

Notes

Designed and Developed a Sport Management Api system with bearer JWT (Json Web Token) authentication.

All the tasks including the Stretch goals are completed.

Tech Stack:

Language/Framework : C# /.net Core
Database : EFCore In Memory DB
Cloud Services: AWS Cloud Services

Date

13-Oct-2021

Location of deployed application

Deployed in AWS via AWS Elastic Beanstalk Service.

Base URL: <http://sportsmanagementapi-dev.ca-central-1.elasticbeanstalk.com/>

Isalive URL: <http://sportsmanagementapi-dev.ca-central-1.elasticbeanstalk.com/isalive>

API Doc URL : <http://sportsmanagementapi-dev.ca-central-1.elasticbeanstalk.com/swagger>

Time spent

23 hrs

Assumptions made

- A New user/manager is created via the managers endpoint
- Manager logs in with the credentials that was used when the manager was created. The login endpoint provides JWT bearer token. All the sport management API endpoints are authorized based on this token.
- Manager can create a team, delete a team, add players, get player details, remove a player, update player details of that team.
- Only Player Name and Details can be updated at the moment.
- Manager can create a schedule (or a game) between two teams with only his team as a home team. He can get the schedule with results, delete or update (past or future) the schedule (no past (scheduledTime is older than current time) game/schedule data except scheduleDetails can be changed).
- Manager can post a result of a game and get/update/delete the results of past or future games.
- Only Result Details can be updated at the moment.

Shortcuts/Compromises made

- To speed up development, I used an in-memory database.
- Haven't written any Unit/Integration Tests to save time. Definitely the next item in my list if I were to continue with this.
- Some Input validations are missing from in the endpoints.
- Didn't implement the OAUTH2 client server model where the client asks for a token from the server and the endpoints are validated with it. It requires a bit of understanding regarding how the whole .netcore identity flow works and that would have taken a lot longer. I settled with a basic jwt bearer token based auth where you have to copy the token as a header to get authorized.
- I put a lot of restrictions (during patching specially) and also didn't spend too much time on db design so there are room for improvements in those areas.

Stretch goals attempted

Stretch goals are completed-

- Api Documentation is generated using .net core swagger implementation (using swashbuckle.aspnetcore package) of open API (the swagger doc url is provided above)
- The API has been deployed to AWS (dev environment) using AWS Elastic Beanstalk service (please find the swagger doc and deploy url above).

Instructions to run assignment locally

- Install docker (<https://docs.docker.com/get-docker/>)
- Download the zip file
- Go to ..\sport management api\SportsManagementAPI where the dockerfile resides
- Build it with the following command- `docker build -t sport-management-api`.
- Run the docker container that was build (by the above command) with the following – `docker run -d -p 3334:80 sport-management-api`
- Open a browser and hit the following url - <http://localhost:3334/isalive> to check if the isalive endpoint is working.
- To get the api doc, use the following url - <http://localhost:3334/swagger>
- You can use a tool (i.e, postman) to check the api according to the specification. Also please find the sample api request below.

What did you not include in your solution that you want us to know about?

- A good set of tests
- A relational db(postgres) to permanently add data. So, as long as the app is running it will keep persisting data but the moment it restarts it deletes the data. This is done for testing and rapid development and definitely not production ready. Currently the app is running in aws and will run till the interview is over, so it will keep persisting data.

Other information about your submission that you feel it's important that we know if applicable.

Below I will post request and responses for some of the endpoints (the rest are similar to these below). For lack of time I couldn't create detailed swagger examples (it's showing the contract currently but no example request or response is present) so I will be providing those examples here (all the endpoint url below based on the aws deployed service's base URL. You can replace the baseurl with <http://localhost:3334/> if you want to hit this endpoints after running locally via a docker container) -

- Create Manager Endpoint
url - <http://sportsmanagementapi-dev.ca-central-1.elasticbeanstalk.com/api/managers/>
Request Example :

```
{
  "email" : "test2@test.com",
  "password": "1dfds232"
}
```

Response Example:

```
{
  "id": "cc88be75-8e08-4dc2-8941-611984e69542",
  "email": "test2@test.com"
}
```

- Login Endpoint-
url - <http://sportsmanagementapi-dev.ca-central-1.elasticbeanstalk.com/api/login/>
Request Example: // same email and password used to create the manager needs to be provided here or you get a bad request with invalid credentials msg.

```
{
  "email" : "test2@test.com",
  "password": "1dfds232"
}
```

Response Example- // the access token part is the bearer token that you need to copy and put it in the authorization header when you hit the other endpoints. Also, the expiration is datetime type representing ticks. The expiration time is set to 12000s from the current time so that we have ample time to test. We can easily change this setting in the appsettings.json (change the value of the property AccessTokenExpiration which is in seconds).

```
{
  "accessToken":
    "eyJhbGciOiJodHRwOi8vd3d3LnczLm9yZy8yMDAxLzA0L3htbGRzaWctbW9yZSNOYWJLXNoYTI1NiIsInR5cCI6IkpXVCJ9.eyJqdGkiOiJyZg4YmU3NS04ZTA4LTRkYzltODk0MS02MTE5ODRINjk1NDIiLCJzZWliOiJ0ZXN0MkBOZXN0LmNvbSIsIm5iZil6MTYzNDE2NzcwNywiZXhwIjoxNjM0MTc5NzA3LCJpc3MiOiJ0JTYyZG9ydE1hbmFnZW1lbnRBUEkiLCJhdWQiOiJTYW1wbGVbdWRpZW5jZSJ9.Gdu7Zm7P1abL7ccGzRxtUYVVsFAniAFraCnBsLG7xCUs",
  "refreshToken":
    "ADN7YM0fUvKPeMKtKKwFRmDTf4eZ2gP+P+qojfMwgTMk7NwyaikrmR91fk9+kccJbg==",
  "expiration": 637697765074745328
}
```

- Create Team Endpoint

url - <http://sportsmanagementapi-dev.ca-central-1.elasticbeanstalk.com/api/sportmanagement/createTeam>

Request Example:

Headers:

Authorization: Bearer

eyJhbGciOiJodHRwOi8vd3d3LnczLm9yZy8yMDAxLzA0L3htbGRzaWctbW9yZSNOYWJLXNoYTI1NiIsInR5cCI6IkpXVCJ9.eyJqdGkiOiJyZg4YmU3NS04ZTA4LTRkYzltODk0MS02MTE5ODRINjk1NDIiLCJzZWliOiJ0ZXN0MkBOZXN0LmNvbSIsIm5iZil6MTYzNDE2NzcwNywiZXhwIjoxNjM0MTc5NzA3LCJpc3MiOiJ0JTYyZG9ydE1hbmFnZW1lbnRBUEkiLCJhdWQiOiJTYW1wbGVbdWRpZW5jZSJ9.Gdu7Zm7P1abL7ccGzRxtUYVVsFAniAFraCnBsLG7xCUs // the token after Bearer is the same token we got from the login endpoint. Without a valid token you will get an 401 unauth.

Example picture with authorization header :

POST http://sportsmanagementapi-dev.ca-central-1.elasticbeanstalk.com/api/sportmanagement/createTeam

Params Authorization Headers (10) Body Pre-request Script Tests Settings

Headers 8 hidden

KEY	VALUE
<input checked="" type="checkbox"/> Content-Type	application/json
<input checked="" type="checkbox"/> Authorization	Bearer eyJhbGciOiJodHRwOi8vd3d3LnczLm9yZy8yMDAxLzA0L3htbGRzaWctbW9yZSNOYWJLXNoYTI1NiIsInR5cCI6IkpXVCJ9.eyJqdGkiOiJyZg4YmU3NS04ZTA4LTRkYzltODk0MS02MTE5ODRINjk1NDIiLCJzZWliOiJ0ZXN0MkBOZXN0LmNvbSIsIm5iZil6MTYzNDE2NzcwNywiZXhwIjoxNjM0MTc5NzA3LCJpc3MiOiJ0JTYyZG9ydE1hbmFnZW1lbnRBUEkiLCJhdWQiOiJTYW1wbGVbdWRpZW5jZSJ9.Gdu7Zm7P1abL7ccGzRxtUYVVsFAniAFraCnBsLG7xCUs
Key	Value

Body Cookies Headers (5) Test Results

Pretty Raw Preview Visualize JSON

```
1 {
2   "id": "1e289afd-1e03-41d3-9ef2-c8f6bf69e12c",
3   "name": "Thinkific FC",
4   "managerId": "cc88be75-8e08-4dc2-8941-611984e69542"
5 }
```

Body:

```
{
```

```
"Name": "Thinkific FC"
}
```

Response Example:

```
{
  "id": "1e289afd-1e03-41d3-9ef2-c8f6bf69e12c",
  "name": "Thinkific FC",
  "managerId": "cc88be75-8e08-4dc2-8941-611984e69542"
}
```

- Create Player Endpoint (all the subsequent endpoints would require the bearer token from now on)

url - <http://sportsmanagementapi-dev.ca-central-1.elasticbeanstalk.com/api/sportmanagement/createPlayer>

Request Example:

Headers

Authorization : Bearer TOKEN (replace with the actual access token here)

Body :

```
{
  "Name" : "Anton Griezman",
  "TeamName": "Thinkific FC",
  "Details": "Midfielder. Age 25. Player rating 10 out 10"
}
```

Response:

```
{
  "id": "1b77a16a-4047-4146-ae6b-bbe63fbedcc5",
  "name": "Anton Griezman",
  "teamId": "1e289afd-1e03-41d3-9ef2-c8f6bf69e12c",
  "managerId": "cc88be75-8e08-4dc2-8941-611984e69542",
  "details": "Midfielder. Age 25. Player rating 10 out 10"
}
```

- GetPlayers endpoint – // this will return all the players that the current manager (based on the auth token) manages.

url- <http://sportsmanagementapi-dev.ca-central-1.elasticbeanstalk.com/api/sportmanagement/getPlayers>

Request Example:

Headers

Authorization : Bearer TOKEN (replace with the actual access token here)

Response:

```
[
  {
    "id": "1b77a16a-4047-4146-ae6b-bbe63fbedcc5",
    "name": "Anton Griezman",
    "teamId": "1e289afd-1e03-41d3-9ef2-c8f6bf69e12c",
    "managerId": "cc88be75-8e08-4dc2-8941-611984e69542",
    "details": "Midfielder. Age 25. Player rating 10 out 10"
  }
]
```

- Delete Player Endpoint:

url - <http://sportsmanagementapi-dev.ca-central-1.elasticbeanstalk.com/api/sportmanagement/deletePlayer/45d3fdf9-b943-467c-9dea-26d9cdea3bed> // the id at the end is the playerId

Request Example:

Headers

Authorization : Bearer TOKEN (replace with the actual access token here)

Response:

204 no content

- Create Schedule Endpoint-

url - <http://sportsmanagementapi-dev.ca-central-1.elasticbeanstalk.com/api/sportmanagement/schedule>

Request Example:

Headers

Authorization : Bearer TOKEN (replace with the actual access token here)

Body :

```
{
  "HomeTeamName": "Thinkific FC",
  "AwayTeamName": "Udemy FC",
  "ScheduledTime": "2021-04-21T18:25:43-05:00Z",
  "ScheduleDetails": "League Game"
}
```

Response:

```
{
  "gameId": "9367eea1-c217-4393-a94d-c9d66e2643b8",
  "homeTeamId": "1e289afd-1e03-41d3-9ef2-c8f6bf69e12c",
  "homeTeamName": "Thinkific FC",
  "awayTeamId": "3ff6a229-9b2d-4f61-9689-12792f12128f",
  "awayTeamName": "Udemy FC",
  "scheduledTime": "2021-04-21T23:25:43+00:00",
  "managerId": "cc88be75-8e08-4dc2-8941-611984e69542",
  "scheduleDetails": "League Game"
}
```

// A manager can only schedule his team's home game, that means the HomeTeamName must match the team that the manager owns. Otherwise, it's going to throw an error.

- Create Result Endpoint –

url - <http://sportsmanagementapi-dev.ca-central-1.elasticbeanstalk.com/api/sportmanagement/result>

Request Example:

Headers

Authorization : Bearer TOKEN (replace with the actual access token here)

Body : // a valid gameId (can be found from the get schedule endpoint) and valid winnerTeam and LoserTeamName are required otherwise it will throw error.

```
{
  "gameId": "9367eea1-c217-4393-a94d-c9d66e2643b8",
  "winnerTeamName": "Thinkific FC",
  "loserTeamName": "Udemy FC",
  "ResultDetails": "Thinkific FC won. Final Score 3-0"
}
```

Response:

```
{
  "gameId": "9367eea1-c217-4393-a94d-c9d66e2643b8",
  "winnerTeamId": "1e289afd-1e03-41d3-9ef2-c8f6bf69e12c",
  "loserTeamId": "3ff6a229-9b2d-4f61-9689-12792f12128f",
}
```

```

    "resultDetails": "Thinkific FC won. Final Score 3-0",
    "managerId": "cc88be75-8e08-4dc2-8941-611984e69542"
  }

```

- GetScheduleWithResults endpoint- // returns all the schedules and corresponding results (if any was posted) created by that manager (based on the auth token)

url- <http://sportsmanagementapi-dev.ca-central-1.elasticbeanstalk.com/api/sportmanagement/getSchedulesWithResults>

Request Example:

Headers

Authorization : Bearer TOKEN (replace with the actual access token here)

```

[
  {
    "gameId": "9367eea1-c217-4393-a94d-c9d66e2643b8",
    "homeTeamId": "1e289afd-1e03-41d3-9ef2-c8f6bf69e12c",
    "homeTeamName": "Thinkific FC",
    "awayTeamId": "3ff6a229-9b2d-4f61-9689-12792f12128f",
    "awayTeamName": "Udemy FC",
    "scheduledTime": "2021-04-21T23:25:43+00:00",
    "managerId": "cc88be75-8e08-4dc2-8941-611984e69542",
    "result": {
      "gameId": "9367eea1-c217-4393-a94d-c9d66e2643b8",
      "winnerTeamId": "1e289afd-1e03-41d3-9ef2-c8f6bf69e12c",
      "loserTeamId": "3ff6a229-9b2d-4f61-9689-12792f12128f",
      "resultDetails": "Thinkific FC won. Final Score 3-0",
      "managerId": "cc88be75-8e08-4dc2-8941-611984e69542"
    }
  }
]

```

- patchPlayer endpoint

url - <http://sportsmanagementapi-dev.ca-central-1.elasticbeanstalk.com/api/sportmanagement/patchPlayer/1b77a16a-4047-4146-ae6b-bbe63fbedcc5> // the id at the end is the player Id managed by the the manager requesting to update the player details)

Request Example:

Headers

Authorization : Bearer TOKEN (replace with the actual access token here)

Body : // the player details looks like below -

```

public Guid Id { get; set; }
public string Name { get; set; }
public Guid TeamId { get; set; }
public Guid ManagerId { get; set; }
public string Details { get; set; }

```

In the below body example, you can add the above properties in the path(i.e, Details) and in the value section add the value that you wanted to update with and the `op` will be replace and the following will be updated. At the moment, the no ID's are allowed to be changed (Id, TeamId, ManagerId).

```

[
  {

```

```

    "value": "Midfielder. Age 25. Player rating 8 out 10",
    "path": "/details",
    "op": "replace"
  }
]

```

Response:

204 no content. // Validate by calling get players endpoint.

- **PatchSchedule Endpoint-**

url - <http://sportsmanagementapi-dev.ca-central-1.elasticbeanstalk.com/api/sportmanagement/patchSchedule/9367eea1-c217-4393-a94d-c9d66e2643b8> // the id at the end is the game Id created by the manager requesting to update the game/schedule details)

Request Example:

Headers

Authorization : Bearer TOKEN (replace with the actual access token here)

Body : // the schedule looks like below -

```

public Guid Gameld { get; set; }
public Guid HomeTeamId { get; set; }
public string HomeTeamName { get; set; }
public Guid AwayTeamId { get; set; }
public string AwayTeamName { get; set; }
public DateTime ScheduledTime { get; set; }
public Guid ManagerId { get; set; }
public string ScheduleDetails { get; set; }

```

In the below body example, you can add the above properties in the path(i.e, ScheduleDetails) and in the value section add the value that you wanted to update with and the `op` will be replace and the property in the Path will be updated. At the moment, the no Gameld and ManagerId are not allowed to be changed (Id, TeamId, ManagerId) and also for past games (scheduledTime is older than current time) you can't update TeamIds, ScheduleTime information (scheduleDetails can be updated for past games).

```

[
  {
    "value": "this was a semi final game",
    "path": "/ScheduleDetails",
    "op": "replace"
  }
]

```

Response:

204 no content. // Validate by calling getScheduleAndResults endpoint.

- **PatchResult Endpoint**

url - <http://sportsmanagementapi-dev.ca-central-1.elasticbeanstalk.com/api/sportmanagement/patchResult/9367eea1-c217-4393-a94d-c9d66e2643b8> // the id at the end is the game Id created by the manager requesting to update the result details)

Request Example:

Headers

Authorization : Bearer TOKEN (replace with the actual access token here)

Body : // the result looks like below -

```
public Guid Gameld { get; set; }
public Guid WinnerTeamId { get; set; }
public Guid LoserTeamId { get; set; }
public string ResultDetails { get; set; }
public Guid ManagerId { get; set; }
```

In the below body example, you can add the above properties in the path(i.e, ResultDetails) and in the value section add the value that you wanted to update with and the `op` will be replace and the property in the Path will be updated. No ID's are allowed to be updated for a result (as it's already posted). Only Result details can be updated at the moment.

```
[
  {
    "value": "Thinkific FC won.Final Score 3-1",
    "path": "/resultDetails",
    "op": "replace"
  }
]
```

Response:

204 no content. // Validate by calling getResults endpoint.

deleteSchedule , getResults endpoints are similar as above (like deletePlayer (put a valid gameld in the query string for deleteschedule and getPlayers (put a valid token for the getResults which will return all the results created by that manager)

Your feedback on this technical challenge

I enjoyed working on this project very much. I wasn't too familiar with the technical implementation of the bearer token based authentication so learning and implementing it was fun. Definitely one of my better technical interview experiences.