

# Snap Fitness Center - V1

**Grade settings:** Maximum grade: 100

**Based on:** [Snap Fitness Center - V1](#)

**Run:** Yes **Evaluate:** Yes

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

GiB **Maximum number of processes:** 10000

## Snap Fitness center

Snap Fitness center is a famous GYM in the city. It offers various packages to the customers to join in their center. They are in need of a web application for their new customers to show their available packages and the estimated cost for it. Help them by developing an application

Create a Spring MVC Spring Boot Application for developing a Snap fitness center Application. Design a Snap Enquiry page to choose an appropriate package (Aerobics, Zumba, PersonalTraining, Yoga, RegularGYM) and enter the Number of months

On clicking Package cost button, the application should calculate the package cost depending on the package type is chosen and the cost per month which the customers need to pay for the package. The customer has to then be redirected to **snapEstimation.jsp** page that displays the message **Welcome to Snap Fitness Center. Your preferred package cost is RS. <<cost>>Rs.**

### Application Work Flow:

- **FitnessController** is the Controller class.
- **FitnessCenter** is the model class with three attributes **preferredPackage**, **costPerMonth** and **numberOfMonths** along with its getters and setters.
- **FitnessService** is the Service class, has a method called **calculatePackageCost** that takes **FitnessCenter** as its argument and returns double.
- This method needs to set the **costPerMonth** instance variable based on the cost per month (in Rs) value given in the below mentioned table.
- Calculate the package cost depending on the **costPerMonth** for the **preferredPackage** chosen and the **numberOfMonths**.
- Package cost should be returned as double.

Package Type	cost per month(in Rs)
Aerobics	1500.0
Zumba	1750.0
RegularGYM	1250.0
PersonalTraining	2500.0
Yoga	1300.0

Initially, the customer should be routed via the **FitnessController**'s **snapEnquiryPage** method to **snapEnquiryPage.jsp** that allows user to choose the **preferredPackage** and enter the duration in hours.

**[Note: snapEnquiryPage method has to be written inside the FitnessController]**

A method in the **FitnessController** known as **buildState** should be annotated with the **ModelAttribute** `@packageList`. This method should populate the preferred packages (Aerobics, Zumba, RegularGYM, PersonalTraining, Yoga ) in the Map as key-value pair. Both key and value be the same preferredPackage. (Example: Key- Aerobics, Value- Aerobics, Key- Zumba, Value- Zumba, etc) and then return the Map. This should be then used to auto-populate the preferredPackage in the **snapEnquiryPage.jsp**

**[Note: buildState method should be written inside the FitnessController]**

On clicking the Package Cost button, the **FitnessController's calculatePackageCost** method should be called. This method takes three arguments - model attribute named `snap` which holds the form populated **FitnessCenter** Object, **BindingResult** and the **ModelMap**.

- This method should calculate the package cost by invoking the **calculatePackageCost** method of the **FitnessService**.

**Screen designs and Expected Output:**

The screenshot shows a web form titled "Snap Fitness Center". It contains a dropdown menu labeled "Select Package" with "RegularGYM" selected. Below it is a text input field labeled "Package duration (In months)" with the value "0". At the bottom is a button labeled "Package Cost".

If the **numberOfMonths** entered by the customer is less than 1 then an error message "**Number of months should be minimum one**" has to be displayed in the **snapEnquiryPage.jsp**. To do this validation use appropriate annotation above **numberOfMonths** attribute in **FitnessCenter** model class.

**snapEnquiryPage.jsp**

This screenshot shows the same form as the previous one, but with an error message. The "Package duration (In months)" input field now contains "0", and a red error message "Number of Months should be minimum one" is displayed to the right of the input field. The "Package Cost" button remains at the bottom.

- Redirect the user to **snapEstimation.jsp** page with a message `Welcome to Snap Fitness center. Your estimated package cost is Rs. <<cost>>`

**snapEstimation.jsp**

**Welcome to Snap Fitness Center. Your preferred package cost is Rs. 3750.0**

**Design Constraints:**

**UI Design Constraints:**

snapFitnesspage.jsp		
Component	ID	Constraints
Select	preferredPackage	Should be auto-populated using the model attribute written above the buildState method inside the FitnessController. Do not hard code the values
Textbox	numberOfMonths	Should not be less than 1
submit	submit (Name)	-

**Note:** In snapEstimation.jsp, the Result has to be rendered in the **<h2>** tag

## Component Specification

### Controller

FitnessController			
AttributeName	AttributeType	Access Specifier	Constraints
FitnessService	service	private	Use annotation to Autowire

FitnessController			
Method Name	Method Argument name:type	Return type	RequestMapping URL & Request Method
snapEnquiryPage	modelAttribute â€œsnapâ€• :FitnessCenter	String	/snapEnquiryPage & GET
calculatePackageCost	modelAttribute â€œsnapâ€• :FitnessCenter , result:BindingResult, model:ModelMap	String	/snapEstimation & POST
buildState		Map <String,String>	Should be annotated with ModelAttribute with name â€œpackageListâ€•

**Service:**

FitnessService		
Method Name	Method Argument name:type	Return type
calculatePackageCost	fitness :FitnessCenter	double

**Model:**

FitnessCenter	
AttributeName	AttributeType
preferredPackage	String
costPerMonth	double
numberOfMonths	int

**Overall Design Constraints:**

- FitnessController should be inside the package com.controller
- FitnessService should be inside the package com.service
- FitnessCenter should be in the package com.model
- Use appropriate annotation to configure FitnessService as a Service
- Use appropriate annotation to configure FitnessController as a Controller
- SnapFitnessService should be autowired inside the FitnessController.
- Use annotations to implement the business Validation as specified in the screenshot [That is when the duration in hours is less than 1 then error message "**Number of months should be minimum one**" 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[+]

### SnapFitness/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 http://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>BuffetRestaurant</artifactId>
13    <version>0.0.1-SNAPSHOT</version>
14    <name>BuffetRestaurant</name>
15    <description>Demo project for Spring Boot</description>
16
17    <properties>
18        <java.version>1.8</java.version>
19    </properties>
20
21    <dependencies>
22        <dependency>
23            <groupId>org.springframework.boot</groupId>
24            <artifactId>spring-boot-starter-web</artifactId>
25        </dependency>
26
27        <dependency>
28            <groupId>org.apache.tomcat.embed</groupId>
29            <artifactId>tomcat-embed-jasper</artifactId>
30            <scope>provided</scope>
31        </dependency>
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

```

Snapshot/src/main/java/com/controller/FitnessController.java

```

1 package com.controller;
2
3 import java.util.HashMap;

```

```

4 import java.util.Map;
5
6 import javax.validation.Valid;
7
8 import org.springframework.beans.factory.annotation.Autowired;
9 import org.springframework.stereotype.Controller;
10 import org.springframework.ui.ModelMap;
11 import org.springframework.validation.BindingResult;
12 import org.springframework.web.bind.annotation.ModelAttribute;
13 import org.springframework.web.bind.annotation.RequestMapping;
14 import org.springframework.web.bind.annotation.RequestMethod;
15
16 import com.model.FitnessCenter;
17 import com.service.FitnessService;
18
19 @Controller
20 public class FitnessController {
21
22     @Autowired
23     private FitnessService service;
24
25     @RequestMapping(value = "/snapEnquiryPage", method = RequestMethod.GET)
26     public String showPage(@ModelAttribute("snap") FitnessCenter fitness) {
27         System.out.println("in controller");
28         return "snapEnquiryPage";
29     }
30
31
32     @ModelAttribute("packageList")
33     public Map<String, String> buildState(){
34
35         Map<String, String> packMap = new HashMap<String, String>();
36         packMap.put("Aerobics", "Aerobics");
37         packMap.put("Zumba", "Zumba");
38         packMap.put("RegularGYM", "RegularGYM");
39         packMap.put("Yoga", "Yoga");
40         packMap.put("PersonalTraining", "PersonalTraining");
41
42         return packMap;
43     }
44
45     @RequestMapping(value = "/packEstimation", method = RequestMethod.POST)
47     public String calculatePackageCost(@ModelAttribute("snap") @Valid FitnessCenter fitness,
BindingResult result,
48         ModelMap model) {
49
50         if (result.hasErrors()) {
51             System.out.println("Error : " + result.toString());
52             return "snapEnquiryPage";
53         }
54         double cost=service.calculatePackageCost(fitness);
55         model.addAttribute("cost", cost);
56         return "snapEstimation";
57     }
58
59
60 }
61

```

SnapFitness/src/main/java/com/example/demo/SnapFitnessApplication.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 SnapFitnessApplication {
10
11     public static void main(String[] args) {
12         SpringApplication.run(SnapFitnessApplication.class, args);
13     }
14
15 }
16

```

## SnapFitness/src/main/java/com/model/FitnessCenter.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 FitnessCenter {
9
10     private String preferredPackage;
11     private double costPerMonth;
12     @Min(value = 1, message = "Number of Months should be minimum one")
13     private int numberOfMonths;
14
15
16     public String getPreferredPackage() {
17         return preferredPackage;
18     }
19     public void setPreferredPackage(String preferredPackage) {
20         this.preferredPackage = preferredPackage;
21     }
22     public double getCostPerMonth() {
23         return costPerMonth;
24     }
25     public void setCostPerMonth(double costPerMonth) {
26         this.costPerMonth = costPerMonth;
27     }
28     public int getNumberOfMonths() {
29         return numberOfMonths;
30     }
31     public void setNumberOfMonths(int numberOfMonths) {
32         this.numberOfMonths = numberOfMonths;
33     }
34
35
36
37
38 }
39
40

```

## SnapFitness/src/main/java/com/service/FitnessService.java

```

1 package com.service;
2
3 import org.springframework.stereotype.Service;
4
5 import com.model.FitnessCenter;
6
7 @Service
8 public class FitnessService {
9
10     public double calculatePackageCost (FitnessCenter fitness) {

```



```

11
12         double packageCost=0.0;
13
14         if(fitness.getPreferredPackage().equalsIgnoreCase("Aerobics") &&
fitness.getNumberOfMonths()>=1)
15             {
16                 fitness.setCostPerMonth(1500.0);
17                 packageCost=fitness.getCostPerMonth()*fitness.getNumberOfMonths();
18             }
19         else if(fitness.getPreferredPackage().equalsIgnoreCase("Zumba") &&
fitness.getNumberOfMonths()>=1 )
20             {
21                 fitness.setCostPerMonth(1750.0);
22                 packageCost=fitness.getCostPerMonth()*fitness.getNumberOfMonths();
23             }
24         else if(fitness.getPreferredPackage().equalsIgnoreCase("RegularGYM") &&
fitness.getNumberOfMonths()>=1)
25             {
26                 fitness.setCostPerMonth(1250.0);
27                 packageCost=fitness.getCostPerMonth()*fitness.getNumberOfMonths();
28             }
29         if(fitness.getPreferredPackage().equalsIgnoreCase("PersonalTraining") &&
fitness.getNumberOfMonths()>=1)
30             {
31                 fitness.setCostPerMonth(2500.0);
32                 packageCost=fitness.getCostPerMonth()*fitness.getNumberOfMonths();
33             }
34         else if(fitness.getPreferredPackage().equalsIgnoreCase("Yoga") && fitness.getNumberOfMonths()>=1)
35         {
36             fitness.setCostPerMonth(1300.0);
37             packageCost=fitness.getCostPerMonth()*fitness.getNumberOfMonths();
38         }
39         return packageCost;
40
41     }
42
43 }
44

```

## SnapFitness/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

```

## SnapFitness/src/main/webapp/WEB-INF/views/snapEnquiryPage.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 <%@ taglib prefix="form" uri="http://www.springframework.org/tags/form" %>
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> Snap Fitness Center </center></h1>
12 <form:form method="post" action="packEstimation" modelAttribute="snap">
13 <table style="margin: 0px auto; margin-left: auto; margin-right:auto">
14
15         <tr>
16
17                 <td>Select Package</td>
18
19         </tr>
20
21     </table>
22
23 </form>
24

```

```

18             <form:select path="preferredPackage" id="preferredPackage">
19                 <form:options items="${packageList}"/>
20             </form:select>
21
22             </td>
23         </tr>
24
25         <tr>
26             <td>Package duration (In months)</td>
27             <td><form:input path="numberOfMonths" id="numberOfMonths"/></td>
28             <td><form:errors path="numberOfMonths" id="errorMsg"/></td>
29         </tr>
30
31         <tr>
32             <td><input type="submit" value="Package Cost" name="submit"
33         </td>
34     </tr>
35 </table>
36 </form:form>
37
38
39 </body>
40 </html>
41

```

SnapFitness/src/main/webapp/WEB-INF/views/snapEstimation.jsp

```

1 <%@page isELIgnored="false" %>
2 <html>
3 <body bgcolor="lavender">
4 <h2>Welcome to Snap Fitness Center. Your preferred package cost is Rs. ${cost} </h2>
5 </body>
6 </html>

```

## Grade

Reviewed on Friday, 18 February 2022, 1:09 PM by Automatic grade

**Grade 100 / 100**

**Assessment report**

**Assessment Completed Successfully**

**[\[-\]Grading and Feedback](#)**

*Feature Test - 7.50 / 7.50(Success)*

*\* check whether the Controller / Service class with the required annotation is implemented - 2.50 / 2.50*

*\* check whether the Controller class with the required annotation is implemented - 1.50 / 1.50*

*\* check whether the Service class with the required annotation is implemented - 1.50 / 1.50*

*\* check whether the model class with the required annotation is implemented - 2.00 / 2.00*

*Functional testing - 30.00 / 30.00(Success)*

*\* check whether the package cost for the Aerobics type is calculated properly - 6.00 / 6.00*

*\* check whether the package cost for the Zumba type is calculated properly - 6.00 / 6.00*

*\* check whether the package cost for the Yoga type is calculated properly - 6.00 / 6.00*

*\* check whether the package cost for the PersonalTraining type is calculated properly - 6.00 / 6.00*

*\* check whether the package cost for the RegularGYM type is calculated properly - 6.00 / 6.00*

*Spring Testing in the Controller - 45.00 / 45.00(Success)*

- \* check whether the request is mapped for snapEnquiryPage and redirected to snapEnquiryPage.jsp page - 6.00 / 6.00
- \* check whether the request is mapped for snapEstimation and redirected to snapEstimation.jsp page - 6.00 / 6.00
- \* check whether the model attribute is specified with the name above the buildState method that holds the Map containing the Preferred package - 6.00 / 6.00
- \* check whether the buildstate method is configured properly - 7.00 / 7.00
- \* check whether the request is mapped and redirected to the snapEstimation page for valid details - 10.00 / 10.00
- \* check whether the request is mapped and redirected to the same page for Invalid details - 10.00 / 10.00

UI Testing - 17.50 / 17.50(Success)

- \* Check whether preferredPackage Aerobics is autopopulated as per the requirement - 3.00 / 3.00
- \* Check whether preferredPackage Zumba is autopopulated as per the requirement - 3.00 / 3.00
- \* Check whether preferredPackage PersonalTraining is autopopulated as per the requirement - 3.00 / 3.00
- \* Check whether preferredPackage Yoga is autopopulated as per the requirement - 5.00 / 5.00
- \* Check whether preferredPackage RegularGYM is autopopulated as per the requirement - 0.50 / 0.50
- \* check whether the snapEnquiryPage.jsp is rendered with proper constraints for numberOfMonths - 0.50 / 0.50
- \* check whether UI from snapEnquiryPage to snapEstimator page is redirected after calculating the package cost for Zumba - 0.50 / 0.50
- \* check whether UI from snapEnquiryPage to snapEstimator page is redirected after calculating the package cost for PersonalTraning - 0.50 / 0.50
- \* check whether UI from snapEnquiryPage to snapEstimator page is redirected after calculating the package cost for Aerobics - 1.00 / 1.00
- \* check whether the numberOfMonths field validated properly - 0.50 / 0.50

Test Case Passed