

About the code:

Add points:

The code will take 3 parameters from the body

Points, Company, Date.

If the company already exist in the JSON array, points and date are updated for that company.

If not, a new object is created and pushed into the array.

Delete Points:

The code will take one parameter from the body

Points (to be deleted)

If total points < points, code will return "not enough points"

Else, Points will be deducted from the object with the least timestamp value, the operation is repeated until points = 0

Check Points:

Will return array of JSON objects.

Instructions to run the service.

- 1) Go to the project folder using terminal and run the following command
npm run dev
- 2) Get request links

Request	Route
Add Points	http://localhost:3000/addPoints
Delete Points	http://localhost:3000/deletePoints
Check points	http://localhost:3000/points

I have used postman to make the calls, following are the screenshots of responses I received:

Add points:

Body of requests

```
{"payer": "DANNON", "points": 1000, "timestamp": "2020-11-02T14:00:00Z"}
```

```
{"payer": "UNILEVER", "points": 200, "timestamp": "2020-10-31T11:00:00Z"}
```

```
{"payer": "DANNON", "points": -200, "timestamp": "2020-10-31T15:00:00Z"}
```

```
{"payer": "MILLER COORS", "points": 10000, "timestamp": "2020-11-01T14:00:00Z"}
```

```
{"payer": "DANNON", "points": 300, "timestamp": "2020-10-31T10:00:00Z"}
```

GET http://localhost:3000/addPoints Send

Params Authorization Headers (9) Body ● Pre-request Script Tests Settings Cookies

Body Cookies Headers (7) Test Results 200 OK 7 ms 440 B Save Response

Pretty Raw Preview Visualize JSON

```
1 [
2   {
3     "company": "DANNON",
4     "points": 1100,
5     "date": "2020-11-02T14:00:00Z"
6   },
7   {
8     "company": "UNILEVER",
9     "points": 200,
10    "date": "2020-10-31T11:00:00Z"
11  },
12  {
13    "company": "MILLER COORS",
14    "points": 10000,
15    "date": "2020-11-01T14:00:00Z"
16  }
17 ]
```

Bootcamp Runner Trash

Delete Points

Body: {"points": 5000}

GET http://localhost:3000/deletePoints Send

Params Authorization Headers (9) Body ● Pre-request Script Tests Settings Cookies

Body Cookies Headers (7) Test Results 200 OK 6 ms 346 B Save Response

Pretty Raw Preview Visualize JSON

```
1 [
2   {
3     "company": "UNILEVER",
4     "points": "-200",
5     "date": "now"
6   },
7   {
8     "company": "MILLER COORS",
9     "points": "-4800",
10    "date": "now"
11  }
12 ]
```

Bootcamp Runner Trash

Check Points

The screenshot shows a REST client interface with a GET request to `http://localhost:3000/points`. The response is a JSON array of three objects, each representing a company's points and a timestamp. The response status is 200 OK, with a response time of 7 ms and a body size of 437 B.

Request: GET `http://localhost:3000/points`

Response: 200 OK, 7 ms, 437 B

Body (JSON):

```
[
  {
    "company": "DANNON",
    "points": 1100,
    "date": "2020-11-02T14:00:00Z"
  },
  {
    "company": "UNILEVER",
    "points": 0,
    "date": "2020-10-31T11:00:00Z"
  },
  {
    "company": "MILLER COORS",
    "points": 5200,
    "date": "2020-11-01T14:00:00Z"
  }
]
```

The interface includes tabs for Params, Authorization, Headers (9), Body, Pre-request Script, Tests, Settings, and Cookies. The Body tab is selected, and the response is displayed in a Pretty view. The bottom status bar shows icons for Bootcamp, Runner, Trash, and a window management icon.