Peer-graded Assignment: Capstone Assignment 4.2 – Connect app to Elasticsearch

In this assignment you will be challenged to fully connect the app to a remote Elasticsearch server.

How to create your assignment

The provided code base already allows you to store and access both user and item data remotely using Elasticsearch. In this assignment, you will modify the code to implement remote data storage and access for bids.

To accomplish this, you will find it useful to study how Elasticsearch has been implemented for items and users.

Specifically, you will need to implement the following methods in the ElasticSearchManager class:

- RemoveBidTask() - Delete bid from remote server using bid_id
- AddBidTask() - Add bid to remote server
- GetBidListTask() - Returns all remote bids from server

Modify the command classes:

- AddBidCommand
- DeleteBidCommand

Modify the BidList class:

- Add getRemoteBids()
- Remove loadBids()

Modify the following methods in the BidListController class

- addBid()
- removeBid()
- removeItemBids()

- Add getRemoteBids()
- Remove loadBids()

Finally, you will modify the following views and adapter:

- ViewItemActivity
- ViewItemBidsActivity
- ItemFragmentAdapter

Before running the code, you must change the value of INDEX in the ElasticSearchManager. You will lose marks if you fail to do this.

```
/**
* For remote machine: SERVER = "http://34.202.206.222:8080"
* ----------------------------------------------------------------------------
* curl -XDELETE 'http://34.202.206.222:8080/INDEX' - can be used to delete all
* objects on the server (items, users, and bids) at that index
* view an item at: http://34.202.206.222:8080/INDEX/items/item_id
* view a user at: http://34.202.206.222:8080/INDEX/users/user_id
* view a bid at: http://34.202.206.222:8080/INDEX/bids/bid_id
* Where INDEX is replaced with the random number string you generate as per the
* assignment instructions. Note item_ids and user_ids are printed to the log
* (See the Android Monitor) as each user/item is added.
```

You can run the following curl command in your terminal to select a random number to use as an index.

curl 'https://www.random.org/integers/?num=1&min=1000000&max=1000000000&col=1&base=10&format=plain&rnd=new

'

Note: If you do not already have curl, you may download it from your terminal as follows:

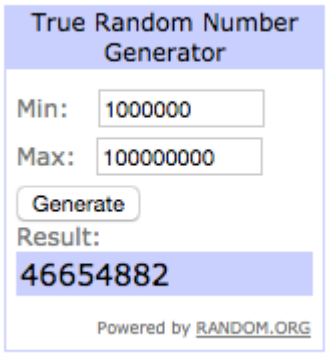
- Linux users: sudo apt-get install curl
- Mac users: brew install curl
- Windows users: please follow this tutorial https://develop.zendesk.com/hc/en-us/articles/360001068567-Installing-and-using-cURL#curl_win

After you have run this command, edit your index value correspondingly. In the end, you will have something like:

INDEX = “565706608”

Do not use this above value -- is is for illustrative purposes only. **Again, your index must be unique. You will lose marks if you fail to do this.** If you use the same index as another learner you will be able to add, edit, and delete another learner’s data, which would make debugging your code a nightmare.

- If you are unable to install curl, you may alternatively visit https://www.random.org/
- in a web browser to generate a random number. The site provides a graphical random number generator. Use min = 1000000, max = 100000000.



Hint: Read the comments near the top of the ElasticSearchManager file to learn how to view and delete server contents for your index by running a command in your terminal. For debugging purposes, it will be useful for you to be able to delete all the items, users, and bids stored remotely. Additionally, it may be helpful to view the server contents.

Submission Instructions

When you are ready to submit your code, include the following files in a folder:

- **ElasticSearchManager.java**
- **AddBidCommand.java**
- **DeleteBidCommand.java**
- **BidList.java**
- **BidListController.java**
- **ViewItemActivity.java**
- **ViewItemBidsActivity.java**
- **ItemFragmentAdapter.java**

Compress the folder into a ZIP folder. Windows users can use 7zip or WinRAR. Upload it where prompted.