Intro

- This exercise is an introduction to OAuth and OIDC, using an INDIGO IAM instance as the OIDC Provider (OP).
- We will be using the INDIGO IAM instance ran by the IRIS UK Community, the IRIS IAM.
  - INDIGO IAM is the application of choice for the WLCG token-based infrastructure.
- The main page is found at `public/index.html` and the auth code is in `auth/iam.js`.
  `.env` is where you can configure environment variables for your app.
  - These are the only three areas within Glitch you will need to edit.

Exercises

Exercise 0: Setup

To be completed in Glitch

To complete these exercises you will be modifying this Glitch App. In order to do this, you will need to "Remix" this app, to create your own personal project.

- First of all, you will need to register an account with Glitch. You can do this either by registering via GitHub or Google (using OAuth/OIDC!) or with an email link. You can register at https://glitch.com/signup.
- Once you have your glitch account, you will need to create a personal copy of the exercise materials. To do this, please select the **Remix your own** button from the project page, found at: https://glitch.com/~oauth-oidc-exercises.

Exercise 1

Register for IRIS IAM

To be completed in IRIS IAM

The first step is to register a user account within the IRIS IAM, the **OP** and **Authorisation Server** for this exercise. You will use this account in your role as the **Resource Owner** and to register a **Client**.

Navigate to https://iris-iam.stfc.ac.uk, and from here you should register for an account.
Registering via eduGAIN is the preferred process, however if you do not have membership of an eduGAIN IdP, please follow the **Local Account** instructions

- Registration with an eduGAIN IdP

  1. Select **"Sign in with your institution via eduGAIN"**.
  2. In the resulting field, search for your institution name, e.g. **"CERN"**. Please note that some IdP's may have their acronyms expanded - for example, **STFC** appears as **Science and Technology Facilities Council**.
  3. You will here be redirected to your institutional sign-in page.
  4. After signing in, you will be redirected to the IRIS IAM to complete registration. Please enter **"CSC Exercises"** within the notes field.
  5. Submit your registration request.

- Registration for a Local Account

  1. Select **"Apply for an Account"**
  2. In the resulting fields, enter you name, email and username. Please enter **"CSC Exercises"** within the notes field.
  3. You will here be redirected to your institutional sign-in page.
  4. After signing in, you will be redirected to the IRIS IAM to complete registration. Please enter **"CSC Exercises"** within the notes field.
  5. Submit your registration request.

Some Notes on eduGAIN

- "The eduGAIN interfederation service connects identity federations around the world, simplifying access to content, services and resources for the global research and education community. eduGAIN comprises over 80 participant federations connecting more than 8,000 Identity and Service Providers."
  - *from https://edugain.org/*

IRIS IAM is registered to eduGAIN via the UK Access Management Federation (UKAMF) as a **Service Provider**, allowing it to act as an **Identity Proxy Service** and enabling users to register for the IRIS IAM using their eduGAIN-registered **Identity Providers (IdPs)**. The UKAMF and eduGAIN communicate with services via **SAML**, Security Assertion Markup Language.

Exercise 2

Register a Client within IRIS IAM

To be completed in IRIS IAM

Now that you have an account within IRIS IAM, you now need to register your Glitch Project as a **Client**.

1. From your IRIS IAM Dashboard, select the **My Clients** button in the left hand menu, and then select **New Client**.
2. In the client registration, the following options are compulsory:

- *On the **Main** tab:*
  - **Client Name** - Set a human-readable name for your client
  - **Redirect URIs** - The redirect URL for your app.
    This can be constructed from your Glitch project, which will have been given a generated random name at creation time. You can get the URL directly

by selecting preview, and then Preview in a New Window. This will open your project page, and then you will need to append `/iam/redirect/`. For example, the demo project it is: `https://oauth-oidc-exercises.glitch.me/iam/redirect/`

- *On the **Scopes** tab:*
    - The following scopes must be selected: **email, openid, profile**.
- *On the **Grant Types** tab:*
    - The grant type **authorization_code** is selected

3. Once you have saved the client, you need to identify your **Client ID** and **Client Secret** by clicking back into your client:

- You can find your **Client ID** on the **Main** tab
- You can find your **Client Secret** on the **Credentials** tab

Exercise 3

Complete details within Glitch

To be completed in Glitch

Now that you have registered your App with IAM, you need to populate the details within Glitch. These should be stored within the `.env` file on Glitch. The following fields will need to be completed:

- `IAM_CLIENT_ID`
- `IAM_CLIENT_SECRET`

Exercise 4

Get a token from IRIS IAM

To be completed in Glitch

Once you have registered your client and set things up Glitch-side, you can get a token from IRIS IAM.

- Select **Preview** from the bottom bar in Glitch, and then **Preview in a new window**
- In the newly opened tab, click login with IAM and follow the on-screen steps
- You should be redirected to your Glitch App, with the encoded token and its contents displayed. You can use https://jwt.io/ to decode the token, and understand what the string represents.

Exercise 5

Use Token Claims

To be completed in Glitch

Now that you have your token from IAM, you should look to use the claims from it - similar to how authorisation decisions may be made about token claims.

- For this exercise you should be looking at the `groups` token claim. At this stage, the groups claim for you will be empty, and so you will need to join a group within IAM.
    - You should navigate to the profile page at https://iris-iam.stfc.ac.uk/dashboard#!/home
    - in the search field, search and request to join the group `CSC/testing`

Once you are a member of a group, you will now need to write the decision logic.
In order to see the changes to your groups, you will need to reauthenticate with the IRIS IAM to update your token, and you should see your token's `groups` claim and that it now contains `CSC/testing`.
If it does, you should now look to hve Glitch read the token claims and display whether a user is a member of a group or not.
For this exercise, you should just print whehter a user's `groups` claim contains `CSC/testing` or not. In a real scenario, you would use this membership to gate authorisation to a page or resource.

- You can find where you will need to code this within `public/index.html` at line 85
- The token is a JSON object, containing the ID token claims.
  The claim values can be read by using their names - you can find examples of how to do this by investigating the `auth/iam.js` file, where the `email` and `fullname` values are extracted from the token.

Exercise 6

Extensions

To be completed in Glitch and IRIS IAM

- Now that you have made decisions based off of groups, you could also investigate other token claims, and look to make decisions based on these. Some services may limit users to a certain email domain, for example.
- You could also look to activate the `refresh_token` scope - you can find demo code to do this within `iam.js`. In order to enable this, you will need to give your app the `refresh_token` grant within the IAM settings for your client.
- INDIGO IAM supports different token profiles, including the WLCG token profile and the AARC (Authentication and Authorisation for Research and Collaboration) profile, which ensures groups follow the guidance of AARC G002. You can activite different token profiles by requesting the corresponding scope, and adjusting your other requested scopes to reflect the new profile. Have a look at the IAM JWT token profiles documentation and try requesting both the `wlcg` and `aarc` token profiles for your client.
- Interested in a more complete view about token claims? The Oidc ProvidEr featUre Support (OrPhEUS) Portal by KIT provides a similar view to this simple

Glitch tool, and allows you to authentucate with different OPs - such as Google. You may also test the DeviceCode flow here as well.