED
403 – FORBIDDEN
404 – NOT FOUND

Ejemplos:

GET /rest/api/pedido/{id}
DELETE /rest/api/pedido/{id}
POST /rest/api/pedido
PUT /rest/api/pedido/{id}/producto/{idProducto}

Devuelve
JSON/XML

```
{
    "nombre": "Pepito",
    "coches" : ["Ford", "Fiat", "Audi"]
}
```

Explicación

- Se trata de un protocolo SIN ESTADO. Por tanto si el servicio requiere de seguridad/autenticación se tienen que pasar SIEMPRE los datos necesarios para que pueda autenticar en cada llamada (usuario y contraseña, token, …)
- La manera más estándar de interactuar con REST es a través de los verbos HTTP. Por ejemplo, si lo que quiero hacer es leer datos de un servicio REST utilizaría el HTTP GET.
- Los servicios REST suelen devolver un fichero JSON.
- El estado de la operación se realiza a través del HTTP STATUS. ¿A que cuando pides una página web que no existe el HTTP STATUS es 404? De igual manera si pides vía REST los datos de un usuario que no existe en la base de datos también devería devolver un 404.

como ejemplo: suponemos que hay un servicio meteorológico que si le pasamos el nombre de la ciudad nos devuelve un JSON con los datos del tiempo de la próxima semana.

Llamaríamos al servicio a través del verbo GET de esta manera:

`https://serviciodeltiempoquemedaelrest.com/eltiempo?ciudad=Madrid`

La respuesta sería un JSON

```
{
	"ciudad": "Madrid",
	"temperatura_actual": "17ºC",
	"prediccion": {
		"lunes": "15ºC",
		"martes": "16ºC",
		"miercoles": "21ºC",
		"jueves": "20ºC",
		"viernes": "18ºC",
		"sabado": "18ºC",
		"domingo": "12ºC"
```

```
        }
}
```