Sports API

Program : Data Engineer

Difficulty : 8.5/10

Description :
This project aims to create a sports API similar to Transfermarkt to provide comprehensive and reliable information about teams, players, matches, and results from various sports leagues around the world. This API will be developed using a rigorous data engineering approach, which includes the collection, storage, transformation, and distribution of data from multiple sources. The main objective of this project is to create an interactive and user-friendly platform that will allow users to easily access accurate and up-to-date information about their favorite teams and players.

Step	Description	Goal	Courses/ Masterclass / Templates	Conditions of validation
1	Collecting data	Go through Sports APIS to get the historical data of the games and the predictions of the sports bets. We can focus on the soccer version. Some freemium APIS from Rapid API may also prove useful. Another solution is to use Prediction API (1 year of history, 300 requests per hour). We can also use webscrappers from sites like Transfermarket, Who scored, Sports Reference This step is important, you must understand the data you can retrieve and choose the endpoints to use	Use of the requests library or the Postman tool (to test) Webscraping techniques (133 Beautiful Soup, Selenium)	Explanatory file of the treatment and the various accessible data (doc / pdf) A sample of collected data
2	Data modeling	There are several options available to us. In the previous step, we observed that there are several "types" of data. We will qualify **fixed data** (star schema) as information on players, teams, leagues, matches and variables, predictions on matches/market value of players This diversification of data will lead to the use of different databases.	142 - SQL Elasticsearch 143 - MongoDB Neo4j	A relational database UML diagram A SQL query file to show that it works Same rendering but examples of Elastic/Mongo queries
3	Data consumption	Games Prediction: Use the bettors' odds on games or the financial value of players to propose prediction models on	Dash	Appli Dash

		games or detect nuggets. Analytics: The second use is a reporting tool, instead it will be necessary to make an API to query these different databases with nice dashboards (for players with pentagon/hexagon performance metrics). You may find inspiration from similar projects : this one, that one or this project about NBA	Plotly	API FastAPI
4	Deployment	Make a Docker container of each project component(BDD,API) and make a working docker-compose.	FastAPI Docker	API FastAPI Docker, docker-compose file
5	Automation	It is necessary to retrieve live data from the Sports API according to a well-defined rhythm to update the databases and send it to the various consumers of the data.	Airflow	Python file for the Airflow DAG
6	Defense	Demonstrate their application and explain the reasoning behind their project.	X	Defense Documentation

Useful Links : https://github.com/n0shake/Public-APIs#sport