Institute of Data

2023

Data Science and AI

Module 3
Part 2:

APIs

Agenda: Module 3 Part 2

- What is an API?
- APIs for data services
- APIs for analytic services
- APIs for visualisation services
- APIs for cognitive services
- Creating an API

What is an API?

- Definition, examples
- Interfaces
- Authentication protocols
- Documentation

What is an API?

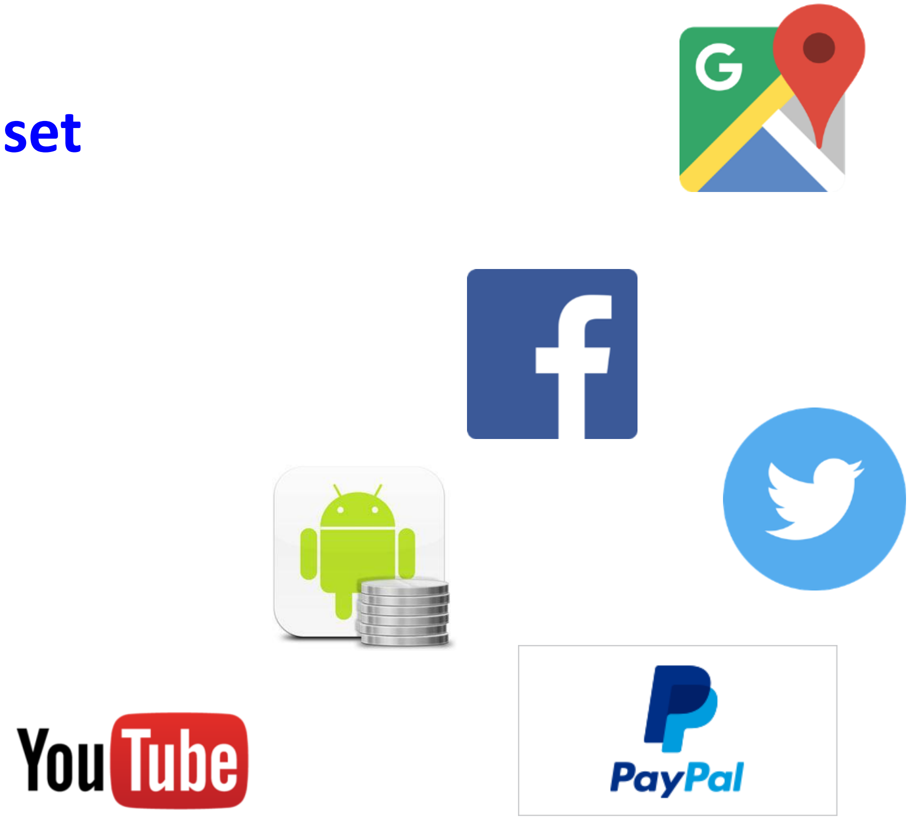
- What does “API” stand for?
  - Application Programming Interface

- Examples?
  - automation in Microsoft Office
    - e.g. generating a Word document or an Outlook reminder from another application
  - high-level database drivers
    - e.g. PyMongo
  - programming libraries for mobile & wearable devices
  - programmable web services
  - other?

Use Cases for APIs

- integrate remote data access
  - repetitive analyses of an **evolving dataset**
  - **up-to-the-moment** forecasting

- **integrate** familiar functionality
  - location sharing using Google Maps
  - simplified app login via Facebook
  - in-app purchases
  - in-app YouTube viewing



Some Popular Web Service APIs

Name	Nature	URL
Twitter	Networking, marketing, trending	https://developer.twitter.com/en.html
Facebook	Networking, marketing	https://developers.facebook.com/tools/
Amazon S3	Cloud storage, Big Data analytics	https://aws.amazon.com/s3/
LinkedIn	Networking	https://developer.linkedin.com/
eBay	E-commerce	https://developer.ebay.com/
Google API Console	Data access & analytics, e-commerce, etc.	https://developers.google.com/apis-explorer/#p/
New York Times	News	http://developer.nytimes.com/

Interfaces for Web Service APIs

- SOAP
  - *Simple Object Access Protocol*
  - early, widespread web service protocol
  - exposes components of application logic as services
  - XML

XML (eXtensible Markup Language) is one such format. XML is a markup language that defines rules for encoding documents in a format that is both human-readable and machine-readable.

- REST
  - *Representational State Transfer*
  - now > 70% of public APIs
  - accesses data
  - variety of data formats, coupled with JSON
  - generally faster and uses less bandwidth
  - easier to integrate with existing websites

Overview of RESTful API Description Languages:
https://en.wikipedia.org/wiki/Overview_of_RESTful_API_Description_Languages

roll your own:
https://www.restapitutorial.com/
https://aws.amazon.com/api-gateway

HTTP

- HyperText Transfer Protocol
- underlies RESTful APIs

- 4 major methods

GET	fetches data from web server
PUT	edits data on web server
POST	adds new data
DELETE	removes data

- HTTP Status Codes

1xx	informational
2xx	success
3xx	redirection
4xx	client error
5xx	server error

https://www.restapitutorial.com/httpstatuscodes.html

Elements of an API call

- ***endpoint***
  - URL of a server page that provides data or functionality via ***requests*** and ***responses***
- ***protocol***
  - the communication standard for passing requests to an endpoint
- ***authentication***
  - secure **identification** of user making request
  - if a developer creates an app for other users, the app needs to obtain **authorisation** from the owner of the API for both the developer's access *and* the user's access

Authentication Protocols

- HTTP Basic Access Authentication
  - username + password
  - transmitted in header of HTTP request
  - weakly encoded, no encryption
- OAuth 1.0
  - uses encrypted tokens
- OAuth 2.0
  - simpler, more robust than OAuth 1.0

OAuth 2.0

- token-based
  - e.g. ***client_id*** & ***client_secret***
  - allows a 3rd-party app to access a user's/developer's account **without knowing the account password**
  - allows an end-user to access an API via *your* app, using *their* token
- redirect URL
  - **registered** when app created
  - OAuth 2.0 service **returns user to this URL** after authorising (and issuing a user token)
  - protects access token from **interception**

https://www.oauth.com/oauth2-servers/background/

Developer Access

- some API's have **a developer mode** that may allow access without requesting a user token
- options for connect/request include:
  - use developer's *user_id* and *password*
  - use *app_id*, developer's *client_id*, developer's *secret*
- access granted **may** include
  - read developer's posts, comments, profile, etc.
  - post to developer's account
  - read other users' posts, comments, profiles, etc.

Python Libraries: Utilities

requests

- HTTP library (“elegant and simple”)
- http://www.python-requests.org/en/latest/
- returns JSON-formatted byte strings

json

- JSON <-> lists, dictionaries
- https://docs.python.org/2/library/json.html

untangle, xmltodict

- parses XML to Pythonic data structures

BeautifulSoup (bs4)

- parses HTML, XML to Pythonic data structures

Python Libraries: API Wrappers

- simplify usage of APIs by introducing a Python API into the loop
- use data types & structures familiar to Python developers

pyfacebook

linkedin

praw (Reddit)

bucketstore (Amazon S3)

python-forecastio (weather)

foursquare (location-based networking)

GooPyCharts (Google Charts)

indeed (indeed.com)

kiteconnect (stock trading)

pymaps (Google Maps)

pymed (PubMed)

pyspotify (Spotify)

newsapi

rottentomatoes (crowd-based movie reviews)

sportradar (sport APIs)

tesserocr (OCR)

bowshock (NASA)

geopy (geocoding)

https://github.com/realpython/list-of-python-api-wrappers

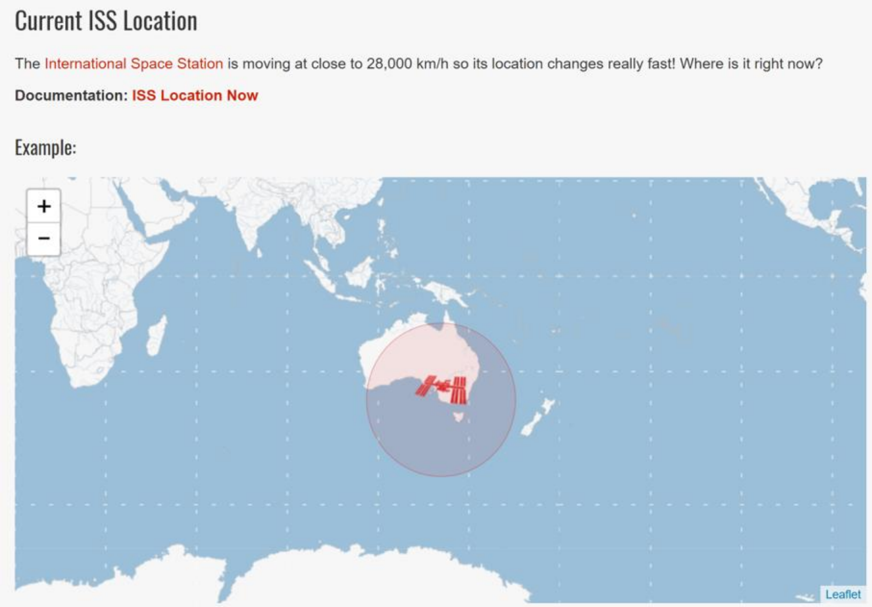
Lab 3.2.1: Querying the ISS

- Purpose:
  - To become familiar with basic API requests and responses

- Resources:
  - API for the International Space Station: **OpenNotify**

    http://open-notify.org/Open-Notify-API/
  - HTTP response codes

    https://www.restapitutorial.com/httpstatus



Current ISS Location

The International Space Station is moving at close to 28,000 km/h so its location changes really fast! Where is it right now?

Documentation: **ISS Location Now**

Example:

- Materials:
  - ‘Lab 3.2.1.ipynb’

Extracting Data from APIs

- Reddit API
- Google Public Data and BigQuery API

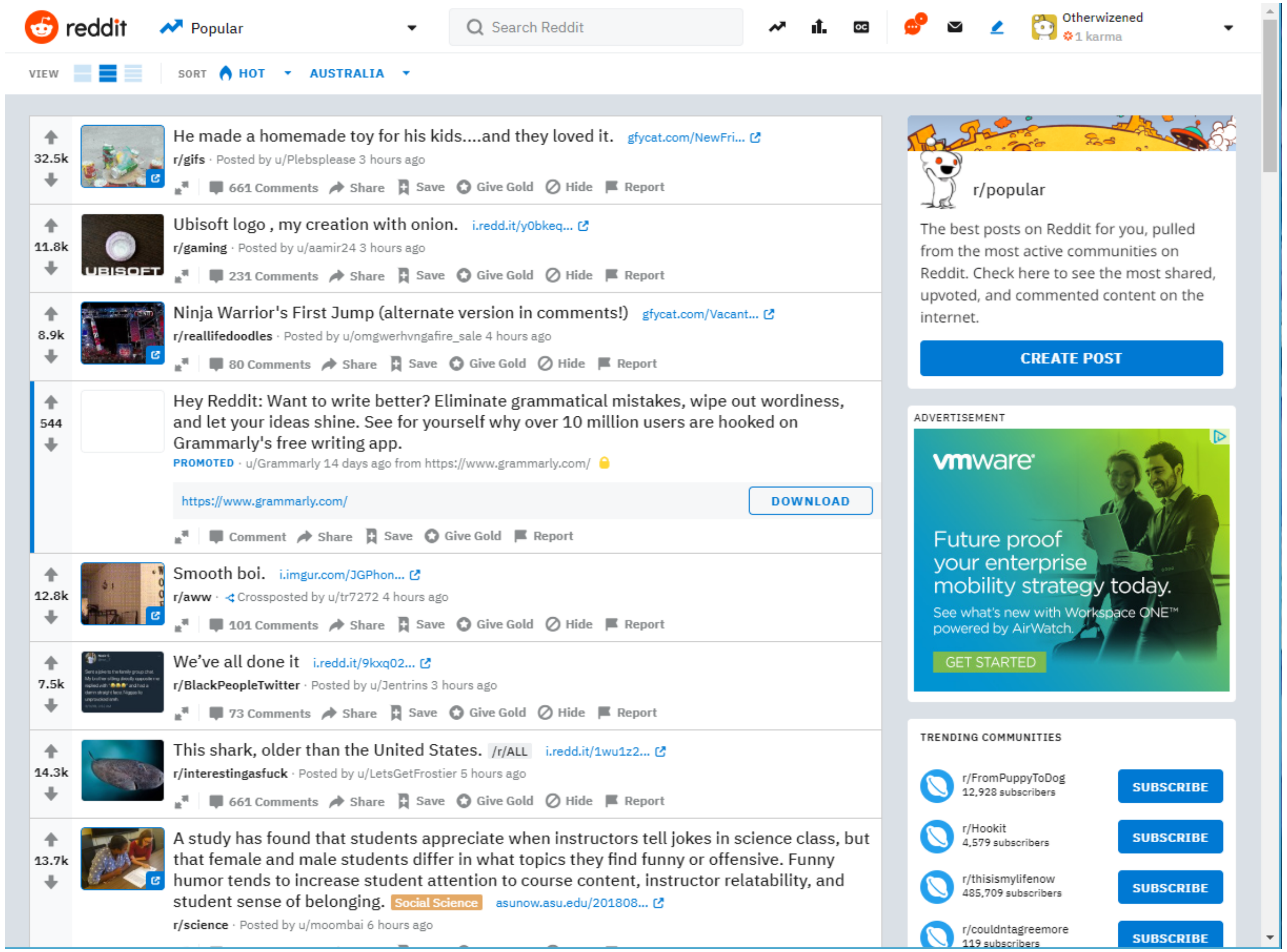
Reddit API

- Introduction to Reddit
- API structure
- Developer access
- Reddit API: Using Python

Reddit

- why Reddit?
  - good example of a social media product
  - rich content
  - large user base
  - highly structured API
  - immediately accessible

https://www.reddit.com/wiki/faq



32.5k He made a homemade toy for his kids....and they loved it. gfycat.com/NewFri...
r/gifs · Posted by u/Plebsplease 3 hours ago
661 Comments Share Save Give Gold Hide Report

11.8k Ubisoft logo , my creation with onion. i.redd.it/y0bkeq...
r/gaming · Posted by u/aamir24 3 hours ago
231 Comments Share Save Give Gold Hide Report

8.9k Ninja Warrior's First Jump (alternate version in comments!) gfycat.com/Vacant...
r/reallifedoodles · Posted by u/omgwerhvngafire_sale 4 hours ago
80 Comments Share Save Give Gold Hide Report

544 Hey Reddit: Want to write better? Eliminate grammatical mistakes, wipe out wordiness, and let your ideas shine. See for yourself why over 10 million users are hooked on Grammarly's free writing app.
PROMOTED · u/Grammarly 14 days ago from https://www.grammarly.com/
https://www.grammarly.com/ DOWNLOAD
Comment Share Save Give Gold Report

12.8k Smooth boi. i.imgur.com/JGPhon...
r/aww · Crossposted by u/tr7272 4 hours ago
101 Comments Share Save Give Gold Hide Report

7.5k We’ve all done it i.redd.it/9kxq02...
r/BlackPeopleTwitter · Posted by u/Jentrins 3 hours ago
73 Comments Share Save Give Gold Hide Report

14.3k This shark, older than the United States. /r/ALL i.redd.it/1wu1z2...
r/interestingasfuck · Posted by u/LetsGetFrostier 5 hours ago
661 Comments Share Save Give Gold Hide Report

13.7k A study has found that students appreciate when instructors tell jokes in science class, but that female and male students differ in what topics they find funny or offensive. Funny humor tends to increase student attention to course content, instructor relatability, and student sense of belonging. Social Science asunow.asu.edu/201808...
r/science · Posted by u/moombai 6 hours ago

r/popular

The best posts on Reddit for you, pulled from the most active communities on Reddit. Check here to see the most shared, upvoted, and commented content on the internet.

CREATE POST

ADVERTISEMENT

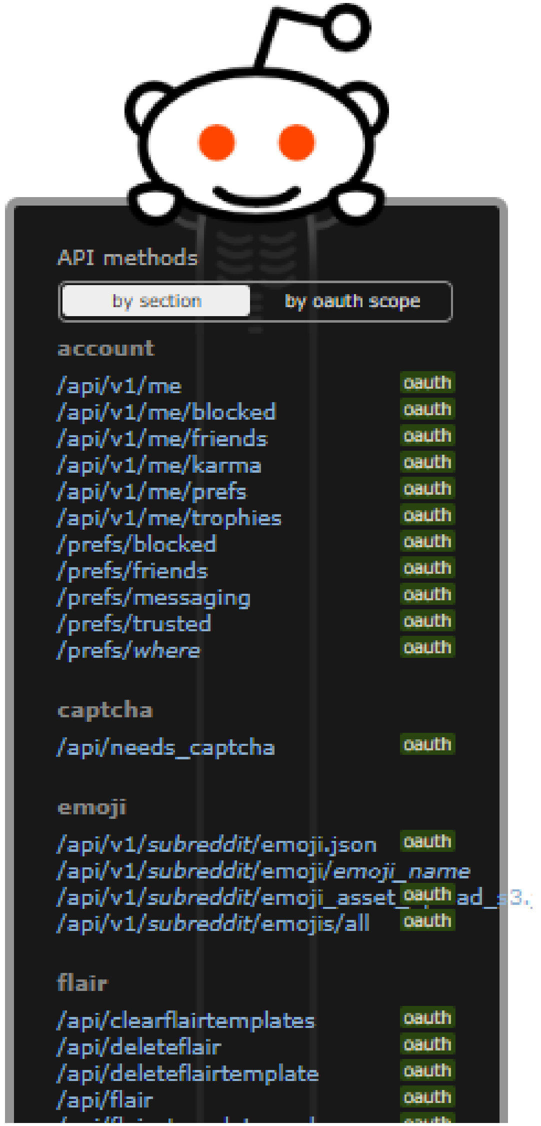
vmware
Future proof your enterprise mobility strategy today.
See what's new with Workspace ONE™ powered by AirWatch.
GET STARTED

TRENDING COMMUNITIES

r/FromPuppyToDog 12,928 subscribers SUBSCRIBE
r/Hookit 4,579 subscribers SUBSCRIBE
r/thisismylifenow 485,709 subscribers SUBSCRIBE
r/couldntagreemore 119 subscribers SUBSCRIBE

Reddit API

- *Account* endpoints:
  - *me*, *me/friends*, *me/prefs*, …
- Links & comments endpoint:
  - *comment*, *vote*, *report*, …
- *Listing* endpoints:
  - categories
    - *hot*, *new*, *random*, …
  - navigation (pagination) and filtering
    - *before*, *after*, *count*, *show*
- and many more …



API methods

account

/api/v1/me	oauth
/api/v1/me/blocked	oauth
/api/v1/me/friends	oauth
/api/v1/me/karma	oauth
/api/v1/me/prefs	oauth
/api/v1/me/trophies	oauth
/prefs/blocked	oauth
/prefs/friends	oauth
/prefs/messaging	oauth
/prefs/trusted	oauth
/prefs/*where*	oauth

captcha

/api/needs_captcha	oauth

emoji

/api/v1/*subreddit*/emoji.json	oauth
/api/v1/*subreddit*/emoji/*emoji_name*	
/api/v1/*subreddit*/emoji_asset_ ad_s3.	oauth
/api/v1/*subreddit*/emojis/all	oauth

flair

/api/clearflairtemplates	oauth
/api/deleteflair	oauth
/api/deleteflairtemplate	oauth
/api/flair	oauth

https://www.reddit.com/dev/api

Reddit API: Developer Access

1. Open a Reddit user account
2. Create a Reddit app
3. Register the app for API access
4. Store your credentials
   - for accessing your account:
     - user name
     - password
   - for authenticating your app:
     - user agent (information describing your app)
     - client ID (a unique identifier for your app)
     - client secret (secure token for authorising your app to access the API)

Reddit API: Using Python

- install PRAW package
- import praw
- create a connection object (to Reddit API)
- invoke API methods on the connection object
  - send requests that GET or PUT data to/from Reddit objects
- do something with data!

https://www.reddit.com/r/popular/

https://www.reddit.com/wiki/faq

https://praw.readthedocs.io/en/stable/getting_started/quick_start.html

Lab 3.2.2: Mining Social Media with Reddit

- Purpose:
  - To develop skills in using a media-rich API

- Resources:
  - Python library for Reddit API: ***PRAW***
    https://praw.readthedocs.io/en/stable/getting_started/quick_start.html

- Materials:
  - ‘Lab 3.2.2.ipynb’

Google Cloud Platform

- public data sets / BigQuery
- APIs based on data science products

Google Cloud Platform

Google Cloud SDK	• https://cloud.google.com/sdk/gcloud/ • https://cloud.google.com/sdk/docs/initializing
Google Cloud Platform	• https://github.com/GoogleCloudPlatform/python-docs-samples • https://googlecloudplatform.github.io/google-cloud-python/ • https://googlecloudplatform.github.io/google-cloud-python/latest/
Google API Client Libraries	• https://developers.google.com/api-client-library/
Google BigQuery	• https://cloud.google.com/bigquery/public-data/ • https://cloud.google.com/bigquery/docs/quickstarts/quickstart-web-ui • https://cloud.google.com/bigquery/docs/reference/libraries • https://cloud.google.com/bigquery/create-simple-app-api • https://github.com/GoogleCloudPlatform/google-cloud-python/tree/master/bigquery

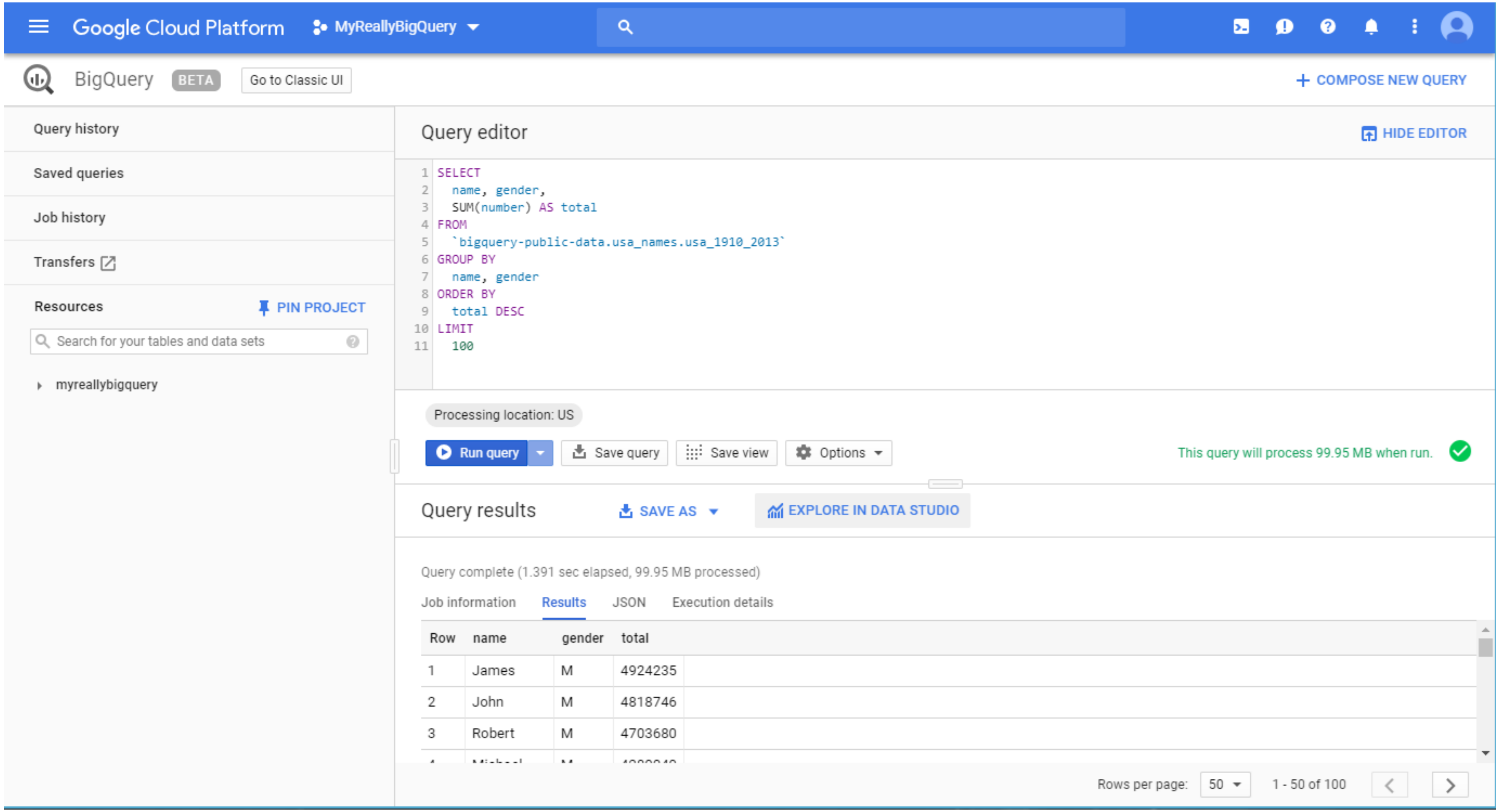
Google Public Data sets

- accessible via Google BigQuery
- free for 1st TB / month
- subject areas:
  - genomics
  - medicine & epidemiology
  - geo imagery (Earth science, weather, etc.)
  - transport & service utilisation
  - annotated images
  - etc.
- https://cloud.google.com/public-datasets/

Google BigQuery

Quickstart to BigQuery Web UI:

https://cloud.google.com/bigquery/docs/quickstarts/quickstart-web-ui



```sql
SELECT
  name, gender,
  SUM(number) AS total
FROM
  `bigquery-public-data.usa_names.usa_1910_2013`
GROUP BY
  name, gender
ORDER BY
  total DESC
LIMIT
  100
```

This query will process 99.95 MB when run.

Query complete (1.391 sec elapsed, 99.95 MB processed)

Row	name	gender	total
1	James	M	4924235
2	John	M	4818746
3	Robert	M	4703680

BigQuery API: Authentication

Service accounts

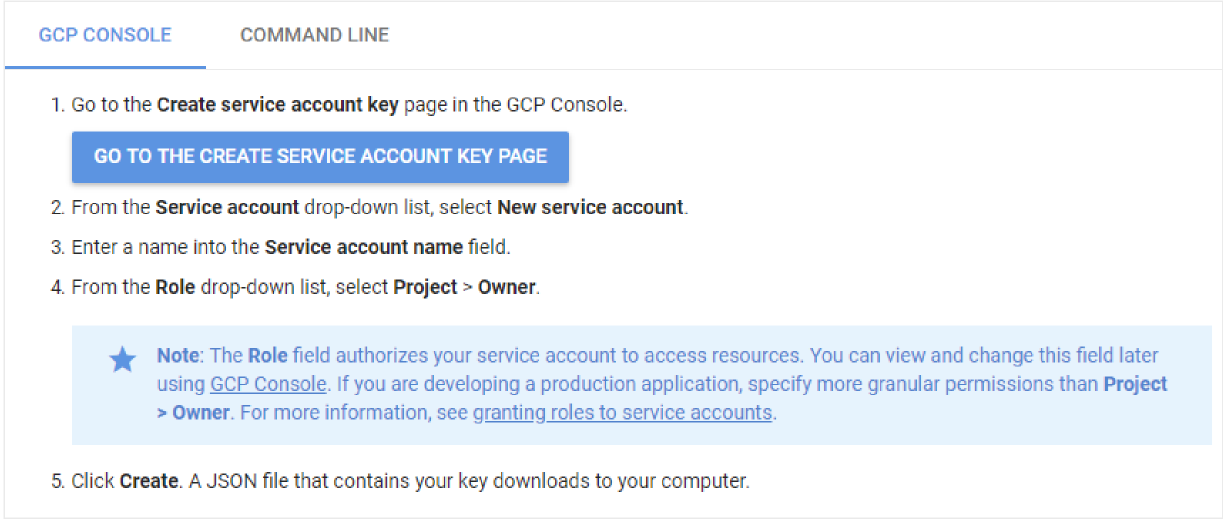
- for client apps that you will run
  - e.g. dev/test, batch processing pipelines
- authentication via your service credentials

User accounts

- for apps you create for other end-users
  - e.g. data products
- authentication via end-users credentials
  - app can only access BigQuery tables that the end-user is authorised to access
  - end-user gets billed for queries

https://cloud.google.com/bigquery/docs/authentication/

BigQuery API: Authentication – cont’d



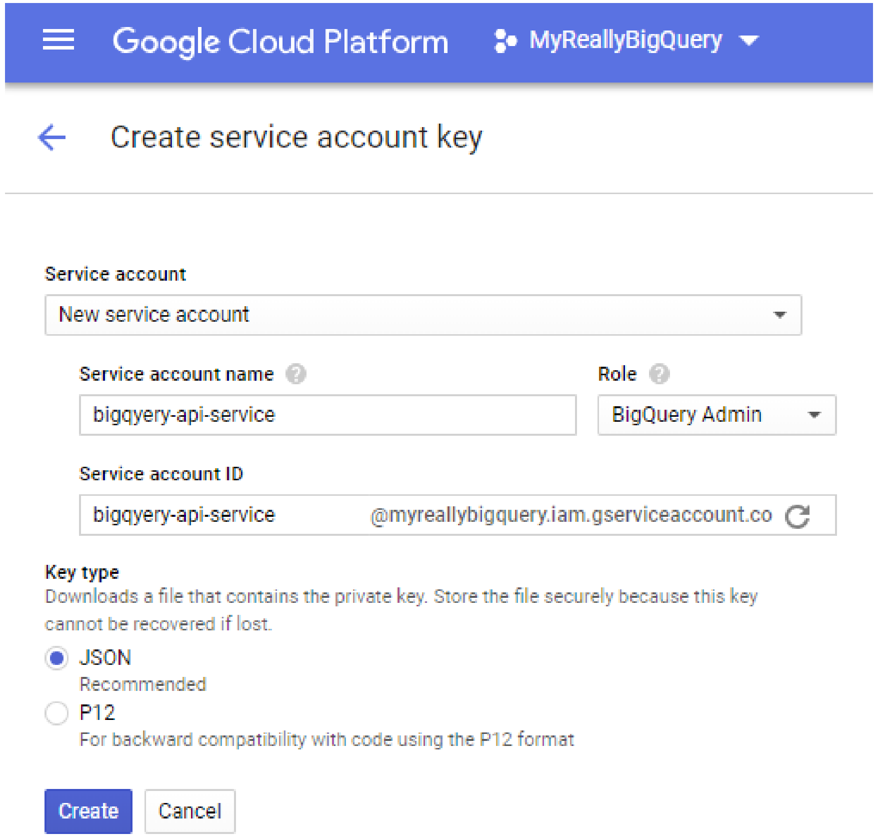
1. Go to the **Create service account key** page in the GCP Console.

2. From the **Service account** drop-down list, select **New service account**.
3. Enter a name into the **Service account name** field.
4. From the **Role** drop-down list, select **Project** > **Owner**.

> ★ **Note**: The **Role** field authorizes your service account to access resources. You can view and change this field later using GCP Console. If you are developing a production application, specify more granular permissions than **Project > Owner**. For more information, see granting roles to service accounts.

5. Click **Create**. A JSON file that contains your key downloads to your computer.

https://cloud.google.com/docs/authentication/production



Create service account key

Service account: New service account

Service account name: bigqyery-api-service

Role: BigQuery Admin

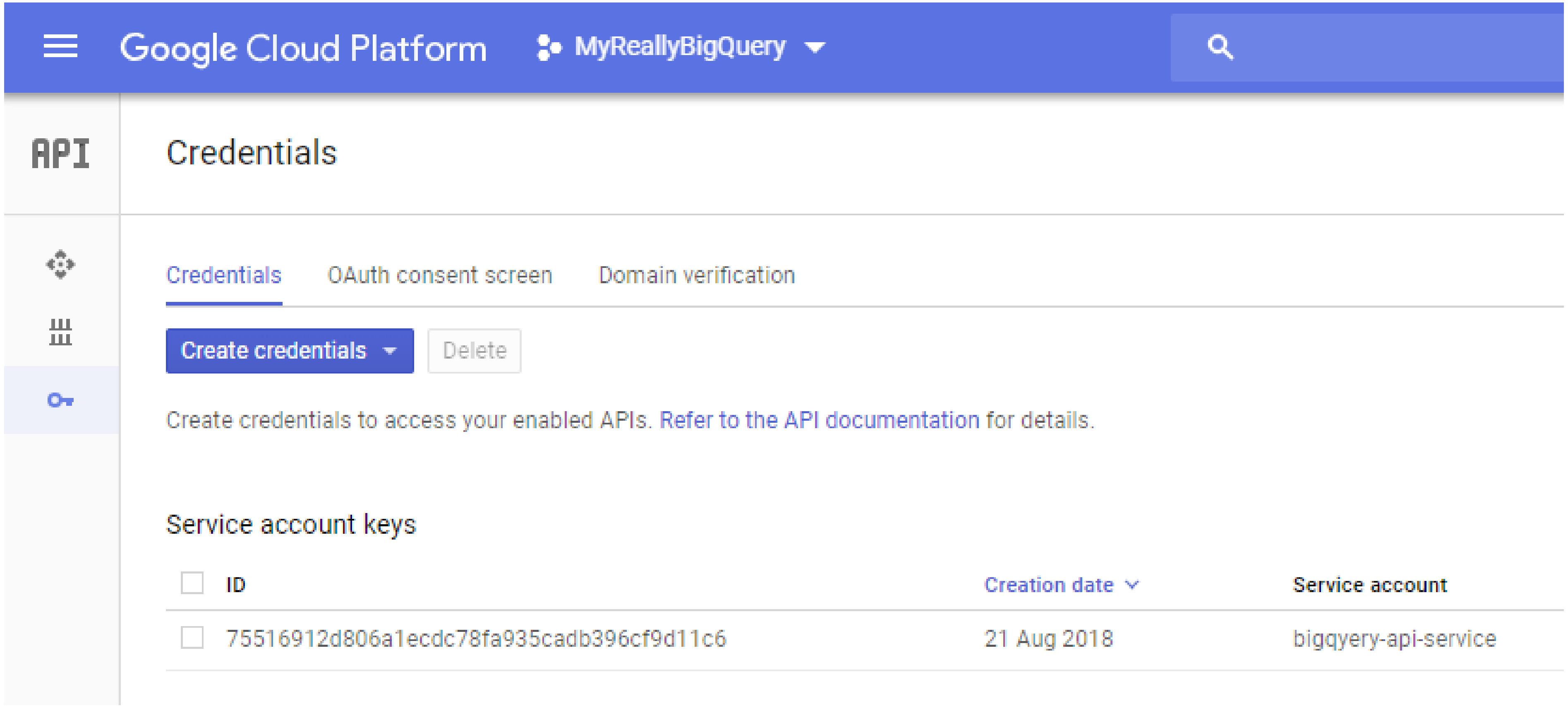
Service account ID: bigqyery-api-service @myreallybigquery.iam.gserviceaccount.co

Key type
Downloads a file that contains the private key. Store the file securely because this key cannot be recovered if lost.

● JSON
Recommended

○ P12
For backward compatibility with code using the P12 format

BigQuery API: Authentication – cont’d



Credentials

Create credentials to access your enabled APIs. Refer to the API documentation for details.

Service account keys

□	ID	Creation date	Service account
□	75516912d806a1ecdc78fa935cadb396cf9d11c6	21 Aug 2018	bigqyery-api-service

Using the Google Authentication Key

Option 1: Set GOOGLE_APPLICATION_CREDENTIALS environment variable

- Linux / MacOS

```
$ export GOOGLE_APPLICATION_CREDENTIALS="[PATH]"
```

- Windows

```
$ set GOOGLE_APPLICATION_CREDENTIALS="[PATH]"
```

Option 2: Pass the path to the service account key in code

```
from google.cloud import storage
storage_client = storage.Client.from_service_account_json(‘[PATH]’)
```

where ‘[PATH]’ is the full file path of the json key file

Google BigQuery API: Top-Level Object

client object:

- connection
  - authenticated connection to the BigQuery service
  - determines credentials
    - implicitly from the environment,
    - or directly via *from_service_account_json* and *from_service_account_p12*
- project
  - top-level container
  - tied to billing
  - can provide default access control across all its datasets
  - access control list (ACL)
    - grants reader / writer / owner permission to one or more entities
    - must be managed using the Google Developer Console (not API)

BigQuery API Object Hierarchy

```
bigquery
    .projects
    .datasets
        .get, .delete, .insert, .list, .update, ...
    .tabledata
    .tables
    .jobs
        .get, .cancel, .insert, .list, .query, ...
    . . .
```

https://developers.google.com/apis-explorer/#p/bigquery/v2/

Lab 3.2.3: Big Data Analytics with BigQuery

- Purpose:
  - (1) To learn how to the Google BigQuery Web UI for discovering public data sets and performing basic analytics.
  - (2) To become proficient with the Google BigQuery API for wrangling Google’s public datasets.

- Materials:
  - ‘Lab 3.2.3.ipynb’

Lab 3.2.3 – cont'd

- Python packages :
  - pyarrow (pip)
  - google-cloud-bigquery (conda-forge)
  - google-cloud-storage (conda-forge)

- Resources:
  - Google BigQuery Public Datasets https://cloud.google.com/bigquery/public-data/
  - BigQuery UI https://cloud.google.com/bigquery/docs/quickstarts/quickstart-web-ui
  - Python client for BigQuery API https://github.com/GoogleCloudPlatform/google-cloud-python/tree/master/bigquery

Discussion

- Extracting data using APIs
    - applications?

Lab/ HOMEWORK

1. Create a mini-project based on any skills from the course so far:
   - select an interesting public data set or form a question you are interested answer and identify data needed to answer the question
   - use Jupyter Notebook to access, analyse and visualise the data
2. Prepare a 5-minute presentation
   - use Jupyter Notebook
   - organise as:
     - question
     - dataset & analysis
     - conclusion
3. plan to present to the class

Presentations

- each team
  - 5 minute presentation

Analytics-Based APIs

- **Google**
  - Google Analytics
    - https://developers.google.com/analytics/
  - Google Cloud Vision
    - https://cloud.google.com/vision/
  - Google Cloud AI
    - https://cloud.google.com/products/ai/
- **IBM Watson**
  - Developer Cloud
    - https://www.ibm.com/watson/developercloud/
    - https://github.com/watson-developer-cloud/python-sdk
  - Mashups
    - https://www-01.ibm.com/common/ssi/cgi-bin/ssialias?subtype=SP&infotype=PM&htmlfid=LBS03048USEN&attachment=LBS03048USEN.PDF

Analytics-Based APIs – cont’d

- AWS
  - Boto3
    - low-level (“client”) and high-level (“resource”) APIs for all AWS products
    - https://aws.amazon.com/sdk-for-python/
  - API Explorer
    - https://developers.google.com/apis-explorer/#search/analytics/analytics/v3/

- Azure
  - Code samples, Cognitive Services API, etc.
    - https://docs.microsoft.com/en-us/python/azure/?view=azure-python
  - Python API Browser
    - https://docs.microsoft.com/en-au/python/api/?view=azure-python

Machine Vision APIs

- use cases:
  - autonomous vehicles
  - industrial control & QA
  - face recognition
  - number plate recognition
  - biometric identity verification
  - print & handwriting transcription
  - image annotation
    - detecting and labelling objects or themes in an image

Creating APIs

- Why would a data scientist/engineer want to create their own API?
  - for building an interface to your data product
  - for enforcing control over how your application’s data and services can be used
  - for isolating the IP that your data product is based on

- References:
  - https://www.fullstackpython.com/application-programming-interfaces.html

Discussion

More APIs

- List of Free APIs (Rapid API)
  https://rapidapi.com/collection/list-of-free-apis/

- Public APIs List
  https://apislist.com/

- todmotto Public APIs
  https://github.com/toddmotto/public-apis

HOMEWORK

1. Investigate a data or analytic API for one of the following:
    - AWS
    - Microsoft Azure
    - IBM Cloud
2. Create a Jupyter notebook that demonstrates some basic operations (e.g. transporting, querying, or visualising data).

NOTES:

- The offerings of these platforms are myriad and complex. It may not be obvious which API you need to use at first, so try to start with published code examples.
- APIs (and the libraries that wrap them) change. Online examples may not work as documented.

Questions?

End of Presentation!