

Stock Exchange Controller Unit Test

micro_service - stockexchange/src/main/resources/application.properties - Eclipse IDE

File Edit Navigate Search Project Run Window Help

Project Explorer

- bangtanrestsql
- movie-catalog-service
- movie-info-service
- MyWebApp
- ratings-data-service
- stockexchange
 - src/main/java
 - src/main/resources
 - static
 - templates
 - application.properties
 - src/test/java
 - com.wellsfargo.stocks.stockexcha
 - StockexchangeApplicationTests
 - StockExchangeServiceTests.java
- JRE System Library [JavaSE-1.8]
- Maven Dependencies
- JUnit 4
- src
- target
 - HELP.md
 - mvnw
 - mvnw.cmd
 - pom.xml

StockexchangeA... StockExchangeC... application.pr... StockExchangeS... StockExchangeS...

```
1 server.port = 3797
2 spring.datasource.url=jdbc:mysql://localhost:3306/wells?verifyServerCertificate=false&useSSL=false
3 spring.datasource.username=root
4 spring.datasource.password=Swaathi7*
5 spring.datasource.driver-class-name=com.mysql.cj.jdbc.Driver
6 spring.jpa.database-platform = org.hibernate.dialect.MySQL57InnoDBDialect
7 spring.jpa.properties.hibernate.format_sql = true
8 spring.jpa.properties.hibernate.generate_statistics = true
9 spring.jpa.generate-ddl=true
10 spring.jpa.hibernate.ddl-auto = update
11 spring.jpa.show-sql = true
12 spring.jpa.properties.hibernate.enable_lazy_load_no_trans=true
13
```

JUnit

Finished after 11.399 seconds

Runs: 3/3 Errors: 0 Failures: 0

StockexchangeApplicationTests [Runner: JUnit 5] (1.104 s)

- getStockExchangesListTest() (0.877 s)
- addStockExchangeTest() (0.165 s)
- getCompaniesListTest() (0.062 s)

Failure Trace

Console

<terminated> StockexchangeApplicationTests [JUnit] C:\Program Files\Java\jdk1.8.0_221\bin\javaw.exe (Sep 1, 2020, 2:22:53 AM)

2:23 AM
9/1/2020

Stock Exchange Service Unit test

The screenshot shows the Eclipse IDE interface with the following components:

- Project Explorer (Left):** Displays the project structure for 'com.wellsfargo.stocks.stockexchange'. The 'src/test/java' directory is expanded, showing the test classes: 'com.wellsfargo.stocks.stockexchange.StockExchangeApplicationTests', 'com.wellsfargo.stocks.stockexchange.StockExchangeServiceTests', and 'com.wellsfargo.stocks.stockexchange.StockExchangeServiceTests.java' (which is selected).
- Editor (Center):** Shows the code for 'StockExchangeServiceTests.java'. The code includes:

```
115     }  
116  
117     public String getCeo() {  
118         return ceo;  
119     }  
120  
121     public void setCeo(String ceo) {  
122         this.ceo = ceo;  
123     }
```
- Console (Bottom):** Shows the output of the Java application, indicating that 21870700 nanoseconds were spent executing 15 JDBC statements and 0 nanoseconds were spent executing 0 JDBC batches.
- JUnit (Bottom):** Shows the test results for 'com.wellsfargo.stocks.stockexchange.StockExchangeServiceTests'. The test suite finished after 7.831 seconds. The results are summarized as:
Runs: 4/4
Errors: 0
Failures: 0
The individual test results are listed below:
- getStockExchangesListTest (0.057 s)
- addStockExchangeTest (0.010 s)
- contextLoads (0.005 s)
- getCompaniesListTest (0.007 s)

Get Company List Service Result

The screenshot shows the Postman application interface. On the left, the 'History' tab is active, displaying a list of requests from 'Today' and 'Yesterday'. The main workspace shows an 'Untitled Request' with the following details:

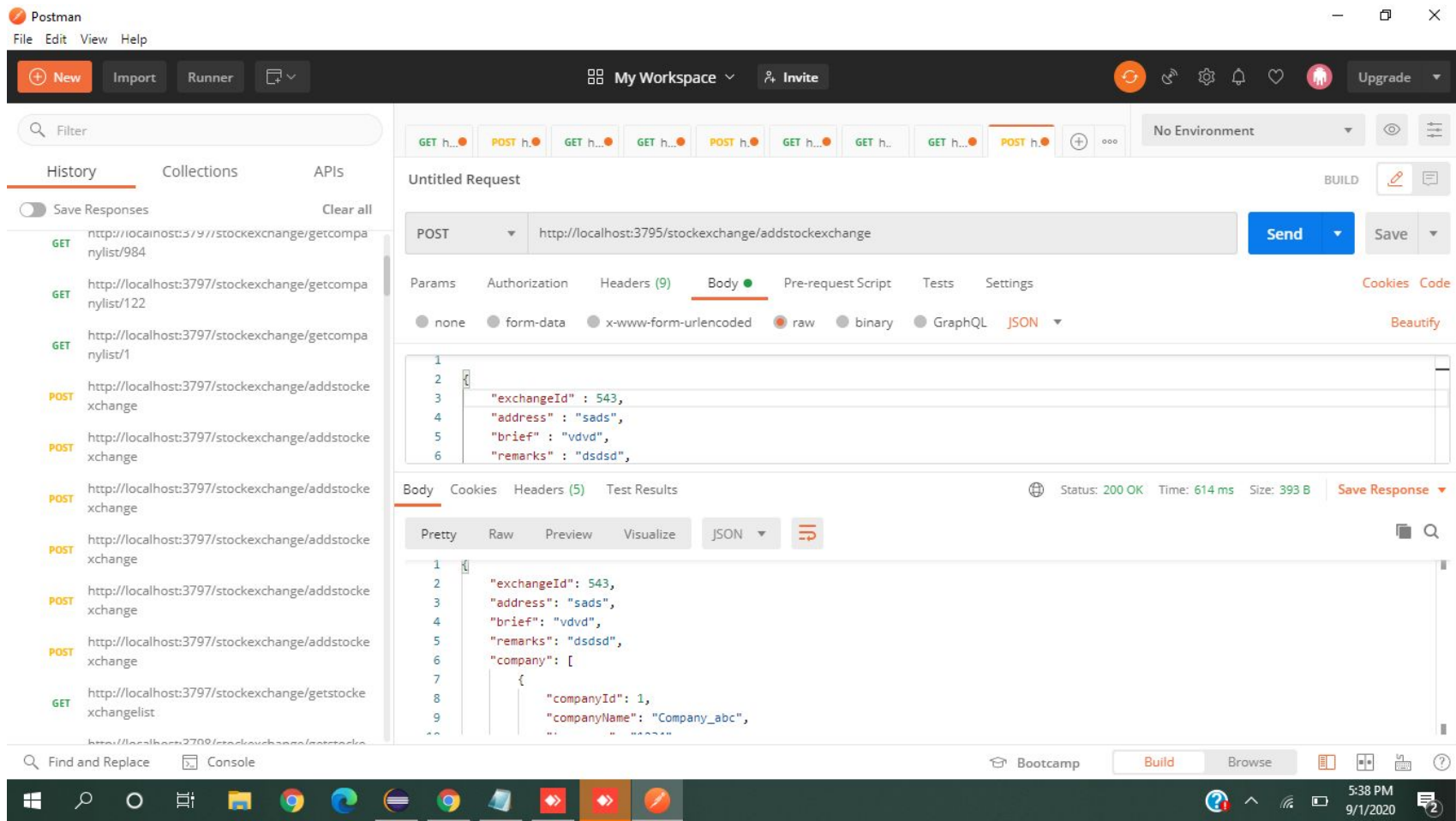
- Method:** GET
- URL:** http://localhost:3795/stockexchange/getcompanylist/984
- Params:** Query Params table with one entry: Key, Value, Description.
- Body:** JSON format, showing the response data.

The response status is 200 OK, with a time of 835 ms and a size of 312 B. The response body is displayed in JSON format:

```
1 {
2   "companyId": 1,
3   "companyName": "Company_abc",
4   "turnover": "1234",
5   "ceo": "bcd",
6   "boardOfDirectors": "abc",
7   "writeUp": "writeup",
8   "ipo": null,
9   "stockPrices": []
10 }
11
```

The bottom of the screen shows the Windows taskbar with various application icons and the system clock indicating 5:37 PM on 9/1/2020.

Add Stock Exchange Result



Get Stock Exchange Result

The screenshot shows the Postman application interface. The top bar includes the Postman logo, menu items (File, Edit, View, Help), and buttons for New, Import, Runner, and My Workspace. The right side of the top bar has an Invite button, a refresh button, a settings gear, a bell, a heart, a user icon, and an Upgrade button.

The left sidebar contains a search bar and three tabs: History, Collections, and APIs. The History tab is active, showing a list of requests. The first request is a GET request to `http://localhost:3795/stockexchange/getstockexchangelist`, which is highlighted. Other requests in the history include GET requests to `http://localhost:3797/stockexchange/getcompa` and POST requests to `http://localhost:3797/stockexchange/addstocke`.

The main area displays the details of the selected request, titled "Untitled Request". The request method is GET, and the URL is `http://localhost:3795/stockexchange/getstockexchangelist`. The request is sent to the "No Environment" environment. The response is displayed in the "Body" tab, showing a JSON object with the following structure:

```
{
  "exchangeId": 543,
  "address": "sads",
  "brief": "vdvd",
  "remarks": "dsdsd",
  "company": [
    {
      "companyId": 1,
      "companyName": "Company_abc",
      "turnover": "1234",
      "ceo": "bcd",
      "boardOfDirectors": "abc",
      "writeUp": "writeup",
      "ipo": null,
      "stockPrices": []
    }
  ]
}
```

The response status is 200 OK, the time is 534 ms, and the size is 967 B. The bottom status bar shows the Bootcamp logo, Build and Browse buttons, and system icons for the taskbar.