

Introduction to MongoDB

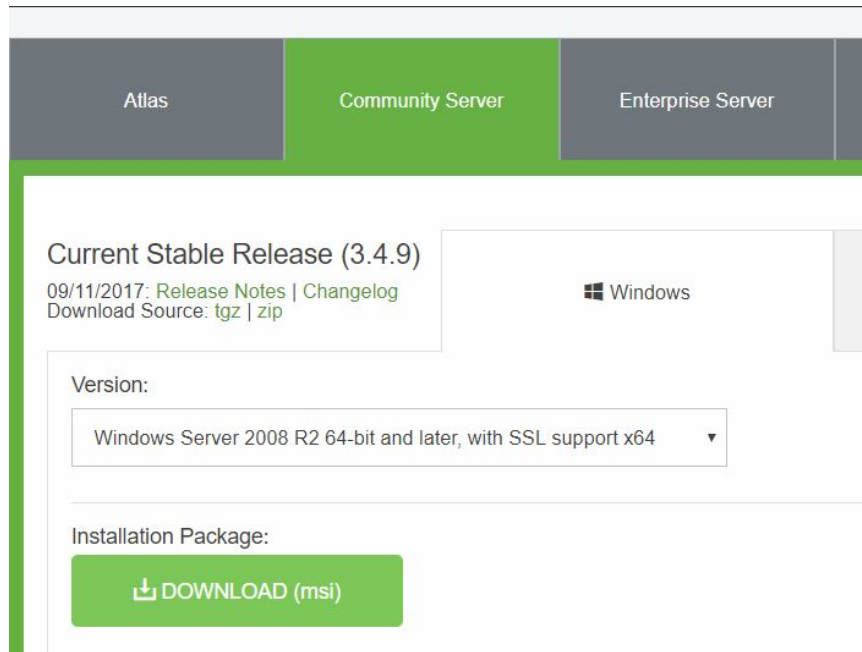
26/03/2018

What is MongoDB

- It's a free NoSQL document oriented database.
 - Not only SQL, since SQL queries are supported.
- It uses JSON-like documents to save data.
 - Flexible schemas - Allows missing fields.
- High performance, high availability, and automatic scaling.
 - Used by large companies like Amazon.

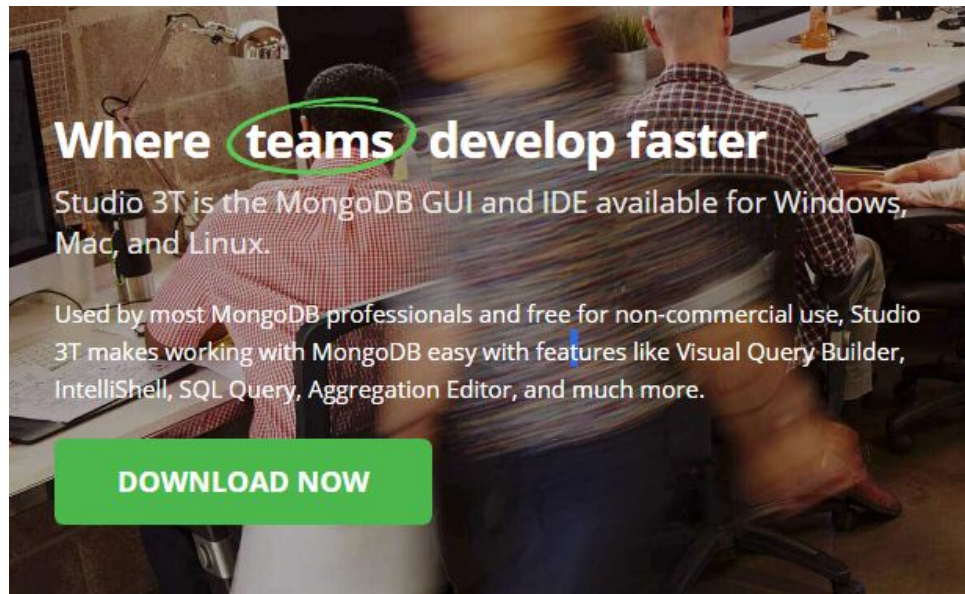
Installation

MongoDB community server



The screenshot shows the MongoDB download page for the Community Server. At the top, there are three tabs: 'Atlas', 'Community Server' (which is selected and highlighted in green), and 'Enterprise Server'. Below the tabs, the 'Current Stable Release (3.4.9)' is displayed, along with the date '09/11/2017' and links for 'Release Notes' and 'Changelog'. The 'Download Source' is listed as 'tgz' or 'zip'. A Windows logo icon is visible next to the 'Windows' operating system selection. Below this, a 'Version:' dropdown menu is set to 'Windows Server 2008 R2 64-bit and later, with SSL support x64'. At the bottom, under 'Installation Package:', there is a green button with a download icon and the text 'DOWNLOAD (msi)'.

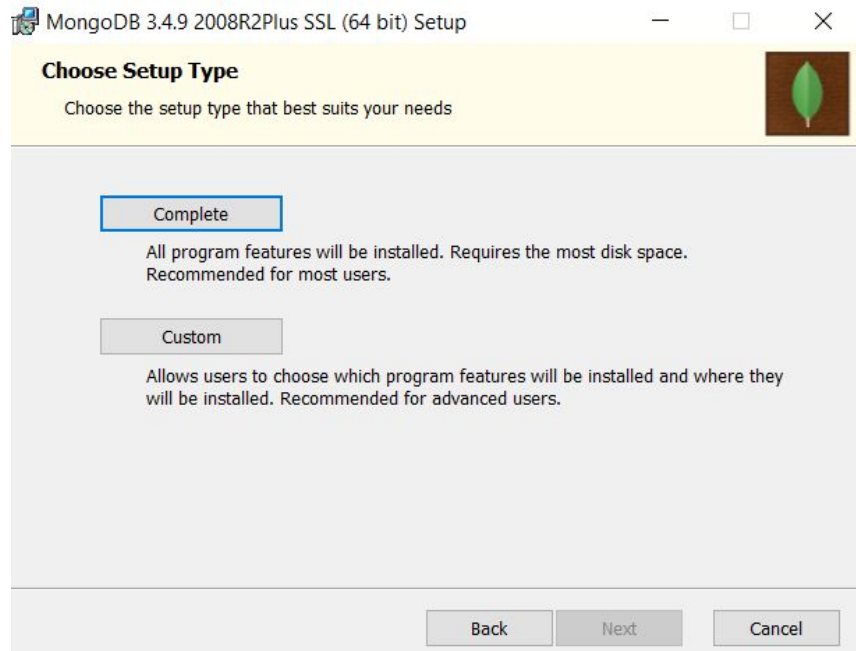
Studio 3T



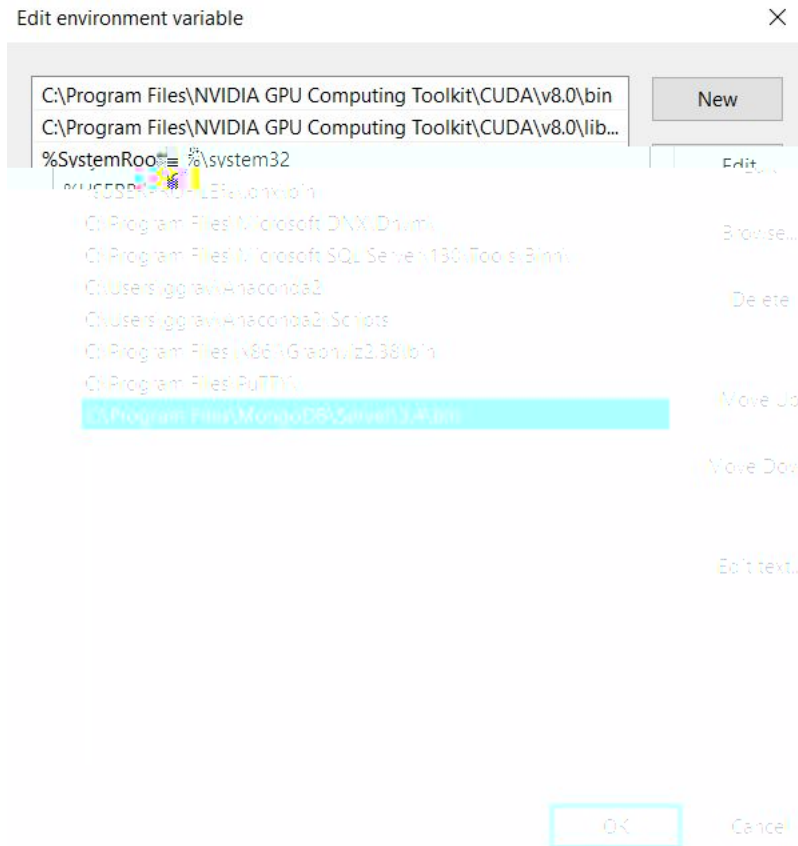
The advertisement for Studio 3T features a background image of two people working at a desk with multiple monitors. Overlaid on the image is the text 'Where teams develop faster', with the word 'teams' circled in green. Below this, it states 'Studio 3T is the MongoDB GUI and IDE available for Windows, Mac, and Linux.' Further down, it mentions 'Used by most MongoDB professionals and free for non-commercial use, Studio 3T makes working with MongoDB easy with features like Visual Query Builder, IntelliShell, SQL Query, Aggregation Editor, and much more.' At the bottom, there is a prominent green button with the text 'DOWNLOAD NOW'.

Installation

Step 1



Step 2



What is a document

- A sum of key / value pairs

```
{  
  name: "sue",  
  age: 26,  
  status: "A",  
  groups: [ "news", "sports" ]  
}
```



What is a collection



Operations

- Run / Connect to mongo

- Import documents

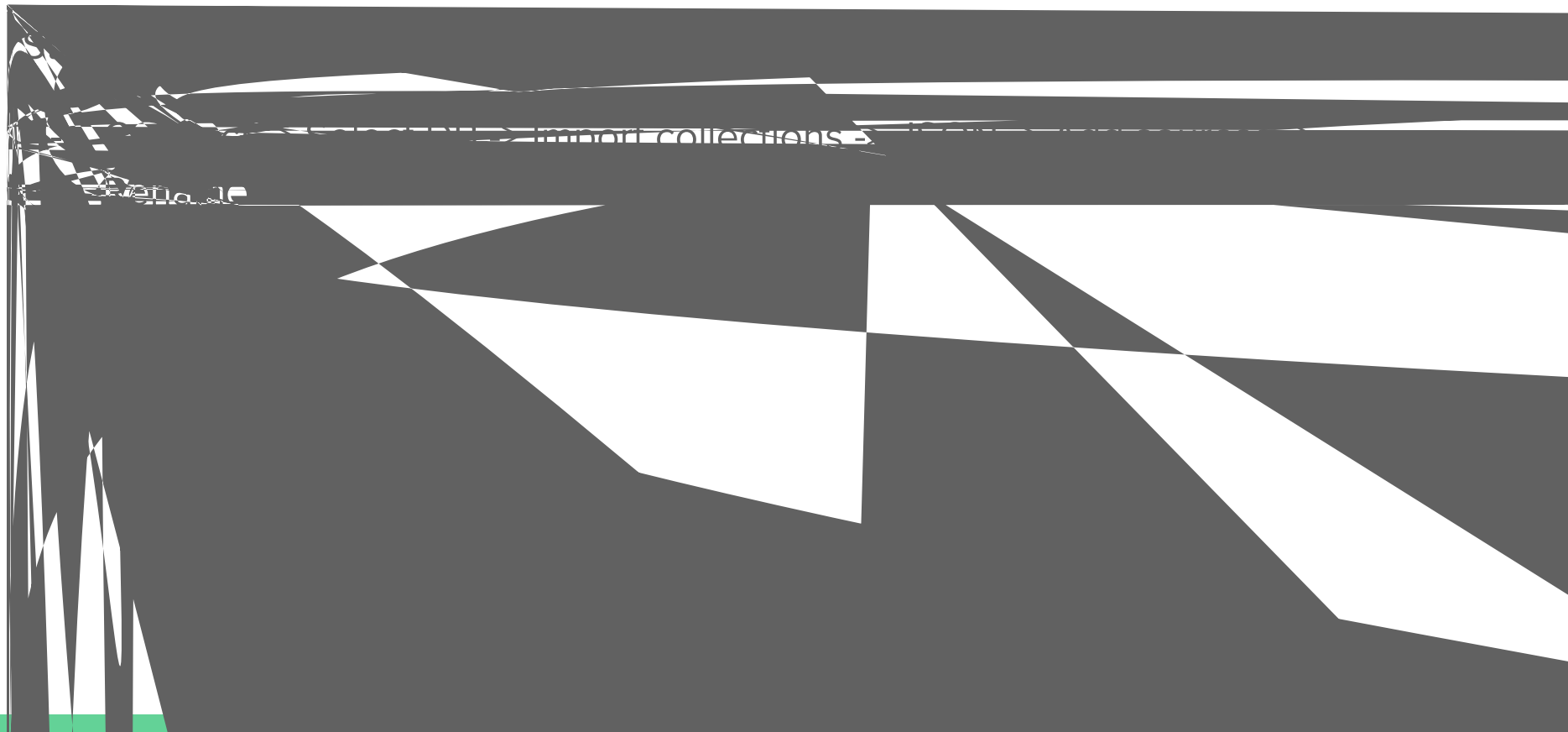
- Insert documents

- C

Connection



Import dataset



Insert documents

Python:

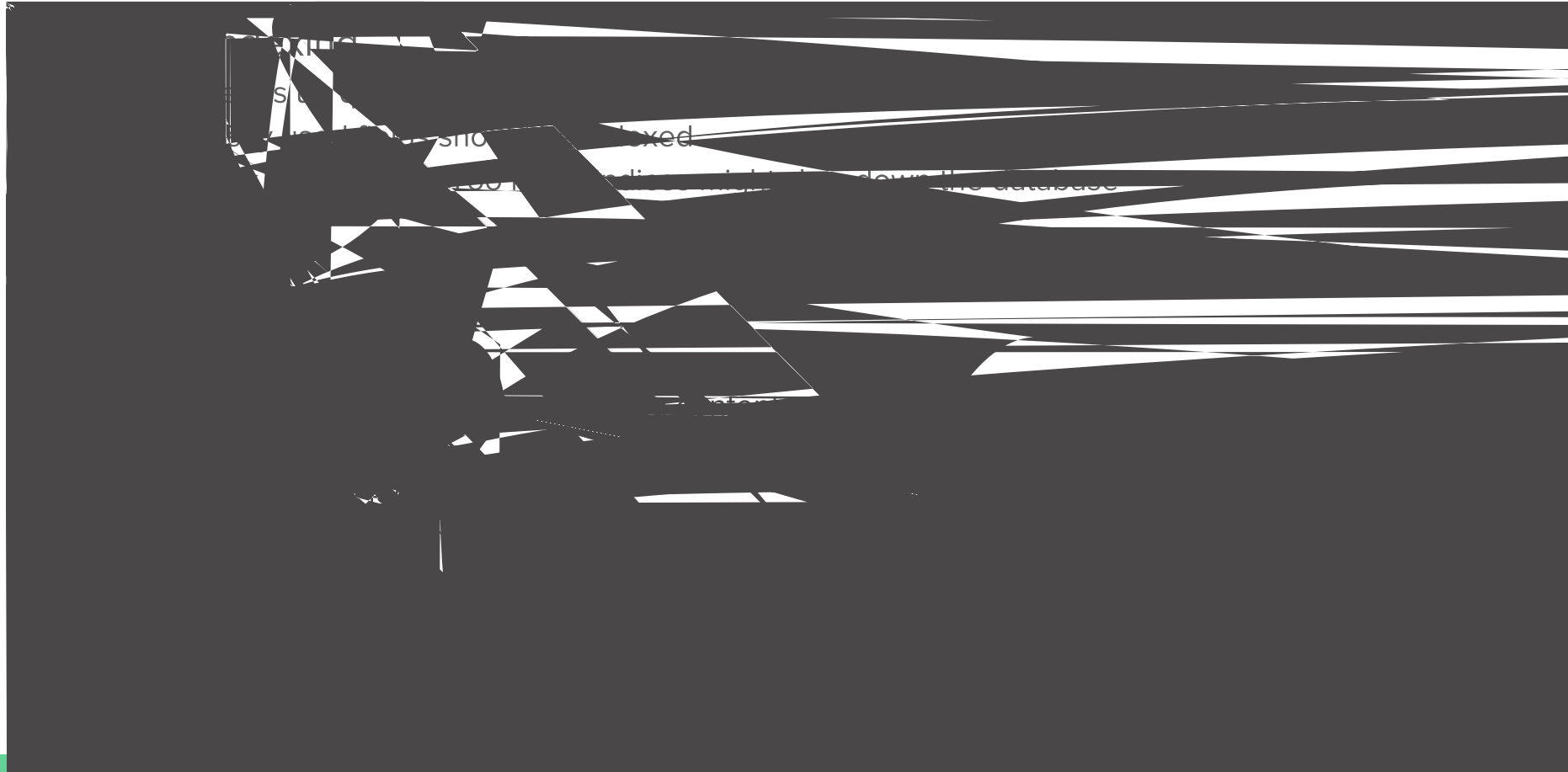
```
38     # Find the coordinates for each restaurant and
39     # save them to an external collection
40     all_restaurants = find_all_restaurants()
41     for restaurant in all_restaurants:
42         json_obj = {
43             'name': restaurant['name'],
44             'business_id': restaurant['business_id'],
45             'longitude': restaurant['longitude'],
46             'latitude': restaurant['latitude']
47         }
48         insert_to_db(json_obj, 'restaurants_coordinates')
```


Data lookups - For quick data checking

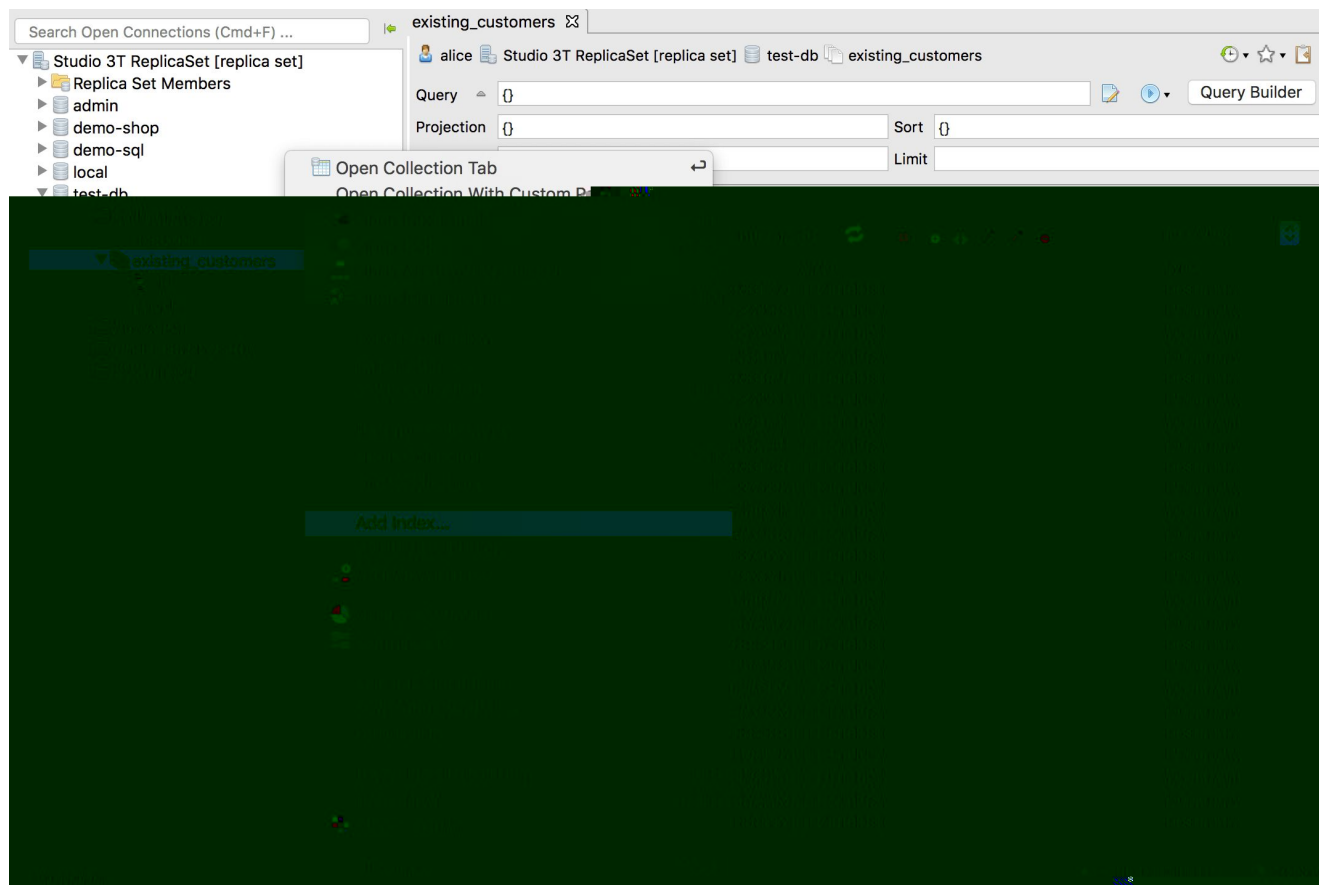
"this lecture isn't mandatory,
however it will offer you valuable
informa-"



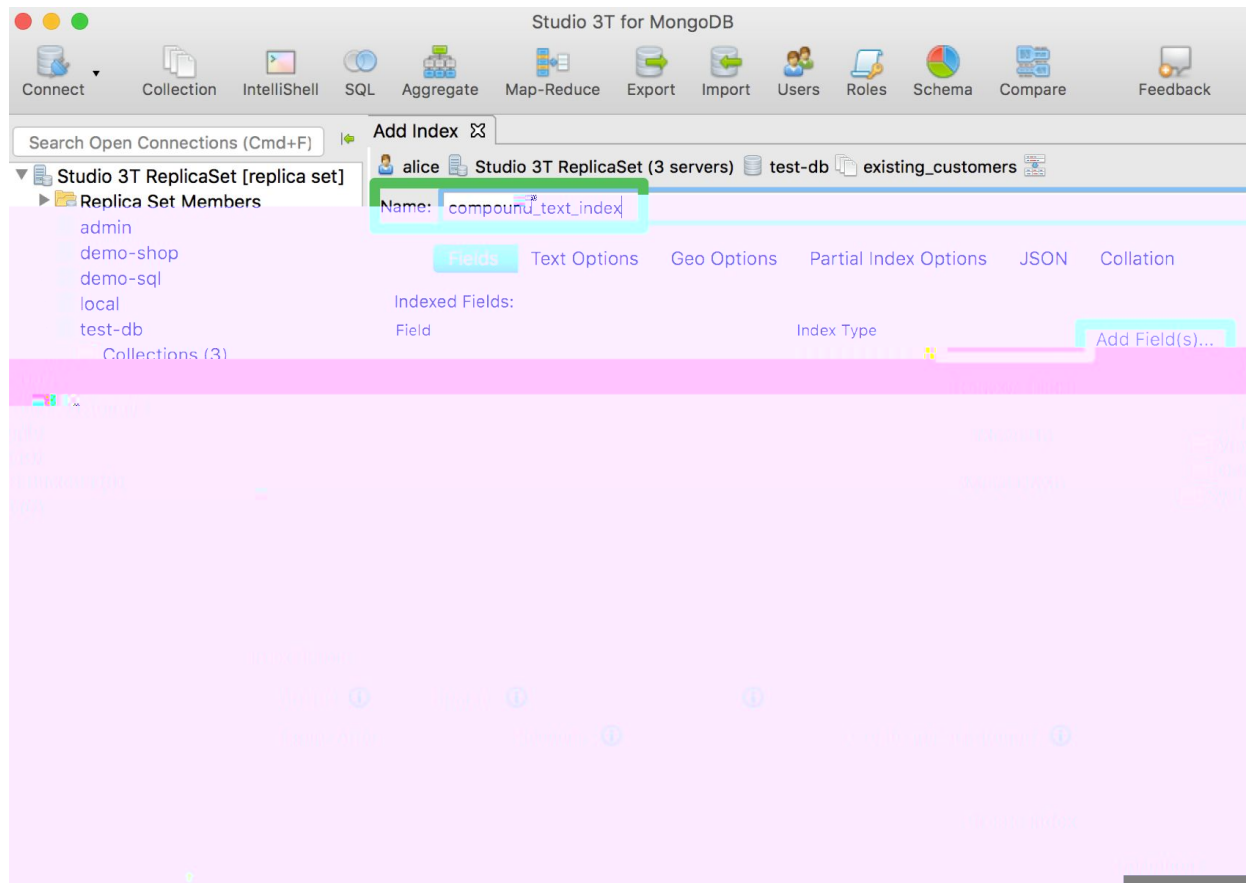
Indexing



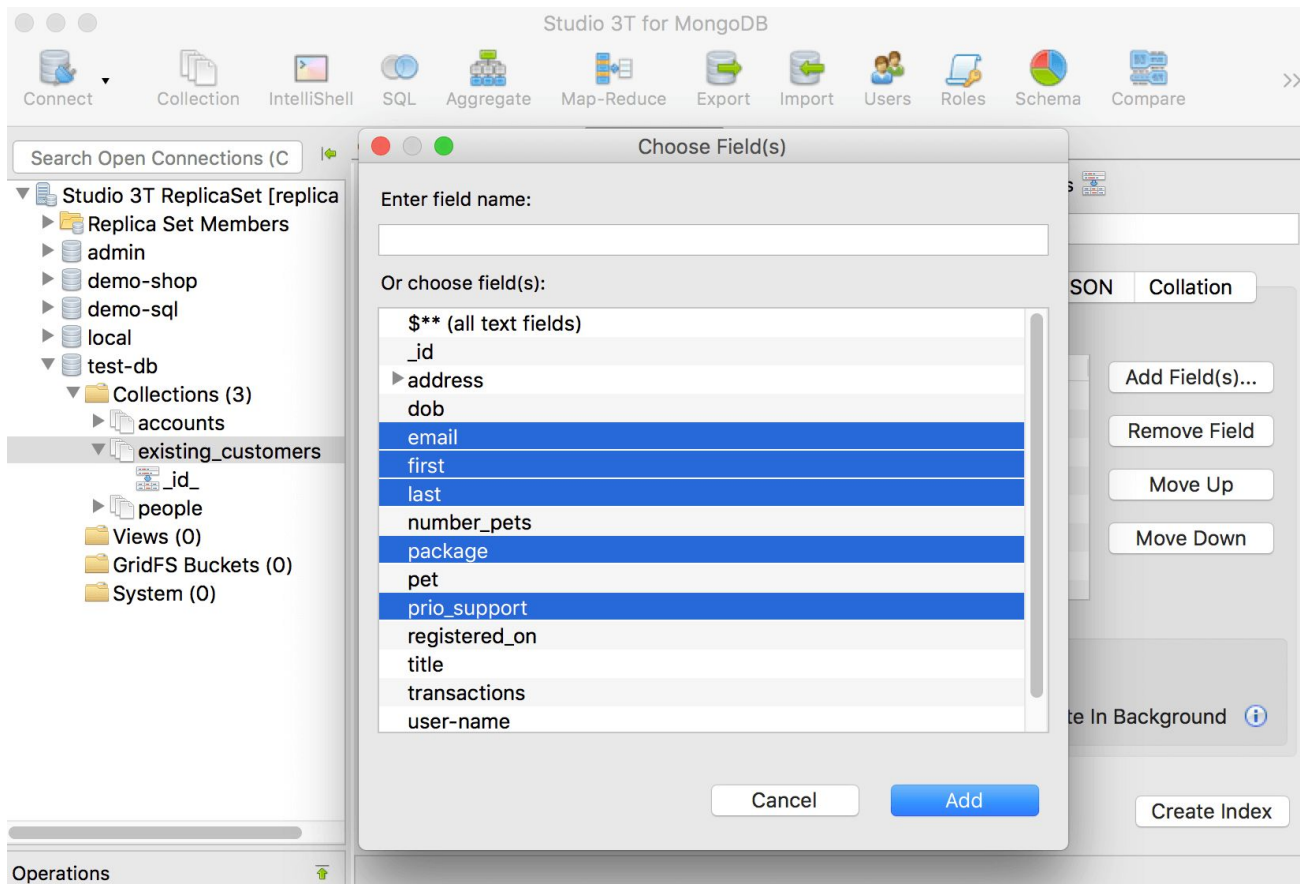
How to - Text indexing



How to - Text indexing



How to - Text indexing



How to - Text indexing

The screenshot shows the Studio 3T for MongoDB interface. The left sidebar displays the database structure: Studio 3T ReplicaSet [replica set] > test-db > Collections (3) > existing_customers. The main window is the 'Add Index' dialog for the 'existing_customers' collection in the 'test-db' database. The index name is 'compound_text_index'. The 'Indexed Fields' table lists the fields to be indexed:

Field	Index Type
email	text
first	ascending
last	descending
package	hashed
prio_support	text
	2dsphere
	2d
	geoHaystack

A dropdown menu is open for the 'Index Type' of the 'first' field, showing options: ascending, descending, hashed, text (highlighted), 2dsphere, 2d, and geoHaystack. To the right of the table are buttons: 'Add Field(s)...', 'Remove Field', 'Move Up', and 'Move Down'. Below the table, there are sections for 'Index Options' (including 'Sparse' and 'Expire After') and a 'Create Index' button at the bottom right.

How to - Text indexing

Studio 3T for MongoDB

Connect Collection IntelliShell SQL Aggregate Map-Reduce Export Import Users Roles Schema Compare Feedback

Search Open Connections (Cmd+F) Add Index

alice Studio 3T ReplicaSet (3 servers) test-db existing_customers

Name: compound_text_index

Fields Text Options Geo Options Partial Index Options JSON Collation

Indexed Fields:

Field	Index Type
email	text
first	text
last	text
package	text
prio_support	text

Add Field(s)... Remove Field Move Up Move Down

Index Options

☐ Unique ☐ Sparse ☐ Drop Duplicates

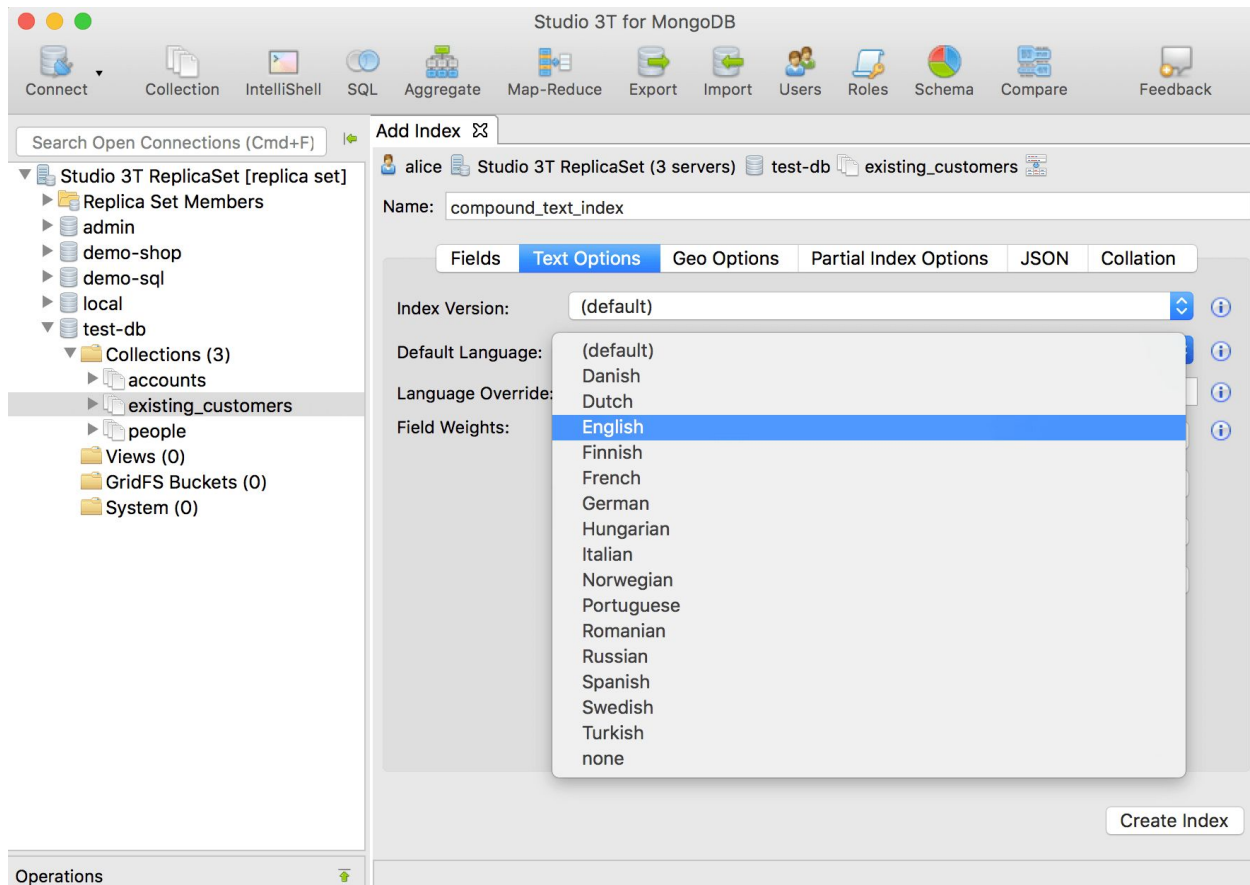
☐ Expire After Seconds

☒ Create In Background

Create Index

Operations

How to - Text indexing

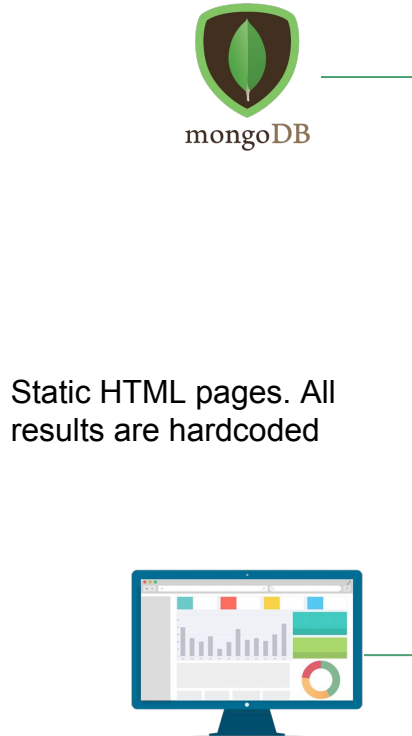


Querying with text indices

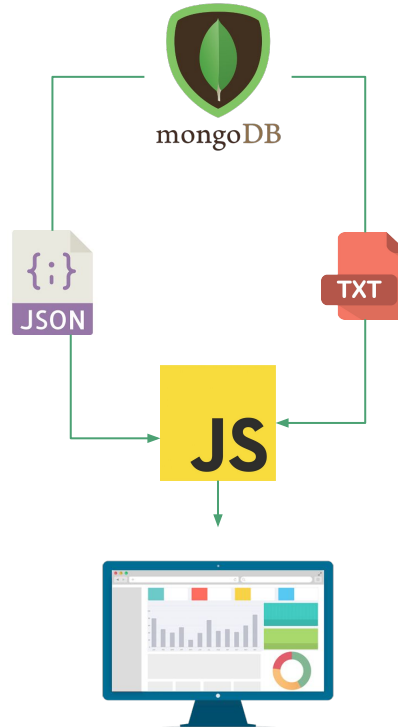
- `db.stores.find({ $text: { $search: "java coffee shop" } })`
 - `$text` tokenizes the search string and performs a logical OR
 - Will search on all indexed fields
 - Results include a relevance score for each record
- `db.stores.find({ $text: { $search: "java \"coffee shop\"" } })`
 - Will match exact phrases “java” OR “coffee shop”
-

Web app development

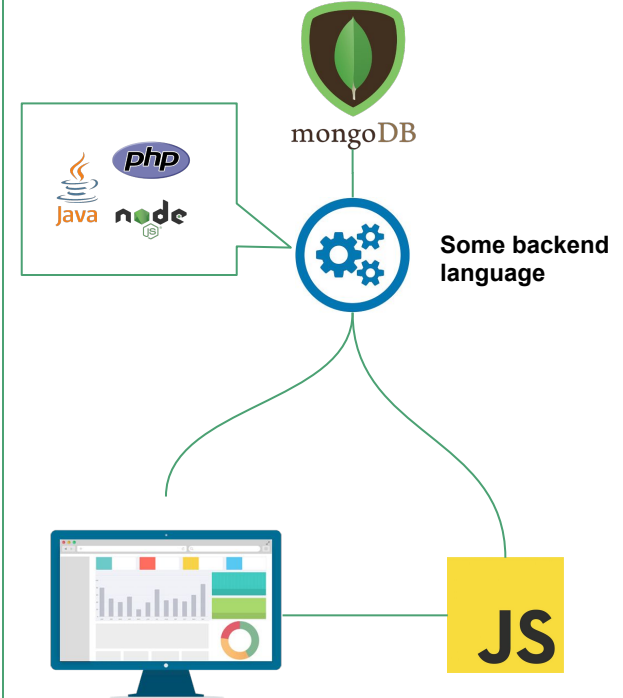
Possibility #1



Possibility #2



Possibility #3





A python-based microframework, suitable for small-scale applications.

How to create a simple Flask App:

app.py

```
from flask import Flask, render_template

app = Flask(__name__, template_folder='views')

@app.route("/")
def home():
    uni_name = "Aristotle University of Thessaloniki"
    return render_template('some_html_file.html', uni_name=uni_name)

if __name__ == "__main__":
    app.run(debug=True, host='127.0.0.1', port=5110)
```


some_html_file.html

```
<!doctype html>
<html>
  <head>
    <meta charset="utf-8">
  </head>
  <body>
    <p>Welcome to your first Flask application!</p>
    <!-- Variable is passed, using the jinja2 templating engine -->
    <h2>{{ uni_name }}</h2>
  </body>
</html>
```



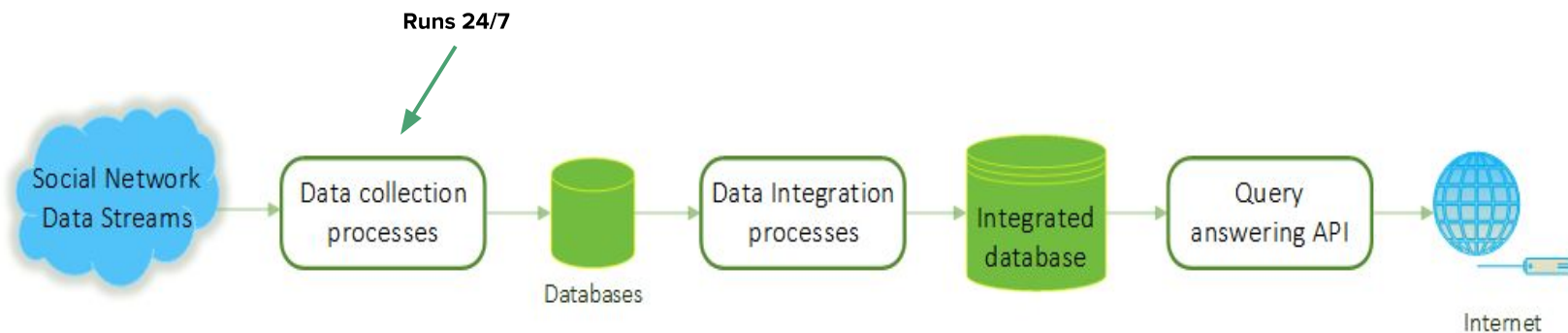
Welcome to your first Flask application!

Aristotle University of Thessaloniki

Diligent - A Social Media Data Integration platform

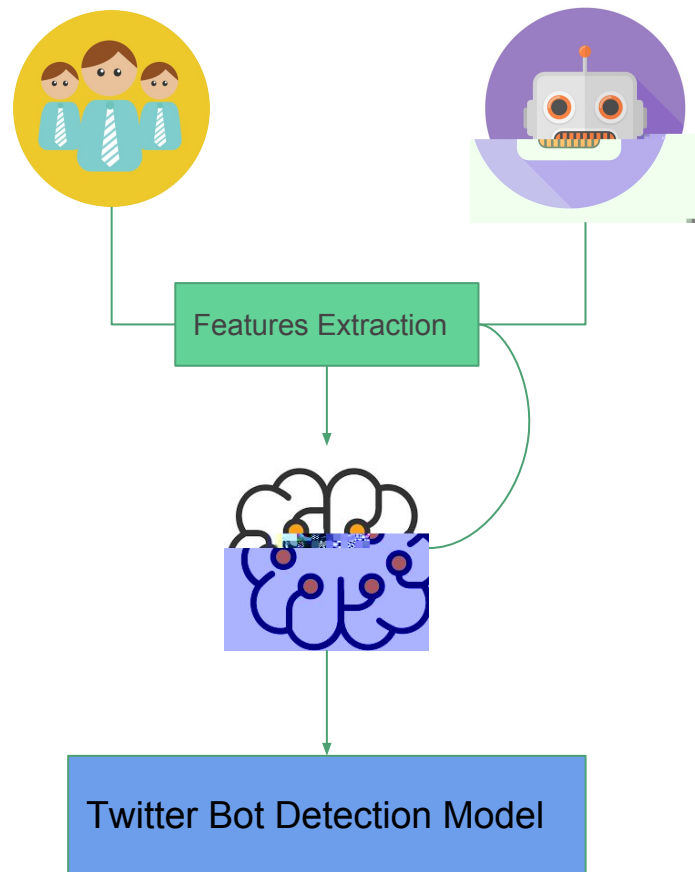
- Social media data explosion - Information Era
 - All data and metadata are saved into databases
 - Including private information (fingerprint, face, political views, etc.)
- Various data sources can be Integrated
 - Extract valuable insights
 - Personalized advertising
 - Sentiment analysis
 - -> Build a complete image - Increase data value \$\$
- Machine Learning opportunities
 - Integrated data as input

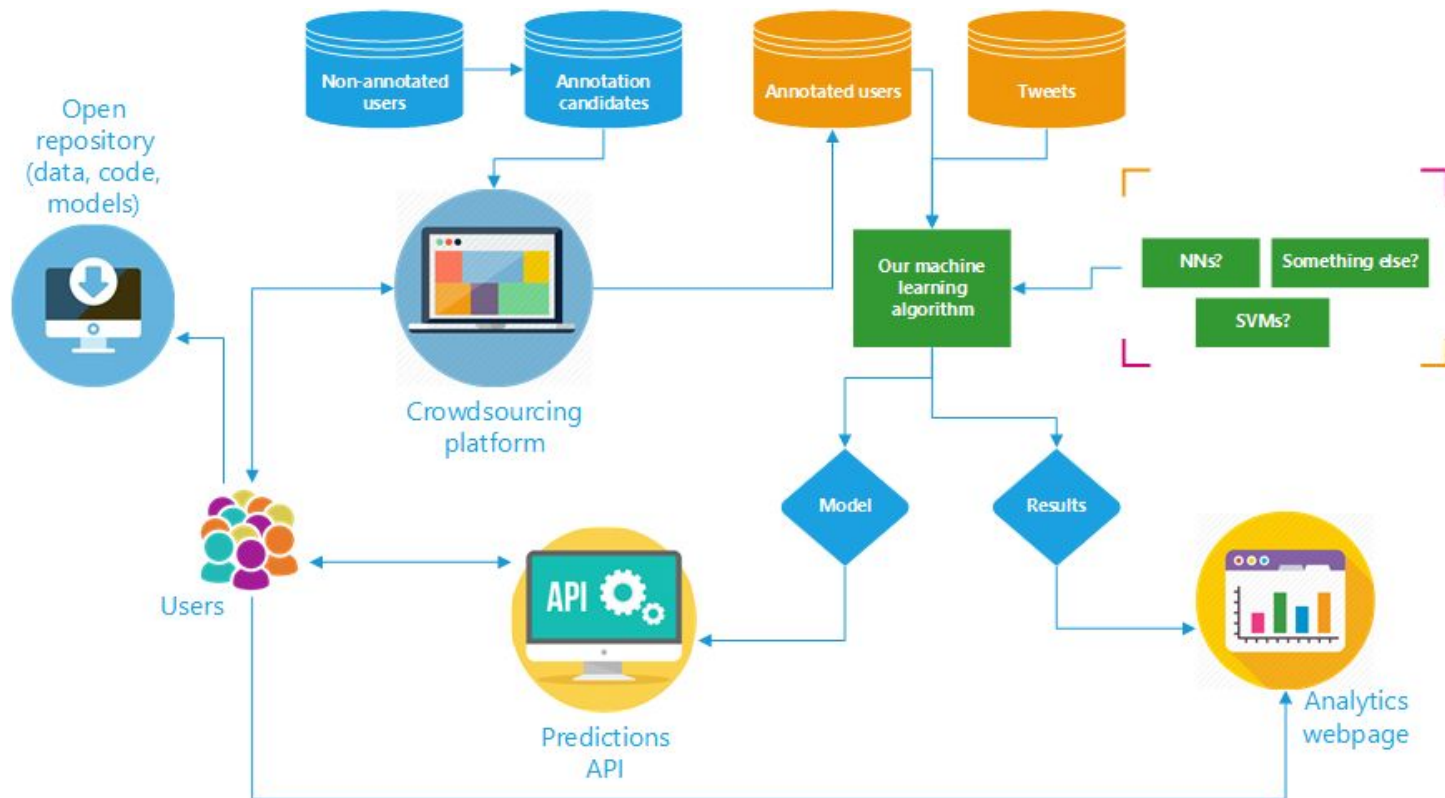
Diligent - System pipeline



Twitter bots detection

- Own a set of pre-annotated users as bots or humans
- Extract features from each user (focus on sentiment)
 - E.g. “num of tweets”, “tweets entropy”, “average sentiment polarity”
- Feed a machine learning algorithm all the features vectors
- Evaluate and extract model





Resources

- [Dataset jsons](#)
- [Github link](#)
- [MongoDB](#)
- [Flask](#)
- [Flask-RESTful](#)
- [Python](#)