

README FILE

1. Installed Dependencies:

- Using Spring Boot Initializer to create a Spring Boot application with Java.
- JDK 17.
- Apache Maven 3.9.9.

2. Applications Used:

- **Visual Studio Code** to implement the application.
- **Postman** to test the entry point APIs.
- **MongoDB Compass** for creating data models from the application (Note: I used MySQL, but had many issues when trying to use it).

3. Used Port:

- I use **8081** because **8080** is already in use.

4. To Run the Application:

- Set up the application configuration in `Champion\src\main\resources\application.properties`, such as:
`spring.data.mongodb.uri=mongodb://localhost:27017/Champion`
`spring.data.mongodb.database=Champion`
`spring.datasource.username=sa`
`spring.datasource.password=`
- Open a command prompt in the application path and type the following command:
`mvnw.cmd spring-boot:run`
- Test if the application is running using the browser at:
<http://localhost:8081>.

To ensure that the application is running without issues, you should receive the following message:

Welcome to the Chess League API!

5. Add Dependencies:

- Add the necessary dependencies required to implement functionalities in `pom.xml`.

6. Unit Testing:

- Add unit testing for the match service and participant service.

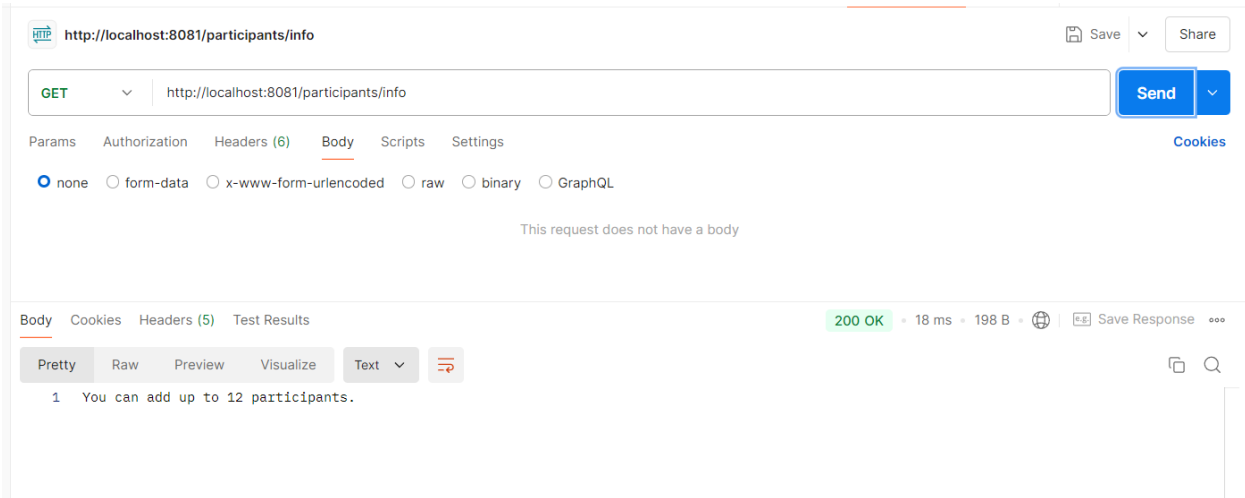
7. The League (Bonus):

- Point-2 Submit league champion

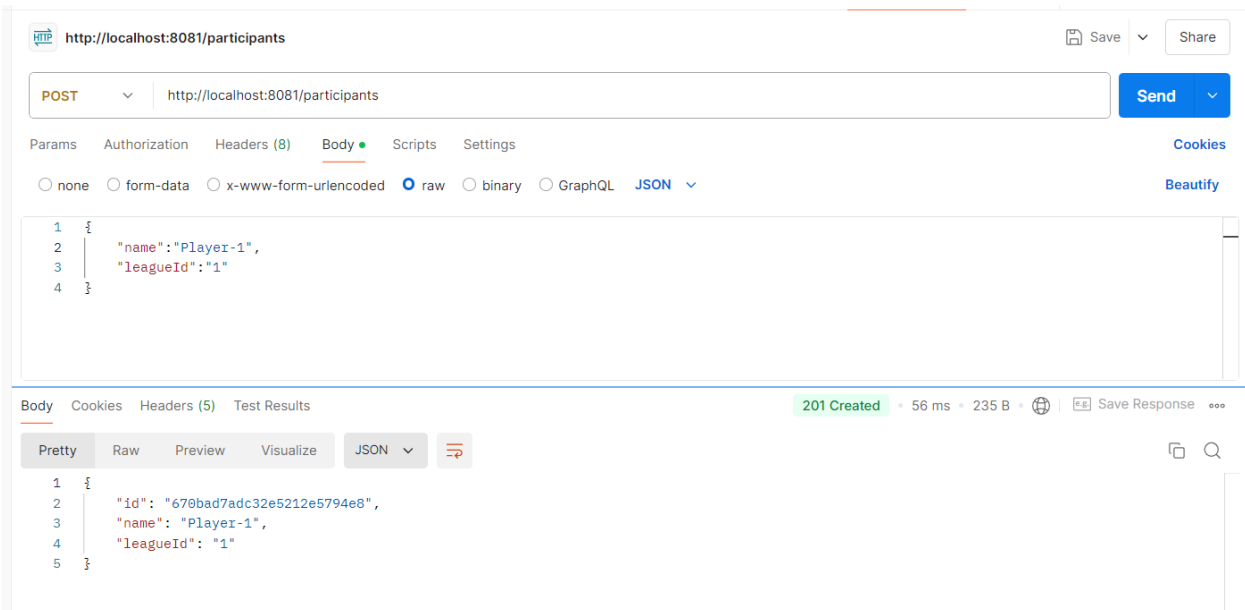
Test APIs on Postman:

1- Participants

- Get Information before starting
 - **Method** : GET
 - **Entrypoint** : http://localhost:8081/participants/info
 - **Expected Result** :



- Submit a Participant Request (Max is 12 participants)
 - **Method** : POST
 - **Entrypoint** : http://localhost:8081/participants
 - **Expected Result** :



You can create up to 12 participants.

The screenshot shows a REST client interface with the URL `http://localhost:8081/participants`. The request method is `POST`. The request body is a JSON object: `{ "name": "Player-12", "leagueId": "1" }`. The response status is `201 Created`, with a response time of `13 ms` and a size of `236 B`. The response body is a JSON object: `{ "id": "670bae5fdc32e5212e5794f3", "name": "Player-12", "leagueId": "1" }`.

```
1 {
2   "name": "Player-12",
3   "leagueId": "1"
4 }
```

```
1 {
2   "id": "670bae5fdc32e5212e5794f3",
3   "name": "Player-12",
4   "leagueId": "1"
5 }
```

If you try to create another participant after reaching the limit of 12, you will receive:

The screenshot shows the same REST client interface, but the response status is `400 Bad Request`, with a response time of `18 ms` and a size of `187 B`. The response body is a text message: `Error: Maximum number of participants is 12`.

```
1 {
2   "name": "Player-12",
3   "leagueId": "1"
4 }
```


```
1 Error: Maximum number of participants is 12
```

- Get a List of All Participants
 - **Method** : GET
 - **Entrypoint** : http://localhost:8081/participants
 - **Expected Result** :

The screenshot shows a REST client interface with the following details:

- URL:** http://localhost:8081/participants
- Method:** GET
- Body:** This request does not have a body.
- Response:** 200 OK, 19 ms, 972 B.
- Response Format:** JSON (Pretty view).
- Response Data:**

```
1  [
2    {
3      "id": "670bad7adc32e5212e5794e8",
4      "name": "Player-1",
5      "leagueId": "1"
6    },
7    {
8      "id": "670bae32dc32e5212e5794e9",
9      "name": "Player-2",
10     "leagueId": "1"
11    },
12    {
13      "id": "670bae37dc32e5212e5794ea",
14      "name": "Player-3",
15      "leagueId": "1"
16    },
17    {
18      "id": "670bae3bdc32e5212e5794eb",
19      "name": "Player-4",
20      "leagueId": "1"
21    }
22  ],
```

 http://localhost:8081/participants

GET

http://localhost:8081/participants

Send

ParamsAuthorizationHeaders (6)BodyScriptsSettings

☒ none

☐ form-data

☐ x-www-form-urlencoded

☐ raw

☐ binary

☐ GraphQL


This request does not have a body


BodyCookiesHeaders (5)Test Results


200 OK

19 ms


972 B




 Save Response



PrettyRawPreviewVisualizeJSON





```
22  {
23    "id": "670bae3fdc32e5212e5794ec",
24    "name": "Player-5",
25    "leagueId": "1"
26  },
27  {
28    "id": "670bae44dc32e5212e5794ed",
29    "name": "Player-6",
30    "leagueId": "1"
31  },
32  {
33    "id": "670bae49dc32e5212e5794ee",
34    "name": "Player-7",
35    "leagueId": "1"
36  },
37  {
38    "id": "670bae4ddc32e5212e5794ef",
39    "name": "Player-8",
40    "leagueId": "1"
41  },
```

 http://localhost:8081/participants

GET

http://localhost:8081/participants

Send

ParamsAuthorizationHeaders (6)BodyScriptsSettings

☒ none

☐ form-data

☐ x-www-form-urlencoded

☐ raw

☐ binary

☐ GraphQL

This request does not have a body

BodyCookiesHeaders (5)Test Results

200 OK

19 ms

972 B



 Save Response



PrettyRawPreviewVisualizeJSON





```
42  {
43    "id": "670bae52dc32e5212e5794f0",
44    "name": "Player-9",
45    "leagueId": "1"
46  },
47  {
48    "id": "670bae56dc32e5212e5794f1",
49    "name": "Player-10",
50    "leagueId": "1"
51  },
52  {
53    "id": "670bae5bdc32e5212e5794f2",
54    "name": "Player-11",
55    "leagueId": "1"
56  },
57  {
58    "id": "670bae5fdc32e5212e5794f3",
59    "name": "Player-12",
60    "leagueId": "1"
61  }
62  ]
```

Postbot

 Runner

 Start Proxy

 Cookies

 Vault


 Trash

- Group Randomly Participants into (n) Groups
 - **Method :** POST
 - **Entrypoint :**
http://localhost:8081/participants/group?numberOfGroups=4
 - **Expected Result :**

The screenshot displays a REST client interface with the following details:

- URL:** http://localhost:8081/participants/group?numberOfGroups=4
- Method:** POST
- Body:** This request does not have a body
- Response:** 200 OK, 24 ms, 1020 B
- Response Body (JSON):**

```
[{"Group 1": [{"id": "670bae32dc32e5212e5794e9", "name": "Player-2", "leagueId": "1"}, {"id": "670bae37dc32e5212e5794ea", "name": "Player-3", "leagueId": "1"}, {"id": "670bad7adc32e5212e5794e8", "name": "Player-1", "leagueId": "1"}]}]
```

 http://localhost:8081/participants/group?numberOfGroups=4

SaveShare

POSThttp://localhost:8081/participants/group?numberOfGroups=4Send

ParamsAuthorizationHeaders (7)BodyScriptsSettingsCookies


☒ none ☐ form-data ☐ x-www-form-urlencoded ☐ raw ☐ binary ☐ GraphQL

This request does not have a body

BodyCookiesHeaders (5)Test Results200 OK • 24 ms • 1020 B • Save Response

PrettyRawPreviewVisualizeJSON

```
1 {
2   "Group 2": [
3     {
4       "id": "670bae56dc32e5212e5794f1",
5       "name": "Player-10",
6       "leagueId": "1"
7     },
8     {
9       "id": "670bae49dc32e5212e5794ee",
10      "name": "Player-7",
11      "leagueId": "1"
12    },
13    {
14      "id": "670bae44dc32e5212e5794ed",
15      "name": "Player-6",
16      "leagueId": "1"
17    }
18  ],
```

 http://localhost:8081/participants/group?numberOfGroups=4

SaveShare

POSThttp://localhost:8081/participants/group?numberOfGroups=4Send

ParamsAuthorizationHeaders (7)BodyScriptsSettingsCookies

☒ none ☐ form-data ☐ x-www-form-urlencoded ☐ raw ☐ binary ☐ GraphQL

This request does not have a body

BodyCookiesHeaders (5)Test Results200 OK • 24 ms • 1020 B • Save Response

PrettyRawPreviewVisualizeJSON

```
53 "Group 3": [
54   {
55     "id": "670bae3fdc32e5212e5794ec",
56     "name": "Player-5",
57     "leagueId": "1"
58   },
59   {
60     "id": "670bae5fdc32e5212e5794f3",
61     "name": "Player-12",
62     "leagueId": "1"
63   },
64   {
65     "id": "670bae3bdc32e5212e5794eb",
66     "name": "Player-4",
67     "leagueId": "1"
68   }
69 ]
```

REST client interface showing a POST request to `http://localhost:8081/participants/group?numberOfGroups=4`. The response is a JSON array of 3 player objects for Group 4.

```
36 "Group 4": [  
37   {  
38     "id": "670bae52dc32e5212e5794f0",  
39     "name": "Player-9",  
40     "leagueId": "1"  
41   },  
42   {  
43     "id": "670bae5bdc32e5212e5794f2",  
44     "name": "Player-11",  
45     "leagueId": "1"  
46   },  
47   {  
48     "id": "670bae4ddc32e5212e5794ef",  
49     "name": "Player-8",  
50     "leagueId": "1"  
51   }  
52 ],
```

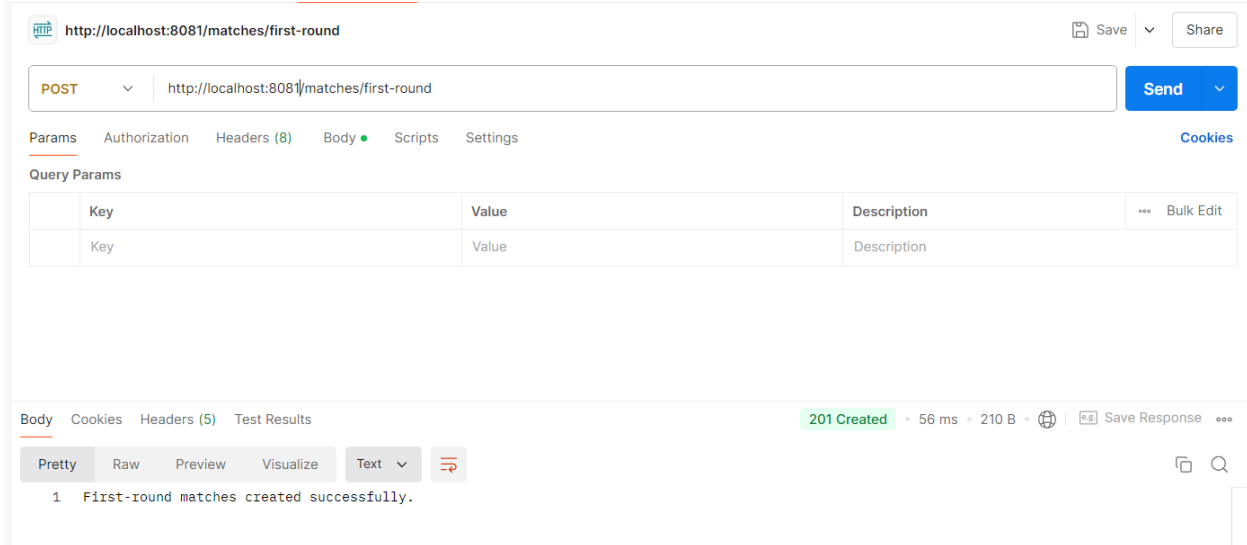
2- Matches

- Get Information Before Starting
 - **Method** : GET
 - **Entrypoint** : `http://localhost:8081/matches/info`
 - **Expected Result** :

REST client interface showing a GET request to `http://localhost:8081/matches/info`. The response is a text message.

```
1 You can create up to 3 matches in a day by requesting http://localhost:8081/matches/first-round the 3 matches will created automatically.
```


- Get List of All Automatically Created First-Round Matches
 - **Method :** POST
 - **Entrypoint :** http://localhost:8081/matches/first-round
 - (This endpoint will create only 3 matches per day automatically)
 - **Expected Result :**



http://localhost:8081/matches/first-round

POST http://localhost:8081/matches/first-round

Send

Params Authorization Headers (8) Body Scripts Settings Cookies

Query Params

Key	Value	Description	Bulk Edit
Key	Value	Description	

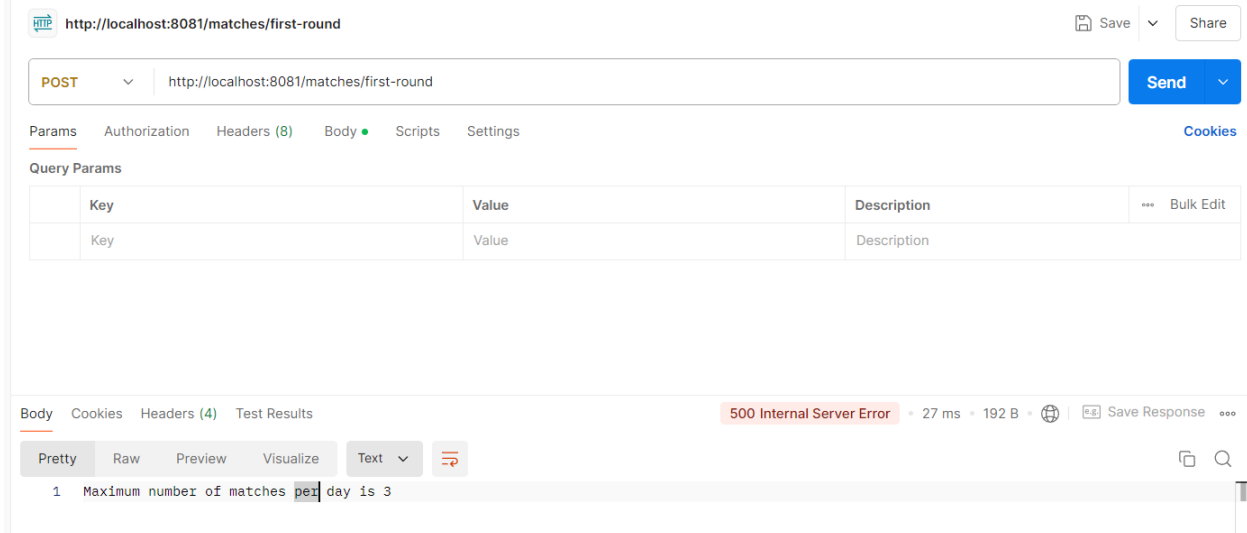
Body Cookies Headers (5) Test Results

201 Created • 56 ms • 210 B • Save Response

Pretty Raw Preview Visualize Text

1 First-round matches created successfully.

If you try to create more matches using `http://localhost:8081/matches/first-round`, you will receive:



http://localhost:8081/matches/first-round

POST http://localhost:8081/matches/first-round

Send

Params Authorization Headers (8) Body Scripts Settings Cookies

Query Params

Key	Value	Description	Bulk Edit
Key	Value	Description	


Body Cookies Headers (4) Test Results

500 Internal Server Error • 27 ms • 192 B • Save Response

Pretty Raw Preview Visualize Text

1 Maximum number of matches per day is 3

- Retrieve All Created Matches
 - **Method :** GET
 - **Entrypoint :** http://localhost:8081/matches
 - **Expected Result :**

 http://localhost:8081/matches

SaveShare

GEThttp://localhost:8081/matchesSend

ParamsAuthorizationHeaders (6)BodyScriptsSettingsCookies


☒ none☐ form-data☐ x-www-form-urlencoded☐ raw☐ binary☐ GraphQL

This request does not have a body

BodyCookiesHeaders (5)Test Results200 OK35 ms807 BSave Response

PrettyRawPreviewVisualizeJSON

```
1  [
2    {
3      "id": "670bb3c0dc32e5212e5794f4",
4      "participantOneId": "670bae4ddc32e5212e5794ef",
5      "participantTwoId": "670bad7adc32e5212e5794e8",
6      "winnerId": null,
7      "leagueId": "1234",
8      "result": "Pending",
9      "closed": false,
10     "date": "2024-10-13"
11   },
12   {
13     "id": "670bb3c0dc32e5212e5794f5",
14     "participantOneId": "670bae3bdc32e5212e5794eb",
15     "participantTwoId": "670bae3fdc32e5212e5794ec",
16     "winnerId": null,
17     "leagueId": "1234",
18     "result": "Pending",
19     "closed": false,
20     "date": "2024-10-13"
21   },
22 ]
```

 http://localhost:8081/matches

SaveShare

GEThttp://localhost:8081/matchesSend

ParamsAuthorizationHeaders (6)BodyScriptsSettingsCookies

☒ none☐ form-data☐ x-www-form-urlencoded☐ raw☐ binary☐ GraphQL

This request does not have a body

BodyCookiesHeaders (5)Test Results200 OK35 ms807 BSave Response

PrettyRawPreviewVisualizeJSON

```
22  [
23    {
24      "id": "670bb3c0dc32e5212e5794f6",
25      "participantOneId": "670bae52dc32e5212e5794f0",
26      "participantTwoId": "670bae44dc32e5212e5794ed",
27      "winnerId": null,
28      "leagueId": "1234",
29      "result": "Pending",
30      "closed": false,
31      "date": "2024-10-13"
32    }
33  ]
```

- Update Match Winner and Results / Close Round

- **Method :** PUT
- **Entrypoint :** <http://localhost:8081/matches/{matchId}/result>
- Ex:
<http://localhost:8081/matches/670bb3c0dc32e5212e5794f5/result>

- **Expected Result :**

When updating the match to set the winner and results, the match will close automatically, and the close field will be set to true.

The screenshot displays a REST client interface with the following details:

- URL:** `http://localhost:8081/matches/670bb3c0dc32e5212e5794f5/result`
- Method:** PUT
- Body (Request):**

```
1 {
2   "winnerId": "670bae4ddc32e5212e5794ef",
3   "result": "2-1"
4 }
```
- Response:** 200 OK, 221 ms, 394 B
- Body (Response):**

```
1 {
2   "id": "670bb3c0dc32e5212e5794f5",
3   "participantOneId": "670bae3bdc32e5212e5794eb",
4   "participantTwoId": "670bae3fdc32e5212e5794ec",
5   "winnerId": "670bae4ddc32e5212e5794ef",
6   "leagueId": "1234",
7   "result": "2-1",
8   "closed": true,
9   "date": "2024-10-13"
10 }
```

3- The League | BONUS

- Submit League Champion

- **Method** : POST
- **Entrypoint** : `http://localhost:8081/league/submit-champion`
- **Expected Result**:

The screenshot shows a REST client interface with the URL `http://localhost:8081/api/league/submit-champion`. The method is set to **POST**. The body is a JSON object: `{ "championId": "670be2dc2f95241f246d66bc" }`. The response status is **200 OK** with a response time of 36 ms and a body size of 203 B. The response body is `1 League champion submitted successfully.`

To run unit tests, navigate to the project path in the command prompt and execute the following command:

```
statistics: [DefaultContextCache@2fedfff1 size = 1, maxSize = 32, parentContextCount = 0, hitCount = 10, missCount = 1, failureCount = 0]
2024-10-13T15:08:51.010+03:00 DEBUG 13308 --- [Champion] [main] org.springframework.test.context.cache : Spring test ApplicationContext cache
statistics: [DefaultContextCache@2fedfff1 size = 1, maxSize = 32, parentContextCount = 0, hitCount = 11, missCount = 1, failureCount = 0]
2024-10-13T15:08:51.011+03:00 DEBUG 13308 --- [Champion] [main] .c.s.DirtiesContextTestExecutionListener : After test method: class [DemoApplica
tionTests], method [contextLoads], class annotated with @DirtiesContext [false] with mode [null], method annotated with @DirtiesContext [false] with mode [
null]
2024-10-13T15:08:51.011+03:00 DEBUG 13308 --- [Champion] [main] o.s.t.c.w.ServletTestExecutionListener : Resetting RequestContextHolder for te
st class com.example.Champion.DemoApplicationTests
2024-10-13T15:08:51.029+03:00 DEBUG 13308 --- [Champion] [main] org.springframework.test.context.cache : Spring test ApplicationContext cache
statistics: [DefaultContextCache@2fedfff1 size = 1, maxSize = 32, parentContextCount = 0, hitCount = 12, missCount = 1, failureCount = 0]
2024-10-13T15:08:51.030+03:00 DEBUG 13308 --- [Champion] [main] .c.s.DirtiesContextTestExecutionListener : After test class: class [DemoApplicat
ionTests], class annotated with @DirtiesContext [false] with mode [null]
[INFO] Tests run: 1, Failures: 0, Errors: 0, Skipped: 0, Time elapsed: 10.19 s -- in com.example.Champion.DemoApplicationTests
[INFO] Running com.example.Champion.service.MatchServiceTest
[INFO] Tests run: 7, Failures: 0, Errors: 0, Skipped: 0, Time elapsed: 0.541 s -- in com.example.Champion.service.MatchServiceTest
[INFO] Running com.example.Champion.service.ParticipantServiceTest
[INFO] Tests run: 5, Failures: 0, Errors: 0, Skipped: 0, Time elapsed: 0.041 s -- in com.example.Champion.service.ParticipantServiceTest
[INFO] Results:
[INFO] Tests run: 13, Failures: 0, Errors: 0, Skipped: 0
[INFO]
[INFO] BUILD SUCCESS
[INFO]
[INFO] Total time: 16.261 s
[INFO] Finished at: 2024-10-13T15:08:51+03:00
[INFO]
```