

Google APIs

Grupo 1

- Renato Josue Flores Perez, **201709244**
- Lester Fernando Mazariegos Navarro, **201403610**
- Jorge Luis Salazar Peralta, **201404215**
- Astrid Edith Hernández González, **201213223**
- Bryan Eduardo Chacón López, **201504290**
- Wendy Aracely Chamalé Boch, **201504284**
- María de Los Angeles Herrera Sumalé, **201504399**

Que son las APIs de Google?

Las APIs de Google son interfaces de programacion desarrolladas por Google que hacen posible la comunicacion con los servicios de Google y en consecuencia hacen poible la integracion con otros servicios o aplicaciones.

- Sheets
- Docs
- Maps
- Drive

Para ver la lista completa de APIs disponibles pueden visitar la **Biblioteca de APIs disponibles**

REST

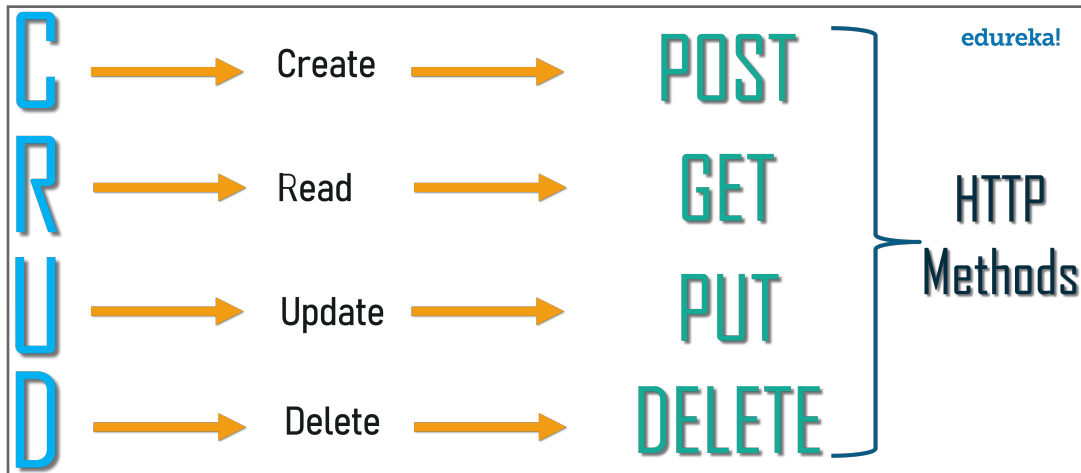
Las APIs de Google son APIs REST estandar.
Esto significa que se componen de:

- Metodo
- URL base
- Resource path
- Body
- Response

Esto a su vez implica que en teoria pueden ser accesadas desde cualquier dispositivo y cualquier aplicacion capaz de hacer una http request.

Metodo

El metodo en una REST API indica que tipo de operacion se desea realizar. Existen 4 estandar.



URL base

La URL base de una API REST indica el endpoint al cual se va a realizar la petición. La URL base indica la parte constante en todas las solicitudes que se realizan a la API. En nuestro caso, esta URL apunta a los servidores oficiales de Google y cambia dependiendo de que servicio se este solicitando.

Un ejemplo seria la URL base de Google's Spreadsheets:

<https://sheets.googleapis.com/v4>

Resource path

El resource path es la parte variable en la petición, e indica que recurso se desea solicitar. Un ejemplo seria el path para recibir una Overview de una hoja de calculo almacenada en Google Sheets:

/spreadsheets/{spreadsheetId}

Los resource paths por lo general requieren uno o mas parametros que se pueden especificar directamente en la URL.

En este caso, la ultima parte **{spreadsheetId}** es un URL parameter, que debe ser sustituido por el ID de interes.

Body

```
{
  "majorDimension": "ROWS",
  "values": [
    [
      "Hello",
      "World"
    ],
    [
      "Practicas",
      "Intermedias"
    ],
    [
      "Google",
      "APIs"
    ]
  ]
}
```

Response

```
{
  "config": {
    "url":
      "https://sheets.googleapis.com/v4/spreadsheets/19NV-XzX7KdZd1TfiapXlZGQpjNnLNo4oTe_6jgguIV
      E/values/A2%3AD501?dateTimeRenderOption=FORMAT
      TED_STRING&majorDimension=ROWS&
      valueRenderOption=FORMATTED_VALUE",
    "method": "GET",
    "userAgentDirectives": [
      {
        "product": "google-api-nodejs-
        client",
        "version": "4.4.0",
        "comment": "gzip"
      }
    ],
    "headers": {
      "x-goog-api-client": "gdcl/4.4.0
      gl-node/10.19.0 auth/6.0.6",
      "Accept-Encoding": "gzip",
      "User-Agent": "google-api-
      nodejs-client/4.4.0 (gzip)",
      "Authorization": "Bearer
      ya29.a0AfH6SMAZeb-
```



```

ce_6Jpgi0jud3aI6QcMagwJDZd_afGASeelB7w
CusH3B1arwu3AbEc_844Fz9eQhNIXUuCq-
AjUVdIMhvhbb8tIgXkI06t0rlQdJd8NxI30ByX
2u0tg3-t8BtI92n-
j-YQQm4lM0pLZrT7CGMChgzSqTqIw",
    "Accept": "application/json"
  },
  "params": {
    "dateTimeRenderOption":
"FORMATTED_STRING",
    "majorDimension": "ROWS",
    "valueRenderOption":
"FORMATTED_VALUE"
  },
  "retry": true,
  "responseType": "json"
},
"data": {
  "range": "'Hoja 1'!A2:D501",
  "majorDimension": "ROWS",
  "values": [
    ["2013045278", "Ana Elizabeth",
"IPC1", "13"],
    ["201502458", "Aaron Burr",
"Practicas Intermedias", "55"],
    ["2013045278", "Ana Elizabeth",
"IPC1", "13"],
    ["201502458", "Aaron Burr",
"Practicas Intermedias", "55"],
    ["2013045278", "Ana Elizabeth",
"IPC1", "13"],
    ["245112", "Alex Flores",

```

```

"IPC2", "22"]
    ],
    "headers": {
        "alt-svc": "h3-29=\":443\";
ma=2592000,h3-27=\":443\";
ma=2592000,h3-T051=\":443\";
ma=2592000,h3-T050=\":443\";
ma=2592000,h3-Q050=\":443\";
ma=2592000,h3-Q046=\":443\";
ma=2592000,h3-Q043=\":443\";
ma=2592000,quic=\":443\"; ma=2592000;
v=\"46,43\"",
        "cache-control": "private",
        "connection": "close",
        "content-encoding": "gzip",
        "content-type": "application/json;
charset=UTF-8",
        "date": "Wed, 09 Sep 2020 19:56:28
GMT",
        "server": "ESF",
        "transfer-encoding": "chunked",
        "vary": "Origin, X-Origin,
Referer",
        "x-frame-options": "SAMEORIGIN",
        "x-xss-protection": "0"
    },
    "status": 200,
    "statusText": "OK",
    "request": {
        "responseURL":
"https://sheets.googleapis.com

```

```
/v4/spreadsheets/19NV-  
XzX7KdZd1TfiapXlZGQpjNnLNo4oTe_6jgguIV  
E/values  
/A2%3AD501?dateTimeRenderOption=FORMAT  
TED_STRING&majorDimension=ROWS&  
valueRenderOption=FORMATTED_VALUE"  
  }  
}
```

Partes de una respuesta

- **Config:** contiene metada sobre el tipo de solicitud, el seridor, la version y otra informacion sobre el tipo de solicitud realizada.
- **Data:** la informacion que solicitamos.
- **Headers:** headers de la respuesta.
- **Status:** codigo de estatus de la solicitud.
Siempre sera 200 si la solicitud fue exitosa.
- **StatusText:** Una pequeña descripcion del status de la respuesta. Por lo general es "OK" si todo fue bien.
- **Request :** Indica el respurce path utilizado para realizar esta request.

Autenticacion

Si bien es cierto que las APIs de Google pueden ser accesadas desde basicamente cualquier dispositivo con una conexion a internet, su consumo en realidad no es tan simple. **TODAS** Las apis de google requieren de una autenticacion por parte del usuario para poder utilizarlas.

Los procesos de autenticacion disponibles incluyen:

- API Keys
- Personal Tokens
- Service Accounts

API Keys

Una API key es la forma mas simple de autenticacion.

Ventajas

Desventajas

Facil de generar

Solo pueden accesar a informacion publica

Facil de utilizar

Solo pueden realizar operaciones de solo lectura

Se pueden utilizar desde cualquier cliente

No todas las APIs admiten autenticacion por API keys

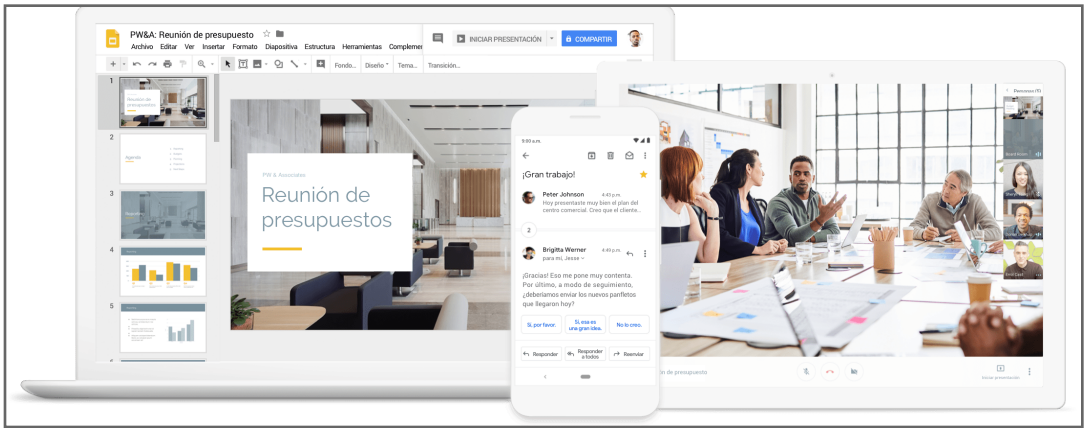
OAuth2 personal authentication

1. Crear un proyecto de desarrollador de Google.
2. Habilitar las APIs a utilizar por el proyecto.
3. Crear un proyecto local en el lenguaje de eleccion.
4. Descargar e instalar las librerias para autenticacion con OAuth2.
5. Descargar las credenciales de autenticacion de la consola de desarrollo de Google.
6. Utilizar la libreria de OAuth2 y las credenciales descargadas para comenzar el proceso de autenticacion.
7. Aprobar el acceso de sus recursos a la aplicacion.
8. Utilizar la API

Lenguajes soportados

- Browser Javascript
- Go
- Java
- Node
- PHP
- Python
- Ruby
- .NET
- Android SDK
- Objective-C

Service Accounts



Scopes

El Scope (Alcance) indica de forma especifica a que recurso (API) se tiene acceso y a que permisos (leer, editar) dentro de esa API se tiene acceso. Un Scope se indica como un URL al momento de realizar la autenticacion inicial.

- <https://www.googleapis.com/auth/drive.readonly>
 - <https://www.googleapis.com/auth/spreadsheets.readonly>
 - <https://www.googleapis.com/auth/drive.file>
 - <https://www.googleapis.com/auth/spreadsheets>
 - <https://www.googleapis.com/auth/admin.directory.resource.calendar>
- [Lista entera de Scopes y su descripcion](#)

Estructura de una Google API

RECURSOS

Todas las APIs de google se dividen en una serie de recursos. Estos definen agrupaciones logicas de operaciones que es posible realizar en la API. Estos recursos a su vez se dividen en metodos.

METODOS

Los metodos dentro de una API de google indica que operacion especifica se desea realizar. Sin embargo estos no son metodos REST. Es simplemente la parte final del resource path entero. El resource path final se compone del endpoint (URL base) + resource + metodo. Aparte esta el REST method, los URL parameters y el body.

googleapis library

1. Conectarse y Autenticarse
2. Crear el objeto cliente que representa la API
3. Identificar el recurso al que se desea acceder
4. Utilizar el metodo especifico
5. Enviar el body esperado de acuerdo al metodo.
6. Enviar un callback
7. Procesar la informacion devuelta

Estructura de una instancia de una Google API

```
gAPIinstance = {  
  recurso1 : {  
    metodo1 : function  
    (body,callback){  
      // ... process internal API call  
      ...  
      callback(error,response);  
    },  
    metodo2 : function  
    (body,callback){},  
    metodoN : function  
    (body,callback){}  
  },  
  recurso2 : { /* ... */ },  
  recursoN : { /* ... */ }  
};
```

Callbacks & Promises

```
function metodo1(body){
  return new Promise((resolve, reject)
=> {

gAPIinstance.recurso1.metodo1(body,
(err, res) => {
    if (err)
reject('The API returned an error: ' +
err);
    resolve(res);
  });
});
}
```