Extra notes - Client-side Design and Development

- Dr Nick Hayward

JS - Google APIs Overview

A brief overview to working with Google APIs, OAuth 2.0, &c.

Contents

- Intro
- Common options
- Setup
- App access

Intro

Google's JS client library enables an app to interact with Google APIs, including Drive &c.

API requests can be made in a number of different ways using the JS client library. The basic pattern is as follows,

- app loads JS client library
- app initialises library with API key, OAuth client ID, and API discovery doc (where applicable)
- app sends request and then processes the response

Common options

There are three common ways to use the JS client library,

- load API discovery document, then assemble request
- use `gapi.client.request`
- use CORS

Setup

To access Google APIs, you'll need to setup the following prior to coding an app.

- create a Google account (assuming you don't already have one suitable for app development)
- create a Google project using the Google API Console
- enable Google APIs
  - choose required Google APIs for the project - use APIs explorer
  - use the Google API console to choose APIs, select one, then click the *enable* button
  - follow the remaining instructions...

App access

Per app, and API, Google defines two permitted level of API access,

- **simple** - API call cannot access a user's private data - requires API key
- **authorised** - API call can read and write a user's private data &c. - requires OAuth 2.0 credentials

For each app, we can acquire a key and credentials as follows,

- API key

    - in the API console, open the Credentials page
    - choose **Create credentials > API key** - select required key
    - follow guidelines for securing an app's key - best practices

- OAuth 2.0 credentials
    - in the API console, open the Credentials page
    - choose **Create credentials > OAuth client ID** - select appropriate app type