**Repository Location:** [imhardik14/phonebook (github.com)](https://github.com/imhardik14/phonebook)

**Time Spent:** 3.5 hours

**Technical Stack**

* Oracle Java 8
* Spring boot 2.6.3
* H2 in-memory DB
* Spring JPA
* Spring security for basic authentication

**Setup**

* Checkout project and import as existing maven project in the STS
* Run project with spring boot dashboard

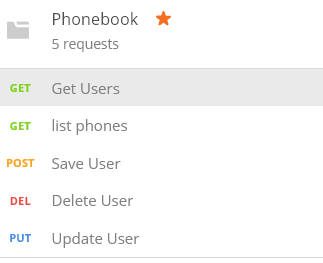
**Run Locally**

* Checkout project and go the cmd where pom.xml
* Run ***mvn clean install*** command
* It should create phonebook-0.0.1-SNAPSHOT.jar <project\_root>/target folder
* move to target folder in cmd
* run below command - ***java -jar phonebook-0.0.1-SNAPSHOT.jar***
* it should run system on localhost:8080

**Test Locally**

Once server is in running mode – you can import this Postman collection and it is having below Operations. You need to use basic authentication for performing below operation in postman

https://github.com/imhardik14/phonebook/blob/main/Phonebook.postman\_collection.json

otherwise you can use below URL and method

GET - http://localhost:8080/user/list - list of users

GET - http://localhost:8080/user/phones/{userId} - Get list of Phones

POST - <http://localhost:8080/user/>{userId} – Save User

PUT - <http://localhost:8080/user/>{userId} –Update User

DELETE - <http://localhost:8080/user/>{userId} – Delete User

**In-memory Spring Security User**

|  |  |  |
| --- | --- | --- |
| **Username** | **Password** | **Role** |
| user | password | USER |
| admin | admin | ADMIN |

**Current Working Features**

1. List of users
2. Find by username
3. Add User to system
4. Update user to system
5. Delete user from system
6. User Controller Junit test cases for get user list and save user
7. Logging configuration

**Current Issues**

* While saving or updating User object – it is not storing child phones object

**Next Scope**

1. Swagger implementation
2. Spring security implementation using user entity
3. UI capabilities
4. Role based authorization
5. Advanced unit test cases implementation
6. Validation
7. Password store with encryption
8. Exception handling