

Lab OAuth

Grupo: **G06**

Integrantes:

Nombre	Padrón
<i>Pedro Flynn</i>	<i>105742</i>
<i>Agustina Schmidt</i>	<i>103409</i>
<i>Agustina Fraccaro</i>	<i>103199</i>
<i>Kevin Gadacz</i>	<i>104531</i>
<i>Abraham Osco</i>	<i>102256</i>
<i>Ricardo Luizaga</i>	<i>87528</i>

1. Flujo de Autorización OAuth2.0

1.1. Crear cliente OAuth 2.0

Nombre del proyecto *

OAuth

GUARDAR

ID de proyecto
inner-melody-440716-f0

Número de proyecto:
108276432383

Ubicación

fi.uba.ar

Más información

Se creó el cliente de OAuth

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

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

ID de cliente	108276432383- arplo7p1i0kd1gi0thcbp29fopvai3s6.apps.g oogleusercontent.com
Secreto del cliente	GOCSPX-k7- 5A9tCwv1CTRU8CMKKKYES3fSo
Fecha de creación	4 de noviembre de 2024, 13:37:19 GMT-3
Estado	Habilitada

DESCARGAR JSON

ACEPTAR

ID de cliente: 108276432383-arplo7p1i0kd1gi0thcbp29fopvai3s6.apps.googleusercontent.com

Secreto del cliente: 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

1.2. Generar código de autorización



8636 Criptografía y Seguridad Informática

Seguridad en Arquitectura de Servicios

Oauth Code Flow Request

Authorization Uri: `accounts.google.com/o/oauth2/v2/auth`

Redirect Uri: `['http://oauthlab.zapto.org/auth/oauth/callback']`

Token Exchange Uri: `https://accounts.google.com/o/oauth2/`

Response Type: `['code']`

Scope: `['profile']`

State: `Ez4oa2d7I0uYgOQPIQLc2uh7cDKPcs`

Send Request

Auth code:

`4/0AVG7fiS0Mnw2KWyz5bM-n_zuer-rzB7BDMBn_9sWN3aW2nthFExZFC4VGiiQsKbBGiABkQ`



8636 Criptografía y Seguridad Informática

Seguridad en Arquitectura de Servicios

Oauth Code Flow Response

Authorization Code:

```
4/0AVG7fiS0Mnw2KWyZ5bM-n_zuer-  
rzB7BDMbn_9sWN3aW2nthFExZFC4VGiiQsKbBGiABkQ
```

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

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

Original state: Ez4oa2d7I0uYgOQPIQLc2uh7cDKPcs

Returned state: Ez4oa2d7I0uYgOQPIQLc2uh7cDKPcs

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

Continue with the Exchange of authorization code for access token

1.3. Obtener Token de Acceso

secret:

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

Auth code:

4/0AVG7fiS0Mnw2KWyZ5bM-n_zuer-rzB7BDMbn_9sWN3aW2nthFExZFC4VGiiQsKbBGiABkQ

MissingTokenError

`oauthlib.oauth2.rfc6749.errors.MissingTokenError: (missing_token) Missing access token parameter.`

Traceback (most recent call last)

File `"/usr/local/lib/python3.12/site-packages/flask/app.py"`, line 1498, in `__call__`

```
    return self.wsgi_app(environ, start_response)
           ^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^
```

File `"/usr/local/lib/python3.12/site-packages/flask/app.py"`, line 1476, in `wsgi_app`

```
    response = self.handle_exception(e)
               ^^^^^^^^^^^^^^^^^^^^^
```

File `"/usr/local/lib/python3.12/site-packages/flask/app.py"`, line 1473, in `wsgi_app`

```
    response = self.full_dispatch_request()
               ^^^^^^^^^^^^^^^^^^^^^^^^^
```

File `"/usr/local/lib/python3.12/site-packages/flask/app.py"`, line 882, in `full_dispatch_request`

1. ¿Con qué objetivo se utiliza la URI de redirección?

La URI de redirección se utiliza para redirigir al usuario después de que haya autorizado la solicitud de acceso. Es la URL a la que se enviará el código de autorización o el token después de que el usuario haya autenticado y autorizado la solicitud.

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

El **scope** define el nivel de acceso que la aplicación solicita. Permite especificar qué recursos o datos puede acceder la aplicación. Por ejemplo, **profile** permite acceder a la información del perfil de usuario, y utilizar scopes más específicos puede proporcionar acceso más limitado, lo que mejora la seguridad.

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

El parámetro que contiene las acciones que el cliente debe autorizar es el **scope**. Este parámetro especifica los permisos solicitados por la aplicación.

4. ¿Por qué se verifica que el estado enviado en la solicitud sea idéntico al obtenido en la respuesta?

Se verifica el parámetro **state** para prevenir ataques de tipo **Cross-Site Request Forgery (CSRF)**. Asegura que la solicitud realizada por el usuario es legítima y que no ha sido manipulada durante el proceso de autorización.

5. ¿Qué beneficio se obtiene de utilizar OAuth para el acceso?

OAuth permite a las aplicaciones acceder a recursos protegidos en nombre de un usuario sin tener que compartir las credenciales del usuario. Esto mejora la seguridad y la privacidad del usuario.

6. Explicar los pasos del "Authorization Code Flow" que se realizaron en esta parte.

1. La aplicación redirige al usuario a la página de autorización de Google.
2. El usuario inicia sesión y autoriza el acceso.
3. Google redirige al usuario de vuelta a la URI de redirección con un código de autorización.
4. La aplicación intercambia el código de autorización por un token de acceso para obtener información del perfil del usuario.

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?

Se devuelve un código de autorización en lugar de un token para mejorar la seguridad. Esto evita que el token de acceso sea expuesto directamente en la URL. Si el token de acceso fuera devuelto directamente, se trataría de un flujo de autorización implícito.

8. ¿Para qué se utiliza el parámetro 'Secret ID'?

El **Secret ID** se utiliza como una clave secreta que valida que la aplicación que está solicitando el token es legítima. Se utiliza junto con el **Client ID** para asegurar que la solicitud de intercambio del código de autorización proviene de una aplicación registrada y autorizada.

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

Para obtener el correo electrónico, se debe modificar el **scope** e incluir **email**, por ejemplo, **profile email**.

10. ¿Qué ventaja se obtiene al utilizar OpenID respecto a OAuth?

OpenID Connect añade una capa de autenticación adicional sobre OAuth. Mientras que OAuth solo proporciona autorización, OpenID también autentica al usuario y proporciona un identificador único del usuario (ID token) que puede ser utilizado para verificar la identidad del usuario.

11. ¿Sobre qué acciones pide permiso el servicio de Google al cliente en este caso?

En el flujo de OpenID, Google solicita permiso para acceder a la información de autenticación, como el perfil del usuario y su dirección de correo electrónico, dependiendo de los scopes solicitados.

12. ¿Qué información se obtiene del JWT? Explicar cada una de sus partes.

Un JWT (JSON Web Token) contiene tres partes:

1. **Header:** Información sobre el tipo de token (JWT) y el algoritmo de firma utilizado.
2. **Payload:** Contiene los datos del usuario, como el nombre, correo electrónico, etc.
3. **Signature:** Se utiliza para verificar la integridad del token y su autenticidad.

13. ¿Qué son los valores 'e' y 'n' de la sección de 'Verify Signature'?

Los valores **e** y **n** corresponden a los parámetros de la clave pública utilizada para verificar la firma del JWT. Son componentes de una clave RSA y son usados para validar que el token no ha sido alterado.

14. ¿Cómo es el proceso de validación del token?

La validación del token implica:

1. Verificar que la firma del token sea válida utilizando la clave pública.
2. Comprobar que no haya expirado (verificando el campo **exp**).
3. Validar que los campos como el **issuer** y el **audience** coincidan con los valores esperados.

Let me know if you need help with more detailed steps or additional explanations for other questions in the lab!