**REST API Design and Implementation Document**

**Introduction**

This document describes the design and implementation of a REST API for managing matches and their performances. The API is implemented using Java and Spring Boot, and uses PostgreSQL as the database. Spring Data JPA is used to access the database.

**API Endpoints**

The following endpoints are provided by the API:

**Matches**

GET /api/matches - get a list of all matches

GET /api/matches/{id} - get a specific match by ID

POST /api/matches - create a new match

PUT /api/matches/{id} - update an existing match by ID

DELETE /api/matches/{id} - delete a match by ID

**Match Odds**

GET /api/match-odds - get a list of all match odds

GET /api/match-odds/{id} - get a specific match odd by ID

POST /api/match-odds - create a new match odd

PUT /api/match-odds/{id} - update an existing match odd by ID

DELETE /api/match-odds/{id} - delete a match odd by ID

Data Model

The following entities are managed by the API:

**Match**

id - a unique identifier for the match

description - a description of the match

match\_date - the date the match will be played

match\_time - the time the match will be played

team\_a - the name of one of the teams playing in the match

team\_b - the name of the other team playing in the match

sport - an enumeration indicating the sport of the match (1 = Football, 2 = Basketball)

**Match Odds**

id - a unique identifier for the match odd

match\_id - the ID of the match the odds are associated with

specifier - a specifier indicating the type of odds (e.g. "X" for a draw in football)

odd - the value of the odds

Error Handling

The API returns error responses in JSON format, with the following structure:

**json**

{

"status": <HTTP status code>,

"message": <error message>

}

**Authentication and Authorization**

The API does not currently require authentication or authorization to use.

**Deployment**

The API can be deployed as a standalone Spring Boot application,

**Conclusion**

This REST API provides a simple and flexible way to manage matches and their performances. It is easy to use and can be deployed as a standalone application or as a Docker container.