



PONTIFÍCIA UNIVERSIDADE CATÓLICA DE MINAS GERAIS
Programa de Pós-Graduação

Implementação prática de uma API REST

APIs e Web Services (AWS)

Grupo:

Eric Jangola

Julia Casseb Maciel

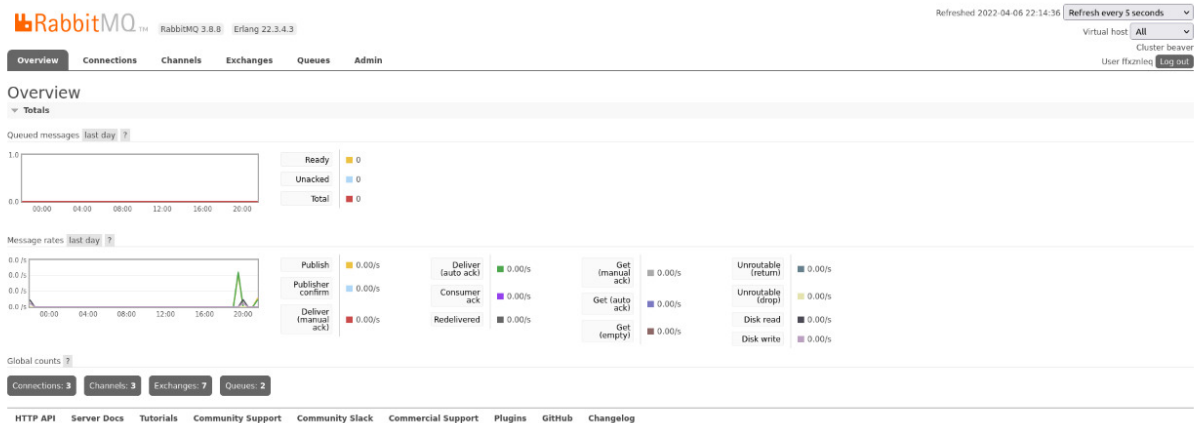
Michael Haruo Nakashima

MECANISMOS EM EXECUÇÃO

1- Chaves salvas no REDIS para Cache dos comandos

```
127.0.0.1:6379> keys *
1) "playerMurilAOS\xc3\xa3o Paulo"
2) "tournamentLibertadores2015-03-24"
3) "transferundefinedWed Apr 06 2022 20:00:26 GMT-0300"
4) "teamatl\xc3\xa9tico mineiro"
5) "tournamentCopa Brasil2015-03-24"
```

2- Fila Rabbit MQ executando as a service

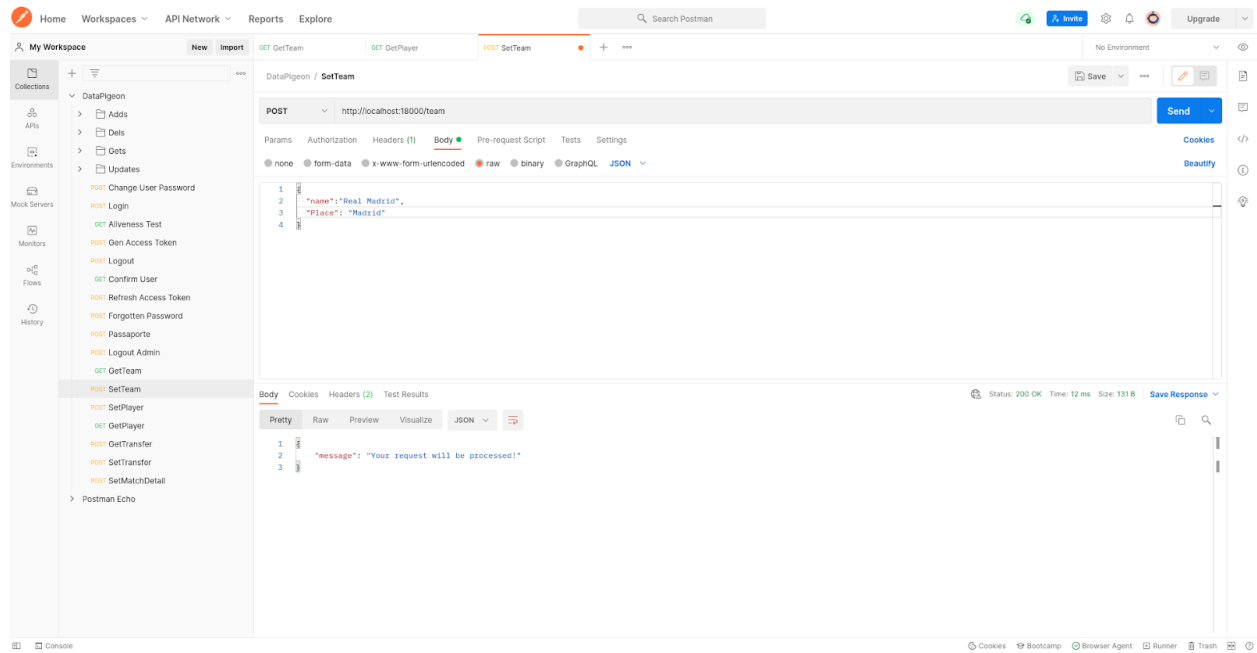


CHAMADAS DA API USANDO O POSTMAN

Team

1- POST : Set Team (Success)

Comando para registro de um novo time.



2- GET: Get All Teams (Success)

Comando para obter todos os times cadastrados

GET http://localhost:18000/team Send

Params Authorization Headers (6) Body Pre-request Script Tests Settings Cookies

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

This request does not have a body

Body Cookies Headers (8) Test Results Status: 200 OK Time: 38 ms Size: 594 B Save Response

Pretty Raw Preview Visualize

```
[{"id":1,"name":"cruzeiro","place":"mineirão","createdAt":"2022-04-07T02:56:23.000Z","updatedAt":"2022-04-07T02:56:23.000Z"}, {"id":2,"name":"atletico mineiro","place":"arena do galo","createdAt":"2022-04-07T02:56:44.000Z","updatedAt":"2022-04-07T02:56:44.000Z"}, {"id":3,"name":"america mineiro","place":"independencia","createdAt":"2022-04-07T02:57:53.000Z","updatedAt":"2022-04-07T02:57:53.000Z"}]
```

3- GET: Get Teams By Id (Success)

Comando para obter um time com um id específico

GET http://localhost:18000/team/2 Send

Params Authorization Headers (6) Body Pre-request Script Tests Settings Cookies

Query Params

KEY	VALUE	DESCRIPTION	...	Bulk Edit
Key	Value	Description		

Body Cookies Headers (8) Test Results Status: 200 OK Time: 26 ms Size: 377 B Save Response

Pretty Raw Preview Visualize JSON

```
1 {
2   "id": 2,
3   "name": "atletico mineiro",
4   "place": "arena do galo",
5   "createdAt": "2022-04-07T02:47:38.000Z",
6   "updatedAt": "2022-04-07T02:47:38.000Z"
7 }
8
9 }
```

4- DELETE: Remove Team (Success)

Remover um time com um id específico

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

- URL:** `http://localhost:18000/team/1`
- Method:** `DELETE`
- Send Button:** A blue button labeled "Send".
- Params:** A tab labeled "Params" is selected, showing a table with columns: KEY, VALUE, DESCRIPTION, and Bulk Edit. The table contains one row: Key, Value, Description.
- Body:** A tab labeled "Body" is selected, showing a JSON response: `{ "message": "Team deleted" }`.
- Status:** 200 OK, Time: 11 ms, Size: 293 B.
- Save Response:** A button labeled "Save Response".

Player

5- POST: Set Player (Success)

Registro de um novo jogador

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

- URL:** `http://localhost:18000/player`
- Method:** `POST`
- Send Button:** A blue button labeled "Send".
- Params:** A tab labeled "Params" is selected, showing a table with columns: KEY, VALUE, DESCRIPTION, and Bulk Edit. The table contains one row: Key, Value, Description.
- Body:** A tab labeled "Body" is selected, showing a JSON request: `{ "name": "Murilo", "birthplace": "São Paulo", "country": "Brasil" }`.
- Status:** 200 OK, Time: 97 ms, Size: 295 B.
- Save Response:** A button labeled "Save Response".

6- POST: Set Player (Failure)

Falha na tentativa de registrar um jogador com o mesmo nome de um jogador já registrado

The screenshot shows the Postman interface for a POST request to `http://localhost:18000/player`. The request body is a JSON object:

```
{
  "name": "MurilAO",
  "birthplace": "São Paulo",
  "country": "Brasil"
}
```

The response is a 500 Internal Server Error with a status of 500, time of 1612 ms, and size of 274 B. The response body is an HTML document with the following content:

```
<!DOCTYPE html>
<html lang="en">
  <head>
    <meta charset="utf-8">
    <title>Error</title>
  </head>
  <body>
    <p>Error: "MurilAO" is already registered</p>
  </body>
</html>
```

7- GET: Get All Players (Success)

Obter todos os jogadores registrados

The screenshot shows the Postman interface for a GET request to `http://localhost:18000/player`. The response is a 200 OK status with a status of 200, time of 23 ms, and size of 597 B. The response body is a JSON array of two player objects:

```
[
  {
    "id": 1,
    "name": "MurilAO",
    "birthplace": "São Paulo",
    "country": "Brasil",
    "createdAt": "2022-04-07T03:12:24.000Z",
    "updatedAt": "2022-04-07T03:12:24.000Z",
    "teamId": null
  },
  {
    "id": 2,
    "name": "Hulk",
    "birthplace": "Campina Grande",
    "country": "Brasil",
    "createdAt": "2022-04-07T03:14:01.000Z",
    "updatedAt": "2022-04-07T03:14:01.000Z",
    "teamId": null
  }
]
```

8- GET: Get Player By Id (Success)

Obter um jogador registrado com um id específico

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

- Method:** GET
- URL:** http://localhost:18000/player/2
- Buttons:** Send, Cookies
- Body:** none
- Status:** 200 OK
- Time:** 16 ms
- Size:** 433 B
- Save Response:** (button)
- Body Content (JSON):**

```
{  "id": 2,  "name": "Hulk",  "birthplace": "Campina Grande",  "country": "Brasil",  "createdAt": "2022-04-07T03:14:01.000Z",  "updatedAt": "2022-04-07T03:14:01.000Z",  "teamId": null}
```

9- DELETE: Remove Player (Success)

Remover um jogador registrado com um id específico

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

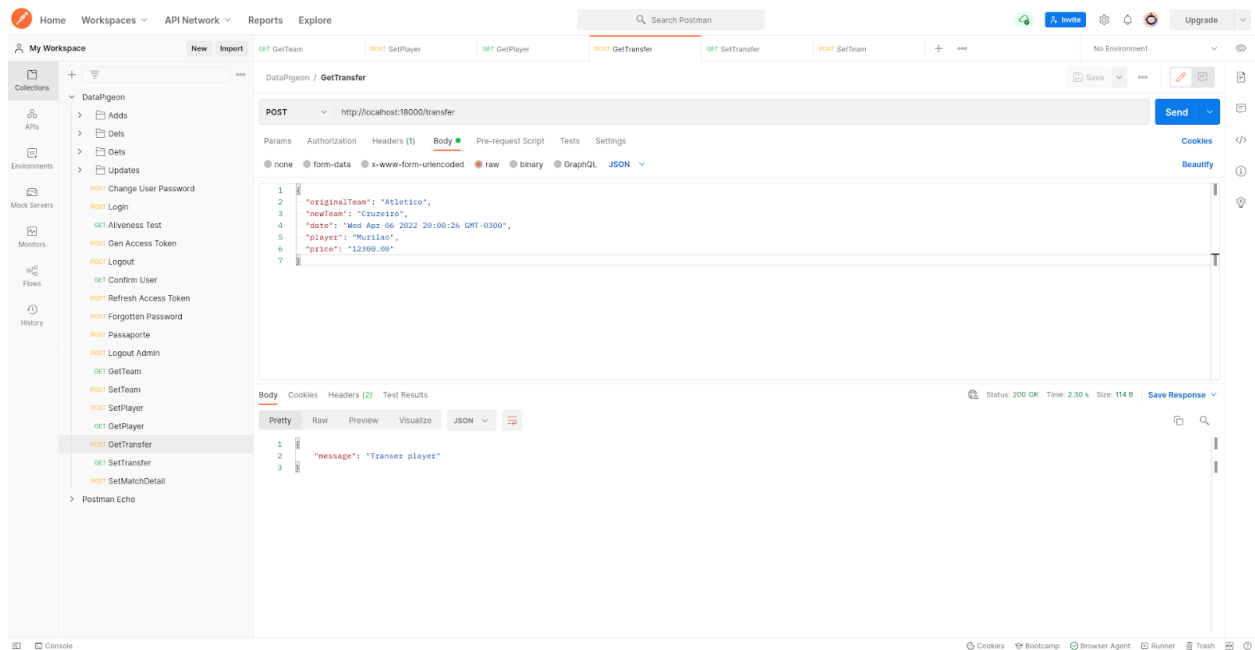
- Method:** DELETE
- URL:** http://localhost:18000/player/1
- Buttons:** Send, Cookies
- Body:** none
- Status:** 200 OK
- Time:** 22 ms
- Size:** 295 B
- Save Response:** (button)
- Body Content (JSON):**

```
{  "message": "Player deleted"}
```

Transfer

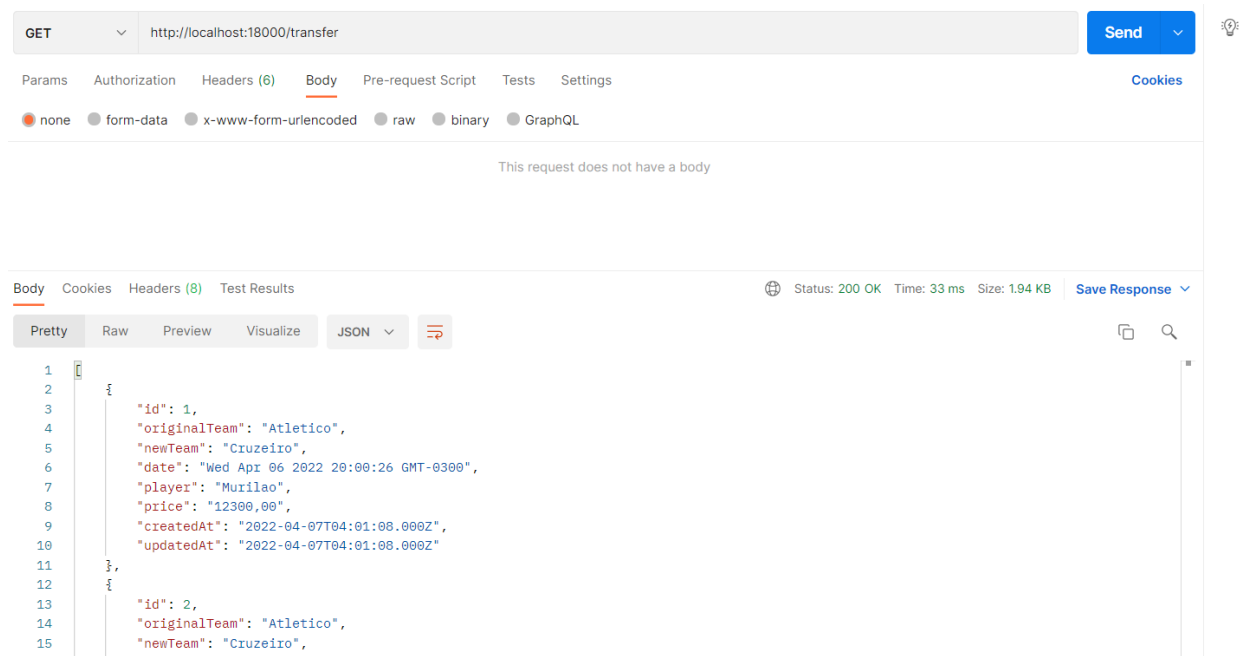
10- POST : Set Transfer (Success)

Registro de uma nova transferência



11- GET: Get All Transfers (Success)

Obter todas as transferências registradas



12- GET: Get Transfer By Id (Success)

Obter uma transferência com um id específico

GET Send

Params Authorization Headers (6) **Body** Pre-request Script Tests Settings Cookies

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

This request does not have a body

Body Cookies Headers (8) Test Results Status: 200 OK Time: 29 ms Size: 1.52 KB Save Response

Pretty Raw Preview Visualize JSON

```
1 {
2   "id": 3,
3   "originalTeam": "Atletico",
4   "newTeam": "Cruzeiro",
5   "date": "Wed Apr 06 2022 20:00:26 GMT-0300",
6   "player": "Murilao",
7   "price": "12300,00",
8   "createdAt": "2022-04-07T04:03:28.000Z",
9   "updatedAt": "2022-04-07T04:03:28.000Z"
10 }
11
```

13- DELETE: Remove Transfer (Success)

Remover uma transferência com um id específico

DELETE Send

Params Authorization **Headers (6)** Body Pre-request Script Tests Settings Cookies

Headers 6 hidden

KEY	VALUE	DESCRIPTION	...	Bulk Edit	Presets
Key	Value	Description			

Body Cookies Headers (8) Test Results Status: 200 OK Time: 7 ms Size: 297 B Save Response

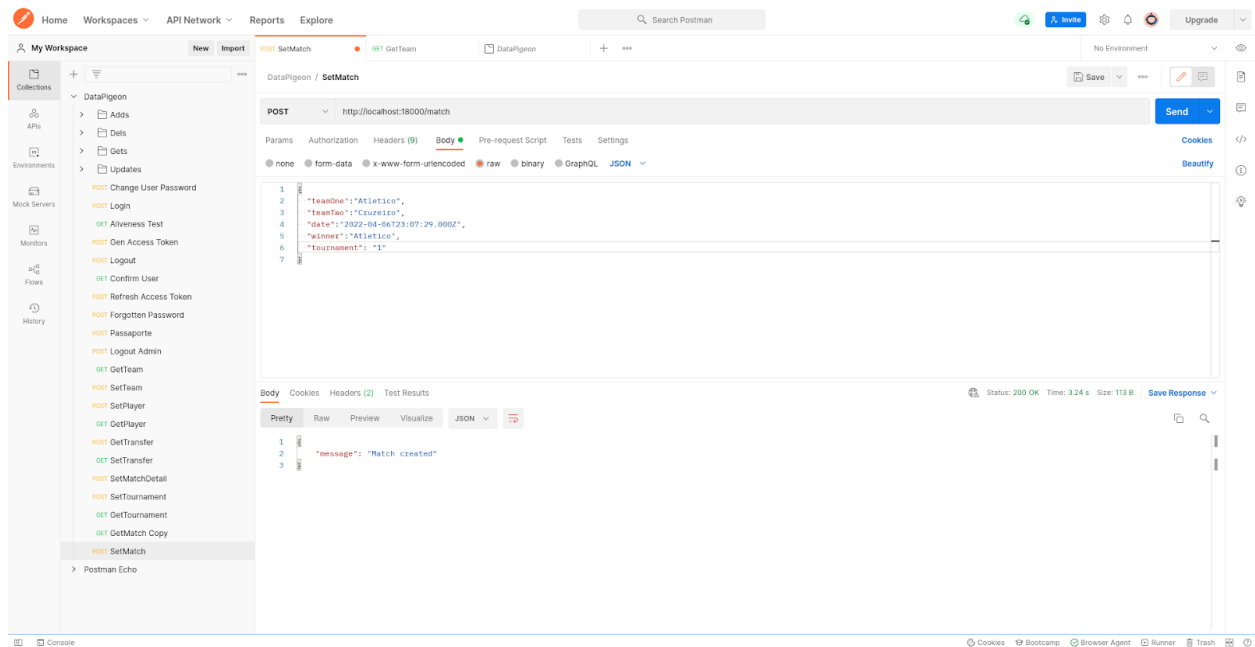
Pretty Raw Preview Visualize JSON

```
1 {
2   "message": "Transfer deleted"
3 }
```

Match

14- POST : Set Match(Success)

Registro de uma nova partida



The screenshot shows the Postman interface for a POST request to `http://localhost:18000/match`. The request body is a JSON object with the following structure:

```
{  "teamOne": "Atletico",  "teamTwo": "Cruzeiro",  "date": "2022-04-06T23:07:29.000Z",  "admission": "Atletico",  "tournament": "1"}
```

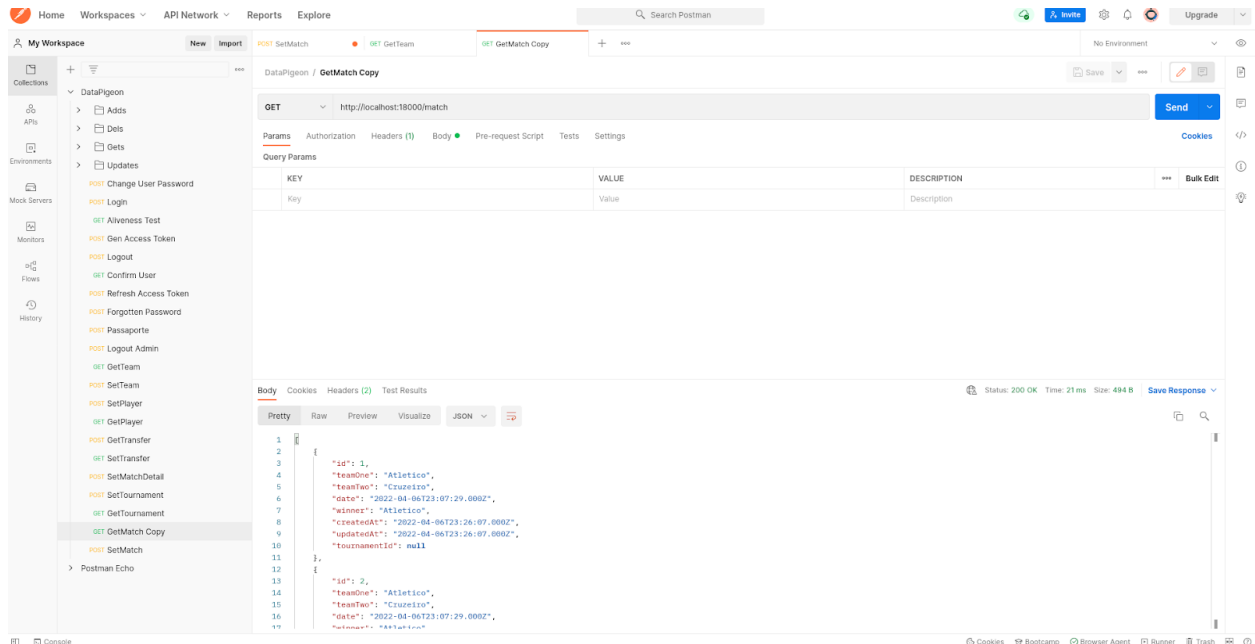
The response is a JSON object with a success message:

```
{  "message": "Match created"}
```

The status is 200 OK, Time: 3.24 s, Size: 113 B.

15- GET: Get All Match(Success)

Obter todas as partidas registradas



The screenshot shows the Postman interface for a GET request to `http://localhost:18000/match`. The response is a JSON array of match objects:

```
[  {    "id": 1,    "teamOne": "Atletico",    "teamTwo": "Cruzeiro",    "date": "2022-04-06T23:07:29.000Z",    "admission": "Atletico",    "createdAt": "2022-04-06T23:26:07.000Z",    "updatedAt": "2022-04-06T23:26:07.000Z",    "tournamentId": null  },  {    "id": 2,    "teamOne": "Atletico",    "teamTwo": "Cruzeiro",    "date": "2022-04-06T23:07:29.000Z",    "admission": "Atletico",    "createdAt": "2022-04-06T23:26:07.000Z",    "updatedAt": "2022-04-06T23:26:07.000Z",    "tournamentId": null  }]
```

The status is 200 OK, Time: 21 ms, Size: 494 B.

16- GET: Get Match By Id (Success)

Obter uma partida registrada com um id específico

GET http://localhost:18000/match/1 Send

Params Authorization Headers (6) **Body** Pre-request Script Tests Settings Cookies

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

This request does not have a body

Body Cookies Headers (8) Test Results Status: 200 OK Time: 42 ms Size: 477 B Save Response

Pretty Raw Preview Visualize JSON

```
1 {
2   "id": 1,
3   "teamOne": "Atletico",
4   "teamTwo": "Cruzeiro",
5   "date": "Wed Apr 06 2022 23:07:29.000Z",
6   "winner": "Atletico",
7   "createdAt": "2022-04-07T04:22:47.000Z",
8   "updatedAt": "2022-04-07T04:22:47.000Z",
9   "tournamentId": null
10 }
11
12
```

17- DELETE: Remove Match(Success)

Remover uma partida com um id específico

DELETE http://localhost:18000/match/2 Send

Params Authorization **Headers (6)** Body Pre-request Script Tests Settings Cookies

Headers 6 hidden

KEY	VALUE	DESCRIPTION	...	Bulk Edit	Presets
Key	Value	Description			

Body Cookies Headers (8) Test Results Status: 200 OK Time: 19 ms Size: 294 B Save Response

Pretty Raw Preview Visualize JSON

```
1 {
2   "message": "Match deleted"
3 }
```

Match Detail

18- POST : Set Match Detail (Success)

Registro de novos detalhes de uma partida

The screenshot shows a Postman interface for a POST request to `http://localhost:18000/match-detail`. The request body is a JSON object with the following fields:

```
{  "matchId": "1",  "start": "Wed Apr 06 2022 16:00:00",  "overtime": "3'",  "replacements": "5",  "warnings": "2",  "end": "Wed Apr 06 2022 18:06:02"}
```

The response is a JSON object with a success message:

```
{  "message": "Match detail created"}
```

The status is 200 OK, Time: 43 ms, Size: 301 B.

19- GET: Get All MatchDetail (Success)

Obter os detalhes de uma partida específica. Antes do Set /tournament/<id>/match/<id>/detail/start:

The screenshot shows a Postman interface for a GET request to `http://localhost:18000/match-detail`. The response is a JSON object with the following fields:

```
{  "id": "1",  "matchId": "1",  "start": null,  "interval": null,  "overtime": null,  "replacements": null,  "warnings": null,  "end": null,  "createdAt": "2022-04-06T23:35:37.000Z",  "updatedAt": "2022-04-06T23:35:37.000Z"}
```

The status is 200 OK, Time: 10 ms, Size: 291 B.

Depois:

The screenshot shows the Postman interface with a GET request to `http://localhost:18000/match-detail`. The response body is a JSON object with the following structure:

```
1 {
2   "id": 1,
3   "matchId": "1",
4   "start": "2022-04-06T22:44:33.000Z",
5   "interval": null,
6   "overtime": null,
7   "replacements": null,
8   "wastings": null,
9   "end": null,
10  "createAt": "2022-04-06T23:35:37.000Z",
11  "updateAt": "2022-04-06T23:50:59.000Z"
12 }
```

20- DELETE: Remove Match Detail (Success)

Remover os detalhes de uma partida sob um id específico

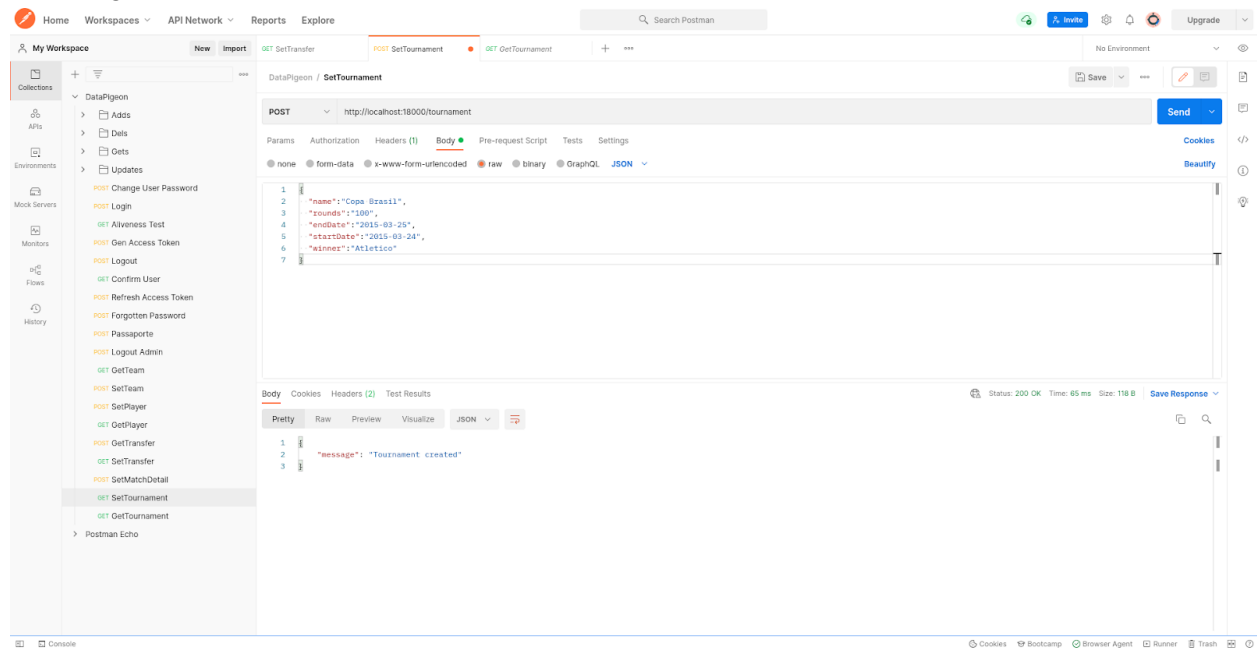
The screenshot shows the Postman interface with a DELETE request to `http://localhost:18000/match-detail/1`. The response body is a JSON object with the following structure:

```
1 {
2   "message": "Match detail deleted"
3 }
```

Tournament

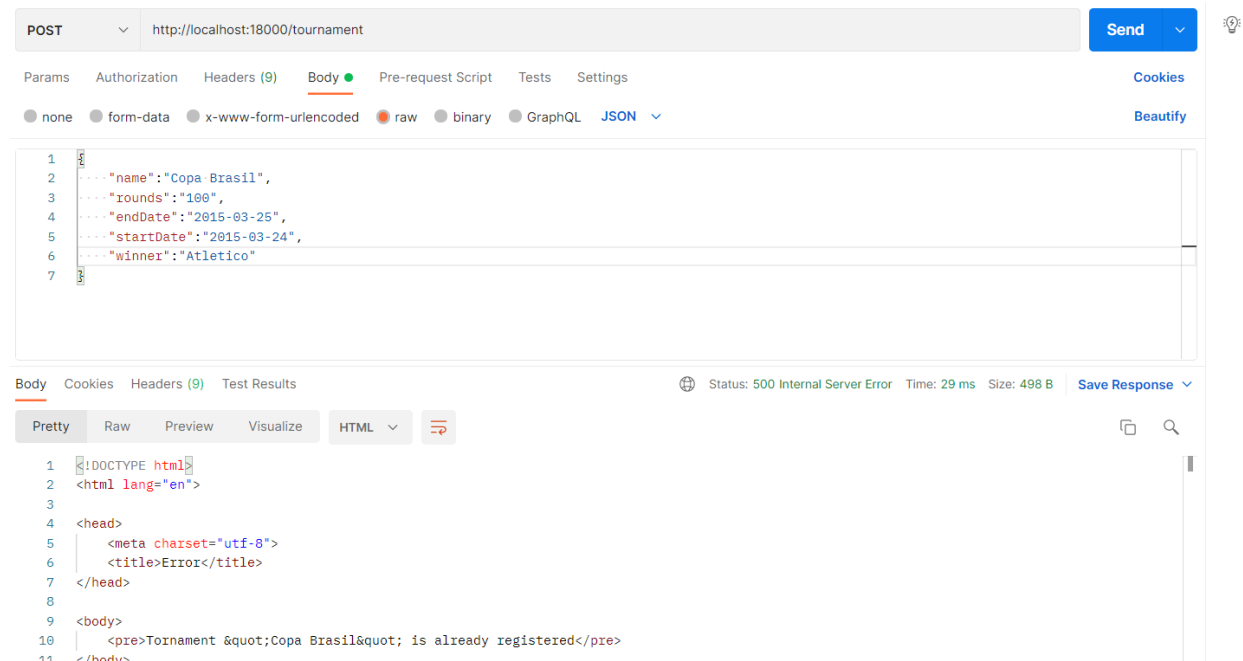
21- POST: Set Tournament (Success)

Novo registro de um torneio



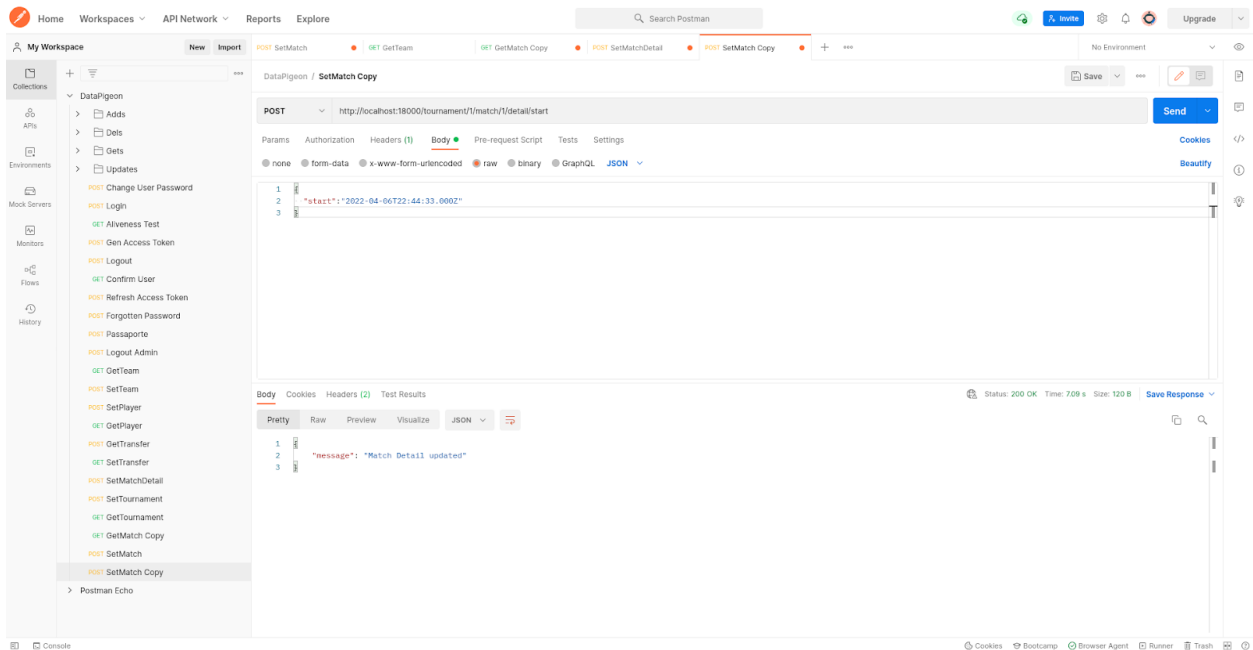
22- POST: Set Tournament (Failure)

Falha na tentativa de registro de um torneio com o mesmo nome de outro torneio ja registrado



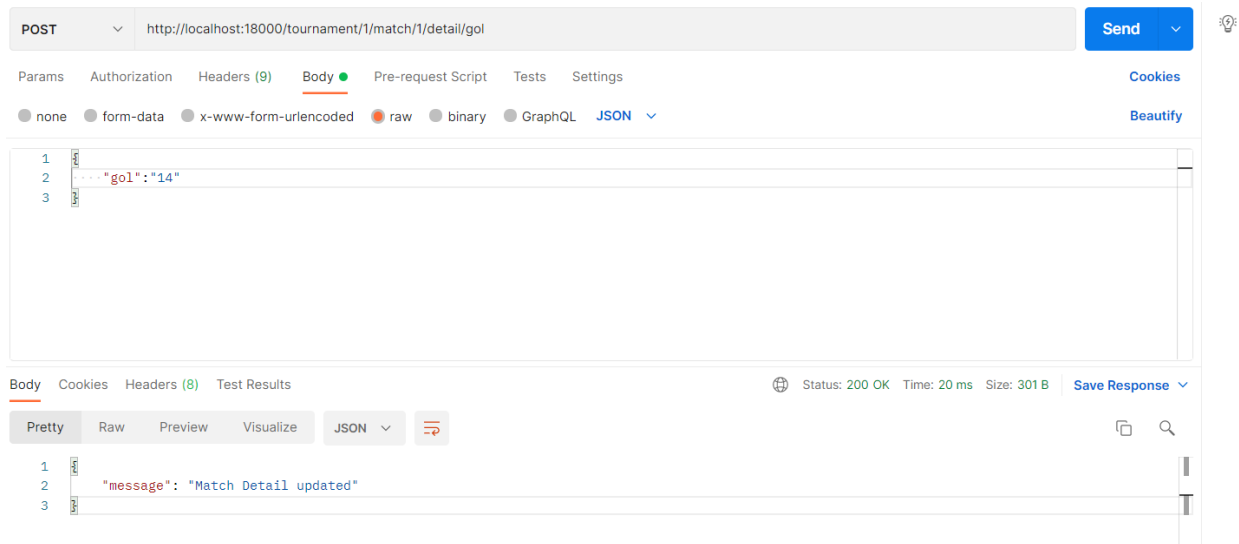
23- POST: Set Início (Success)

Definição do horário de início de uma partida de um torneio



24- POST: Set Gol(Success)

Definição do número de gols para uma partida de um torneio



25- POST: Set Intervalo (Success)

Definição do tempo de intervalo de uma partida de um torneio

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

- Method:** POST
- URL:** `http://localhost:18000/tournament/1/match/1/detail/interval`
- Body:**

```
{  "interval": "25"}
```
- Status:** 200 OK
- Time:** 29 ms
- Size:** 301 B
- Response Body:**

```
{  "message": "Match Detail updated"}
```

26- POST: Set Acréscimo (Success)

Definição do tempo de acréscimo em uma partida de um torneio

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

- Method:** POST
- URL:** `http://localhost:18000/tournament/1/match/1/detail/overtime`
- Body:**

```
{  "overtime": "6"}
```
- Status:** 200 OK
- Time:** 17 ms
- Size:** 301 B
- Response Body:**

```
{  "message": "Match Detail updated"}
```


27- POST: Set Substituição (Success)

Definição das substituições ocorridas em uma partida de um torneio

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

- Method:** POST
- URL:** http://localhost:18000/tournament/1/match/1/detail/replacements
- Body:**

```
{  "replacements": "7-11,1-2,9-6"}
```
- Status:** 200 OK
- Time:** 29 ms
- Size:** 301 B
- Response:**

```
{  "message": "Match Detail updated"}
```

28- POST: Set Advertencia (Success)

Definição do número de advertências em uma partida de um torneio

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

- Method:** POST
- URL:** http://localhost:18000/tournament/1/match/1/detail/warning
- Body:**

```
{  "warning": "16"}
```
- Status:** 200 OK
- Time:** 51 ms
- Size:** 301 B
- Response:**

```
{  "message": "Match Detail updated"}
```

33- POST: Set Fim (Success)

Definição do horário de término de uma partida de um torneio

The screenshot shows a REST client interface with a POST request to `http://localhost:18000/tournament/1/match/1/detail/end`. The request body is a JSON object: `{ "end": "2022-04-07T01:14:21.000Z" }`. The response status is 200 OK, with a time of 18 ms and a size of 301 B. The response body is a JSON object: `{ "message": "Match Detail updated" }`.

```
POST http://localhost:18000/tournament/1/match/1/detail/end

{
  "end": "2022-04-07T01:14:21.000Z"
}
```

Status: 200 OK Time: 18 ms Size: 301 B

```
{
  "message": "Match Detail updated"
}
```

Resultado:

The screenshot shows a REST client interface with a GET request to `http://localhost:18000/match-detail`. The response status is 200 OK, with a time of 40 ms and a size of 529 B. The response body is a JSON object containing match details.

```
GET http://localhost:18000/match-detail

This request does not have a body
```

Status: 200 OK Time: 40 ms Size: 529 B

```
{
  "id": 1,
  "matchId": "1",
  "start": "2022-04-06T22:44:33.000Z",
  "interval": "25",
  "gol": "14",
  "overtime": "6",
  "replacements": "7-11,1-2,9-6",
  "warnings": "16",
  "end": "2022-04-07T01:14:21.000Z",
  "createdAt": "2022-04-07T05:23:57.000Z",
  "updatedAt": "2022-04-07T05:40:11.000Z"
}
```

34- GET: Get All Tournaments (Success)

Comando para obter todos os torneios registrados

The screenshot shows the Postman interface with a GET request to `http://localhost:18000/tournament`. The response is a JSON array containing one tournament object. The status is 200 OK, and the response is saved.

KEY	VALUE	DESCRIPTION
Key	Value	Description

```
1 {
2   "id": 1,
3   "name": "Copa Brasil",
4   "rounds": "160",
5   "endDate": "2015-03-20",
6   "startDate": "2015-03-24",
7   "winner": "Atlético",
8   "createdAt": "2022-04-06T23:16:38.000Z",
9   "updatedAt": "2022-04-06T23:16:38.000Z"
10 }
11
12
```

35- GET: Get Tournament By Id (Success)

Comando para obter um torneio registrado com um id específico

The screenshot shows the Postman interface with a GET request to `http://localhost:18000/tournament/2`. The response is a JSON object representing the tournament with id 2. The status is 200 OK, and the response is saved.

KEY	VALUE	DESCRIPTION
Key	Value	Description

```
1 {
2   "id": 2,
3   "name": "Brasileirão",
4   "rounds": "32",
5   "endDate": "10 de dezembro",
6   "startDate": "hoje",
7   "winner": "none",
8   "createdAt": "2022-04-07T05:43:16.000Z",
9   "updatedAt": "2022-04-07T05:43:16.000Z"
10 }
11
12
```

36- DELETE: Remove Tournament (Success)

Remover um torneio registrado com um id específico

DELETE

http://localhost:18000/tournament/1

Send

ParamsAuthorizationHeaders (6)BodyPre-request ScriptTestsSettingsCookies

Headers6 hidden

KEY	VALUE	DESCRIPTION		Bulk Edit	Presets
Key	Value	Description			

BodyCookiesHeaders (6)Test Results

Status: 200 OKTime: 9 msSize: 299 BSave Response

PrettyRawPreviewVisualizeJSON

```
1  {
2    "message": "Tournament deleted"
3  }
```