



Cleaning data (part 2)

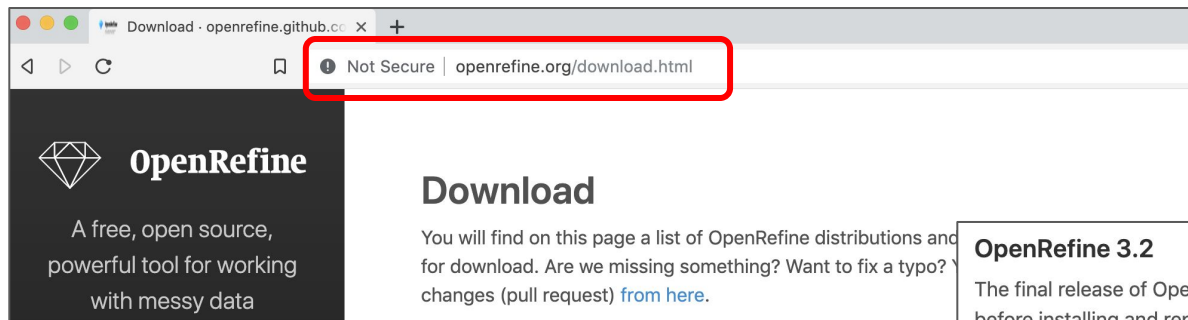
Advanced cleaning with OpenRefine

link for slides: <http://bit.ly/bu-cleaning-data-p2>

First steps:

Make sure you've downloaded and installed OpenRefine:

<http://openrefine.org/download.html>



1) Go to the Downloads page

2) Select the appropriate application for your computer

OpenRefine 3.2

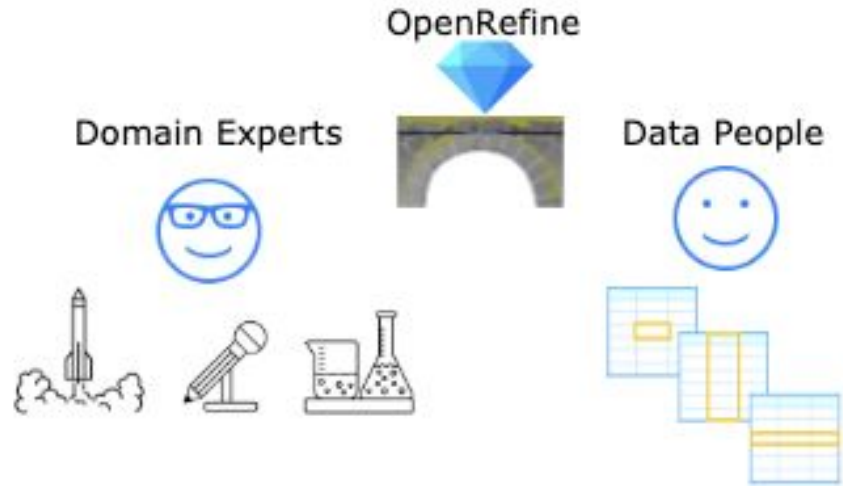
The final release of OpenRefine 3.2. Please BACKUP your workspace directory before installing and report any problems that you encounter.

The final release of 3.2 was released on July 26, 2019. A change log is provided on [the release page](#).

- **Windows kit**, Download, unzip, and double-click on *openrefine.exe*. If you're having issues with the above, try double-clicking on *refine.bat* instead.
- **Mac kit**, Download, open, drag icon into the Applications folder and double click on it.
- **Linux kit**, Download, extract, then type `./refine` to start.

What we'll cover

- First things first (software installation check)
- Web technologies and non-rectangular data types
- Wrectangling your data
- Make it reproducible
- Share your work!



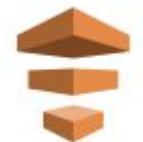


So much data work to do...

What is/where are the data?

Data are everywhere we look, and they're being used to measure nearly anything we can imagine

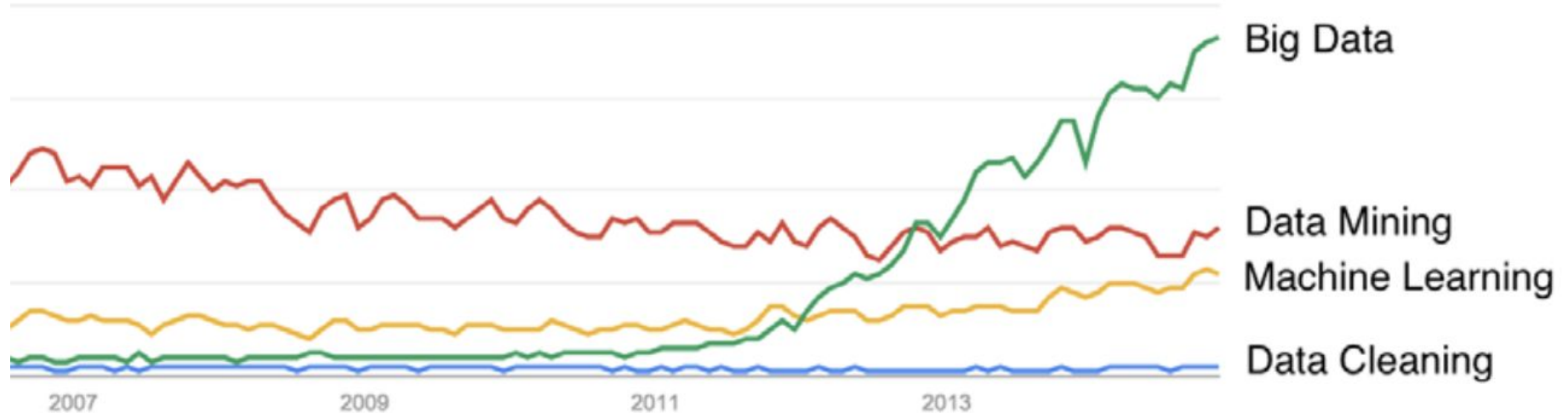
Numbers, text, pictures, audio, videos--there's no limit to what we can capture



But 'data cleaner' and 'data janitor' aren't very sexy...



Popularity of "Data" Terms on Google Trends, 2007-2014

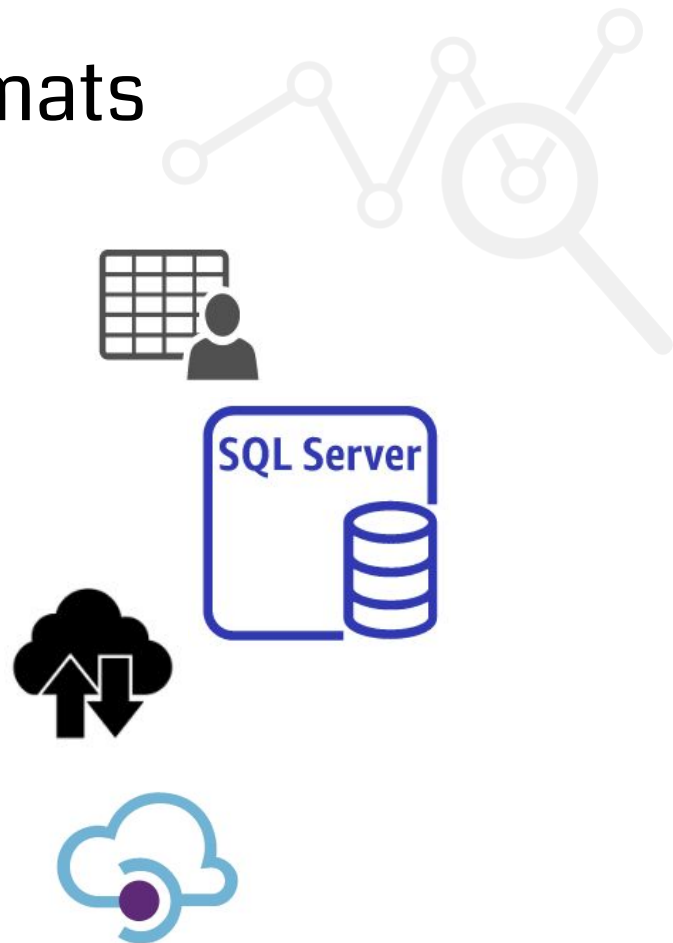




Fundamentals - Data File Types & Structures

Data come in a variety of file formats

1. Someone sends you data in a downloadable file
2. You log into an interactive front-end and retrieve the data from a storage system (i.e. a MS SQL server database system)
3. The data are available via a continuous stream that's capturing web traffic (i.e. social media)
4. You access the data through an **Application Programming Interface (API)**



Types of files: text files

Computer files generally belong to two broad groups, commonly called **text files** and **binary files**

- Files like web pages, computer source code, and open-source programming languages are all text files
- These files can be opened in a text editor (Notepad, Text Edit, etc.) or via the command line
- Humans and computers can read these files



Types of files: binary files

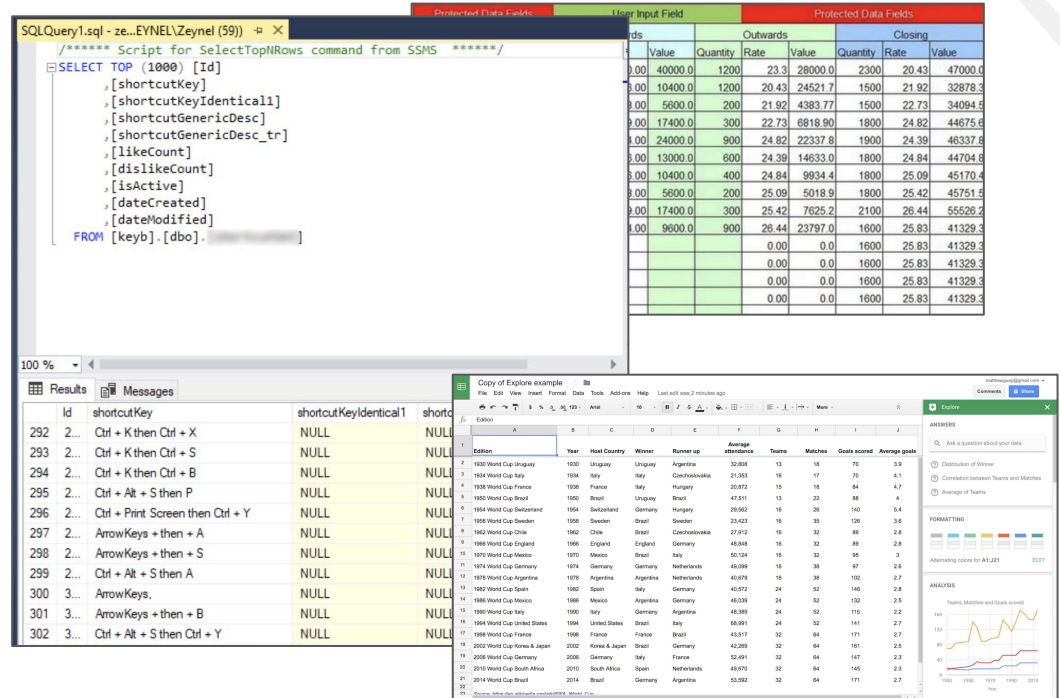
Binary files require software to open them, and are typically not human readable

- MS Word, Excel, and proprietary software files
- Executables and application installation files
(**.dmg** or **.exe**)
- Media files (**.png**, **.jpg**, **.mp4** or **.mov** files)
- Encryption or compression files (**.zip** or **.rar**)
- Humans can't read these files!!



Rectangular files (spreadsheets)

- These are files stored in columns and rows (or variables and observations)
- Rectangular files usually require spreadsheet software (Excel or OpenOffice) or relational database software



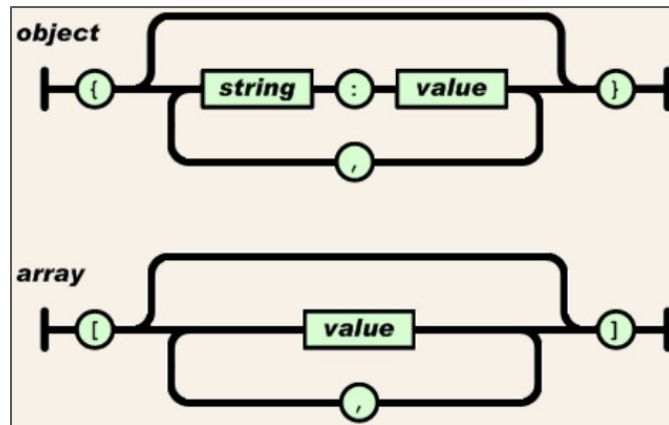
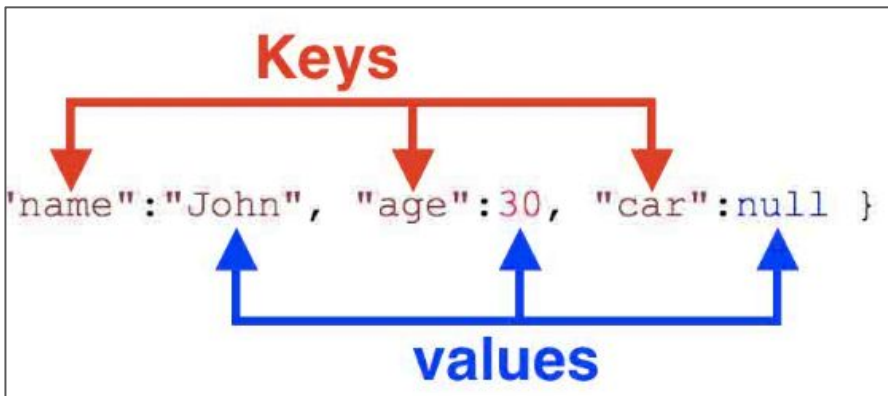
Non-rectangular files (JSON & XML)

- Non-rectangular data files are commonly used on the web for storing and transferring data
- **JSON** (JavaScript Object Notation) was created in 2002 and used for data storage and transfer
- **XML** (extensible markup language) is slightly older technology (created in 1996), but still widely used for the same purpose



JSON data objects

- JSON is an **object notation language** and stores data in **objects and arrays**
- *Why would anyone store data this way?* JSON can store data and the attributes about the data within the same object.



XML data objects

- [XML](#) (eXtensible Markup Language) is a language used to encode web documents and data structures
- XML is also useful for transmitting information between different software systems and through APIs
- XML was designed to be "self-descriptive" and carry data in a readable format

```
<?xml version="1.0"?>
<quiz>
  <qanda seq="1">
    <question>
      Who was the forty-second
      president of the U.S.A.?
    </question>
    <answer>
      William Jefferson Clinton
    </answer>
  </qanda>
  <!-- Note: We need to add
  more questions later.-->
</quiz>
```

XML

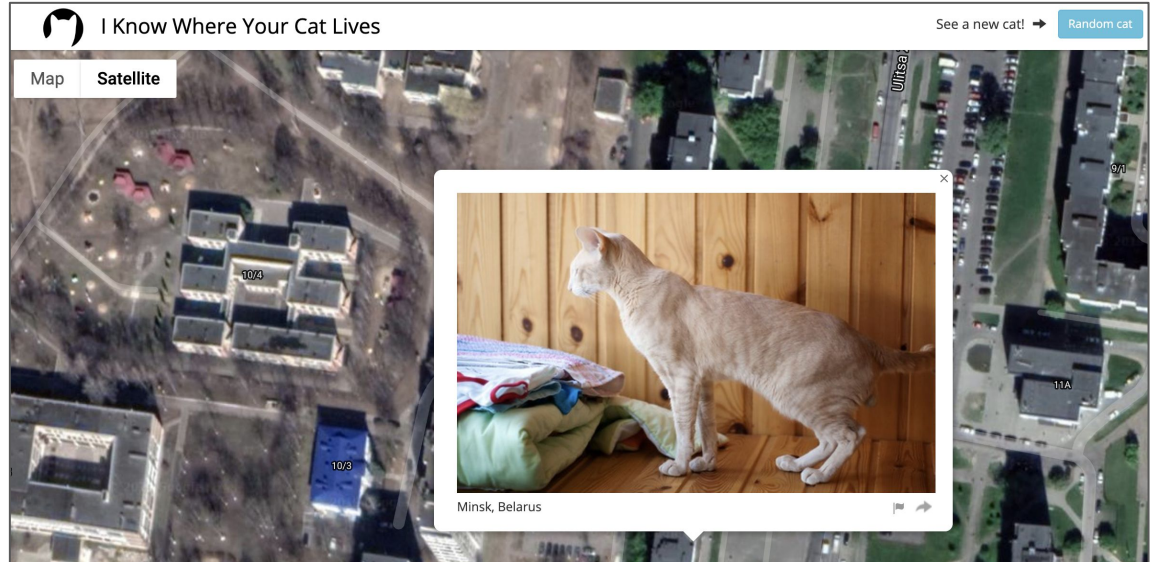


Why bother with these technologies?

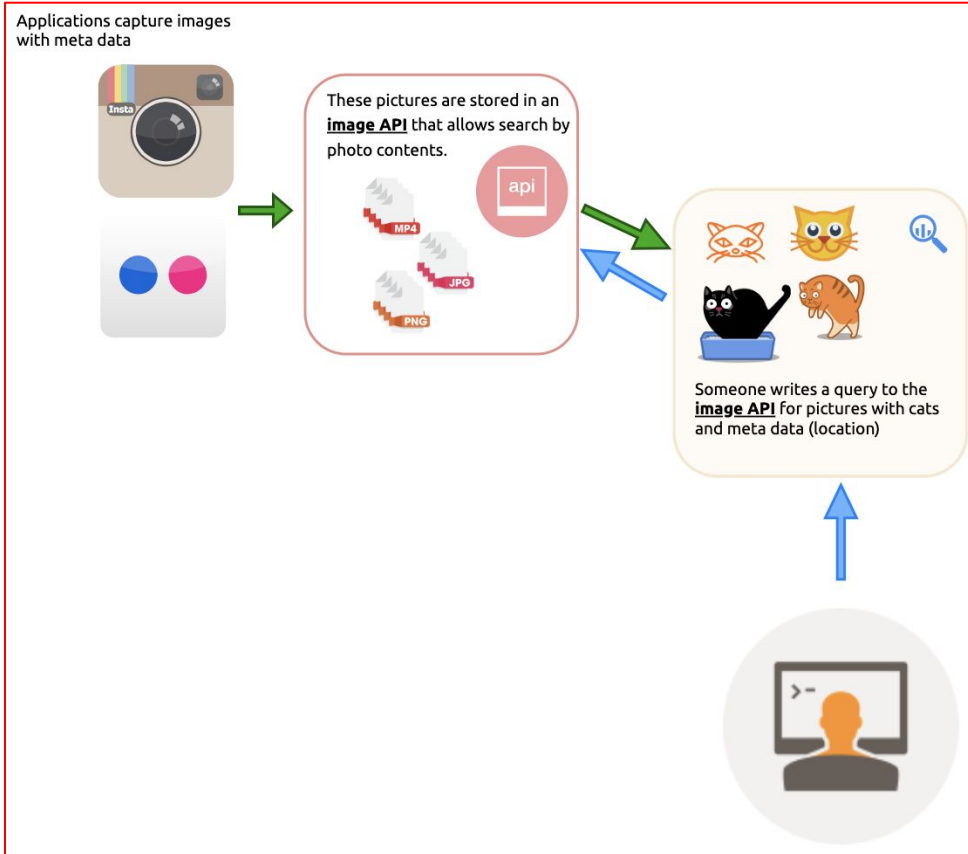
API: Application Programming Interface

An [API](#) is the set of instructions for accessing or transmitting information between software applications

JSON and **XML** are usually how data are transferred in/out of APIs



API: Application Programming Interface



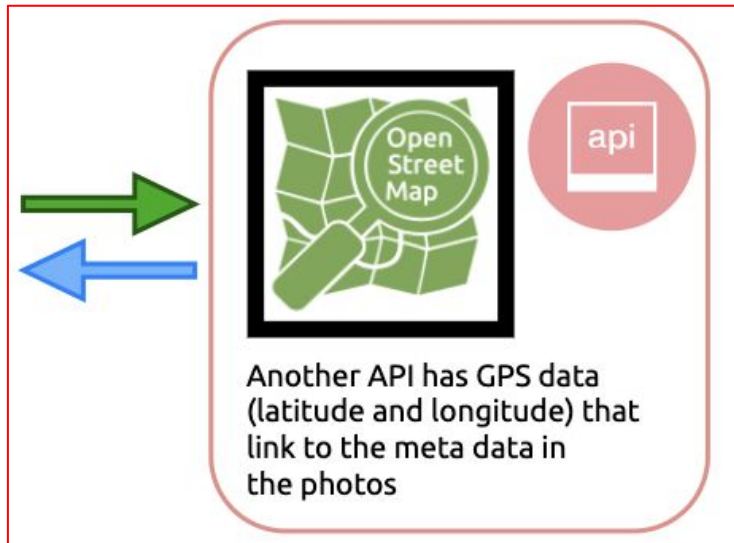
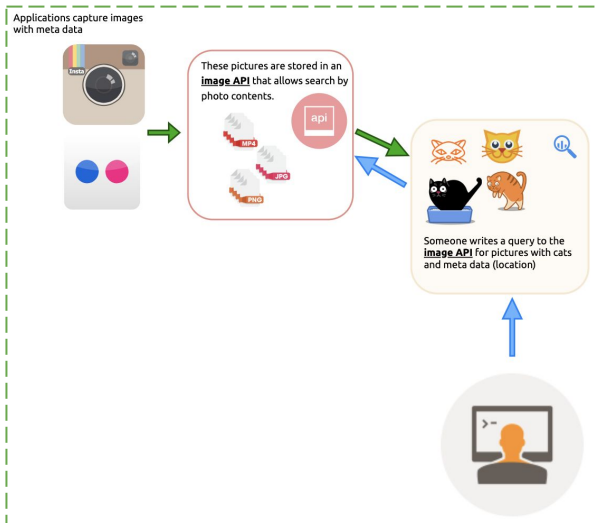
1) Photos from Flickr and Instagram get stored in an API

2) Requests are made for certain photos with GPS data

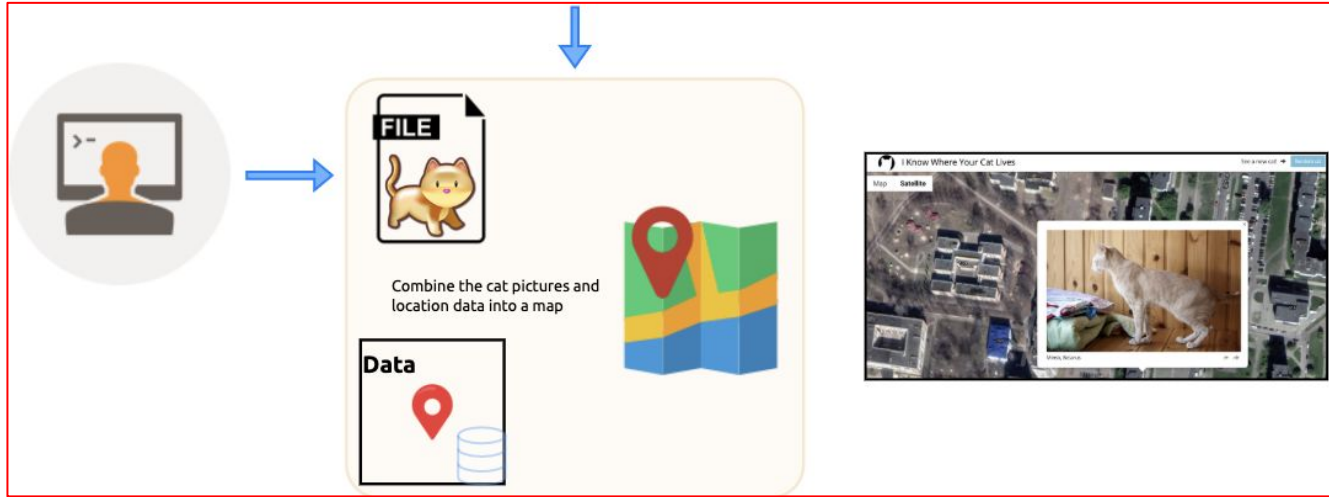
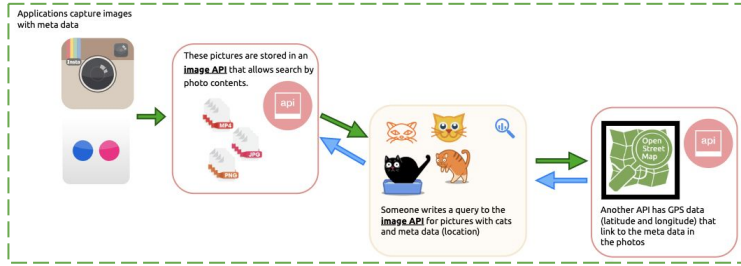
API: Application Programming Interface



3) Combine image meta data with GPS data from OpenStreetMap API



API: Application Programming Interface



4) Use Google Maps to display the image on the satellite image



A Quick Example


Load Address Data into OpenRefine



1. Navigate to this link and load the addresses into OpenRefine:

[Raw CSV address data](#)


2. Create a new project called **us-addresses**

 **OpenRefine** us-addresses [Permalink](#)

Facet / Filter

Undo / Redo 6 / 6

Using facets and filters



Use facets and filters to select subsets of your data to act on. Choose facet and filter methods from the menus at the top of each data column.

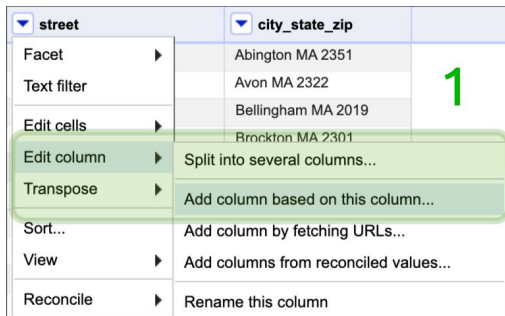
Not sure how to get started?
[Watch these screencasts](#)

234 rows

Show as: **rows** records Show: 5 10 25 50 rows

<input checked="" type="checkbox"/> All	<input checked="" type="checkbox"/> street	<input checked="" type="checkbox"/> city_state_zip	
		1.	777 Brockton Avenue
		2.	30 Memorial Drive
		3.	250 Hartford Avenue
		4.	700 Oak Street
		5.	66-4 Parkhurst Rd
		6.	591 Memorial Dr
		7.	55 Brooksby Village Way
		8.	137 Teaticket Hwy
		9.	42 Fairhaven Commons Way

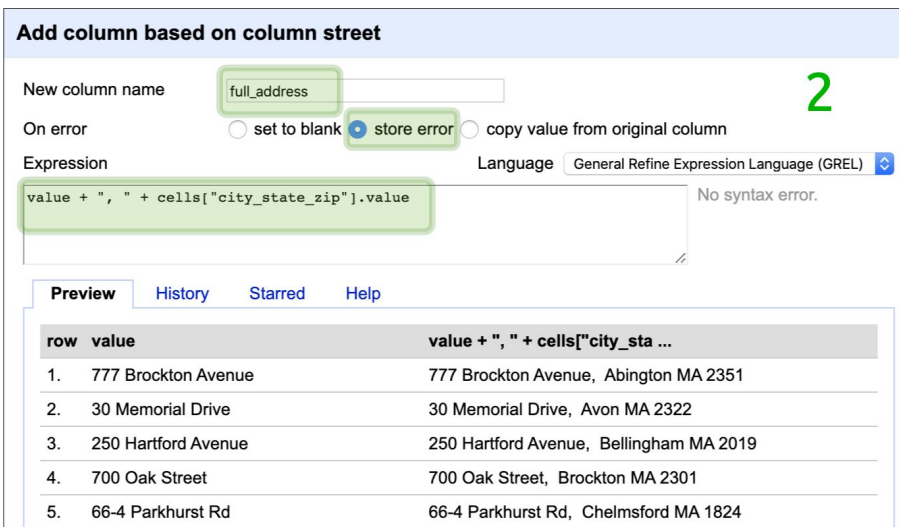
Create `full_address` column



street	city_state_zip
Abington MA 2351	
Avon MA 2322	
Bellingham MA 2019	
Brockton MA 2301	

Select the little arrow on the **street** column and click on,

***Edit column >
Add column based on this column***



Add column based on column street

New column name: `full_address`

On error: ☐ set to blank ☒ store error ☐ copy value from original column

Expression: `value + ", " + cells["city_state_zip"].value` Language: General Refine Expression Language (GREL)

No syntax error.

Preview


row	value	value + ", " + cells["city_sta ...
1.	777 Brockton Avenue	777 Brockton Avenue, Abington MA 2351
2.	30 Memorial Drive	30 Memorial Drive, Avon MA 2322
3.	250 Hartford Avenue	250 Hartford Avenue, Bellingham MA 2019
4.	700 Oak Street	700 Oak Street, Brockton MA 2301
5.	66-4 Parkhurst Rd	66-4 Parkhurst Rd, Chelmsford MA 1824

Use the following GREL code to create a new **`full_address`** column:

`value + ", " + cells["city_state_zip"].value`

Click **OK**.

Arrange columns



full_address	city_state_zip
Facet	Abington MA 2351
Text filter	Avon MA 2322
Edit cells	Bellingham MA 2019
Edit column	Brockton MA 2301
Transpose	
Sort...	
View	
Reconcile	
121 Worcester Rd, Fram	
677 Timpany Blvd, Gard	
337 Russell St, Hadley	
295 Plymouth Street, Ha	
1775 Washington St, Ha	
280 Washington Street,	

Select the little arrow on the new **full_address** column and click on,

Edit column > Move column to the end

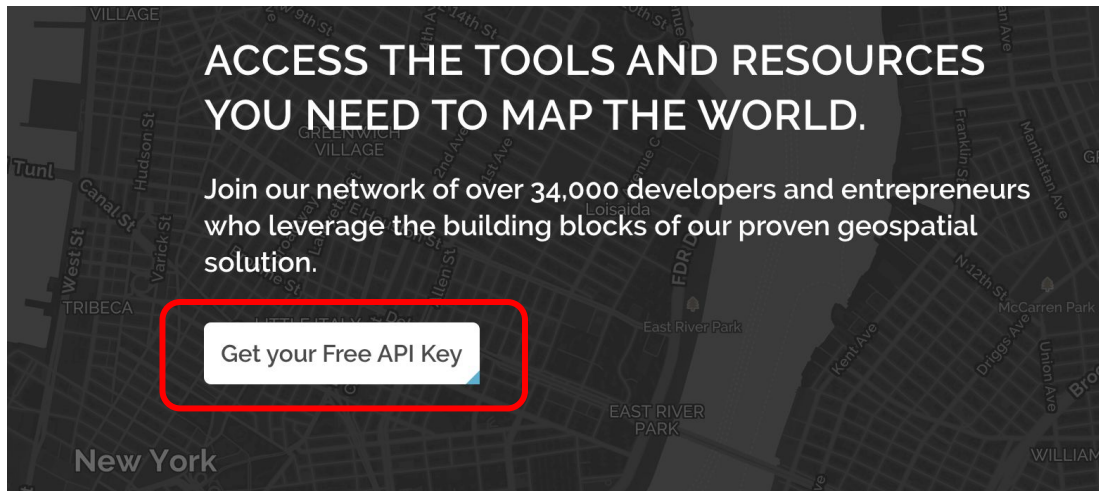
234 rows			
Show as: rows records Show: 5 10 25 50 rows			
All	street	city_state_zip	full_address
1.	777 Brockton Avenue	Abington MA 2351	777 Brockton Avenue, Abington MA 2351
2.	30 Memorial Drive	Avon MA 2322	30 Memorial Drive, Avon MA 2322
3.	250 Hartford Avenue	Bellingham MA 2019	250 Hartford Avenue, Bellingham MA 2019
4.	700 Oak Street	Brockton MA 2301	700 Oak Street, Brockton MA 2301
5.	66-4 Parkhurst Rd	Chelmsford MA 1824	66-4 Parkhurst Rd, Chelmsford MA 1824
6.	591 Memorial Dr	Chicopee MA 1020	591 Memorial Dr, Chicopee MA 1020
7.	55 Brooksby Village Way	Danvers MA 1923	55 Brooksby Village Way, Danvers MA 1923
8.	137 Teaticket Hwy	East Falmouth MA 2536	137 Teaticket Hwy, East Falmouth MA 2536
9.	42 Fairhaven Commons Way	Fairhaven MA 2719	42 Fairhaven Commons Way, Fairhaven MA 2719
10.	374 William S Canning Blvd	Fall River MA 2721	374 William S Canning Blvd, Fall River MA 2721
11.	121 Worcester Rd	Framingham MA 1701	121 Worcester Rd, Framingham MA 1701

MapQuest API

Head over to:

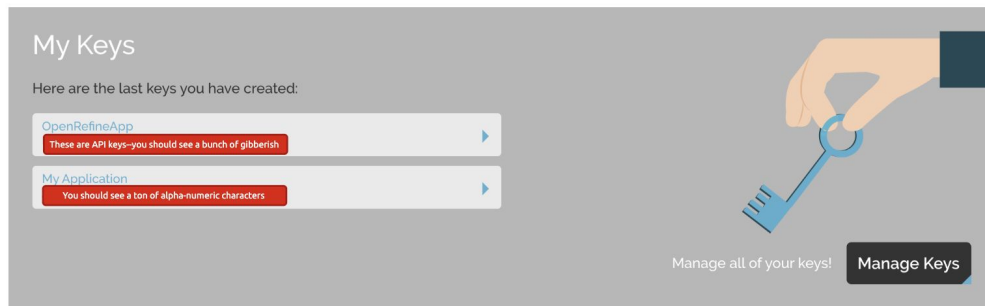
<https://developer.mapquest.com/>

Click on the button to get your **Free API Key**



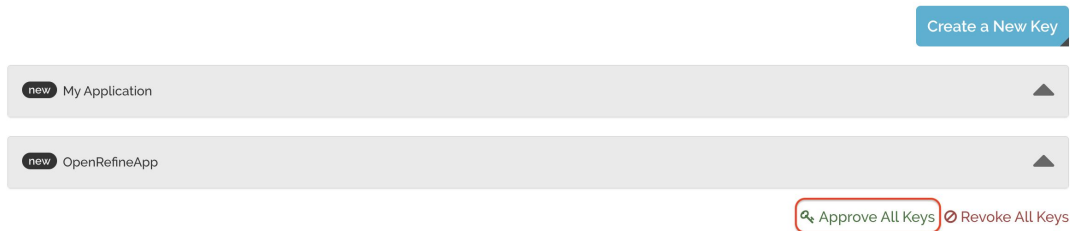
Create and Approve MapQuest API Key

1. You can create a new key or use the one listed under **My Application**



2. Click **Manage Keys** then **Approve All Keys**

▸ Manage Keys





APIs with OpenRefine

Example MapQuest API query



The domain address, proceeded by some additional information to access the server:

`open.mapquestapi.com/nominatim/v1/`

The request, written in `http` or hypertext transfer protocol:

`http://`

The specific resources on the server:

- 1) the (`search.php`) API, followed by '?'
- 2) data requests (`format=json`), followed by "&"
- 3) Our API key (`key=yourkey`), followed by "&"
- 4) The query (`q=[Addresses]`)

`http:// open.mapquestapi.com/nominatim/v1/`

`search.php?format=json&key=JWbcrLmu5UfJ9n1krjAMdr0jz3QIG0ha&q=[Addresses]`

Domain & directory

Example API query parameters with GREL

Click on the new full_address column and select, **Add column by fetching URLs**

Follow the steps in the diagram to the right, but refer to the notes below for the GREL expression:

After you've entered the **Expression**, check the 1st row in the **Preview** section at the bottom of the dialogue box.

Add column by fetching URLs based on column full_address

New column name Throttle delay milliseconds

On error ☐ set to blank ☒ store error ☒ Cache responses

HTTP headers to be used when fetching URLs: [Show](#)

Formulate the URLs to fetch:

Expression Language No syntax error.

Preview History Starred Help

row	value	'http://open.mapquestapi.com/n ...
1.	777 Brockton Avenue, Abington MA 2351	http://open.mapquestapi.com/nominatim/v1/search. format=json&key=JWbcrLmu5UfJ9n1krjAMdr0jz3QI
2.	30 Memorial Drive, Avon MA 2322	http://open.mapquestapi.com/nominatim/v1/search. format=json&key=JWbcrLmu5UfJ9n1krjAMdr0jz3QI
3.	250 Hartford Avenue, Bellingham MA 2019	http://open.mapquestapi.com/nominatim/v1/search. format=json&key=JWbcrLmu5UfJ9n1krjAMdr0jz3QI
4.	700 Oak Street, Brockton MA 2301	http://open.mapquestapi.com/nominatim/v1/search. format=json&key=JWbcrLmu5UfJ9n1krjAMdr0jz3QI

OK Cancel

Example API query parameters with GREL

This takes time!




Create column mapquest_locations at index 3 by fetching URLs based on column full_address using expression
grel:'http://open.mapquestapi.com/nominatim/v1/search.php?'+
'format=json&' +
'key=JWbcrLmu5UfJ9n1krjAMdr0jz3QIG0ha&'+ 'q='+
escape(value, "url")



78% complete **Cancel**

JSON data

When you're done, you should see the following JSON data in the new **mapquest_locations** column


 **OpenRefine** us-addresses [Permalink](#)

Open... Export ▾ Help

Facet / Filter [Undo](#) / [Redo](#) 14 / 14

234 rows Extensions: Wikidata ▾

Show as: **rows** records Show: 5 10 25 50 rows « first < previous **1 - 50** next > last »

Using facets and filters

Use facets and filters to select subsets of your data to act on. Choose facet and filter methods from the menus at the top of each data column.

Not sure how to get started?
[Watch these screencasts](#)

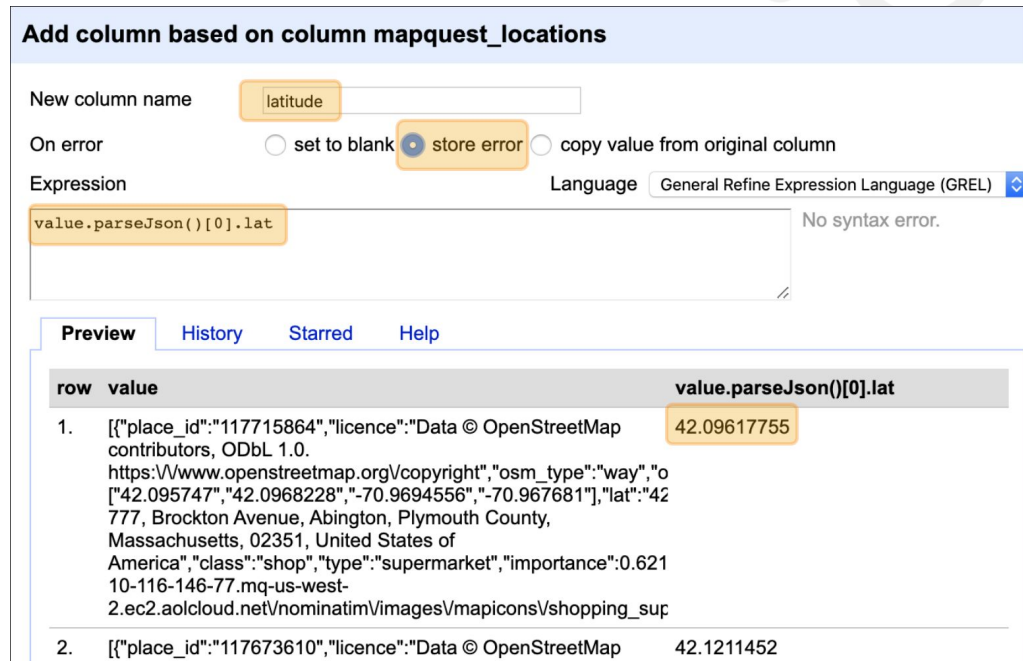
		street	city_state_zip	full_address	mapquest_locations
☆	1.	777 Brockton Avenue	Abington MA 2351	777 Brockton Avenue, Abington MA 2351	[{"place_id":"117715864","licence":"Data © OpenStreetMap contributors, ODbL 1.0. https://www.openstreetmap.org/Vcopyright","osm_type":"way","osm_id":"196033306","boundingbox":["42.095747","42.0968228","-70.9694556","-70.967681"],"lat":"42.09617755","lon":"-70.9685309348312","display_name":"Walmart, 777, Brockton Avenue, Abington, Plymouth County, Massachusetts, 02351, United States of America","class":"shop","type":"supermarket","importance":0.621,"icon":"http://ip-10-116-146-77.mq-us-west-2.ec2.aolcloud.net/nominatim/Vimages/Vmapicons/Vshopping_supermarket.p.20.png"}]
☆	2.	30 Memorial Drive	Avon MA 2322	30 Memorial Drive, Avon MA 2322	[{"place_id":"117673610","licence":"Data © OpenStreetMap contributors, ODbL 1.0. https://www.openstreetmap.org/Vcopyright","osm_type":"way","osm_id":"196505139","boundingbox":["42.1205242","42.1218355","-71.0308302","-71.0293731"],"lat":"42.1211452","lon":"-71.0300878730199","display_name":"Walmart Supercenter, 30, Memorial Drive, Avon, Norfolk County, Massachusetts, 02322, United States of America","class":"shop","type":"supermarket","importance":0.621,"icon":"http://ip-10-116-136-130.mq-us-west-2.ec2.aolcloud.net/nominatim/Vimages/Vmapicons/Vshopping_supermarket.p.20.png"}]

Parsing JSON data in OpenRefine

Click on the new `mapquest_locations` column and select **Edit column > Add column based on this column...**

In the dialogue box, enter the following options:

1. **Name new column** = `latitude`
2. **On error** = `store error`
3. **Expression** = `value.parseJson()[0].lat`
3. Verify this is correct in the **Preview** pane



Add column based on column mapquest_locations

New column name

On error ☐ set to blank ☒ store error ☐ copy value from original column

Expression Language No syntax error.

Preview History Starred Help

row	value	value.parseJson()[0].lat
1.	[{"place_id": "117715864", "licence": "Data © OpenStreetMap contributors, ODbL 1.0.", "url": "https://www.openstreetmap.org/copyright", "osm_type": "way", "lat": "42.095747", "lon": "-70.968228", "addr": "777, Brockton Avenue, Abington, Plymouth County, Massachusetts, 02351, United States of America", "class": "shop", "type": "supermarket", "importance": 0.621}, {"place_id": "117673610", "licence": "Data © OpenStreetMap contributors, ODbL 1.0.", "url": "https://www.openstreetmap.org/copyright", "osm_type": "way", "lat": "42.1211452", "lon": "-70.967681", "addr": "10-116-146-77.mq-us-west-2.ec2.amazonaws.com", "class": "shop", "type": "supermarket", "importance": 0.621}]	42.09617755
2.	[{"place_id": "117673610", "licence": "Data © OpenStreetMap contributors, ODbL 1.0.", "url": "https://www.openstreetmap.org/copyright", "osm_type": "way", "lat": "42.1211452", "lon": "-70.967681", "addr": "10-116-146-77.mq-us-west-2.ec2.amazonaws.com", "class": "shop", "type": "supermarket", "importance": 0.621}]	42.1211452

Parsing JSON data in OpenRefine

Add column based on column `mapquest_locations`

New column name

On error ☐ set to blank ☒ store error ☐ copy value from original column


Expression Language

No syntax error.

Preview History Starred Help

row	value	value.parseJson()[0].lon
1.	[{"place_id": "117715864", "licence": "Data © OpenStreetMap contributors, ODbL 1.0. https://www.openstreetmap.org/copyright", "osm_type": "way", "coordinates": "[42.095747, 42.0968228, -70.9694556, -70.967681]", "lat": 42.0968228, "lon": -70.967681, "display_name": "777, Brockton Avenue, Abington, Plymouth County, Massachusetts, 02351, United States of America", "class": "shop", "type": "supermarket", "importance": 0.62}, {"place_id": "117673610", "licence": "Data © OpenStreetMap contributors, ODbL 1.0. https://www.openstreetmap.org/copyright", "osm_type": "way", "coordinates": "[42.095747, 42.0968228, -70.9694556, -70.967681]", "lat": 42.0968228, "lon": -70.967681, "display_name": "10-116-146-77.mq-us-west-2.ec2.aolcloud.net/nominatim/images/mapicons/shopping_supermarket.png"}]	-70.9685309348312
2.	[{"place_id": "117673610", "licence": "Data © OpenStreetMap contributors, ODbL 1.0. https://www.openstreetmap.org/copyright", "osm_type": "way", "coordinates": "[42.095747, 42.0968228, -70.9694556, -70.967681]", "lat": 42.0968228, "lon": -71.0300878730199, "display_name": "10-116-146-77.mq-us-west-2.ec2.aolcloud.net/nominatim/images/mapicons/shopping_supermarket.png"}]	-71.0300878730199

OK Cancel



Repeat the same steps on the **mapquest_locations** column, but this time create a **longitude** column

Check out these links to learn more!

1. John R Little has a great [Pragmatic Datafication workshop](#) with some great materials ([slides](#) and [workbook](#))
2. Check out the OpenRefine [GREL documentation on Github](#) (lots of great tidbits in here!)
3. Good 'ol fashioned YouTube! Check out the following video to learn more:
 - [Data Journalism - Cleaning Data in Workbench and OpenRefine](#)
 - [OpenRefine Beginners Tutorial](#)
 - [Clean Your Data: Getting Started with OpenRefine](#)

