



# Practical Knowledge Graph Example

Protege,  
Stardog and  
Peeps



# Today's exercise

1. Look at a simple ontology for information about people and their relations in Protégé
2. Look at some instance data in Protégé
3. Run the DL and rule reasoner in Protégé
4. Load the ontology and data into Stardog
5. Browse and query the resulting knowledge graph in Stardog

# Preliminaries

- On your own computer (Windows, Mac, Linux)
  - Download and install Protégé
  - Download, install and configure the community edition of Stardog 5
  - Clone the 691 peeps repository

# Peeps files

- The peeps repo has five files
  - **README.md**
  - **catalog-v001.xml** – protégé config file
  - **load\_peeps.sh** – bash script to load peeps into stardog
  - **mypeeps.ttl** – data encoded using peeps ontology
  - **peeps.ttl** – the peeps ontology
  - **prefixes.ttl** – list of prefixes, used by stardog’s query component

# Separate ontology and data?

- An ontology is a knowledge graph schema
  - peeps:Man owl:disjointWith peeps:Woman .
- We talk about populating it with instance data
  - :janeDoe a peeps:Woman; foaf:givenName “Jane” .
- Good practice for real applications is to keep the ontology and data separate
  - i.e., in different files
- Hence, peeps.ttl and mypeeps.ttl

# Why separate ontology and data?

- It really depends on the usecase
- Some facts are part of an ontology if they're important, unchanging knowledge
- Maybe the ontology is a one-off, and will never be used with any other data
- Maybe you added data while developing the ontology for testing and debugging
- But many ontologies are intended for reuse or to represent datasets that change frequently

# Namespaces

- Promoting reuse also entails giving the ontology and a knowledge graph that uses it with different namespaces
- Namespace = uri = unique identifier
- Example
  - <http://dbpedia.org/resource/>
  - <http://dbpedia.org/ontology/>
- BTW, lookup prefixes at <http://prefix.cc>
- Ideally, the uris are ones you control and no one else will use

# Namespace best practice

- Ideally, the namespace should resolve to a file containing the ontology or data
  - Maybe not the data if it's big or proprietary
- Enables other ontologies to **import and use** yours just from its URI
- If you don't control a long-lived URI ...
  - You might use a file on github
  - You might use purl to create a “permanent url” that redirects to the current location

# Peeps.ttl in Protégé

The screenshot shows the Protégé ontology editor interface with the following details:

- Title Bar:** peeps.ttl (<https://raw.githubusercontent.com/UMBC-CMSC-491-691-F18-Knowledge-Graphs/master/peeps.ttl>) : [/Users/finin/Desktop...]
- Toolbars:** Standard Mac OS X toolbar with icons for back, forward, search, and file operations.
- Tab Bar:** Data Properties, Annotation Properties, Individuals by class, DL Query, SWRLTab, Active Ontology, Entities, Object Properties.
- Annotations Tab:** Active tab, showing:
  - Ontology header:** Ontology IRI: <https://raw.githubusercontent.com/UMBC-CMSC-491-691-F18-Knowledge-Graphs/master/peeps.ttl>
  - Ontology Version IRI:** e.g. <https://raw.githubusercontent.com/UMBC-CMSC-491-691-F18-Knowledge-Graphs/master/peeps.mast>
- Annotations Panel:** Shows annotations for the rdfs:label predicate. One annotation is present: "An example ontology for people created in Protege OWL 5.5".
- OWL/XML rendering:** Shows the ontology's XML representation.
- OWL functional syntax rendering:** Shows the ontology's functional syntax representation.
- Ontology imports:** Direct Imports (empty), Indirect Imports (empty).
- Imported ontologies:** Empty list.
- Rules Panel:** Shows four rules:
  - $\text{hasParent}(\text{?p1}, \text{?p2}), \text{Woman}(\text{?p2}) \rightarrow \text{hasMother}(\text{?p1}, \text{?p2})$
  - $\text{hasParent}(\text{?p1}, \text{?p2}) \rightarrow \text{youngerThan}(\text{?p1}, \text{?p2})$
  - $\text{hasAge}(\text{?p1}, \text{?a1}), \text{hasAge}(\text{?p2}, \text{?a2}), \text{lessThan}(\text{?a1}, \text{?a2}) \rightarrow \text{youngerThan}(\text{?p1}, \text{?p2})$
- Bottom Status Bar:** Git: master, To use the reasoner click Reasoner > Start reasoner, Show Inferences checked.

# Mypeeps.ttl

The screenshot shows the Protégé ontology editor interface for the file `mypeeps.ttl`. The top navigation bar includes tabs for Active Ontology, Entities, Individuals by class, and DL Query. Below the tabs are buttons for Annotations, Selected entailments, Rules, and Ontology prefixes. A toolbar with icons for search, zoom, and other functions is at the top right.

The main area displays the Ontology header, which includes the Ontology IRI (`https://raw.githubusercontent.com/UMBC-CMSC-491-691-F18-Knowledge-Graphs/peeps/master/mypeeps.ttl`) and the Ontology Version IRI (e.g., `https://raw.githubusercontent.com/UMBC-CMSC-491-691-F18-Knowledge-Graphs/peeps/master/mypeeps.ttl/1.0.0`). There is also a section for Annotations with a plus sign icon.

Below the header, there are tabs for Ontology imports, General axioms, RDF/XML rendering, OWL/XML rendering, and OWL functional syntax rendering. The Imported ontologies section shows a single direct import from the same GitHub repository, with the URL `<https://raw.githubusercontent.com/UMBC-CMSC-491-691-F18-Knowledge-Graphs/peeps/master/peeps.ttl>` and the label `peeps.ttl`. It also displays the Ontology IRI and Location.

At the bottom left, it says "Git: master". At the bottom right, there is a note to "To use the reasoner click Reasoner > Start reasoner" and a checked checkbox for "Show Inferences".

# When to import an ontology

- In Protégé, we import an ontology if we want a reasoner to understand its vocabulary
- It does not add the ontology to the file that will be saved
- Plus: the knowledge may be important or essential in testing
- Minus: big ontologies may add a lot of useless data
- Here mypeeps.ttl imports peeps, but not foaf or schema

# Stardog Graph Platform

The screenshot shows the official website for Stardog. At the top, there's a navigation bar with a logo of a blue and orange dog-like creature on the left, followed by links for 'PLATFORM', 'CUSTOMERS', 'RESOURCES', 'ABOUT', and a prominent orange 'DOWNLOAD' button. Below the navigation is a large headline: 'The Knowledge Graph Platform for the Enterprise'. Underneath this, a sub-headline reads: 'With Stardog you can unify, query, search, and analyze all your data. Say goodbye to data silos forever.' To the right of the text is a photograph of a brown and white dog wearing a white motorcycle helmet. In the bottom left corner of the main content area, there's a red-bordered button with the text 'Try Stardog' in white. At the very bottom of the page, there's a footer bar with a 'LEAVE A MESSAGE' button and a small square icon.

Stardog: The Enterprise Knowl... X +

https://www.stardog.com

PLATFORM CUSTOMERS RESOURCES ABOUT DOWNLOAD

The Knowledge Graph Platform for the Enterprise

With Stardog you can unify, query, search, and analyze all your data. Say goodbye to data silos forever.

Try Stardog

LEAVE A MESSAGE

# Stardog Graph Platform

- Stardog is easy to install and use, but rich in features
- It has a Web interface, good command-line tools and a Java API
- We'll look at how to
  - Load the peeps example files
  - Browse the results
  - Query the graph via the Web console

# Start Stardog

- This command will start Stardog listening to its default port (5820) and disable security

**stardog-admin server start --disable-security**

- Enter the URL <http://localhost:5820> to access the Web console

Use admin for both the user and password

Stardog Admin Web Console X +

localhost:5820

Stardog Admin Databases Security Query Management admin

**Server**

**Stardog Home:** /Users/finin/stardog      **Stardog Version:** 5.3.5

**Databases**

Name	Status
------	--------

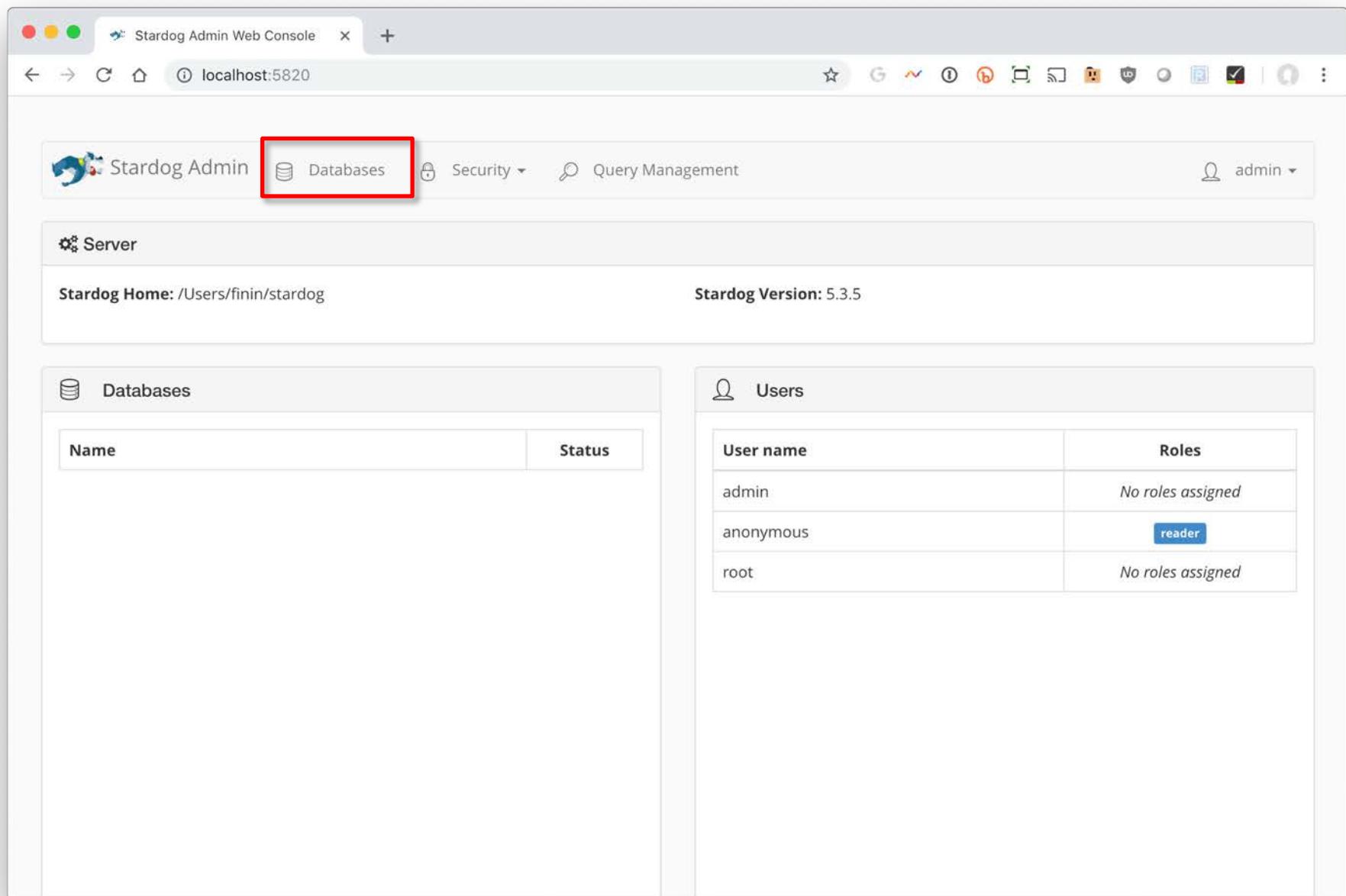
**Users**

User name	Roles
admin	No roles assigned
anonymous	reader
root	No roles assigned

# Stardog script

- load\_peeps.sh is a bash script for loading the peeps data and ontology
- Use variations for other systems or shells
- Once loaded go to <http://localhost:5820/> to use Stardog's web interface

# Stardog's web interface



The screenshot shows the Stardog Admin Web Console interface. At the top, there is a navigation bar with tabs for "Databases" (which is highlighted with a red box), "Security", and "Query Management". On the right side of the top bar, there is a user dropdown set to "admin". Below the top bar, there are two main sections: "Server" on the left and "Users" on the right. The "Server" section displays "Stardog Home: /Users/finin/stardog" and "Stardog Version: 5.3.5". The "Users" section lists three users: "admin" (with "No roles assigned"), "anonymous" (with "reader" role), and "root" (with "No roles assigned").

**Databases**

Name	Status
------	--------

**Users**

User name	Roles
admin	No roles assigned
anonymous	reader
root	No roles assigned

# Create a database

The screenshot shows the Stardog Admin Web Console interface. At the top, the title bar reads "Stardog Admin Web Console" and the address bar shows "localhost:5820/#/databases". The header includes links for "Stardog Admin", "Databases", "Security", and "Query Management", along with a user dropdown for "admin". Below the header, the main content area has a title "Databases" with a database icon. To the right of the title is a green "New DB" button, which is highlighted with a red rectangular box. Below the title is a search bar with a magnifying glass icon and the placeholder text "Type to search by database name". A table follows, with columns labeled "Status", "Name", "Last Activity", "Features", and "Actions".

Status	Name	Last Activity	Features	Actions
--------	------	---------------	----------	---------

# Name it mypeeps and accept the defaults

The screenshot shows the Stardog Admin Web Console interface. The title bar says "Stardog Admin Web Console" and the address bar shows "localhost:5820/#/databases/new". The main content area is titled "New database". It contains a wizard for creating a new database. The "Database name" field is set to "mypeeps" and is highlighted with a red box. The "Database namespaces" section lists several prefixes: rdf, rdfs, xsd, owl, and stardog, each with a red "X" button to remove them. There is also a blue "Add namespace" button. On the left, there are sections for "Database archetypes" (None selected), "Database online" (ON), "Strict Parsing" (ON), and "Preserve BNode identifiers" (ON). At the bottom, there are "Finish" and "Next" buttons, with the "Next" button being highlighted with a red box.

This wizard will help you create a new Stardog database. It will go through all the options available for setting up a new DB. All the options are filled up with the default values. If all you need are the default options, just go ahead and click Finish, otherwise click Next.

**Database name**  
mypeeps

**Database archetypes**  
None selected ▾

**Database online**  
ON

**Strict Parsing**  
ON

**Preserve BNode identifiers**  
ON

**Database namespaces**

- rdf=http://www.w3.org/1999/02/22-rdf-syntax-ns#
- rdfs=http://www.w3.org/2000/01/rdf-schema#
- xsd=http://www.w3.org/2001/XMLSchema#
- owl=http://www.w3.org/2002/07/owl#
- stardog=tag:stardog:api:

Add namespace

Finish

Next ➔

Stardog Admin Web Console X +

localhost:5820/#/databases/mypeeps

Stardog Admin Databases Security Query Management admin

Database created!

Database **mypeeps** was created, go to **mypeeps console** to add data

Query Browse Edit Optimize Drop ON

## mypeeps

Database

Database archetypes	
Database name	mypeeps
Database namespaces	rdf=http://www.w3.org/1999/02/22-rdf-syntax-ns# rdfs=http://www.w3.org/2000/01/rdf-schema# xsd=http://www.w3.org/2001/XMLSchema# owl=http://www.w3.org/2002/07/owl# stardog=tag:stardog:api: =http://api.stardog.com/
Database creation time	Tuesday, October 30th 2018, 10:48:07 pm -04:00
database modification time	Tuesday, October 30th 2018, 10:48:08 pm -04:00

# Click on *data* and select *+Add*

localhost:5820/mypeeps#!/webconsole

Admin Console    Query    Browse    Data    + Add    Remove    Export

Database Metadata

Database Name mypeeps

Database Online Yes

Last Modified

Index Type Disk

Index Literals Canonical Yes

Index Statistics Update Automatic Yes

Index Differential Enable Limit (Triples) 10000

Database Namespaces

- rdf=http://www.w3.org/1999/02/22-rdf-syntax-ns#
- rdfs=http://www.w3.org/2000/01/rdf-schema#
- xsd=http://www.w3.org/2001/XMLSchema#
- owl=http://www.w3.org/2002/07/owl#
- stardog=tag:stardog:api:
- =http://api.stardog.com/

Database Time Creation 2018-10-30T23:01:14.437-04:00

Index Size (Triples) 0

Index Persist Yes

Index Named Graphs Yes

Add the files

- peeps.ttl
- mypeeps.ttl

# Go to Browse to explore the graph

The screenshot shows a web browser window for the 'Schema Browser' at the URL `localhost:5820/mypeeps#/schema`. The browser's top navigation bar includes icons for back, forward, refresh, and search, along with tabs for 'Admin Console', 'Query', 'Browse' (which is highlighted with a red box), and 'Data'. A 'Search' bar and a help icon are also present.

A modal window titled 'Instructions' provides guidance: 'Click on the + / - icons to expand or collapse node elements in the Schema Tree.' The main content area is titled 'Schema Browser' and features a sidebar with 'Classes' selected (highlighted in blue) and 'Properties'.

The central part of the screen displays a hierarchical schema tree under the 'Classes' tab:

- Thing
  - Male person
    - Boy
  - Person
    - Adult person
      - Minor
    - Person
      - Woman

# Go to Query to enter a SPARQL query

localhost:5820/mypeeps#!/query/prefix%20rdf%3A%20<http%3A%2F%2Fwww.w...

Admin Console    Query    Browse    Data

## Query Panel

Hide SPARQL Editor

Explore File Print Help

Reasoning OFF Execute Clear

Prefixes:

x rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#> x owl: <http://www.w3.org/2002/07/owl#> x xsd: <http://www.w3.org/2001/XMLSchema#>  
x rdfs: <http://www.w3.org/2000/01/rdf-schema#> x foaf: <http://xmlns.com/foaf/0.1/>

```
1 select * where {?person foaf:givenName ?name}
```

The query

select \* where {?person foaf:givenName ?name}

Finds variable assignments that satisfy the where clause

# Go to Query to enter a SPARQL query

The screenshot shows a web browser window with the URL `localhost:5820/mypeeps#!/query`. The page displays SPARQL results for a query named "person". The results are listed in a table with two columns: "URI" and "name". There are four rows, each corresponding to a person: Alan, Robert, Carol, and Diana. Each row includes a link to the original RDF triple. A red box highlights the "Export as..." button and a dropdown menu below it, which contains options for RDF/XML, JSON, TSV, and CSV.

URI	name
<a href="https://raw.githubusercontent.com/UMBC-CMSC-491-691-F18-Knowledge-Graphs/peeps/master/mypeeps.ttl#alan">https://raw.githubusercontent.com/UMBC-CMSC-491-691-F18-Knowledge-Graphs/peeps/master/mypeeps.ttl#alan</a>	Alan
<a href="https://raw.githubusercontent.com/UMBC-CMSC-491-691-F18-Knowledge-Graphs/peeps/master/mypeeps.ttl#bob">https://raw.githubusercontent.com/UMBC-CMSC-491-691-F18-Knowledge-Graphs/peeps/master/mypeeps.ttl#bob</a>	Robert
<a href="https://raw.githubusercontent.com/UMBC-CMSC-491-691-F18-Knowledge-Graphs/peeps/master/mypeeps.ttl#carol">https://raw.githubusercontent.com/UMBC-CMSC-491-691-F18-Knowledge-Graphs/peeps/master/mypeeps.ttl#carol</a>	Carol
<a href="https://raw.githubusercontent.com/UMBC-CMSC-491-691-F18-Knowledge-Graphs/peeps/master/mypeeps.ttl#diana">https://raw.githubusercontent.com/UMBC-CMSC-491-691-F18-Knowledge-Graphs/peeps/master/mypeeps.ttl#diana</a>	Diana

It found four solutions. The data can be exported to your computer as a file in any of several formats (e.g., rdf, json, csv, tsv)

SPARQL Results RDF/XML

SPARQL Results JSON

TSV

CSV

Export as...

Page 1

The screenshot shows a web browser window with a red box highlighting the top navigation bar and a large red box highlighting the error message and configuration panel.

**Top Bar:** The browser's top bar includes the URL `localhost:5820/mypeeps#!/query`, a search field, and various bookmark and extension icons.

**Error Message:** A red box highlights the error message "Error! Unknown prefix: peeps" displayed in a pink box.

**Query Panel:** A large red box highlights the "Query Panel" section.

- Buttons:** "Hide SPARQL Editor", "Explore", "Reasoning" (set to OFF), "Execute", and "Clear".
- Prefixed:** A list of prefixes:
  - `rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>`
  - `owl: <http://www.w3.org/2002/07/owl#>`
  - `xsd: <http://www.w3.org/2001/XMLSchema#>`
  - `rdfs: <http://www.w3.org/2000/01/rdf-schema#>`
  - `foaf: <http://xmlns.com/foaf/0.1/>`
- SPARQL Editor:** A code editor containing the query:

```
1 select * where {?person a peeps:Man}
```

**Text Overlay:** A large red box contains the text: "The query systems needs to know (independently) about any namespace prefixes you want to use (other than the common ones). Enter these when you create the database."

# Command line commands

Running a simple bash script will create or refresh the peeps knowledge graph example

```
#!/bin/bash
# loads peeps.ttl, mypeeps.ttl and associated namespaces into a Stardog database.

PORT="5820"
SERVER="http://localhost:$PORT"
DBNAME="mypeeps"
DBURL="$SERVER/$DBNAME"

# stop server in case one is already running
stardog-admin --server $SERVER server stop

# start server
stardog-admin server start --port $PORT --disable-security
# drop database $DBNAME in case it exists already
stardog-admin --server $SERVER db drop -n $DBNAME

# create database $DBNAME with reasoning and search enabled
stardog-admin --server $SERVER db create -o reasoning.sameas=FULL -o search.enabled=true -n $DBNAME

# load ontology and data
stardog data add $DBURL peeps.ttl mypeeps.ttl

# add namespace prefixes for the query system to use
stardog namespace import --verbose $DBURL prefixes.ttl
```

# Query from Python

- Stardog serves as a endpoint for SPARQL queries
- Use this URL to send queries to the mypeeps database  
<http://localhost:5820/mypeeps/query/>
- There are packages that help do this in many languages, including Python
- See [query.py](#) in the peeps repository