

# Santex Back-end Developer Hiring Test

The goal is to create a **project** that exposes a REST API.

## What should this API do?

We'll be hitting <http://www.football-data.org/> API (*you can see the documentation in the site, use the API v4*) to populate the data locally and then expose it.

## Import League:

There should be an endpoint to import a league, named **"import league"**, that takes a **"league code"** as input.

The import league implementation must get data using the given **league code**, by making requests to the <http://www.football-data.org/> API, and **import** the data into a **DB**. Any SQL or NoSQL DB can be used, as long as there are **clear instructions** on how to run the project locally as well as an explanation for the decision in the README.

The data we're importing is:

- Competition ("name", "code", "areaName")
- Team ("name", "tla", "shortName", "areaName", "address")
- Player ("name", "position", "dateOfBirth", "nationality")

Note: When making the implementation, if there is no data (no players) inside team **squads**, then, instead of importing *players*, you should import only the **coach**:

- Coach("name", "dateOfBirth", "nationality") (import that data even if the values are null)

Feel free to add to this data structure any other field that you might need.

## Information to retrieve:

Additionally, expose the following endpoints, that should rely exclusively on the data saved inside the DB (it **must not** access the API [football-data.org](http://www.football-data.org/)):

- **players**: takes **league code** as a parameter and returns the players that belong to all teams participating in the given league. If the given **league code** is not present in the DB, it should respond with an error message. Add an optional input to the endpoint to filter players also by team name.
- **team**: takes a name and returns the corresponding team. Additionally, if requested in the query, it should resolve the players for that team (or *coaches*, if players are not available at the moment of implementation).
- **players of a team**: should resolve the players for the given team (or *coaches*, if players are not available at the moment of implementation).

### What we expect:

- **You must push all the relevant files to the repository provided to you**
  - If your name is John Doe, then the repository name should be <https://github.com/SantexGroup/be-challenge-jdoe>
  - Make sure to include anything related to project configuration and/or dependency management
  - Use good git practices
- Please notice that even though this is a paid API, you can get a free token and perform your testing with some specific competitions (PD, CL, PL).
- It's important that the code handles in some way the limit frequency to the requests performed with a free-token.
- You are allowed to use any library related to the language in which you are implementing the project.
- All the mentioned DB entities must keep their proper relationships (the players with which team they belong to; the teams in which leagues participate).
- It might happen that when a given **league code** is being imported, the league has participant teams that are already imported (because each team might belong to one or more leagues). For these cases, it **must** add the relationship between the league and the team(s) (and omit the process of the preexistent teams and their players).
- Please explain your train of thought and your decision making (for libraries/frameworks used) in the README or another doc inside the project.

### Nice to have:

- It is a plus that the project automatically generates any necessary schemas when it runs the first time.
- Usage of Docker or any other containerization is also a plus.
- Think about this as a real product and add anything you feel could add value (like other endpoints). **SHOW ALL YOUR SKILLS, SURPRISE US!**