
# Social Network Mining and Analysis

Bruno Gonçalves

www.data4sci.com/newsletter

github.com/DataForScience/DataMining

# Requirements

github.com/DataForScience/DataMining



3.7


# Lesson I - Web scraping

# urllib

docs.python.org/3/library/urllib.html

- Extensible library for opening and manipulating URLs
- https://foursquare.com/tyayayayaa/checkin/5304b652498e734439d8711f?s=ScMqmpSLg1buhGXQicDJS4A_FVY&ref=tw
- https <- protocol
- foursquare.com <- server
- /tyayayayaa/checkin/5304b652498e734439d8711f <- resource within server
- s=ScMqmpSLg1buhGXQicDJS4A_FVY&ref=tw <- Query string

# urllib

docs.python.org/3/library/urllib.html

- Extensible library for opening and manipulating URLs
- https://foursquare.com/tyayayayaa/checkin/5304b652498e734439d8711f?s=ScMqmpSLg1buhGXQicDJS4A_FVY&ref=tw
- https <- protocol
- foursquare.com <- server
- /tyayayayaa/checkin/5304b652498e734439d8711f <- resource within server
- s=ScMqmpSLg1buhGXQicDJS4A_FVY&ref=tw <- Query string

```
from urllib import parse

url = "https://foursquare.com/tyayayayaa/checkin/5304b652498e734439d8711f?s=ScMqmpSLg1buhGXQicDJS4A_FVY&ref=tw"

parsed = parse.urlparse(url)
query = parsed.query
query_dict = parse.parse_qs(query)

print(parsed)
print(query_dict)
```

# urllib

docs.python.org/3/library/urllib.html

- urllib2.urlopen(url) opens a url for reading and returns a "file handle"-like object
- Information about the webpage can be obtained with the .info() method in the form of an HTTPMessage
- The HTTPMessage object obeys the usual Python dictionary interface
- The .geturl() method returns the final location of the webpage.
  .urlopen() follows redirects until it connects with the final content.
- .getcode() returns the status code of the call
  - 200 OK
  - 404 File Not Found
  - 500 Internal Server Error

# posixpath

docs.python.org/3/library/os.path.html

- Manipulate paths in a POSIX operating system
- Also useful to extract information from remote resource paths
- Aliased to os.path if your operating systems is POSIX
- https://foursquare.com/tyayayayaa/checkin/5304b652498e734439d8711f
  -> Path in remote filesystem

- .basename(path) -> returns the file name (if there is one) 5304b652498e734439d8711f
- .dirname(path) -> return the directory portion /tyayayayaa/checkin

```
from urllib import parse
import posixpath

url = "https://foursquare.com/tyayayayaa/checkin/5304b652498e734439d8711f?s=ScMqmpSLg1buhGXQicDJS4A_FVY&ref=tw"

parsed = parse.urlparse(url)
filename = posixpath.basename(parsed.path)
directory = posixpath.dirname(parsed.path)

print(filename, directory)
```

# requests

requests.readthedocs.org/en/latest/

- "HTTP for Humans" - Simplified HTTP requests:
    - authentication (basic authentication, OAuth1, OAuth2, etc)
    - header manipulation
    - error handling
    - etc...

# requests

requests.readthedocs.org/en/latest/

- .get(url) open the given url for reading and returns a Response
- Response.status_code is a field that contains the calls status code
- Response.headers is a dict containing all the returned headers
- Response.text is a field that contains the content of the returned page
- Response.url contains the final url after all redirections
- Response.json() parses a JSON response (throws a JSONDecodeError exception if response is not valid JSON). Check “content-type” header field.

# json

docs.python.org/3/library/json.html

- JavaScript Object Notation - Serialization format originally developed for Javascript
- Currently widely accepted format for data dissemination
- Most languages have excellent libraries to handle it
- json.loads(obj_str) - load JSON data from a string - returns native Python object
- json.load(fp) - load JSON data from a file handle - returns native Python object
- json.dumps(obj) - convert JSON data to a string
- json.dump(obj, fp) - write the string version of obj to the file handle fp

# Challenge

docs.python.org/3/library/json.html

- Access the JSON file:

http://www.bgoncalves.com/test.json

- and extract all the friend pairs

# Basic Structure of a web page

www.w3schools.com/tags/

```
<!DOCTYPE html>
<html>
<head>
<meta charset="utf-8" />
<title>CSS Basics: A Cool Button</title>

<link href="style.css" rel="stylesheet" type="text/css" media="screen" />
</head>

<body>

    <div id="container">
        <a href="#" class="btn">Push the button</a>
    </div>

</body>
</html>
```

Doctype

Page title

Link to CSS stylesheet

Container div to centre things up

Anchor with class of "btn"

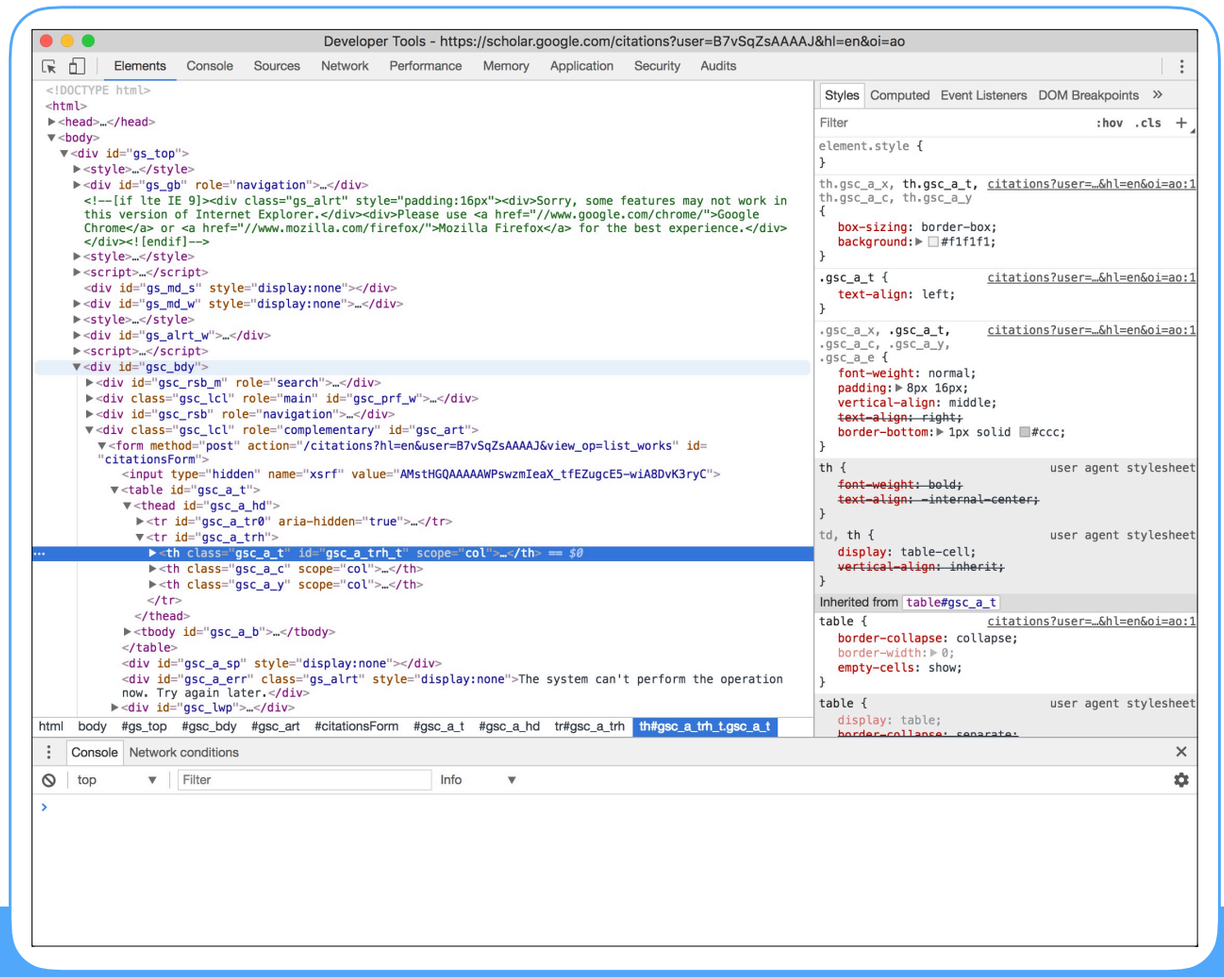
- Tree-like structure (DOM)
- Nested <tags> with attributes and content
- Two main sections under <html>:
  - <head> - meta data and resource location
  - <body> - page contents

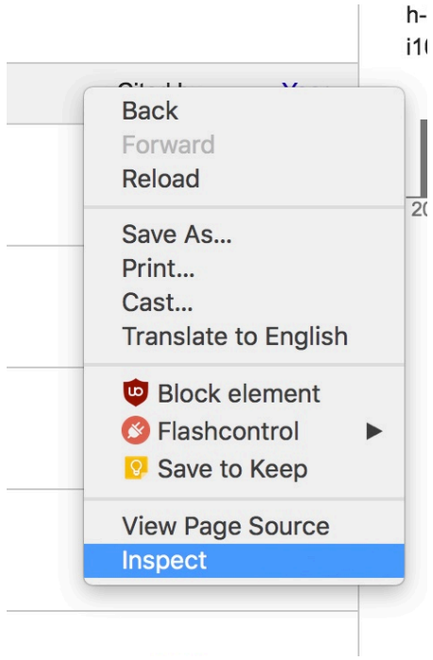
# Chrome Developer Tools

- Extremely powerful and intuitive set of tools
- Comes standard with Google Chrome. Just right click anywhere on the page and select "Inspect"
- Allows you to interactively change the DOM of any "live" webpage and find which element corresponds to which part of the page.

# Chrome Developer Tools

- Extremely powerful and intuitive set of tools
- Comes standard with Google Chrome. Just right click anywhere on the page and select "Inspect"
- Allows you to interactively change the DOM of any "live" webpage and find which element corresponds to which part of the page.



# Demo

# BeautifulSoup

crummy.com/software/BeautifulSoup/

- Parses html and xml files into a tree.
- **BeautifulSoup(page)** where page is a string or a "file handle"-like object
- BeautifulSoup parses the contents of the page and returns a BeautifulSoup object, corresponding to the root of the document tree.
- Each leaf of the tree is a Tag object:
  - can be used as dicts to access tag attributes,
  - contains pointers to children (**.findChildren()**), siblings (**.findSiblings()**) and parent (**.findParent()**)
  - can be accessed recursively by name (**head.title.content**)
  - modifying the contents of a tag modifies the contents of the document

# BeautifulSoup

crummy.com/software/BeautifulSoup/

- .findAll() returns a list of all tags matching a certain criteria
  - .findAll(name="a") find all "a" tags (links)
  - .findAll(name=["a", "div"]) find all "a" and "div" tags
  - .findAll(attrs = {"class": "btn"}) find all tags with class "btn", regardless of tag name
  - .findAll(name="a", attrs = {"class": "btn"}, limit=2) find the first two "a" tags with class "btn"
- Some servers use the User-agent string to decide how to format the output
  - Correctly handle specific versions of web browsers
  - Provide lighter/simplified versions to users on their mobiles
  - Refusing access to automated tools, etc

# Challenge

crummy.com/software/BeautifulSoup/

- Extract the title of Feynman's 100 most cited papers from Google Scholar

**Richard Feynman**

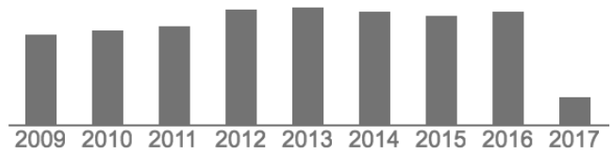
California Institute of Technology
quantum mechanics, quantum electrodynamics
No verified email

Follow

Google Scholar

| Title 1–100 | Cited by | Year |
|---|---|---|
| The Feynman lectures on physics<br>RP Feynman, RB Leighton, M Sands, SB Treiman<br>Physics Today 17, 45 | 14650 * | 1964 |

| Citation indices | All | Since 2012 |
|---|---|---|
| Citations | 82210 | 21241 |
| h-index | 59 | 45 |
| i10-index | 93 | 72 |



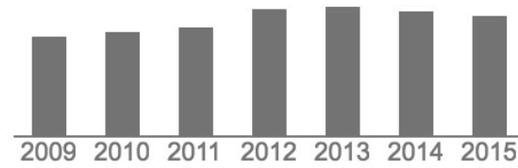
# Richard Feynman

California Institute of Technology
quantum mechanics, quantum electrodynamics
No verified email





| Citation indices | All | S |
|---|---|---|
| Citations | 82210 | |
| h-index | 59 | |
| i10-index | 93 | |




| Title 1–100 | Cited by | Year |
|---|---|---|
| The Feynman lectures on physics<br>RP Feynman, RB Leighton, M Sands, SB Treiman<br>Physics Today 17, 45 | 14650 * | 1964 |
| Quantum mechanics and path integration<br>RP Feynman, AR Hibbs<br>McGraw–Hill | 11256 * | 1965 |
| Simulating physics with computers<br>RP Feynman<br>International journal of theoretical physics 21 (6), 467-488 | 5715 | 1982 |
| Space-time approach to non-relativistic quantum mech<br>RP Feynman<br>Reviews of Modern Physics 20 (2), 367 | 4146 | 1948 |
| Forces in molecules<br>RP Feynman<br>Physical Review 56 (4), 340 | 3411 | 1939 |
| There's plenty of room at the bottom<br>RP Feynman<br>Engineering and Science 23 (5), 22-36 | 3342 | 1960 |
| Very high-energy collisions of hadrons<br>RP Feynman<br>Physical Review Letters 23 (24), 1415-1417 | 2801 | 1969 |
| Theory of the Fermi interaction | | |

Richard Feynman - Google Sch ×

Secure https://scholar.google.com/citations?hl=en&user=B7vSqZsAAAAJ&view_op=list_wo

Images More...

# Richard Feynman

California Institute of Technology

quantum mechanics, quantum electrodynamics

No verified email

td.gsc_a_t | 585 × 68

The Feynman lectures on physics
RP Feynman, RB Leighton, M Sands, SB Treiman
Physics Today 17, 45

Quantum mechanics and path integration
RP Feynman, AR Hibbs
McGraw–Hill

Simulating physics with computers
RP Feynman
International journal of theoretical physics 21 (6), 467-488

Space-time approach to non-relativistic quantum mechanics
RP Feynman
Reviews of Modern Physics 20 (2), 367

Forces in molecules
RP Feynman
Physical Review 56 (4), 340

There's plenty of room at the bottom
RP Feynman
Engineering and Science 23 (5), 22-36

Very high-energy collisions of hadrons
RP Feynman
Physical Review Letters 23 (24), 1415-1417

Theory of the Fermi interaction

Developer Tools - https://scholar.google.com/

Elements Console Sources Network Timeline Profiles

```
<!--[if lte IE 9]><div class="gs_alrt" style="padding:16px"><di
this version of Internet Explorer.</div><div>Please use <a href
Chrome</a> or <a href="//www.mozilla.com/firefox/">Mozilla Fire
</div><![endif]-->
▶<style>…</style>
▶<script>…</script>
 <div id="gs_md_s" style="display:none"></div>
▶<div id="gs_md_w" style="display:none">…</div>
▶<style>…</style>
▶<div id="gs_alrt_w">…</div>
▶<script>…</script>
▼<div id="gsc_bdy">
 ▶<div id="gsc_rsb_m" role="search">…</div>
 ▶<div class="gsc_lcl" role="main" id="gsc_prf_w">…</div>
 ▶<div id="gsc_rsb" role="navigation">…</div>
 ▼<div class="gsc_lcl" role="complementary" id="gsc_art">
  ▼<form method="post" action="/citations?hl=en&user=B7vSqZsAA
  "citationsForm">
     <input type="hidden" name="xsrf" value="AMstHGQAAAAAWPiWa3
   ▼<table id="gsc_a_t">
    ▼<thead id="gsc_a_hd">
      ▶<tr id="gsc_a_tr0" aria-hidden="true">…</tr>
      ▶<tr id="gsc_a_trh">…</tr>
      </thead>
    ▼<tbody id="gsc_a_b">
      ▼<tr class="gsc_a_tr">
... ▼<td class="gsc_a_t"> == $0
          <a href="/citations?
          view_op=view_citation&hl=en&user=B7vSqZsAAAAJ&pages
          x6o8ySG0sC" class="gsc_a_at">The Feynman lectures
          <div class="gs_gray">RP Feynman, RB Leighton, M San
        ▶<div class="gs_gray">…</div>
          </td>
        ▶<td class="gsc_a_c">…</td>
        ▶<td class="gsc_a_y">…</td>
        </tr>
      ▶<tr class="gsc_a_tr">…</tr>
      ▶<tr class="gsc_a_tr">…</tr>
      ▶<tr class="gsc_a_tr">…</tr>
      ▶<tr class="gsc_a_tr">…</tr>
      ▶<tr class="gsc_a_tr">…</tr>
```

html body #gs_top div#gsc_bdy div#gsc_art.gsc_lcl form#citationsForm table#gs

Console Network conditions

top ▼ Preserve log

>

# pyquery

pyquery.readthedocs.io/en/latest/

- Python version of the popular **jQuery** javascript package.
- More powerful than BeautifulSoup but also more complex.
- It defines three type of selectors:
  - element selector - retrieve all instances of a given HTML element (div, p, li, etc…)
  - #id selector - retrieve the element with id given by id
  - .class selector - retrieve all elements of a given class
- It also defines the usual jQuery pseudo-classes:
  - :first - first element
  - :last - last element
  - :even - even elements (0, 2, 4, …)
  - :odd - odd element (1, 3, 5, …)
  - :eq - a specific element (equals)
  - :lt - less than
  - :gt - greater than

# pyquery

pyquery.readthedocs.io/en/latest/

- pyQuery(url=url) or pyQuery(string) to parse a given external url of the html code in a specific string
- .attr("attr") returns a specific attribute of a given object.
- .addClass("bla") - add a css class
- .toggleClass("bla ble") - toggle class
- .removeClass("ble") - remove class
- .css("style": "value") - define css style value ("font-size", "15px")
- .items() - iterate over results

# Authentication Methodologies

# Authentication Methodologies

- Much of the content available online is only accessible to specific individuals for privacy, copyright protection, etc…

# Authentication Methodologies

- Much of the content available online is only accessible to specific individuals for privacy, copyright protection, etc…

- Three main ways of authenticating users:

# Authentication Methodologies

- Much of the content available online is only accessible to specific individuals for privacy, copyright protection, etc…

- Three main ways of authenticating users:

    - BasicAuth - The first and most basic one. Plain text user name and password sent to the server

# Authentication Methodologies

- Much of the content available online is only accessible to specific individuals for privacy, copyright protection, etc…

- Three main ways of authenticating users:

    - BasicAuth - The first and most basic one. Plain text user name and password sent to the server

    - OAuth 1 - Developed by a consortium of Industry leaders to provide transparent and secure authentication.

# Authentication Methodologies

- Much of the content available online is only accessible to specific individuals for privacy, copyright protection, etc…

- Three main ways of authenticating users:

    - BasicAuth - The first and most basic one. Plain text user name and password sent to the server

    - OAuth 1 - Developed by a consortium of Industry leaders to provide transparent and secure authentication.

    - OAuth 2 - An improvement on OAuth 1 designed to allow users to more easily share they're content on social media, etc…

# Authentication Methodologies

- Much of the content available online is only accessible to specific individuals for privacy, copyright protection, etc…

- Three main ways of authenticating users:

    - BasicAuth - The first and most basic one. Plain text user name and password sent to the server

    - OAuth 1 - Developed by a consortium of Industry leaders to provide transparent and secure authentication.

    - OAuth 2 - An improvement on OAuth 1 designed to allow users to more easily share they're content on social media, etc…

    - OpenID - A predecessor to OAuth that has gone out of favor.

# BasicAuth

requests.readthedocs.org/en/latest/

# BasicAuth

requests.readthedocs.org/en/latest/

- "The mother of all authentication protocols"

# BasicAuth

requests.readthedocs.org/en/latest/

- "The mother of all authentication protocols"
- Insecure but easy to use with standard implementations in all networking tools

# BasicAuth

requests.readthedocs.org/en/latest/

- "The mother of all authentication protocols"
- Insecure but easy to use with standard implementations in all networking tools
- In particular, in requests:

# BasicAuth

requests.readthedocs.org/en/latest/

- "The mother of all authentication protocols"
- Insecure but easy to use with standard implementations in all networking tools
- In particular, in requests:
    - requests.get(url, auth=("user", "pass")) open the given url and authenticate with username="user" and password="pass"

# OAuth 1

hueniverse.com/oauth/
tools.ietf.org/html/rfc5849

# OAuth 1

hueniverse.com/oauth/
tools.ietf.org/html/rfc5849

- "An open protocol to allow secure authorization in a simple and standard method from web, mobile and desktop applications."

# OAuth 1

hueniverse.com/oauth/
tools.ietf.org/html/rfc5849

- "An open protocol to allow secure authorization in a simple and standard method from web, mobile and desktop applications."


- The idea is to allow for a safe way to share privileges without divulging private credentials

# OAuth 1

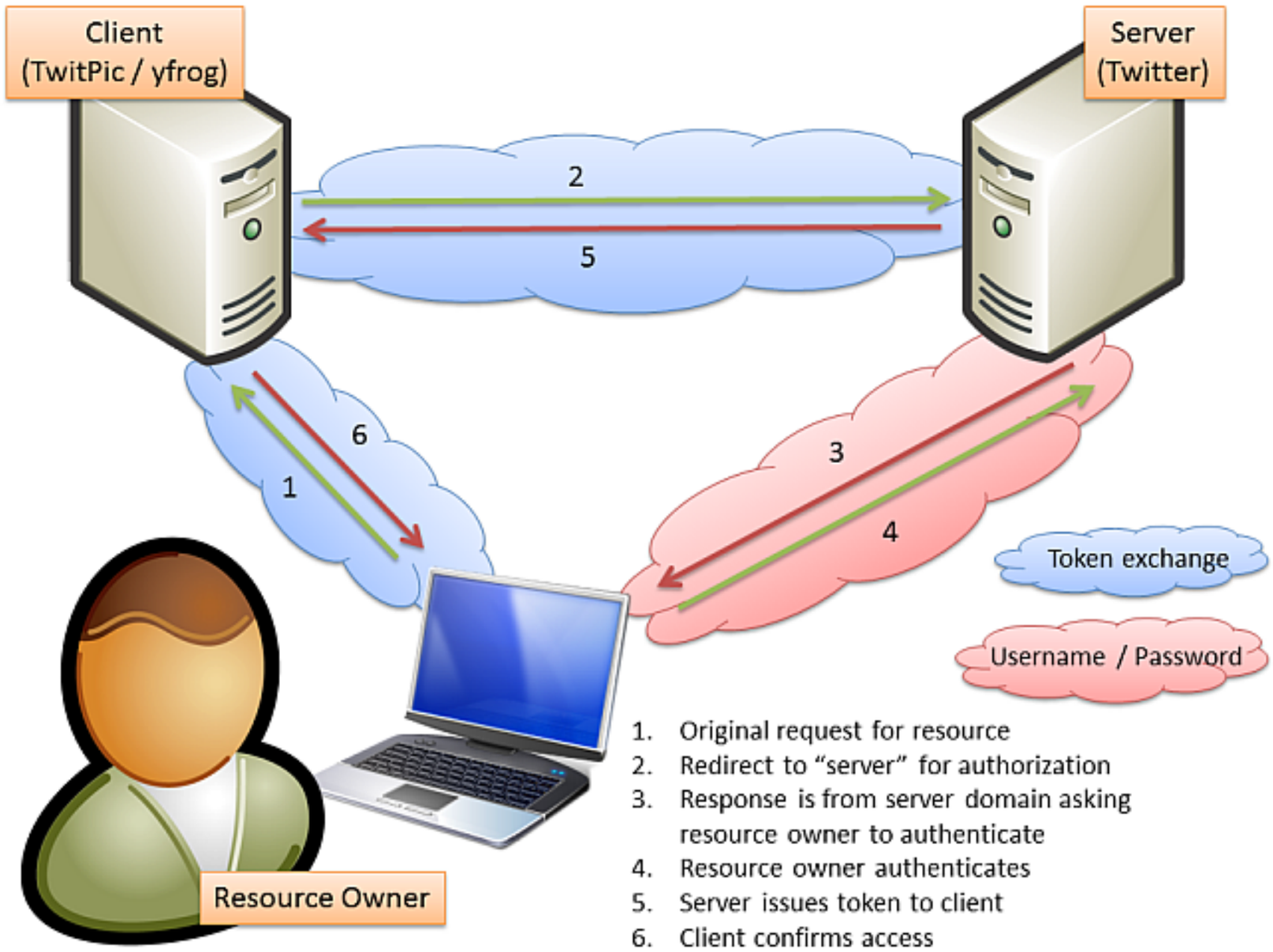
hueniverse.com/oauth/
tools.ietf.org/html/rfc5849

- "An open protocol to allow secure authorization in a simple and standard method from web, mobile and desktop applications."


- The idea is to allow for a safe way to share privileges without divulging private credentials
- Give XPTO Application permission to post to your Twitter account without having to trust the developers of XPTO with your username/password and while being able to unilaterally revoke privileges.

# OAuth 1

hueniverse.com/oauth/
tools.ietf.org/html/rfc5849



# OAuth 1

hueniverse.com/oauth/
tools.ietf.org/html/rfc5849

# OAuth 1

- After the "OAuth dance" is concluded, client application has two sets of keys:

hueniverse.com/oauth/
tools.ietf.org/html/rfc5849

# OAuth 1

hueniverse.com/oauth/
tools.ietf.org/html/rfc5849

- After the "OAuth dance" is concluded, client application has two sets of keys:
    - one that uses to identify itself as a valid application (api_key, api_secret)

# OAuth 1

hueniverse.com/oauth/
tools.ietf.org/html/rfc5849


- After the "OAuth dance" is concluded, client application has two sets of keys:
    - one that uses to identify itself as a valid application (api_key, api_secret)
    - one that uses to identify the user it wants to access (token, token_secret)

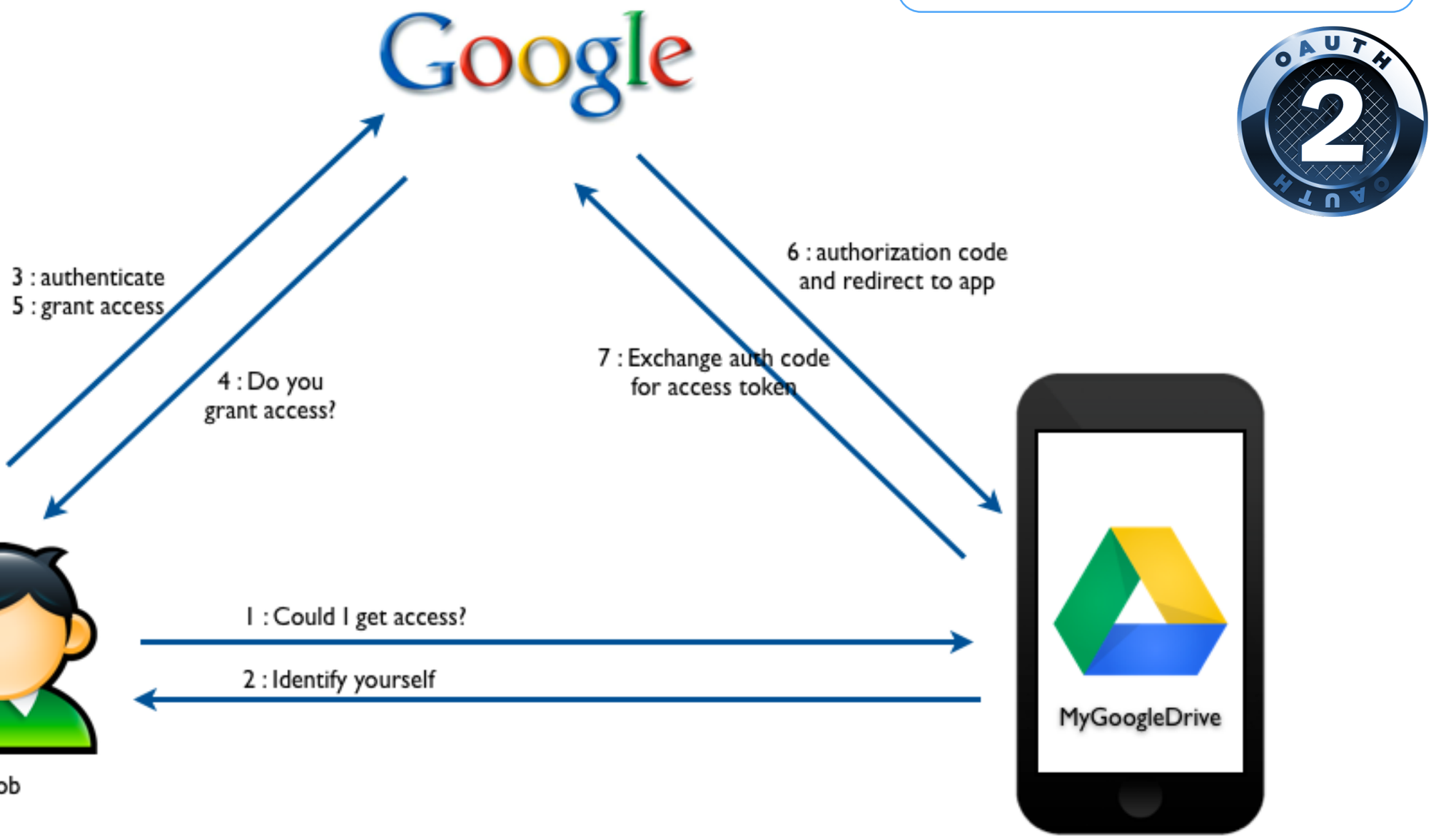
# OAuth 1

hueniverse.com/oauth/
tools.ietf.org/html/rfc5849

- After the "OAuth dance" is concluded, client application has two sets of keys:
  - one that uses to identify itself as a valid application (api_key, api_secret)
  - one that uses to identify the user it wants to access (token, token_secret)
- You can revoke access at any time by letting the token provider that a given app is no long authorized (invalidating token and token_secret).

# OAuth 2

tools.ietf.org/html/rfc6749
tools.ietf.org/html/rfc6750



# OAuth 2

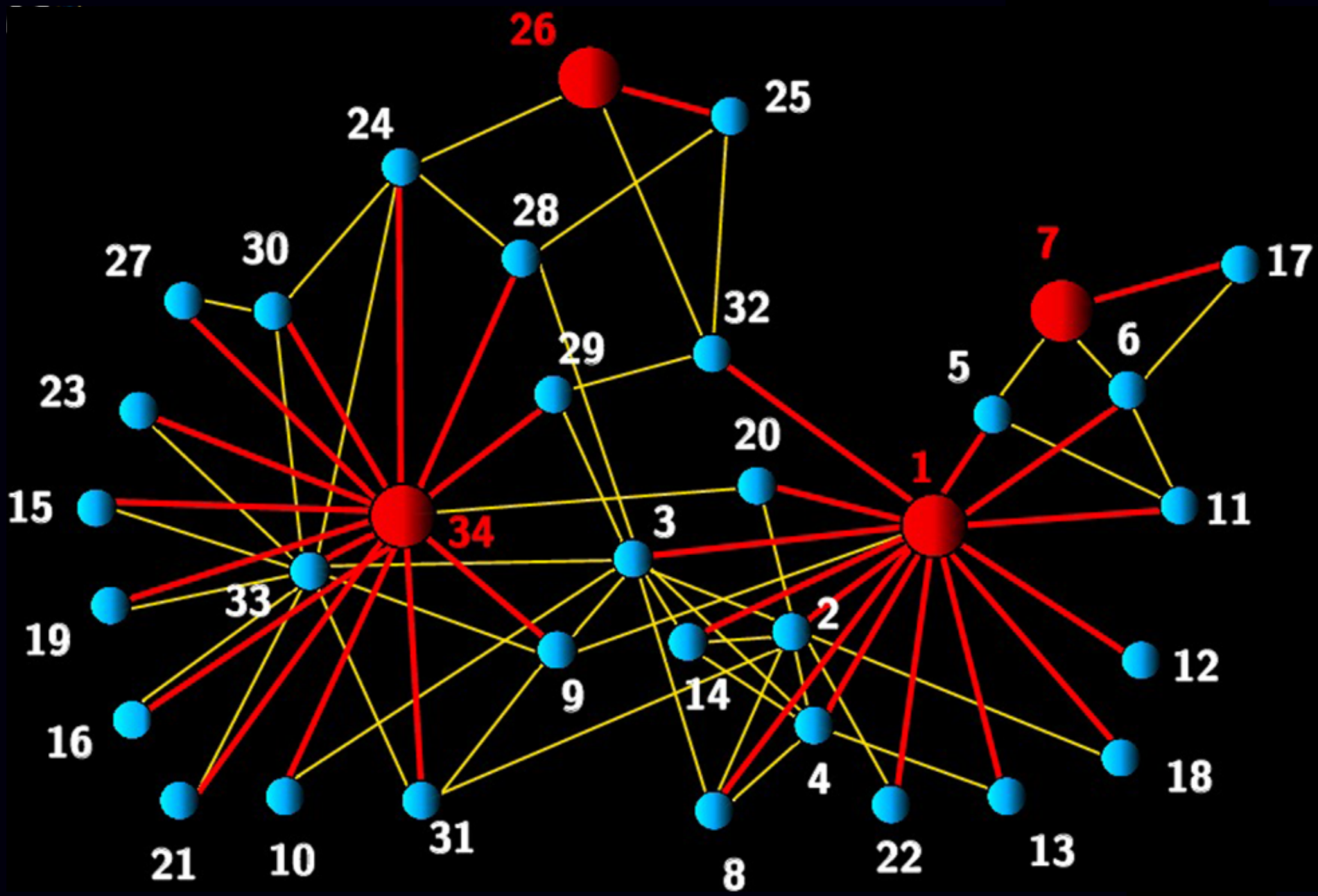
tools.ietf.org/html/rfc6749
tools.ietf.org/html/rfc6750


- Latest version of OAuth protocol
  - Similar “dance” required
  - Allows for “bearer tokens” - access is given to anyone able to provide a valid token without any further restrictions or authentication
    - access tokens are provided along with the request for the resource through a secure connection
    - tokens can expire automatically
- We will use both OAuth and OAuth2 over the next few days

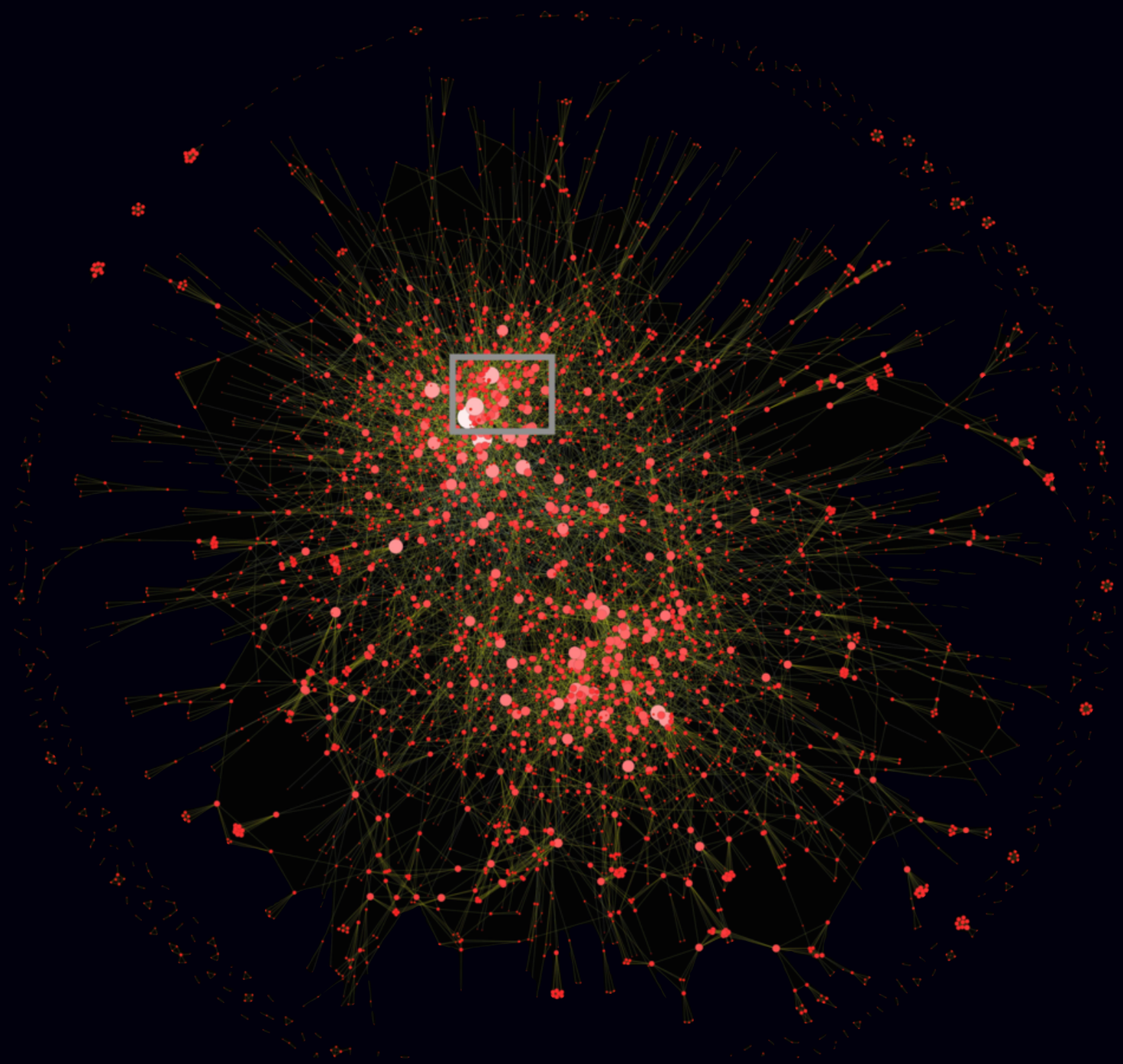

# Lesson II - Networks

# Facebook

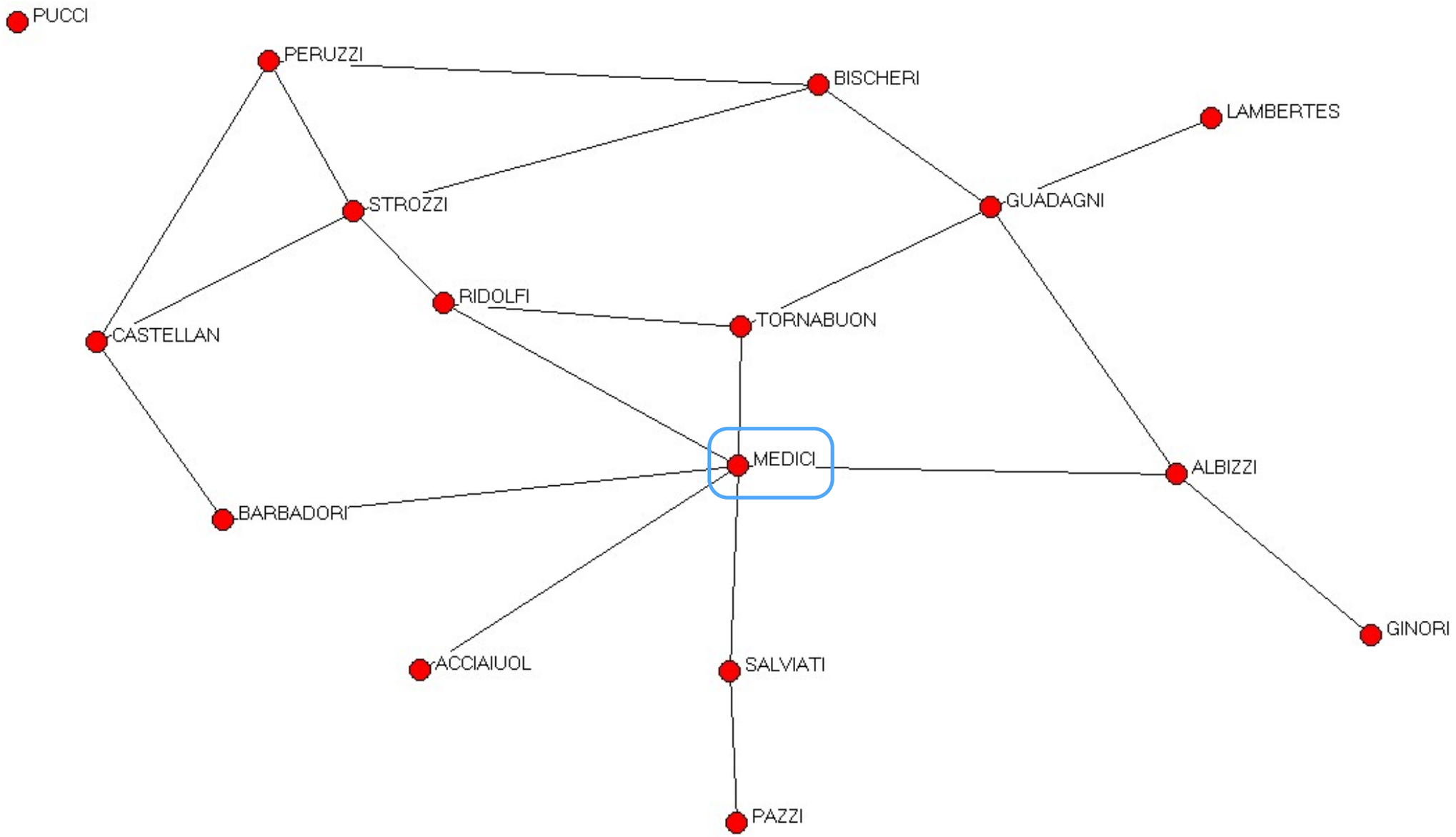
# Zachary Karate Club



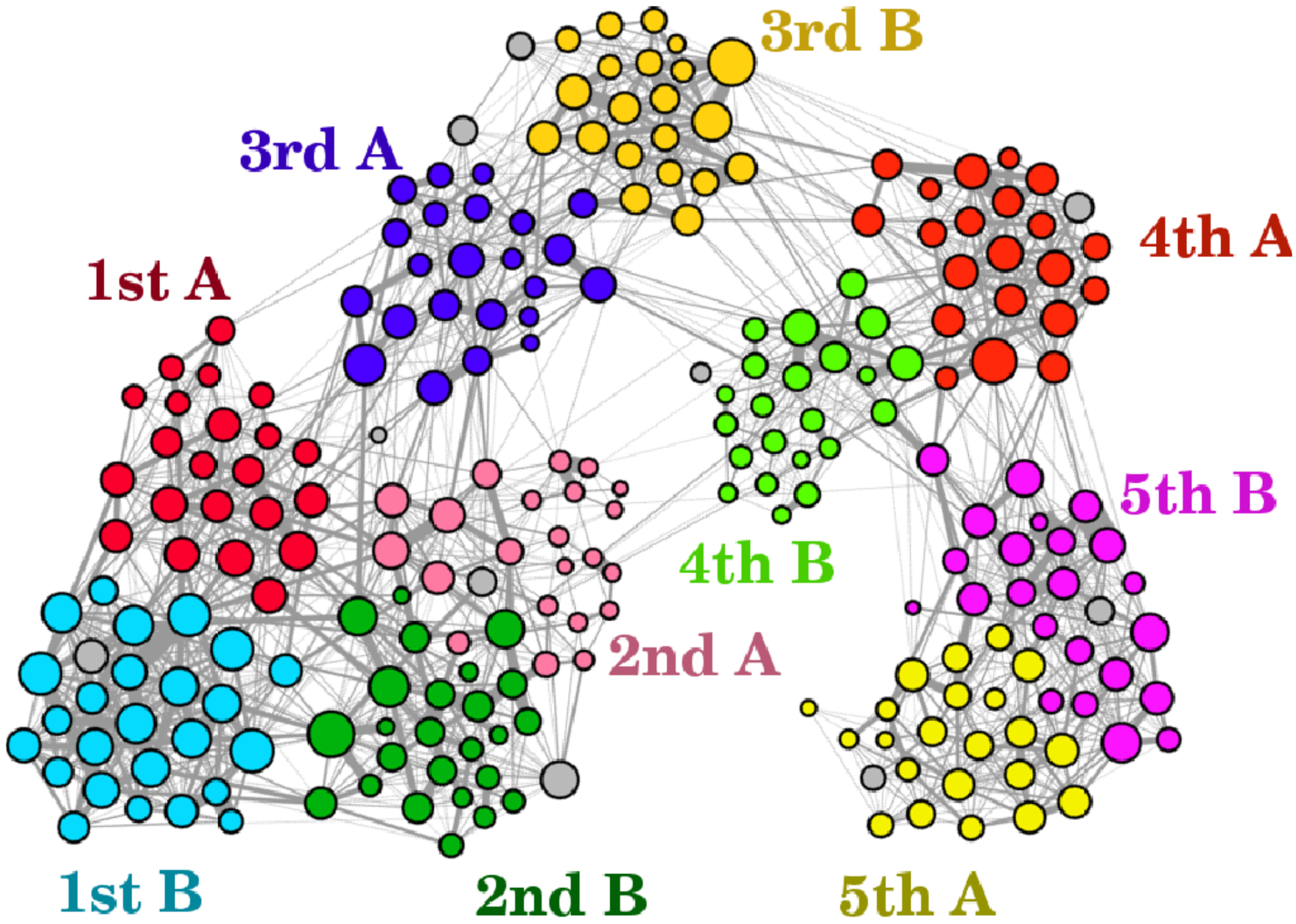
# Scientific Collaboration



# Florentine Weddings

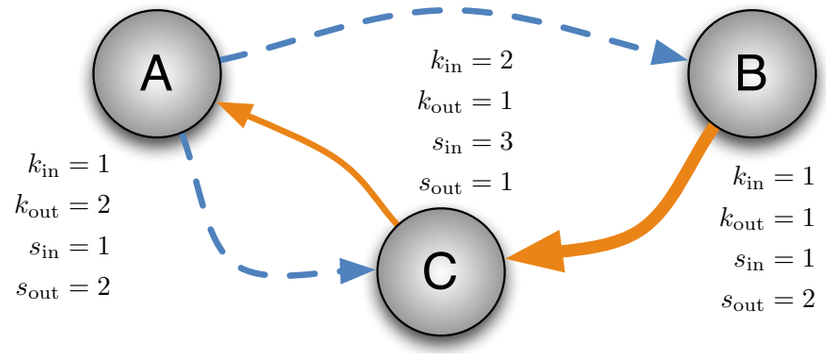


# Face-to-Face Contact in a Primary School



# Graph Theory

- Mathematical object, with set of nodes and edges




- **Node** - Individual Element

- **Edge** - Connection between element

- **Degree** - Number of edges connected to a node

- **Weighted Edge** - Edge with a weight associated

- **Direct Edge** - "One way street"

# NetworkX

networkx.github.io

# NetworkX

networkx.github.io

- High productivity software for complex networks

# NetworkX

networkx.github.io

- High productivity software for complex networks
- Simple Python interface

# NetworkX

networkx.github.io

- High productivity software for complex networks
- Simple Python interface
- Four types of graphs supported:

# NetworkX

networkx.github.io

- High productivity software for complex networks
- Simple Python interface
- Four types of graphs supported:
    - Graph - UnDirected

# NetworkX

networkx.github.io

- High productivity software for complex networks
- Simple Python interface
- Four types of graphs supported:
    - Graph - UnDirected
    - DiGraph - Directed

# NetworkX

networkx.github.io

- High productivity software for complex networks
- Simple Python interface
- Four types of graphs supported:
    - Graph - UnDirected
    - DiGraph - Directed
    - MultiGraph - Multi-edged Graph

# NetworkX

networkx.github.io

- High productivity software for complex networks
- Simple Python interface
- Four types of graphs supported:
    - Graph - UnDirected
    - DiGraph - Directed
    - MultiGraph - Multi-edged Graph
    - MultiDiGraph - Directed Multigraph

# NetworkX

networkx.github.io

- High productivity software for complex networks
- Simple Python interface
- Four types of graphs supported:
    - Graph - UnDirected
    - DiGraph - Directed
    - MultiGraph - Multi-edged Graph
    - MultiDiGraph - Directed Multigraph
- Similar interface for all types of graphs

# NetworkX

networkx.github.io

- High productivity software for complex networks
- Simple Python interface
- Four types of graphs supported:
    - Graph - UnDirected
    - DiGraph - Directed
    - MultiGraph - Multi-edged Graph
    - MultiDiGraph - Directed Multigraph
- Similar interface for all types of graphs
- Nodes can be any type of Python object - Practical way to manage relationships

# Growing Graphs

# Growing Graphs

- .add_node(node_id) Add a single node with ID node_id

# Growing Graphs

- .add_node(node_id) Add a single node with ID node_id

- .add_nodes_from() Add a list of node ids

# Growing Graphs

- .add_node(node_id) Add a single node with ID node_id
- .add_nodes_from() Add a list of node ids
- .add_edge(node_i, node_j) Adds an edge between node_i and node_j

# Growing Graphs

- .add_node(node_id) Add a single node with ID node_id
- .add_nodes_from() Add a list of node ids
- .add_edge(node_i, node_j) Adds an edge between node_i and node_j
- .add_edges_from() Adds a list of edges. Individual edges are represented by tuples

# Growing Graphs

- .add_node(node_id) Add a single node with ID node_id
- .add_nodes_from() Add a list of node ids
- .add_edge(node_i, node_j) Adds an edge between node_i and node_j
- .add_edges_from() Adds a list of edges. Individual edges are represented by tuples
- .remove_node(node_id)/.remove_nodes_from() Removing a node removes all associated edges

# Growing Graphs

- .add_node(node_id) Add a single node with ID node_id
- .add_nodes_from() Add a list of node ids
- .add_edge(node_i, node_j) Adds an edge between node_i and node_j
- .add_edges_from() Adds a list of edges. Individual edges are represented by tuples
- .remove_node(node_id)/.remove_nodes_from() Removing a node removes all associated edges
- .remove_edge(node_i, node_j)/.remove_edges_from()

# Graph Properties

# Graph Properties

- .nodes() Returns the list of nodes

# Graph Properties

- .nodes() Returns the list of nodes

- .edges() Returns the list of edges

# Graph Properties

- .nodes() Returns the list of nodes
- .edges() Returns the list of edges
- .degree() Returns a dict with each nodes degree .in_degree()/.out_degree() returns dicts with in/out degree for DiGraphs

# Graph Properties

- .nodes() Returns the list of nodes
- .edges() Returns the list of edges
- .degree() Returns a dict with each nodes degree .in_degree()/.out_degree() returns dicts with in/out degree for DiGraphs
- .is_connected() Returns true if the node is connected

# Graph Properties

- .nodes() Returns the list of nodes

- .edges() Returns the list of edges

- .degree() Returns a dict with each nodes degree .in_degree()/.out_degree() returns dicts with in/out degree for DiGraphs

- .is_connected() Returns true if the node is connected

- .is_weakly_connected()/.is_strongly_connected() for DiGraph

# Graph Properties

- .nodes() Returns the list of nodes

- .edges() Returns the list of edges

- .degree() Returns a dict with each nodes degree .in_degree()/.out_degree() returns dicts with in/out degree for DiGraphs

- .is_connected() Returns true if the node is connected

- .is_weakly_connected()/.is_strongly_connected() for DiGraph

- .connected_components() A list of nodes for each connected component

# Challenge

networkx.github.io

- Create an **Undirected Graph** containing these 6 edges

A - B

A - C

B - C

D - E

E - F

D- F

- And identify the **Connected Components**

# Snowball Sampling

- Commonly used in Social Science and Computer Science

    1. Start with a single node (or small number of nodes)

    2. Get "friends" list

    3. For each friend get the "friend" list

    4. Repeat for a fixed number of layers or until enough users have been connected

- Generates a connected component from each seed

- Quickly generates a *lot* of data/API calls

# Snowball Sampling

```python
def snowball(net, seed, max_depth = 3, maxnodes=1000):
    seen = set()
    queue = set()

    queue.add(seed)
    queue2 = set()

    for _ in range(max_depth+1):
        while queue:
            user_id = queue.pop()
            seen.add(user_id)

            NN = net.neighbors(user_id)

            for node in NN:
                if node not in seen:
                    queue2.add(node)

        queue.update(queue2)
        queue2 = set()

    return seen

net = NX.connected_watts_strogatz_graph(10000, 4, 0.01)
neve = snowball(net, 0)

print(neve)
```

snowball.py

# Challenge - Random Sampling

networkx.github.io

- Implement a random sampling approach on a graph. This is just a random walk along the edges of the graph, where at each step one of the edges is chosen at random.

- Notice that this is a small simplification of the SnowBall Sampling approach discussed before


# Lesson III - Wikipedia

# Wikipedia



WIKIPEDIA
The Free Encyclopedia

Main page
Contents
Featured content
Current events
Random article
Donate to Wikipedia
Wikipedia store

Interaction
Help
About Wikipedia
Community portal
Recent changes
Contact page

Tools
What links here
Related changes
Upload file
Special pages
Permanent link
Page information
Wikidata item
Cite this page

Print/export
Create a book
Download as PDF
Printable version

In other projects

## Turin

From Wikipedia, the free encyclopedia

Coordinates: 45°04′N 07°42′E

*For other uses, see Turin (disambiguation).*
*"Torino" redirects here. For other uses, see Torino (disambiguation).*

**Turin** (/tjʊˈrɪn/ *tewr-ɪn*; Italian: *Torino*, pronounced [toˈriːno] (listen); Piedmontese: *Turin*, pronounced [tyˈrin])[2] is a city and an important business and cultural centre in northern Italy, capital of the Piedmont region and was the first capital city of Italy. The city is located mainly on the western bank of the Po River, in front of Susa Valley and surrounded by the western Alpine arch and by the Superga Hill. The population of the city proper is 892,649 (August 2015) while the population of the urban area is estimated by Eurostat to be 1.7 million inhabitants. The Turin metropolitan area is estimated by the OECD to have a population of 2.2 million.[3]

In 1997 a part of the historical center of Torino was inscribed in the World Heritage List under the name Residences of the Royal House of Savoy.

The city has a rich culture and history, and is known for its numerous art galleries, restaurants, churches, palaces, opera houses, piazzas, parks, gardens, theatres, libraries, museums and other venues. Turin is well known for its Renaissance, Baroque, Rococo, Neo-classical, and Art Nouveau architecture.

Many of Turin's public squares, castles, gardens and elegant *palazzi* such as Palazzo Madama, were built between the 16th and 18th centuries. This was after the capital of the Duchy of Savoy (later Kingdom of Sardinia) was moved to Turin from Chambery (now in France) as part of the urban expansion.

The city used to be a major European political center. Turin was Italy's first capital city in 1861 and home to the House of Savoy, Italy's royal family.[4] From 1563, it was the capital of the Duchy of Savoy, then of the Kingdom of Sardinia ruled by the Royal House of Savoy and finally the first capital of the unified Italy.[5] Turin is sometimes called "the cradle of Italian liberty" for having been the birthplace and home of notable politicians and people who contributed to the *Risorgimento*, such as Cavour.[6]

The city currently hosts some of Italy's best universities, colleges, academies, lycea and gymnasia, such as the University of Turin, founded in the 15th century, and the Turin Polytechnic. In addition, the city is home to museums such as the Museo Egizio[7] and the Mole Antonelliana . Turin's attractions make it one of the world's top 250 tourist destinations and the tenth most visited city in Italy in 2008.[8]

Even though much of its political significance and importance had been lost by World War II, Turin became a major European crossroad for industry, commerce and trade, and is part of the famous "industrial triangle" along with Milan and Genoa. Turin is ranked third in Italy, after Milan and Rome, for economic strength.[9] With a GDP of $58 billion, Turin is the world's 78th richest city by purchasing power.[10] As of 2010, the city has

**Turin**
***Torino***

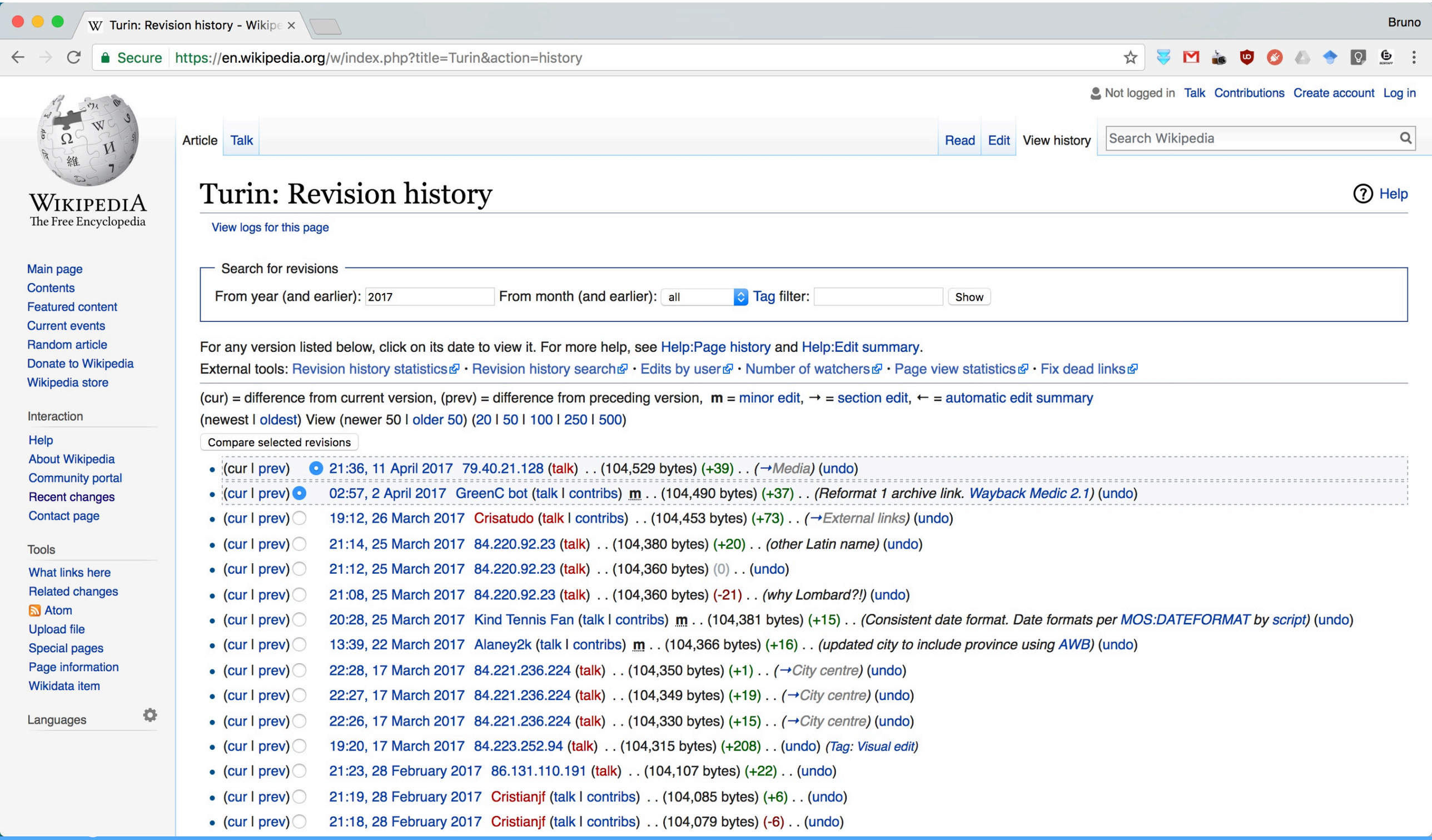
***Comune***

***Città di Torino***

**Flag**

**Coat of arms**

# Wikipedia


Turin: Revision history - Wikipe
Bruno
Secure https://en.wikipedia.org/w/index.php?title=Turin&action=history
Not logged in Talk Contributions Create account Log in
Article Talk Read Edit View history Search Wikipedia
WIKIPEDIA
The Free Encyclopedia
Turin: Revision history
Help
View logs for this page
Search for revisions
From year (and earlier): 2017 From month (and earlier): all Tag filter: Show
For any version listed below, click on its date to view it. For more help, see Help:Page history and Help:Edit summary.
External tools: Revision history statistics · Revision history search · Edits by user · Number of watchers · Page view statistics · Fix dead links
(cur) = difference from current version, (prev) = difference from preceding version, m = minor edit, → = section edit, ← = automatic edit summary
(newest | oldest) View (newer 50 | older 50) (20 | 50 | 100 | 250 | 500)
Compare selected revisions
(cur | prev) 21:36, 11 April 2017 79.40.21.128 (talk) . . (104,529 bytes) (+39) . . (→Media) (undo)
(cur | prev) 02:57, 2 April 2017 GreenC bot (talk | contribs) m . . (104,490 bytes) (+37) . . (Reformat 1 archive link. Wayback Medic 2.1) (undo)
(cur | prev) 19:12, 26 March 2017 Crisatudo (talk | contribs) . . (104,453 bytes) (+73) . . (→External links) (undo)
(cur | prev) 21:14, 25 March 2017 84.220.92.23 (talk) . . (104,380 bytes) (+20) . . (other Latin name) (undo)
(cur | prev) 21:12, 25 March 2017 84.220.92.23 (talk) . . (104,360 bytes) (0) . . (undo)
(cur | prev) 21:08, 25 March 2017 84.220.92.23 (talk) . . (104,360 bytes) (-21) . . (why Lombard?!) (undo)
(cur | prev) 20:28, 25 March 2017 Kind Tennis Fan (talk | contribs) m . . (104,381 bytes) (+15) . . (Consistent date format. Date formats per MOS:DATEFORMAT by script) (undo)
(cur | prev) 13:39, 22 March 2017 Alaney2k (talk | contribs) m . . (104,366 bytes) (+16) . . (updated city to include province using AWB) (undo)
(cur | prev) 22:28, 17 March 2017 84.221.236.224 (talk) . . (104,350 bytes) (+1) . . (→City centre) (undo)
(cur | prev) 22:27, 17 March 2017 84.221.236.224 (talk) . . (104,349 bytes) (+19) . . (→City centre) (undo)
(cur | prev) 22:26, 17 March 2017 84.221.236.224 (talk) . . (104,330 bytes) (+15) . . (→City centre) (undo)
(cur | prev) 19:20, 17 March 2017 84.223.252.94 (talk) . . (104,315 bytes) (+208) . . (undo) (Tag: Visual edit)
(cur | prev) 21:23, 28 February 2017 86.131.110.191 (talk) . . (104,107 bytes) (+22) . . (undo)
(cur | prev) 21:19, 28 February 2017 Cristianjf (talk | contribs) . . (104,085 bytes) (+6) . . (undo)
(cur | prev) 21:18, 28 February 2017 Cristianjf (talk | contribs) . . (104,079 bytes) (-6) . . (undo)
Main page
Contents
Featured content
Current events
Random article
Donate to Wikipedia
Wikipedia store
Interaction
Help
About Wikipedia
Community portal
Recent changes
Contact page
Tools
What links here
Related changes
Atom
Upload file
Special pages
Page information
Wikidata item
Languages

# Wikipedia



# User:Kind Tennis Fan

From Wikipedia, the free encyclopedia


This editor is a **Veteran Editor IV** and is entitled to display this **Gold Editor Star**.

## About me [ edit ]

I'm a male, born in the United Kingdom, and I've spent the vast majority of my life so far living in the south of England. My marital status is single and I currently have a girlfriend.

I registered as a Wikipedia user in July 2013, as I have many different interests and subjects that I like to read about.

## Passions [ edit ]

Tennis

Golf

Association football (more commonly known as football or soccer.)

Rock music (In particular: melodic Alternative rock, Soft rock, Art rock and New wave music)

Pugs (Highly recommended as nice gentle pets. They have a unique character and are one of the least aggressive breeds in the world.)

## Other interests [ edit ]

Politics

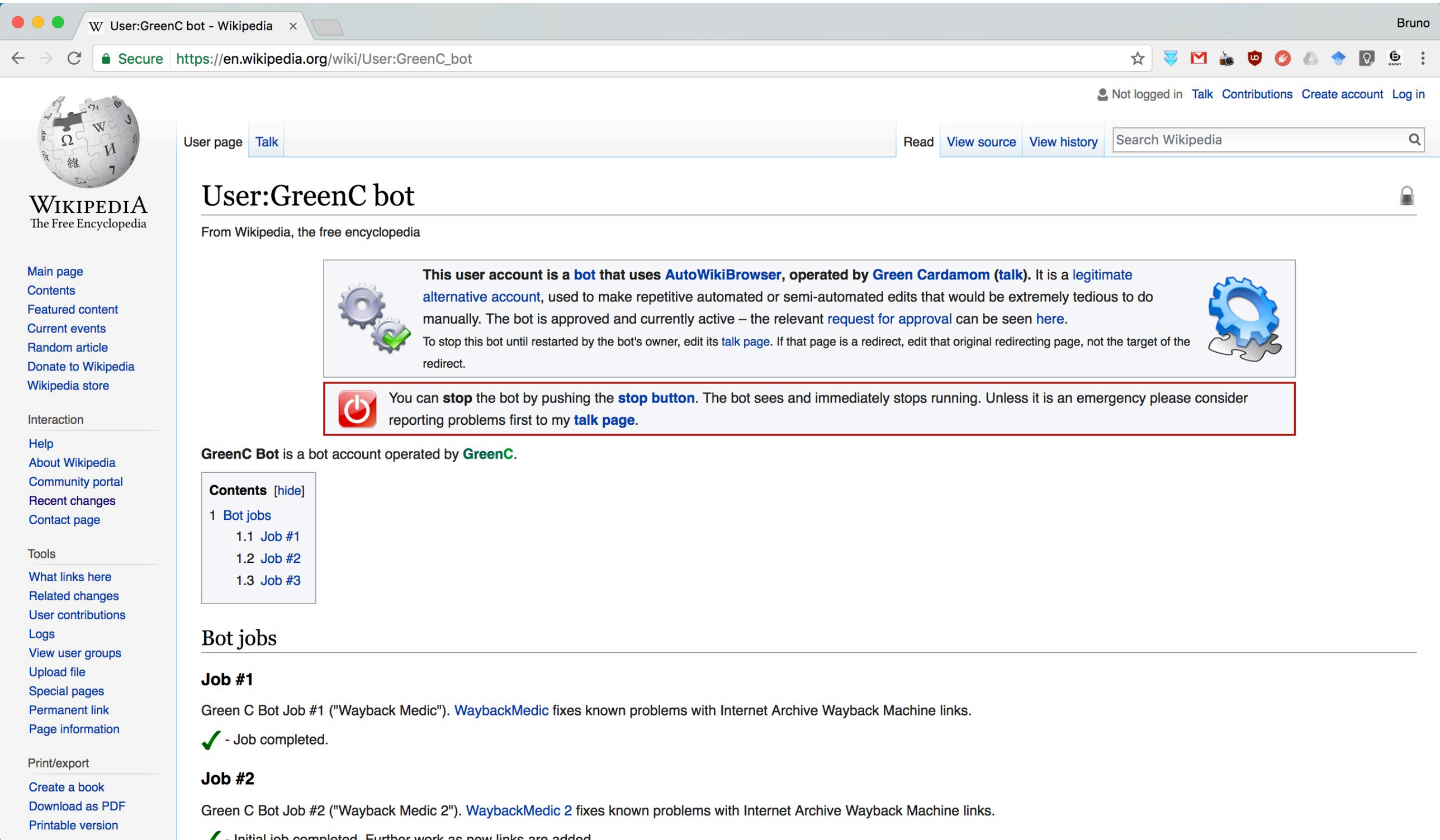
Contemporary history (Particularly the tragic conflict known as The Troubles where more than 3,500 people have been killed and over 50,000 people wounded. I would like to see the two main communities continue to work towards peace and reconciliation.)

Restaurants and Cuisine

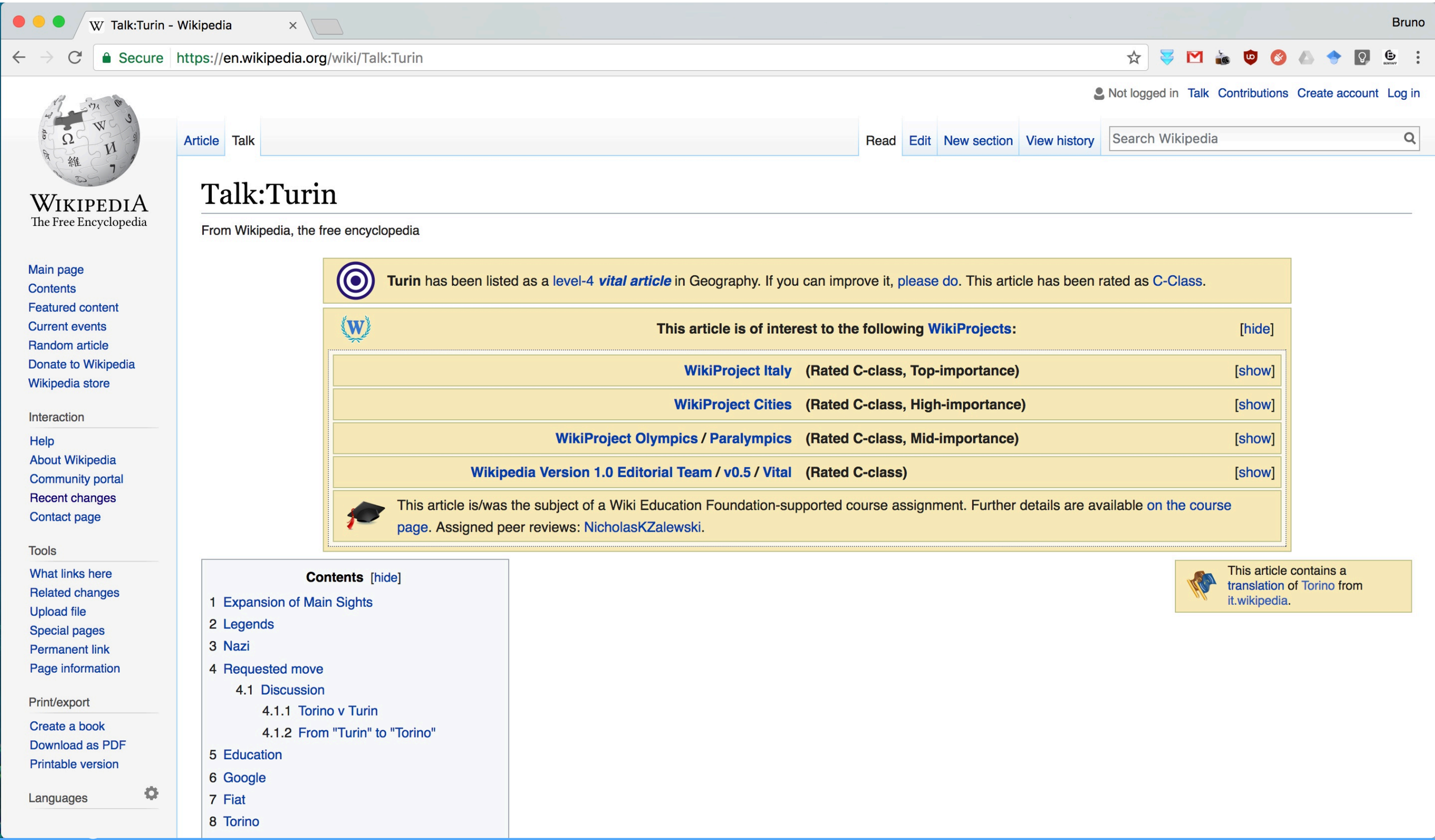
British New Wave films with an element of social realism. (Such as *Room at the Top* and *Saturday Night and Sunday Morning*.)

Psychology

# Wikipedia


User:GreenC bot - Wikipedia
Bruno
Secure https://en.wikipedia.org/wiki/User:GreenC_bot
Not logged in Talk Contributions Create account Log in
User page Talk
Read View source View history
Search Wikipedia
WIKIPEDIA
The Free Encyclopedia
Main page
Contents
Featured content
Current events
Random article
Donate to Wikipedia
Wikipedia store
Interaction
Help
About Wikipedia
Community portal
Recent changes
Contact page
Tools
What links here
Related changes
User contributions
Logs
View user groups
Upload file
Special pages
Permanent link
Page information
Print/export
Create a book
Download as PDF
Printable version
User:GreenC bot
From Wikipedia, the free encyclopedia
This user account is a bot that uses AutoWikiBrowser, operated by Green Cardamom (talk). It is a legitimate alternative account, used to make repetitive automated or semi-automated edits that would be extremely tedious to do manually. The bot is approved and currently active – the relevant request for approval can be seen here.
To stop this bot until restarted by the bot's owner, edit its talk page. If that page is a redirect, edit that original redirecting page, not the target of the redirect.
You can stop the bot by pushing the stop button. The bot sees and immediately stops running. Unless it is an emergency please consider reporting problems first to my talk page.
GreenC Bot is a bot account operated by GreenC.
Contents [hide]
1 Bot jobs
1.1 Job #1
1.2 Job #2
1.3 Job #3
Bot jobs
Job #1
Green C Bot Job #1 ("Wayback Medic"). WaybackMedic fixes known problems with Internet Archive Wayback Machine links.
- Job completed.
Job #2
Green C Bot Job #2 ("Wayback Medic 2"). WaybackMedic 2 fixes known problems with Internet Archive Wayback Machine links.
- Initial job completed. Further work as new links are added

# Wikipedia


Talk:Turin - Wikipedia
Bruno
Secure https://en.wikipedia.org/wiki/Talk:Turin
Not logged in Talk Contributions Create account Log in
Article Talk Read Edit New section View history Search Wikipedia
WIKIPEDIA
The Free Encyclopedia
Main page
Contents
Featured content
Current events
Random article
Donate to Wikipedia
Wikipedia store
Interaction
Help
About Wikipedia
Community portal
Recent changes
Contact page
Tools
What links here
Related changes
Upload file
Special pages
Permanent link
Page information
Print/export
Create a book
Download as PDF
Printable version
Languages
Talk:Turin
From Wikipedia, the free encyclopedia
Turin has been listed as a level-4 vital article in Geography. If you can improve it, please do. This article has been rated as C-Class.
This article is of interest to the following WikiProjects: [hide]
WikiProject Italy (Rated C-class, Top-importance) [show]
WikiProject Cities (Rated C-class, High-importance) [show]
WikiProject Olympics / Paralympics (Rated C-class, Mid-importance) [show]
Wikipedia Version 1.0 Editorial Team / v0.5 / Vital (Rated C-class) [show]
This article is/was the subject of a Wiki Education Foundation-supported course assignment. Further details are available on the course page. Assigned peer reviews: NicholasKZalewski.
Contents [hide]
1 Expansion of Main Sights
2 Legends
3 Nazi
4 Requested move
4.1 Discussion
4.1.1 Torino v Turin
4.1.2 From "Turin" to "Torino"
5 Education
6 Google
7 Fiat
8 Torino
This article contains a translation of Torino from it.wikipedia.

# Wikipedia Dumps

https://dumps.wikimedia.org

Wikimedia Downloads

Bruno

Secure https://dumps.wikimedia.org

# Wikimedia Downloads

If you are reading this on Wikimedia servers, please note that we have rate limited downloaders and we are capping the number of per-ip connections to 2. This will help to ensure that everyone can access the files with reasonable download times. Clients that try to evade these limits may be blocked. Our mirror sites do not have this cap.

## Data downloads

The Wikimedia Foundation is requesting help to ensure that as many copies as possible are available of all Wikimedia database dumps. Please **volunteer to host a mirror** if you have access to sufficient storage and bandwidth.

### Database backup dumps

A complete copy of all Wikimedia wikis, in the form of wikitext source and metadata embedded in XML. A number of raw database tables in SQL form are also available.
These snapshots are provided at the very least monthly and usually twice a month. If you are a regular user of these dumps, please consider subscribing to xmldatadumps-l for regular updates.

### Mirror Sites of the XML dumps provided above

Check the complete list.

### Static HTML dumps

A copy of all pages from all Wikipedia wikis, in HTML form.
These are currently not running.

### DVD distributions

# Wikipedia Dumps

https://dumps.wikimedia.org

# Wikipedia Dumps

https://dumps.wikimedia.org

- The Wikimedia foundation makes freely available regular dumps of all Wikimedia project databases.

# Wikipedia Dumps

https://dumps.wikimedia.org

- The Wikimedia foundation makes freely available regular dumps of all Wikimedia project databases.

- In particular, for the various language editions of Wikipedia, we have:

# Wikipedia Dumps

https://dumps.wikimedia.org

- The Wikimedia foundation makes freely available regular dumps of all Wikimedia project databases.

- In particular, for the various language editions of Wikipedia, we have:

    - *.pages-articles.xml.bz2 - Complete wiki page and revision content.

# Wikipedia Dumps

https://dumps.wikimedia.org

- The Wikimedia foundation makes freely available regular dumps of all Wikimedia project databases.
- In particular, for the various language editions of Wikipedia, we have:
  - *.pages-articles.xml.bz2 - Complete wiki page and revision content.
  - *.stub-meta-history.xml.gz - Wiki page revision metadata

# Wikipedia Dumps

https://dumps.wikimedia.org

- The Wikimedia foundation makes freely available regular dumps of all Wikimedia project databases.
- In particular, for the various language editions of Wikipedia, we have:
  - *.pages-articles.xml.bz2 - Complete wiki page and revision content.
  - *.stub-meta-history.xml.gz - Wiki page revision metadata
  - *.pagelinks.sql.gz - Wiki page-to-page link records

# Wikipedia Dumps

https://dumps.wikimedia.org

- The Wikimedia foundation makes freely available regular dumps of all Wikimedia project databases.
- In particular, for the various language editions of Wikipedia, we have:
  - *.pages-articles.xml.bz2 - Complete wiki page and revision content.
  - *.stub-meta-history.xml.gz - Wiki page revision metadata
  - *.pagelinks.sql.gz - Wiki page-to-page link records
  - *.geo_tags.sql.gz - List of pages' geographical coordinates

# Wikipedia Dumps

https://dumps.wikimedia.org

- The Wikimedia foundation makes freely available regular dumps of all Wikimedia project databases.
- In particular, for the various language editions of Wikipedia, we have:
  - *.pages-articles.xml.bz2 - Complete wiki page and revision content.
  - *.stub-meta-history.xml.gz - Wiki page revision metadata
  - *.pagelinks.sql.gz - Wiki page-to-page link records
  - *.geo_tags.sql.gz - List of pages' geographical coordinates
  - *.externallinks.sql.gz - Wiki external URL link records.

# Wikipedia Dumps

https://dumps.wikimedia.org

- The Wikimedia foundation makes freely available regular dumps of all Wikimedia project databases.
- In particular, for the various language editions of Wikipedia, we have:
  - *.pages-articles.xml.bz2 - Complete wiki page and revision content.
  - *.stub-meta-history.xml.gz - Wiki page revision metadata
  - *.pagelinks.sql.gz - Wiki page-to-page link records
  - *.geo_tags.sql.gz - List of pages' geographical coordinates
  - *.externallinks.sql.gz - Wiki external URL link records.
  - *.page.sql.gz - Base per-page data (id, title, old restrictions, etc).

# Wikipedia Dumps

https://dumps.wikimedia.org

- The Wikimedia foundation makes freely available regular dumps of all Wikimedia project databases.
- In particular, for the various language editions of Wikipedia, we have:
  - *.pages-articles.xml.bz2 - Complete wiki page and revision content.
  - *.stub-meta-history.xml.gz - Wiki page revision metadata
  - *.pagelinks.sql.gz - Wiki page-to-page link records
  - *.geo_tags.sql.gz - List of pages' geographical coordinates
  - *.externallinks.sql.gz - Wiki external URL link records.
  - *.page.sql.gz - Base per-page data (id, title, old restrictions, etc).
  - *.langlinks.sql.gz - Wiki interlanguage link records

# Wikipedia Dumps

https://dumps.wikimedia.org

- The Wikimedia foundation makes freely available regular dumps of all Wikimedia project databases.
- In particular, for the various language editions of Wikipedia, we have:
  - *.pages-articles.xml.bz2 - Complete wiki page and revision content.
  - *.stub-meta-history.xml.gz - Wiki page revision metadata
  - *.pagelinks.sql.gz - Wiki page-to-page link records
  - *.geo_tags.sql.gz - List of pages' geographical coordinates
  - *.externallinks.sql.gz - Wiki external URL link records.
  - *.page.sql.gz - Base per-page data (id, title, old restrictions, etc).
  - *.langlinks.sql.gz - Wiki interlanguage link records

# Wikipedia Dumps

https://dumps.wikimedia.org

- The Wikimedia foundation makes freely available regular dumps of all Wikimedia project databases.
- In particular, for the various language editions of Wikipedia, we have:
  - *.pages-articles.xml.bz2 - Complete wiki page and revision content.
  - *.stub-meta-history.xml.gz - Wiki page revision metadata
  - *.pagelinks.sql.gz - Wiki page-to-page link records
  - *.geo_tags.sql.gz - List of pages' geographical coordinates
  - *.externallinks.sql.gz - Wiki external URL link records.
  - *.page.sql.gz - Base per-page data (id, title, old restrictions, etc).
  - *.langlinks.sql.gz - Wiki interlanguage link records

# Wikipedia Dumps

https://dumps.wikimedia.org

- The Wikimedia foundation makes freely available regular dumps of all Wikimedia project databases.
- In particular, for the various language editions of Wikipedia, we have:
  - *.pages-articles.xml.bz2 - Complete wiki page and revision content.
  - *.stub-meta-history.xml.gz - Wiki page revision metadata
  - *.pagelinks.sql.gz - Wiki page-to-page link records
  - *.geo_tags.sql.gz - List of pages' geographical coordinates
  - *.externallinks.sql.gz - Wiki external URL link records.
  - *.page.sql.gz - Base per-page data (id, title, old restrictions, etc).
  - *.langlinks.sql.gz - Wiki interlanguage link records

# (Wikipedia Dump "Dumping")

# (Wikipedia Dump "Dumping")

- I've written a simple script to easily download the most recent version of specific files for many different languages.

# (Wikipedia Dump "Dumping")

- I've written a simple script to easily download the most recent version of specific files for many different languages.

- You can find it in the GitHub repo: **wikidump.py**

# (Wikipedia Dump "Dumping")

- I've written a simple script to easily download the most recent version of specific files for many different languages.

- You can find it in the GitHub repo: **wikidump.py**

- To customize to your needs, you just need to list the files you want in the allowed_files list and the languages you're interested in allowed_wikis

# (Wikipedia Dump "Dumping")

- I've written a simple script to easily download the most recent version of specific files for many different languages.

- You can find it in the GitHub repo: **wikidump.py**

- To customize to your needs, you just need to list the files you want in the allowed_files list and the languages you're interested in allowed_wikis

- If you want to download all the files from **acewiki** required for this tutorial, you would simply set:

```
allowed_files = ["stub-meta-history.xml.gz",
                 "geo_tags.sql.gz",
                 "langlinks.sql.gz",
                 ]

allowed_wikis = ["acewiki"]
```

# (Wikipedia Dump "Dumping")

- I've written a simple script to easily download the most recent version of specific files for many different languages.

- You can find it in the GitHub repo: **wikidump.py**

- To customize to your needs, you just need to list the files you want in the allowed_files list and the languages you're interested in allowed_wikis

- If you want to download all the files from **acewiki** required for this tutorial, you would simply set:

```
allowed_files = ["stub-meta-history.xml.gz",
                 "geo_tags.sql.gz",
                 "langlinks.sql.gz",
                 ]

allowed_wikis = ["acewiki"]
```

- But with what you learn so far you should be able to easily write your own version 

# SQL files

# SQL files

- Standard format, well suited for loading the data directly to a relational database (MySQL, MariaDB, PostgreSQL, etc…)

# SQL files

- Standard format, well suited for loading the data directly to a relational database (MySQL, MariaDB, PostgreSQL, etc…)

- Databases are optimized for fast querying of information, but not suitable for large scale processing where you touch all or most rows.

# SQL files

- Standard format, well suited for loading the data directly to a relational database (MySQL, MariaDB, PostgreSQL, etc…)

- Databases are optimized for fast querying of information, but not suitable for large scale processing where you touch all or most rows.

- mysqldump_to_csv.py - convert a wikipedia dump to a CSV file.

# SQL files

- Standard format, well suited for loading the data directly to a relational database (MySQL, MariaDB, PostgreSQL, etc…)

- Databases are optimized for fast querying of information, but not suitable for large scale processing where you touch all or most rows.

- mysqldump_to_csv.py - convert a wikipedia dump to a CSV file.

  - Slightly modified version of https://github.com/jamesmishra/mysqldump-to-csv

# SQL files

- Standard format, well suited for loading the data directly to a relational database (MySQL, MariaDB, PostgreSQL, etc…)

- Databases are optimized for fast querying of information, but not suitable for large scale processing where you touch all or most rows.

- mysqldump_to_csv.py - convert a wikipedia dump to a CSV file.

  - Slightly modified version of https://github.com/jamesmishra/mysqldump-to-csv

  - Available in the courses GItHub repository

# SQL files

- Standard format, well suited for loading the data directly to a relational database (MySQL, MariaDB, PostgreSQL, etc…)

- Databases are optimized for fast querying of information, but not suitable for large scale processing where you touch all or most rows.

- mysqldump_to_csv.py - convert a wikipedia dump to a CSV file.

  - Slightly modified version of https://github.com/jamesmishra/mysqldump-to-csv

  - Available in the courses GItHub repository

  - First row is column names as defined in the SQL file

# langlinks

# langlinks

- Just a few fields:

# langlinks

- Just a few fields:
  - 0 - ll_from - The page in this wikipedia edition

# langlinks

- Just a few fields:
  - 0 - ll_from - The page in this wikipedia edition
  - 1 - ll_lang - The language it's linking to

# langlinks

- Just a few fields:
  - 0 - **ll_from** - The page in **this** wikipedia edition
  - 1 - **ll_lang** - The language it's linking to
  - 2 - **ll_title** - The title of the page in the **target** wikipedia edition

# langlinks

- Just a few fields:
  - 0 - **ll_from** - The page in **this** wikipedia edition
  - 1 - **ll_lang** - The language it's linking to
  - 2 - **ll_title** - The title of the page in the **target** wikipedia edition
- This is a good example of some of the problems of working with wikipedia data, or any other self organize collaboration platform

# langlinks

- Just a few fields:
  - 0 - **ll_from** - The page in **this** wikipedia edition
  - 1 - **ll_lang** - The language it's linking to
  - 2 - **ll_title** - The title of the page in the **target** wikipedia edition
- This is a good example of some of the problems of working with wikipedia data, or any other self organize collaboration platform
- Many of the file formats and conventions were created in an ad hoc way, to serve one very specific need and ended up becoming adopted as "standard".

# langlinks

- Just a few fields:
  - 0 - **ll_from** - The page in **this** wikipedia edition
  - 1 - **ll_lang** - The language it's linking to
  - 2 - **ll_title** - The title of the page in the **target** wikipedia edition
- This is a good example of some of the problems of working with wikipedia data, or any other self organize collaboration platform
- Many of the file formats and conventions were created in an ad hoc way, to serve one very specific need and ended up becoming adopted as "standard".
  - How can we convert the language/title pairs into a unique page_id in the target wikipedia?

# langlinks

- Just a few fields:
  - 0 - **ll_from** - The page in **this** wikipedia edition
  - 1 - **ll_lang** - The language it's linking to
  - 2 - **ll_title** - The title of the page in the **target** wikipedia edition
- This is a good example of some of the problems of working with wikipedia data, or any other self organize collaboration platform
- Many of the file formats and conventions were created in an ad hoc way, to serve one very specific need and ended up becoming adopted as "standard".
  - How can we convert the language/title pairs into a unique page_id in the target wikipedia?
  - Can we be sure that two pages didn't accidentally switch titles?

# langlinks

- Just a few fields:
  - 0 - **ll_from** - The page in **this** wikipedia edition
  - 1 - **ll_lang** - The language it's linking to
  - 2 - **ll_title** - The title of the page in the **target** wikipedia edition
- This is a good example of some of the problems of working with wikipedia data, or any other self organize collaboration platform
- Many of the file formats and conventions were created in an ad hoc way, to serve one very specific need and ended up becoming adopted as "standard".
  - How can we convert the language/title pairs into a unique page_id in the target wikipedia?
  - Can we be sure that two pages didn't accidentally switch titles?
  - As pages get edited, their titles change. To when (which revision) do these titles correspond to?

# langlinks

- Just a few fields:
  - 0 - **ll_from** - The page in **this** wikipedia edition
  - 1 - **ll_lang** - The language it's linking to
  - 2 - **ll_title** - The title of the page in the **target** wikipedia edition
- This is a good example of some of the problems of working with wikipedia data, or any other self organize collaboration platform
- Many of the file formats and conventions were created in an ad hoc way, to serve one very specific need and ended up becoming adopted as "standard".
  - How can we convert the language/title pairs into a unique page_id in the target wikipedia?
  - Can we be sure that two pages didn't accidentally switch titles?
  - As pages get edited, their titles change. To when (which revision) do these titles correspond to?
  - Does a link A -> B imply a link B -> A?

# langlinks

- convert the

data/acewiki-20170420-langlinks.sql.gz

- SQL file to CSV using the mysqldum2csv.py script.

```
./mysqldump_to_csv.py data/acewiki-<date>-langlinks.sql.gz | gzip -c > data/acewiki-<date>-langlinks.csv.gz
```

# geo_tags

- Several interesting fields:
  - 0 - **gt_id** - Unique geo tag ID
  - 1 - **gt_page_id** - Corresponding Page ID
  - 2 - **gt_globe** - Not all coordinates are on Earth (Mars, Moon, Venus, Titan, etc…)
  - 4 - **gt_lat** - Latitude
  - 5 - **gt_lon** - Longitude
  - 7 - **gt_type** - city, railwaystation, landmark, airport, etc…

# expat - (semi) sane XML parsing

https://docs.python.org/3/library/pyexpat.html

# expat - (semi) sane XML parsing

https://docs.python.org/3/library/pyexpat.html

- C library for parsing XML with bindings in most modern programming languages

# expat - (semi) sane XML parsing

https://docs.python.org/3/library/pyexpat.html

- C library for parsing XML with bindings in most modern programming languages

- Extremely fast

# expat - (semi) sane XML parsing

https://docs.python.org/3/library/pyexpat.html

- C library for parsing XML with bindings in most modern programming languages
- Extremely fast
- Well suited to handle large xml files:

# expat - (semi) sane XML parsing

https://docs.python.org/3/library/pyexpat.html

- C library for parsing XML with bindings in most modern programming languages
- Extremely fast
- Well suited to handle large xml files:
  - Stream oriented - Reads the file line by line

# expat - (semi) sane XML parsing

https://docs.python.org/3/library/pyexpat.html

- C library for parsing XML with bindings in most modern programming languages
- Extremely fast
- Well suited to handle large xml files:
  - Stream oriented - Reads the file line by line
  - Non-validating - doesn't check for the validity of the XML file (expensive and prone to failure)

# expat - (semi) sane XML parsing

https://docs.python.org/3/library/pyexpat.html

- C library for parsing XML with bindings in most modern programming languages
- Extremely fast
- Well suited to handle large xml files:
  - Stream oriented - Reads the file line by line
  - Non-validating - doesn't check for the validity of the XML file (expensive and prone to failure)
- In Python it lives inside the xml.parsers package

```
from xml.parsers import expat
```

# expat - (semi) sane XML parsing

https://docs.python.org/3/library/pyexpat.html

- C library for parsing XML with bindings in most modern programming languages
- Extremely fast
- Well suited to handle large xml files:
  - Stream oriented - Reads the file line by line
  - Non-validating - doesn't check for the validity of the XML file (expensive and prone to failure)
- In Python it lives inside the xml.parsers package

```
from xml.parsers import expat
```

- The .ParserCreate() method returns a new xmlparser instance

# expat - (semi) sane XML parsing

https://docs.python.org/3/library/pyexpat.html

# expat - (semi) sane XML parsing

https://docs.python.org/3/library/pyexpat.html

- Defines event handlers that get called whenever it encounters something “interesting”

# expat - (semi) sane XML parsing

https://docs.python.org/3/library/pyexpat.html

- Defines event handlers that get called whenever it encounters something "interesting"
- The default behavior is to do nothing (very efficient!) but you can override the ones that you are interested in.

# expat - (semi) sane XML parsing

https://docs.python.org/3/library/pyexpat.html

- Defines event handlers that get called whenever it encounters something "interesting"
- The default behavior is to do nothing (very efficient!) but you can override the ones that you are interested in.
- In particular:

# expat - (semi) sane XML parsing

https://docs.python.org/3/library/pyexpat.html

- Defines event handlers that get called whenever it encounters something "interesting"
- The default behavior is to do nothing (very efficient!) but you can override the ones that you are interested in.
- In particular:
    - .StartElementHandler(name, attrs) - every time it encounters a <name …>

# expat - (semi) sane XML parsing

https://docs.python.org/3/library/pyexpat.html

- Defines event handlers that get called whenever it encounters something "interesting"
- The default behavior is to do nothing (very efficient!) but you can override the ones that you are interested in.
- In particular:
    - .StartElementHandler(name, attrs) - every time it encounters a <name …>
    - .EndElementHandler(name) - whenever it encounter a </name>

# expat - (semi) sane XML parsing

https://docs.python.org/3/library/pyexpat.html

- Defines event handlers that get called whenever it encounters something "interesting"
- The default behavior is to do nothing (very efficient!) but you can override the ones that you are interested in.
- In particular:
  - .StartElementHandler(name, attrs) - every time it encounters a <name …>
  - .EndElementHandler(name) - whenever it encounter a </name>
  - .CharacterDataHandler(data) - any textual data in between the opening and closing of a tag:

# expat - (semi) sane XML parsing

https://docs.python.org/3/library/pyexpat.html

- Defines event handlers that get called whenever it encounters something "interesting"
- The default behavior is to do nothing (very efficient!) but you can override the ones that you are interested in.
- In particular:
  - .StartElementHandler(name, attrs) - every time it encounters a <name …>
  - .EndElementHandler(name) - whenever it encounter a </name>
  - .CharacterDataHandler(data) - any textual data in between the opening and closing of a tag:
    - <name>data</name>

# expat - (semi) sane XML parsing

https://docs.python.org/3/library/pyexpat.html

- Defines event handlers that get called whenever it encounters something "interesting"
- The default behavior is to do nothing (very efficient!) but you can override the ones that you are interested in.
- In particular:
  - .StartElementHandler(name, attrs) - every time it encounters a <name …>
  - .EndElementHandler(name) - whenever it encounter a </name>
  - .CharacterDataHandler(data) - any textual data in between the opening and closing of a tag:
    - <name>data</name>
    - If the amount of data between these two tags is too large, it sometimes results in multiple char_data events. You should always concatenate the results as you get it

# expat - (semi) sane XML parsing

https://docs.python.org/3/library/pyexpat.html

- Defines event handlers that get called whenever it encounters something "interesting"
- The default behavior is to do nothing (very efficient!) but you can override the ones that you are interested in.
- In particular:
    - .StartElementHandler(name, attrs) - every time it encounters a <name …>
    - .EndElementHandler(name) - whenever it encounter a </name>
    - .CharacterDataHandler(data) - any textual data in between the opening and closing of a tag:
        - <name>data</name>
        - If the amount of data between these two tags is too large, it sometimes results in multiple char_data events. You should always concatenate the results as you get it
- After you overwrite the relevant methods, you can process the file by providing a file handle to .ParserFile(fp)

<mediawiki xmlns="http://www.mediawiki.org/xml/export-0.10/" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:schemaLocation="http://www.mediawiki.org/xml/export-0.10/ http://www.mediawiki.org/xml/export-0.10.xsd"
version="0.10" xml:lang="en">
  <siteinfo>
    <sitename>Wikipedia</sitename>
    <dbname>enwiki</dbname>
    <base>https://en.wikipedia.org/wiki/Main_Page</base>
    <generator>MediaWiki 1.29.0-wmf.20</generator>
    <case>first-letter</case>
    <namespaces>
      <namespace key="-2" case="first-letter">Media</namespace>
      <namespace key="-1" case="first-letter">Special</namespace>
      <namespace key="0" case="first-letter" />
      <namespace key="1" case="first-letter">Talk</namespace>
      <namespace key="2" case="first-letter">User</namespace>
      <namespace key="3" case="first-letter">User talk</namespace>
      <namespace key="4" case="first-letter">Wikipedia</namespace>
      <namespace key="5" case="first-letter">Wikipedia talk</namespace>
      <namespace key="6" case="first-letter">File</namespace>
      <namespace key="7" case="first-letter">File talk</namespace>
      <namespace key="8" case="first-letter">MediaWiki</namespace>
      <namespace key="9" case="first-letter">MediaWiki talk</namespace>
      <namespace key="10" case="first-letter">Template</namespace>
      <namespace key="11" case="first-letter">Template talk</namespace>
      <namespace key="12" case="first-letter">Help</namespace>
      <namespace key="13" case="first-letter">Help talk</namespace>
      <namespace key="14" case="first-letter">Category</namespace>
      <namespace key="15" case="first-letter">Category talk</namespace>
      <namespace key="100" case="first-letter">Portal</namespace>
      <namespace key="101" case="first-letter">Portal talk</namespace>
      <namespace key="108" case="first-letter">Book</namespace>
      <namespace key="109" case="first-letter">Book talk</namespace>
      <namespace key="118" case="first-letter">Draft</namespace>
      <namespace key="119" case="first-letter">Draft talk</namespace>
      <namespace key="446" case="first-letter">Education Program</namespace>
      <namespace key="447" case="first-letter">Education Program talk</namespace>
      <namespace key="710" case="first-letter">TimedText</namespace>
      <namespace key="711" case="first-letter">TimedText talk</namespace>
      <namespace key="828" case="first-letter">Module</namespace>
      <namespace key="829" case="first-letter">Module talk</namespace>
      <namespace key="2300" case="first-letter">Gadget</namespace>
      <namespace key="2301" case="first-letter">Gadget talk</namespace>
      <namespace key="2302" case="case-sensitive">Gadget definition</namespace>
      <namespace key="2303" case="case-sensitive">Gadget definition talk</namespace>
    </namespaces>
  </siteinfo>

```
<mediawiki xmlns="http://www.mediawiki.org/xml/export-0.10/" xmlns:xsi="http://
www.w3.org/2001/XMLSchema-instance" xsi:schemaLocation="http://www.mediawiki.org/xml/
export-0.10/ http://www.mediawiki.org/xml/export-0.10.xsd" version="0.10"
xml:lang="en">
  <siteinfo>
    <sitename>Wikipedia</sitename>
    <dbname>enwiki</dbname>
    <base>https://en.wikipedia.org/wiki/Main_Page</base>
    <generator>MediaWiki 1.29.0-wmf.20</generator>
    <case>first-letter</case>
    <namespaces>
      <namespace key="-2" case="first-letter">Media</namespace>
      <namespace key="-1" case="first-letter">Special</namespace>
      <namespace key="0" case="first-letter" />
      <namespace key="1" case="first-letter">Talk</namespace>
      <namespace key="2" case="first-letter">User</namespace>
      <namespace key="3" case="first-letter">User talk</namespace>
      <namespace key="4" case="first-letter">Wikipedia</namespace>
      <namespace key="5" case="first-letter">Wikipedia talk</namespace>
      <namespace key="6" case="first-letter">File</namespace>
      <namespace key="7" case="first-letter">File talk</namespace>
      <namespace key="8" case="first-letter">MediaWiki</namespace>
      <namespace key="9" case="first-letter">MediaWiki talk</namespace>
      <namespace key="10" case="first-letter">Template</namespace>
      <namespace key="11" case="first-letter">Template talk</namespace>
      <namespace key="12" case="first-letter">Help</namespace>
      <namespace key="13" case="first-letter">Help talk</namespace>
      <namespace key="14" case="first-letter">Category</namespace>
      <namespace key="15" case="first-letter">Category talk</namespace>
      <namespace key="100" case="first-letter">Portal</namespace>
      <namespace key="101" case="first-letter">Portal talk</namespace>
      <namespace key="108" case="first-letter">Book</namespace>
      <namespace key="109" case="first-letter">Book talk</namespace>
      <namespace key="118" case="first-letter">Draft</namespace>
      <namespace key="119" case="first-letter">Draft talk</namespace>
      <namespace key="446" case="first-letter">Education Program</namespace>
      <namespace key="447" case="first-letter">Education Program talk</namespace
      <namespace key="710" case="first-letter">TimedText</namespace>
      <namespace key="711" case="first-letter">TimedText talk</namespace>
      <namespace key="828" case="first-letter">Module</namespace>
      <namespace key="829" case="first-letter">Module talk</namespace>
      <namespace key="2300" case="first-letter">Gadget</namespace>
      <namespace key="2301" case="first-letter">Gadget talk</namespace>
      <namespace key="2302" case="case-sensitive">Gadget definition</namespace>
      <namespace key="2303" case="case-sensitive">Gadget definition talk</namespace>
    </namespaces>
  </siteinfo>
```

**Wikipedia data structure**

**Namespaces**

| Subject namespaces | | Talk namespaces | |
|---|---|---|---|
| 0 | (Main/Article) | Talk | 1 |
| 2 | User | User talk | 3 |
| 4 | Wikipedia | Wikipedia talk | 5 |
| 6 | File | File talk | 7 |
| 8 | MediaWiki | MediaWiki talk | 9 |
| 10 | Template | Template talk | 11 |
| 12 | Help | Help talk | 13 |
| 14 | Category | Category talk | 15 |
| 100 | Portal | Portal talk | 101 |
| 108 | Book | Book talk | 109 |
| 118 | Draft | Draft talk | 119 |
| 446 | Education Program | Education Program talk | 447 |
| 710 | TimedText | TimedText talk | 711 |
| 828 | Module | Module talk | 829 |
| 2300 | Gadget | Gadget talk | 2301 |
| 2302 | Gadget definition | Gadget definition talk | 2303 |

**Virtual namespaces**

| | |
|---|---|
| -1 | Special |
| -2 | Media |

# Revision file format

```
<page>
  <title>Ôn Keuë</title>
  <ns>0</ns>
  <id>1</id>
  <revision>
    <id>1028</id>
    <timestamp>2008-04-13T07:53:23Z</timestamp>
    <contributor>
      <ip>125.162.38.87</ip>
    </contributor>
    <comment>New page: Jinoë droën neuh ka neutamong lam Wikipèdia  Acèh. Wikipèdia  Acèh
nyoë mantöng geu'ujoë, geukalön peuë ék na soë peudawôk peuë h'an. Meunyoë le nyang pakoë,
Wikipèdi...</comment>
    <model>wikitext</model>
    <format>text/x-wiki</format>
    <text id="815" bytes="3106" />
    <sha1>43iy7hfjh19xt1683z27ii0ie35z9am</sha1>
  </revision>
  <revision>
    <id>1029</id>
    <parentid>1028</parentid>
    <timestamp>2008-04-13T08:01:10Z</timestamp>
    <contributor>
      <username>Si Gam Acèh</username>
      <id>0</id>
    </contributor>
    <comment>Removing all content from page</comment>
    <model>wikitext</model>
    <format>text/x-wiki</format>
    <text id="816" bytes="0" />
    <sha1>phoiac9h4m842xq45sp7s6u21eteeq1</sha1>
  </revision>
</page>
```

data/page.xml

# Revision file format

- To extract the article revision information onto a csv file with the format:

page_id, revision_id, timestamp, title

In the file:

data/acewiki-20170420-stub-meta-history.xml.gz

```
<page>
  <title>Ôn Keuë</title>
  <ns>0</ns>
  <id>1</id>
  <revision>
    <id>1028</id>
    <timestamp>2008-04-13T07:53:23Z</timestamp>
    <contributor>
      <ip>125.162.38.87</ip>
    </contributor>
    <comment>New page: Jinoë droën neuh ka neutan
nyoë mantöng geu'ujoë, geukalön peuë ék na soë pe
Wikipèdi...</comment>
    <model>wikitext</model>
    <format>text/x-wiki</format>
    <text id="815" bytes="3106" />
    <sha1>43iy7hfjh19xt1683z27ii0ie35z9am</sha1>
  </revision>
  <revision>
    <id>1029</id>
    <parentid>1028</parentid>
    <timestamp>2008-04-13T08:01:10Z</timestamp>
    <contributor>
      <username>Si Gam Acèh</username>
      <id>0</id>
    </contributor>
    <comment>Removing all content from page</comm
    <model>wikitext</model>
    <format>text/x-wiki</format>
    <text id="816" bytes="0" />
    <sha1>phoiac9h4m842xq45sp7s6u21eteeq1</sha1>
  </revision>
</page>
```

data/page.xml

# Revision file format

- To extract the article revision information onto a csv file with the format:

page_id, revision_id, timestamp, title

In the file:

data/acewiki-20170420-stub-meta-history.xml.gz

There are 3 <id> tags. You have to keep track of which one you're in!

```
<page>
  <title>Ôn Keuë</title>
  <ns>0</ns>
  <id>1</id>
  <revision>
    <id>1028</id>
    <timestamp>2008-04-13T07:53:23Z</timestamp>
    <contributor>
      <ip>125.162.38.87</ip>
    </contributor>
    <comment>New page: Jinoë droën neuh ka neutan
nyoë mantöng geu'ujoë, geukalön peuë ék na soë pe
Wikipèdi...</comment>
    <model>wikitext</model>
    <format>text/x-wiki</format>
    <text id="815" bytes="3106" />
    <sha1>43iy7hfjh19xt1683z27ii0ie35z9am</sha1>
  </revision>
  <revision>
    <id>1029</id>
    <parentid>1028</parentid>
    <timestamp>2008-04-13T08:01:10Z</timestamp>
    <contributor>
      <username>Si Gam Acèh</username>
      <id>0</id>
    </contributor>
    <comment>Removing all content from page</comm
    <model>wikitext</model>
    <format>text/x-wiki</format>
    <text id="816" bytes="0" />
    <sha1>phoiac9h4m842xq45sp7s6u21eteeq1</sha1>
  </revision>
</page>
```

data/page.xml

# Revision file format

- To extract the article revision information onto a csv file with the format:

page_id, revision_id, timestamp, title

In the file:

data/acewiki-20170420-stub-meta-history.xml.gz

There are 3 <id> tags. You have to keep track of which one you're in!

```
<page>
  <title>Ôn Keuë</title>
  <ns>0</ns>
  <id>1</id>
  <revision>
    <id>1028</id>
    <timestamp>2008-04-13T07:53:23Z</timestamp>
    <contributor>
      <ip>125.162.38.87</ip>
    </contributor>
    <comment>New page: Jinoë droën neuh ka neutan
nyoë mantöng geu'ujoë, geukalön peuë ék na soë pe
Wikipèdi...</comment>
    <model>wikitext</model>
    <format>text/x-wiki</format>
    <text id="815" bytes="3106" />
    <sha1>43iy7hfjh19xt1683z27ii0ie35z9am</sha1>
  </revision>
  <revision>
    <id>1029</id>
    <parentid>1028</parentid>
    <timestamp>2008-04-13T08:01:10Z</timestamp>
    <contributor>
      <username>Si Gam Acèh</username>
      <id>0</id>
    </contributor>
    <comment>Removing all content from page</comm
    <model>wikitext</model>
    <format>text/x-wiki</format>
    <text id="816" bytes="0" />
    <sha1>phoiac9h4m842xq45sp7s6u21eteeq1</sha1>
  </revision>
</page>
```

data/page.xml

# Matching titles

# Matching titles

- As we saw before, matching titles and page_id is not easy. The only place where the two field appear together is in the revisions files. Fortunately, we already know how to process those.

# Matching titles

- As we saw before, matching titles and page_id is not easy. The only place where the two field appear together is in the revisions files. Fortunately, we already know how to process those.

- Even more fortunately, the Wikimedia foundation also makes available the stub-meta-current.xml.gz that have a similar format to the stub-meta-history.xml.gz files but include only the current revision of each page.

# Matching titles

- As we saw before, matching titles and page_id is not easy. The only place where the two field appear together is in the revisions files. Fortunately, we already know how to process those.

- Even more fortunately, the Wikimedia foundation also makes available the stub-meta-current.xml.gz that have a similar format to the stub-meta-history.xml.gz files but include only the current revision of each page.

- This is done in two parts:

# Matching titles

- As we saw before, matching titles and page_id is not easy. The only place where the two field appear together is in the revisions files. Fortunately, we already know how to process those.

- Even more fortunately, the Wikimedia foundation also makes available the stub-meta-current.xml.gz that have a similar format to the stub-meta-history.xml.gz files but include only the current revision of each page.

- This is done in two parts:

    - Convert data/abwiki-20170420-stub-meta-current.xml.gz to CSV

# Matching titles

- As we saw before, matching titles and page_id is not easy. The only place where the two field appear together is in the revisions files. Fortunately, we already know how to process those.

- Even more fortunately, the Wikimedia foundation also makes available the stub-meta-current.xml.gz that have a similar format to the stub-meta-history.xml.gz files but include only the current revision of each page.

- This is done in two parts:

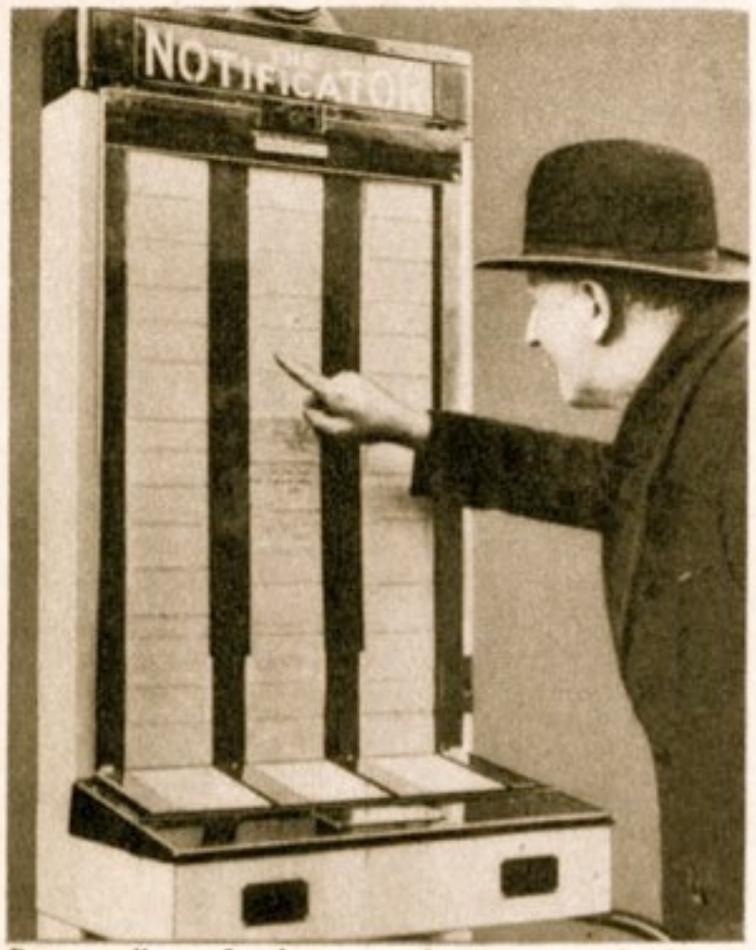
  - Convert data/abwiki-20170420-stub-meta-current.xml.gz to CSV

  - Use the newly generated data/abwiki-20170420-stub-meta-current.csv.gz to match the titles in data/acewiki-20170420-langlinks.csv.gz to the page_id in for the abwiki wikipedia edition.


# Lesson IV - Twitter

# Twitter

## Robot Messenger Displays Person-to-Person Notes In Public




For a small sum Londoners may leave messages for friends in public places. When written on "notificator," message moves up behind window, remaining in view for two hours.

TO AID persons who wish to make or cancel appointments or inform friends of their whereabouts, a robot message carrier has been introduced in London, England.

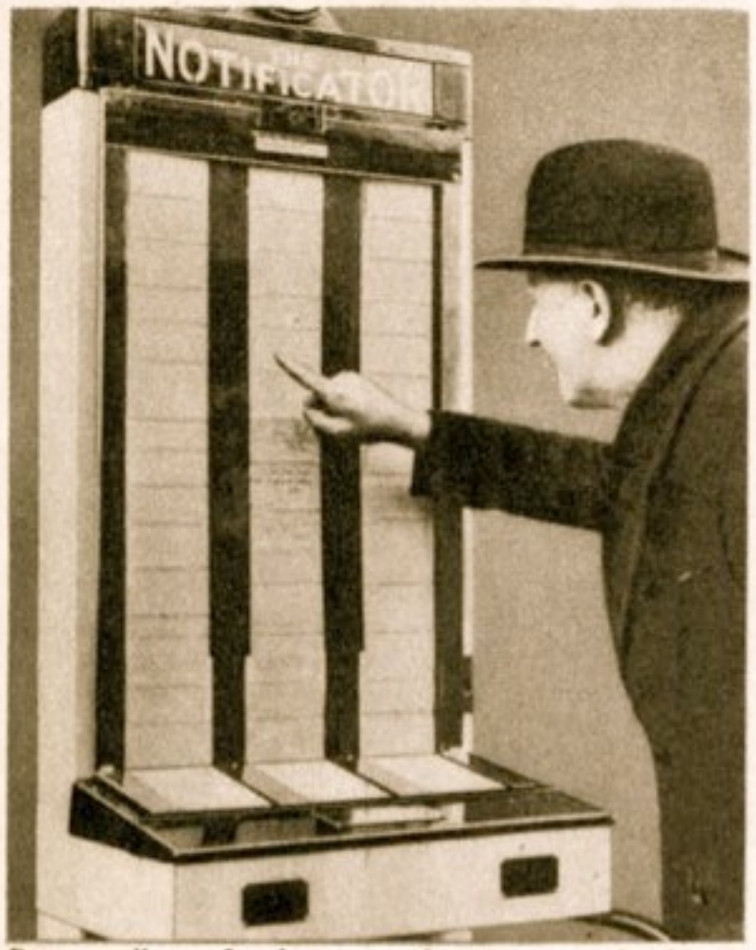
Known as the "notificator," the new machine is installed in streets, stores, railroad stations or other public places where individuals may leave messages for friends.

The user walks up on a small platform in front of the machine, writes a brief message on a continuous strip of paper and drops a coin in the slot. The inscription moves up behind a glass panel where it remains in public view for at least two hours so that the person for whom it is intended may have sufficient time to observe the note at the appointed place. The machine is similar in appearance to a candy-vending device.

Source: Modern Mechanix (Aug, 1935)

# Twitter

## Robot Messenger Displays Person-to-Person Notes In Public




For a small sum Londoners may leave messages for friends in public places. When written on "notificator," message moves up behind window, remaining in view for two hours.

TO AID persons who wish to make or cancel appointments or inform friends of their whereabouts, a robot message carrier has been introduced in London, England.

Known as the "notificator," the new machine is installed in streets, stores, railroad stations or other public places where individuals may leave messages for friends.

The user walks up on a small platform in front of the machine, writes a brief message on a continuous strip of paper and drops a coin in the slot. The inscription moves up behind a glass panel where it remains in public view for at least two hours so that the person for whom it is intended may have sufficient time to observe the note at the appointed place. The machine is similar in appearance to a candy-vending device.

Source: Modern Mechanix (Aug, 1935)

# Anatomy of a Tweet

# Anatomy of a Tweet


```
[u'contributors',
 u'truncated',
 u'text',
 u'in_reply_to_status_id',
 u'id',
 u'favorite_count',
 u'source',
 u'retweeted',
 u'coordinates',
 u'entities',
 u'in_reply_to_screen_name',
 u'in_reply_to_user_id',
 u'retweet_count',
 u'id_str',
 u'favorited',
 u'user',
 u'geo',
 u'in_reply_to_user_id_str',
 u'possibly_sensitive',
 u'lang',
 u'created_at',
 u'in_reply_to_status_id_str',
 u'place',
 u'metadata']
```

# Anatomy of a Tweet

```
[u'contributors',
 u'truncated',
 u'text',
 u'in_reply_to_status_id',
 u'id',
 u'favorite_count',
 u'source',
 u'retweeted',
 u'coordinates',
 u'entities',
 u'in_reply_to_screen_name',
 u'in_reply_to_user_id',
 u'retweet_count',
 u'id_str',
 u'favorited',
 u'user',
 u'geo',
 u'in_reply_to_user_id_str',
 u'possibly_sensitive',
 u'lang',
 u'created_at',
 u'in_reply_to_status_id_str',
 u'place',
 u'metadata']
```

```
[u'follow_request_sent',
 u'profile_use_background_image',
 u'default_profile_image',
 u'id',
 u'profile_background_image_url_https',
 u'verified',
 u'profile_text_color',
 u'profile_image_url_https',
 u'profile_sidebar_fill_color',
 u'entities',
 u'followers_count',
 u'profile_sidebar_border_color',
 u'id_str',
 u'profile_background_color',
 u'listed_count',
 u'is_translation_enabled',
 u'utc_offset',
 u'statuses_count',
 u'description',
 u'friends_count',
 u'location',
 u'profile_link_color',
 u'profile_image_url',
 u'following',
 u'geo_enabled',
 u'profile_banner_url',
 u'profile_background_image_url',
 u'screen_name',
 u'lang',
 u'profile_background_tile',
 u'favourites_count',
 u'name',
 u'notifications',
 u'url',
 u'created_at',
 u'contributors_enabled',
 u'time_zone',
 u'protected',
 u'default_profile',
 u'is_translator']
```

# Anatomy of a Tweet

```
[u'contributors',
 u'truncated',
 u'text',
 u'in_reply_to_status_id',
 u'id',
 u'favorite_count',
 u'source',
 u'retweeted',
 u'coordinates',
 u'entities',
 u'in_reply_to_screen_name',
 u'in_reply_to_user_id',
 u'retweet_count',
 u'id_str',
 u'favorited',
 u'user',
 u'geo',
 u'in_reply_to_user_id_str',
 u'possibly_sensitive',
 u'lang',
 u'created_at',
 u'in_reply_to_status_id_str',
 u'place',
 u'metadata']
```

```
u"I'm at Terminal Rodovi\xe1rio de Feira de Santana
 (Feira de Santana, BA) http://t.co/WirvdHwYMq"
```

```
u'<a href="http://foursquare.com" rel="nofollow">
  foursquare</a>'
```

```
[u'symbols',
 u'user_mentions',
 u'hashtags',
 u'urls']
```

```
[u'type',
 u'coordinates']
```

# Anatomy of a Tweet

```
[u'contributors',
 u'truncated',
 u'text',
 u'in_reply_to_status_id',
 u'id',
 u'favorite_count',
 u'source',
 u'retweeted',
 u'coordinates',
 u'entities',
 u'in_reply_to_screen_name',
 u'in_reply_to_user_id',
 u'retweet_count',
 u'id_str',
 u'favorited',
 u'user',
 u'geo',
 u'in_reply_to_user_id_str',
 u'possibly_sensitive',
 u'lang',
 u'created_at',
 u'in_reply_to_status_id_str',
 u'place',
 u'metadata']
```

```
u"I'm at Terminal Rodovi\xe1rio de Feira de Santana
 (Feira de Santana, BA) http://t.co/WirvdHwYMq"
```

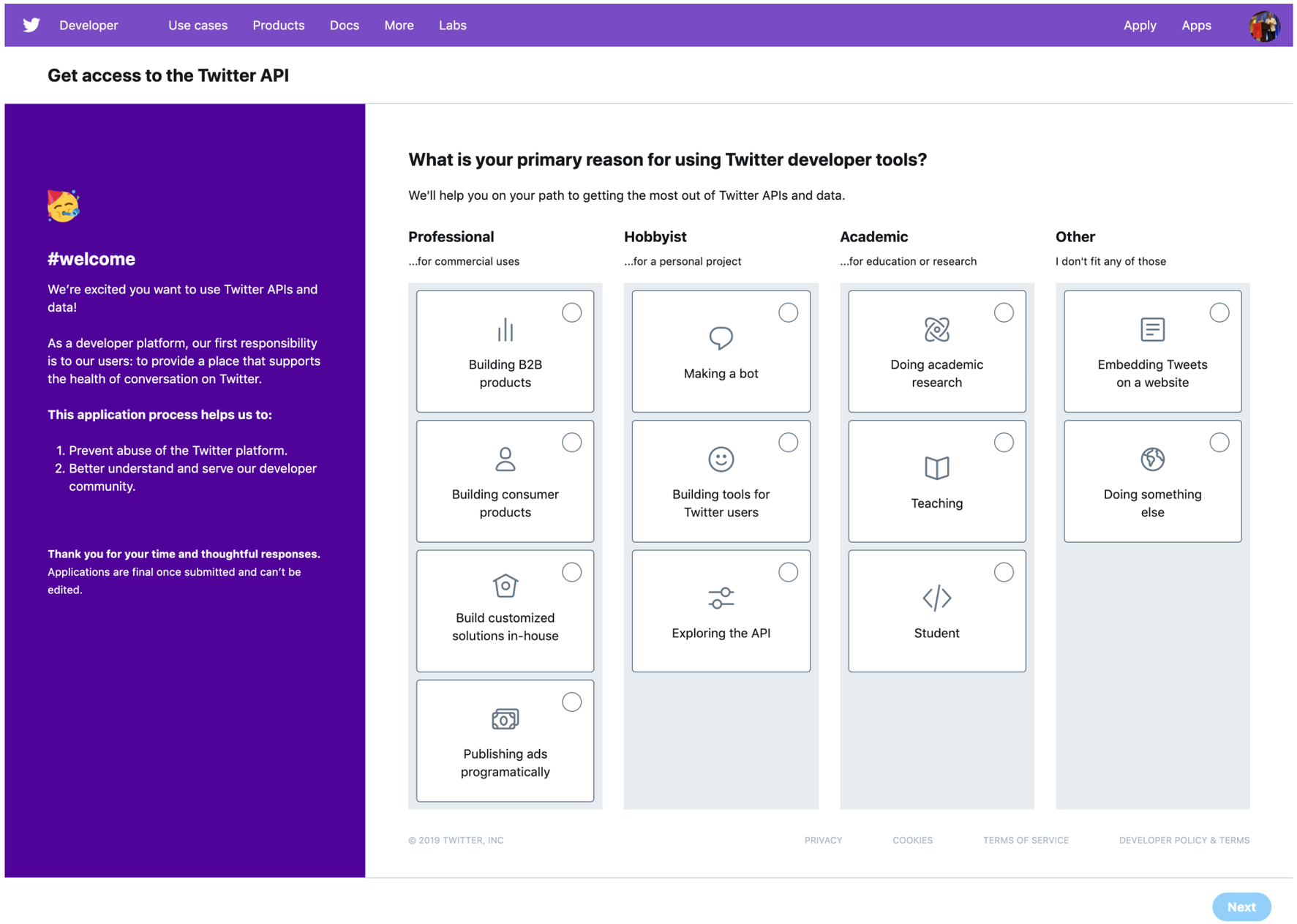
```
u'<a href="http://foursquare.com" rel="nofollow">
  foursquare</a>'
```

```
[u'symbols',
 u'user_mentions',
 u'hashtags',
 u'urls']
```

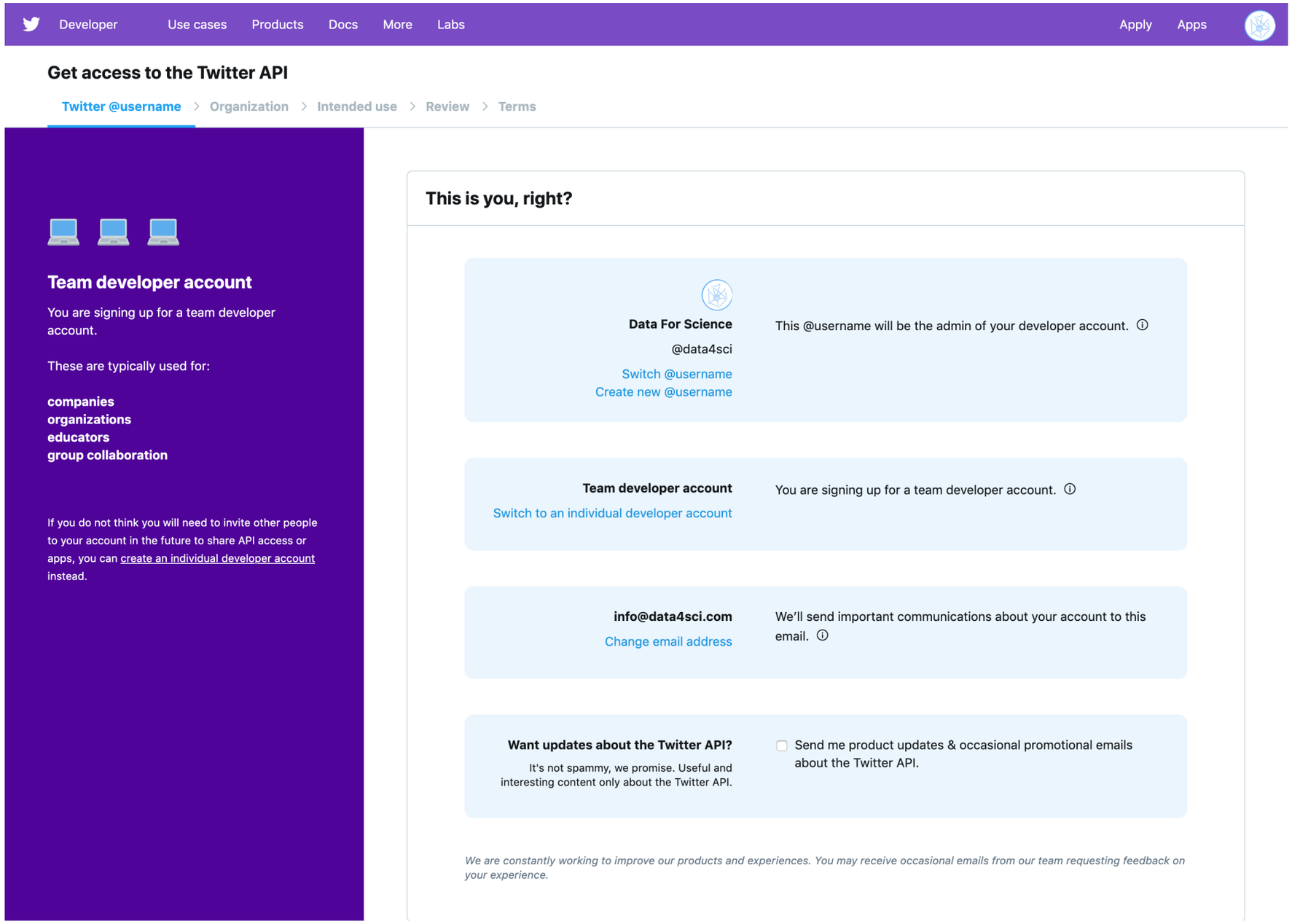
```
{u'display_url': u'4sq.com/1k5MeYF',
 u'expanded_url': u'http://4sq.com/1k5MeYF',
 u'indices': [70, 92],
 u'url': u'http://t.co/WirvdHwYMq'}
```

```
[u'type',
 u'coordinates']
```

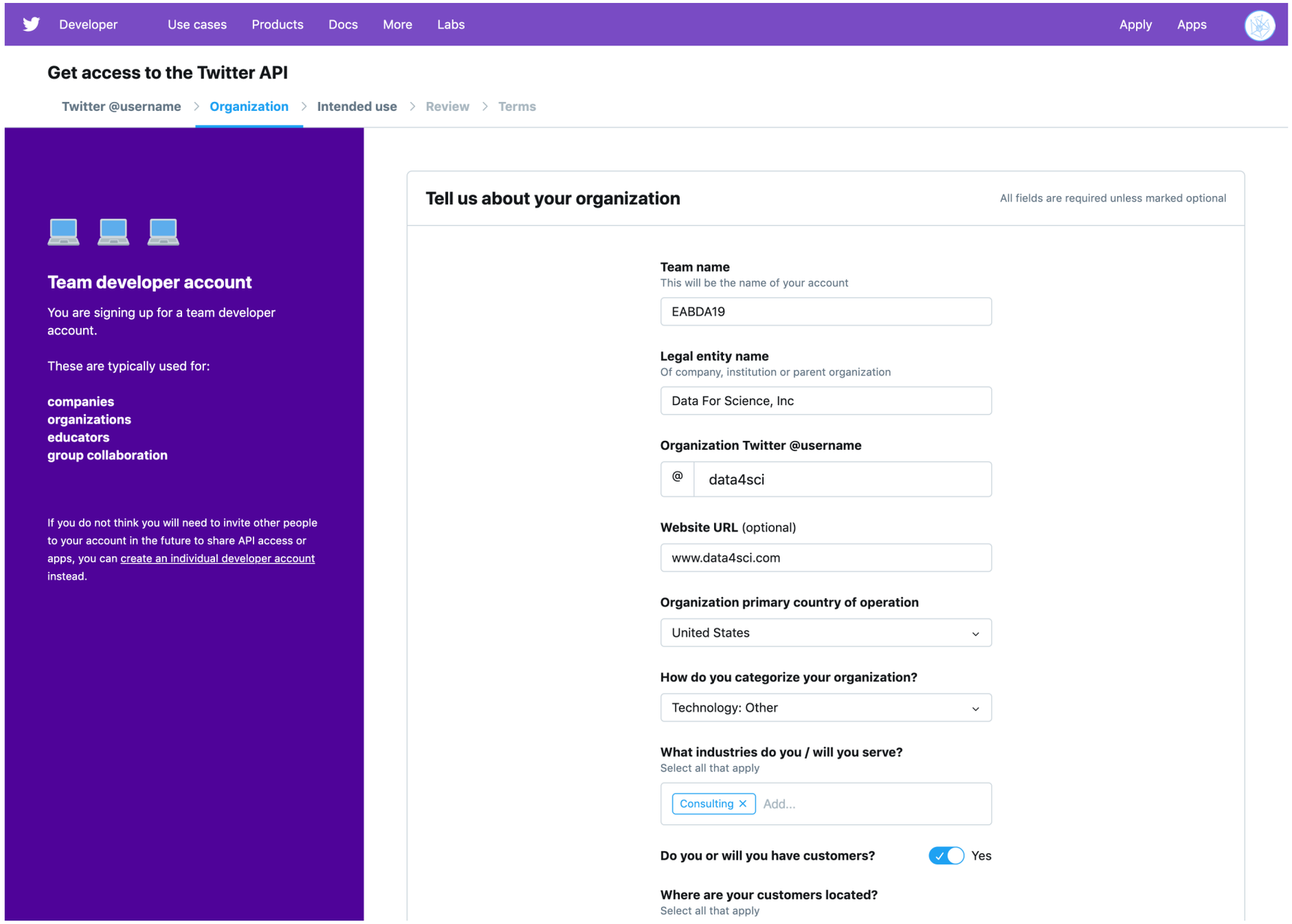
# Registering an Application



# Registering an Application



# Registering an Application



# Registering an Application

Developer Use cases Products Docs More Labs Apply Apps

**Get access to the Twitter API**

Twitter @username > Organization > Intended use > Review > Terms

**Key things to keep in mind**

This section of the application helps us ensure that users of our data are complying with Twitter's Developer Policies.

This review process and our policies help us keep Twitter a safe and healthy space for public conversation.

**Restricted uses**

Some activities (like surveillance) are never allowed on Twitter. Take a look at our restricted uses page to ensure that your use case is policy-compliant before you submit an application.

**Automation**

Be sure to review the automation rules if you plan on enabling any sort of automated activity on the platform.

**Be thorough**

We need to completely understand your use case before we can approve it. So, please include as much detail as possible in your application.

**How will you use the Twitter API or Twitter data?**

All fields are required unless marked optional

**In your words**

In English, please describe how you plan to use Twitter data and/or APIs. For students and teachers, please include the name of the school, the name of the instructor and the course number (if available). The more detailed the response, the easier it is to review and approve.

Please be thoughtful and thorough

Required

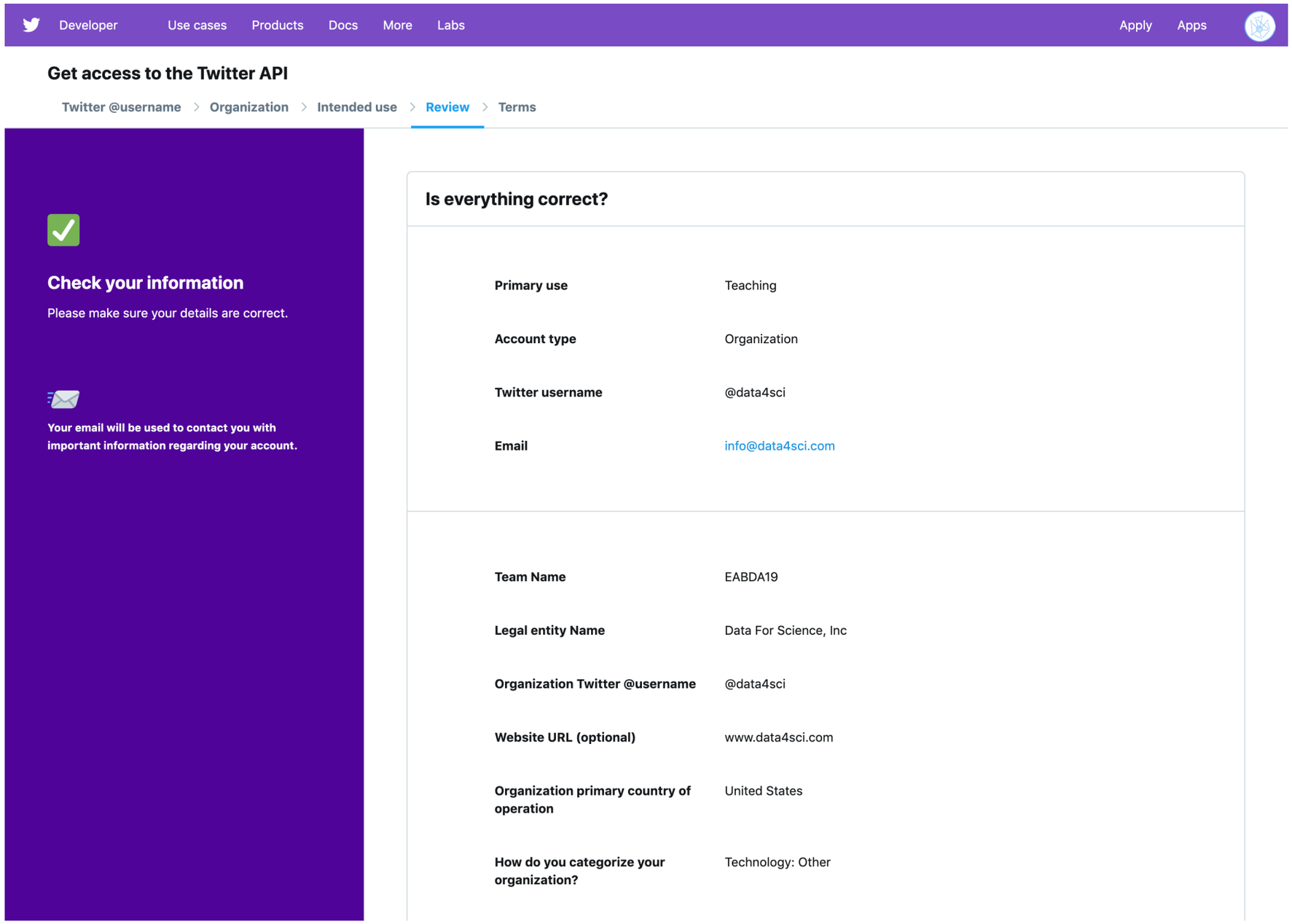
Response must be at least 200 characters 200

**The specifics**

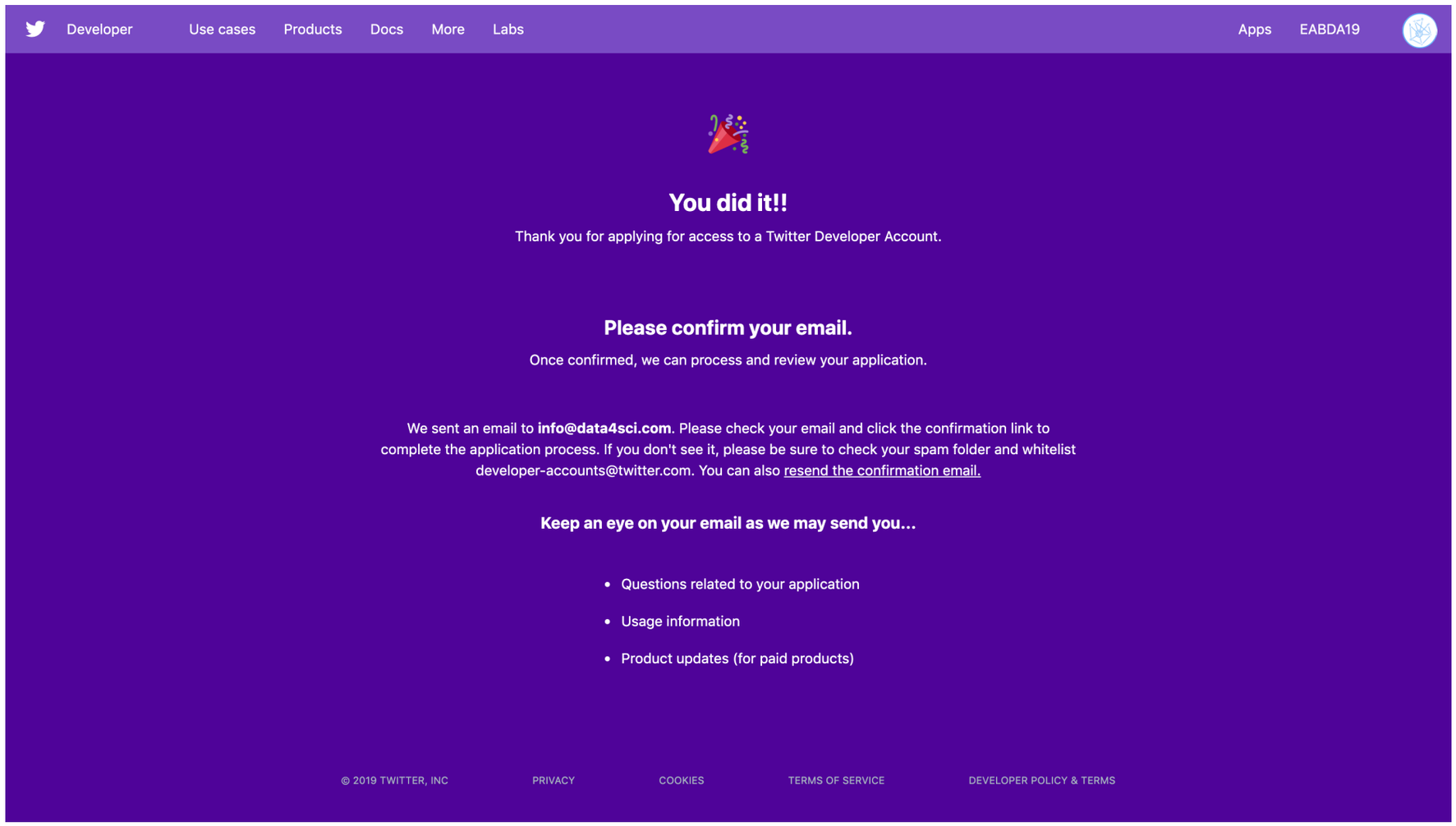
Please answer each of the following with as much detail and accuracy as possible. Failure to do so could result in delays to your access to the Twitter developer platform or rejected applications.

Are you planning to analyze Twitter data? No

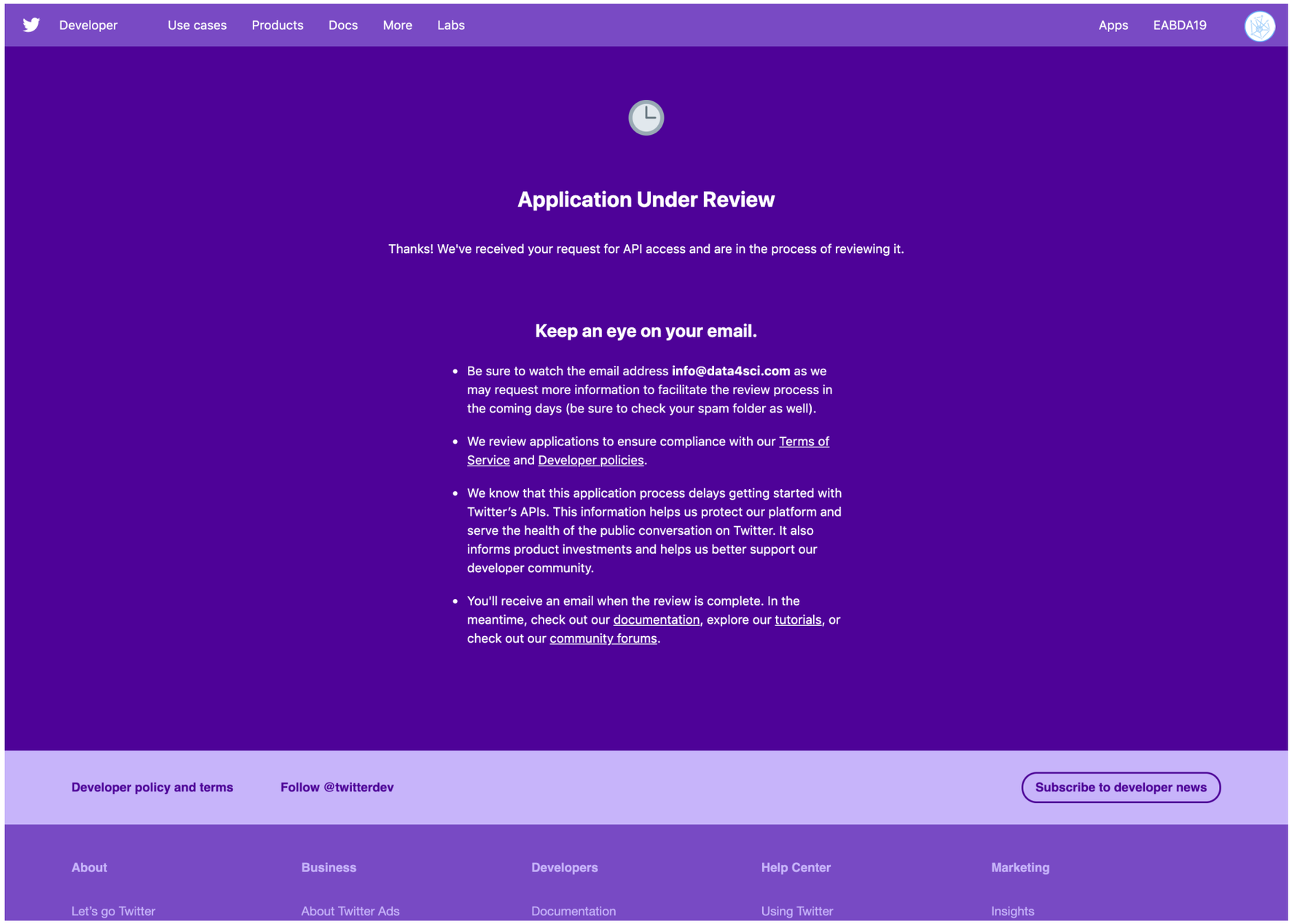
# Registering an Application



# Registering an Application



# Registering an Application



# API Basics

developer.twitter.com/en/docs/

# API Basics

developer.twitter.com/en/docs/

- The twitter module provides the oauth interface. We just need to provide the right credentials.

# API Basics

developer.twitter.com/en/docs/

- The twitter module provides the oauth interface. We just need to provide the right credentials.

- Best to keep the credentials in a dict and parametrize our calls with the dict key. This way we can switch between different accounts easily.

# API Basics

developer.twitter.com/en/docs/

- The twitter module provides the oauth interface. We just need to provide the right credentials.

- Best to keep the credentials in a dict and parametrize our calls with the dict key. This way we can switch between different accounts easily.

- .Twitter(auth) takes an OAuth instance as argument and returns a Twitter object that we can use to interact with the API

# API Basics

developer.twitter.com/en/docs/

- The twitter module provides the oauth interface. We just need to provide the right credentials.
- Best to keep the credentials in a dict and parametrize our calls with the dict key. This way we can switch between different accounts easily.
- .Twitter(auth) takes an OAuth instance as argument and returns a Twitter object that we can use to interact with the API
- Twitter methods mimic API structure

# API Basics

developer.twitter.com/en/docs/

- The twitter module provides the oauth interface. We just need to provide the right credentials.

- Best to keep the credentials in a dict and parametrize our calls with the dict key. This way we can switch between different accounts easily.

- .Twitter(auth) takes an OAuth instance as argument and returns a Twitter object that we can use to interact with the API

- Twitter methods mimic API structure

- 4 basic types of objects:

# API Basics

developer.twitter.com/en/docs/

- The twitter module provides the oauth interface. We just need to provide the right credentials.

- Best to keep the credentials in a dict and parametrize our calls with the dict key. This way we can switch between different accounts easily.

- .Twitter(auth) takes an OAuth instance as argument and returns a Twitter object that we can use to interact with the API

- Twitter methods mimic API structure

- 4 basic types of objects:

    - Tweets

# API Basics

developer.twitter.com/en/docs/

- The twitter module provides the oauth interface. We just need to provide the right credentials.
- Best to keep the credentials in a dict and parametrize our calls with the dict key. This way we can switch between different accounts easily.
- .Twitter(auth) takes an OAuth instance as argument and returns a Twitter object that we can use to interact with the API
- Twitter methods mimic API structure
- 4 basic types of objects:
  - Tweets
  - Users

# API Basics

developer.twitter.com/en/docs/

- The twitter module provides the oauth interface. We just need to provide the right credentials.
- Best to keep the credentials in a dict and parametrize our calls with the dict key. This way we can switch between different accounts easily.
- .Twitter(auth) takes an OAuth instance as argument and returns a Twitter object that we can use to interact with the API
- Twitter methods mimic API structure
- 4 basic types of objects:
  - Tweets
  - Users
  - Entities

# API Basics

developer.twitter.com/en/docs/

- The twitter module provides the oauth interface. We just need to provide the right credentials.
- Best to keep the credentials in a dict and parametrize our calls with the dict key. This way we can switch between different accounts easily.
- .Twitter(auth) takes an OAuth instance as argument and returns a Twitter object that we can use to interact with the API
- Twitter methods mimic API structure
- 4 basic types of objects:
  - Tweets
  - Users
  - Entities
  - Places

# Authenticating with the API

```
import tweepy
from twitter_accounts import accounts

app = accounts["social"]

auth = tweepy.OAuthHandler(app["api_key"], app["api_secret"])
auth.set_access_token(app["token"], app["token_secret"])

twitter_api = tweepy.API(auth)
```

- In the remainder of this course, the accounts dict will live inside the twitter_accounts.py file
- 4 basic types of objects:
  - Tweets
  - Users
  - Entities
  - Places

# Searching for Tweets

developer.twitter.com/en/docs/

# Searching for Tweets

developer.twitter.com/en/docs/

- .search(query, count)

# Searching for Tweets

developer.twitter.com/en/docs/

- .search(query, count)
    - query is the content to search for

# Searching for Tweets

developer.twitter.com/en/docs/

- .search(query, count)
  - query is the content to search for
  - count is the maximum number of results to return

# Searching for Tweets

developer.twitter.com/en/docs/

- .search(query, count)
  - query is the content to search for
  - count is the maximum number of results to return
- returns SearchResults object with a list of "statuses" and some other meta_data:

```
max_id: 438088492577345536
since_id: 0
refresh_url: '?since_id=4380884925773455536&q=soccer&include_entities=1'
completed_in: 0.027
query': 'soccer'
count': 15
next_results: '?max_id=438088485145034752&q=soccer&include_entities=1'
```

# Searching for Tweets

developer.twitter.com/en/docs/

- .search(query, count)
  - query is the content to search for
  - count is the maximum number of results to return
- returns SearchResults object with a list of "statuses" and some other meta_data:

```
max_id: 438088492577345536
since_id: 0
refresh_url: '?since_id=438088492577345536&q=soccer&include_entities=1'
completed_in: 0.027
query': 'soccer'
count': 15
next_results: '?max_id=438088485145034752&q=soccer&include_entities=1'
```

- next_results can be used to get the next page of results

# Streaming data

developer.twitter.com/en/docs/

# Streaming data

developer.twitter.com/en/docs/

- The Streaming api provides realtime data, subject to filters

# Streaming data

developer.twitter.com/en/docs/

- The Streaming api provides realtime data, subject to filters

- Use Stream instead of API object (.Stream(auth=auth, listener=listen))

# Streaming data

developer.twitter.com/en/docs/

- The Streaming api provides realtime data, subject to filters

- Use Stream instead of API object (.Stream(auth=auth, listener=listen))

- Stream also requires a StreamListener object to be passed at instantiation.

# Streaming data

developer.twitter.com/en/docs/

- The Streaming api provides realtime data, subject to filters

- Use Stream instead of API object (.Stream(auth=auth, listener=listen))

- Stream also requires a StreamListener object to be passed at instantiation.

- You must implement your own version of StreamListener to handle the data you are interested in.

# Streaming data

developer.twitter.com/en/docs/

- The Streaming api provides realtime data, subject to filters

- Use Stream instead of API object (.Stream(auth=auth, listener=listen))

- Stream also requires a StreamListener object to be passed at instantiation.

- You must implement your own version of StreamListener to handle the data you are interested in.

- .filter(track=[q]) will obtain tweets that match the query q in real time. For each tweet received, it will call the StreamListener.on_data to process the data and StreamListener.on_error in case it encounters any error. These are the two main functions you must override in your implementation

# Streaming data

developer.twitter.com/en/docs/

```
class StdOutListener(tweepy.StreamListener):
    def on_status(self, status):
        print(status.text)
        return True

    def on_error(self, status):
        print(status)
```

# User profiles

developer.twitter.com/en/docs/

# User profiles

developer.twitter.com/en/docs/

- .get_user() returns user profile information for a given user_id or screen_name
- It returns a User object with detailed information about the user, including:
  - name - The user "display" name
  - location - Their stated location
  - description - Their bio
  - followers_count/friends_count/statuses_count - number of followers, friends and tweets
  - created_at - Date that the account was created
  - verified - Wether the account has been verified or not
  - status - Their most recent tweet

# Social Connections

developer.twitter.com/en/docs/

- .friends() and .followers() returns a list of a users friends or followers for a given screen_name or user_id
- If the number of friends or followers is smaller than the count requested, these methods simply return a list of results. If the number of friends or followers exceeds the count requested, these methods also return pagination information.
- To facilitate the processing of multiple pages of results, we can simply use a Cursor, a wrapper object that simply takes care of all the pagination details in the background.
  - Instead of calling the .friends() and .followers() methods (or any other than can return multiple pages of results) we simply pass this function to the Cursor module
  - Any other arguments we wish to pass along can be provided to Cursor module directly:

```
screen_name = "stephen_wolfram"

for follower in tweepy.Cursor(twitter_api.followers, screen_name=screen_name):
    print(i, follower.screen_name)
```

# User Timeline

developer.twitter.com/en/docs/

# User Timeline

developer.twitter.com/en/docs/

- user_timeline() returns a set of tweets posted by a single user

# User Timeline

developer.twitter.com/en/docs/

- user_timeline() returns a set of tweets posted by a single user
- Important options:

# User Timeline

developer.twitter.com/en/docs/

- user_timeline() returns a set of tweets posted by a single user
- Important options:
    - screen_name - screen_name of the user we are interested in

# User Timeline

developer.twitter.com/en/docs/

- user_timeline() returns a set of tweets posted by a single user
- Important options:
    - screen_name - screen_name of the user we are interested in
    - count=200 number of tweets to return in each call

# User Timeline

developer.twitter.com/en/docs/

- user_timeline() returns a set of tweets posted by a single user
- Important options:
    - screen_name - screen_name of the user we are interested in
    - count=200 number of tweets to return in each call
    - max_id=1234 to include only tweets with an id lower than 1234

# User Timeline

developer.twitter.com/en/docs/

- user_timeline() returns a set of tweets posted by a single user
- Important options:
  - screen_name - screen_name of the user we are interested in
  - count=200 number of tweets to return in each call
  - max_id=1234 to include only tweets with an id lower than 1234
- Returns at most 200 tweets in each call. Can get all of a users tweets (up to 3200) with multiple calls using max_id or a Cursor

# Streaming Geocoded data

developer.twitter.com/en/docs/

# Streaming Geocoded data

developer.twitter.com/en/docs/

- The Streaming api provides realtime data, subject to filters

# Streaming Geocoded data

developer.twitter.com/en/docs/

- The Streaming api provides realtime data, subject to filters

- Use TwitterStream instead of Twitter object (.TwitterStream(auth=twitter_api.auth))

# Streaming Geocoded data

developer.twitter.com/en/docs/

- The Streaming api provides realtime data, subject to filters
- Use TwitterStream instead of Twitter object (.TwitterStream(auth=twitter_api.auth))
- .status.filter(track=q) will return tweets that match the query q in real time

# Streaming Geocoded data

developer.twitter.com/en/docs/

- The Streaming api provides realtime data, subject to filters
- Use TwitterStream instead of Twitter object (.TwitterStream(auth=twitter_api.auth))
- .status.filter(track=q) will return tweets that match the query q in real time
- Returns generator that you can iterate over

# Streaming Geocoded data

developer.twitter.com/en/docs/

- The Streaming api provides realtime data, subject to filters

- Use TwitterStream instead of Twitter object (.TwitterStream(auth=twitter_api.auth))

- .status.filter(track=q) will return tweets that match the query q in real time

- Returns generator that you can iterate over

- .status.filter(locations=bb) will return tweets that occur within the bounding box bb in real time

# Streaming Geocoded data

developer.twitter.com/en/docs/

- The Streaming api provides realtime data, subject to filters
- Use TwitterStream instead of Twitter object (.TwitterStream(auth=twitter_api.auth))
- .status.filter(track=q) will return tweets that match the query q in real time
- Returns generator that you can iterate over
- .status.filter(locations=bb) will return tweets that occur within the bounding box bb in real time
- bb is a comma separated pair of lon/lat coordinates.

# Streaming Geocoded data

developer.twitter.com/en/docs/

- The Streaming api provides realtime data, subject to filters
- Use TwitterStream instead of Twitter object (.TwitterStream(auth=twitter_api.auth))
- .status.filter(track=q) will return tweets that match the query q in real time
- Returns generator that you can iterate over
- .status.filter(locations=bb) will return tweets that occur within the bounding box bb in real time
- bb is a comma separated pair of lon/lat coordinates.
  - -180,-90,180,90 - World

# Streaming Geocoded data

developer.twitter.com/en/docs/

- The Streaming api provides realtime data, subject to filters
- Use TwitterStream instead of Twitter object (.TwitterStream(auth=twitter_api.auth))
- .status.filter(track=q) will return tweets that match the query q in real time
- Returns generator that you can iterate over
- .status.filter(locations=bb) will return tweets that occur within the bounding box bb in real time
- bb is a comma separated pair of lon/lat coordinates.
  - -180,-90,180,90 - World
  - -74,40,-73,41 - NYC

# Streaming Geocoded data

developer.twitter.com/en/docs/

# Streaming Geocoded data

developer.twitter.com/en/docs/

- The Streaming api provides realtime data, subject to filters

# Streaming Geocoded data

developer.twitter.com/en/docs/

- The Streaming api provides realtime data, subject to filters
- Use Stream instead of API object (.Stream(auth=auth, listener=listen))

# Streaming Geocoded data

developer.twitter.com/en/docs/

- The Streaming api provides realtime data, subject to filters
- Use Stream instead of API object (.Stream(auth=auth, listener=listen))
- Stream also requires a StreamListener object to be passed at instantiation.

# Streaming Geocoded data

developer.twitter.com/en/docs/

- The Streaming api provides realtime data, subject to filters
- Use Stream instead of API object (.Stream(auth=auth, listener=listen))
- Stream also requires a StreamListener object to be passed at instantiation.
- You must implement your own version of StreamListener to handle the data you are interested in.

# Streaming Geocoded data

developer.twitter.com/en/docs/

- The Streaming api provides realtime data, subject to filters
- Use Stream instead of API object (.Stream(auth=auth, listener=listen))
- Stream also requires a StreamListener object to be passed at instantiation.
- You must implement your own version of StreamListener to handle the data you are interested in.
- .filter(track=[q]) will obtain tweets that match the query q in real time. For each tweet received, it will call the StreamListener.on_data to process the data and StreamListener.on_error in case it encounters any error. These are the two main functions you must override in your implementation

# Streaming Geocoded data

developer.twitter.com/en/docs/

- The Streaming api provides realtime data, subject to filters
- Use Stream instead of API object (.Stream(auth=auth, listener=listen))
- Stream also requires a StreamListener object to be passed at instantiation.
- You must implement your own version of StreamListener to handle the data you are interested in.
- .filter(track=[q]) will obtain tweets that match the query q in real time. For each tweet received, it will call the StreamListener.on_data to process the data and StreamListener.on_error in case it encounters any error. These are the two main functions you must override in your implementation
- .filter(locations=bb) will return tweets that occur within the bounding box bb in real time

# Streaming Geocoded data

developer.twitter.com/en/docs/

- The Streaming api provides realtime data, subject to filters
- Use Stream instead of API object (.Stream(auth=auth, listener=listen))
- Stream also requires a StreamListener object to be passed at instantiation.
- You must implement your own version of StreamListener to handle the data you are interested in.
- .filter(track=[q]) will obtain tweets that match the query q in real time. For each tweet received, it will call the StreamListener.on_data to process the data and StreamListener.on_error in case it encounters any error. These are the two main functions you must override in your implementation
- .filter(locations=bb) will return tweets that occur within the bounding box bb in real time
- bb is a list of lon/lat coordinates.

# Streaming Geocoded data

developer.twitter.com/en/docs/

- The Streaming api provides realtime data, subject to filters
- Use Stream instead of API object (.Stream(auth=auth, listener=listen))
- Stream also requires a StreamListener object to be passed at instantiation.
- You must implement your own version of StreamListener to handle the data you are interested in.
- .filter(track=[q]) will obtain tweets that match the query q in real time. For each tweet received, it will call the StreamListener.on_data to process the data and StreamListener.on_error in case it encounters any error. These are the two main functions you must override in your implementation
- .filter(locations=bb) will return tweets that occur within the bounding box bb in real time
- bb is a list of lon/lat coordinates.
  - -180,-90,180,90 - World

# Streaming Geocoded data

developer.twitter.com/en/docs/

- The Streaming api provides realtime data, subject to filters
- Use Stream instead of API object (.Stream(auth=auth, listener=listen))
- Stream also requires a StreamListener object to be passed at instantiation.
- You must implement your own version of StreamListener to handle the data you are interested in.
- .filter(track=[q]) will obtain tweets that match the query q in real time. For each tweet received, it will call the StreamListener.on_data to process the data and StreamListener.on_error in case it encounters any error. These are the two main functions you must override in your implementation
- .filter(locations=bb) will return tweets that occur within the bounding box bb in real time
- bb is a list of lon/lat coordinates.
  - -180,-90,180,90 - World
  - -74,40,-73,41 - NYC

# Streaming Geocoded data

developer.twitter.com/en/docs/

- The Streaming api provides realtime data, subject to filters
- Use Stream instead of API object (.Stream(auth=auth, listener=listen))
- Stream also requires a StreamListener object to be passed at instantiation.
- You must implement your own version of StreamListener to handle the data you are interested in.
- .filter(track=[q]) will obtain tweets that match the query q in real time. For each tweet received, it will call the StreamListener.on_data to process the data and StreamListener.on_error in case it encounters any error. These are the two main functions you must override in your implementation
- .filter(locations=bb) will return tweets that occur within the bounding box bb in real time
- bb is a list of lon/lat coordinates.
  - -180,-90,180,90 - World
  - -74,40,-73,41 - NYC
- locations acts as a parallel filter to track, so the results obtained in this way will match the query q, the bounding box bb, or both.

# Events

www.data4sci.com/newsletter


**Graphs and Network Algorithms from Scratch**

- Sep 16, 2019 - 5am-9pm (PST)

**Deep Learning from Scratch**

- Sept 30, 2019 - 5am-9pm (PST)

**Deep Learning from Scratch**

- Sept 24, 2019 - Strata NYC

Strata Data Conference
PRESENTED WITH CLOUDERA
Sep 23–26, 2019

livelessons
Natural Language Processing
Bruno Gonçalves
video

**Natural Language Processing (NLP) from Scratch**

http://bit.ly/LiveLessonNLP - On Demand