## **JobFair 2021**

## **Nordeus - Data Engineering Challenge**

### **Description**

At Nordeus we love working with football-related data, as you can imagine. So, by no surprise, we have decided to make this task all about our favorite sport. As input you are given a dataset ([JSONL format](https://jsonlines.org/)) where each line is a valid JSON object that represents an event generated by a football mobile game. Each event emitted from the game has several fields/ parameters that describe the action that is happening inside the game. Some examples of events are: user logged in, match started, goal scored, match ended, user logged out, etc. [Link to the input dataset (events.jsonl).](https://drive.google.com/file/d/1Q3Xuk31A_utc8pg54De0bVWmVHb-Dzbx/view?usp=sharing)

Your task is to process, clean and transform the source dataset into a new data model that will be capable of answering certain questions about the state of the game, specifically, we are interested in the current state of the leaderboard for each league.

Dataset (events.jsonl)

| Parameter Name | Parameter Type | Parameter Description |
| --- | --- | --- |
| event\_id | INT | Unique identifier representing an event |
| event\_timestamp | INT | Time of event represented as [Unix time](https://en.wikipedia.org/wiki/Unix_time) |
| event\_type | STRING | One of the following: match\_start, goal, or match\_end |
| event\_data | JSON OBJECT | JSON object containing all event-specific data (check event data below) |

Required event data for match\_start

| Parameter Name | Parameter Type | Parameter Description |
| --- | --- | --- |
| match\_id | INT | Unique identifier representing a match |
| league\_id | INT | Unique identifier representing a league |
| home\_club | STRING | Name of home club |
| away\_club | STRING | Name of away club |

Required event data for goal

| Parameter Name | Parameter Type | Parameter Description |
| --- | --- | --- |
| match\_id | INT | Unique identifier representing a match |
| scoring\_club | STRING | One of the following: home or away |

Required vent data for match\_end

| Parameter Name | Parameter Type | Parameter Description |
| --- | --- | --- |
| match\_id | INT | Unique identifier representing a match |

### **Requirements**

Data cleaning requirements:

* Discard duplicate events (event\_id uniquely identifies an event)
* Discard events that are missing required fields in event\_data
* Discard goal events that happened before match\_start or after match\_end
* Only matches with valid match\_start and match\_end events count towards the leaderboard

API requirements:

* Get leaderboard for specific league
  + The winning team gets 3 points, and in the case of draw, both teams are getting 1 point
  + If multiple teams have the same number of points, order them by goal difference (goals scored - goals conceded) in decreasing order
  + If multiple teams have the same number of points and the same goal difference, order these teams by club name in increasing alphabetical order
  + Input: an integer representing a league id
  + Output: Ordered list/array of tuples/objects (club\_name, points, goal difference) representing the leaderboard for the input league

### 

### **Expectations**

The input file should be processed **only once** and the target data model should be somehow persisted (in-memory, file, database, etc.), so that queries can be executed efficiently. It is not recommended to process the input file each time a query is executed.

You have freedom to implement the API in any way you find appropriate and in any language (Python, Java, SQL, etc.) for example:

* REST API implemented in any language
* CLI APP implemented in any language
* Other

One important thing here is that you should add a **documentation** file (possibly README.md) where you explain which approach you have chosen and how a user should interact with the application (how to start the app and get the leaderboard for some league).

Bonus points for:

* Implementing a detailed and careful data cleaning process
* Storing the target data model in a SQL database and using SQL to get the leaderboard
* Implementing the solution as a well-documented REST API

### **Submission**

* Upload the solution to a new public repository in your GitHub profile
* Send an email to [jobfair@nordeus.com](mailto:jobfair@nordeus.com) with a link to your Github repository (email subject: Data Engineering challenge)

### **Example**

The correct state of the leaderboard for league with ***league\_id=8,***after discarding incorrect events is:

| Rank | Club | Points | Goal difference |
| --- | --- | --- | --- |
| 1. | Win FC | 29 | 14 |
| 2. | Talent FC | 22 | 0 |
| 3. | Little Spartans | 17 | 9 |
| 4. | Captain FC | 16 | 3 |
| 5. | Sprinters | 13 | -2 |
| 6. | Losers | 13 | -6 |
| 7. | Wolves | 13 | -6 |
| 8. | Golden Knights | 9 | -5 |
| 9. | The Braves | 7 | -7 |

**Open until November 8**, end of day.