< Summary

Methods/Properties

- .ctor(FuelQuoteApp_p1.Provider.IFuelQu…
- ClientProfileDataValidation(FuelQuoteAp…

Information

Class:	FuelQuoteApp_p1.Controllers.ClientController
Assembly:	FuelQuoteApp_p1
File(s):	E:\UH\Summer\SD\FuelQuoteApp_p1\FuelQuoteApp_p1\Controllers\ClientController.cs

Line coverage

100%

Covered lines:	24
Uncovered lines:	0
Coverable lines:	24
Total lines:	119
Line coverage:	100%

Branch coverage

100%

Covered branches:	18
Total branches:	18
Branch coverage:	100%

Method coverage

Method coverage is only available for sponsors.

Upgrade to PRO version

Metrics

Method	Branch coverage	Cyclomatic complexity	Line coverage
.ctor(...)	100%	1	100%
ClientProfileDataValidation(...)	100%	18	100%

File(s)

E:\UH\Summer\SD\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 FuelQuoteApp_p1.Provider;
      4  using FuelQuoteApp_p1.Repository;
      5  using Microsoft.AspNetCore.Http;
      6  using Microsoft.AspNetCore.Mvc;
      7  using Newtonsoft.Json;
      8  using System;
      9  using System.Diagnostics.CodeAnalysis;
     10
     11  namespace FuelQuoteApp_p1.Controllers
     12  {
     13      public class ClientController : Controller
     14      {
     15          private readonly IFuelQuoteProvider _FuelQuotePro;
     16
1    17          public ClientController(IFuelQuoteProvider FuelQuotePro)
1    18          {
1    19              _FuelQuotePro = FuelQuotePro;
     20
1    21          }
     22          [HttpGet]
     23          [ExcludeFromCodeCoverage]
     24          public IActionResult ClientDashBoard()
     25          {
     26              int usrID = _FuelQuotePro.GetUserID(User.Identity.Name);
     27              Client client = _FuelQuotePro.GetClient(usrID);
     28              HttpContext.Session.SetString("ClientDetails", JsonConvert.SerializeObject(client));
     29
     30              return View();
     31          }
     32          [HttpGet]
     33          [ExcludeFromCodeCoverage]
     34          public IActionResult ClientProfile()
     35          {
     36              return View();
     37          }
     38
     39
     40          [HttpPost]
     41          [ExcludeFromCodeCoverage]
     42          public ActionResult ClientProfile(Profile clientProfile)
     43          {
     44              if (ModelState.IsValid)
     45              {
     46                  Client client = new Client
     47                  {
     48                      FullName = clientProfile.FullName,
     49                      Address1 = clientProfile.Address1,
     50                      Address2 = clientProfile.Address2,
     51                      City = clientProfile.City,
     52                      State = clientProfile.State.ToString(),
     53                      ZipCode = clientProfile.ZipCode,
     54                      Email = User.Identity.Name
     55
     56                  };
     57
```

```
                    //User usrDetails = new User();
                    // client.User_ID = usrDetails;
                    client.User_Id = _FuelQuotePro.GetUserID(User.Identity.Name);
                    Client AddedClient = _FuelQuotePro.AddClient(client);

                    return RedirectToAction("ClientDashBoard", "Client");
                }
                else
                {
                    return View();
                }

            }

            [HttpGet]
            [ExcludeFromCodeCoverage]
            public IActionResult DisplayProfile()
            {
                Client client = new Client();
                client = JsonConvert.DeserializeObject<Client>(HttpContext.Session.GetString("ClientDetails"));


                Profile cl = new Profile()
                {
                    FullName = client.FullName,
                    Address1 = client.Address1,
                    Address2 = client.Address2,
                    City = client.City,
                    State = States.AK,
                    ZipCode = client.ZipCode
                };

                return View(cl);
            }

            public bool ClientProfileDataValidation(Profile data)
8           {
8               bool flag = false;
8               if ((data.FullName.Length <= 50) && (data.FullName != String.Empty))
6               {
6                   if (((data.Address1.Length <= 100) && (data.Address1 != String.Empty)) && (data.Address2.Length <= 100))
5                   {
5                       if ((data.City.Length <= 100) && (data.City != String.Empty))
3                       {
3                           if (data.ZipCode.Length <= 9 && data.ZipCode.Length >= 5)
2                           {
2                               flag = true;
2                           }
3                       }
5                   }
6               }
                else
2               {
2                   flag = false;
2               }

8               return flag;
8           }


        }
    }
```

Generated by: ReportGenerator 5.1.9.0
17-07-2022 - 23:06:29
GitHub | www.palmmedia.de