



API USE FOR RESEARCH

SESSION 1



Course tutors

Dr. Somya Iqbal, Alex Crest, & Dr. Aybuke Atalay



AYBUKE ATALAY



ALEX CREST



SOMYA IQBAL



Housekeeping and itinerary

Session 1 - Day 1 - 26/11/25

Remit: Understanding fundamentals and demonstration

- [14:00-14:20] - **Talk:** Introduction to APIs and use cases
- [14:20-14:30] - **Q & A** - main goals from attendees for API use and domain
- [14:30-14:40] - **Talk & brief demo:** Web based access for APIs (searchable interface) a short demonstration
- [14:40-15:00] -**Talk:** Understanding API documentation
- [15:00-15:20] - **Talk :** Understanding file outputs with JSON/XML structures

10-minute break

- [15:30-15:45] - **Talk** and **demo:** Understanding API user keys/authentication
- **Demonstration** (Google APIs)

Final 15 minutes

- Discussion: **Q & A**
- Next session overview for practical & exercise
- Record attendance & feedback



Introduction

Application Programming Interface

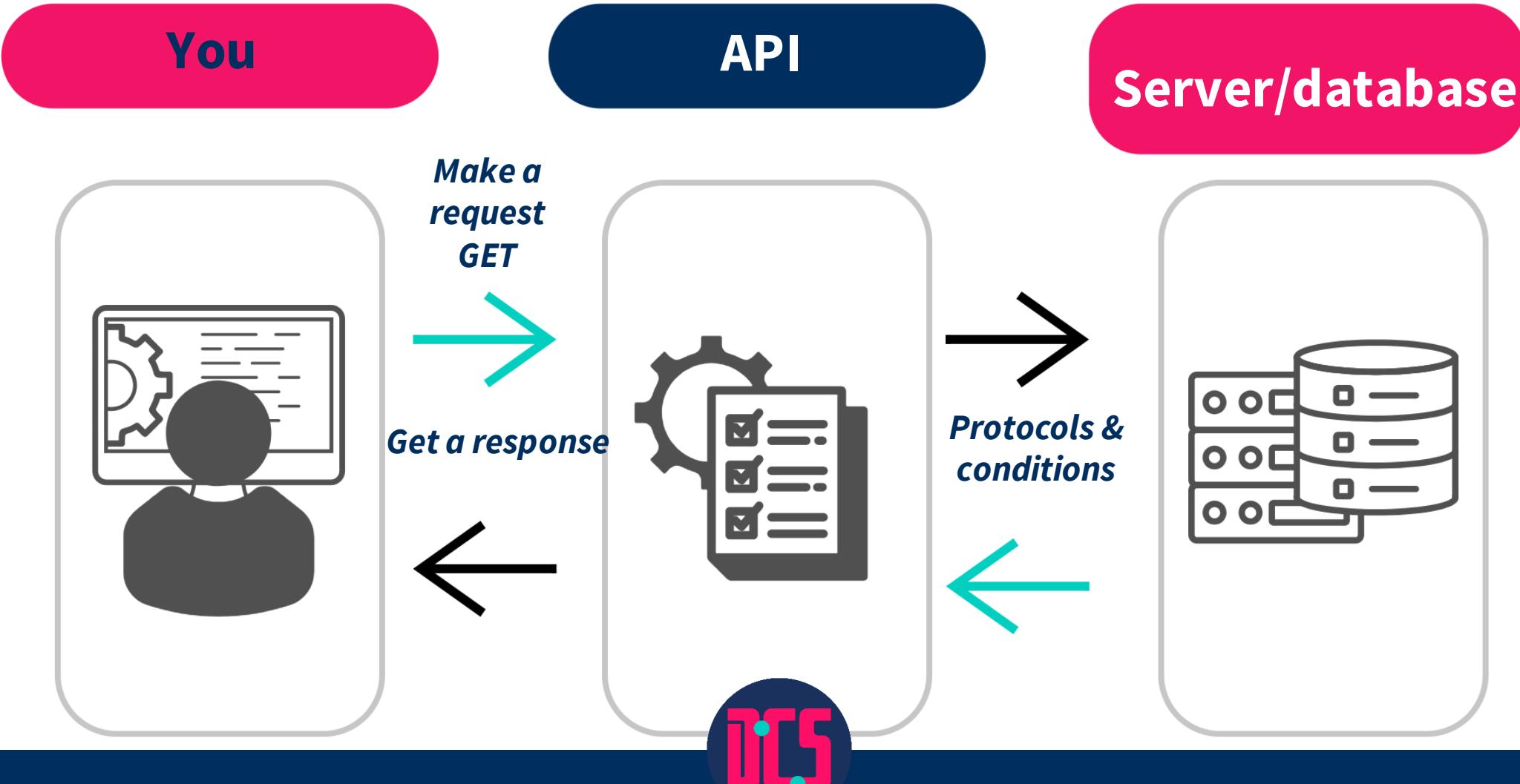


Image: SI

API Types & uses

Are they open? public? Private? Internal?

Differing architectures in how these are setup

REST

Clients use HTTP verbs--> GET (read), POST (create), PUT/PATCH (update), DELETE (remove).
For example, GET /api/objects/123 will get object #123

SPARQL

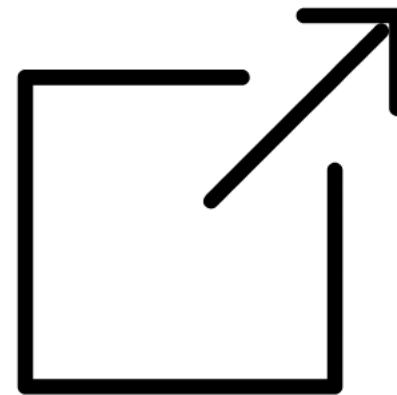
(Sparkle). This is a query language specifically for RDF (Resource Description Framework) data (the semantic web standard). SPARQL queries graph-like data (subject-predicate-object triples) and returns results.

GraphQL

Instead of fixed endpoints, GraphQL lets clients request exactly the data they need via a query language. It operates over a single HTTP endpoint.



HTTPS and the web...



https://api.UNIPROT.org/v1/genes/P12345/search?species=mouse&limit=20

Protocol

Base URL

Version

Endpoint

Path

Parameter

Query
Paramaters





403. That's an error.

We're sorry, but you do not have access to this document.
That's all we know.

404

Not Found

The resource requested could not be found on this server!

Discussion

**What are your main goals &
interest in APIs?**



GUI (Graphical User Interface)

- Different from an API ---> A GUI is a visual interface that allows users to interact with applications using elements like windows, buttons, and menus.
- Designed to be intuitive and user-friendly, enabling real-time data manipulation and interaction
- Interactive and platform specific
- Requires minimal programming skill
- Often used to make initial assessments, like a broad search or distinct information.



GUI - brief demonstration

The screenshot shows the STRING homepage. At the top left is the STRING logo with a molecular graph icon. At the top right are links for Search, Download, Help, and My Data. The main title "Welcome to STRING" is centered above a subtitle "Protein-Protein Interaction Networks" and "Functional Enrichment Analysis". Below this, three statistics are displayed: "ORGANISMS 12535", "PROTEINS 59.3 mio", and "INTERACTIONS >20 bln". A large "SEARCH" button is at the bottom.

https://string-db.org/cgi/input?sessionId=bxhaGntDTKFg&input_page_show=search=off





Search

Download

Help

My Data

Protein by name



Multiple proteins



Proteins by sequences



Proteins with Values/Ranks



Protein families ("COGs")



Pathway / Process / Disease



Add organism



Organisms

STRING chat New

Examples



SEARCH

Multiple Proteins by Names / Identifiers

List Of Names: (one-per-line or CSV; examples: #1 #2 #3)

... or, upload a file:

Organisms:

GUI

The screenshot displays the UIS Data Browser interface. At the top, there is a dark blue header bar with the UNESCO logo and navigation links for HOME, BROWSE DATA, RESOURCES, ABOUT, and VIEW DATA. Below this, a large white banner on a dark blue background reads "Welcome to the UIS Data Browser". To the left of the banner, a text block states: "The data browser allows users to **view** and **filter data** and **metadata**, **visualize** and **share** it or **download** it in various formats (csv, excel)". A blue button labeled "Browse Data" is located below this text, with an orange arrow pointing towards it from the bottom left. On the right side of the banner, there is a detailed data visualization. It shows a table with columns for Country (Country, 2000, 2001, 2002, 2003, 2004) and a row for "Percentage of teachers in pre-primary education who are qualified according to national standards and who are female (%)" for ISCED 02 - Pre-primary education and Female. The table lists countries like Albania, Algeria, Andorra, Angola, Antigua and Barbuda, Armenia, Azerbaijan, and Bahamas. The year 2003 is highlighted with an orange arrow. The interface includes a sidebar for "View" options (Overview, Time series, Latest year, Filter by country or region, Time range, Table), a "Customize View" section, and various filtering and sharing options at the bottom.

[UIS Data Browser](#)



Software/library examples

Scikit learn is a well-known library for machine learning users

- Composite API
- Can be accessed using Python programming
- Contains validated algorithms for users to apply with ease
- Encourages reproducibility

<https://scikit-learn.org/stable/api/index.html>

The screenshot shows the scikit-learn API Reference page for the `sklearn` module. The top navigation bar includes links for Install, User Guide, API (which is underlined), Examples, Community, More, and a search bar. The main content area has a header "sklearn" and a sub-header "Configure global settings and get information about the working environment". On the left is a "Section Navigation" sidebar with a tree view of the `sklearn` module's submodules: `config_context`, `get_config`, `set_config`, `show_versions`, `sklearn.base`, `sklearn.calibration`, `sklearn.cluster`, `sklearn.compose`, and `sklearn.covariance`. To the right of the sidebar are four API entries:

config_context	Context manager to temporarily change the global scikit-learn configuration.
get_config	Retrieve the current scikit-learn configuration.
set_config	Set global scikit-learn configuration.
show_versions	Print useful debugging information"

Plotly is a graphing library -> for visualisation (web-based) and interactive

- Plotly has well laid out API for making flexible objects from your visualisations
- We use the library features to describe what we want to view and implementation is quick
- Outputs in web formats so users can have interactive features.

<https://plotly.com/python-api-reference/>

The screenshot shows the Python API reference for plotly. The top navigation bar includes links for Site, Page, and a search bar. The main content area has a header "Python API reference for plotly" and a sub-header "This is the reference of plotly's API. Also see plotly's documentation website.". Below this is a section titled "Submodules" with a bulleted list of available submodules:

- Plotly Express: high-level interface for data visualization
- Graph Objects: low-level interface to figures, traces and layout
- Subplots: helper function for layout out multi-plot figures
- Figure Factories: helper methods for building specific complex charts
- I/O: low-level interface for displaying, reading and writing figures
- `plotly.colors`: colorscales and utility functions
- `plotly.datasets`: datasets for demonstration, educational and test purposes



Understanding documentation for APIs

- The user manual for an application programming interface.
- Each API has a **controlled vocabulary** but differs between them.
- Instructions are clear and updated as well as details about versions.
- Specific R and Python projects also have documentation for APIs.



content - content

fetches content based on id/url of an object.

GET

`https://api.si.edu/openaccess/api/v1.0/content/:id`

Parameter

Field	Type	Description
id	String	Row id, url.
api_key	String	the API KEY you received from https://api.data.gov/signup/

Success-Response: HTTP/1.1 200 OK { "status": 200,

Example from Smithsonian: [Documentation](#)



Common file outputs & types

- JSON **Java S cript O bject N otation**
- XML **E xtensible M arkup L anguage**
- RDF **R esource D escription F ramework**
- CSV **C omma S eparated V alues**



File outputs & types cont....

The screenshot shows a web-based JSON viewer tool. At the top, there's a navigation bar with links for "JSON BEAUTIFIER", "JSON PARSER", "XML FORMATTER", "JSBEAUTIFIER", "SAVE", "RECENT LINKS", and "LOGIN". Below the navigation is a title "JSON Viewer" and a toolbar with various icons. On the left, there's a code editor window showing a JSON object with numbered lines 1 through 16. Lines 1 through 10 show basic key-value pairs, while lines 11 through 16 show a nested "constituents" array. In the center, there are three buttons: "Load Data", "JSON Viewer", "Format JSON", and "Download". To the right, there's a tree view of the JSON structure with expandable nodes. A search bar is also present at the top right.

```
1 "objectID": 437112,
2 "isHighlight": false,
3 "acquisitionNumber": "29.100.107",
4 "acquisitionYear": "1929",
5 "isPublicDomain": false,
6 "primaryImage": "",
7 "primaryImageSmall": "",
8 "additionalImages": [],
9 "constituents": [
10   {
11     "constituentID": 162135,
12     "role": "Artist",
13     "name": "Claude Monet",
14     "constituentULAN_URL": "http://vocab.getty.edu/page/ulan/500019484",
15     "constituentWikidata_URL": "https://www.wikidata.org/wiki/Q296"
16   }
]
```

Select a node...

- ▼ object {57}
 - objectID : 437112
 - isHighlight : false
 - acquisitionNumber : 29.100.107
 - acquisitionYear : 1929
 - isPublicDomain : false
 - primaryImage : [value]
 - primaryImageSmall : [value]
- ▼ additionalImages [0]
 - (empty array)
- ▼ constituents [1]
 - ▼ 0 {6}
 - constituentID : 162135
 - role : Artist
 - name : Claude Monet

<https://jsonformatter.org/>



File outputs & types cont....

XML Formatter

Input XML

```
1  <?xml version="1.0" encoding="UTF-8"?>
2  <breakfast_menu>
3  <food>
4    <name>Belgian Waffles</name>
5    <price>$5.95</price>
6    <description>Two of our famous Belgian
      Waffles with plenty of real maple syrup
      </description>
7    <calories>650</calories>
8  </food>
9  <food>
10   <name>Strawberry Belgian Waffles</name>
11   <price>$7.95</price>
12   <description>Light Belgian waffles covered
      with strawberries and whipped cream
      </description>
13   <calories>900</calories>
14 </food>
15 <food>
16   <name>Berry-Berry Belgian Waffles</name>
17   <price>$8.95</price>
18   <description>Light Belgian waffles covered
      with an assortment of fresh berries and
```

Load Data

2 Tab Space

Format / Beautify

XML Tree

Minify / Compact

XML to JSON

Download

How to create XML
File?

Formatted XML

```
1  <?xml version="1.0" encoding="UTF-8"?>
2  <breakfast_menu>
3  <food>
4    <name>Belgian Waffles</name>
5    <price>$5.95</price>
6    <description>Two of our famous Belgian
      Waffles with plenty of real maple
      syrup</description>
7    <calories>650</calories>
8  </food>
9  <food>
10   <name>Strawberry Belgian Waffles</name>
11   <price>$7.95</price>
12   <description>Light Belgian waffles
      covered with strawberries and
      whipped cream</description>
13   <calories>900</calories>
14 </food>
15 <food>
16   <name>Berry-Berry Belgian Waffles</name>
17   <price>$8.95</price>
18   <description>Light Belgian waffles
```



10-MINUTE BREAK



www.ccds.ed.ac.uk

What are keys & authentication?

- Authentication mechanisms are used to verify the identity of users and applications when accessing APIs.
- An API key is a unique identifier used to authenticate requests to an API, ensuring that the caller has the necessary permissions.
- Authentication is the broader process of verifying that a user or application is who it claims to be, often involving credentials like API keys, tokens, or user passwords to protect access and data
- API Keys are embedded in the scripts researchers use to make calls to any API. They require the researcher to request an access key from the institution or API they are attempting to access and have this prior to running their script.



Google Developer Console

Required: Google Account

Google has many APIs that access different features. We will be creating a project to access information on Places – Points marked on Google maps.

This API Requires a Billing Account – but features can be accessed for free.

I will be generating keys for the account to use in any script – you may use this in the next session or in your own time to generate data.

<https://console.cloud.google.com/apis/dashboard>



Q & A



www.ccds.ed.ac.uk

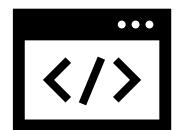
Next session overview



- **To do:** 'Bring' an API you are interested in and associated documentation to go through next session for the exercise

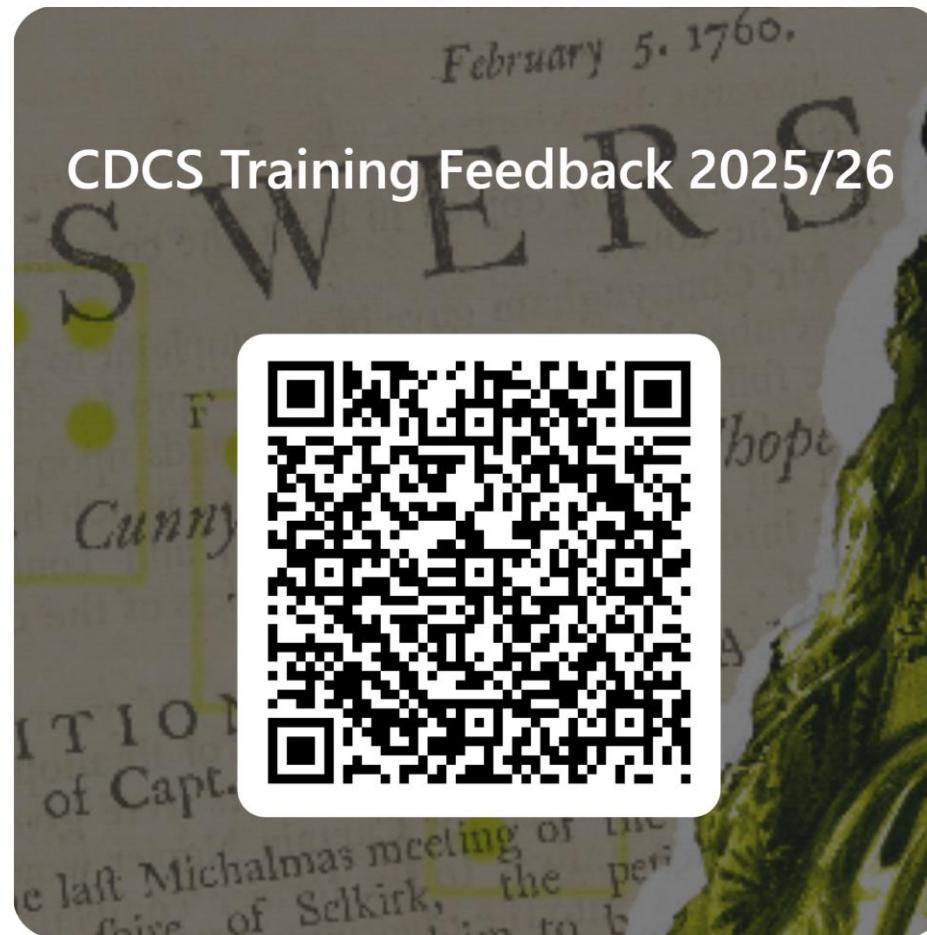


- During the exercise we encourage you to work in pairs and go through the documentation with tutors stopping by to discuss
- Identify any features useful to you, as well as any bottlenecks you notice.



- **Next week** we will also run through a group practical working with a web API (REST) for the Met Museum.







API USE FOR RESEARCH

SESSION 2



Housekeeping and itinerary

Session 2 - Day 2 03/12/25

APIs in practice

- [14:00-14:15]: Introduction to session and recap
- [14:15-14:40]: Exercise - working through API documentation
- [14:40-14:50]: Feedback to group on exercise

10-minute break & setup

- [15:00-15:45 +- 5 mins]: Practical with code book (accessing an API)
- [15:45-15:50]: Discussion

Last 10 minutes

- In closing: signpost to resources
- Record attendance & feedback



Recap talk & plans



www.ccds.ed.ac.uk

Exercise 1

1. Which API are you interested in?
2. What service does your chosen API offer?
3. Group together and go through some of key features listed in the documentation for API access.
4. Is it open or requires authentication?
5. Discuss and lay out the core features you would need



Quick feedback from the group exercise

1. Any bottlenecks
2. Is the API suitable for your use case?
3. Any questions?



10 – MINUTE BREAK (practical setup)



Practical exercise

Programmatically accessing a web API to collect data

- Groups (4)
- Setup (4 notebooks open)
- Guided walk through with Somya (notebook on screen)



GITHUB repository



www.ccds.ed.ac.uk



Edina

You have arrived at Noteable without any *Course* details: the Assignments system will not be available to you.

Please select a personal notebook

server

Standard Python 3

Start

Help and Guides

- Helpful resources for students, lecturers, school teachers and users new to notebooks
- Guides for assignments, collaboration, multiple markers in a course and more



DCS-training / APIs_Forum_2025

Centre for Da... DCS-tra...
Edinburgh
108 repositories 34 members

main ▾ 2 Branches 0 Tags

SoSomya Minor text edits

Notebooks_&_Outputs Delete Notebook

Introduction_&_Overview.md Minor text edits

README.md Update README

README_Auth_Setup.md Rename README

README

Type / to search

Wiki Security Insights Settings

Edit Pins Watch Fork Star

Go to file + <> Code ▾

Local Codespaces

Clone

HTTPS SSH GitHub CLI

https://github.com/DCS-training/APIs_Forum_2025

Clone using the web URL.

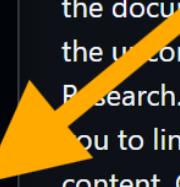
Open with GitHub Desktop

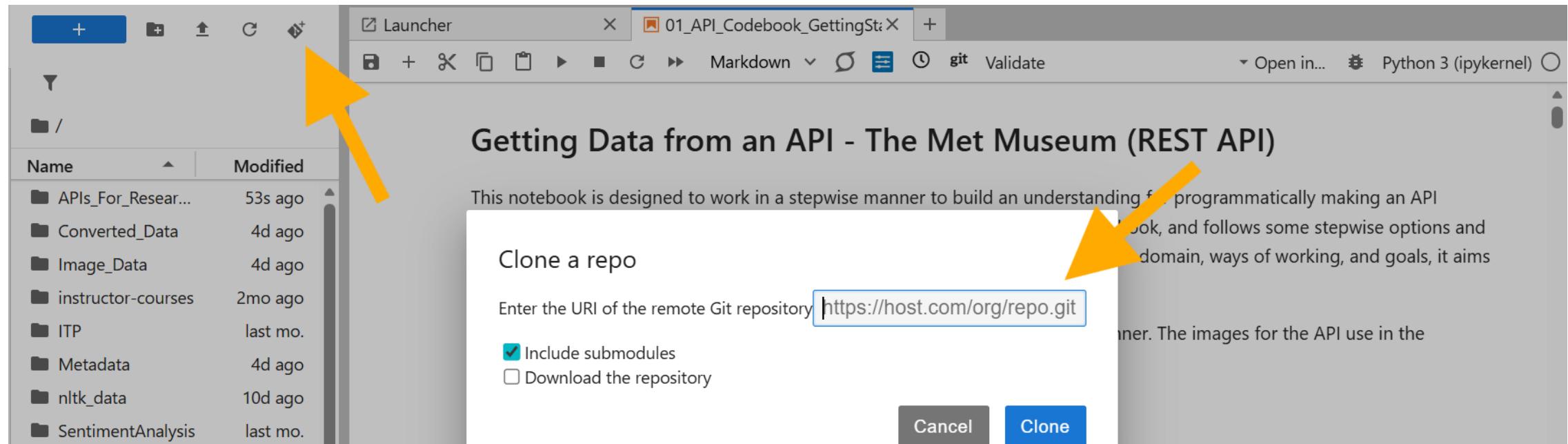
Download ZIP

About

In this repository you are going to find the documents we produced to support the upcoming course on API use for Research. These documents/files will help you to line up activities from the in-class content. Go to the readme file to start with.

Readme Activity Custom properties 0 stars





This notebook is designed to work in a stepwise manner to build an understanding of programmatically making an API request. It follows some stepwise options and domain, ways of working, and goals, it aims to teach the user how to use the API.

Getting Data from an API - The Met Museum (REST API)

Clone a repo

Enter the URI of the remote Git repository

Include submodules
 Download the repository

Cancel Clone

Name	Modified
APIs_For_Research	53s ago
Converted_Data	4d ago
Image_Data	4d ago
instructor-courses	2mo ago
ITP	last mo.
Metadata	4d ago
nltk_data	10d ago
SentimentAnalysis	last mo.



Q & A



www.ccds.ed.ac.uk

