

# Red Hat Technical Assessment | MODE - Data Engineer

---

*This is meant to assess a number of things for potential candidates:*

- *Comfort with the programming language - Python*
- *Comfort with potentially troubleshooting problem when implementing a technology - Setting up MySQL database locally*
- *Proficiency with documentation*

*There are some requirements for the tasks:*

- *You cannot use the Pandas library to pull, push, or transform data.*
    - *We recommend using the Requests library to perform API requests.*
  - *Use Python3.x not Python2.x*
  - *You will not be allowed to use the Python wrapper for the xkcd API.*
  - *You should use MySQL for your database.*
  - *SQL statements will be limited to select or insert or create.*
  - *Data transformations should take place in Python during runtime.*
  - *You should document all work so that a user can easily setup your project and verify your results.*
  - *Your project needs to be hosted on Github*
- 

For the tasks below you will be using the **xkcd API** - <https://xkcd.com/>

xkcd API which allows you to fetch various XKCD comics and their metadata.

If you want to fetch comics and metadata automatically, you can use the JSON interface

The URL

<https://xkcd.com/info.0.json> (current comic)

or

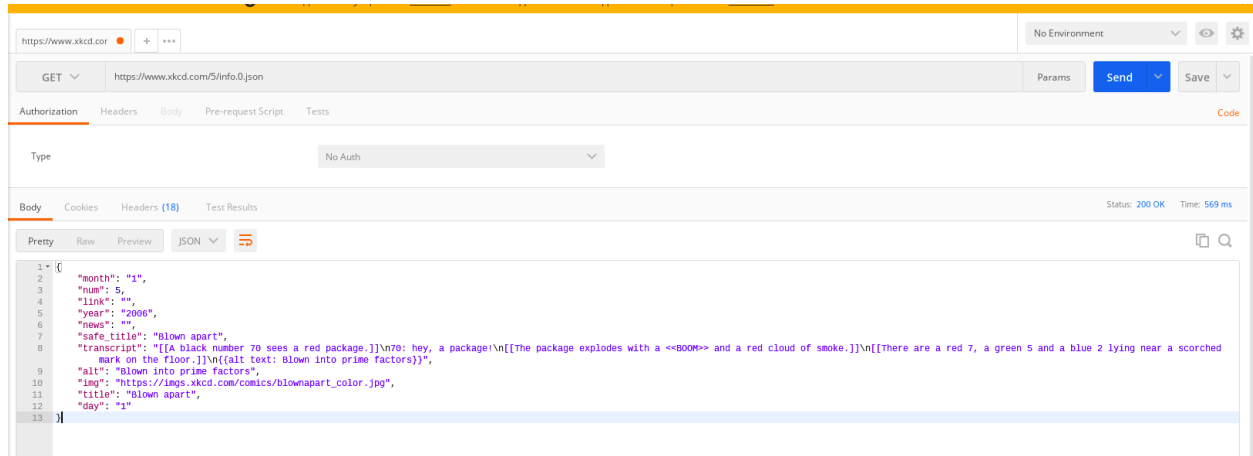
<http://xkcd.com/2472/info.0.json> (comic #2472)

Those files contain, in a plaintext and easily-parsed format:

comic titles, URLs, post dates, transcripts (when available), and other metadata.

## Task

The xkcd API lists 87 comics in the xkcd webcomic. For the task we would like you to use a random number generator that picks a number between 1-87. Using these random numbers you will be pulling 15 comics from the API using Python. (*reference below*)



We would like you to:

1. GET 15 random comics and following details in using Python.
  - a. Get the names of the comic
  - b. Get the alt-text of the comic
  - c. Get the number of the comic
  - d. Get the link of the comic
  - e. Get the image of the comics
  - f. Get the image Link of the comics
2. Insert into MySQL - Please include SQL database schema(s) for any table(s) created in your Github repo
3. Write a script called *task\_one.py* that when called will output something like this to the console. (*reference below*).

```
[
  {
    "comic": "Barrel - Part 1",
    "comic meta": {
      "alt_text": "Don't we all.",
      "number": 1,
      "link": "https://www.xkcd.com/1",
      "image": "barrel_cropped_(1).jpg",
      "image_link": "https://imgs.xkcd.com/comics/barrel_cropped_(1).jpg"
    }
  },
  {
    "comic": "Petit Trees (sketch)",
    "comic meta": {
      "alt_text": "'Petit' being a reference to Le Petit Prince, which I only thought about halfway through the sketch",
      "number": 2,
      "link": "https://www.xkcd.com/2",
      "image": "tree_cropped_(1).jpg",
      "image_link": "https://imgs.xkcd.com/comics/tree_cropped_(1).jpg"
    }
  },
  {
    "comic": "Island (sketch)",
    "comic meta": {
      "alt_text": "Hello, island",
      "number": 3,
      "link": "https://www.xkcd.com/3",
      "image": "island_color.jpg",
      "image_link": "https://imgs.xkcd.com/comics/island_color.jpg"
    }
  }
]
```

## Bonus Task

- We would like you to write unit tests for your functions and end to end testing for the tasks.

This is not required but if you would like to attempt it, please do so:

- Ask your to show comics like -
  - Would you like to read comics? Please provide a number/Title and read on the web!

ARCHIVE  
WHAT IF?  
BLAG  
HOW TO  
STORE  
ABOUT  
FEED • EMAIL  
TW • FB • IG

xkcd

A WEBCOMIC OF ROMANCE,  
SARCASM, MATH, AND LANGUAGE.

BLACK LIVES MATTER

 [HOW TO HELP](#)

BARREL - PART 1

[<](#)

[< PREV](#)

[RANDOM](#)

[NEXT >](#)

[>](#)



[<](#)

[< PREV](#)

[RANDOM](#)

[NEXT >](#)

[>](#)

PERMANENT LINK TO THIS COMIC: <https://xkcd.com/1/>

IMAGE URL (FOR HOTLINKING/EMBEDDING): [https://imgs.xkcd.com/comics/barrel\\_cropped\\_\(1\).jpg](https://imgs.xkcd.com/comics/barrel_cropped_(1).jpg)



Good luck, and may the force be with you!!