

Digital Home Brokering - V1

Grade settings: Maximum grade: 100

Based on: [Digital Home Brokering - V1](#)

Run: Yes **Evaluate:** Yes

Automatic grade: Yes **Maximum execution time:** 240 s **Maximum memory used:** 1.50

GiB Maximum number of processes: 10000

Digital Home Broker

[Click here to download the code skeleton](#)

Develop a web application for Digital home broker. Through this application anyone can search rented home/apartments based on location, BHK type and rent.

Create a Spring MVC Spring Boot Application for developing a Digital home broker Application. Design a search page to search homes/apartments based on location,bhkType and rent

On clicking search button, the application should list the available home/apartments based on the given search criteria. The customer has to then be redirected to homeListPage.jsp page that displays the list of available homes in a tabular format.

Application Work Flow:

- **BrokerController** is the Controller class.
- **Home** is the model class with attributes **location,address,bhkType,contactPerson,contactNumber and rent** along with its getters and setters.
- BrokerService is the Service class which has a method called **searchHome** that takes **Home** as its argument and returns **ArrayList<Home>**.
- **Attribute ArrayList<Home> list;** defined in service class and the value for this list was populated through **addHomes()** method. Don't change the code in this method.
- Through **searchHome** method, have to search the **Home** objects based on the given input location, bhkType and rent. (Iterate the list, get the Home objects which have the same location,bhkType and rent entered by customer)

Initially, the customer should be routed via the **BrokerController's searchPage** method to searchPage.jsp that allows user to choose the location, bhkType and rent.

[Note: searchPage method has to be written inside the **BrokerController]**

A method in the **BrokerService** known as **addHomes** is already available. which is used to add the **Home** objects into the **ArrayList<Home>**

On clicking the program search button, the **BrokerController's searchHome** method should be called. This method takes three arguments - model attribute named **homeBean** which holds the form populated **Home** Object, **BindingResult** and the **ModelMap**.

- This method should search the Home objects from the list based on the location, bhkType and rent. **(Should be case insensitive search for location and bhkType)**

Screen designs and Expected Output:

localhost:9088/searchPage

Digital Home Broker

Search homes|Apartments !!!

Search location	<input type="text" value="Chennai"/>
Search BHK Type	<input type="text" value="2bhk"/>
Preferred Rent	<input type="text" value="14000"/>
<input type="button" value="Search"/>	

If the rent entered by the customer is less than 5000 then an error message "**Minimum rent should be 5000**" has to be displayed in the searchPage.jsp. To do this validation use appropriate annotation above **rent** attribute in **Home** model class.

searchPage.jsp

localhost:9088/searchPage

Digital Home Broker

Search homes|Apartments !!!

Search location	<input type="text" value="Chennai"/>
Search BHK Type	<input type="text" value="2bhk"/>
Preferred Rent	<input type="text" value="14000"/>
<input type="button" value="Search"/>	

homeListPage.jsp

localhost:9088/homeList			
<h1>Digital Home Broker</h1> <h2>Search homes Apartments !!!</h2>			
Location	BhkType	Address	ContactPerson
Chennai	2BHK	#212/4, 10th cross, T-nagar, PIN-60008	Annesh
Chennai	2BHK	#556/4, 5th cross, Tambaram, PIN-60010	Jisha
Chennai	2BHK	#556/4,Nila paradise, Tambaram, PIN-60010	Jisha

- If there are no Home objects available based on the search then display the message **“No home/Apartment found for this search”** in **homeListPage.jsp**

localhost:9088/homeList			
<h1>Digital Home Broker</h1> <h2>Search homes Apartments !!!</h2> <h3>No home/Apartment found for this search</h3>			

Design Constraints:

UI Design Constraints:

searchPage.jsp		
Component	ID	Constraints
Textbox	location	Name and id attribute should be “location” •
Textbox	bhkType	Name and id attribute should be “bhkType” •
Textbox	rent	Should not be less than 5000
submit	submit	Name and id attribute value should be submit

Component Specification

Controller

BrokerController			
AttributeName	AttributeType	Access Specifier	Constraints
BrokerService	service	private	Use annotation to Autowire

BrokerController			
Method Name	Method Argument name:type	Return type	RequestMapping URL & Request Method
searchPage	modelAttribute @ModelAttribute • :Home	String	/searchPage & GET
searchHome	modelAttribute @ModelAttribute • :Home, result:BindingResult, model:ModelMap	String	/homeList & POST

Service:

BrokerService		
Method Name	Method Argument name:type	Return type
searchHome	homeBean:Home	ArrayList<Home>

Model:

Home	
AttributeName	AttributeType
location	String
address	String
bhkType	String
contactPerson	String
contactNumber	String
rent	double

Overall Design Constraints:

- **BrokerController** class should be inside the package **com.controller**

- **BrokerService** class should be inside the package **com.service**
- **Home** class should be in the package **com.model**
- Use appropriate annotation to configure **BrokerService** as a Service
- Use appropriate annotation to configure **BrokerController** as a Controller
- **BrokerService** should be autowired inside the **BrokerController**.
- Use annotations to implement the business Validation as specified in the screenshot [That is when the rent is less than 5000 then error message "**Minimum rent should be 5000**" should be rendered in the UI.]
- Do not change the property name given in the application.properties files, you can change the value and you can include additional property if needed.
- In the pom.xml you are provided with all the dependencies needed for developing the application.
- You will not be evaluated based on the UI design (layout, color, formatting, etc.). You are free to have a basic UI with all the required UI components (input fields, buttons, labels, etc.). Expected components with the id alone should be designed as per the requirement.
- Adhere to the design specifications mentioned in the case study.
- **Do not change or delete the class/method names or return types that are provided to you as a part of the base code skeleton.**
- Please make sure that your code does not have any compilation errors while submitting your case study solution.

Result Description

Submitted by sathya .

Automatic evaluation[+]

DigitalHomeBroker/pom.xml

```

1 <?xml version="1.0" encoding="UTF-8"?>
2 <project xmlns="http://maven.apache.org/POM/4.0.0" xmlns:xsi="http://www.w3.org/2001/XMLSchema-
instance"
3     xsi:schemaLocation="http://maven.apache.org/POM/4.0.0 https://maven.apache.org/xsd/maven-
4.0.0.xsd">
4     <modelVersion>4.0.0</modelVersion>
5     <parent>
6         <groupId>org.springframework.boot</groupId>
7         <artifactId>spring-boot-starter-parent</artifactId>
8         <version>2.1.8.RELEASE</version>
9         <relativePath/> <!-- lookup parent from repository -->
10    </parent>
11    <groupId>com.example</groupId>
12    <artifactId>DigitalHomeBroker</artifactId>
13    <version>0.0.1-SNAPSHOT</version>
14    <name>FlawlessAcademy</name>
15    <description>Demo project for Spring Boot</description>
16    <properties>
17        <java.version>1.8</java.version>
18    </properties>
19
20    <dependencies>
21        <dependency>
22            <groupId>org.springframework.boot</groupId>

```

```

23             <artifactId>spring-boot-starter-web</artifactId>
24         </dependency>
25
26         <dependency>
27             <groupId>org.apache.tomcat.embed</groupId>
28             <artifactId>tomcat-embed-jasper</artifactId>
29             <scope>provided</scope>
30         </dependency>
31
32         <dependency>
33             <groupId>javax.servlet</groupId>
34             <artifactId>servlet-api</artifactId>
35             <version>2.5</version>
36             <scope>provided</scope>
37         </dependency>
38         <dependency>
39             <groupId>javax.servlet.jsp</groupId>
40             <artifactId>jsp-api</artifactId>
41             <version>2.1</version>
42             <scope>provided</scope>
43         </dependency>
44
45         <dependency>
46             <groupId>taglibs</groupId>
47             <artifactId>standard</artifactId>
48             <version>1.1.2</version>
49         </dependency>
50         <dependency>
51             <groupId>javax.servlet</groupId>
52             <artifactId>jstl</artifactId>
53             <version>1.2</version>
54         </dependency>
55         <!-- Spring test dependencies -->
56         <dependency>
57             <groupId>org.mockito</groupId>
58             <artifactId>mockito-all</artifactId>
59             <version>1.10.19</version>
60             <scope>test</scope>
61         </dependency>
62
63         <!-- <dependency>
64             <groupId>org.seleniumhq.selenium</groupId>
65             <artifactId>selenium-java</artifactId>
66             <version>2.53.0</version>
67         </dependency>-->
68
69         <dependency>
70             <groupId>org.seleniumhq.selenium</groupId>
71             <artifactId>htmlunit-driver</artifactId>
72             <version>2.26</version>
73         </dependency>
74
75         <!-- https://mvnrepository.com/artifact/org.w3c.css/sac -->
76         <dependency>
77             <groupId>org.w3c.css</groupId>
78             <artifactId>sac</artifactId>
79             <version>1.3</version>
80         </dependency>
81
82
83         <dependency>
84             <groupId>org.springframework</groupId>
85             <artifactId>spring-test</artifactId>
86             <version>5.1.7.RELEASE</version> <!-- 4.0.5 -->
87             <scope>test</scope>
88         </dependency>
89         <!-- End of spring test dependencies -->

```

```

90
91
92
93         <dependency>
94             <groupId>org.springframework.boot</groupId>
95             <artifactId>spring-boot-starter-test</artifactId>
96             <scope>test</scope>
97             <exclusions>
98                 <exclusion>
99                     <groupId>org.junit.vintage</groupId>
100                     <artifactId>junit-vintage-engine</artifactId>
101                 </exclusion>
102             </exclusions>
103         </dependency>
104     </dependencies>
105
106     <build>
107         <plugins>
108             <plugin>
109                 <groupId>org.springframework.boot</groupId>
110                 <artifactId>spring-boot-maven-plugin</artifactId>
111             </plugin>
112         </plugins>
113     </build>
114
115 </project>
116

```

DigitalHomeBroker/src/main/java/com/controller/BrokerController.java

```

1  package com.controller;
2
3  import java.util.ArrayList;
4  import javax.validation.Valid;
5  import org.springframework.beans.factory.annotation.Autowired;
6  import org.springframework.stereotype.Controller;
7  import org.springframework.ui.ModelMap;
8  import org.springframework.validation.BindingResult;
9  import org.springframework.web.bind.annotation.ModelAttribute;
10 import org.springframework.web.bind.annotation.RequestMapping;
11 import org.springframework.web.bind.annotation.RequestMethod;
12
13 import com.model.Home;
14 import com.service.BrokerService;
15
16 @Controller
17 public class BrokerController {
18
19     @Autowired
20     private BrokerService service;
21
22     @RequestMapping(value = "/searchPage", method = RequestMethod.GET)
23     public String searchPage(@ModelAttribute("homeBean") Home homeBean) {
24         System.out.println("in controller");
25         return "searchPage";
26     }
27
28     @RequestMapping(value = "/homeList", method = RequestMethod.POST)
29     public String searchHome(@ModelAttribute("homeBean") @Valid Home homeBean, BindingResult
result,
30                             ModelMap model) {
31         String str="";
32         if (result.hasErrors()) {
33             System.out.println("Error : " + result.toString());
34             str="searchPage";
35         }else{
36             ArrayList<Home> list=service.searchHome(homeBean);

```

```

37         model.addAttribute("homeList", list);
38         str= "homeListPage";
39     }
40     return str;
41 }
42 }
43

```

DigitalHomeBroker/src/main/java/com/example/demo/DigitalHomeBrokerApplication.java

```

1  package com.example.demo;
2
3  import org.springframework.boot.SpringApplication;
4  import org.springframework.boot.autoconfigure.SpringBootApplication;
5  import org.springframework.context.annotation.ComponentScan;
6
7  @SpringBootApplication
8  @ComponentScan({"com.*"})
9  public class DigitalHomeBrokerApplication {
10
11     public static void main(String[] args) {
12         SpringApplication.run(DigitalHomeBrokerApplication.class, args);
13     }
14
15 }
16

```

DigitalHomeBroker/src/main/java/com/model/Home.java

```

1  package com.model;
2
3  import javax.validation.constraints.Min;
4
5  import org.springframework.stereotype.Component;
6
7  @Component
8  public class Home {
9
10     private String location;
11     private String address;
12     private String contactNumber;
13     private String bhkType;
14     private String contactPerson;
15     @Min (value=5000, message="Minimum rent should be 5000")
16     private double rent;
17     public double getRent() {
18         return rent;
19     }
20     public void setRent(double rent) {
21         this.rent = rent;
22     }
23     public String getLocation() {
24         return location;
25     }
26     public void setLocation(String location) {
27         this.location = location;
28     }
29     public String getAddress() {
30         return address;
31     }
32     public void setAddress(String address) {
33         this.address = address;
34     }
35     public String getContactNumber() {
36         return contactNumber;
37     }
38

```



```

38     public void setContactNumber(String contactNumber) {
39         this.contactNumber = contactNumber;
40     }
41     public String getBhkType() {
42         return bhkType;
43     }
44     public void setBhkType(String bhkType) {
45         this.bhkType = bhkType;
46     }
47     public String getContactPerson() {
48         return contactPerson;
49     }
50     public void setContactPerson(String contactPerson) {
51         this.contactPerson = contactPerson;
52     }
53
54     public Home(String location, String address, String contactNumber, String bhkType, String
contactPerson, double rent) {
55         super();
56         this.location = location;
57         this.address = address;
58         this.contactNumber = contactNumber;
59         this.bhkType = bhkType;
60         this.contactPerson = contactPerson;
61         this.rent=rent;
62     }
63     public Home()
64     {}
65
66 }
67
68

```

DigitalHomeBroker/src/main/java/com/service/BrokerService.java

```

1  package com.service;
2
3  import java.util.ArrayList;
4  import org.springframework.stereotype.Service;
5  import com.model.Home;
6
7
8  @Service
9  public class BrokerService {
10     ArrayList<Home> list = new ArrayList<Home>();
11
12     public ArrayList<Home> searchHome(Home homeBean) {
13         ArrayList<Home> newList=new ArrayList<Home>();
14         System.out.println(list);
15
16         for(int i=0;i<list.size();i++) {
17             if(list.get(i).getLocation().equalsIgnoreCase(homeBean.getLocation()) &&
list.get(i).getBhkType().equalsIgnoreCase(homeBean.getBhkType())&&
list.get(i).getRent()==homeBean.getRent()) {
18                 newList.add(list.get(i));
19             }
20         }
21         return newList;
22     }
23
24     public BrokerService() {
25         addHomes();
26     }
27
28     public ArrayList<Home> addHomes(){
29

```

```

30         list.add(new Home("Coimbatore", "#412, Shipitha cascade, Ganapathy, PIN-
641001", "9917678456", "1BHK", "Malavika", 8000));
31         list.add(new Home("Coimbatore", "#12, shardhalotus, Ram Nagar, PIN-
641001", "9917100989", "2BHK", "Karthik", 12000));
32         list.add(new Home("Coimbatore", "#12, Anandha nilayam, Kuniamuthur, PIN-
641001", "9917100989", "2BHK", "Karthik", 12000));
33         list.add(new Home("Coimbatore", "#201, Shipitha cascade, Saravanampatti, PIN-
641001", "9917123456", "3BHK", "Sowparnika", 17000));
34
35         list.add(new Home("Chennai", "#34/45, RV Homes, 5th cross, Anssari street, PIN-
600001", "7788900989", "1BHK", "Mukunthan", 6000));
36         list.add(new Home("Chennai", "#212/4, 10th cross, T-nagar, PIN-
60008", "9917100989", "2BHK", "Annesh", 14000));
37         list.add(new Home("Chennai", "#556/4, 5th cross, Tambaram, PIN-
60010", "9917100989", "2BHK", "Jisha", 14000));
38         list.add(new Home("Chennai", "#556/4, Nila paradise, Tambaram, PIN-
60010", "9917100989", "2BHK", "Jisha", 14000));
39         list.add(new Home("Chennai", "#412/4, 10th cross, Velacheri, PIN-
60009", "9917156989", "3BHK", "Rakesh", 18000));
40
41         list.add(new Home("Bangalore", "#45, 12Th main , GM Palya, PIN-
560075", "9917122119", "1BHK", "Senthil", 11000));
42         list.add(new Home("Bangalore", "#67/2, 3rd cross , Indra nagar, PIN-
560077", "9944550989", "2BHK", "RamKumar", 17000));
43         list.add(new Home("Bangalore", "#302, Sakthi enclave , 1st main , Malleshpalya, PIN-
560075", "9917668889", "3BHK", "Saradha", 25000));
44         return list;
45
46     }
47 }
48
49

```

DigitalHomeBroker/src/main/resources/application.properties

```

1 server.port=9088
2 spring.mvc.view.prefix = /WEB-INF/views/
3 spring.mvc.view.suffix = .jsp
4 spring.mvc.static-path-pattern=/resources/**
5

```

DigitalHomeBroker/src/main/webapp/WEB-INF/views/homeListPage.jsp

```

1 <%@page isELIgnored="false" %>
2 <%@ taglib prefix="c" uri="http://java.sun.com/jsp/jstl/core" %>
3 <%@ taglib uri="http://java.sun.com/jsp/jstl/functions" prefix="fn" %>
4 <html>
5 <body bgcolor="lavender">
6 <h1><center> Digital Home Broker </center></h1>
7 <h2><center>Search homes|Apartments !!! </center></h2>
8 <center>
9         <c:if test="${fn:length(homeList) == 0}">
10             <center><h2> No home/Apartment found for this search</h2></center>
11             </c:if>
12             <c:if test="${fn:length(homeList) > 0}">
13                 <table border="1" cellpadding="6">
14                     <tr>
15                         <th>Location</th>
16                         <th>BhkType</th>
17                         <th>Address</th>
18                         <th>ContactPerson</th>
19                         <th>ContactNumber</th>
20                         <th>Rent</th>
21                     </tr>
22                     <c:forEach var="obj" items="${homeList}">
23
24

```

```

25         <td>${obj.location}</td>
26         <td>${obj.bhkType}</td>
27         <td>${obj.address }</td>
28         <td>${obj.contactPerson }</td>
29         <td>${obj.contactNumber }</td>
30         <td>${obj.rent }</td>
31     </tr>
32 </c:forEach>
33 </table></center>
34 </c:if>
35 </body>
36 </html>
37

```

DigitalHomeBroker/src/main/webapp/WEB-INF/views/searchPage.jsp

```

1 <%@ taglib prefix="spring" uri="http://www.springframework.org/tags" %>
2 <%@ taglib prefix="sf" uri="http://www.springframework.org/tags/form" %>
3 <%@ taglib uri="http://www.springframework.org/tags/form" prefix="form"%>
4 <%@ page language="java" contentType="text/html; charset=ISO-8859-1"
5     pageEncoding="ISO-8859-1" isELIgnored="false"%>
6
7
8 <!DOCTYPE html PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN"
"http://www.w3.org/TR/html4/loose.dtd">
9 <html>
10 <body style="background-color:lavender">
11 <h1><center> Digital Home Broker </center></h1>
12 <h2><center>Search homes|Apartments !!! </center></h2>
13
14 <form:form method="post" action="/homeList" modelAttribute="homeBean">
15 <table style="margin: 0px auto; margin-left: auto; margin-right:auto">
16
17         <tr>
18             <td>Search location</td>
19             <td><form:input path="location" id="location" /></td>
20         </tr>
21         <tr>
22             <td> Search BHK Type</td>
23             <td><form:input path="bhkType" id="bhkType"/></td>
24         </tr>
25         <tr>
26             <td>Preferred Rent</td>
27             <td><form:input path="rent" id="rent"/></td>
28         <td><form:errors path="rent" id="errorMsg"/></td>
29         <tr>
30             <td><input type="submit" value="Search" name="submit" /></td>
31
32         </tr>
33
34 </table>
35 </form:form>
36
37
38 </body>
39 </html>
40

```

Grade

Reviewed on Friday, 18 February 2022, 12:20 AM by Automatic grade

Grade 100 / 100

Assessment report

Assessment Completed Successfully

[\[+\]Grading and Feedback](#)