



Práctica de Implementación de OAuth y OpenID Connect

En este laboratorio, exploraremos la implementación de dos protocolos fundamentales de seguridad y autenticación en APIs: OAuth 2.0 y OpenID Connect. En la primera parte de este trabajo, nos centraremos en el flujo de autorización de OAuth 2.0, conocido como 'Authorization Code Flow'. A través de este flujo, se obtendrá autorización para acceder a recursos protegidos en nombre de nuestro usuario de Google, en este caso, obtendremos la información del correo electrónico. La segunda parte del trabajo tendrá como objetivo la obtención de un Token JWT utilizando el protocolo OpenID Connect, que agrega una capa de autenticación a OAuth 2.0

Indice

Flujo de Autorización OAuth2.0.....	2
1. Crear Cliente OAuth 2.0 en el servicio de autorización de Google.....	2
2. Generar código de autorización.....	4
3. Obtener Token de Acceso.....	6
4. Obtener la información del correo electrónico de google	6
Autenticación con OpenID Connect.....	7
1. Configurar Cliente OAuth 2.0 en el servicio de autorización	7
2. Obtener el token JWT.....	7
3. Validar el token JWT.....	8



Acceder al Laboratorio de Oauth

Ingresa al siguiente [link](#). Debería observar la siguiente página:



(Opcional) Instalación de la aplicación Lab Oauth

En el primer inciso, se realizará la instalación y puesta en marcha de la aplicación para el laboratorio de OAuth y OpenID. Previamente, es necesario tener descargado **Docker** y, en caso de encontrarse en Windows, tener **Docker Desktop** en ejecución.

En primer lugar, se deberá clonar el repositorio [Oauth Lab](#) y luego, estando ubicado dentro de la carpeta "OauthLab", ejecutar el comando "docker-compose build".

Finalmente, acceda en el navegador a la URL "localhost:5000" donde podrá observar la aplicación funcionando.





Flujo de Autorización Oauth2.0

En este inciso, se simulará el flujo de autenticación conocido como 'Authorization Code' en OAuth 2.0. En primer lugar, se crea un cliente en el servicio de autorización OAuth de Google. Luego se obtiene un código de autorización para acceder a información protegida en nombre de nuestro usuario de Google. A continuación, se procede intercambiándolo por un token de acceso y, finalmente, se obtiene la información del perfil del usuario de Google a través de su API.

1. Crear Cliente OAuth 2.0 en el servicio de autorización de Google

Para lograr que una aplicación acceda a las APIs de Google en representación de un usuario utilizando el protocolo de autorización OAuth, es esencial que se cuente con credenciales de autorización que permitan la identificación de la aplicación ante el servicio de autorización OAuth 2.0 de Google.

Para lograrlo se deben seguir los siguientes pasos:

a. Ingresar a la pagina [Apis y Servicios Google](#) y crear un nuevo proyecto:

APIs y servicios

APIs y servicios habilitados

Para ver esta página, selecciona un proyecto. [CREAR PROYECTO](#)

Nombre del proyecto *
Oauth

ID de proyecto: sylvan-arcadia-403818. No se podrá cambiar más tarde. [EDITAR](#)

Organización *
fi.uba.ar

Selecciona una organización para vincularla a un proyecto. No podrás cambiar esta selección más adelante.

Ubicación *
fi.uba.ar [EXPLORAR](#)

Organización o carpeta superior

[CREAR](#) [CANCELAR](#)



b. Luego, ingresar a la vista de [pantalla de consentimiento](#) y completar los siguientes campos:

- **User type:** Externos
- **Nombre de la Aplicación:** OAuth
- **Correo de asistencia e Información de contacto:** mail de la cuenta de google

c. Finalmente, se debe acceder a [Credenciales](#) → Crear Credenciales → ID de cliente de OAuth, seleccionar como tipo de aplicación “Aplicacion web” y, por ultimo, en url de redirección autorizadas ingresar la siguiente url: <http://oauthlab.zapto.org/auth/oauth/callback>

Tipo de aplicación *
Aplicación web

Nombre *
Cliente web 1

URI de redireccionamiento autorizados ?

Para usar con solicitudes de un servidor web

URI 1 *
<https://oauthdebugger.com/debug>

+ AGREGAR URI

Al terminar de crear el cliente de OAuth, se muestra un pop-out como el siguiente. **Es importante anotarse el valor del secret id ya que no podrá ser visto a posteriori:**

Se creó el cliente de OAuth

Puedes acceder al ID de cliente y el secreto desde “Credenciales” en API y servicios

i El acceso OAuth está restringido a los [usuarios de prueba](#) que aparecen en la [pantalla de consentimiento de OAuth](#)

ID de cliente	616079903194-cfmng6r8rst4tl525qb4d2o313efho5r.apps.googleusercontent.com
Secreto del cliente	GOCSPX-3ZaUiczszy0pPjS0hFn1jUZ83mz
Fecha de creación	1 de noviembre de 2023, 16:25:15 GMT-3
Estado	✓ Habilitada



2. Generar código de autorización

En el navegador, acceder a la url "localhost:5000" donde podrá observar la aplicación funcionando y presionar la opción de **"OAuth Code Flow"** para iniciar el flujo de OAuth conocido como "authentication code flow".

Luego, ingresar los parametros correspondientes pedidos por la aplicación y presionar **review request**:

- **Authorize URI:** <https://accounts.google.com/o/oauth2/v2/auth>
- **Token Exchange URI:** <https://accounts.google.com/o/oauth2/token>
- **Redirect URI:** <http://oauthlab.zapto.org/auth/oauth/callback>
- **Client ID:** Client ID obtenido en la sección 1.c
- **Scope:** profile



8636 Criptografía y Seguridad Informática

Seguridad en Arquitectura de Servicios

Authorization URI:

<https://accounts.google.com/o/oauth2/v2/auth>

Redirect URI:

<http://localhost:5000/auth/oauth/callback>

Token exchange URI:

<https://accounts.google.com/o/oauth2/token>

Client ID:

673704949781-0oa69b9pomqknm8j4poehbij938cb8mq.app

Scope:

profile

1) ¿Con que objetivo se utiliza la URI de redirección?

2) ¿Qué ventaja se obtiene al utilizar el scope?




Una vez enviado el request, el servicio de autenticación de google le solicitará que ingrese las credenciales de su cuenta de google para continuar:



Una vez ingresadas, se le pedirá permisos para realizar ciertas acciones en específico:

Esto permitirá a **oauthClaseCripto** hacer lo siguiente:

-  Consultar tu información personal, incluida la que has compartido públicamente

3) ¿De qué parámetro del request obtiene el servicio de Google las acciones que el cliente debe autorizar?



Finalmente, luego de darle permitir, se obtendra un código de autorización

OAuth Code Flow Response

Authorization Code:

```
4/0AdLIrYeVZP6G1i41o1w6awW46qwHhPM-  
LF22gWqHPbJz2d2PDFMptwefX5pt5pq0aiwzg
```

Copy the authorization code to use it in the next step!

Scope: profile <https://www.googleapis.com/auth/userinfo.profile>

Original state: 22pfUSOtBDxFxAli0QTBkqmePkdCOR

Returned state: 22pfUSOtBDxFxAli0QTBkqmePkdCOR

**The callback state is identical to the one sent in the request!
You can continue with OAuth Code Flow.**

- 4) ¿Por qué se verifica que el estado enviado en la solicitud sea idéntico al obtenido en la respuesta?
- 5) ¿Qué beneficio se obtiene de utilizar OAuth para el acceso?
- 6) Explique los pasos del "Authorization Code Flow" que se realizaron en esta parte.
- 7) ¿Por qué se devuelve un código y no directamente el token de acceso? Si la respuesta fuera el token de acceso, ¿de qué tipo de flujo de autorización se trataría?



3. Obtener Token de Acceso

Una vez obtenido el código de autorización, es necesario realizar el intercambio por el token de acceso para poder obtener la información del usuario. Para lograrlo, se debe continuar con el flujo propuesto de la aplicación presionando **Continue with the exchange of authorization code for access token**.

Luego, se debe completar los datos pedidos por la aplicación y presionar **Exchange authorization code for access token**:

- **Authorization Code:** El obtenido en el paso anterior
- **Secret ID:** El obtenido en el paso 1.c



8636 Criptografía y Seguridad Informática

Seguridad en Arquitectura de Servicios

Authorization Code:

4/0AdLlrYfObw40xdb12cIN3-kB-ePbRtDFceNNIjAjd-RSKFi

Secret ID:

GOCSPX-JpS

Exchange authorization code for access token

Como respuesta a la solicitud, se obtendrá un token de acceso:

OAuth Code Flow Response

Access Token:

```
ya29.a0AXooCgsbyxQNm-  
cud7lBwpywrHXCssdqNPCBQzZ0iA7Ygh6r5zenk1vZYH70VH5Jxr3KzXTUayI9sQ7MH  
0Gr01UDjA8JpLvZpZHNaz2GLN3q4ArJy7uvTKYmDFX2U6KFbwfDnbkq71zkuwmUHFmb  
73gxXwzECrSNnVumaCgYKAQISARISFQHGx2Miyim70k99Q3Edh2Y42X_ng0171
```

8) ¿Para qué se utiliza el parametro 'Secret ID'?



4. Obtener la información del correo electrónico de google

Finalmente, una vez obtenido el token de acceso del cliente, se buscará obtener la información del perfil de Google. Con este objetivo, se debe realizar una solicitud GET al endpoint `oauth2/v2/userinfo`, indicando el token de acceso. Esto se logra presionando el botón **Continue to get user info** y completando los campos requeridos:

- **Access Token:** El obtenido en el paso anterior
- **Request user information uri:** <https://www.googleapis.com/oauth2/v2/userinfo>



8636 Criptografía y
Seguridad Informática

Seguridad en Arquitectura de Servicios

User Info



Pablo Peiretti (ppeiretti@fi.uba.ar)

Home

9) ¿Qué parámetro deberíamos modificar para obtener el email además de la información del perfil?

10) [Opcional] Realizar el mismo flujo con el cambio de la pregunta 9)



Autenticación con OpenID Connect

La segunda parte del trabajo tendrá como objetivo obtener un token a obtención de un Token JWT utilizando el protocolo OpenID Connect.

1. Obtener el token JWT

En el navegador, acceder nuevamente al laboratorio o presionar el botón de **Home** y presionar la opción de **Oauth OpenID Flow** para iniciar el flujo de Open ID.

Luego, ingresar los parametros correspondientes pedidos por la aplicación y presionar **review request**:

- **Authorize URI:** <https://accounts.google.com/o/oauth2/v2/auth>
- **Redirect URI:** <http://oauthlab.zapto.org/auth/oauth/callback>
- **Client ID:** Client ID obtenido en la sección 1.c
- **Scope:** openid

Finalmente, seguir con todo el flujo de la aplicación de la misma forma que en el inciso anterior.

- 1) *¿Qué ventaja se obtiene al utilizar OpenID respecto a Oauth?*
- 2) *¿Sobre qué acciones pide permiso el servicio de google al cliente en este caso?*

2. Validar el token JWT

Una vez obtenido el Token JWT, poner en el navegador la siguiente URL:
“https://oauth2.googleapis.com/tokeninfo?id_token=YOUR_ID_TOKEN”

- 3) *¿Qué información se obtiene del JWT? Explicar cada una de sus partes.*

Ahora ingrese a la URL <https://www.googleapis.com/oauth2/v3/certs>

- 4) *¿Qué son los valores e y n de la sección de ‘Verify Signature’?*
- 5) *¿Cómo es el proceso de validación del token?*



Práctica de Implementación de OAuth y OpenID