Chapter 08 - What Data Scientists Should Know About APIs

Using Variety of API Styles

- To interact with an Web (interface) API we interact through HTTP (protocol) verbs.

HTTP - Hypertext Transfer Protocol

It's a protocol. Protocols are a specific way to communicate with something, HTTP doing the same thing, it determines how things "talk"

IP: Port

Protocols need an address and a port

HTTP is built on TCP/IP - Transmission Control Protocol/Internet Protocol

How lots of devices communicate

- gRPC enables cross-language remote procedure calls $\xrightarrow{\text{means}}$ your program code can call external gPRC service like a local one (?)
  - mostly used when calling a machine learning model such as LLMs
- Difference of gPRC with other APIs architecture $\xrightarrow{\text{result into}}$ faster communication and support two-way data streaming
  - gRPC uses data format called "protocol buffer" instead of JSON
  - gRPC uses HTTP/2 communication protocol (HTTP/1 uses by GraphQL & REST)

HTTP Basics

- Only use APIs with "HTTPS" in the URL $\xrightarrow{\text{means}}$ API traffic will be encrypted in transit.

HTTP Verbs

- HTTP Verbs $\overset{\text{HTTP Standard Document}}{=}$ HTTP Method
- Indicates the purpose for which the client has made this request and what is expected by the client as a successful result
    - `GET`: Read a resource or list of resources.
    - `POST`: Create a new resource.
    - `PUT`: Update an existing resource.
    - `DELETE`: Remove an existing resource
    - For `GET`, `DELETE` providing the URL is enough
        - But `POST`, `PUT` need some data to do the action
        - Solution = HTTP Message Body: Body contains JSON or XML data that the API uses to perform the action
            - For GraphQL you always sending POST request $\rightarrow$ the body of the message contains the query that you are sending to the API

HTTP Status Codes

- 2XX: indicate success.
    - 200 OK: The request was successful
    - 201 Created: A POST method successfully created a resource
- 3XX: indicate redirection.
    - 301 or 308 Mover Permanently: The API address has moved permanently, so you should change your API call
    - 302 Moved Found: The API address redirected temporarily. Keep using the address you used.
- 4XX: indicate client error.
    - 400 Bad Request: Your request has an error or invalid request
    - 401 Unauthorized: Invalid credentials to make the API call
    - 404 Not Found: The resource doesn't exist or the address is wrong.
- 5XX: indicate server error
    - 500 Internal Server Error: Something failed unexpectedly on the server.
    - 503 service Unavailable: Temporary issue with service. Retry may be appropriate.

How to Consume API Responsibly

- Follow the Terms of Service
  It provide what you should expect and the requirements of API providers have for you to use their API.
- Handle Retries Gently
  To avoid overwhelming the service with your automated retry process, Implement backoff and retry process
- Handle Credentials Safely
  Register to use API $\overset{\text{way to}}{=}$ Monitor the user activity
  You will have username, passwords, API keys, secret keys, tokens etc. $\rightarrow$ store them securely and implement them in your code by use of secret manager or environment variables
  Google Tips on securely using API keys
- Validate Inputs and Outputs
  You should handle data you receive from APIs carefully $\rightarrow$ SQL injection, etc.
  Send expected data to APIs
- Log and Diagnose Erros
  When you using an API in a recurring data pipeline handles and log errors $\rightarrow$ handle them ini a organized fashion $\rightarrow$ make debugging sessions easier

Separation of Concerns: Using SDKs or Creating API Clients

- Separation of Concerns (SoC) = Important principle in software development
  - Means: Computer Program should be broken up into chunks that performa a specific task
    - Using API responsibly and calling it should be separate things
- If SDK is available use it, it makes your life easier + it already implemented advance features such as backoff, retry, data validation, error handling, logging.

How to Build APIs

- You can create API as an "Inference Endpoint" to share your statistical model or Machine learning model

How to Test APIs

- API Producer: Perform test through their Development, Deployment and Maintenance phases of hosting and API

        - Responsible for: ensuring API is reliable + lives up to customer expectations + Service Level Agreements (SLA)
            - SLA = formal agreements that producers make with consumers about uptime, performance, or other aspects of API service
- API consumers: test them before using them into your system
- Postman Recommendation for API testing:
    - Contract Testing: Verifies the format and behavior of each endpoint
    - Unit Testing: Confirms the behavior of an individual endpoint
    - End-to-End Testing: Tests workflows that use multiple endpoints
    - Load Testing: Verifies performance items such as the number of concurrent request that can be processed at peak times and the response time for individual requests
        - Locust: Python load-testing library
- Other types of Testing Resources:
    - The Agile Testing Quadrants from Janet Gregory and Lisa Crispin
        - Comprehensive Testing:
            - Technology-facing tests:
                - Unit testing
                - Performance testing
            - Business-facing tests:
                - Prototyping test
                - Usability test
- Don't forget to include your API documentation and SDKs in your testing

API Deployment and Containerization

Containerization: Packaging your program code into a reusable package that can be run locally or on another server or cloud provider. → Docker is a software that do Containerization

Using Version Control

Version Control is a way of tracking what changes have been made to a codebase, and it allows multiple people to work on the same code easily

Project II:

Chapter 9: Using APIs in data analytics products using Jupyter Notebook
Chapter 10: Using APIs in data pipelines using Apache Airflow
Chapter 11: Using APIs in a Streamlit data application

Additional Resource:

- Public GraphQL API for information about Countries