(Software Design Document)

Nikaylah Woody

December 10, 2019

Contents

Revisions

Date	Description	Version
November 25, 2019	Created document	1.0
November 26, 2019	Added documentation for the packages and classes	1.1
December 09, 2019	Added more in-depth explainations of functionality within application	1.2

1 Introduction

The purpose of this project is to design and implement a REST API for a website to complete a test assesment as part of my interview with Intuit.

2 Overview

This project is a marketplace mvc application for contractors. This marketplace allows clients to post projects(jobs) and allows contractors to bid for these projects.

3 Implemented Components

/controllers

(a) provides the implementation of the Bid, Contractor, Project, and Index Controllers

(b) provides the model data to the view allowing button clicks and form submissions

(c) list() injects the model and returns the dependencies and requests model to the show by taking the model in and adding attributes for "projects", "contractors", and "bids" and using the service to grab the list of data

(d) show() get() uses @pathVariable which matches up the property in request mapping with the properties of the method signature

(e) edit() gets the data and renders it to the form page

(f) new() passes in an empty model

(g) saveOrUpdate() posts the most accurate match, captures data by calling method from services and saves it, comes back and redirects to the show url. It dynamically creates the id url (request comes in and Spring binds the form parameter to our data properties dynamically)

(h) delete()

/domain

(a) provides the implementation of the models and data of the Bid.java, Contractor.java, and Project.java

(b) DomainObject interface that show the id properties to use for AbstractMapService class

(c) AbstractMapService returns a DomainObject

(d) projects properties include id, version, name , description and maxBudget

(e) version property performs jpa optimistic bocking that allows a version number to be created for each new update without losing previous updates

(f) contractors properties include firstName, lastName, email, phoneNumber, and address info

(g) bids properties include name and bidAmount

/services

(a) services are used to separate our concerns within the application

(b) provides the Bid, Project, and Contractor service classes and interfaces to perform operations within the various models

(c) using CrudRepository interface with a generic type to automatically implement getById(), saveOrUpdate(), and delete() methods at runtime for Bid, Project and Contractor models

(d) listAll() using DomainObject interface and lists all of the provided data

(e) saveOrUpdate() takes in data, and if method id is null it goes out an gets the maximum value from our map and append 1 to it then puts that project in out map

(f) delete() annotates @PathVariable to bind the id from the web request. It goes into the data and removes that existing data, then redirects

(g) loadDomainObjects() provides HashMap of data

/resources

(a) contains form.html, list.html, and show.html for each model class and an index.html for the home page

/test

(a) utilizes a mock web context to test web requests and its interactions with the Spring MVC dispatcher servlet

(b) Mock requests go through the dispatcher servlet and the dispatcher servlets reacts with the contollers as it would with the web container(Tomcat)

(c) Mokito is used to test the injected dependent objects

(d) use @Mock to set up a service that gets injected into the project controller

(e) use @InjectMocks which creates a controller, and injects mock objects into it

(f) andExpect(staus()) expects it to return an http status of OK

(g) andExpect(view()) to represent where we are going

(h) andExpect(model()) to show the correct attribute

(i) status().is3xxRedirection() verifying a redirect to the object url

(j) any() takes in any class of the object

(k) ArgumentCaptor capture the argument that comes into the mockito mock, and verifies the properties of the bound object

3.1 Anticipated Updates

(a) Provide additional frontend such as jQuery to provide a more user firendly experience

(b) Provide Spring Security to allow both Clients and Contrators to log in with username and password

(c) Provide a bid currency converter that allows Clients and Contractors to bid in the currency of their choice without conflicting interferance

(d) Provide project filter feature to allow Contractors to narrow their project search by category

(e) Allow notifications feature that sends notifiation when a new project is created within the users category preferences

(f) Allow contractors to provide descriptions of their qualifications at the time of providing a bid