Vert.x Lab

How to submit

1. **Create git repository ;**
2. **Upload**

**repoName : yourName-yourid-vertxlab**

**-lab-1**

**Functional Programming**

Time : 15 mins

1.Create Login class, which should have method called validate.

Rules:

1. validate method must take two two args
2. The first arg is one functional Interface called “Resolve”, method name you can decide
3. The Second args is another functional interface “Reject”, method name you can decide
4. Write a biz logic to validate username and password; you can hardcode username and password values.
5. If validation is success, send “Login success” else “Login failed”.

Time : 30 mins

2.Implementation of functional application

Build Customer-Manager App using functional approach

* Repository
  + Have repository interface having methods findAll,findById,save,update,remove
  + Have repository implementation
  + Results Should be handled by Functional Interfaces like Consumer, Supplier.
  + Use Array List to store,update,remove,findAll Customer data
  + Customer Entity should be id,firstName,lastName,city.
* MainApp
  + Call apis for findAll, findbyid, saving, delete, update

**Reactive Programming**

**Time : 10 mins**

3 .Create an Observable Stream which should emit sequence of Employee Objects.

**Time : 6 mins**

3 .Create an Observable Stream which should emit sequence of Employee Objects**, Each Employee name must be transformed into uppercase**

**Time :8 mins**

**4.** Create an Observable Stream which should emit sequence of Employee Objects, **list all employees who are onsite** in upper case. Note : add new property in Employee Entity called “onsite=true”. Populate 5 employees.

**Time : 6 mins**

5.Create Employee Object, Departement, EmployeeDepartment**, zip data from Employee, Department, combine them , return new EmployeeDepartment Object.**

**Vert.x**

**Future and Promise**

**Time : 30 mins**

**6.Create login api , which should accept username and password as parameter.**

1. You have to return “Login Success” and “Login failed” message when username and password matches to “admin” and “admin”.
2. You have to handle result using onComplete,onSuccess, onFailure,function as a parameter syntax, you can return Future via “Future.future()” , and “static factory api”.
3. Later you have to provide Promise version as well.

**Time : 15mins**

**7.Write a simple simulation of callback hell and how to avoid using Future.compose method**

1. Api name is prepareDatabase- which should return Boolean called isConnected. If it is connected ,
2. Second api is startWebServer which should get status from prepareDatabase , startWebServer should have Boolean flag called isServerReady, if it is true
3. Call third api called startWebContainer which should get status from

If all servers are up, you have to print “All servers are up” else which server is down , you print and terminate flow.

Implementation:

1.Use callback hell Syntax

2.Start refactoring using compose method with all flavours.

**Time : 6 mins**

**8.** You have to return message json to client with some delay may be after 5000ms.Create simple web end point to return data to client.

**Tip: use timer.**