Home Assignment

Imagine a client has asked you to build a tool that can display data on nearby asteroids using NASA's APIs. We don't expect all the bells and whistles you would implement in a real solution, but your solution must provide at least the following functionality:

- Allow the end user to input a time period (start date and end date).
- Output detailed information of the asteroid that passed closest to Earth during that time period. For example, between 10th December 2015 and 31st December 2015, the closest asteroid was: "id": "3738548", "name": "(2015 YJ)", "miss_distance.kilometers": "71228.647194367". **Note: the API only returns 7 days of data at a time, but your solution needs to accommodate time ranges longer than 7 days.**
- Set up at least a few tests to demonstrate your approach to testing.

The assignment will give us material to discuss during your technical interview. We will consider your approach to problem-solving, coding style, and knowledge. We appreciate clear and clean code with a consistent style, naming and comments when needed.

Choose your approach

As long as your solution fulfills the requirements, you can choose your own approach, languages, frameworks, etc. Choose technologies that you know well: we really want to see your core strengths demonstrated. You can submit a minimal solution if you are short on time, or you can make a larger effort and show off. Example approaches:

A) **Minimal solution for busy candidates**: a command-line solution (e.g. Python or NodeJS) that takes start date and end date as command line arguments, fetches data from the API and outputs results to command line.
B) **Front-end show-off:** create front-end only. Your front-end displays the data from the API in a table, a chart, or in another visual representation form. Show off your design and accessibility chops, or other aspects that you feel are important in front-end development.
C) **Full-stack show-off:** create front-end and back-end. Your backend service fetches data from NASA's API, processes it and provides an API for a simple front-end. Show off how to divide responsibilities between the front-end and back-end, how to design the API, and how to glue everything together. We expect your backend to support a large number of concurrent users.

Please return the assignment with instructions on how to run it either in a Git repository or a zip file.

API to use

Asteroids - NeoWs

Go to https://api.nasa.gov/. Navigate to Browse APIs and look for: Asteroids - NeoWs.
The API you need for this task is called Neo - Feed

The NASA APIs require you to sign up here (it's free), this will allow you approximately 1000 API calls per hour. You can generate an API key at https://api.nasa.gov/ and select Generate API key.