

LeaderboardAPI Service Documentation

Introduction

This is an API for calculating leaderboard and rewards for a given month. The API is built using ASP.NET Core and MongoDB. The LeaderboardAPI is a service that allows users to create and view leaderboards and rewards. It includes several classes and methods to handle the different functionalities. The API also allows for listing of user rewards and leaderboards, and for filtering by user ID and month. The rewards are calculated based on user's point data and the leaderboard is created based on the user's ranking.

Getting Started

1. Install the latest version of .Net Core 6.0 and MongoDB, and ensure that it is running on your local machine.
2. In your project, install NuGet and then install the MongoDB.Bson and MongoDB.Driver packages from NuGet and add it to your dependencies.
3. In your appsettings.json file, include the following information for connecting to your MongoDB database:

```
"MongoDb": { "ConnectionString":  
  "mongodb+srv://<username>:<password>@cluster0.skqjzkj.mongodb.net/<dbname>?retryWrites=true&w=majority", "Database": "<dbname>" }
```

4. In the Startup class, configure the service collection to include the MongoDBConnection service and add it to the service provider.
5. In the Program class, use the RewardsCalculator service to create the leaderboard and distribute rewards for a specific month.
6. Use the RewardsCalculator service to list the leaderboard and user rewards, including the ability to filter by user ID.

The main classes include:

- **MongoDBConnection:** This class is responsible for connecting to the MongoDB database, and includes methods for adding and retrieving data from the collections.

- **RewardsCalculator:** This class is responsible for calculating the rewards and rankings based on the user point data. It includes methods for creating the leaderboard, calculating rewards, and listing both the leaderboard and rewards.
- **LeaderboardController:** This class is responsible for handling the HTTP requests and routes, and includes methods for creating, listing and getting the leaderboard and rewards.

To use the API, the user must first instantiate the MongoDBConnection class and pass in the MongoDB connection string and database name. Then, the user can use the methods provided by the RewardsCalculator and LeaderboardController classes to create and view the leaderboards and rewards.

Creating the leaderboard:

- The CreateLeaderboard() method in the RewardsCalculator class is used to create a new leaderboard. It takes the month in the format of MMYYYY or MMMMYYYY. If the leaderboard of the requested month already exists, it will give the error "Leaderboard of the requested month already exists" and return the existing leaderboard.

Listing the leaderboard:

- The ListLeaderboard() method in the RewardsCalculator class is used to retrieve the leaderboard. It takes the month in the format of MMYYYY or MMMMYYYY. It will return a list of leaderboard data containing user_id, rank and total points
- The ListLeaderboardByUserId() method in the RewardsCalculator class is used to retrieve the leaderboard by user id. It takes one argument, userId in string format. It will return a list of leaderboard data containing user_id, rank and total points

Calculating Rewards:

- The CalculateRewards() method in the RewardsCalculator class is used to calculate rewards based on the point data. It takes one argument, the month in the format of MMYYYY. It will first retrieve the approved user point data, then calculate the rankings based on the point data. It will remove any existing rewards and distribute rewards based on the rankings. It will then create the leaderboard.

Listing the rewards:

- The ListAllRewards() method in the RewardsCalculator class is used to list all the rewards. It takes no argument. It will return a list of rewards data containing user_id, rank and rewards
- The ListUserRewards() method in the RewardsCalculator class is used to list the rewards by user id. It takes one argument, userId in string format. It will return a list of rewards data containing user_id, rank and rewards.

Endpoints

/leaderboard/create - This endpoint is used to create a leaderboard for a specific month. The month should be passed as a parameter in the format of "MMYYYY". If the leaderboard for that month already exists, it will return an error message "Leaderboard of the requested month already exists".

/leaderboard/list - This endpoint is used to list the leaderboard for a specific month. The month should be passed as a parameter in the format of "MMYYYY". It returns an array of leaderboard objects, which includes the user_id, rank, and total points.

/leaderboard/list/{userId} - This endpoint is used to list the leaderboard for a specific user for a specific month. The month should be passed as a parameter in the format of "MMYYYY" and userId should be passed as a path parameter.

/rewards/list - This endpoint is used to list the rewards for a specific month. The month should be passed as a parameter in the format of "MMYYYY". It returns an array of rewards objects, which includes the user_id and rewards.

/rewards/list/{userId} - This endpoint is used to list the rewards for a specific user for a specific month. The month should be passed as a parameter in the format of "MMYYYY" and userId should be passed as a path parameter.

Models

1. **Points** - This model represents the user's point data.
2. **UserRewards** - This model represents the rewards that are distributed to the users based on their ranking.

3. **Leaderboard** - This model represents the leaderboard that is created based on the user's ranking.

Data Models

Points

- user_id (ObjectId)
- point (int)
- id (string)
- approved (bool)

Leaderboard

- user_id (string)
- point (int)
- rank (int)
- _id (ObjectId)

UserRewards

- user_id (ObjectId)
- reward (string)
- prize (int)
- _id (ObjectId)

Dependencies

1. MongoDB - This API service uses MongoDB as the database to store and retrieve data.
2. NuGet Libraries - MongoDB.Bson and MongoDB.Driver are required.

Data

The data for the leaderboard is taken from the "Points" collection in MongoDB. It contains the user_id, point, approved, and month fields. Only the approved points are considered while calculating the leaderboard and rewards.

Indexing

Indexing is applied on the "approved" field in the "Points" collection to improve the performance of the API.

Error Handling

If the month passed as a parameter is not in the correct format "MMYYYY", the API returns an error message "Month must be in the format MMYYYY".

If there is any other error while processing the request, the API returns a general error message "An error occurred while processing the request."

Note

1. Only approved scores are taken into account when calculating the leaderboard and rewards. Additionally, the program is designed to work with MongoDB database and a collection named points and Leaderboard_<month> respectively
2. The API does not have the functionality to delete or update the leaderboard and rewards once they are created.
3. The API assumes that there is already data present in the "rPoints" collection in MongoDB and it is in the correct format.
4. The API does not have any authentication or authorization.
5. The API does not have any validation for the userId passed as a path parameter.
6. API expects the userId to be passed as a string in ObjectId format. The API also expects the month to be passed in the format of "MMYYYY" or "MMMM YYYY" when calling the "ListLeaderboard" and "ListUserRewards" endpoints.
7. The API also has an endpoint called "CreateLeaderboard" which creates the leaderboard for the given month. If the leaderboard for the given month already exists, the API will return an error message "Leaderboard of the requested month already exists".
8. The API has another endpoint called "CalculateRewards" which calculates the rewards for the given month. If the rewards for the given month already exist, it will replace the existing rewards with the new rewards.
9. The API also has an endpoint called "ListAllRewards" which lists all the rewards for all months.

10. The API utilizes MongoDB as the database and has implemented indexing on the 'approved' field to improve performance.

Conclusion

Overall, The LeaderboardAPI service is a useful tool for tracking progress and rewards in a leaderboard system. It allows for the calculation and distribution of rewards, the creation of a leaderboard, and the listing of both user rewards and the leaderboard. With its simple and easy to use methods, it is a great addition to any project that needs to track user progress and rewards. the LeaderboardAPI is a powerful tool for working with user points data and creating leaderboards and rewards, and it is easy to use with the help of the detailed documentation provided.