

Bowling API Documentation:

1) Create New Bowlers:

POST Request: <http://localhost:8080/bowling/createBowler>


POST ▾ http://localhost:8080/bowling/createBowler

Authorization | Headers (0) | **Body** | Pre-request script | Tests

☐ form-data ☐ x-www-form-urlencoded ☒ raw ☐ binary Text ▾

```
1 {  
2   name: "Tom"  
3 }
```

Body | Cookies | Headers (4) | Tests | Status **200 OK** Time **655 ms**

Pretty Raw Preview | HTML ▾ | 

```
i 1 {"name": "Tom", "ID": 1, "DOJ": "Tue Nov 17 12:09:50 CST 2015"}
```

2) Get all Bowlers:

GET Request: <http://localhost:8080/bowling/Bowlers/getBowlers>

GET <http://localhost:8080/bowling/Bowlers/getBowlers> Params

Authorization Headers (0) Body Pre-request script Tests

No Auth

Body Cookies Headers (4) Tests Status 200 OK Time 359 ms

Pretty Raw Preview JSON

```
[
  {
    "name": "Willas",
    "id": 4,
    "doj": "Tue Nov 17 12:12:15 CST 2015"
  },
  {
    "name": "Jerry",
    "id": 2,
    "doj": "Tue Nov 17 12:12:08 CST 2015"
  },
  {
    "name": "Tom",
    "id": 1,
    "doj": "Tue Nov 17 12:09:50 CST 2015"
  },
  {
    "name": "Bruce",
    "id": 3,
    "doj": "Tue Nov 17 12:12:11 CST 2015"
  },
  {
    "name": "Hardy",
    "id": 5,
    "doj": "Tue Nov 17 12:12:11 CST 2015"
  }
]
```

3) Get Bowlers by ID:

GET Request: <http://localhost:8080/bowling/Bowlers/getBowler/{bowlerID}>

GET <http://localhost:8080/bowling/Bowlers/getBowler/1>

Authorization Headers (0) Body Pre-request script Tests

No Auth

Body Cookies Headers (4) Tests Status 200 OK Time 128 ms

Pretty Raw Preview HTML

```
{
  "name": "Tom",
  "ID": 1,
  "DOJ": "Tue Nov 17 12:09:50 CST 2015"
}
```

4) Create League:

POST Request: <http://localhost:8080/bowling/createLeague>

POST ▼

http://localhost:8080/bowling/createLeague

Authorization

Headers (0)

Body

Pre-request script

Tests

☐ form-data

☐ x-www-form-urlencoded

☒ raw

☐ binary

Text ▼

1 {

2 name: "Chicago League"

3 }

Body

Cookies

Headers (4)

Tests

Status 200 OK Time 57 ms

Pretty

Raw

Preview

HTML ▼

≡

i 1

{"ID":1,"name":"Chicago League","bowlers":[],"lotteries":[]}

5) Get all Leagues:

GET Request: <http://localhost:8080/bowling/Leagues/getLeagues>

GET ▾

<http://localhost:8080/bowling/Leagues/getLeagues>

Authorization

Headers (0)

Body

Pre-request script

No Auth ▾

Body

Cookies

Headers (4)

Tests

Status 200 OK

Time 103 ms

Pretty

Raw

Preview

JSON ▾

≡

```
1 [
2   {
3     "name": "Illinois League",
4     "bowlers": [],
5     "lotteries": [],
6     "id": 2
7   },
8   {
9     "name": "Elite League",
10    "bowlers": [],
11    "lotteries": [],
12    "id": 3
13  },
14  {
15    "name": "Chicago League",
16    "bowlers": [],
17    "lotteries": [],
18    "id": 1
19  }
20 ]
```

6) Get League by ID:

GET Request:

<http://localhost:8080/bowling/Leagues/getLeague/{leagueID}>

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

- Request Bar:** Method **GET** and URL `http://localhost:8080/bowling/Leagues/getLeague/1`.
- Request Tabs:** **Authorization**, **Headers (0)**, **Body**, and **Pre-request script**.
- Auth Section:** A dropdown menu showing **No Auth**.
- Response Bar:** **Body** (selected), **Cookies**, **Headers (4)**, **Tests**, **Status 200 OK**, and **Time 73 ms**.
- Response Format:** Buttons for **Pretty** (selected), **Raw**, and **Preview**. A dropdown menu shows **HTML**.
- Response Body:** A JSON object: `{ "ID": 1, "name": "Chicago League", "bowlers": [], "lotteries": [] }`.

7) Join League:

POST Request:

<http://localhost:8080/bowling/Leagues/joinLeague>

The screenshot shows a REST client interface. At the top, the method is set to POST and the URL is <http://localhost:8080/bowling/Leagues/joinLeague>. Below the URL bar are tabs for Authorization, Headers (0), Body, Pre-request script, and Tests. The Body tab is selected, and the format is set to raw. The request body contains the following JSON:

```
1 {  
2   bowlerID: "1",  
3   leagueID: "1"  
4 }
```

Below the request body, the response status is 200 OK and the time taken is 53 ms. The response body is displayed in the Pretty view, showing the following JSON:

```
i 1 {"ID":1,"name":"Chicago League","bowlers":[1],"lotteries":[]}
```

8) Get Bowlers who Joined a League:

GET Request: <http://localhost:8080/bowling/Leagues/{leagueID}/getBowlers>

GET ▼
http://localhost:8080/bowling/Leagues/1/getBowlers

Authorization
Headers (0)
Body
Pre-request script

No Auth ▼

Body
Cookies
Headers (4)
Tests
Status 200 OK Time 63 ms

Pretty
Raw
Preview
JSON ▼
≡

```

1  [
2    {
3      "name": "Willas",
4      "id": 4,
5      "doj": "Tue Nov 17 12:12:15 CST 2015"
6    },
7    {
8      "name": "Jerry",
9      "id": 2,
10     "doj": "Tue Nov 17 12:12:08 CST 2015"
11   },
12   {
13     "name": "Tom",
14     "id": 1,
15     "doj": "Tue Nov 17 12:09:50 CST 2015"
16   },
17   {
18     "name": "Bruce",
19     "id": 3,
20     "doj": "Tue Nov 17 12:12:11 CST 2015"
21   },
22   {
23     "name": "Hardy",
24     "id": 5,
25     "doj": "Tue Nov 17 12:12:18 CST 2015"
26   }
27 ]

```

9) Create Lottery:

POST Request: <http://localhost:8080/bowling/createLottery>

POST ▾

http://localhost:8080/bowling/createLottery

Authorization

Headers (0)

Body

Pre-request script

Tests

☐ form-data

☐ x-www-form-urlencoded

☒ raw

☐ binary

Text ▾

1 {

2 leagueID: "1",

3 name: "Lottery 1"

4 }

Body

Cookies

Headers (4)

Tests

Status 200 OK Time 53 ms

Pretty

Raw

Preview

HTML ▾


≡

i 1


{ "name": "Lottery 1", "ID": 1, "Jackpot_balance": 0.0, "bowler_tickets": [] }

10) Get all Lotteries for a given LeagueID:



GET Request: <http://localhost:8080/bowling/Leagues/{leagueID}/getLotteries>

GET  | http://localhost:8080/bowling/Leagues/1/getLotteries

Authorization | Headers (0) | Body | Pre-request script

No Auth 


Body | Cookies | Headers (4) | Tests | Status 200 OK | Time 106 ms

Pretty | Raw | Preview | JSON  


```
1 [
2   {
3     "name": "Lottery 1",
4     "id": 1
5   },
6   {
7     "name": "Lottery 2",
8     "id": 2
9   }
10 ]
```

11) Get a specific lottery based on Lottery ID and League ID:



GET Request: <http://localhost:8080/bowling/Leagues/{leagueID}/getLottery/{lotteryID}>

GET  | http://localhost:8080/bowling/Leagues/1/getLottery/1

Authorization | Headers (0) | Body | Pre-request scrip

No Auth 

Body | Cookies | Headers (4) | Tests | Status 200 OK | Time 61 ms

Pretty | Raw | Preview | JSON  

```
1 {
2   "name": "Lottery 1",
3   "ID": 1,
4   "Jackpot_balance": 0,
5   "bowler_tickets": []
6 }
```

12) Purchase Ticket:

POST Request: <http://localhost:8080/bowling/purchaseTicket>

POST ▾

http://localhost:8080/bowling/purchaseTicket

Authorization

Headers (0)

Body

Pre-request script

Tests

☐ form-data

☐ x-www-form-urlencoded

☒ raw

☐ binary

Text ▾

```
1 {
2   bowlerID: "4",
3   leagueID: "1",
4   lotteryID: "1"
5 }
```

Body

Cookies

Headers (4)

Tests

Status 200 OK

Time 57 ms

Pretty

Raw

Preview

HTML ▾

```
i 1 [{"name": "Lottery 1", "ID": 1, "Jackpot_balance": 3.0, "bowler_tickets": [1, 2, 4]}]
```

13) Get all purchased Tickets for the current week based on Lottery ID and League ID:

GET Request:

<http://localhost:8080/bowling/Leagues/{leagueID}/Lottery/{lotteryID}/getPurchasedTickets>

Output:

A HashMap object with Key as BowlerID and value as the number of tickets purchased by that bowler.

GET <http://localhost:8080/bowling/Leagues/1/Lottery/1/getPurchasedTickets>

Authorization Headers (0) Body Pre-request script Tests

No Auth

Body Cookies Headers (4) Tests Status 200 OK Time 66 ms

Pretty Raw Preview JSON

```
{
  "1": 1,
  "2": 1,
  "4": 2,
  "5": 1
}
```

14) Get purchased Ticket for a particular bowler ID based on Lottery ID and League ID:

GET Request:

<http://localhost:8080/bowling/Leagues/{leagueID}/Lottery/{lotteryID}/getPurchasedTicket/{BowlerID}>

Output:

An integer value that provides the number of tickets bought by the given bowler.

GET <http://localhost:8080/bowling/Leagues/1/Lottery/1/getPurchasedTicket/4>

Authorization Headers (0) Body Pre-request script Tests

No Auth

Body Cookies Headers (4) Tests Status 200 OK Time 59 ms

Pretty Raw Preview HTML

```
2
```

15) Draw Winner for a specific lottery ID in a League:

GET Request: <http://localhost:8080/bowling/Leagues/{leagueID}/Lottery/{lotteryID}/drawWinner>

Output:

A Randomly selected Bowler from all the bowlers who bought ticket for the draw.

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

- Method:** GET
- URL:** `http://localhost:8080/bowling/Leagues/1/Lottery/1/drawWinner`
- Authorization:** No Auth
- Headers:** (0)
- Body:** (empty)
- Pre-request script:** (empty)
- Status:** 200 OK
- Time:** 55 ms
- Response Body (JSON):**

```
{
  "name": "Willas",
  "ID": 4,
  "DOJ": "Tue Nov 17 12:12:15 CST 2015"
}
```

16) Providing a chance to the winner of the draw to claim Jackpot by throwing a strike for the given lottery:

GET Request:

<http://localhost:8080/bowling/Leagues/{leagueID}/Lottery/{lotteryID}/throwStrike/{winnerID}>

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

- Method:** GET
- URL:** `http://localhost:8080/bowling/Leagues/1/Lottery/1/throwStrike/4`
- Authorization:** No Auth
- Headers:** (0)
- Body:** (empty)
- Pre-request script:** (empty)
- Status:** 200 OK
- Time:** 55 ms
- Response Body (JSON):**

```
{
  "leagueID": 1,
  "lotteryID": 1,
  "bowlerID": 4,
  "roll": 4,
  "winning_amount": 0.5,
  "remaining_balance": 4.5,
  "DrawDate": "Tue Nov 17 13:06:56 CST 2015"
}
```

Note: After throwStrike request, balance gets carry forward for the next weeks draw. Also the tickets arraylist is cleared.

GET ⌵ | http://localhost:8080/bowling/Leagues/1/getLottery/1

Authorization | Headers (0) | Body | Pre-request script

No Auth ⌵

Body | Cookies | Headers (4) | Tests | Status 200 OK | Time 61 ms

Pretty | Raw | Preview | JSON ⌵ | ≡

```
1 {  
2   "name": "Lottery 1",  
3   "ID": 1,  
4   "Jackpot_balance": 4.5,  
5   "bowler_tickets": []  
6 }
```

Data Storage:

All data is getting stored in HashSet as shown below. As serialization is not performed at start of the server the HashSet objects get reinitialized.

```
package com.shouvik.bean;  
  
import java.util.HashSet;  
  
public class BowlingClub{  
  
    public static HashSet<Bowler> bowlers = new HashSet<Bowler>();  
    public static HashSet<League> leagues = new HashSet<League>();  
    public static HashSet<Payment> payments = new HashSet<Payment>();
```

Things remaining:

- Fixing a day for the draw and error handling if DrawWinner request is called before that day.
- Allowing ticket sale till a day before the draw date. Freezing the ticket sale on the draw date.
- Displaying the Jackpot payout history data which is stored in the data model named payments.
- Serializing Datastore into a local file when the server stops to maintain persistence.