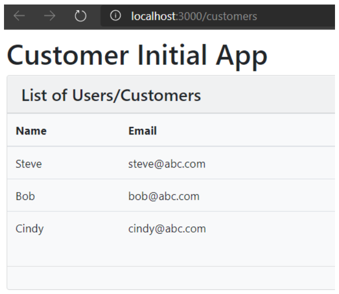
Sprint 1 (Day 2/2)

1. Design a build process using Gradle that will be used to build your REST services
2. Create classes for the three domain objects mentioned in the Domains section of this documentation.
3. Create an in-memory repository/persistence layer. Ensure you have a repository/persistence class for each domain object.
    1. Create methods to support only the **findAll** and **findById** operations (i.e. read-only operations. In this sprint, you don't need to support the rest of the CRUD operations, such as insert, update, and delete.)
    2. Hard-code 2-3 sample records in each repository.
4. Create a Service layer (both the interface and implementation) for each repository.
    1. Use the auto-wiring feature to inject the repository
5. Create the three APIs as mentioned in the APIs section of this documentation.
6. Test all APIs using Postman.
7. Test all APIs using the provided react-client application.
    1. Run the client application using the command: **npm start**
    2. localhost:3000/customers should the following page (Note: Your data can be different)

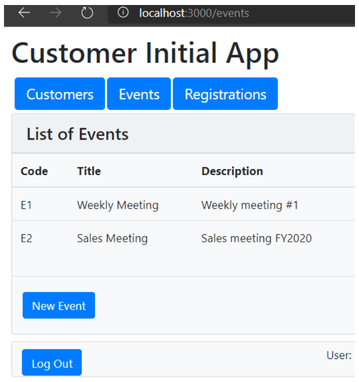


Customer Initial App

List of Users/Customers

Name	Email
Steve	steve@abc.com
Bob	bob@abc.com
Cindy	cindy@abc.com

    3. localhost:3000/events should the following page:

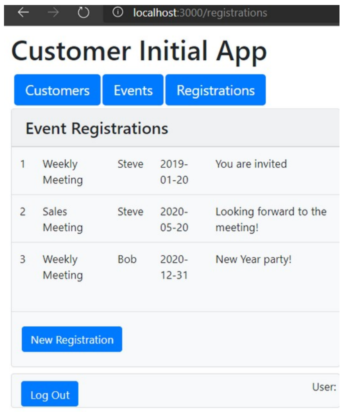


Customer Initial App

List of Events

Code	Title	Description
E1	Weekly Meeting	Weekly meeting #1
E2	Sales Meeting	Sales meeting FY2020

    4. localhost:3000/registrations should show the following page: (Note: you can also click Customers, Events, and Registrations button on the page).


localhost:3000/registrations

Customer Initial App

Event Registrations

1	Weekly Meeting	Steve	2019-01-20	You are invited
2	Sales Meeting	Steve	2020-05-20	Looking forward to the meeting!
3	Weekly Meeting	Bob	2020-12-31	New Year party!

8. Prepare to explain and demonstrate the service and build process
9. Commit the completed code to your local repository and also push it to the Github remote repository.

Domains

Customers

Property	Type
id	long
name	String
password	String
email	String

Events

Property	Type
id	long
code	String
title	String
description	String

Registrations

Property	Type
id	long
event_id	long
customer_id	long
registration_date	Date
notes	String

APIs

1. Configure the following in application.properties.
    1. Use /api as the context-path
    2. Use 8080 as the port
2. Customers
    1. endpoint (complete URL http://localhost:8080/api/customers)
    2. support the following operations: (**Note: Data must be returned in JSON format)**
        1. GET
            1. get all records. (/api/customers)
            2. get customer by id. (/api/customers/{id})
            3. get customer by name (/api/customers/byname/{name})
    3. Here's a sample JSON:
       [{"id":1,"name":"Steve","password":"pass","email":"steve@abc.com"},
       {"id":2,"name":"Bob","password":"pass","email":"bob@abc.com"},
       {"id":3,"name":"Cindy","password":"pass","email":"cindy@abc.com"}]
3. Events
    1. endpoint (complete URL http://localhost:8080/api/events)
    2. support the following operations: (**Note: Data must be returned in JSON format)**
        1. GET
            1. get all records. (/api/events)
            2. get customer by id. (/api/events/{id})
    3. Here's a sample JSON
       [{"id":1,"code":"E1","title":"Weekly Meeting","description":"Weekly meeting #1"},
       {"id":2,"code":"E2","title":"Sales Meeting","description":"Sales meeting FY2020"}]
4. Registrations
    1. endpoint (complete URL http://localhost:8080/api/registrations)
    2. support the following operations: (**Note: Data must be returned in JSON format)**
        1. GET
            1. get all records. (/api/registrations)
            2. get registration by id. (/api/registrations/{id})
    3. Here's a sample JSON
       [{"id":1,"event_id":1,"customer_id":1,"registration_date":"2019-01-20T05:00:00.000+0000","notes":"You are invited"},
       {"id":2,"event_id":2,"customer_id":1,"registration_date":"2020-05-20T04:00:00.000+0000","notes":"Looking forward to the meeting!"},
       {"id":3,"event_id":1,"customer_id":2,"registration_date":"2020-12-31T05:00:00.000+0000","notes":"New Year party!"}]