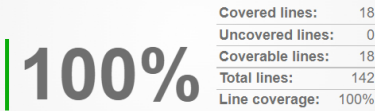


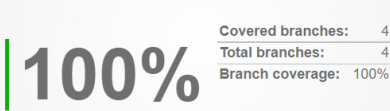
Information

Class: FuelQuoteApp_p1.Controllers.QuoteController
Assembly: FuelQuoteApp_p1
File(s): E:\UHSummer\SD\FuelQuoteApp_p1\FuelQuoteApp_p1\Controllers\QuoteController.cs

Line coverage



Branch coverage



Method coverage

Method coverage is only available for sponsors.

Upgrade to PRO version

- .ctor(FuelQuoteApp_p1.Provider.IFuelQu...
- QuoteDataValidation(FuelQuoteApp_p1....

Metrics

Method	Branch coverage ⓘ	Cyclomatic complexity ⓘ	Line coverage ⓘ
.ctor(...)	100%	1	100%
QuoteDataValidation(...)	100%	4	100%

File(s)

E:\UHSummer\SD\FuelQuoteApp_p1\FuelQuoteApp_p1\Controllers\QuoteController.cs

```
# Line Line coverage
1 using FuelQuoteApp_p1.BusinessLayer.BL;
2 using FuelQuoteApp_p1.EntModels.Models;
3 using FuelQuoteApp_p1.Models.Quote;
4 using FuelQuoteApp_p1.Provider;
5 using FuelQuoteApp_p1.Repository;
6 using Microsoft.AspNetCore.Http;
7 using Microsoft.AspNetCore.Mvc;
8 using Newtonsoft.Json;
9 using System;
10 using System.Collections.Generic;
11 using System.Diagnostics.CodeAnalysis;
12 using System.Globalization;
13
14 namespace FuelQuoteApp_p1.Controllers
15 {
16     public class QuoteController : Controller
17     {
18         private readonly IFuelQuoteProvider _FuelQuotePro;
19         public QuoteController(IFuelQuoteProvider FuelQuotePro)
20         {
21             _FuelQuotePro = FuelQuotePro;
22         }
23
24         [HttpGet]
25         [ExcludeFromCodeCoverage]
26         public IActionResult GetQuote()
27         {
28             FQuote quote = new FQuote();
29             Client client = new Client();
30             client = JsonConvert.DeserializeObject<Client>(HttpContext.Session.GetString("ClientDetails"));
31             quote.DeliveryAddress = client.Address1 + " " + client.Address2;
32             return View(quote);
33         }
34
35         [HttpPost]
36         [ExcludeFromCodeCoverage]
37         public IActionResult GetQuote(FQuote quote)
38         {
39
40             Price pricetdetails = new Price();
41             pricetdetails = GetPrice(quote);
42             quote.PricePerGallon = pricetdetails.PricePerGallon;
43             quote.TotalAmount = pricetdetails.TotalAmount;
44             HttpContext.Session.SetString("QuoteDetails", JsonConvert.SerializeObject(quote));
45
46             return RedirectToAction("GetFinalQuote");
47         }
48     }
49
50     [HttpGet]
51     [ExcludeFromCodeCoverage]
52     public IActionResult QuoteHistory()
53     {
54
55         int usrID = _FuelQuotePro.GetUserID(User.Identity.Name);
56         IEnumerable<Quote> quotess = _FuelQuotePro.GetQuoteHistory(usrID);
57
58     }
```

```

58         return View(quotess);
59     }
60
61     [ExcludeFromCodeCoverage]
62     public Price GetPrice(FQuote quote)
63     {
64         int usrID = _FuelQuotePro.GetUserID(User.Identity.Name);
65
66         Client client = new Client();
67         client = JsonConvert.DeserializeObject<Client>(HttpContext.Session.GetString("ClientDetails"));
68         QuoteDetails quoteInfo = new QuoteDetails
69         {
70             DateRequested = quote.DateRequested,
71             GallonsRequested = quote.GallonsRequested,
72             State = client.State,
73             quoteHistory = _FuelQuotePro.GetQuoteHistoryCount(usrID)
74         };
75
76
77         Price price = new Price { PricePerGallon = 5, TotalAmount = 1500 };    //Passing Hard Coded values, since pr
78
79         return price;
80     }
81
82     [HttpGet]
83     [ExcludeFromCodeCoverage]
84     public IActionResult GetFinalQuote()
85     {
86         FQuote quote = new FQuote();
87         Client client = new Client();
88         quote = JsonConvert.DeserializeObject<FQuote>(HttpContext.Session.GetString("QuoteDetails"));
89         client = JsonConvert.DeserializeObject<Client>(HttpContext.Session.GetString("ClientDetails"));
90         quote.DeliveryAddress = client.Address1 + " " + client.Address2;
91         return View(quote);
92     }
93
94
95     [HttpPost]
96     [ExcludeFromCodeCoverage]
97     public IActionResult GetFinalQuote(FQuote quote)
98     {
99         quote = JsonConvert.DeserializeObject<FQuote>(HttpContext.Session.GetString("QuoteDetails"));
100         Client client = new Client();
101         client = JsonConvert.DeserializeObject<Client>(HttpContext.Session.GetString("ClientDetails"));
102         quote.DeliveryAddress = client.Address1 + " " + client.Address2;
103
104         Quote dbquote = new Quote
105         {
106             DateRequested = quote.DateRequested,
107             DeliveryAddress = quote.DeliveryAddress,
108             GallonsRequested = quote.GallonsRequested,
109             PricePerGallon = quote.PricePerGallon,
110             TotalAmount = quote.TotalAmount,
111
112         };
113         User userinfo = new User();
114         dbquote.User_Id = userinfo.Id;
115         dbquote.User_Id = _FuelQuotePro.GetUserID(User.Identity.Name);
116         dbquote = _FuelQuotePro.AddQuote(dbquote);
117         return View("SavedQuote", dbquote);
118     }
119
120
121
122
123     public bool QuoteDataValidation(Quote data)
3 124     {
3 125         bool flag = false;
3 126         if (data.GallonsRequested != 0) /*Default value is 0*/
2 127         {
2 128             if (DateTime.Compare(data.DateRequested, DateTime.Now) > 0)
1 129             {
1 130                 flag = true;
1 131             }
132
2 133         }
134         else
1 135         {
1 136             flag = false;
1 137         }
138
3 139         return flag;
3 140     }
141 }
142

```