



Facebook

# Uso de Graph API

Diego Sánchez

CIF 9159-1 TALLER INTEGRADO

# Graph API

- Graph API es la forma principal de ingresar y obtener datos de la plataforma de Facebook. Es una API basada en HTTP que las aplicaciones pueden usar para consultar datos , publicar nuevas historias, administrar anuncios, cargar fotos y realizar una amplia variedad de otras tareas.

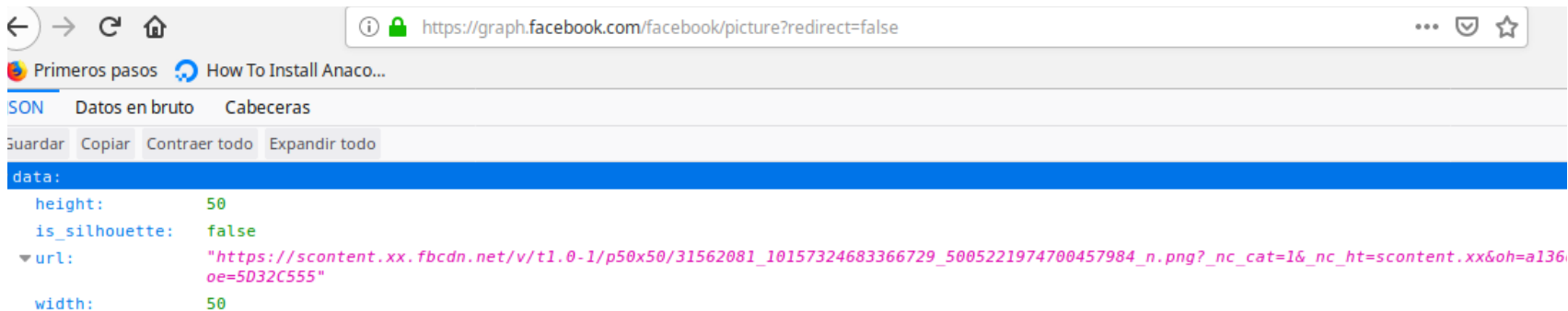
<https://developers.facebook.com/docs/graph-api/overview/>

# Graph API

- está basada en HTTP, por lo que funciona con cualquier idioma que tenga una biblioteca HTTP, como cURL y urllib.

```
curl -i -X GET \  
  "https://graph.facebook.com/facebook/picture  
  ?redirect=false"
```

Como ejemplo en la imagen anterior, con esa url hace una consulta a la API para solicitar una imagen y este entrega un json con la información para obtenerla



# Graph API

**Su USO ES GRATIS PERO**

## **Limitación de frecuencia de uso de la app**

define límites sobre la cantidad de llamadas que pueden realizarse a la API, el número total de llamadas que puede realizar la app por hora es 200 veces el número por usuario.

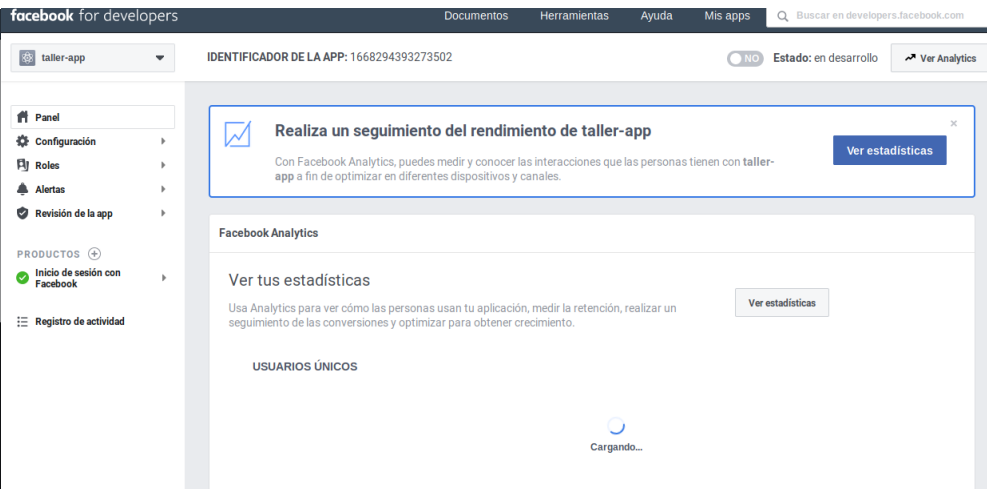
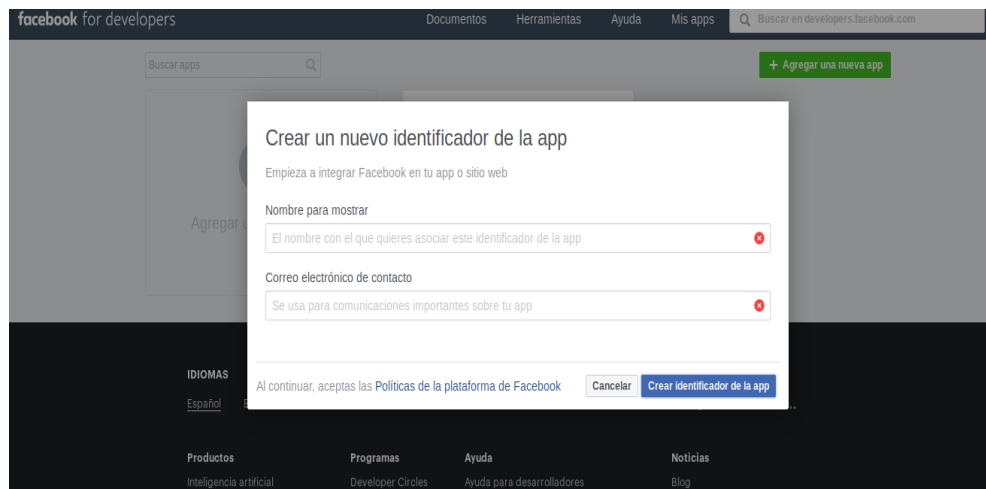
<https://developers.facebook.com/docs/graph-api/advanced/rate-limiting#user-level-rate-limiting>

<https://developers.facebook.com/docs/graph-api/advanced/rate-limiting#application-level-rate-limiting>

# Graph API uso

Para este ejemplo obtendremos datos de usuarios de Facebook que nos conceden el permiso.

Para esto crearemos un identificador de app en Facebook developers.

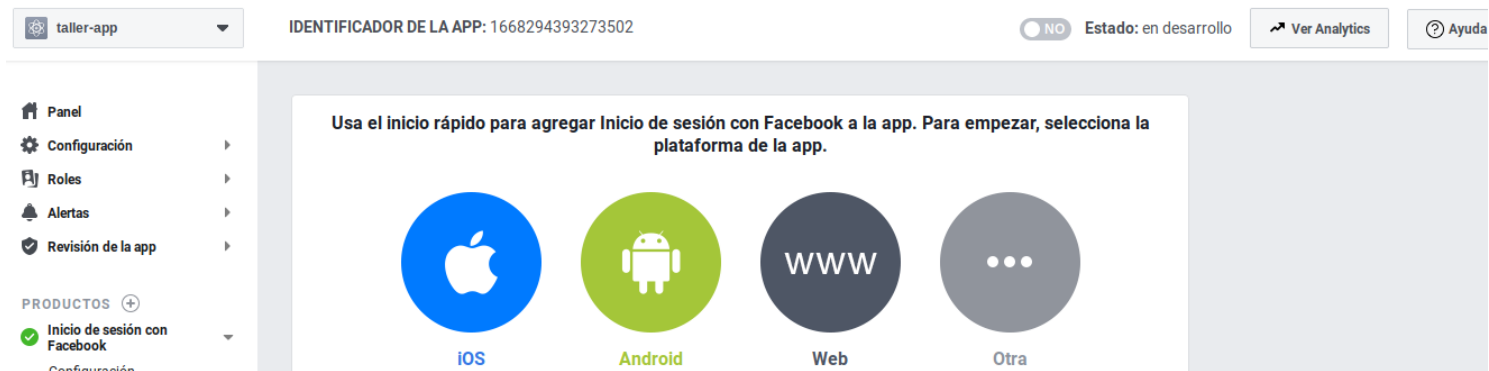


# Graph API uso

Para que la API nos entregue los datos, se necesita un token de acceso

```
curl -i -X GET \  
  "https://graph.facebook.com/object-id  
  ?access_token=your-access-token"
```

Para obtener el token de acceso se utilizo el sdk de Facebook para que el usuario de permisos a la aplicación.



# Graph API uso

Utilizando el sdk de javascript para login de facebook se logra que el usuario nos de permisos y obtener el token de acceso.

```
}  
  
window.fbAsyncInit = function() {  
  FB.init({  
    appId      : '{1668294393273502}', ← Id de la aplicación  
    cookie     : true, |  
  
    xfbml      : true,  
    version    : '{v3.2}' ← Versión de Graph API  
  });  
};
```

El código javascript se coloca en los templates de Flask, en este ejemplo fue en “home.html” el demas codigo del sdk que fue usado se obtiene de: [link de sdk javascript facebook](#)

```
});  
FB.getLoginStatus(function(response) {  
  if (response.status === 'connected') {  
  
    //obtengo el token del usuario  
    var x = response.authResponse.accessToken;  
    //envio token por la url  
    location.href='http://localhost:5000/'+x;
```

Ahora a la variable “x” se asigno el valor del token del usuario por el evento  
“response.authResponse.accessToken”  
Y después redirecciono con la url generada que contiene el token de acceso.

# Obtención de datos con python

```
ex.py
from flask import Flask, render_template
import requests
import json
app = Flask(__name__)
```

Primero se importan la librerías del framework Flask, requests para obtener el json con los datos y json para manipular el json obtenido.

```
@app.route("/")
@app.route('/<token>')
def home(token=None):
    if token is None:
        return render_template("home.html")
    else:
        link = 'https://graph.facebook.com/v3.2/me?fields=id%2Cname%2Cage_range%2C'
        dato = requests.get(link)
        dt = json.loads(dato.content.decode("utf-8"))
        return render_template("datos.html", name=dt)
    pass
pass
if __name__ == '__main__':
```

A través de “@app.route('/<token>')” la función “home” obtiene el token que se envió por la url.

En la variable “link” se genera una url con los datos que nosotros especifiquemos más el token obtenido.

[url lista de datos](#)

En la variable “datos” obtiene el json con los datos obtenidos por la url, después en la variable “dt” carga el json y finalmente es enviado a la templates “datos.html” con el nombre “name”.



# Visualización de datos

```
</script>
</head>
<body class="er">
  <div class="container">
    <div class="row">
      <div class="col-md-4"></div>
      <div class="col-md-4 mt-3 mb-3">
        <div class="card text-center">
          
          <div class="card-body">
            <h5 class="card-title">{{ name.name }}</h5>
            <p class="card-text">{{ name.id }}</p>
          </div>
          <ul class="list-group list-group-flush">
            <li class="list-group-item">{{ name.age_range.min }}</li>
            <li class="list-group-item">{{ name.email }}</li>
            <li class="list-group-item">{{ name.gender }}</li>
            <li class="list-group-item">{{ name.birthday }}</li>
          </ul>
        </div>
      </div>
      <div class="col-md-4"></div>
    </div>
  </div>
</body>
</html>
```

A través de esta pagina web se va mostrando los distintos datos que obtuvieron al usar la API.

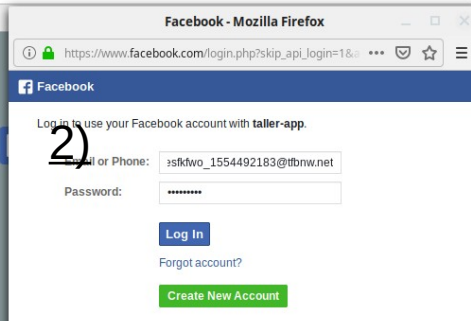
# FINAL

1)



Log Out

2)



3)



← foto perfil

```
id: "103565500834578"
name: "Isabella Alcedcchdhded Putnamescu"
age_range:
  max: 20
  min: 18
email: "wsiesfkfwo_1554492183@tfbnw.net"
gender: "female"
birthday: "04/09/2000"
picture:
  data:
    height: 50
    is_silhouette: false
    url: "https://platform-lookaside.fbsbx.com/platform/profilepic/?asid=103565500834578&he
    width: 50
```

Isabella Alcedcchdhded  
Putnamescu

← Nombre

103565500834578

← id

18

← Edad

wsiesfkfwo\_1554492183@tfbnw.net

← Email

female

← Genero

04/09/2000

← Fecha de cumpleaños

Link: [codigo en githud](#)