

N26

Backend Regular Engineer Test

🕒 1d 16h  
to test end

## ★ Transactions statistics



1

We would like to have a RESTful API for our statistics. The main use case for the API is to calculate realtime statistics for the last 60 seconds of transactions.

The API needs the following endpoints:

- **POST /transactions** – called every time a transaction is made. It is also the sole input of this rest API.
- **GET /statistics** – returns the statistic based of the transactions of the last 60 seconds.
- **DELETE /transactions** – deletes all transactions.

You can complete the challenge offline using an IDE of your choice. To download the application skeleton, please enable **Use Git** in the editor and follow the instructions on screen. Please make sure you push your changes to the master branch and test your solution on HackerRank before submitting.

### Specs

#### **POST /transactions**

This endpoint is called to create a new transaction.

Body:

```
{
  "amount": "12.3343",
  "timestamp": "2018-07-17T09:59:51.312Z"
}
```

Where:

- **amount** – transaction amount; a string of arbitrary length that is parsable as a BigDecimal
- **timestamp** – transaction time in the ISO 8601 format **YYYY-MM-DDThh:mm:ss.sssZ** in the UTC timezone (this is not the current timestamp)

Returns: Empty body with one of the following:

- 201 – in case of success
- 204 – if the transaction is older than 60 seconds
- 400 – if the JSON is invalid
- 422 – if any of the fields are not parsable or the transaction date is in the future

### GET /statistics

This endpoint returns the statistics computed on the transactions within the last 60 seconds.

Returns:

```
{
  "sum": "1000.00",
  "avg": "100.53",
  "max": "200000.49",
  "min": "50.23",
  "count": 10
}
```

Where:

- **sum** – a **BigDecimal** specifying the total sum of transaction value in the last 60 seconds
- **avg** – a **BigDecimal** specifying the average amount of transaction value in the last 60 seconds
- **max** – a **BigDecimal** specifying single highest transaction value in the last 60 seconds
- **min** – a **BigDecimal** specifying single lowest transaction value in the last 60 seconds
- **count** – a **long** specifying the total number of transactions that happened in the last 60 seconds

All BigDecimal values always contain exactly two decimal places and use `HALF\_ROUND\_UP` rounding. eg: 10.345 is returned as 10.35 10.8 is returned as 10.80

### DELETE /transactions

This endpoint causes all existing transactions to be deleted

The endpoint should accept an empty request body and return a 204 status code.

## Requirements

These are the additional requirements for the solution:

- You are free to choose any JVM language to complete the challenge in, but **your application has to run in Maven**.
- The API has to be threadsafe with concurrent requests.
- The solution has to work without a database (this also applies to in-memory databases).
- Unit tests are *\*mandatory\**.
- **mvn clean install** and **mvn clean integration-test** must complete successfully.
- Please ensure that no changes are made to the **src/it** folder since they contain automated tests that will be used to evaluate the solution.

File

Terminal

View

Find

Run

Use Git



**PROJECT**

```

Application.java ×  >_ Terminal - Project Test ×
v project 2018-09-15 08:44:07.911 INFO 2175 --- [          Thread-
  v src 2] o.s.w.c.s.GenericWebApplicationContext : Closing org.springframework
    v it framework.web.context.support.GenericWebApplicationContext@4808bc9b: sta
      > ... f context hierarchy
      > ... [INFO]
    v main [INFO] Results:
      v ... [INFO]
        ... [INFO] Tests run: 1, Failures: 0, Errors: 0, Skipped: 0
          ... [INFO]
            [INFO] -----
            -----
            [INFO] BUILD SUCCESS
            [INFO] -----
            -----
            [INFO] Total time: 25.896 s
            [INFO] Finished at: 2018-09-
v target 15T08:44:08+00:00
  > cla... [INFO] Final Memory: 35M/213M
  > ge... [INFO] -----
  > ma... -----
pom.xml | | ( ) | | | | \

```

119 x 17

Run Tests

Submit code &amp; continue

*You can change your submission later.*

---

[About](#) [Privacy Policy](#) [Terms of Service](#)