Introduction to Dataverse APIs

Gustavo Durand, IQSS, Harvard University

Jim Myers, GDCC

The Dataverse® Project

IQSS
The Institute for Quantitative Social Science

HARVARD LIBRARY

HARVARD UNIVERSITY
Information Technology

The Dataverse® Project

Gustavo Durand
Technical Lead and Architect of Dataverse

Jim Myers
GDCC Senior Developer, Architect

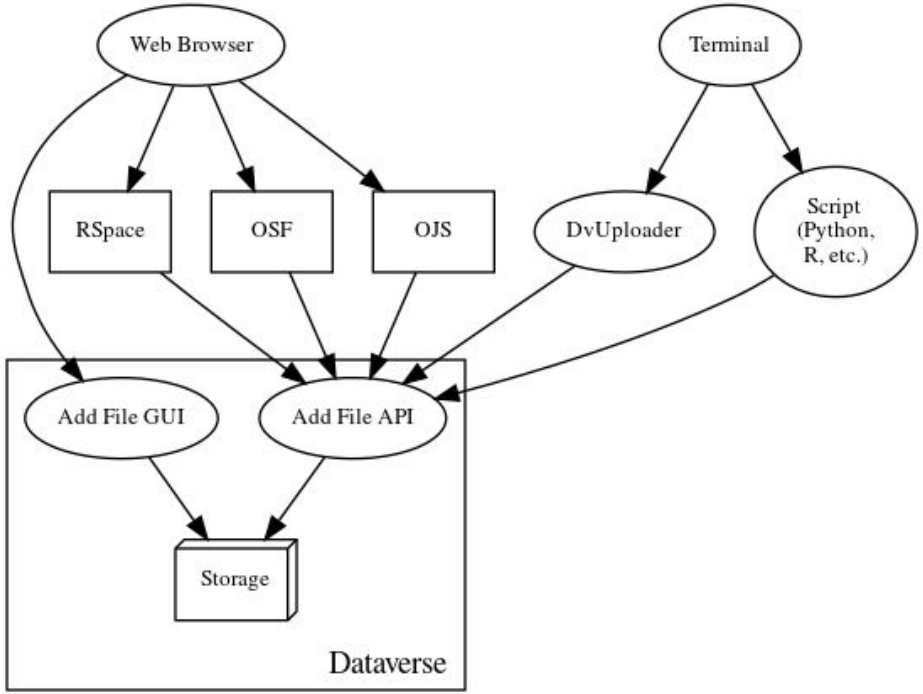
Agenda

- What and Why of APIs
- Modularity in Dataverse
- Using APIs
- Tools that Use APIs
  - Tools for Adding Many/Large Files to Dataverse
- Frontend Rearchitecture

What and Why of APIs

What is an API?

API stands for “Application Programming Interface”



Why APIs?

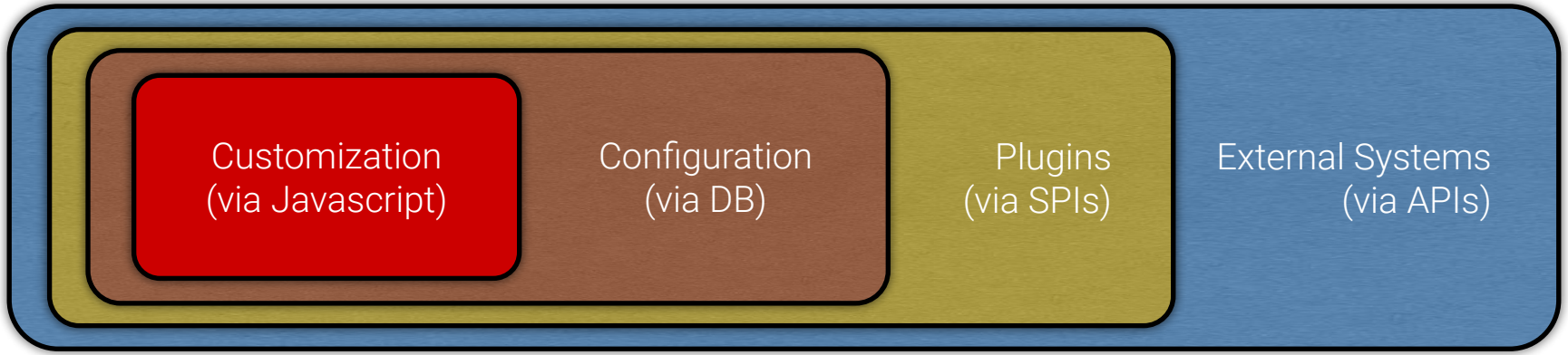
- Enable modularity in the design, development, and deployment of Dataverse, to support multiple types of data, users, and workflows
- Allow programmatic access for bulk access
- Allow interoperability with “external tools” and other repositories / software

Modularity in Dataverse

Modularity within Dataverse

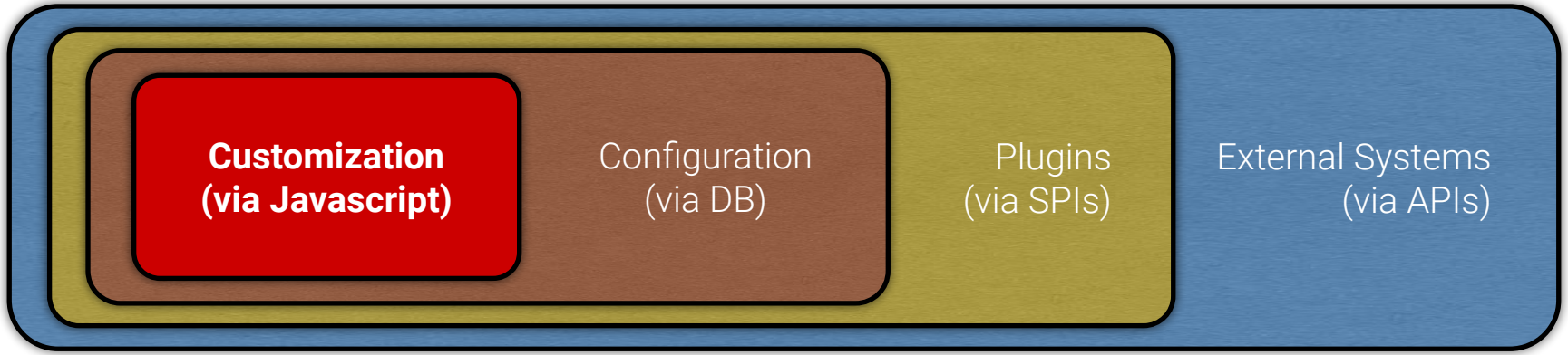
In order for institutions to use the Dataverse software with different workflows, different domains, and different organizational models, we needed to allow flexibility in the way to configure key aspects of the software.

Additionally, we designed the Dataverse software itself to focus on the core functionality for a data repository, namely publishing, versioning, sharing, and citing, while allowing easy interoperability with other tools for exploration and visualization.



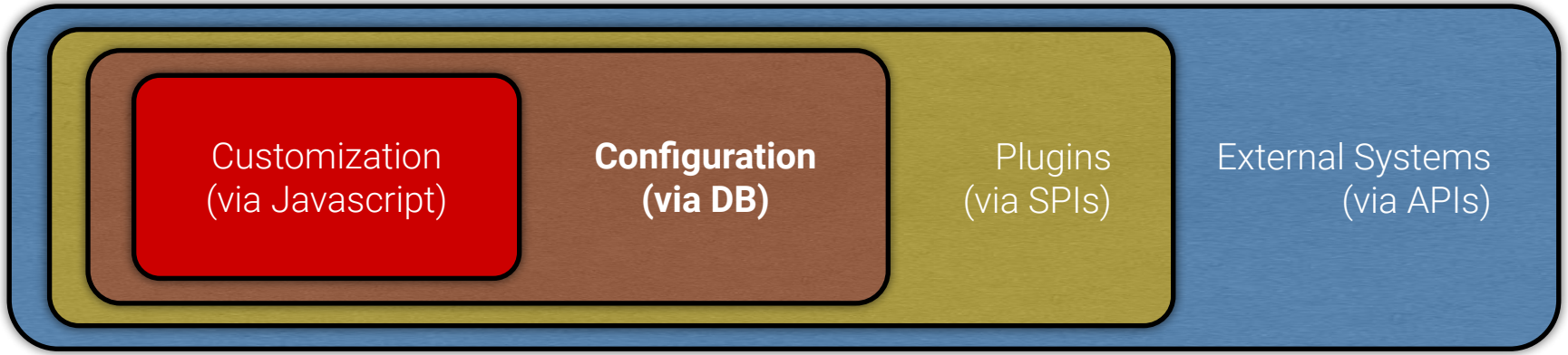
Customization (via Javascript)

Dataverse allows admins to customize their installations with HTML/Javascript in a few areas.



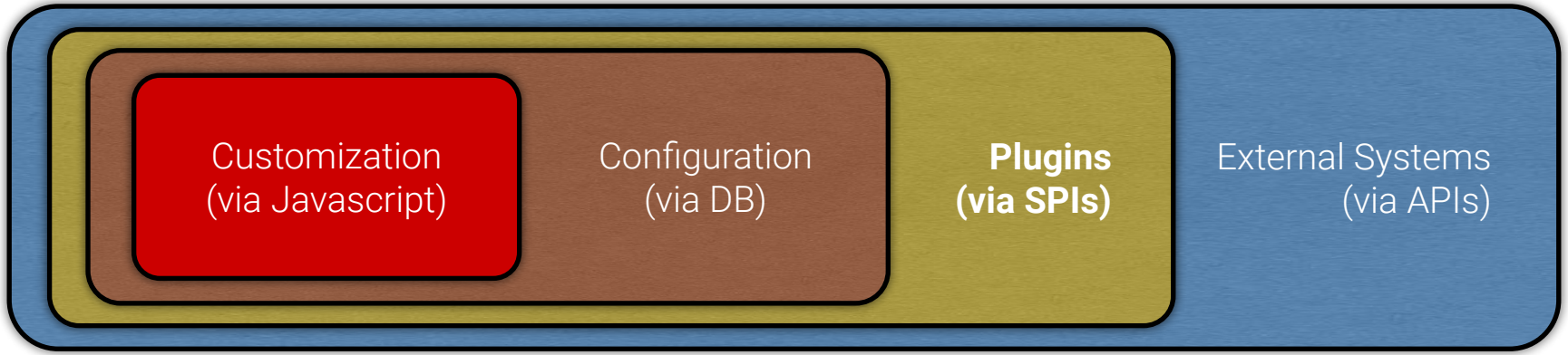
Configuration (via DB)

Several areas of functionality are defined by configuration via the database, rather than in the code itself, allowing the same code to be deployed by different institutions with different needs.



Plugins (via SPIs)

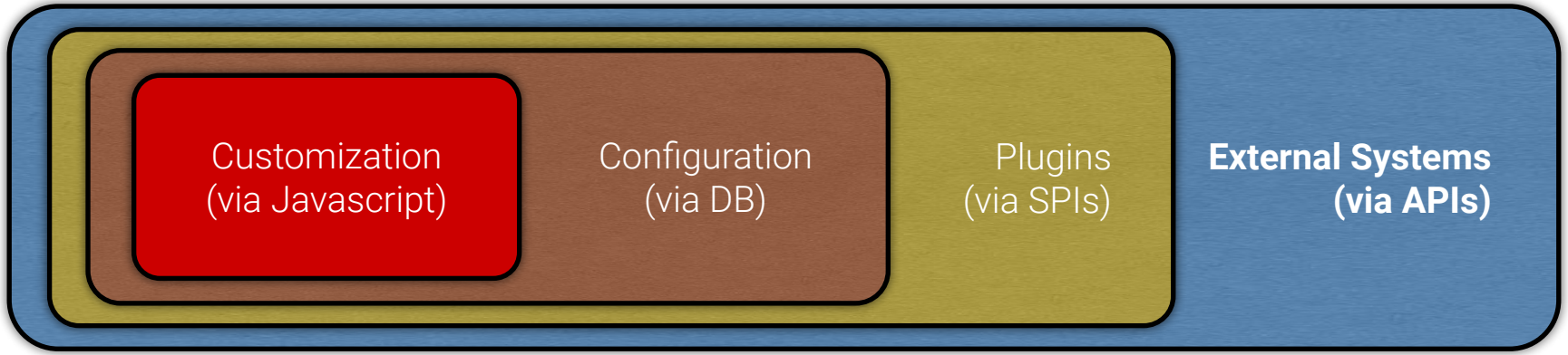
Plugins allow developers to extend the functionality of the core code without having to make a separate fork of the repository. In Dataverse, we enable this via the SPI (Service Provider Interface) model.



External Systems (via APIs)

From Dataverse 4 onward, APIs have been a major focus of the software and a majority of the functionality that is available via the UI is also available via API.

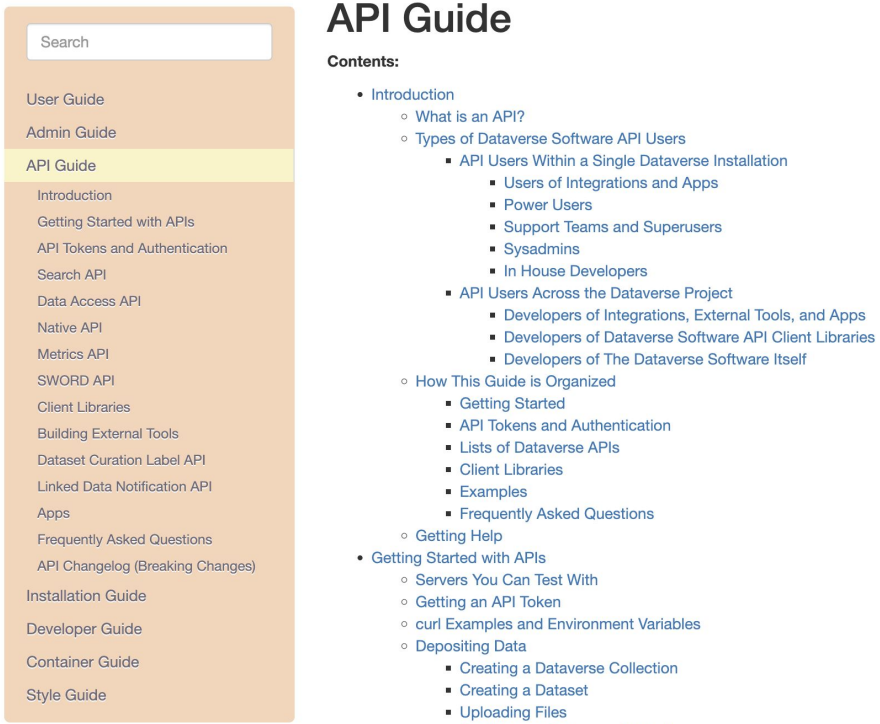
This allows external developers to develop other applications, which we often refer to as **external tools**, using whatever technology is most effective for their purpose.



Using APIs

API Guide

https://guides.dataverse.org/en/latest/api



API Guide

Contents:

- Introduction
  - What is an API?
  - Types of Dataverse Software API Users
    - API Users Within a Single Dataverse Installation
      - Users of Integrations and Apps
      - Power Users
      - Support Teams and Superusers
      - Sysadmins
      - In House Developers
    - API Users Across the Dataverse Project
      - Developers of Integrations, External Tools, and Apps
      - Developers of Dataverse Software API Client Libraries
      - Developers of The Dataverse Software Itself
  - How This Guide is Organized
    - Getting Started
    - API Tokens and Authentication
    - Lists of Dataverse APIs
    - Client Libraries
    - Examples
    - Frequently Asked Questions
  - Getting Help
- Getting Started with APIs
  - Servers You Can Test With
  - Getting an API Token
  - curl Examples and Environment Variables
  - Depositing Data
    - Creating a Dataverse Collection
    - Creating a Dataset
    - Uploading Files
    - Publishing a Dataverse Collection

How to Access APIs

- From a command line, use: `cURL`
  - stands for client URL
  - a command line tool that developers use to transfer data to and from a server
  - response will (usually) be in json format
  - case insensitive, so in the examples you will typically just see `curl`

Components of a API call

```
curl -H "X-Dataverse-key:xxxxxxxx-xxxx-xxxx-xxxx-xxxxxxxxxxxx" -X POST
"https://demo.dataverse.org/api/dataverses/argentina/datasets" --upload-file messi-10.json
-H 'Content-type:application/json'
```

Components of a API call

```
curl -H "X-Dataverse-key:xxxxxxxx-xxxx-xxxx-xxxx-xxxxxxxxxxxx" -X POST
"https://demo.dataverse.org/api/dataverses/argentina/datasets" --upload-file messi-10.json
-H 'Content-type:application/json'
```

- **Endpoint**
  - the Dataverse server you're contacting
  - for our examples, we'll be using: https://demo.dataverse.org

Components of a API call

```
curl -H "X-Dataverse-key:xxxxxxxx-xxxx-xxxx-xxxx-xxxxxxxxxxxx" -X POST
"https://demo.dataverse.org/api/dataverses/argentina/datasets" --upload-file messi-10.json
-H 'Content-type:application/json'
```

- **An HTTP method**
  - GET (default) is used to retrieve information or a resource from a server
  - POST is used to sends data to the server and creates a new resource
  - PUT is most often used to update an existing resource
  - DELETE is used to delete a resource

Components of a API call

```
curl -H "X-Dataverse-key:xxxxxxxx-xxxx-xxxx-xxxx-xxxxxxxxxxxx" -X POST
"https://demo.dataverse.org/api/dataverses/argentina/datasets" --upload-file messi-10.json
-H 'Content-type:application/json'
```

- **Body**
  - contains the data that we want to send
  - Generally, used with POST AND PUT

Components of a API call

```
curl -H "X-Dataverse-key:xxxxxxxx-xxxx-xxxx-xxxx-xxxxxxxxxxxx" -X POST
"https://demo.dataverse.org/api/dataverses/argentina/datasets" --upload-file messi-10.json
-H 'Content-type:application/json'
```

- **Additional Headers**
  - These contain metadata about the request
  - Examples:
    - API token
    - Content type of body

Public APIs

- APIs which anyone can access
  - Basic info about the installation (e.g. server, version)

```
curl "https://demo.dataverse.org/api/info/version"
```

  - Some Settings
    - Example: Custom Popup Text for Publishing Datasets
    - Example: Maximum Embargo Duration In Months

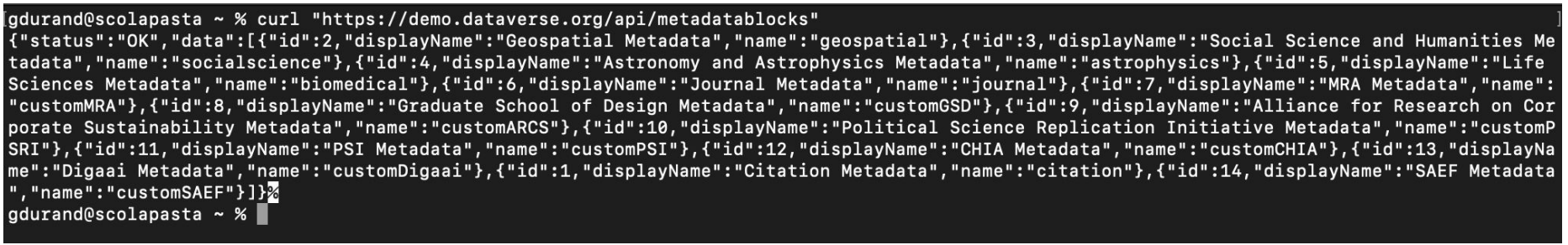
```
curl "https://demo.dataverse.org/api/info/settings/:DatasetPublishPopupCustomText"
curl "https://demo.dataverse.org/api/info/settings/:MaxEmbargoDurationInMonths"
```

  - Other Configuration Info
    - Metadata Blocks
    - Info About Single Metadata Block

```
curl "https://demo.dataverse.org/api/metadatablocks"
curl "https://demo.dataverse.org/api/metadatablocks/citation"
```

Sample Response (raw)

```
curl "https://demo.dataverse.org/api/metadatablocks"
```



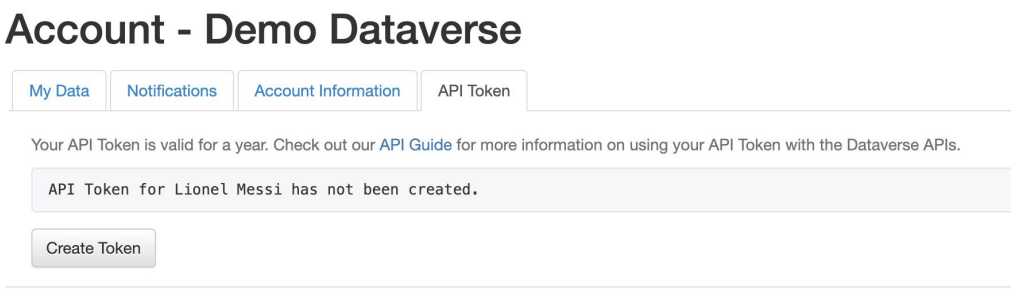
Sample Response (with “jq”)

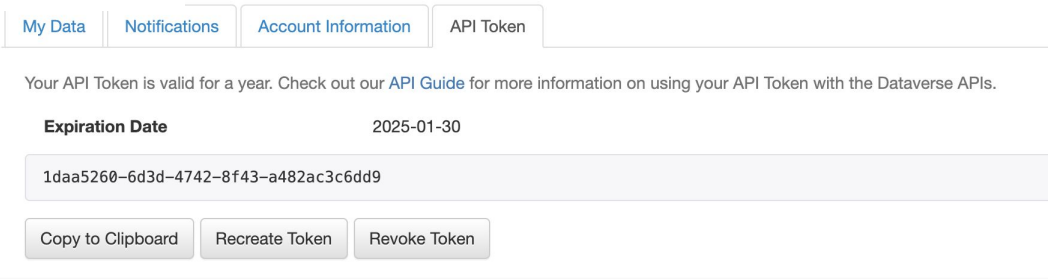
```
curl "https://demo.dataverse.org/api/metadatablocks" |jq
```

```
gdurand@scolapasta ~ % curl "https://demo.dataverse.org/api/metadatablocks" | jq
  % Total    % Received % Xferd  Average Speed   Time    Time     Time  Current
                                 Dload  Upload   Total   Spent    Left  Speed
100  1025  100  1025    0     0  13334      0 --:--:-- --:--:-- --:--:-- 13311
{
  "status": "OK",
  "data": [
    {
      "id": 2,
      "displayName": "Geospatial Metadata",
      "name": "geospatial"
    },
    {
      "id": 3,
      "displayName": "Social Science and Humanities Metadata",
      "name": "socialscience"
    },
    {
      "id": 4,
      "displayName": "Astronomy and Astrophysics Metadata",
      "name": "astrophysics"
    },
    {
      "id": 5,
      "displayName": "Life Sciences Metadata",
      "name": "biomedical"
    },
    {
```

API Token

- Many APIs are not fully public; you will need a **API Token**
- An API token is similar to a password and allows you to authenticate to Dataverse Software APIs to perform actions as you
- Your API token is unique to the server you are using





Search APIs

- The Search API supports the same searching operations as UI
- Required Parameter “q” - the query

```
curl https://demo.dataverse.org/api/search?q=messi
```

Search APIs

- Full set of optional parameters for sorting, faceting, highlighting, and other operations
  - Example (narrowed to Show Relevance and Facets):

```
curl https://demo.dataverse.org/api/search?q=messi&show_relevance=true&show_facets=true&fq=publicationDate:2022
```

- To also search unpublished content, you must pass in an API token
  - Example (Narrowed to only datasets within a specific collection):

```
curl -H X-Dataverse-key:xxxxxxxx-xxxx-xxxx-xxxx-xxxxxxxxxxxx https://demo.dataverse.org/api/search?q=messi&type=dataset&subtree=argentina
```

Access APIs

- Basic access URI: `/api/access/datafile/$id`
- Download a single file (Note: use "-L" header to follow redirects):

```
curl -L https://demo.dataverse.org/api/access/datafile/10
```

- Download just a range of that file:

```
curl -H "Range:bytes=0-9" https://demo.dataverse.org/api/access/datafile/10
```

- Download multiple files at once (as a .zip file):

```
curl -L https://demo.dataverse.org/api/access/datafiles/10,11,12 --output arg.zip
```

- Download all files from a dataset (as a .zip file):

```
curl -L -O -J -H X-Dataverse-key:xxxxxxxx-xxxx-xxxx-xxxx-xxxxxxxxxxxx
https://demo.dataverse.org/api/access/dataset/:persistentId/?persistentId=doi:10.70122/FK2/N2XGBJ
```

Deposit APIs

- Create a Dataverse Collection

```
curl -H "X-Dataverse-key:xxxxxxxx-xxxx-xxxx-xxxx-xxxxxxxxxxxx" -X POST
"https://demo.dataverse.org/api/dataverses/argentina" --upload-file
dataverse-argentina.json
```

- Deposit a Dataset

```
curl -H "X-Dataverse-key:xxxxxxxx-xxxx-xxxx-xxxx-xxxxxxxxxxxx" -X POST
"https://demo.dataverse.org/api/dataverses/argentina/datasets" --upload-file
messi-10.json -H 'Content-type:application/json'
```

Dataverse Collection sample json

```
{
  "name": "Argentina Dataverse Collection" ,
  "alias": "argentina",
  "dataverseContacts": [
    {
      "contactEmail": "scaloni@argentina.org"
    },
    {
      "contactEmail": "messi@argentina.org"
    }
  ],
  "affiliation": "Argentina",
  "description": "Argentina National Team Research." ,
  "dataverseType": "ORGANIZATIONS_INSTITUTIONS"
}
```

Dataset sample json


```
{
  "datasetVersion": {
    "license": {
      "name": "CC0 1.0",
      "uri": "http://creativecommons.org/publicdomain/zero/1.0"
    },
    "metadataBlocks": {
      "citation": {
        "fields": [
          {
            "value": "Messi",
            "typeClass": "primitive",
            "multiple": false,
            "typeName": "title"
          },
          {
            "value": [
              {
                "authorName": {
                  "value": "Messi, Lionel",
                  "typeClass": "primitive",
                  "multiple": false,
                  "typeName": "authorName"
                }, ...
```

Administration APIs (user)

- APIs for administering your dataverse collections and datasets
- Examples:
  - List Role Assignments in a Dataset

```
curl -H "X-Dataverse-key:xxxxxxxx-xxxx-xxxx-xxxx-xxxxxxxxxxxx"
"https://demo.dataverse.org/api/datasets/2347/assignments"
```

  - Delete a Dataverse Collection

```
curl -H "X-Dataverse-key:xxxxxxxx-xxxx-xxxx-xxxx-xxxxxxxxxxxx" -X DELETE
"https://demo.dataverse.org/api/dataverses/brasil"
```

Administration APIs (superuser)

- APIs for some advanced configurations or user management
- Examples:
  - Configure a Dataset to Store All New Files in a Specific File Store

```
curl -H "X-Dataverse-key: xxxxxxxx-xxxx-xxxx-xxxx-xxxxxxxxxxxx" -X PUT -d largeBucket
"https://demo.dataverse.org/api/datasets/1022/storageDriver"
```

  - Change User Identifier

```
curl -H "X-Dataverse-key: xxxxxxxx-xxxx-xxxx-xxxx-xxxxxxxxxxxx" -X POST
"https://demo.dataverse.org/api/users/messi/changeIdentifier/lionelmessi"
```

Administration APIs (installation admin)

- Only available with direct access to server
- Don’t require API token (actions are not “user” actions)
- Examples:
  - Reindex a dataset

```
curl http://localhost:8080/api/admin/index/dataverses/10
```

  - Delete cached metrics results

```
curl - X DELETE http://localhost:8080/api/admin/clearMetricsCache
```

Other APIs

- **Migration APIs**
  - Native Migration APIs for **json** and **ddi** imports
  - Sword API
    - “Simple Web-service Offering Repository Deposit”
    - interoperability standard created in 2007
- **Metric APIs**
  - The Metrics API provides counts of downloads, datasets created, files uploaded, and more
  - Used to power https://dataverse.org/metrics
- **Harvesting APIs**
  - Dataverse expose structured metadata via OAI-PMH
    - OAI-PMH is a set of six verbs or services that are invoked within HTTP
    - Dataverse can harvest remote datasets from other installations or other repositories that support OAI-PMH

Client Libraries

- Several client libraries have been created to help developers interact with Dataverse APIs from other languages, including:
  - C/C++
  - Go
  - Java
  - Javascript
  - Julia
  - PHP
  - Python
  - R
  - Ruby

API Guide (Native API contents)

https://guides.dataverse.org/en/latest/api

Tools that use APIs

External Tools

- Tools that talk to Dataverse
  - generally used to deposit data into Dataverse (via Deposit API)
  - usually don't require anything special to be set up in the Dataverse repository
- Tools that Dataverse talks to
  - user starts on Dataverse and is directed to the external tool
    - require manifest files
  - have predefined areas in the UI where these would plug into (**Explore** tools)
  - **OR**, are embedded into the Dataverse UI directly (**Preview** tools and **Query** Tools)
- Tools that do both
  - user starts on Dataverse and is directed to the external tool
    - require manifest files
  - also have predefined areas in the UI where these would plug into (**Configure** tools)
  - will also send something back to Dataverse, so need an API token that has "write" privileges

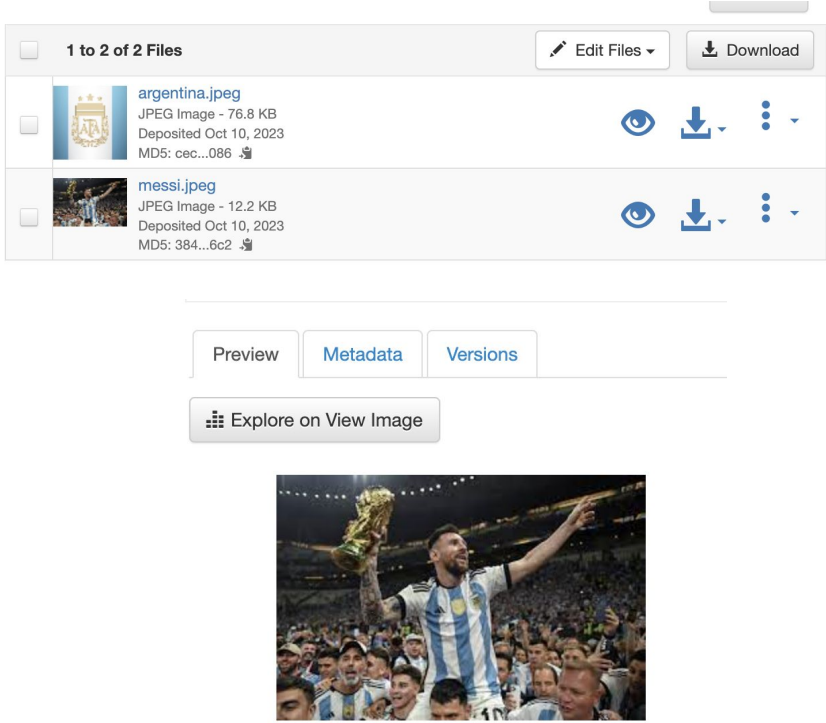
External Tools Manifest

- External tools must be expressed in an external tool manifest file
- Can be uploaded to a Dataverse installation via API

```
{
  "displayName": "Awesome Tool",
  "description": "The most awesome tool.",
  "type": "explore",
  "toolUrl": "https://awesometool.com",
  "toolParameters": {
    "queryParameters": [
      {
        "fileid": "{fileId}"
      },
      {
        "key": "{apiToken}"
      }
    ]
  }
}
```

File Previewers

- A set of tools that display the content of files, allowing them to be viewed without downloading the file, including
- audio
- html
- Hypothes.is annotations
- images
- PDF
- text
- video
- tabular data
- spreadsheets
- GeoJSON
- Zip files
- NcML files
- Previewers are available through the preview (eye) icon on Dataset pages
- And also embedded as a tab on Datafile pages



File Previewers (more examples)

Preview | Metadata | Versions

Explore on Read Document

Previous | Next | Page: 13 / 59

GEOMETRICAL SOLUTIONS DERIVED FROM MECHANICS.

ARCHIMEDES TO ERATOSTHENES, GREETING:

Some time ago I sent you some theorems I had discovered, writing down only the propositions because I wished you to find their demonstrations which had not been given. The propositions of the theorems which I sent you were the following:

1. If in a perpendicular prism with a parallelogram[1] for base a cylinder is inscribed which has its bases in the opposite parallelograms[1] and its surface touching the other planes of the prism, and if a plane is passed through the center of the circle that is the base of the cylinder and one side of the square lying in the opposite plane, then that plane will cut off from the cylinder a section which is bounded by two planes, the intersecting plane and the one in which the base of the cylinder lies, and also by as much of the surface of the cylinder as lies between these same planes: and the detached

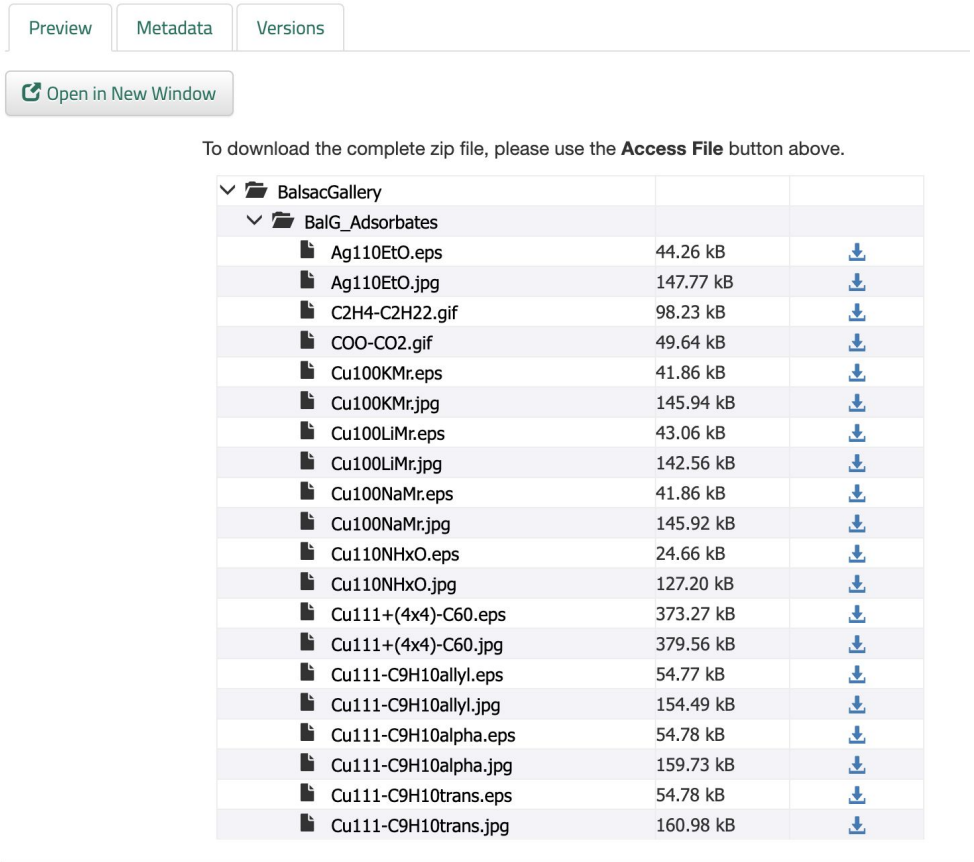
File Tools | Metadata | Versions

File Tools | Open in New Window

	NameFootballPlayer	International
1	Pedro Bonifacio Suárez Pérez \"Arico\" (Pedro Suárez)	Argentina
2	Milovan \"El Grande Milovan\" Jakšić	Yugoslavia
3	Ernest Libérati	France
4	Alexandre Villaplane (Captain)	France
5	Roberto Gayón Márquez	Mexico
6	Andrew \"Andy\" Auld	United States
7	James \"Jim\" Brown	United States
8	Jimmy Gallagher	United States
9	Alfred Eisenbeisser	Romania
10	Ladislau Raffinsky	Romania
11	Bartholomew \"Bertie\" or \"Bart\" McGhee (Bart McGhee)	United States
12	George Moorhouse	United States
13	Alexander “Alec” Wood	United States
14	Lorenzo Fernández	Uruguay
15	Constantino Urbieta Sosa	Argentina
16	Štefan Čambal	Czechoslovakia
17	Ferdinand Daučík	Czechoslovakia
18	Géza Kalocsay	Czechoslovakia
19	František Svoboda	Czechoslovakia
20	Matthias Sindelar (born as Matěj Šindelář)	Austria
21	Joseph Alcazar	France
22	Roger Courtois	France

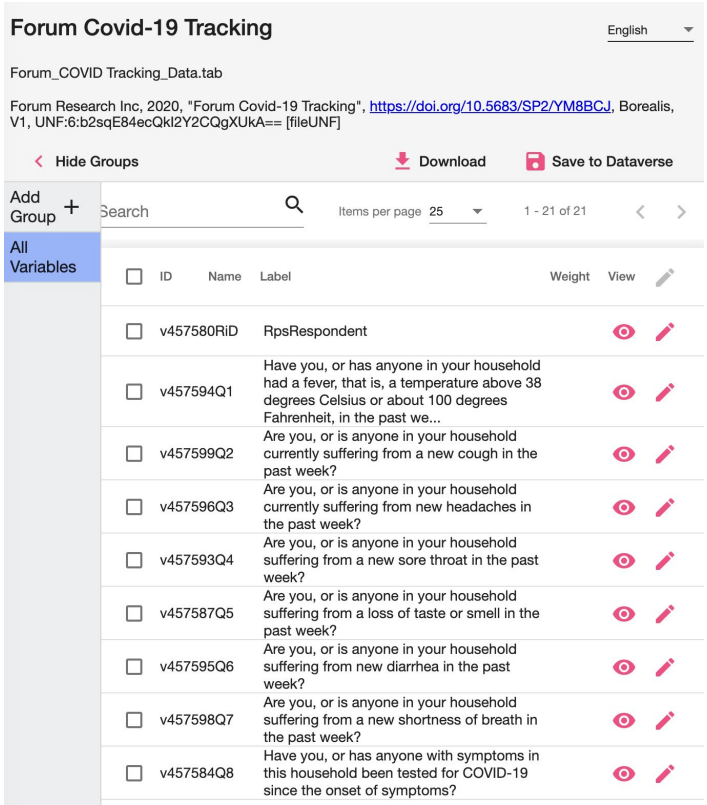
Zip File Previewer +

- A previewer that will show you the internal content and structure of a zip file (or electronic lab notebook)
- Uses the Range functionality in our Access api; so it's not just a viewer, it's an individual file unpacker and downloader too



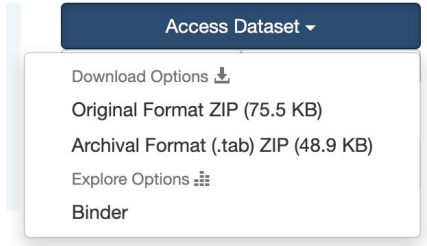
File Exploration, Configuration, and Query Tools

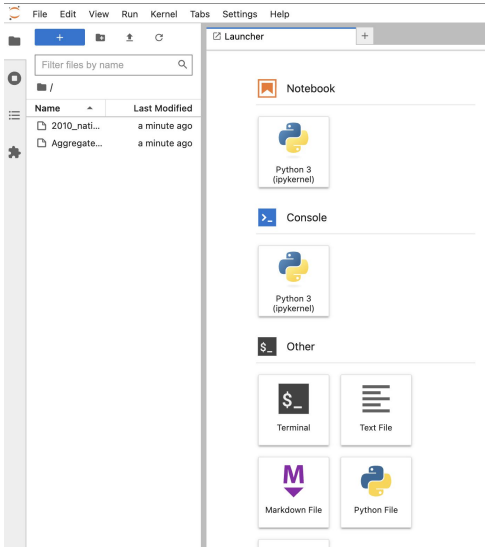
- File level **explore** tools provide a variety of features from data visualization to statistical analysis
- File level **query** tools allow the user to ask questions (e.g. natural language queries) of a data table's contents without having to download the file
- File level **configure** tools allow (authorized) users to send metadata about the file back to Dataverse



Dataset External Tools

- Dataset level **explore** tools allow the user to explore all the files in a dataset - common use case is reproducibility
  - **WholeTale** - creates reproducible research packages based on popular tools such as Jupyter and RStudio
  - **Binder** - spins up custom computing environments in the cloud (including Jupyter notebooks)
- Dataset level **configure** tools allow (authorized) users to send metadata about the dataset back to Dataverse
  - **Turbo Curator** *(coming soon)* - uses Open AI's ChatGPT & ICPSR best practices to provides recommendation to enhance metadata & generate meaningful titles, descriptions, and keywords





Tools for Adding Many/Large Files to Dataverse

Use Cases Where Other Tools Are Useful

- You have many (100+) files to upload
- You have large files that may take hours to upload
- You have files in many subdirectories / want files to have paths in Dataverse
- You want to sync a local directory tree with a dataset / upload only new files
- You want to include or exclude files with a given name pattern
- You want to automate file transfer
- You don’t have a browser where the files are

Available Tools

- DVWebLoader
  - Easiest, configured as an option in the Dataverse UI
- Python DVUploader
  - Simple command-line tool/library
- DVUploader
  - Most comprehensive, but ugliest - a reference implementation exercising the Dataverse API

Note:

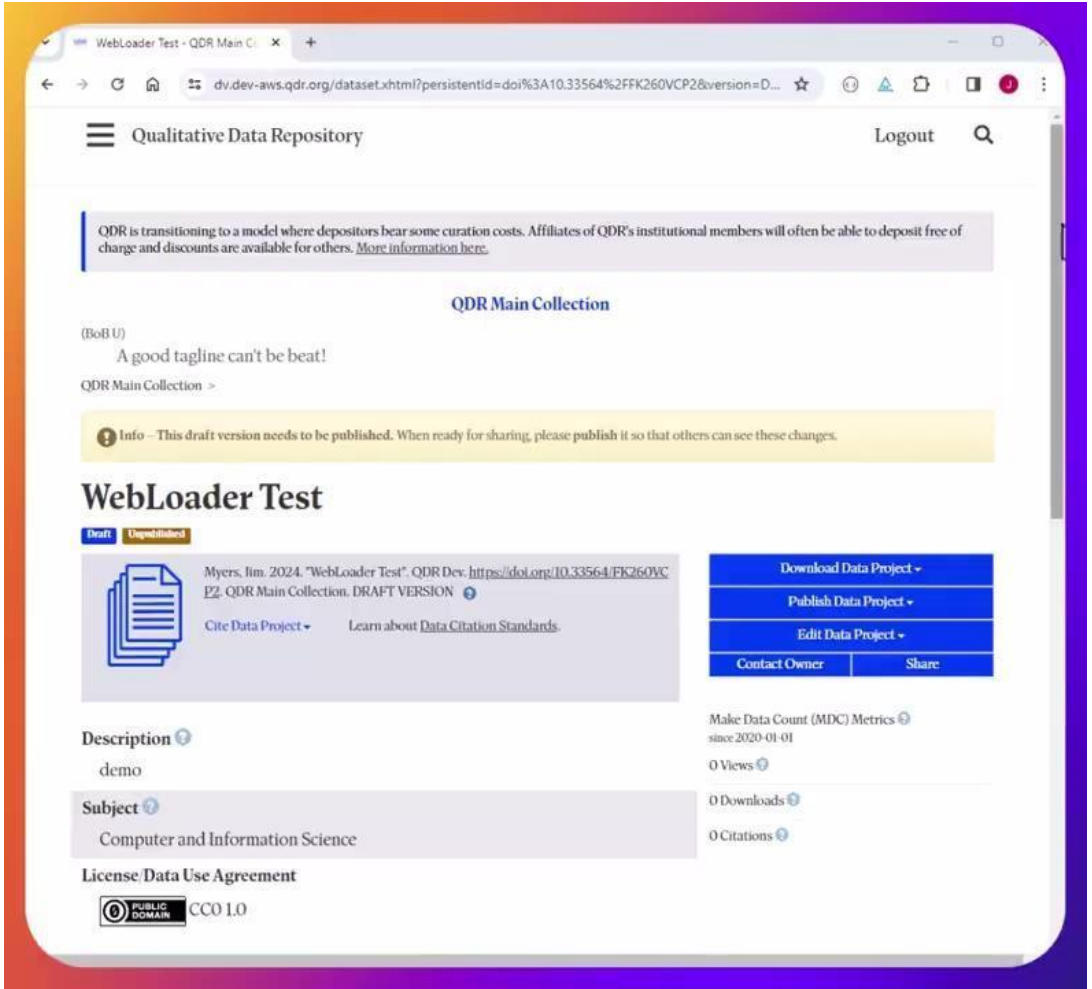
All these tools are most-efficient when Dataverse has been configured to use S3 storage and direct uploads. Some only support this case.

All use the documented Dataverse API

DVWebloader

- Click “Upload a Folder”
- Select Directory
- Agree to pop-up
- Adjust upload list
- Click “Start Upload”
- Return to Dataverse when complete

* Must be configured at your installation



Python DVUploader

Command-line or as a library

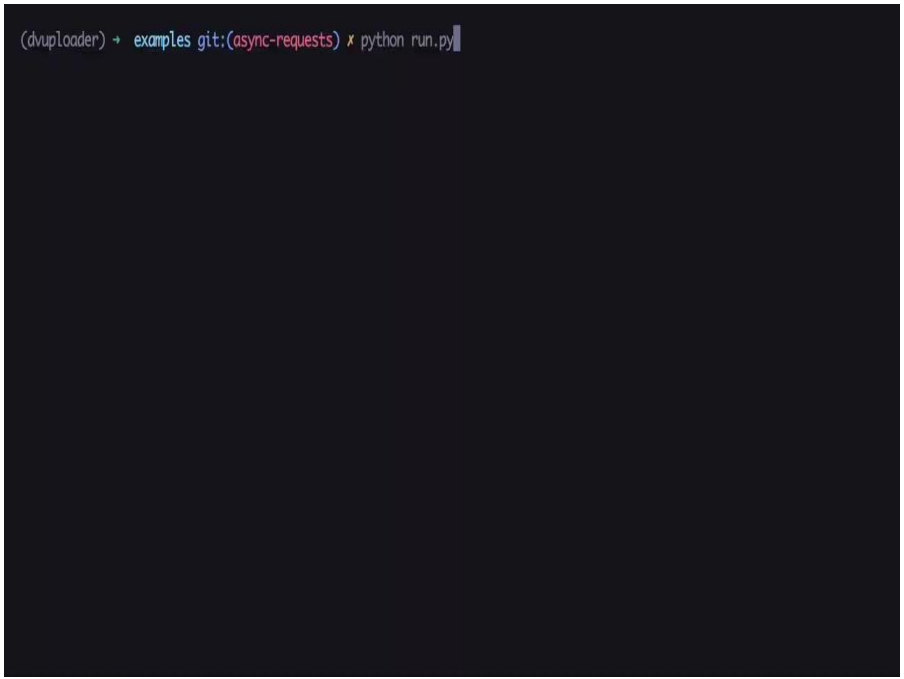
- File list:

```
dvuploader my_file.txt my_other_file.txt \
          --pid doi:10.70122/XXX/XXXXX \
          --api-token
XXXXXX-XXXX-XXXX-XXXX-XXXXXXXXXXXX \
            --dataverse-url
  https://demo.dataverse.org/
```

- Config file - with added metadata

```
dvuploader --config-path config.yml
```

- `filepath`: Path to the file to upload.
- `directoryLabel`: Optional directory label
- `description`: Optional description of the file.
- `mimetype`: Mimetype of the file.
- `categories`: Optional list of categories
- `restrict`: whether to restrict access or not



DVUploader

Java Command-line tool

- Requires Java and downloading the DVUploader jar file

```
java -jar DVUploader-v1.2.0beta3.jar -key=<api key> -did=<dataset doi> -server=<server URL> <dir or file names>
```

- Options
  - -listonly
  - -limit=<X>
  - -ex=<regexp>
  - -verify
  - -recurse
  - -maxlockwait=<X>
  - -uploadviaserver
  - -trustall
  - -singlefile
  - -skip=<X>
  - -forcenew

Detailed logging

Also supports re-creating datasets from RDA-conformant archival Bags

```
Dataverse Mode: Uploading files to a Dataverse instance
Using apiKey: 8599b802-659e-49ef-823c-20abd8efc05c
Adding content to: doi:10.5072/FK2/TUNNVEUsing server:
https://dataverse.tdl.orgRequest to upload: testdir

PROCESSING(C): testdirFound as: doi:10.5072/FK2/TUNNVE

PROCESSING(D): testdir\Capture3.JPGDoes not yet exist
on server.UPLOADED as:
MD5:b2d8726f4ddba30705259143dbb283e3CURRENT TOTAL: 1
files :9506 bytes

PROCESSING(D): testdir\Capture4.GIFDoes not yet exist
on server.UPLOADED as:
MD5:3b9b536bd0abaf9c2677846f62d77ed9CURRENT TOTAL: 2
files :23973 bytes

PROCESSING(D): testdir\Capture5.PNGDoes not yet exist
on server.UPLOADED as:
MD5:ce26585c19bd1470b7229b2cfcc879f0CURRENT TOTAL: 3
files :35448 bytes
```

The Dataverse Direct Upload API

- Request a signed URL(s) from Dataverse for a given file (more than one URL if multipart upload is required)
- Perform the upload(s) to the S3 store
  - For multipart, make the api call to abort/complete it
- Call Dataverse to add the file, providing the name, description, hash value, directoryLabel, etc.
  - Optionally tell Dataverse to add many files at once
  - Optionally tell Dataverse to replace an existing file with the new one

Notes

- Handling larger files/more files requires Dataverse to be configured and resourced appropriately - if you are pushing boundaries, expect to coordinate with the Dataverse admins (TLDR: don’t upload TBs w/o asking!)
- Don’t run multiple copies
  - All of these tools decide what to upload based on the current dataset contents
  - They already parallelize uploads
- If you have really, really big data, consider Globus and/or referencing data at remote sites (requires additional setup at the installation)

Acknowledgements

DVuploader has been supported by SEAD, QDR, GDCC, RDA

DVWebloader was initiated by DataverseNO

Python DVUploader was initiated by Jan Range

All are open source, available on github, and welcome contributions from anyone in the Dataverse community

Frontend Rearchitecture

Frontend Rearchitecture Project

- Separate frontend and backend applications
  - API-first Dataverse backend
  - Single Page Application (SPA) front end
- Benefits
  - modernize the frontend technologies
  - speed up development of new UI/UX ideas
  - empower the community
  - extend modularity
- **100% of functionality available via API**
- Initial beta release planned for Q2 2024

Thank you

Dataverse Community Meeting 2024
March 4-8, 2024
CIMMYT in Texcoco, Mexico

CIMMYT MR
International Maize and Wheat Improvement Center

The Dataverse® Project

Open source research data repository software

Researchers

Enjoy full control over your data. Receive *web visibility, academic credit,* and *increased citation counts.* A personal dataverse is easy to set up, allows you to display your data on your personal website, can be branded uniquely as your research program, makes your data more discoverable to the research community, and satisfies data management plans. Want to set up your personal dataverse?

Journals

Seamlessly manage the submission, review, and publication of data associated with published articles. Establish an *unbreakable link* between *articles in your journal* and *associated data.* Participate in the open data movement by using Dataverse as part of your journal data policy or list of repository recommendations. Want to find out more about journal dataverses?

Institutions

Establish a research data management solution for your community. Federate with a growing list of Dataverse repositories worldwide for increased discoverability of your community's data. Participate in the drive to set norms for sharing, preserving, citing, exploring, and analyzing research data. Want to install a Dataverse repository?

Developers

Participate in a vibrant and growing community that is helping to drive the norms for sharing, preserving, citing, exploring, and analyzing research data. Contribute code extensions, documentation, testing, and/or standards. *Integrate research analysis, visualization* and *exploration tools,* or other research and data archival systems with Dataverse. Want to contribute?

https://dataverse.org
https://github.com/iqss/dataverse