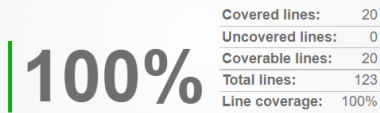


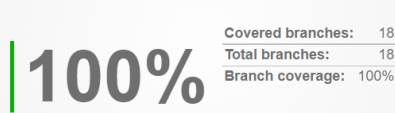
Information

Class: FuelQuoteApp_p1.Controllers.ClientController
Assembly: FuelQuoteApp_p1
File(s): C:\Users\Dell\OneDrive\Desktop\FuelQuoteApp_p1\FuelQuoteApp_p1\Controllers\ClientController.cs

Line coverage



Branch coverage



Method coverage

Method coverage is only available for sponsors.

[Upgrade to PRO version](#)

Metrics

| Method | Branch coverage | Cyclomatic complexity | Line coverage |
|----------------------------------|-----------------|-----------------------|---------------|
| ClientProfileDataValidation(...) | 100% | 18 | 100% |

File(s)

C:\Users\Dell\OneDrive\Desktop\FuelQuoteApp_p1\FuelQuoteApp_p1\Controllers\ClientController.cs

```
# Line Line coverage
1 using FuelQuoteApp_p1.EntModels.Models;
2 using FuelQuoteApp_p1.Models.Client_Profile;
3 using Microsoft.AspNetCore.Http;
4 using Microsoft.AspNetCore.Mvc;
5 using Newtonsoft.Json;
6 using System;
7 using System.Diagnostics.CodeAnalysis;
8
9 namespace FuelQuoteApp_p1.Controllers
10 {
11     public class ClientController : Controller
12     {
13         [HttpGet]
14         [ExcludeFromCodeCoverage]
15         public IActionResult ClientDashBoard()
16         {
17             int usrID = 1; //HardCoded
18             Client client = new Client
19             {
20                 User_ID = new User()
21                 {
22                     Id = usrID,
23                     Email="sainarne15@gmail.com",
24                     UserName="sainarne15@gmail.com"
25                 },
26                 FullName = "Narne Lakshmi Narasimha Sai",
27                 Address1 = "Kirby Dr",
28                 Address2 = "",
29                 City = "Houston",
30                 State = States.TX.ToString(),
31                 ZipCode = 77054.ToString()
32             };
33             HttpContext.Session.SetString("ClientDetails", JsonConvert.SerializeObject(client));
34
35             return View();
36         }
37
38         [HttpGet]
39         [ExcludeFromCodeCoverage]
40         public IActionResult ClientProfile()
41         {
42             return View();
43         }
44
45         [HttpPost]
46         [ExcludeFromCodeCoverage]
47         public ActionResult ClientProfile(Profile clientProfile)
48         {
49             if (ModelState.IsValid)
50             {
51                 Client client = new Client
52                 {
53                     FullName = clientProfile.FullName,
54                     Address1 = clientProfile.Address1,
55                     Address2 = clientProfile.Address2,
56                     City = clientProfile.City,
57                     State = clientProfile.State.ToString(),
58                     ZipCode = clientProfile.ZipCode.ToString()
59                 };
60                 HttpContext.Session.SetString("ClientDetails", JsonConvert.SerializeObject(client));
61                 return View();
62             }
63             return View();
64         }
65     }
66 }
```

```

59         ZipCode = clientProfile.ZipCode,
60         Email = User.Identity.Name
61     };
62
63     User usrDetails = new User();
64     client.User_ID = usrDetails;
65
66
67     return RedirectToAction("GetQuote", "Quote");
68 }
69 else
70 {
71     return View();
72 }
73
74 }
75
76
77 [HttpGet]
78 [ExcludeFromCodeCoverage]
79 public IActionResult DisplayProfile()
80 {
81     Client client = new Client();
82     client = JsonConvert.DeserializeObject<Client>(HttpContext.Session.GetString("ClientDetails"));
83
84
85     Profile cl = new Profile()
86     {
87         FullName = client.FullName,
88         Address1 = client.Address1,
89         Address2 = client.Address2,
90         City = client.City,
91         State = States.TX,
92         ZipCode = client.ZipCode
93     };
94
95     return View(cl);
96 }
97
98
99 public bool ClientProfileDataValidation(Profile data)
8 100 {
8 101     bool flag = false;
8 102     if ((data.FullName.Length <= 50) && (data.FullName != String.Empty))
6 103     {
6 104     P     if (((data.Address1.Length <= 100) && (data.Address1 != String.Empty)) && (data.Address2.Length <= 100))
5 105     {
5 106     P     if ((data.City.Length <= 100) && (data.City != String.Empty))
3 107     {
3 108     P     if (data.ZipCode.Length <= 9 && data.ZipCode.Length >= 5)
2 109     {
2 110         flag = true;
2 111     }
3 112     }
5 113     }
6 114     }
115     else
2 116     {
2 117         flag = false;
2 118     }
119
8 120     return flag;
8 121     }
122 }
123 }

```