Java Coding Exercise

Solutions Development

Exported on Feb 13, 2019

Table of Contents

1 Introduction

This document will provide details about a coding assignment to be given to Backend Java Developer who have passed the phone interview as an assessment of their coding and design skills.
Based on the candidate level the requirement will be reduced or template project will be provided.

The candidate will be evaluated based on the **submitted project** (implementation completeness, code quality, documentation) and the **in-person presentation** of his work during which related technical questions will be asked.

2 Task Delivery

The following will be provided to the candidate, both should be attached on this document:

- a pdf documentation with the requirements, guidelines and deliverable results.
- a zip file of the template project

3 Requirements

Listed below is the requirement of the project to be delivered, required features are expected to be complete. Candidate who will also implemented optional features will be given more chance.

3.1 Required features

Create a **Maven** project to implement a simple **Spring** based **web application** with a controller exposing the following 4 **APIs**

- GET /user
- POST /user
- PUT /user
- DELETE /user

Those APIs will perform CRUD operations using a persistence framework on a User entity with at minimum the following fields:

- id (long) / primary key
- username (String) / unique
- password
- status / possible values: Activated/Deactivated

Secure those 4 APIs with **Spring Security** using BASIC authentication.

Provide **unit tests** for controllers, services and DAO classes.

The project should be hosted in local **GIT** repository. Please make sure you have committed all of your changes in the Git.

3.2 Optional features

Configure hibernate to use Hazelcast as second level cache

Secure service layer methods

4 Guidelines

- You can use any IDE of your choice (IntelliJ, Eclipse, Netbeans ...)
- You are free to use choose between Spring Annotation or XML configuration
- You are free to use choose between Hibernate Annotation or XML configuration
- For simplicity use an in-memory DB like (HSQLDB or H2)
- Also for simplicity use jetty web server (You can use Spring boot , the embeded webserver, or tomcat)
- Please follow the provided project structure
- The submitted project will be evaluated for implementation completeness, code quality, documentation)
- Be prepared to discuss any part of the application during the in-person interview.

5 Deliverable Results

A zip file of the project containing the source code and .git folder (local GIT repo with a proper history of all commits), without build artifacts and compiled classes. The project should not be checked into any of the remote repository sites.

Document important configurations, references or design concerns in the README.md file.

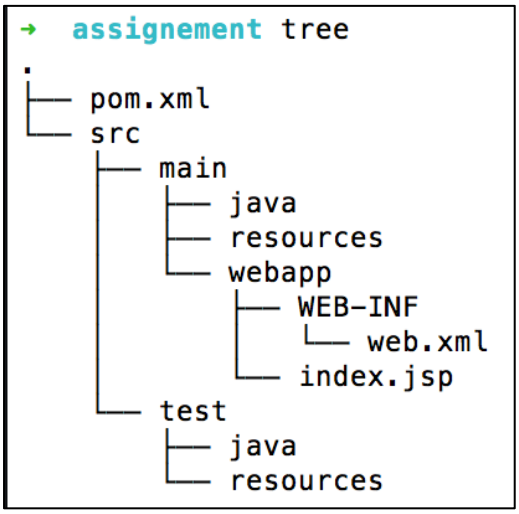
The project should be easily locally executable without any extra configuration with the following commands:

```
mvn clean install
mvn jetty:run (or if you use any other container please provide the
comment how to run and test)
```

The accessed in the browser with this url: http://localhost:9090/assignement

6 Template Project

A web application project based on Maven standard directory layout is provided:



```
➜  assignement tree
.
├── pom.xml
└── src
    ├── main
    │   ├── java
    │   ├── resources
    │   └── webapp
    │       ├── WEB-INF
    │       │   └── web.xml
    │       └── index.jsp
    └── test
        ├── java
        └── resources
```

6.1 POM file

Contains a barely empty **war packaging** POM file without any dependency or plugin.

```
<project xmlns="http://maven.apache.org/POM/4.0.0"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
 xsi:schemaLocation="http://maven.apache.org/POM/4.0.0 http://maven.apache.org/maven-v4_0_0.xsd">
    <modelVersion>4.0.0</modelVersion>
    <groupId>com.uxpsystems.assignement</groupId>
    <artifactId>assignement</artifactId>
    <packaging>war</packaging>
    <version>1.0-SNAPSHOT</version>

    <build>
        <finalName>assignement</finalName>
    </build>
</project>
```

6.2 Package structure

To further emphasize the project structure, in the source folders for java test created the following package:

Package	Description
com.uxpsystems.assignement.config	package for web application configuration classes
com.uxpsystems.assignement.controller	package for Spring controller classes
com.uxpsystems.assignement.service	package for service classes
com.uxpsystems.assignement.dao	package for DAO classes

6.3 Documentation

An empty README.md file is available at the project root directory.