Application Programming Interface

What's in it for a Data Scientist

Francis Jeremiah Majawa

Supervisor: Dr. D. Matekenya

January 19, 2023

What is an API?

An API (Application Programming Interface) is a set of rules/functions and protocols that allows different systems systems to communicate with each other.

Figure: Communation among applications.

Categories of API

1. Web-based system
   - A web API is an interface with a web server.
   - Extensively used for the development of web applications.
2. Operating system
   - Offers the functionality of various OS features taht can be incorporated in creating windows or mac or linux applications.
3. Database system
   - These APIs are defined in a manner to pass out the requested data in a predefined format that is understandable by the requesting client.
   - This makes the process of interaction with databases generalised and thereby enhancing the compatibility of applications with the various databases.

How Do Web API Work?

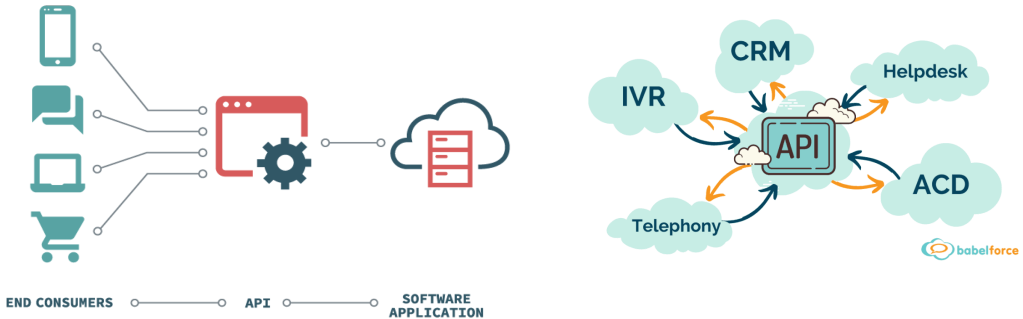

Figure 2: shows how an API works

API in summary

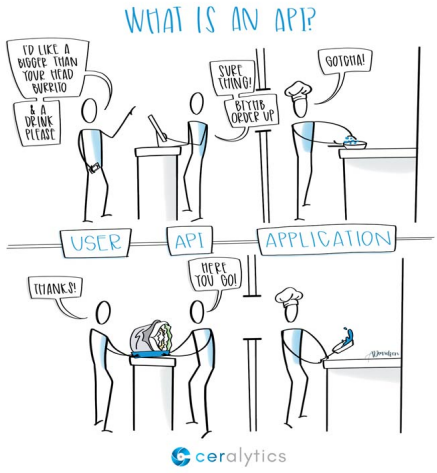

Figure: How API works

Why do Data Scientists need API?

1. To access data from external sources, such as databases or web services, in order to integrate that data into their models and analyses.
2. To expose models and other tools developed by data scientists to other systems and applications, allowing those systems to make use of the data and insights generated by the data science team
3. To automate certain tasks, such as data collection or model training, which can save time and resources for the data science team.

Examples of API for Data Scientists

1. Amazon Machine Learning API
2. IBM Watson Discovery API
3. Google API
4. Twilio API
5. Census.gov API
6. Spotify API
7. Yummly API
8. Reddit API
9. Zillow API
10. Instagram API
11. Twitter API
12. Big ML API
13. Data Science Toolkit API
14. New York Times API

How to build a API for Data Sciences

- Building an API for data science typically involves several steps:
  1. Define the API endpoint(s): Determine what data or functionality the API will provide and how it will be accessed (e.g. via a RESTful interface).
  2. Collect and prepare the data: Gather the necessary data and perform any preprocessing or cleaning that is required.
  3. Develop the API: Use a web framework (such as Flask or Django) to develop the API endpoint(s) and connect them to the data.
  4. Test the API: Test the API to ensure it is functioning correctly and providing the expected results.

Con'td

5. Deploy the API: Deploy the API to a server or hosting platform, making it available for use by other systems and applications.
6. Secure the API: Implement security measures such as authentication and authorization to protect the API and its data from unauthorized access.
7. Document the API: Provide clear documentation for the API, including information on how to access it, what data or functionality it provides, and any constraints or limitations.
8. Monitor and maintain the API: Monitor the API for errors or issues, and make any necessary updates or changes to keep it running smoothly.

Demo

The Architecture and tools

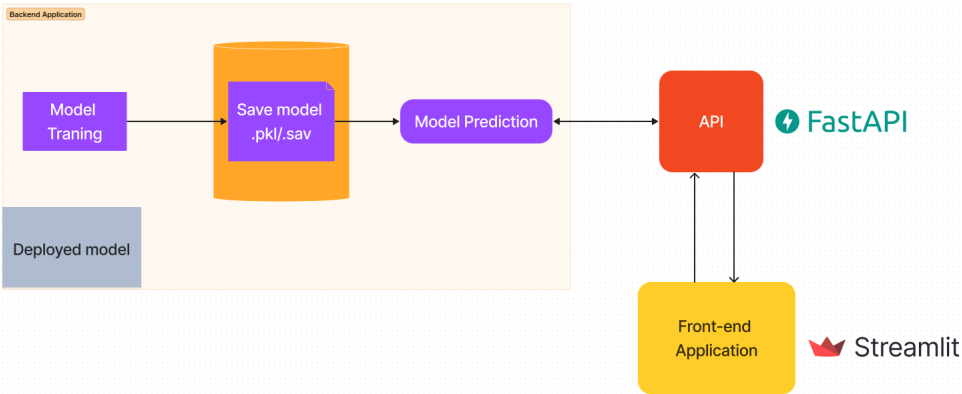

Figure: shows the architecture of the demo application

Thank you for your attention.