

Application Programming Interface

Predicting from a trained model with API call

Primož Konda

AAUBS

March 31, 2023

Outline

1 What is the plan for today?

2 Introduction

3 First example

4 FastAPI

5 Project transformation

Combined lectures

The aim:

- First we have a brief introduction to API

Combined lectures

The aim:

- First we have a brief introduction to API
- We continue from where we ended yesterday

Combined lectures

The aim:

- First we have a brief introduction to API
- We continue from where we ended yesterday
- We will prepare and deploy FastAPI server

Combined lectures

The aim:

- First we have a brief introduction to API
- We continue from where we ended yesterday
- We will prepare and deploy FastAPI server
- Finally, we will make a prediction

Combined lectures

The aim:

- First we have a brief introduction to API
- We continue from where we ended yesterday
- We will prepare and deploy FastAPI server
- Finally, we will make a prediction

Application Programming Interface?

Definition

- 1 It is a set of defined rules that enable different applications to communicate with each other.
- 2 It acts as an intermediary layer that processes data transfers between systems.

Application Programming Interface?

Definition

- 1 It is a set of defined rules that enable different applications to communicate with each other.
- 2 It acts as an intermediary layer that processes data transfers between systems.

Usability

APIs simplify software development and innovation by enabling applications to exchange data and functionality easily and securely

Types regarding availability:

- **Public API** is open and available for use by any outside developer or business. These are also called open or external APIs

Types regarding availability:

- **Public API** is open and available for use by any outside developer or business. These are also called open or external APIs
- **Partner API** is only available to specifically selected and authorized outside developers or API consumers. It facilitates business-to-business activities

Types regarding availability:

- **Public API** is open and available for use by any outside developer or business. These are also called open or external APIs
- **Partner API** is only available to specifically selected and authorized outside developers or API consumers. It facilitates business-to-business activities
- **Private API** is intended only for use within the enterprise to connect systems and data within the business

Types regarding availability:

- **Public API** is open and available for use by any outside developer or business. These are also called open or external APIs
- **Partner API** is only available to specifically selected and authorized outside developers or API consumers. It facilitates business-to-business activities
- **Private API** is intended only for use within the enterprise to connect systems and data within the business
- **Composite API** is a sequence of tasks that run synchronously as a result of the execution and not at the request of a task.

Types regarding protocol:

- 1 **REST** (Representational State Transfer) is a web services API and crucial for modern web applications

Types regarding protocol:

- 1 **REST** (Representational State Transfer) is a web services API and crucial for modern web applications
- 2 **SOAP** (Simple object access protocol) is a well-established protocol but comes with strict rules, rigid standards

Types regarding protocol:

- 1 **REST** (Representational State Transfer) is a web services API and crucial for modern web applications
- 2 **SOAP** (Simple object access protocol) is a well-established protocol but comes with strict rules, rigid standards
- 3 **RPC** (Remote Procedure Call protocol) is the oldest and simplest type of API with a goal for the client to execute code on a server

Types regarding protocol:

- 1 **REST** (Representational State Transfer) is a web services API and crucial for modern web applications
- 2 **SOAP** (Simple object access protocol) is a well-established protocol but comes with strict rules, rigid standards
- 3 **RPC** (Remote Procedure Call protocol) is the oldest and simplest type of API with a goal for the client to execute code on a server
- 4 **Event-driven** or asynchronous APIs transmit information in quasi-real-time. The advantage is that it allows the source to send a response only when the information is new or has changed, useful for stock exchanges

Requests:

- GET - is a read-only operation and doesn't change the state of the resource but only retrieve data
- POST - sends data to a server to create or update a resource and is often used when submitting web forms
- PUT - is used to update an existing resource with new data
- PATCH - is used to apply partial modifications to a resource
- DELETE - used to delete a specified resource from the server

How to get weather data from online API?

TASK: We want to create a simple web and mobile app showing recent and historical weather and air pollution data.

How to get weather data from online API?

TASK: We want to create a simple web and mobile app showing recent and historical weather and air pollution data.

There are many websites but not all offer APIs, e.g., DMI.

How to get weather data from online API?

TASK: We want to create a simple web and mobile app showing recent and historical weather and air pollution data.

There are many websites but not all offer APIs, e.g., DMI.

We can get data for free on <https://open-meteo.com/>.

Exercise competition

You have 15 minutes to get data on Air Quality (PM 2.5) for Roskilde and Beijing

Use open-meteo and you can look at my code. Hint: AirQuality API is in OtherAPIs :) !

FastAPI

FastAPI is a modern, fast (high-performance) web framework for building APIs with Python 3.7+.

FastAPI

FastAPI is a modern, fast (high-performance) web framework for building APIs with Python 3.7+.

- It is fast

FastAPI

FastAPI is a modern, fast (high-performance) web framework for building APIs with Python 3.7+.

- It is fast
- Supports asynchronous code (async and await commands). It can perform multiple tasks concurrently. In that way, it doesn't need to wait until one called to be answered and can continue with a new request

FastAPI

FastAPI is a modern, fast (high-performance) web framework for building APIs with Python 3.7+.

- It is fast
- Supports asynchronous code (async and await commands). It can perform multiple tasks concurrently. In that way, it doesn't need to wait until one called to be answered and can continue with a new request
- Short development line 7-8 lines of code

Uvicorn + FastAPI

Uvicorn is a lightning-fast ASGI server that is used to serve FastAPI applications.

Uvicorn + FastAPI

Uvicorn is a lightning-fast ASGI server that is used to serve FastAPI applications.

FastAPI is built on top of Starlette which itself is built on top of Uvicorn.

Transformation process

Starting point: A folder with working python code for prediction on already trained model saved in sub-folder.

Transformation process

Starting point: A folder with working python code for prediction on already trained model saved in sub-folder.

Steps:

- Install dependencies: fastapi, uvicorn, pydantic

Transformation process

Starting point: A folder with working python code for prediction on already trained model saved in sub-folder.

Steps:

- Install dependencies: fastapi, uvicorn, pydentic
- Create a new python script based on prediction script which initiate fastapi app

Transformation process

Starting point: A folder with working python code for prediction on already trained model saved in sub-folder.

Steps:

- Install dependencies: fastapi, uvicorn, pydantic
- Create a new python script based on prediction script which initiate fastapi app
- define request parameters and their types

Transformation process

Starting point: A folder with working python code for prediction on already trained model saved in sub-folder.

Steps:

- Install dependencies: fastapi, uvicorn, pydantic
- Create a new python script based on prediction script which initiate fastapi app
- define request parameters and their types
- start uvicorn server and define address and fastapi app

Transformation process

Starting point: A folder with working python code for prediction on already trained model saved in sub-folder.

Steps:

- Install dependencies: fastapi, uvicorn, pydentic
- Create a new python script based on prediction script which initiate fastapi app
- define request parameters and their types
- start uvicorn server and define address and fastapi app
- Send request and hope for response 200 :)