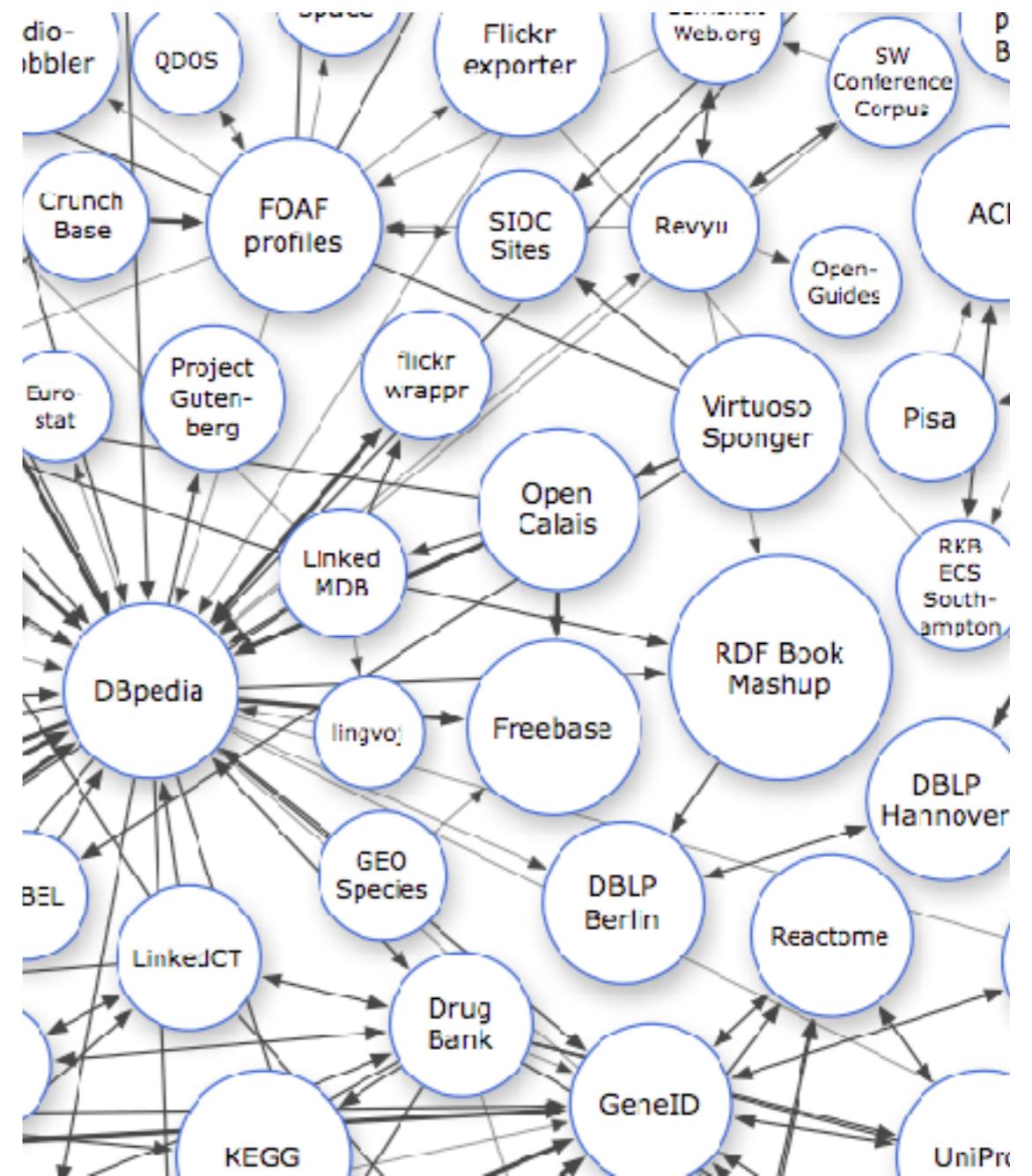


Linking & Querying Data

RDF + APIs

What is Linked Data?

- With the Web, we can link related documents and data
- Linked Data = set of best practices for publishing and connecting structured data on the Web.
- Key technologies that support Linked Data are:
 - URIs, HTTP, RDF
- Linked Open Vocabularies: <https://lov.okfn.org/dataset/lov>
- Pre & Post coordination of names/subjects continues to be a problem



RDF: The Basics

- Resource Description Framework (RDF) is a framework for representing information on the Web.
- RDF is based on **graphs** and **triples**
- One way to search RDF triple stores is through a **SPARQL** endpoint (we can talk more offline, if you're interested)



RDF Triples

Model:

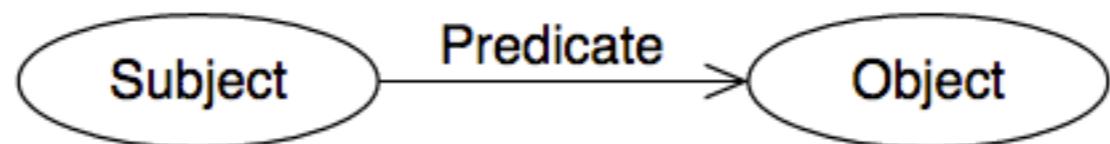
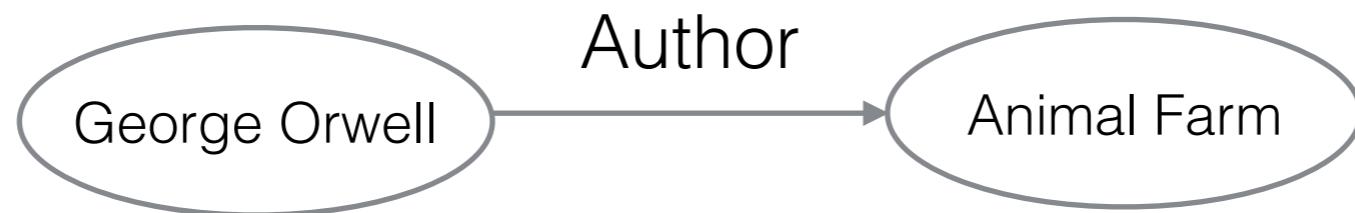


Fig. 1 An RDF graph with two nodes (Subject and Object) and a triple connecting them (Predicate)

Example:



Exercise 1:

Create Some RDF Graphs & Triples

RDF Triples

Model:

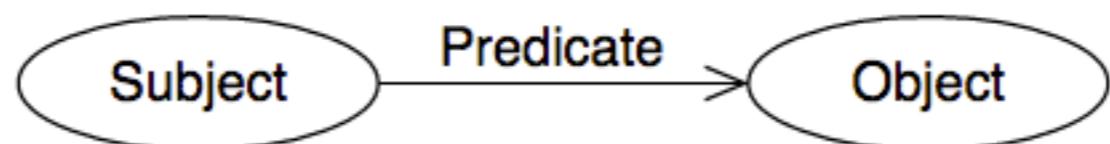


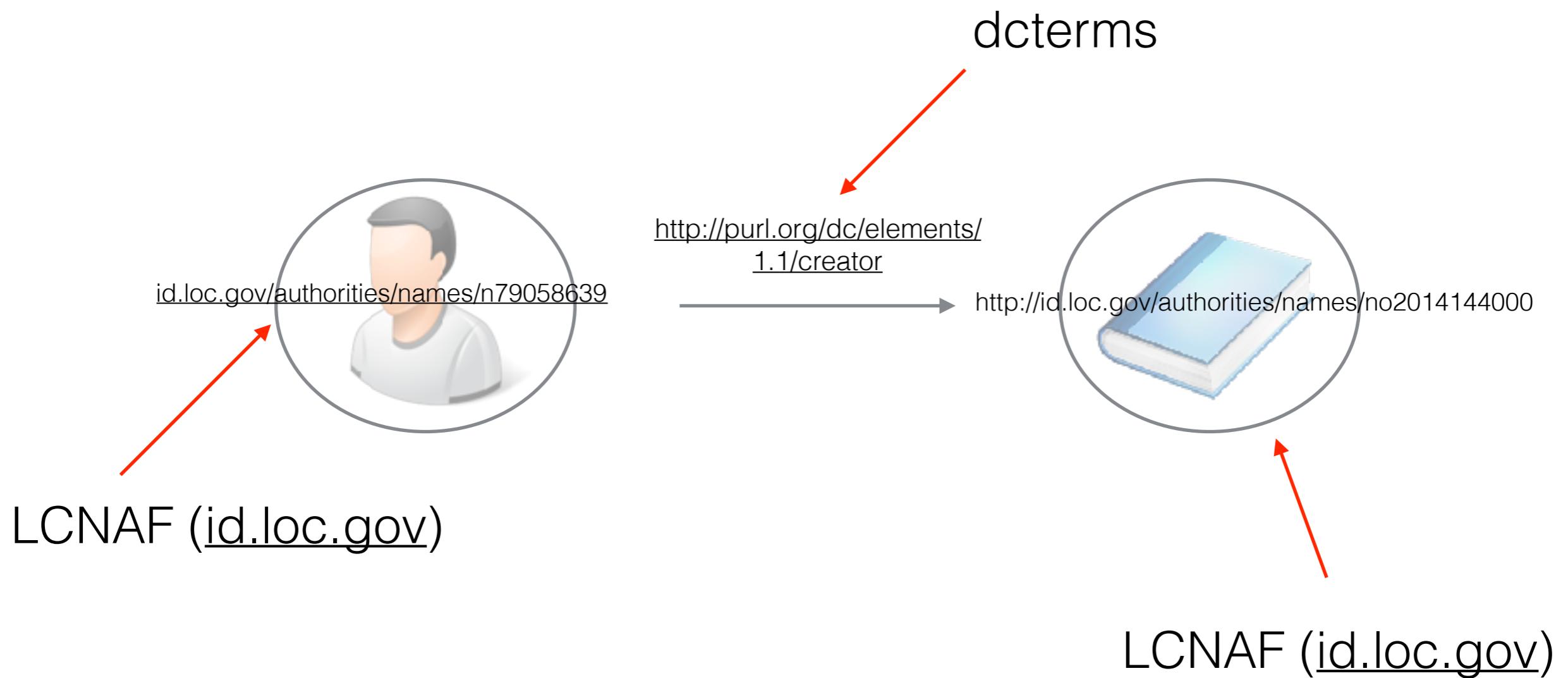
Fig. 1 An RDF graph with two nodes (Subject and Object) and a triple connecting them (Predicate)

Example:



RDF Triples (cont)

Example:



RDF: Predicates

- Predicates are defined in **namespaces**
- Other namespaces being used in Linked Data World:
 - SKOS (<http://www.w3.org/2004/02/skos/core#>)
 - FOAF (<http://xmlns.com/foaf/spec/>)
 - DC Terms (<http://purl.org/dc/terms/>)
 - and many more...

NOTE: you can use multiple namespaces in your dataset



RDF: Serializations

- Serialization = Ways RDF can be written (for machines!)
- Examples include:
 - RDF/XML (<https://www.w3.org/TR/rdf-syntax-grammar/>)
 - Notation-3 (N3) (<https://www.w3.org/TeamSubmission/n3/>)
 - Turtle (<https://www.w3.org/TR/turtle/>)
 - N-Triples (<https://www.w3.org/TR/n-triples/>)
 - RDFa (<https://www.w3.org/TR/rdfa-primer/>)
 - RDF/JSON (<https://www.w3.org/TR/rdf-json/>)



What can Linked Data & RDF actually do?

About 20,400,000 results (0.55 seconds)

George Orwell - Wikipedia, the free encyclopedia

https://en.wikipedia.org/wiki/George_O Orwell

Eric Arthur Blair (25 June 1903 – 21 January 1950), who used the pen name George Orwell, was an English novelist, essayist, journalist, and critic. His work is ...

[Animal Farm - George Orwell bibliography - Orwellian - Sonia Orwell](#)

George Orwell - Author, Journalist - Biography.com

www.biography.com/people/george-orwell-9429833

3 days ago - George Orwell was an English novelist, essayist, and critic most famous for his novels Animal Farm (1945) and Nineteen Eighty-four (1949). ... Born Eric Arthur Blair in Motihari, Bengal, India, in 1903, George Orwell, novelist, essayist and critic, went on to become best known for his ...

In the news



Rare letters of George Orwell, Virginia Woolf now online

[Hindustan Times](#) - 22 hours ago

Letters and manuscripts of Virginia Woolf, Ted Hughes, TS Eliot, George Orwell and other ...

Orwell's Fatal Attraction To Democratic Socialism

[Swarajya](#) - 5 hours ago

[Links and quotes for June 2, 2016: Elon Musk as Willy Wonka, George Orwell, and more - AEI](#)
American Enterprise Institute - 2 days ago

[More news for george orwell](#)

George Orwell | British author | Britannica.com

www.britannica.com/biography/George-Orwell

Sep 24, 2015 - George Orwell, pseudonym of Eric Arthur Blair (born June 25, 1903, Motihari, Bengal, India—died Jan. 21, 1950, London, England, English)



George Orwell

Novelist



Eric Arthur Blair, who used the pen name George Orwell, was an English novelist, essayist, journalist, and critic. [Wikipedia](#)

Born: June 25, 1903, Motihari, India

Died: January 21, 1950, London, United Kingdom

Full name: Eric Arthur Blair

Essays: [Shooting an Elephant](#), Politics and the English Language, more

Quotes

[View 7+ more](#)

In a time of universal deceit - telling the truth is a revolutionary act.

War is peace. Freedom is slavery. Ignorance is strength.

Big Brother is watching you.

Books

[View 30+ more](#)

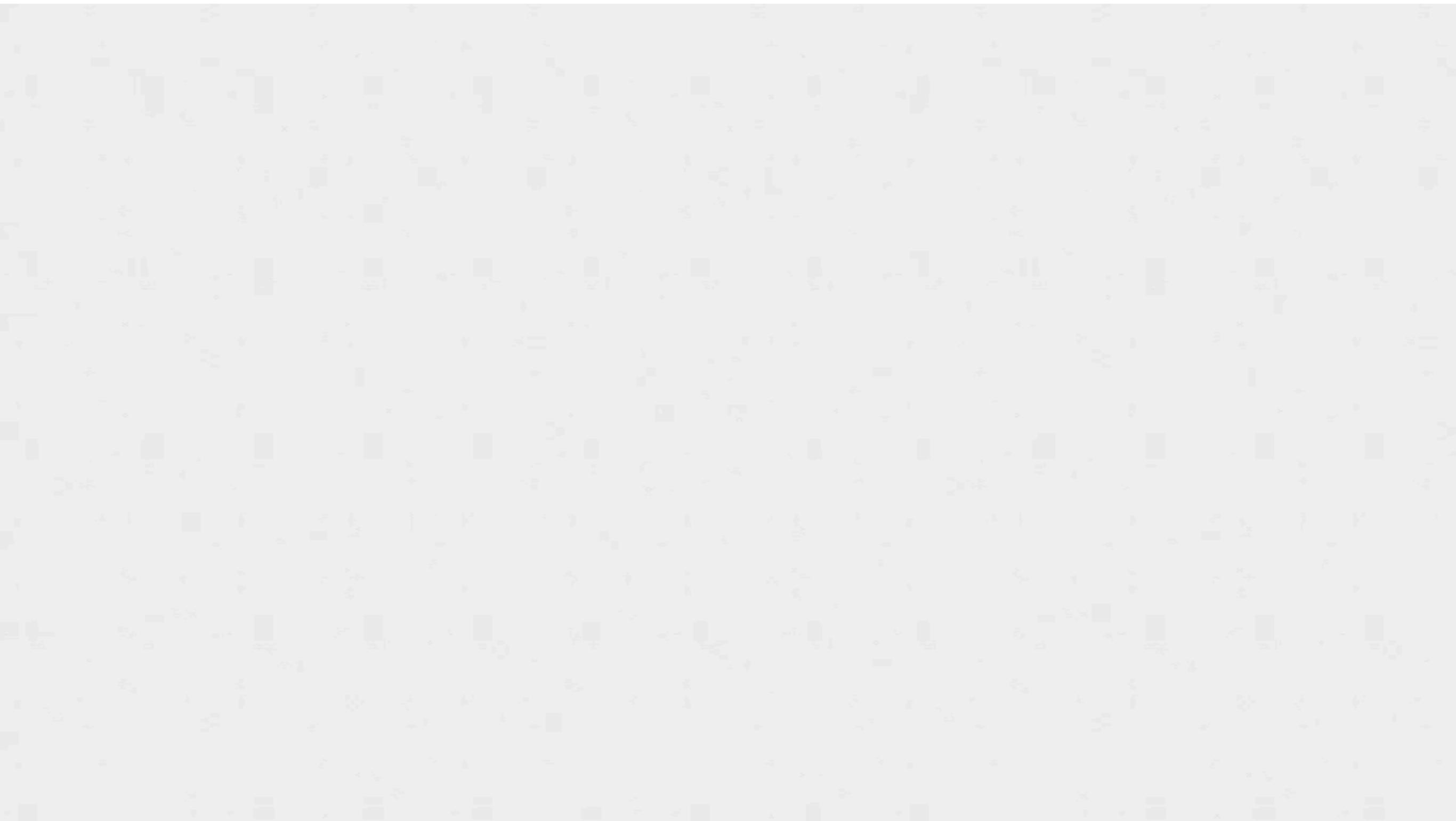


Towards a Public Data Infrastructure for a Large,
Multilingual, Semantic Knowledge Graph



<http://wiki.dbpedia.org/>

Note: Check out featured projects, including:
<http://wiki.dbpedia.org/projects/music-genre-map>



<https://linkedjazz.org/>



OCRE

Online Coins of the Roman Empire



Arco di Severo, e Caracalla. A Forami per
degli ornamenti di bronzo. B Tempio del
C Avanzo del Tempio di Giove Tonante
una rimasta in piedi della Greccostasi
Piranesi Architetto

<http://numismatics.org/ocre/>

Common Ways to Query RDF

SPARQL endpoint

- Pros: Dynamic
- Cons: Expensive to maintain, low availability, low bandwidth

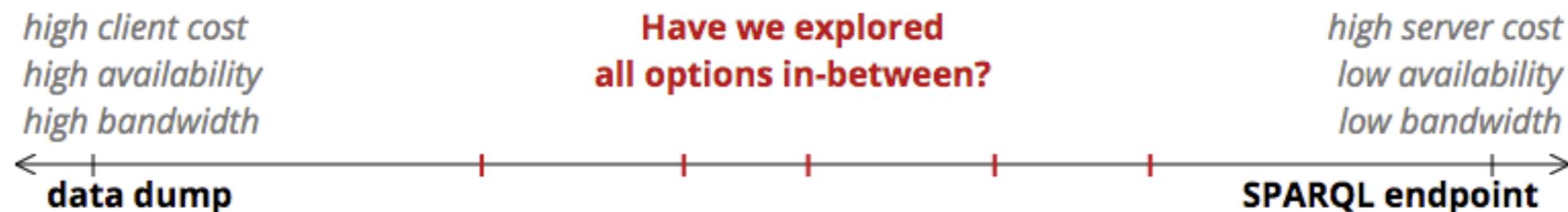
Static data dumps

- Pros: High availability, high bandwidth
- Cons: Not dynamic

Linked Data Fragments

A huge amount of Linked Data is available on the Web. But can live applications use it? SPARQL endpoints are expensive for the server, and not always available for all datasets. Downloadable dumps are expensive for clients, and do not allow live querying on the Web.

With **Linked Data Fragments**, and specifically the **Triple Pattern Fragments** interface, we aim to explore what happens when we redistribute the load between clients and servers. We then measure the impact of such interfaces on clients, servers, and caches.



Such solutions allow you to reliably execute queries against live Linked Data on the Web. You can even perform federated querying—all in your browser.

These are just a few examples of what the Semantic Web can become if we build intelligent clients instead of intelligent servers. Learn more.

<http://linkeddatafragments.org/>



What's an API?

API as Menu

- API = Automated Programming Interface
- APIs are the tools machines use to ask applications for information
- This interaction can be compared to a restaurant and a customer
- You can't order something that's not on the menu and you have to use the same language as the menu



Why use APIs in DH?

- allows you to query large datasets and (depending on the API), export matching data/metadata

THE NEW YORK PUBLIC LIBRARY DIGITAL
COLLECTIONS API[A note about our data model](#)[MODS help](#)[Parameters To Pass To The API](#)[Public Domain Filtering](#)[Image links in the API](#)[Rights statements in the Digital
Collections API](#)

QUERYING THE API

[Method 1: Return Captures for a UUID](#)[Method 2: Return UUID for local
identifier](#)[Method 3: Search in MODS field](#)[Method 4: Search across all MODS
field](#)[Method 5: Return MODS record for a
capture UUID](#)[Method 6: Return Captures Details and
Related Capture Information for a
Given Capture UUID](#)[Method 7: Return All Children
Captures for a Container or Collection
UUID](#)

The New York Public Library Digital Collections API

For more than a century, The New York Public Library has amassed an extraordinary trove of rare and unique material covering the full spectrum of recorded knowledge. Now, for the first time, significant portions of the Library's digitized collections are available as machine-readable data: over one million objects and records for you to search, crawl and compute. Sign up and get hacking today!

Questions/feedback? Write to: repoapi@nypl.org

Sign Up and Authentication

You will need to authenticate in order to use the Digital Collections API -- either via direct login with a username/password, or an authentication token that you will receive when you sign up for API access. (See below for help with authentication).

[Sign Up](#)[Sign In](#)

Access Methods

There are two ways to access the API: via a browser, or via token authentication.

1. Make a call to the API through the browser

- Either provide your username and password when prompted by the browser:

```
http://api.repo.nypl.org/api/v1/items/search?q=cats&publicDomainOnly=true
```

- Or pass your username and password to the browser directly:

```
http://username:password@api.repo.nypl.org/api/v1/items/search?q=cats&publicDomainOnly=true
```

<http://api.repo.nypl.org/>



Example: Ways to Access NYPL's API

- Through the browser
- Via script
- If you want to work with NYPL's API, first step is to sign up and get username and token (<http://api.repo.nypl.org/>)

Call API Through Browser

http://api.repo.nypl.org/api/v1/items/search?
q=cats&publicDomainOnly=true

The diagram illustrates the structure of the API URL. It features a central URL line: "http://api.repo.nypl.org/api/v1/items/search? q=cats&publicDomainOnly=true". Three arrows point from labels to specific parts of the URL: one arrow points from "Search term" to the query parameter "q=cats", another arrow points from "API" to the prefix "api.repo.nypl.org", and a third arrow points from "Rights" to the filter "publicDomainOnly=true".

Call API Through Browser (cont.)

Search term more than one word? Just put it in quotes

[http://api.repo.nypl.org/api/v1/items/search?
q=“United States”&publicDomainOnly=true](http://api.repo.nypl.org/api/v1/items/search?q=United%20States&publicDomainOnly=true)

Note: This works best for simple search queries
(i.e. not Boolean)

Call API Through Browser (cont.)

Your search results will be in JSON. If you want XML, add “.xml” after search

`http://api.repo.nypl.org/api/v1/items/
search.xml?q=dogs&publicDomainOnly=true`

Exercise 2:

Make Calls to NYPL API Through Browser

API Example Calls

1. Images of dogs
2. Images of Vancouver Island
3. Images of baseball (in XML)
4. Images of street cars (in XML)

Note: The metadata used to describe the images will impact your search

Exercise 3: Make Calls to NYPL API With a Script

Using a Script

- Benefits: You have more control over the structure of the data you get from the API.
- We're going to create and run a Python script that will pull metadata from the API based on our search terms, and convert that data to a tabular format

Using a Script (cont.)

Components of our script:

- Store API url with query
- Store access token
- execute call(s) to API
- Parse (xml) responses
- Write to csv and save

Using a Script (cont.)

Python 3

- Interpreted language
- human friendly syntax
- strongly typed
- active development

Using a Script (cont.)

NYPL API demo script

https://github.com/crowesn/nypl_api_demo

Using a Script (cont.)

Run the script:

- Make sure you have python installed (<https://www.python.org/downloads/>) and have signed up for a token with NYPL Labs
- Download “NYPL_api_demo.py” from GitHub: https://github.com/crowesn/nypl_api_demo Edit file in a text editor to add your token and save in the directory of your choice
- On Mac, open terminal and type “idle3”
- On Windows, go to start menu and execute Python IDLE.
- Type (minus the quotes): “python NYPL_api_demo.py”
- CSV output will be in the same directory as the script. File name will be: api_query_results.csv

Sharing your data

In a perfect world...

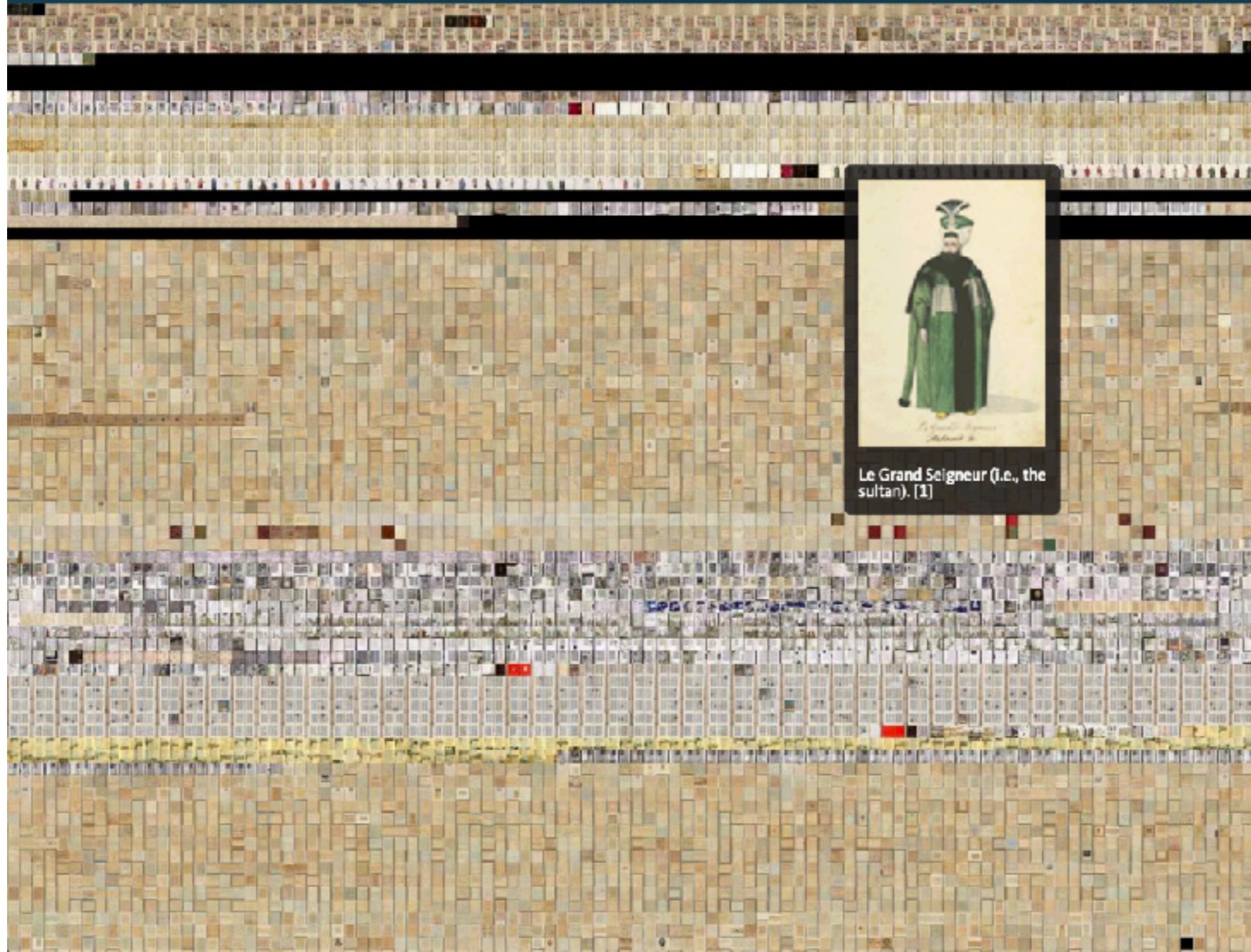
- we would all build RESTful APIs to share our data (https://en.wikipedia.org/wiki/Representational_state_transfer)

In the real world...

- we can make our data available in non-proprietary formats (ex. CSV) through library and discipline specific repositories
- have clear, easy to understand rights statements and “readme” files associated with our data



A public domain remix by Brian Foo of NYPL Labs, using the data released by NYPL. [View source code.](#)



11th century (400)

12th century (7)

13th century (805)

14th century (137)

15th century (6,942)

?

Group By:

Century Created

Genre

Collection

Color

Republic of Letters

1708 to 1779 (72 years)



Voltaire's correspondence, 1755-1776.

Only letters for which complete location information is available are shown.

Data courtesy the Electronic Enlightenment Project, University of Oxford.

Rights Statements

- One of the most important things you can do to make your data reusable; if you have sensitive data such as health records or confidential materials, redaction will be critical to sharing
- Easiest way is to create rights statements is to apply a Creative Commons license (<https://creativecommons.org/licenses/>); depending on your data, you may need a more robust data policy
- Not sure if you have the right to share your data? This is a quick cheatsheet: <http://copyright.cornell.edu/resources/publicdomain.cfm>. When in doubt, ask a copyright expert