Tech Assessment Survey

1. The time the exercise took (after dev environment is set up)

Around 3.5 to 4 hours on core coding.

2. Exercise Difficulty: Easy, Moderate, Difficult, Very Difficult

Between moderate to difficult. Mainly due to limited time. (1 – 4 hours)

3. How did you feel about the exercise itself? (1 lowest, 10 highest—awesome way to assess coding ability)

10.

Great assessment. It is an open-end question, thus it is an easy and straightforward way to examine the coding and design ability of candidates at different levels.

4. How do you feel about coding an exercise as a step in the interview process? (1 lowest, 10 highest—awesome way to assess coding ability)

8.

This assessment is able to examine candidate's ability in both coding and design.

5. What would you change in the exercise and/or process?

The recommended time (1 – 4 hour) on this assessment is a little bit short. It may need longer time if a candidate hope to optimize the assessment.

Assessment Instruction

1. Local Environment setup

1.1 Programming language and tools

Programming Language	Java
JDK	Eclipse JEE
Web Servlet Container	Apache Tomcat
Database	MongoDB (local environment)
Other Tools	Postman

1.2 Project Setup

a) Download whole project to a local machine. Related libraries are included in this peoject.

b) Open the project in Eclipse JEE

c) Setup Apache Tomcat, listen to port 8080.

d) Setup MongoDB, listen to port 27017

e) Start the server. Run DBTableCreation.java.

2. Tech assessment design details

2.1 General purpose

A **marketplace** is built for sellers and self-employed buyers. Sellers are allowed to post projects/jobs with the maximum budget, deadline and other job details. Buyers are able to bid for a project with a lower price. When the deadline is reached, the buyer with the lowest bid wins the bid.

2.2 Models

Three models are designed in this assessment, including seller, buyer, and project. Therefore three tables will be created when DBTableCreation.java is run. The schema of each table is presented below.

- **Table sellers:**

Column Name	seller_id	creation_date	cur_projects	History
Note	String, Primary Key	Date	List<String>	List<String>

- **Table buyers**

Column Name	buyer_id	creation_date	cur_bids	History
Note	String, Primary Key	Date	List<String>	List<String>

- **Table projects**

Column Name	project_id	seller_id	category	description	cur_price	buyer_id	creation_date	expire_date	status
Note	String, Primary Key	String	String	String	Double	String	Date	Date	Boolean

2.3 Available functions

2.3.1 Seller registration / insert rows in sellers table

By using sellerRegister API, a new row could be added into sellers table.

HTTP Method:

POST

URL:

http://localhost:8080/MarketPlace/seller

Input Sample:

{"seller_id":"testSeller"}

Related Java Method:

MongoDBConnection.java: registerSeller(String sellerId)

2.3.2 Buyer registration / insert rows in buyers table

By using buyerRegister API, a new row could be added into sellers table.

HTTP Method:

POST

URL:

http://localhost:8080/MarketPlace/ buyer

Input Sample:

{"buyer_id" : "testBuyer"}

Related Java Method:

MongoDBConnection.java: registerBuyer(String buyerId)

2.3.3 Post projects / insert rows in projects table

By using postProject API, a new row could be added into projects table.

HTTP Method:

POST

URL:

http://localhost:8080/MarketPlace/post

Input Sample:

```
{
    "project_id":"testProject",
    "seller_id":"testSeller",
```

"category":"Software Engineer",

"description":"Web service development",

"cur_price": 2000,

"expire_date": 100000;

}

Note:

For the expire_date, input the post validation duration in milliseconds. For example, **"expire_date": 10,000, means this post will be expired in 10s**.

Project status, creation date and expire date will be automatically generated at backend. Project status is true when a new project is posted.

Related Java Method:

Project.java
MongoDBConnection.java: postProject(Project project)

2.3.4 Search current open projects

Current open projects could be searched by using mainPage API.

HTTP Method:

GET

URL:

http://localhost:8080/MarketPlace/main

Note:

Status is used to indicate whether this project is expired or not. True means this project is still validate and visible for buyers. False means this project has expired. An expired project will not be visible for buyers in main page.

Related Java Method:

MongoDBConnection.java: searchCurProj(), checkStatus()
jsonHelper.java: writeJsonArrayt (HttpServletResponse response, JSONArray array)

2.3.5 Bid for a project

A buyer can bid for a project by using bidProject API.

HTTP Method:

POST

URL:

http://localhost:8080/MarketPlace/bid

Input Sample:

```
{
    "buyer_id": "testBuyer",
    "project_id": "testProject",
    "price": 1200
}
```

Note:

A bid is valid only when the price is lower than project's current price, and project's current bidder is not the same buyer. The project cur_price will be changed via a valid bid.

Once a bid is success, this project ID will be removed from previous buyer's cur_bids column, and added into current buyer's cur_bids column.

Related Java Method:

MongoDBConnection.java: bidProject(String buyerId, double price, String projectId), checkStatus()

jsonHelper.java: writeJsonObject(HttpServletResponse response, JSONObject obj)

2.3.6 Project expiration.

mainPage API could also be used to check whether a project is expired.

HTTP Method:

GET

URL:

http://localhost:8080/MarketPlace/main

Note:

Expired project will not be printed. At backend, when a project is expired, the status of this project will be set to false in "projects" table. In "sellers" table, the seller of this project will move this project's ID from cur_projects column to history column. In "buyers" table, the final bidder of this project will move this project's ID from cur-bids to history.

Related Java Method:

MongoDBConnection.java: searchCurProj(), checkStatus()

jsonHelper.java: writeJsonArray (HttpServletResponse response, JSONArray array)

2.3.7 Search project by project ID

A project could be searched by project ID by using searchByProjectId API, no matter this project is validate or expired.

HTTP Method:

GET

URL (sample):

http://localhost:8080/MarketPlace/searchID?project_id=testProject

Note:

Project details will be printed.

Related Java Method:

MongoDBConnection.java: searchByProjectId (String projectId)

jsonHelper.java: writeJsonObject(HttpServletResponse response, JSONObject obj)

2.3.8 Search project by seller ID

A seller's open project could be searched by seller ID using searchBySeller API.

HTTP Method:

GET

URL (sample):

http://localhost:8080/MarketPlace/searchseller?seller_id=testSeller

Note:

Only open projects will be searched. Expired project will not be present.

Related Java Method:

MongoDBConnection.java: searchBySeller(String sellerId), findProjectIds(String sellerId)

jsonHelper.java: writeJsonArray (HttpServletResponse response, JSONArray array)

2.3.9 Search project by buyer ID

A buyer's project history could be searched by buyer ID.

HTTP Method:

GET

URL:

http://localhost:8080/MarketPlace/searchbuyer?buyer_id=testBuyer03

Note:

Only project history (“history” column) will be searched.

Related Java Method:

MongoDBConnection.java: searchByBidder(String buyerId), findBidIds(String buyerId)

jsonHelper.java: writeJsonArray (HttpServletResponse response, JSONArray array)

2.3.10 Search project by project category

Open projects could be searched by category using searchByCategory API.

HTTP Method:

GET

URL (Sample):

http://localhost:8080/MarketPlace/searchcategory?category=Date+Engineering

Note:

Only open projects in this category will be searched.

Related Java Method:

MongoDBConnection.java: searchByCategory (String category)

jsonHelper.java: writeJsonArray (HttpServletResponse response, JSONArray array)

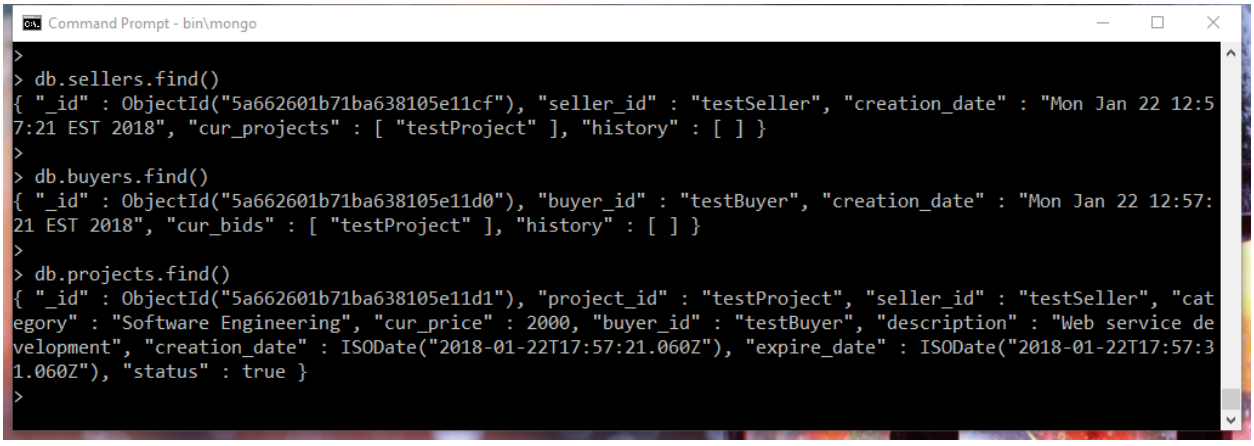
3. Assessment Testing.

In this section, each function of market place will be tested.

3.1 Table creation test

Three tables will be created when DBTableCreation.java is run. A test seller, a test buyer and a test project will be inserted with table creations.

Database screenshot:

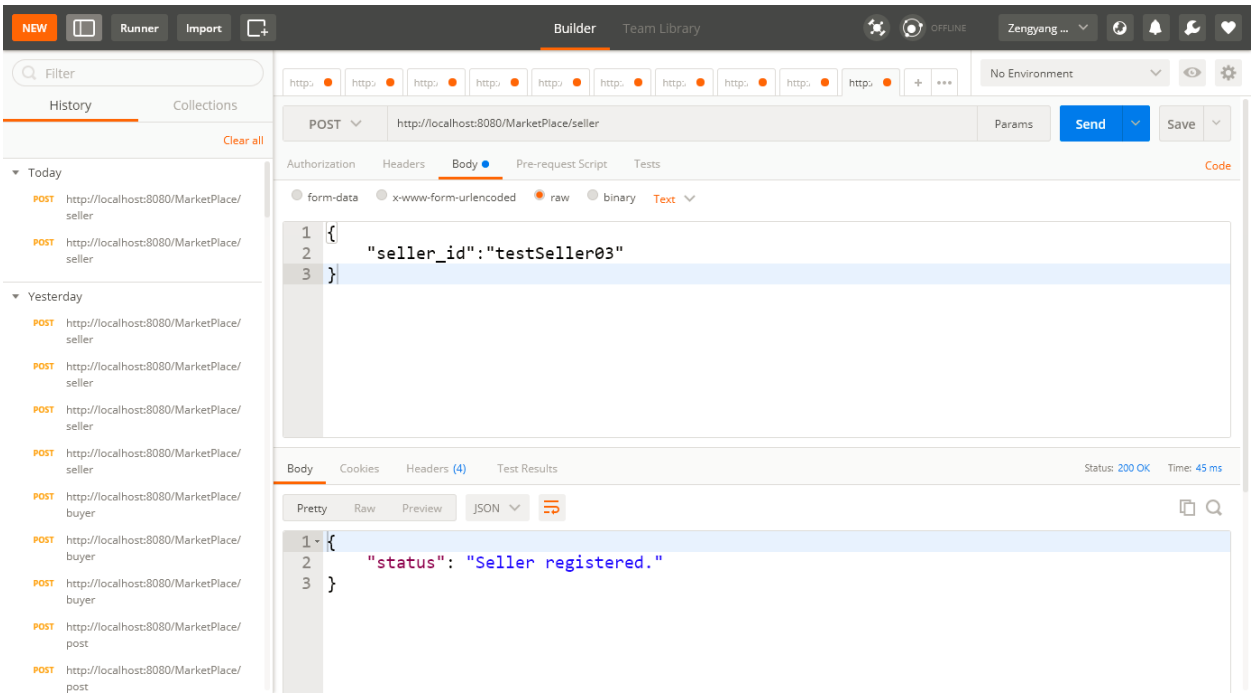


3.2 Seller registration test

Postman is used to send POST request. By using sellerRegister API, two more sellers, testSeller02 and testSeller03, are registered.

Postman screenshot:

Return status: seller registered.



Database screenshot:

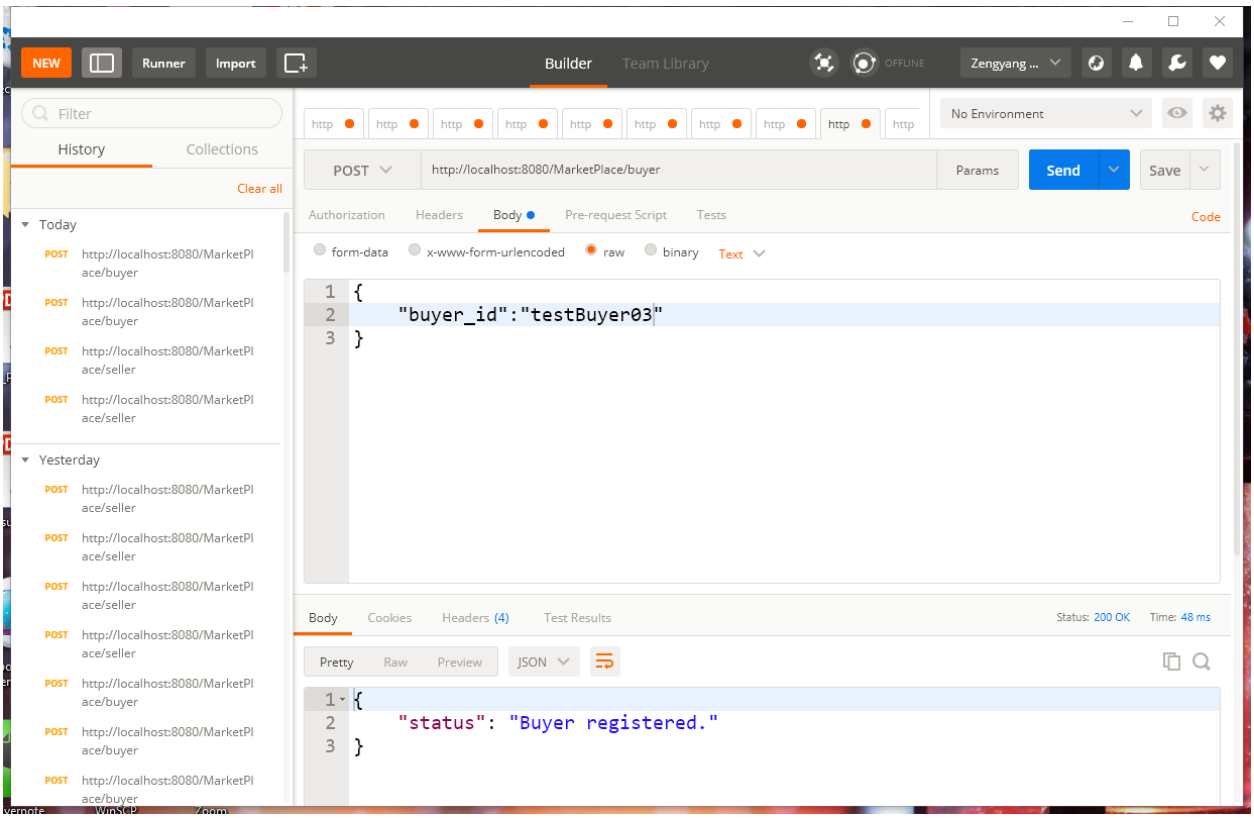
Two more sellers, testSeller02 and testSeller03 were registered.

3.3 Buyer registration test

Postman is used to send POST request. Two more buyers, testBuyer02 and testBuyer03 were registered in this step.

Postman screenshot:

Return status: buyer registered.



Database screenshot:

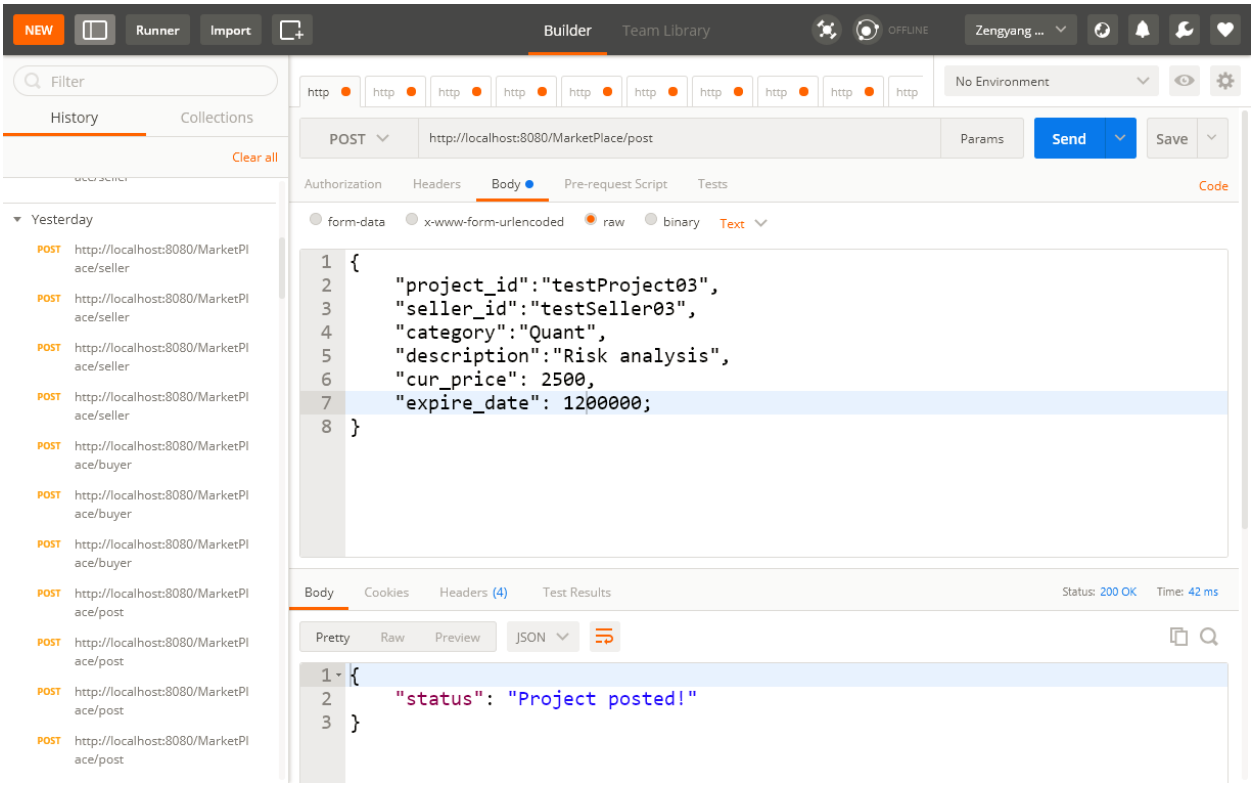
Two more buyers, testBuyer02 and testBuyer03 were registered.

3.4 Project posting test

Postman is used to send POST request. Two more projects, testProject02 by testSeller02, and testBuyer03 by testSeller03, were posted in this step.

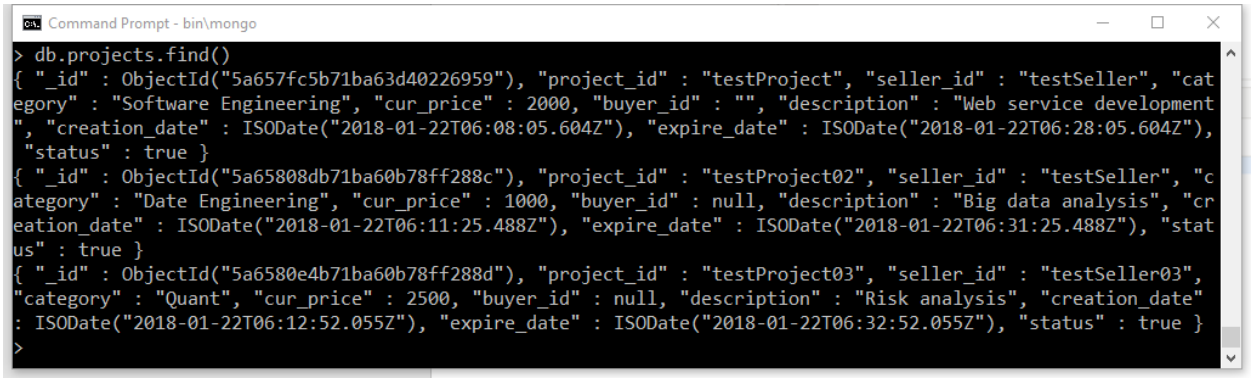
Postman screenshot:

Return status: project posted!



Database screenshot:

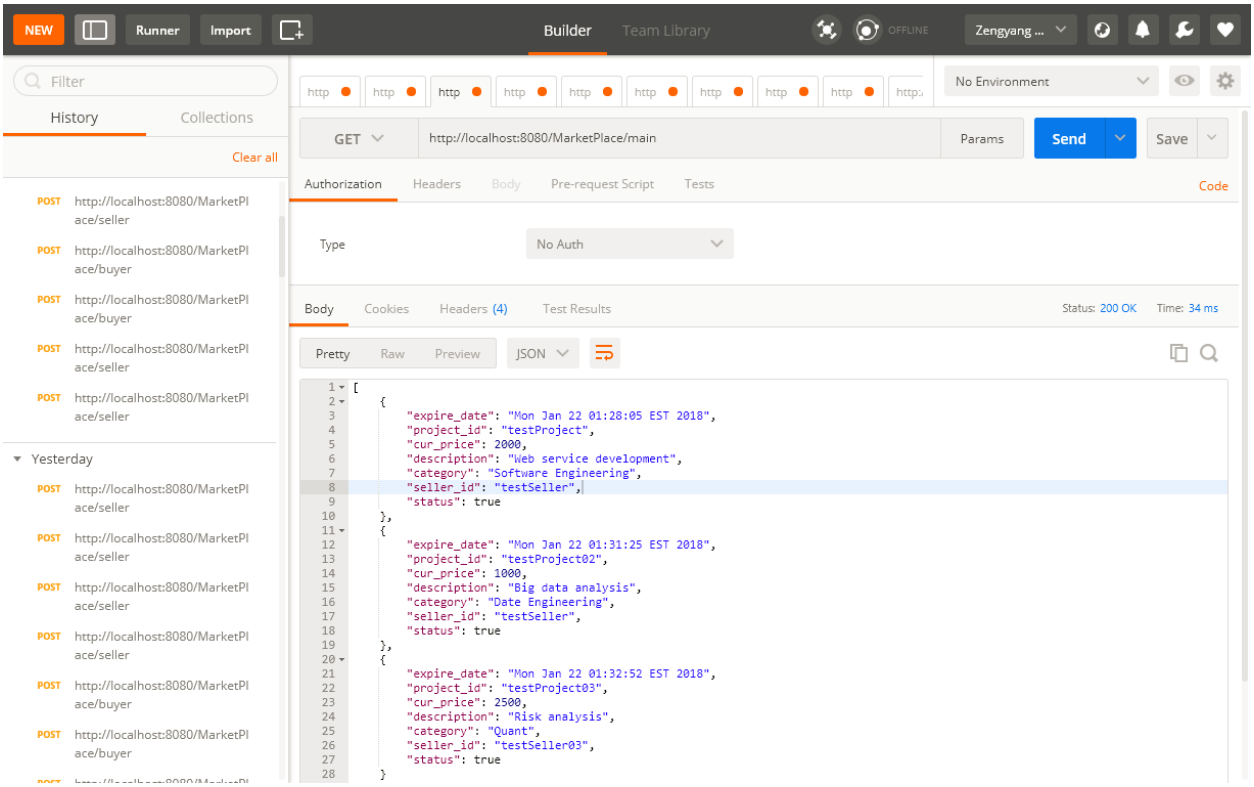
Two more projects, testProject02 by testSeller02, and testBuyer03 by testSeller03, were posted.



3.5 Search current open projects test

Postman screenshot:

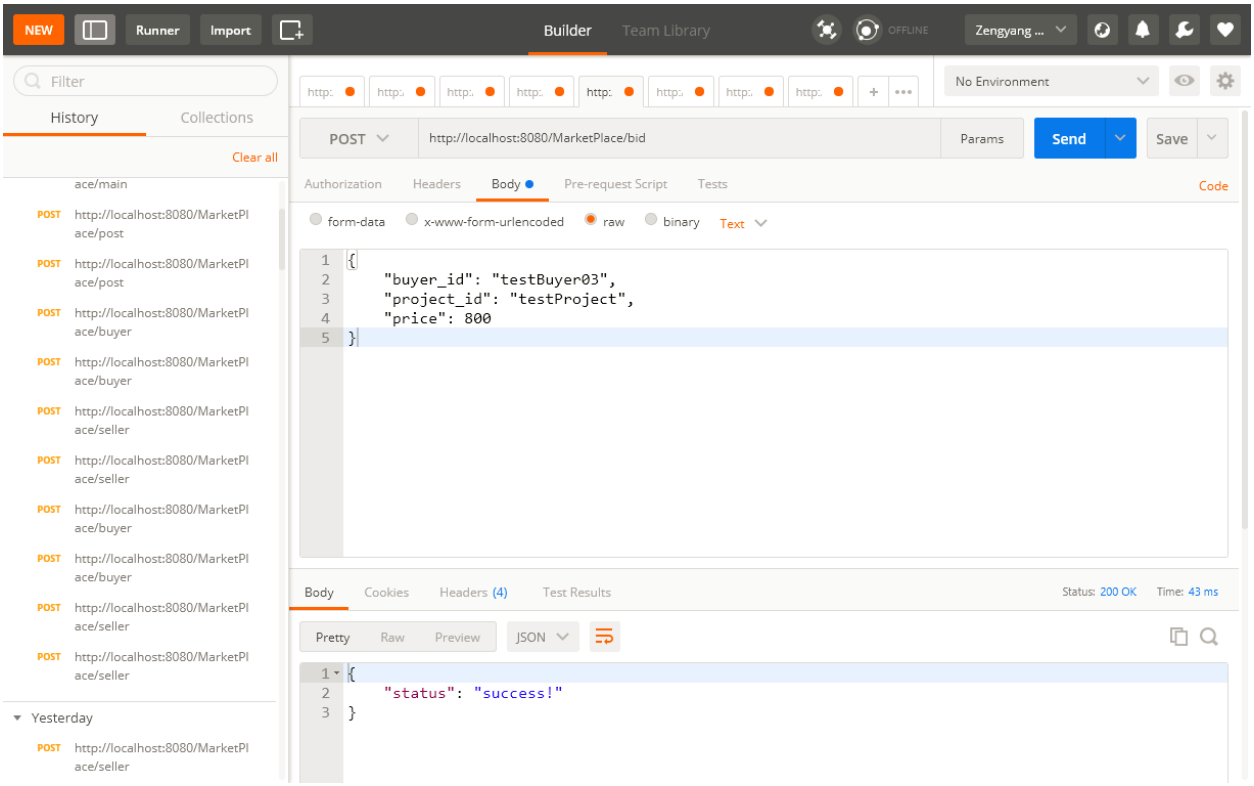
Current open projects will be printed in JSON. Open projects are projects whose status are true.



3.6 Bid for a project test

testBuyer03 bided for testProject, testBuyer02 bided for testProject03

Postman screenshot:



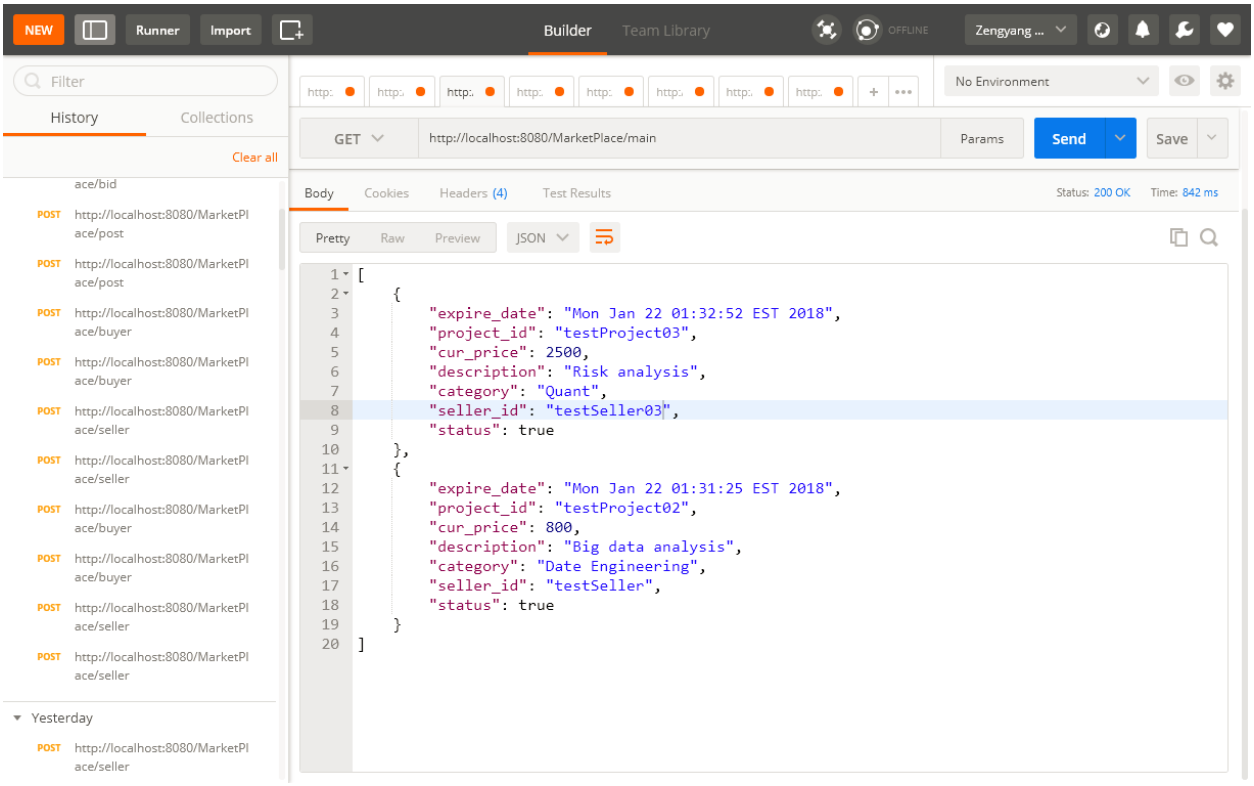
Database screenshot:

Now testBuyer02 and testBuyer03 bid for testProject02 and testProject, respectively.

3.7 **Project expiration test**

Postman screenshot:

Now testProject has expired. It will not be printed at main page.



Database screenshot:

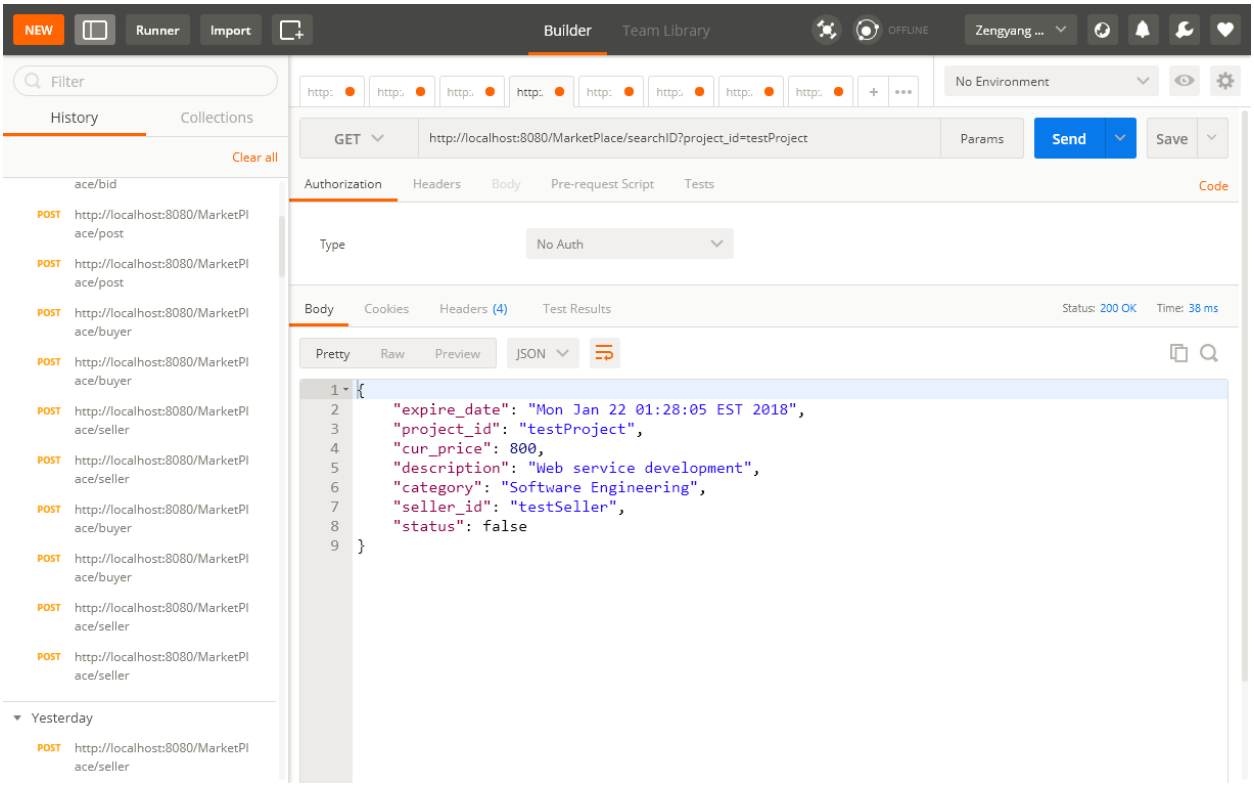
In “sellers” table, for testSeller, the project ID “testProject” has been moved to history from cur_projects.

In “buyers” table, for testBuyer03, the project ID “testProject” has been moved to history from cur_bids.

3.8 Search project by project ID test

Postman screenshot:

Project “testProject” has expired, but it could still be searched by using searchByProjectId API.

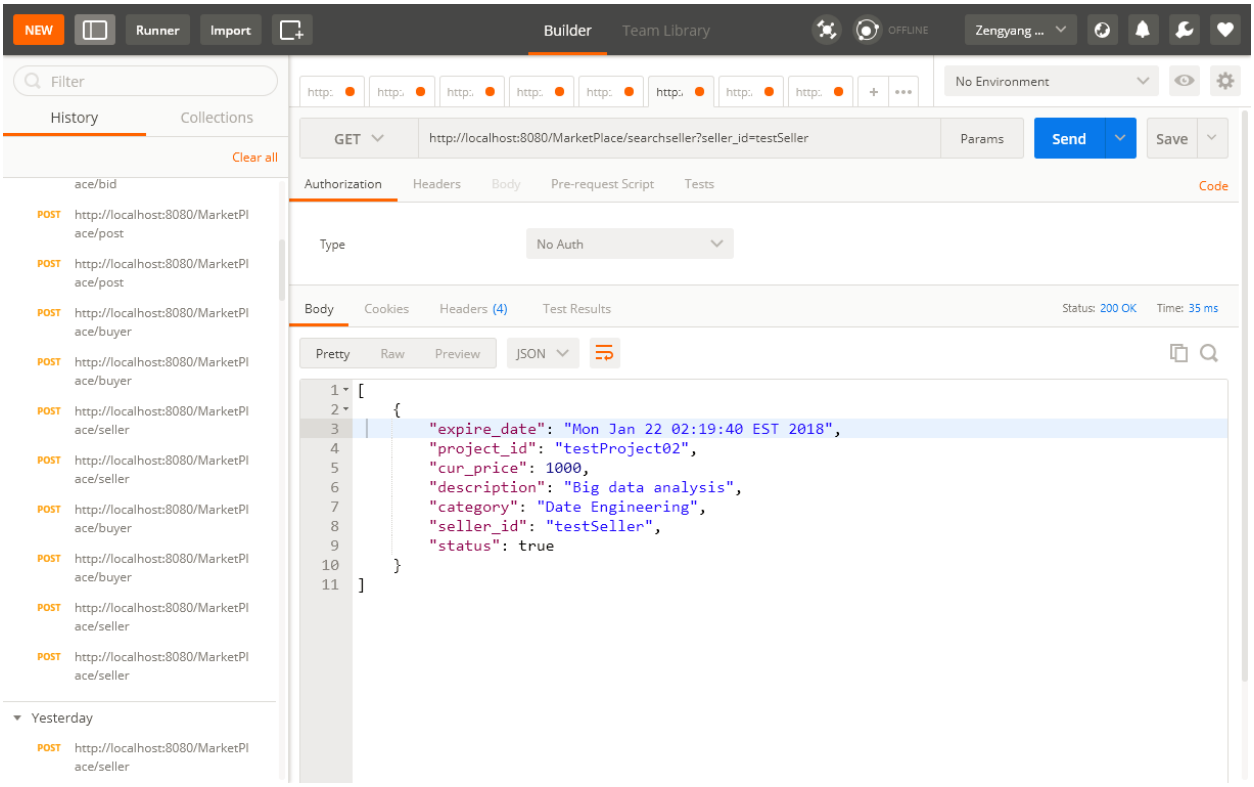


3.9 Search project by seller ID test

Only open projects will be searched. Expired project will not be present.

Postman screenshot:

Searched for testSeller



Database screenshot:

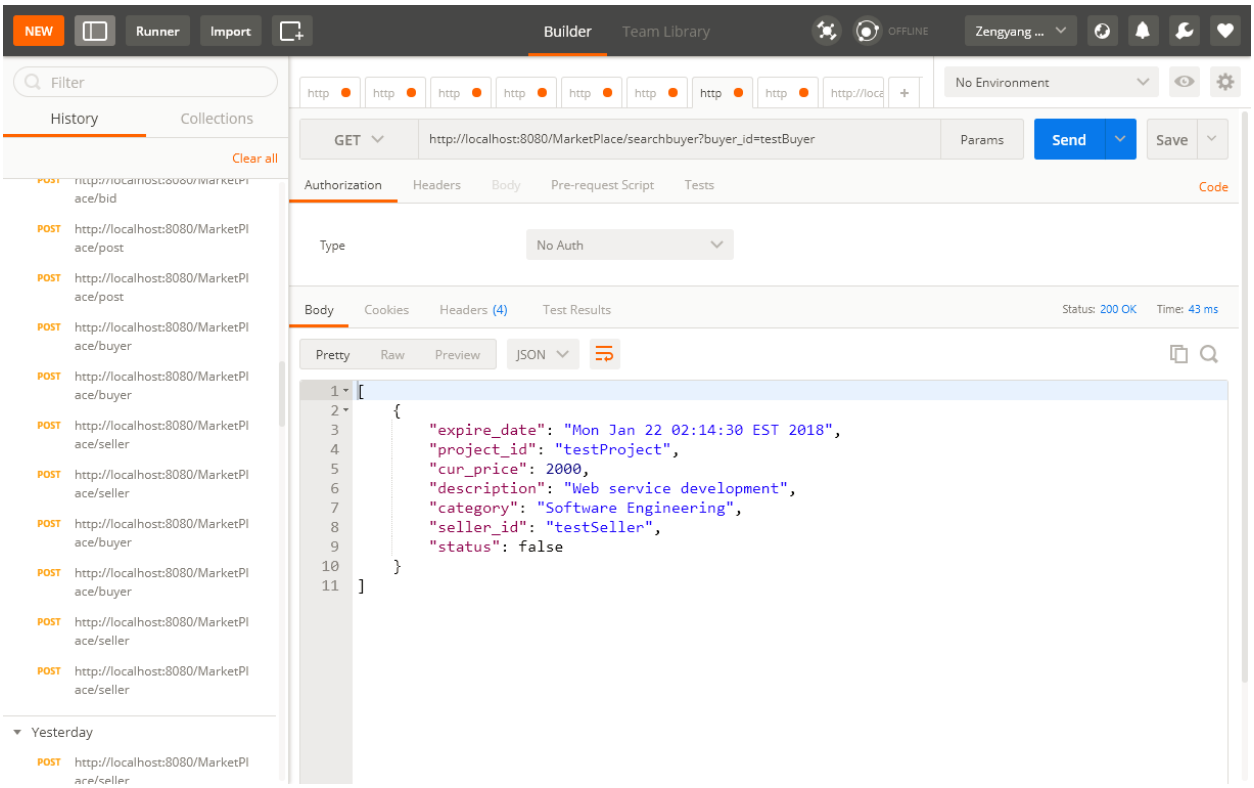
Only project “testProject02” is printed.

3.10 Search project by buyer ID test

Only buyer's project history ("history" column) will be searched.

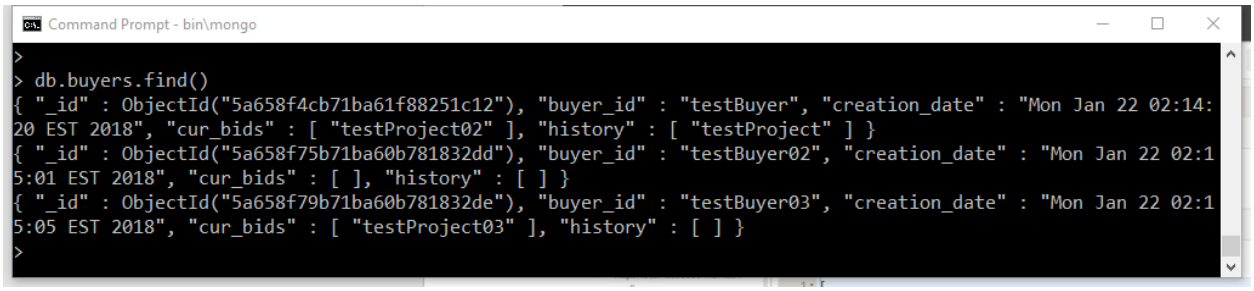
Postman screenshot:

Searched for testBuyer



Database screenshot:

Only project "testProject" is found when searching "testBuyer".



3.11 Search project by project category

Only open projects will be searched.

Postman screenshot:

Open projects in category “Data Engineering” are printed.