

Introducción a API REST

Mervin Díaz

<https://www.linkedin.com/in/mervindiazlugo/>

Terminología

Mervin Díaz

<https://www.linkedin.com/in/mervindiazlugo/>

Interfaz

- Permite que tú puedas consumir los servicios del sistema
- Entonces una API (Application Programming Interface) es una interfaz, que permite que aplicaciones de software puedan intercambiar datos entre ellos.
- La interfaz de software: es utilizada para entregar información, concerniente a los procesos y mecanismos de control, mediante lo que la persona observa frecuentemente en la pantalla.

API Local

- Son sistemas de comunicación que conviven bajo el mismo Sistema operativo, por lo general en forma de servicios del SO o middleware a los programas de aplicación.
- Middleware es software que se sitúa entre un sistema operativo y las aplicaciones que se ejecutan en él. Básicamente, funciona como una capa de traducción oculta para permitir la comunicación y la administración de datos en aplicaciones distribuidas.

¿Qué son los web services?

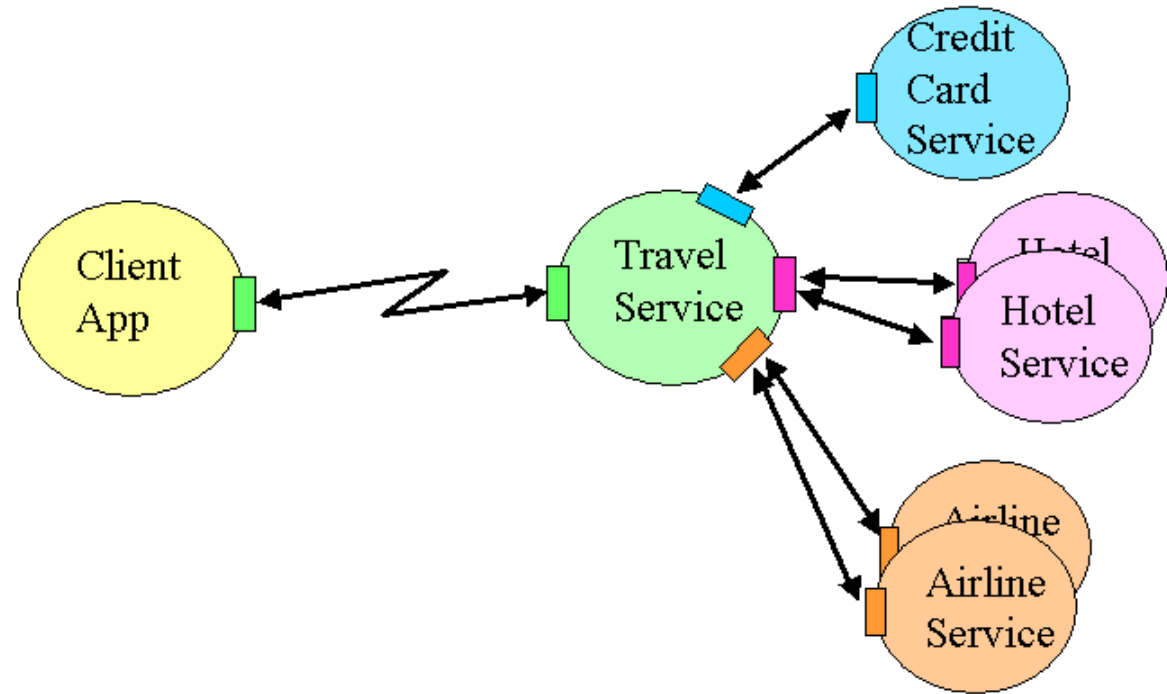
- Un web service es una vía de intercomunicación e interoperabilidad entre máquinas conectadas en Red.
- En el mundo de Internet se han popularizado enormemente, ya se trate de web services públicos o privados.
- Generalmente, la interacción se basa en el envío de solicitudes y respuestas entre un cliente y un servidor, que incluyen datos.

¿Qué son los web services?

- El cliente solicita información, enviando a veces datos al servidor para que pueda procesar su solicitud. El servidor genera una respuesta que envía de vuelta al cliente, adjuntando otra serie de datos que forman parte de esa respuesta.
- Por tanto, podemos entender un servicio web como un tráfico de mensajes entre dos máquinas.

¿Qué son los web services?

Travel Agency Web Service



¿Qué son las APIs y para qué sirven?

Tipos de API

```
graph TD; A[Tipos de API] --> B[Locales]; A --> C[Remotas]; C --> D[Servicios Web]; D --> E[SOAP]; D --> F[Rest]; D --> G[Restful];
```

Locales

Remotas

Servicios
Web

SOAP

Rest

Restful

¿Qué son las APIs y para qué sirven?

- SOAP (Simple Object Access Protocol) no es una arquitectura software, sino un conjunto de restricciones con las que podemos crear un estilo de arquitectura software, la cual podremos usar para crear aplicaciones web respetando HTTP.

SOAP

- SOAP (Simple Object Access Protocol) es un protocolo estándar que define cómo dos objetos en diferentes procesos pueden comunicarse por medio de intercambio de datos XML. No admite otros formatos a diferencia de REST.
- Los servicios SOAP por lo general usan como protocolo de transporte HTTP que es lo más común cuando invocamos servicios web. Pero SOAP es agnóstico al protocolo de transporte, y puede enviar datos usando FTP, POP3, TCP, SMTP, Colas de mensajería (JMS, MQ, etc)...

¿Qué son las APIs y para qué sirven?

- Entre sus ventajas podemos encontrar que al funcionar a través del protocolo de transporte TCP, se pueden utilizar diferentes protocolos de aplicación como: HTTP, SMTP o JMS. También nos brinda la posibilidad de generar cliente/servidor en distintos lenguajes de programación. Y está ampliamente estandarizado, por lo cual hay reglas concretas para formar el mensaje, el contrato entre cliente/servidor o el formato de los datos a enviar, siempre XML.

¿Qué son las APIs y para qué sirven?

- Hoy en día la mayoría de las empresas utilizan API REST para crear servicios. Esto se debe a que es un estándar lógico y eficiente para la creación de servicios web. Por poner algún ejemplo tenemos los sistemas de identificación de Facebook, la autenticación en los servicios de Google (hojas de cálculo, Google Analytics, ...).

¿Qué son las APIs y para qué sirven?

- Las APIs son interfaces que permiten la comunicación entre dos aplicaciones de software siguiendo cierto conjunto de reglas.
- La arquitectura más usada es REST y el formato de envío de datos más usado es JSON.

Rest API Basics

Typical HTTP Verbs:

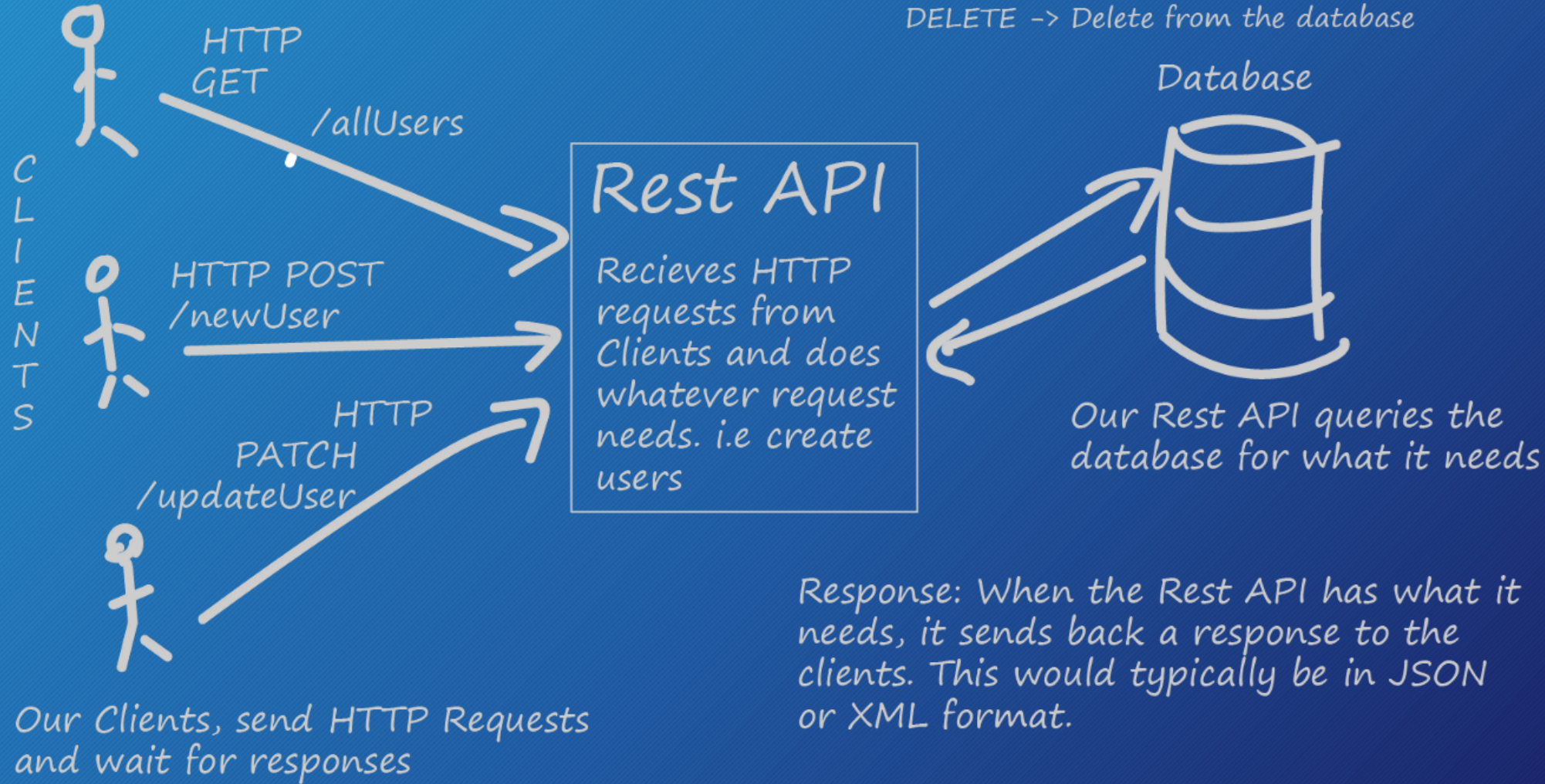
GET -> Read from Database

PUT -> Update/Replace row in Database

PATCH -> Update/Modify row in Database

POST -> Create a new record in the database

DELETE -> Delete from the database



Las APIs y el testing

Mervin Díaz

<https://www.linkedin.com/in/mervindiazlugo/>

APIS y el Testing

- Existen tres niveles de calidad a la hora de aplicar REST en el desarrollo de una aplicación web y que se recogen en un modelo llamado [Richardson Maturity Model](#) en honor al tipo que lo estableció, Leonard Richardson padre de la [arquitectura orientada a recursos](#). Estos niveles son:
- Uso correcto de URIs
- Uso correcto de HTTP.
- Implementar Hypermedia.

APIS y el Testing

- Además de estas tres reglas, **nunca se debe guardar estado en el servidor**, toda la información que se requiere para mostrar la información que se solicita debe estar en la consulta por parte del cliente.

APIS y el Testing

- **Nivel 1: Uso correcto de URIs (uniform resource identifier).**

Cuando desarrollamos una web o una aplicación web, las URLs nos permiten acceder a cada uno de las páginas, secciones o documentos del sitio web.

Cada página, información en una sección, archivo, cuando hablamos de REST, los nombramos como **recursos**.

El recurso por lo tanto es la información a la que queremos acceder o que queremos modificar o borrar,** independientemente de su formato**.

APIS y el Testing

- Nivel 1: Uso correcto de URIs (uniform resource identifier).

Las URL, Uniform Resource Locator , son un tipo de URI, Uniform Resource Identifier, que además de permitir identificar de forma única el recurso, nos permite localizarlo para poder acceder a él o compartir su ubicación.

Una URL se estructura de la siguiente forma:

{protocolo}://{dominio o hostname}[:puerto (opcional)]/{ruta del recurso}?{consulta de filtrado}

`https://petstore.swagger.io/v2/pet/100`

APIS y el Testing

- **Nivel 1: Uso correcto de URIs**

- Los nombres de URI no deben implicar una acción, por lo tanto debe evitarse usar verbos en ellos.
- **Ejemplo de URI incorrecta:** /user/4/edit.
- **Ejemplo de URI correcta:** /user/4 y lo que queremos hacer se indica con el método HTTP como veremos más adelante.
- Deben ser únicas, no debemos tener más de una URI para identificar un mismo recurso.
- Deben ser independiente de formato.
- **Ejemplo de una URI incorrecta** /user/4.pdf.

APIS y el Testing

Nivel 1: Uso correcto de URIs

- Deben mantener una jerarquía lógica.
- Ejemplo de una URI incorrecta: `/user/4//1`.
- ejemplo de una URI correcta: `/organization/1/user/4`
- Los filtrados de información de un recurso no se hacen en la URI.
- Ejemplo de una URI incorrecta: `/users/orden/desc` y ejemplo de una URI correcta: `/users?orderby=organization-DESC`

APIS y el Testing


- Nivel 2: Uso correcto de HTTP


Para desarrollar APIs REST los aspectos claves que hay que dominar y tener claros son:

- Utilizar los códigos de respuesta nativos de HTTP (200,204,409,404...) para informar si la petición se ha realizado correctamente.
- Métodos HTTP
- Aceptación de tipos de contenido

APIS y el Testing

- Nivel 2: Uso correcto de HTTP
 - Utilizar los códigos de respuesta nativos de HTTP (200,204,409,404...) para informar si la petición se ha realizado correctamente.

HTTP STATUS CODE 



Code	Message
200	OK
301	Moved Permanently
302	Moved Temporarily
404	Not Found
500	Internal Server Error
503	Service Unavailable

404
PAGE NOT
FOUND

Un código de estado es una parte de la respuesta devuelta por el servidor cuando un cliente (por ejemplo, un navegador) llama a una URL.

Con la ayuda de un código de estado, el servidor indica al cliente si la solicitud (por ejemplo, "envíame la página www.newyorker.com") se ha procesado correctamente o si se ha producido un error.

APIS y el Testing

- **Nivel 2: Uso correcto de HTTP**

- Usar los métodos HTTP para indicar el tipo de acción que queremos realizar con el recurso.
- GET: Para consultar y leer recursos
- POST: Para crear recursos
- PUT: Para editar recursos
- DELETE: Para eliminar recursos.
- PATCH: Para editar partes concretas de un recurso.

APIS y el Testing

- Nivel 2: Uso correcto de HTTP

Aceptación de tipos de contenido

REST es muy flexible y permite transmitir prácticamente cualquier tipo de datos, definiéndolo en el Header Content-Type, lo que nos permite mandar, XML, JSON, Binarios (imágenes, documentos), Text, etc.

Sin embargo, el tipo de dato que se suele usar en las APIs suele JSON o XML. En REST generalmente se usa JSON, seguramente porque es interpretado de manera natural por JavaScript.

APIS y el Testing

- Nivel 3: Implementar Hypermedia.

A pesar de lo que nos pueda inducir a pensar el término retrofuturista Hypermedia, el concepto y la finalidad que busca describir es bastante sencillo:** conectar mediante vínculos las aplicaciones clientes con las APIs**, permitiendo a dichos clientes despreocuparse por conocer de antemano del cómo acceder a los recursos.

Con Hypermedia básicamente añadimos información extra al recurso sobre su conexión a otros recursos relacionados con él.

APIs y el Testing

Nivel 3: Implementar Hypermedia. Ejemplo

```
<pedido>
  <id>666</id>
  <estado>Procesado</estado>
  <links>
    <link rel="factura">
      http://example.com/api/pedido/666/factura
    </link>
  </links>
</pedido>
```

Herramientas a utilizar

- Lenguaje: Python 3.4 o superior.
- Librerías:
 1. Pytest,
 2. Unittest,
 3. [Requests](#),
 4. Re (Regular Expressions),
 5. Time,
 6. json.
 7. Allure (para reportes)

Fuentes

- ¿Qué son las APIs y para qué sirven?

<https://www.youtube.com/watch?v=u2Ms34GE14U>

<https://ed.team/blog/que-son-las-apis-y-para-que-sirven>