Realizado Por  
Wilder Valencia Ocampo

Contenido

[API (application programming interface) 1](#_Toc37005652)

[API-Raspberry 1](#_Toc37005653)

[API REST 2](#_Toc37005654)

[CRUD 2](#_Toc37005655)

[Método de autentificación 3](#_Toc37005656)

[API Key 3](#_Toc37005657)

[Token 3](#_Toc37005658)

[JSON Web Token 4](#_Toc37005659)

[OAuth 2 4](#_Toc37005660)

[Automatic API Rest 6](#_Toc37005661)

[Instalación 6](#_Toc37005662)

[Mas información y Ejemplo 6](#_Toc37005663)

[Creación de una API REST con Python and Flask 7](#_Toc37005664)

[Proyectos con API REST 7](#_Toc37005665)

[Lecturas Recomendadas 8](#_Toc37005666)

[Manual de desarrollo de APIs 8](#_Toc37005667)

[Arquitectura de una API REST 8](#_Toc37005668)

[Automatic API Rest: Interactúa con tu base de datos en cualquier plataforma 8](#_Toc37005669)

[API REST 8](#_Toc37005670)

[Referencias 9](#_Toc37005671)

# API (application programming interface)

Definición: es un conjunto de subrutinas, funciones y procedimientos (o métodos, en la programación orientada a objetos) que ofrece cierta biblioteca para ser utilizado por otro software como una capa de abstracción.

Las API forman el pegamento de conexión entre las aplicaciones modernas. Casi todas las aplicaciones utilizan APIs para conectarse con fuentes de datos corporativas, servicios de datos de terceros u otras aplicaciones. La creación de un formato de descripción abierto para los servicios de API que sea neutral para los proveedores, portátil y abierto es fundamental para acelerar la visión de un mundo verdaderamente conectado.

API-Raspberry

El «boom» de IoT es reciente y, diferentemente de sistemas convencionales, en esos dispositivos existe la preocupación de consumo de energía y banda, limitación de memoria etc. y todavía hay poca discusión sobre arquitecturas que permitan evolucionar de manera ágil, segura y escalable esas aplicaciones en el mundo IoT, siendo común casos de vulnerabilidades en sistemas críticos como equipos de vigilancia, sensores e incluso monitor cardíaco de bebés. Es esencial durante el desarrollo preocuparse con esos temas y una API REST puede ser una solución ideal.

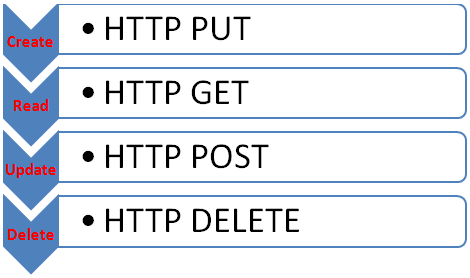
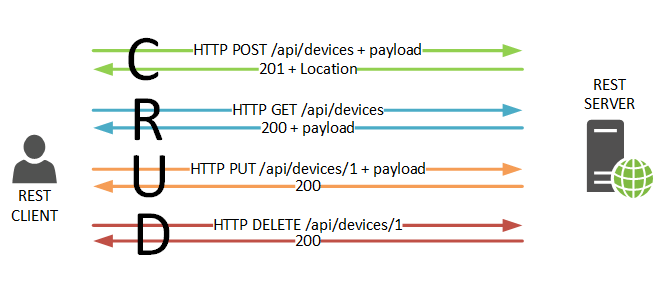
Al utilizar REST, existe una flexibilidad mayor de incrementar nuevas features, facilidad de exponer el sistema a diferentes plataformas (los clientes pueden ser apps, otros dispositivos IoT, navegadores etc) y, si utilizado correctamente y reduce la complejidad del sistema.

# API REST

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.

REST (REpresentational State Transfer): es un tipo de arquitectura de desarrollo web que se apoya totalmente en el estándar HTTP. REST nos permite crear servicios y aplicaciones que pueden ser usadas por cualquier dispositivo o cliente que entienda HTTP, por lo que es increíblemente más simple y convencional que otras alternativas como SOAP y XML-RPC. REST se definió en el 2000 por Roy Fielding, coautor principal también de la especificación HTTP. Podríamos considerar REST como un framework para construir aplicaciones web respetando HTTP.

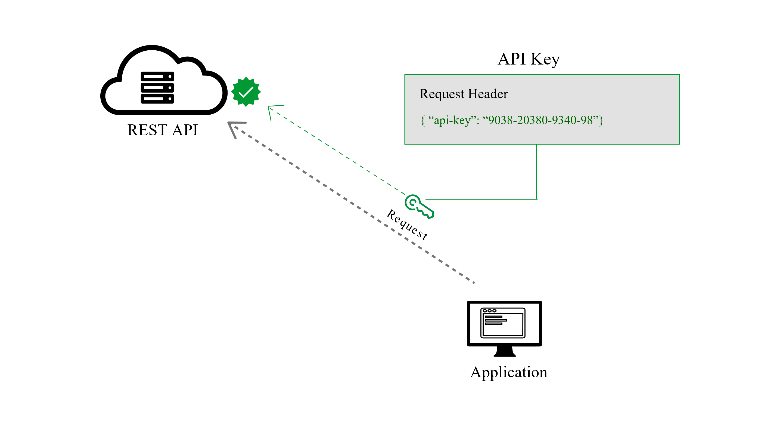
## CRUD



## Método de autentificación

### API Key

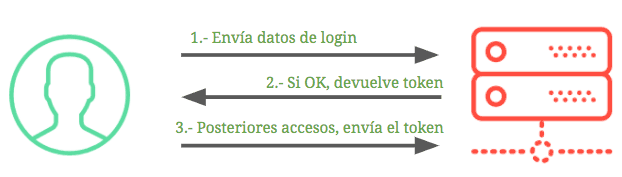
Las API Keys son el primer método que apareció para realizar accesos a este tipo de servicios. Trae aparejados dos grandes problemas que son: su falta de seguridad y la pésima experiencia de usuario. En muchos casos, estas claves son enviadas dentro de la URL en donde se realiza la petición al servicio, lo que permite ser accedida por alguien que no debería. Para esto el estándar nos dice que las keys deberían viajar en el Header de la petición, para dificultar un poco su acceso por parte de terceros, aunque en la práctica se pueden ver en el Header Authorization, en un Header personalizado o en la URL.



### Token

Un token es un valor que nos autentica en el servidor, Normalmente se consigue después de hacer login mediante usuario/contraseña.Un token se genera normalmente como un hash calculado con algún dato (p.ej. login del usuario + clave secreta). Además el token puede llevar datos adicionales como el login.

Cada vez que el cliente deba realizar nuevas solicitudes al servidor, certificando que es un usuario correctamente autenticado, tendrá que enviar el token de nuevo al servidor. Por lo tanto, una vez recibido el token, deberá ser almacenado del lado del cliente para enviarlo con las posteriores solicitudes. En estos casos, el token suele viajar en las cabeceras del HTTP, de modo que llegue al servidor.



El servidor que recibe el token tiene la capacidad de desencriptarlo, de modo que pueda comprobar qué usuario es el que está realizando esta solicitud. Durante el proceso de decodificación del token, el servidor puede comprobar si este es válido y si resulta serlo, puede recuperar toda la información encriptada en el mismo, que suele ser al menos la referencia inequívoca del usuario involucrado. Por supuesto, si en cualquier momento se detecta que el token no es correcto, se obligará al usuario a autenticarse nuevamente.

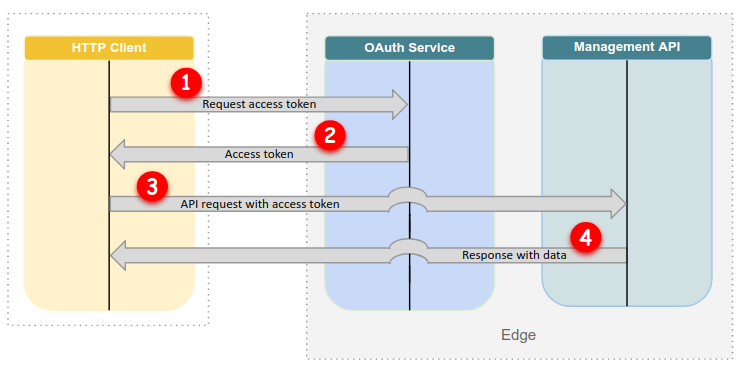
### JSON Web Token

Este tipo de token es mucho más largo que los demás ya que depende de la cantidad de datos que queramos almacenar en él. Sin embargo, una buena utilización del mismo nos permitiría mantener un token que se puede enviar por la red sin ocupar mucho espacio.

Este token está compuesto por varios campos codificados en base64 donde se guarda la información, identificación de usuario, alcance de los permisos del token, y finalmente una firma de todo los datos contenidos del token hecha por el servidor de la API para comprobar que el token es válido. Esta comprobación es mucho más eficiente y rápida que el acceso a la base de datos para determinar a quién pertenece, qué permisos tiene, etc.

### OAuth 2

OAuth es un framework que permite delegar la autorización de acceso a las APIs, NO es un protocolo de autenticación. Para que una app pueda acceder a servicios de terceros sin que el usuario tenga que darle a la app sus credenciales del servicio. Se basa en el uso de un token de sesión.



Este método es el mejor para las APIs que contengan información de usuarios. ¿Por qué? Principalmente por ser la más robusta de las tres alternativas y por cómo está definido el protocolo OAuth nos posibilita declarar permisos para cada aplicación que quiere acceder a la API y así tener varios tipos de autorización para apps diferentes.

# Automatic API Rest

Automatic API Rest nace de la necesidad de gestionar los datos entre distintas plataformas de manera sencilla. Una de las cosas que más pereza me daba a la hora de crear una base de datos es tener que crear todas las clases y funciones (Normalmente en PHP) para dar servicio externo (Android, iOS, Python,etc) a la base de datos.

Automatic API Rest es una herramienta Open Source bajo licencia Apache 2.0 que te permite crear una completa API de tu base de datos MySQL (Por ahora sólo tiene soporte para MySQL o MariaDB).

## Instalación

A continuación se describen los pasos a seguir para la instalación:

* Descargar Automatic API Rest desde aquí. (<http://goo.gl/7kPWWP>)
* Extraer el zip y subir todo el directorio a la raíz principal de vuestro servidor
* Abrir el archivo config.php y rellenar todos los campos con los credenciales de la base de datos.

## Mas información y Ejemplo

<https://geekytheory.com/automatic-api-rest>

# Creación de una API REST con Python and Flask

<https://www.youtube.com/watch?v=s_ht4AKnWZg>

# Proyectos con API REST

<https://riunet.upv.es/bitstream/handle/10251/107770/BIANCHINI%20-%20Desarrollo%20de%20un%20API%20REST%20para%20transmisión%20de%20datos%20de%20sensores%20GPS.pdf?sequence=1&isAllowed=y>

# Lecturas Recomendadas

## Manual de desarrollo de APIs

<https://desarrolloweb.com/manuales/manual-desarrollo-api.html>

## Arquitectura de una API REST

<https://juanda.gitbooks.io/webapps/content/api/arquitectura-api-rest.html>

## Automatic API Rest: Interactúa con tu base de datos en cualquier plataforma

<https://geekytheory.com/automatic-api-rest>

## API REST

<https://aprendiendoarduino.wordpress.com/2019/10/27/api-rest/>

# Referencias

<https://sensedia.com/es/apis/apis-para-dispositivos-iot/>

<https://riunet.upv.es/bitstream/handle/10251/107770/BIANCHINI%20-%20Desarrollo%20de%20un%20API%20REST%20para%20transmisión%20de%20datos%20de%20sensores%20GPS.pdf?sequence=1&isAllowed=y>

<https://desarrolloweb.com/articulos/autenticacion-token.html>

<https://aprendiendoarduino.wordpress.com/2019/10/27/api-rest/>

<https://geekytheory.com/automatic-api-rest>

<https://desarrolloweb.com/manuales/manual-desarrollo-api.html>

<https://juanda.gitbooks.io/webapps/content/api/arquitectura-api-rest.html>

<https://www.youtube.com/watch?v=s_ht4AKnWZg>