

A screenshot of the Microsoft Visual Studio IDE interface. The title bar shows "Debug Any CPU IIS Express (Google Chrome)". The main editor window displays the code for "ProvidersController.cs" under the namespace "UQP.Controllers". The code includes using statements for various System namespaces and Entity-related classes, along with annotations like [EnableCors], [Authorize], and [ProducesResponseType]. The code defines a class "ProvidersController" that inherits from "ApiController". It contains a method that returns a "TextInfo" object with culture settings and XML documentation comments. The status bar at the bottom indicates "100 %".

```
1 using System;
2 using System.Collections.Generic;
3 using System.Data;
4 using System.Data.Entity;
5 using System.Data.Entity.Core.Objects;
6 using System.Data.Entity.Infrastructure;
7 using System.Data.Entity.SqlServer;
8 using System.Globalization;
9 using System.IO;
10 using System.Linq;
11 using System.Net;
12 using System.Net.Http;
13 using System.Net.Http.Headers;
14 using System.Threading.Tasks;
15 using System.Web;
16 using System.Web.Http;
17 using System.Web.Http.Cors;
18 using System.Web.Http.Description;
19 using UQP.Models;
20 using UQP.Utility;
21
22 namespace UQP.Controllers
23 {
24     [EnableCors(origins: "*", headers: "*", methods: "*")]
25     [Authorize]
26     [ProducesResponseType]
27     public class ProvidersController : ApiController
28     {
29
30         TextInfo textInfo = new CultureInfo("en-US", false).TextInfo;
31
32         /// <summary>
33         /// 
34         /// </summary>
35         /// <param name="id"></param>
36         /// 
```

A screenshot of the Microsoft Visual Studio IDE interface. The title bar shows "Microsoft Visual Studio" and the menu bar includes File, Edit, View, Project, Build, Debug, Team, Tools, Test, Analyze, Window, Help. The toolbar has icons for file operations like Open, Save, and Print. The status bar at the bottom right says "Scanned with CamScanner". The code editor window displays a C# file named "UQP.Controllers.ProvidersController.cs". The code is as follows:

```
33     /// 
34     /// </summary>
35     /// <param name="id"></param>
36     /// <returns></returns>
37     [HttpGet]
38     [Route("api/insurancebyproviders/{id}")]
39     public HttpResponseMessage ProviderInsurances(int id)
40     {
41         try
42         {
43             using (UQPEntities db = new UQPEntities(HttpContext.Current.Request.Headers.GetValues("IPA").FirstOrDefault()))
44             {
45                 List<CustomHealthPlan> listhealthPlan = new List<CustomHealthPlan>();
46                 var result = (from i in db.insuranceIpabyprovider_membercount.AsNoTracking()
47                               where i.provider_Id == id && i.healthplan_id != null
48                               select new { i.HealthPlan_Name, i.healthplan_id }).Distinct();
49                 //var lstDist = result.Distinct();
50                 foreach (var ls in result)
51                 {
52                     CustomHealthPlan objhealthPlan = new CustomHealthPlan();
53                     objhealthPlan.health_Name = ls.HealthPlan_Name;
54                     objhealthPlan.healthplan_id = Convert.ToInt32(ls.healthplan_id);
55                     listhealthPlan.Add(objhealthPlan);
56                 }
57                 return Request.CreateResponse(HttpStatusCode.OK, listhealthPlan);
58             }
59         }
60     }
61     catch (Exception ex)
62     {
63         return Request.CreateResponse(HttpStatusCode.InternalServerError, ex.Message);
64     }
65 }
66
67 }
```

```
MembersController.cs ProvidersController.cs X Source Control Explorer ACO_MembersMart.cs Notification_UserSpecific.cs
UQP
68
69
70
71
72 [HttpPost]
73 [Route("api/postprovider")]
74 public HttpResponseMessage PostProvider(Provider_Information prov)
75 {
76     if (!ModelState.IsValid)
77     {
78         return Request.CreateResponse(HttpStatusCode.BadRequest, (ModelState));
79     }
80     try
81     {
82         using (UQPEntities db = new UQPEntities(HttpContext.Current.Request.Headers.GetValues("IPA").FirstOrDefault()))
83         {
84             ProviderDetailsTesting oldprovobj = new ProviderDetailsTesting
85             {
86                 Physician_first_name = prov.Physician_first_name,
87                 Physician_last_name = prov.Physician_last_name,
88                 .NPI = prov.NPI,
89                 emailid = prov.emailid,
90                 Title = prov.Title,
91                 Specialty = prov.Specialty,
92                 status = true,
93                 msgstatus = true,
94                 notifstatus = true,
95                 healthplanid = prov.healthplanid,
96                 visible = true
97
98
99
100
101
102
103
    };
    db.ProviderDetailsTestings.Add(oldprovobj);
    db.SaveChanges();
}
100 %
```

Debug Any CPU IIS Express (Google Chrome)

MembersController.cs ProvidersController.cs Source Control Explorer ACO_MembersMart.cs Notification_UserSpecific.cs

UQP

```
100
101     db.ProviderDetailsTestings.Add(oldprovobj);
102     db.SaveChanges();
103
104     List<providerTesting_address> List_address = new List<providerTesting_address>();
105     foreach(var value in prov.addresslist)
106     {
107         providerTesting_address address = new providerTesting_address()
108         {
109             address = value.address,
110             backline_phone_number = value.backline_phone_number,
111             fax = value.fax,
112             providerid = oldprovobj.providerid,
113             phone = value.phone,
114             npi = value.npi
115         };
116         List_address.Add(address);
117     }
118     db.providerTesting_address.AddRange(List_address);
119     db.SaveChanges();
120     //string[] hpids = prov.HealthplanID.Split(',');
121     //int[] hpintids = hpids.Select(int.Parse).ToArray();
122     //var hpplans = " ";
123     //int i = 0;
124     //foreach (var per in hpintids)
125     //{
126     //    string hp = (from hps in db.HealthPlans
127     //                 where hps.HealthPlan_ID == per
128     //                 select hps.HealthPlan_Name).FirstOrDefault();
129     //    //var hp = db.HealthPlans.Where(x => x.HealthPlan_ID == per).Select(x => x.HealthPlan_Name);
130     //    hpplans = hpplans + "," + hp;
131     //}
132
133
134
135
136
```

```
MembersController.cs ProvidersController.cs > X Source Control Explorer ACO_MembersMart.cs Notification_UserSpecific.cs
UOP UQP Controllers.ProvidersController textinfo
130
131     //    hpplans = hpplans + "," + hp;
132
133     //}
134
135
136
137     //string[] adres = prov.Address_1.Split(',');
138
139
140
141     //foreach (var per in adres)
142     //{
143         // ProviderAddress addobj = new ProviderAddress();
144
145         // addobj.Address = per;
146         // addobj.ProviderID = Convert.ToInt32(oldprovobj.Provider_ID);
147         // addobj.UniqueProviderId = oldprovobj.UniqueProviderID;
148         // db.ProviderAddresses.Add(addobj);
149
150
151     //}
152
153
154
155     //NewProviderTemp provobj = new NewProviderTemp();
156
157     //{
158         // provobj.ProviderID = Convert.ToInt32(oldprovobj.Provider_ID);
159         // provobj.Group_Name = prov.Group_Name == null ? "" : prov.Group_Name;
160         // provobj.Physician_first_name = prov.Physician.first_name;
161         // provobj.Physician_last_name = prov.Physician.last_name;
162         // provobj.Title = prov.Title == null ? "" : prov.Title;
163         // provobj.Address_1 = prov.Address_1 == null ? "" : prov.Address_1;
164         // provobj.Address_2 = "";
165         // provobj.City = "";
166         // provobj.State = "";
167         // provobj.Zip = "";
```

File Edit Team Tools Test Analyze Window Help

Debug Any CPU IIS Express (Google Chrome)

MembersController.cs ProvidersController.cs ACO_MembersMart.cs Notification_UserSpecific.cs

UQP

```
157 // provobj.ProvObjID = Convert.ToInt32(providerProvObj.ProvObjID);
158 // provobj.Group_Name = prov.Group_Name == null ? "" : prov.Group_Name;
159 // provobj.Physician_first_name = prov.Physician_first_name;
160 // provobj.Physician_last_name = prov.Physician_last_name;
161 // provobj.Title = prov.Title == null ? "" : prov.Title;
162 // provobj.Address_1 = prov.Address_1 == null ? "" : prov.Address_1;
163 // provobj.Address_2 = "";
164 // provobj.City = "";
165 // provobj.State = "";
166 // provobj.Zip = "";
167 // provobj.Fax = prov.Fax == null ? "" : prov.Fax;
168 // provobj.Phone = prov.Phone == null ? "" : prov.Phone;
169 // provobj.County = "";
170 // provobj.Backline_Phone_Number = prov.Backline_Phone_Number == null ? "" : prov.Backline_Phone_Number;
171 // provobj.Office_Manager = prov.Office_Manager == null ? "" : prov.Office_Manager;
172 // provobj.Physician_E_mail_or_alt_email_for_office_mgr_ = prov.Physician_E_mail_or_alt_email_for_office_mgr_ == null ? "" : prov.Physician_E_mail_or_alt_email_for_office_mgr_;
173 // provobj.Physician_Cell_ = "";
174 // provobj.CoventryNo = prov.CoventryNo == null ? "" : prov.CoventryNo;
175 // provobj.FreedomNo = prov.FreedomNo == null ? "" : prov.FreedomNo;
176 // provobj.HumanaNo = prov.HumanaNo == null ? "" : prov.HumanaNo;
177 // provobj.OptimumNo = prov.OptimumNo == null ? "" : prov.OptimumNo;
178 // provobj.UltimateNo = prov.UltimateNo == null ? "" : prov.UltimateNo;
179 // provobj.WellcareNo = prov.WellcareNo == null ? "" : prov.WellcareNo;
180 // provobj.SimplyNo = "";
181 // provobj.NPI = prov.NPI;
182 // provobj.Specialty = prov.Specialty;
183 // provobj.UniqueProviderID = prov.UniqueProviderID == null ? "" : prov.UniqueProviderID;
184 // provobj.NotifStatus = true;
185 // provobj.MsgStatus = true;
186 // provobj.Status = true;
187 // provobj.HealthplanID = prov.HealthplanID;
188 // provobj.Medicare_HMO_Plans = hoplans;
189 // provobj.UniqueProviderID = "";
190
191 //}
192 //db.NewProviderItems.Add(provobj);
193
```

```
185 //      provobj.MsgStatus = true;
186 //      provobj.Status = true;
187 //      provobj.HealthplanID = prov.HealthplanID;
188 //      provobj.Medicare_HMO_Plans = hpplans;
189 //      provobj.UniqueProviderID = "";
190
191 //}
192 //db.NewProviderTemps.Add(provobj);
193
194 return Request.CreateResponse(HttpStatusCode.OK, prov);
195 }
196 }
197 catch (Exception ex)
198 {
199     return Request.CreateResponse(HttpStatusCode.InternalServerError, ex.Message);
200 }
201 }
202 }
203
204
205 [HttpGet]
206 [Route("api/ipa")]
207 [ProducesResponseType(typeof(IPADetails), StatusCodes.Status200OK)]
208 public HttpResponseMessage IPADetails([FromUri]InputDate dateobj)
209 {
210     try
211     {
212         using (UQPEntities db = new UQPEntities(HttpContext.Current.Request.Headers.GetValues("IPA").FirstOrDefault()))
213         {
214             List<CustomIPA> lstipa = new List<CustomIPA>();
215             dynamic result = null;
216
217             if (dateobj == null)
218             {
219                 result = db.IPAs;
```

100 %

Output

Show output from: Source Control - Team Foundation

Project Build Debug Team Tools Test Analyze Window Help

Debug Any CPU IIS Express (Google Chrome)

MembersController.cs ProvidersController.cs Source Control Explorer ACO_MembersMait.cs Notification_UserSpecific.cs PostProviderPr

UQP

```
241
242
243     CustomIPA objipa = null;
244     foreach (var lst in result)
245     {
246         objipa = new CustomIPA();
247         objipa.IPA_ID = lst.IPA_ID;
248         objipa.ipacode = lst.IPA_Code;
249         objipa.IPA_Name = lst.IPA_Name;
250
251         lstipa.Add(objipa);
252     }
253
254     return Request.CreateResponse(HttpStatusCode.OK, lstipa);
255
256 }
257
258 catch (Exception ex)
259 {
260     return Request.CreateResponse(HttpStatusCode.InternalServerError, ex.Message);
261 }
262
263
264
265 [HttpGet]
```

```
262
263
264
265 [HttpGet]
266 [Route("api/parentipa")]
267 public HttpResponseMessage ParentIpa()
268 {
269     try
270     {
271         using (UQPEntities db = new UQPEntities())
272         {
273             List<CustomIPA> lstipa = new List<CustomIPA>();
274             int[] accessipa = { 2, 5, 6, 10 };
275             var query = (from i in db.IPAs
276                         where accessipa.Contains(i.IPA_ID)
277                         select new
278                         {
279                             i.IPA_Name,
280                             i.IPA_ID
281                         }).ToList();
282             foreach (var value in query)
283             {
284                 CustomIPA customIPA = new CustomIPA
285                 {
286                     IPA_Name = value.IPA_Name,
287                     IPA_ID = value.IPA_ID
288                 };
289                 lstipa.Add(customIPA);
290             }
291             return Request.CreateResponse(HttpStatusCode.OK, lstipa);
292         }
293     }
294     catch (Exception ex)
295     {
296         return Request.CreateResponse(HttpStatusCode.OK, ex.InnerException);
297     }
298 }
```

Source Control Explorer ACO_MembersMart.cs Notification_UserSpecific.cs

UQP.Controllers.ProvidersController

PostProvider(ProviderInformation prov)

```
288
289
290
291
292
293
294
295
296
297
298
299
300
301
302
303
304
305
306
307
308
309
310
311
312
313
314
315
316
317
318
319
320
321
```

```
1     return Request.CreateResponse(HttpStatusCode.OK, lsthp);
2 }
3 }
4 catch (Exception ex)
{
    return Request.CreateResponse(HttpStatusCode.InternalServerError, ex.Message);
}
}

//[[HttpGet]
//[[Route("api/providerlist/")]
// public List<CustomProviderDetails> ProviderDetails([FromUri]PagingParameterModel pagingparametermodel, [FromUri] Searchprovider obj)
//
//{
//     List<CustomProviderDetails> lstprovider = new List<CustomProviderDetails>();
//     var totalcount = 0;
//     if (obj.startdate.Year.ToString() != "1" && obj.enddate.Year.ToString() != "1")
//     {
//
//         if (obj.name == null)
//         {
//             if (obj.status == null)
//             {
//                 var result = (from mml in db.Members_Mart
//                             //join pr in db.Providers on mml.provider_id equals pr.Provider_ID
//                             where mml.TimeStamp >= obj.startdate && mml.TimeStamp <= obj.enddate && mml.provider_id != null
//                             select new { mml.provider_id }).Distinct().OrderBy(x => x.provider_id).Skip(pagingparametermodel.pageSize *
//((pagingparametermodel.pageNumber - 1)).Take(pagingparametermodel.pageSize);
//                // select new { mml.provider_id, pr.Provider_FirstName, pr.Provider_Email, pr.Provider_Description, pr.Provider_ID, [
//                //pr.Provider_LastName, pr.Provider_MidName, pr.Provider_Level, pr.UniqueProviderID, pr.NPI, pr.Provider_MobileNumber, pr.status }).Distinct();
//                // ;
//                totalcount = (from mml in db.Members_Mart
//                             //join pr in db.Providers on mml.provider_id equals pr.Provider_ID
//                             where mml.TimeStamp >= obj.startdate && mml.TimeStamp <= obj.enddate && mml.provider_id != null
//                             select new { mml.provider_id }).Distinct().Count();
//             }
//         }
//     }
// }
```

A screenshot of the Microsoft Visual Studio IDE. The main window shows a code editor with C# code for a controller named UQP.Controllers.ProvidersController. The code includes logic to calculate a total count of providers based on specific criteria and to map provider data into a CustomProviderDetails object. The Solution Explorer on the right side shows a solution named 'UQP' containing various projects and files.

```
// select new { mml.provider_id, pr.Provider_FirstName, pr.Provider_Email, pr.Provider_Description, pr.Provider_ID, pr.Provider_LastName }
// ;
totalcount = (from mml in db.Members_Mart
               .join pr in db.Providers on mml.provider_id equals pr.Provider_ID
               where mml.TimeStamp >= obj.startdate && mml.TimeStamp <= obj.enddate && mml.provider_id != null
               select new { mml.provider_id }).Distinct().Count();
foreach (var lstdt in result)
{
    var data = (from dt in db.Providers
                where dt.Provider_ID == lstdt.provider_id
                select new
                {
                    dt.Provider_ID,
                    dt.Provider_Email,
                    dt.Provider_Description,
                    dt.Provider_FirstName,
                    dt.Provider_MidName,
                    dt.Provider_LastName,
                    dt.Provider_Level,
                    dt.UniqueProviderID,
                    dt.NPI,
                    dt.Provider_MobileNumber,
                    dt.status
                });
    foreach (var lst in data)
    {
        CustomProviderDetails objprova = new CustomProviderDetails
        {
            id = lst.Provider_ID,
            email = lst.Provider_Email,
            FirstName = lst.Provider_FirstName.ToLower(),
            Midname = lst.Provider_MidName.ToLower(),
            LastName = lst.Provider_LastName.ToLower(),
            level = lst.Provider_Level,
        };
    }
}
```

```
ciller.cs ProvidersController.cs* Source Control Explorer ACO_MembersMart.cs Notification_UserSpecific.cs
UQP.Controllers.ProvidersController PutProviderStatus(int providerid, ModifyStatus providerstatus)
// MidName = lst.Provider_MidName.ToLower(),
// LastName = lst.Provider_LastName.ToLower(),
// level = lst.Provider_Level,
// uniqueproviderno = lst.UniqueProviderID,
// providerno = lst.NPI,
// mobNo = lst.Provider_MobileNumber,
// status = lst.status
// );
// lstprovider.Add(objprova);
// }
// }

// }
// }

else
{
    // totalcount = (from p in db.Providers
    // where p.status == obj.status
    // select p.UniqueProviderID).Distinct().Count();
    // var result = db.Providers.Where(x => x.status == obj.status).GroupBy(x => x.UniqueProviderID).Select(x => x.FirstOrDefault());
    // OrderBy(x => x.UniqueProviderID).Skip(pagingparametermodel.pageSize * (pagingparametermodel.pageNumber - 1))
    // .Take(pagingparametermodel.pageSize);

    // foreach (var lst in result)
    // {
        CustomProviderDetails objprov = new CustomProviderDetails();
        objprov.id = lst.Provider_ID;
        objprov.email = lst.Provider_Email;
        objprov.FirstName = lst.Provider_FirstName.ToLower();
        objprov.MidName = lst.Provider_Midname.ToLower();
        objprov.LastName = lst.Provider_LastName.ToLower();
        objprov.level = lst.Provider_Level;
        objprov.uniqueproviderno = lst.UniqueProviderID;
        objprov.providerno = lst.NPI;
    }
}
```

```
//          //  UQP.Controllers.ProvidersController
//  //PutProviderStatus(int providerid, ModifyStatus providerstatus)
//  //OrderBy(x => x.UniqueProviderID).Skip(pagingparametermodel.pageSize * (pagingparametermodel.pageNumber - 1)).Take(pagingparametermodel.pageSize);
//
//      // foreach (var lst in result)
//      //{
//          // CustomProviderDetails objprov = new CustomProviderDetails();
//          // objprov.id = lst.Provider_ID;
//          // objprov.email = lst.Provider_Email;
//          // objprov.FirstName = lst.Provider_FirstName.ToLower();
//          // objprov.MidName = lst.Provider_MidName.ToLower();
//          // objprov.LastName = lst.Provider_LastName.ToLower();
//          // objprov.level = lst.Provider_Level;
//          // objprov.uniqueproviderno = lst.UniqueProviderID;
//          // objprov.providerno = lst.NPI;
//          // objprov.mobNo = lst.Provider_MobileNumber;
//          // objprov.status = lst.status;
//          // lstprovider.Add(objprov);
//      }
//      //-----
//      //var result = (from p in db.Providers
//      //              where p.status == obj.status
//      //              select new { p.UniqueProviderID,
//      //                          p.Provider_ID,
//      //                          p.Provider_Email,
//      //                          p.Provider_Level,
//      //                          p.Provider_MobileNumber,
//      //                          p.status,
//      //                          p.NPI
//      //                      }).GroupBy(x => x.UniqueProviderID).Select(x => x.FirstOrDefault());
//      //totalcount = 0;
//      //foreach (var lstdt in result)
//      //{
//          // var data = (from dt in db.Members_Mart
//          //             where dt.ProviderID == lstdt.UniqueProviderID
//          //             select new { dt.ProviderID,
//          //                         dt.MemberID,
//          //                         dt.MemberName,
//          //                         dt.MemberAddress,
//          //                         dt.MemberCity,
//          //                         dt.MemberState,
//          //                         dt.MemberZip,
//          //                         dt.MemberPhone,
//          //                         dt.MemberFax,
//          //                         dt.MemberEmail,
//          //                         dt.MemberType,
//          //                         dt.MemberStatus,
//          //                         dt.MemberLastUpdate,
//          //                         dt.MemberLastSync,
//          //                         dt.MemberLastSyncTime,
//          //                         dt.MemberLastSyncUser,
//          //                         dt.MemberLastSyncIP,
//          //                         dt.MemberLastSyncLogon,
//          //                         dt.MemberLastSyncLogoff,
//          //                         dt.MemberLastSyncLogonTime,
//          //                         dt.MemberLastSyncLogoffTime,
//          //                         dt.MemberLastSyncLogonUser,
//          //                         dt.MemberLastSyncLogoffUser,
//          //                         dt.MemberLastSyncLogonIP,
//          //                         dt.MemberLastSyncLogoffIP,
//          //                         dt.MemberLastSyncLogonLogoff,
//          //                         dt.MemberLastSyncLogonLogoffTime,
//          //                         dt.MemberLastSyncLogonLogoffUser,
//          //                         dt.MemberLastSyncLogonLogoffIP,
//          //                         dt.MemberLastSyncLogonLogoffLogon,
//          //                         dt.MemberLastSyncLogonLogoffLogoff,
//          //                         dt.MemberLastSyncLogonLogoffLogonTime,
//          //                         dt.MemberLastSyncLogonLogoffLogoffTime,
//          //                         dt.MemberLastSyncLogonLogoffLogonUser,
//          //                         dt.MemberLastSyncLogonLogoffLogoffUser,
//          //                         dt.MemberLastSyncLogonLogoffLogonIP,
//          //                         dt.MemberLastSyncLogonLogoffLogoffIP,
//          //                         dt.MemberLastSyncLogonLogoffLogonLogoff,
//          //                         dt.MemberLastSyncLogonLogoffLogonLogoffTime,
//          //                         dt.MemberLastSyncLogonLogoffLogonLogoffUser,
//          //                         dt.MemberLastSyncLogonLogoffLogonLogoffIP,
//          //                         dt.MemberLastSyncLogonLogoffLogonLogoffLogon,
//          //                         dt.MemberLastSyncLogonLogoffLogonLogoffLogoff,
//          //                         dt.MemberLastSyncLogonLogoffLogonLogoffLogonTime,
//          //                         dt.MemberLastSyncLogonLogoffLogonLogoffLogoffUser,
//          //                         dt.MemberLastSyncLogonLogoffLogonLogoffLogonIP,
//          //                         dt.MemberLastSyncLogonLogoffLogonLogoffLogoffLogonLogoff,
//          //                         dt.MemberLastSyncLogonLogoffLogonLogoffLogonLogoffTime,
//          //                         dt.MemberLastSyncLogonLogoffLogonLogoffLogonLogoffUser,
//          //                         dt.MemberLastSyncLogonLogoffLogonLogoffLogonLogoffIP,
//          //                         dt.MemberLastSyncLogonLogoffLogonLogoffLogonLogoffLogonLogoffLogon,
//          //                         dt.MemberLastSyncLogonLogoffLogonLogoffLogonLogoffLogonLogoffLogoff,
//          //                         dt.MemberLastSyncLogonLogoffLogonLogoffLogonLogoffLogonLogoffLogonLogoff,
//          //                         dt.MemberLastSyncLogonLogoffLogonLogoffLogonLogoffLogonLogoffLogonLogoffLogon,
//          //                         dt.MemberLastSyncLogonLogoffLogonLogoffLogonLogoffLogonLogoffLogonLogoffLogoff,
//          //                         dt.MemberLastSyncLogonLogoffLogonLogoffLogonLogoffLogonLogoffLogonLogoffLogonLogoff,
//          //                         dt.MemberLastSyncLogonLogoffLogonLogoffLogonLogoffLogonLogoffLogonLogoffLogonLogoffLogoff,
//          //                         dt.MemberLastSyncLogonLogoffLogonLogoffLogonLogoffLogonLogoffLogonLogoffLogonLogoffLogoffLogon,
//          //                         dt.MemberLastSyncLogonLogoffLogonLogoffLogonLogoffLogonLogoffLogonLogoffLogonLogoffLogoffLogoff,
//          //                         dt.MemberLastSyncLogonLogoffLogonLogoffLogonLogoffLogonLogoffLogonLogoffLogonLogoffLogoffLogoffLogon,
//          //                         dt.MemberLastSyncLogonLogoffLogonLogoffLogonLogoffLogonLogoffLogonLogoffLogonLogoffLogoffLogoffLogoff,
//          //                         dt.MemberLastSyncLogonLogoffLogonLogoffLogonLogoffLogonLogoffLogonLogoffLogonLogoffLogoffLogoffLogoff,
//          //                         dt.MemberLastSyncLogonLogoffLogonLogoffLogonLogoffLogonLogoffLogonLogoffLogonLogoffLogoffLogoffLogoffLogon,
//          //                         dt.MemberLastSyncLogonLogoffLogonLogoffLogonLogoffLogonLogoffLogonLogoffLogonLogoffLogoffLogoffLogoffLogoff,
//          //                         dt.MemberLastSyncLogonLogoffLogonLogoffLogonLogoffLogonLogoffLogonLogoffLogonLogoffLogoffLogoffLogoffLogoff,
//          //                         dt.MemberLastSyncLogonLogoffLogonLogoffLogonLogoffLogonLogoffLogonLogoffLogonLogoffLogoffLogoffLogoffLogoffLogon,
//          //                         dt.MemberLastSyncLogonLogoffLogonLogoffLogonLogoffLogonLogoffLogonLogoffLogonLogoffLogoffLogoffLogoffLogoffLogoff,
//          //                         dt.MemberLastSyncLogonLogoffLogonLogoffLogonLogoffLogonLogoffLogonLogoffLogonLogoffLogoffLogoffLogoffLogoff,
//          //                         dt.MemberLastSyncLogonLogoffLogonLogoffLogonLogoffLogonLogoffLogonLogoffLogonLogoffLogoffLogoffLogoff,
//          //                         dt.MemberLastSyncLogonLogoffLogonLogoffLogonLogoffLogonLogoffLogonLogoffLogonLogoffLogoff,
//          //                         dt.MemberLastSyncLogonLogoffLogonLogoffLogonLogoffLogonLogoffLogonLogoffLogonLogoff,
//          //                         dt.MemberLastSyncLogonLogoffLogonLogoffLogonLogoffLogonLogoffLogonLogoff,
//          //                         dt.MemberLastSyncLogonLogoffLogonLogoffLogonLogoffLogonLogoff,
//          //                         dt.MemberLastSyncLogonLogoffLogonLogoffLogonLogoff,
//          //                         dt.MemberLastSyncLogonLogoffLogonLogoffLogon,
//          //                         dt.MemberLastSyncLogonLogoffLogonLogoff,
//          //                         dt.MemberLastSyncLogonLogoffLogon,
//          //                         dt.MemberLastSyncLogonLogoff,
//          //                         dt.MemberLastSyncLogon,
//          //                         dt.MemberLastSync
//          //                     };
//      }
//  }
//}
```

View Project Build Debug Team Tools Test Analyze Window Help

Debug Any CPU IIS Express (Google Chrome)

ProvidersController.cs Source Control Explorer ACO_MembersMart.cs Notification_UserSpecific.cs

PutProviderStatus(int providerid, ModifyStatus providerstatus)

```
17 //      var data = (from dt in db.Members_Mart
18 //                  where dt.TimeStamp >= obj.startdate && dt.TimeStamp <= obj.enddate && dt.provider_id == lstdt.Provider_ID
19 //                  select new
20 //                  {
21 //                      lstdt.Provider_ID,
22 //                      lstdt.Provider_Email,
23 //                      dt.ProviderFirstName,
24 //                      dt.ProviderMiddleName,
25 //                      dt.ProviderLastName,
26 //                      lstdt.Provider_Level,
27 //                      dt.UniqueProviderID,
28 //                      lstdt.NPI,
29 //                      lstdt.Provider_MobileNumber,
30 //                      lstdt.status
31 //                  }).Distinct(); ;
32
33 foreach (var lst in data)
34 {
35     totalcount += 1;
36     CustomProviderDetails objprova = new CustomProviderDetails
37     {
38         id = lst.Provider_ID,
39         email = lst.Provider_Email,
40         FirstName = lst.ProviderFirstName.ToLower(),
41         MidName = lst.ProviderMiddleName.ToLower(),
42         LastName = lst.ProviderLastName.ToLower(),
43         level = lst.Provider_Level,
44         uniqueproviderno = lst.UniqueProviderID,
45         providereno = lst.NPI,
46         mobNo = lst.Provider_MobileNumber,
47         status = lst.status
48     }
49 }
```

```
ersController.cs ProvidersController.cs + X Source Control Explorer ACO_MembersMart.cs Notification_UserSpecific.cs
P UQP.Controllers.ProvidersController PutProviderStatus(int providerid, ModifyStatus providerstatus)
51 //           //      CustomProviderDetails objprova = new CustomProviderDetails
52 //           //      {
53 //           //          id = lst.Provider_ID,
54 //           //          email = lst.Provider_Email,
55 //           //          FirstName = lst.ProviderFirstName.ToLower(),
56 //           //          MidName = lst.ProviderMiddleName.ToLower(),
57 //           //          LastName = lst.ProviderLastName.ToLower(),
58 //           //          level = lst.Provider_Level,
59 //           //          uniqueproviderno = lst.UniqueProviderID,
60 //           //          providerno = lst.NPI,
61 //           //          mobNo = lst.Provider_MobileNumber,
62 //           //          status = lst.status
63 //           //      };
64 //           lstprovider.Add(objprova);
65 //       }
66 //   }
67 // }
68 // }
69 // }
70 // }
71 // }
72 // }
73 // }
74 // }
75 // }
76 // }
77 // }
78 // }
79 // -----
80 // var result = (from mm in db.Members_Mart
81 //               join pr in db.Providers on mm.provider_id equals pr.Provider_ID
82 //               where mm.TimeStamp.Year == obj.startdate.Year
83 //               && mm.TimeStamp.Month == obj.startdate.Month && pr.status == obj.status
84 //               select new
85 //               {
86 //                   pr.UniqueProviderID,
87 //                   pr.Provider_ID,
88 //                   pr.NPI,
89 //                   pr.Provider_Email,
90 //                   pr.Provider_Level,
91 //                   pr.Provider_MobileNumber,
92 //                   pr.status,
93 //                   pr.Provider_FirstName,
94 //                   pr.Provider_LastName,
95 //                   pr.Provider_MidName
96 //               }
97 // }
```

```
Controller.cs    ProvidersController.cs*  X  Source Control Explorer    ACO_MembersMart.cs    Notification_UserSpecific.cs
      ↴ UQP.Controllers.ProvidersController
      ↴ PutProviderStatus(int providerid, ModifyStatus providerstatus)

1 // select new
2 {
3     pr.UniqueProviderID,
4     pr.Provider_ID,
5     pr.NPI,
6     pr.Provider_Email,
7     pr.Provider_Level,
8     pr.Provider_MobileNumber,
9     pr.status,
10    pr.Provider_FirstName,
11    pr.Provider_LastName,
12    pr.Provider_MidName
13 }).GroupBy(x => x.UniqueProviderID).Select(x => x.FirstOrDefault());
14 OrderBy(x => x.UniqueProviderID).Skip(pagingparametermodel.pageSize * (pagingparametermodel.pageNumber - 1))
15
16 //].Take(pagingparametermodel.pageSize);
17
18 // totalcount = (from mm in db.Members_Mart
19 // join pr in db.Providers on mm.provider_id equals pr.Provider_ID
20 // where mm.TimeStamp.Year == obj.startdate.Year
21 // && mm.TimeStamp.Month == obj.startdate.Month && pr.status == obj.status
22 // select new
23 // {
24 //     pr.UniqueProviderID
25 // }).Distinct().Count();
26
27
28 // foreach (var lst in result)
29 // {
30 //     CustomProviderDetails objprova = new CustomProviderDetails
31 //     {
32 //         id = lst.Provider_ID,
33 //         email = lst.Provider_Email,
34 //         FirstName = lst.Provider_FirstName.ToLower(),
35 //         MidName = lst.Provider_MidName.ToLower(),
36 //         LastName = lst.Provider_LastName.ToLower(),
37 //     }
38 // }
```


Debug Any CPU IIS Express (Google Chrome)

ProvidersController.cs

Source Control Explorer

ACO_MembersMart.cs

Notification_UserSpecific.cs

UQP.Controllers.ProvidersController

PutProviderStatus(int providerid, ModifyStatus providerstatus)

```
1 // dt.Provider_FirstName,
2 // dt.Provider_MidName,
3 // dt.Provider_LastName,
4 // dt.Provider_Level,
5 // dt.UniqueProviderID,
6 // dt.NPI,
7 // dt.Provider_MobileNumber,
8 // dt.status
9 // });

0 foreach (var lst in data)
1 {
2     CustomProviderDetails objprova = new CustomProviderDetails
3     {
4         id = lst.Provider_ID,
5         email = lst.Provider_Email,
6         FirstName = lst.Provider_FirstName.ToLower(),
7         MidName = lst.Provider_MidName.ToLower(),
8         LastName = lst.Provider_LastName.ToLower(),
9         level = lst.Provider_Level,
10        uniqeproviderno = lst.UniqueProviderID,
11        providerno = lst.NPI,
12        mobNo = lst.Provider_MobileNumber,
13        status = lst.status
14    };
15    lstprovider.Add(objprova);
16 }
17 }

18 // }
19 // }
20 // }

21 // else
```



```
//         result = db.Providers.GroupBy(x => x.UniqueProviderID).Select(x => x.FirstOrDefault());
//         OrderBy(x => x.UniqueProviderID).Skip(pagingparametermodel.pageSize * (pagingparametermodel.pageNumber - 1)).Take(pagingparametermode
//     }
//     foreach (var lst in result)
//     {
//         objprov = new CustomProviderDetails();
//         objprov.id = lst.Provider_ID;
//         objprov.email = lst.Provider_Email;
//         objprov.FirstName = lst.Provider_FirstName.ToLower();
//         objprov.MidName = lst.Provider_MidName.ToLower();
//         objprov.LastName = lst.Provider_LastName.ToLower();
//         objprov.level = lst.Provider_Level;
//         objprov.uniqueproviderno = lst.UniqueProviderID;
//         objprov.providererno = lst.NPTI;
//         objprov.mobNo = lst.Provider_MobileNumber;
//         objprov.status = lst.status;
//         lstprovider.Add(objprov);
//     }
//     // objprov.count = lstprovider.Count();
//     else
//     {
//         totalcount = (from p in db.Providers where p.status==obj.status
//                     select p.UniqueProviderID).Distinct().Count();
//         var result = db.Providers.Where(x => x.status == obj.status).GroupBy(x => x.UniqueProviderID).Select(x => x.FirstOrDefault());
//         OrderBy(x => x.UniqueProviderID).Skip(pagingparametermodel.pageSize * (pagingparametermodel.pageNumber - 1))].
//         Take(pagingparametermodel.pageSize);
//         var subresult = result.Where(x => x.status == obj.status);
//         foreach (var lst in subresult)
//         {
//             objprov = new CustomProviderDetails();
//             objprov.id = lst.Provider_ID;
//             objprov.email = lst.Provider_Email;
//             objprov.FirstName = lst.Provider_FirstName.ToLower();
//             objprov.MidName = lst.Provider_MidName.ToLower();
//             objprov.LastName = lst.Provider_LastName.ToLower();
//         }
//     }
// }
```

```
UQP.Controllers.ProvidersController          PutProviderStatus(int providerid, ModifyStatus providerstatus)
// var result = db.Providers.Where(x => x.status == obj.status).GroupBy(x => x.UniqueProviderID).Select(x => x.FirstOrDefault());
// OrderBy(x => x.UniqueProviderID).Skip(pagingparametermodel.pageSize * (pagingparametermodel.pageNumber - 1)).Take(pagingparametermodel.pageSize);
// var subresult = result.Where(x => x.status == obj.status);
// foreach (var lst in subresult)
// {
//     objprov = new CustomProviderDetails();
//     objprov.id = lst.Provider_ID;
//     objprov.email = lst.Provider_Email;
//     objprov.FirstName = lst.Provider_FirstName.ToLower();
//     objprov.MidName = lst.Provider_MidName.ToLower();
//     objprov.LastName = lst.Provider_LastName.ToLower();
//     objprov.level = lst.Provider_Level;
//     objprov.uniqueproviderno = lst.UniqueProviderID;
//     objprov.providerno = lst.NPI;
//     objprov.mobNo = lst.Provider_MobileNumber;
//     objprov.status = lst.status;
//     lstprovider.Add(objprov);
// }
// }

// // Get's No of Rows Count
// int count = totalcount;
// // Parameter is passed from Query string if it is null then it default Value will be pageNumber:1
// int CurrentPage = pagingparametermodel.pageNumber;
// // Parameter is passed from Query string if it is null then it default Value will be pageSize:20
// int PageSize = pagingparametermodel.pageSize;
// // Display TotalCount to Records to User
// int TotalCount = count;
// // Calculating Totalpage by Dividing (No of Records / Pagesize)
```

```
UQP UQP.Controllers.ProvidersController PutProviderStatus(int providerid, ModifyStatus providestatus)
647 //      }
648 //    }
649
650
651
652 //        // Get's No of Rows Count
653 //        int count = totalcount;
654
655 //        // Parameter is passed from Query string if it is null then it default Value will be pageNumber:1
656 //        int CurrentPage = pagingparametermodel.pageNumber;
657
658 //        // Parameter is passed from Query string if it is null then it default Value will be pageSize:20
659 //        int PageSize = pagingparametermodel.pageSize;
660
661 //        // Display TotalCount to Records to User
662 //        int TotalCount = count;
663
664 //        // Calculating Totalpage by Dividing (No of Records / Pagesize)
665 //        int TotalPages = (int)Math.Ceiling(count / (double)PageSize);
666
667 //        // Returns List of Customer after applying Paging
668 //        var items = lstprovider;//.Skip((CurrentPage - 1) * PageSize).Take(PageSize).ToList();
669
670 //        // if CurrentPage is greater than 1 means it has previousPage
671 //        var previousPage = CurrentPage > 1 ? "Yes" : "No";
672
673 //        // if TotalPages is greater than CurrentPage means it has nextPage
674 //        var nextPage = CurrentPage < TotalPages ? "Yes" : "No";
675
676 //        // Object which we are going to send in header
677 //        var paginationMetadata = new
678 {
679     //          totalCount = TotalCount,
680     //          pageSize = PageSize,
681     //          currentPage = CurrentPage,
682     //          totalPages = TotalPages,
```

ProvidersController.cs

Source Control Explorer

ACO_MembersMart.cs

Notification_UserSpecific.cs

UOP.Controllers.ProvidersController

PutProviderStatus(int providerid, ModifyStatus providerstatus)

```
677 //         var paginationMetadata = new
678 //         {
679 //             totalCount = TotalCount,
680 //             pageSize = PageSize,
681 //             currentPage = CurrentPage,
682 //             totalPages = TotalPages,
683 //             previousPage,
684 //             nextPage
685 //         };
686 //         // Setting Header
687 //         HttpContext.Current.Response.Headers.Add("Paging-Headers", Newtonsoft.Json.JsonConvert.SerializeObject(paginationMetadata));
688 //         //HttpContext.Current.Response.Headers.Add("Filename", zipFileName);
689 //         //HttpContext.Current.Response.Headers.Add("Access-Control-Expose-Headers", "Paging-Headers");
690 //         // Returing List of Customers Collections
691 //         return items;
692
693
694
695
696
697
698
699
700
701
702
703
704
705
706
707
708
709
710
711
712 }
```

```
710
711
712
713 [HttpPost]
714 [Route("api/PutProviderStatus/{providerid}")]
715 References
716 public HttpResponseMessage PutProviderStatus(int providerid, ModifyStatus providerstatus)
717 {
718     if (!ModelState.IsValid)
719     {
720         return Request.CreateResponse(HttpStatusCode.BadRequest, "Inputs are invalid");
721     }
722
723     try
724     {
725         using (UQPEntities db = new UQPEntities(HttpContext.Current.Request.Headers.GetValues("IPA").FirstOrDefault()))
726         {
727             var entity = (from pr in db.ProviderDetailsTestings where pr.providerid == providerid select pr).ToList();
728
729             if (entity.Count() != 0)
730             {
731                 foreach (var per in entity)
732                 {
733                     per.status = providerstatus.Status;
734                     per.msgstatus = providerstatus.message;
735                     per.notifstatus = providerstatus.notification;
736                     db.SaveChanges();
737
738                 }
739
740             //var entity = (from pr in db.Providers
741             //              where pr.UniqueProviderID == UniqueProviderID
742             //              select pr).ToList();
743         }
744     }
745 }
```

output from: Source Control - Team Foundation

```
39 }  
40  
41 //var entity = (from pr in db.Providers  
42 //      where pr.UniqueProviderID == UniqueProviderID  
43 //      select pr).ToList();  
44  
45 //var newprovidertable_query = (from npr in db.NewProviderTemps  
46 //      where npr.UniqueProviderID == UniqueProviderID  
47 //      select npr).ToList();  
48 //if (entity.Count() != 0 || newprovidertable_query.Count() != 0)  
49 //{
/>50 //    foreach (var per in entity)  
51 //    {  
52 //        per.status = providerstatus.Status;  
53 //        per.MsgStatus = providerstatus.message;  
54 //        per.NotifStatus = providerstatus.notification;  
55 //        db.SaveChanges();  
56  
57 //    }  
58 //    foreach (var nper in newprovidertable_query)  
59 //    {  
60 //        nper.Status = providerstatus.Status;  
61 //        nper.MsgStatus = providerstatus.message;  
62 //        nper.NotifStatus = providerstatus.notification;  
63 //        db.SaveChanges();  
64 //    }  
65 //} else  
66 //return Request.CreateResponse(HttpStatusCode.OK, "Provider status has been updated successfully");  
67 }  
68  
69 else  
70 {  
71     return Request.CreateResponse(HttpStatusCode.NotFound,  
72         "Provider ID= " + providerid + " not found"  
73     );  
74 }
```

UQP - Microsoft Visual Studio

File Edit View Project Build Debug Team Tools Test Analyze Window Help

IIS Express (Google Chrome)

MembersController.cs ProvidersController.cs Source Control Explorer ACO_MembersMart.cs Notification_UserSpecific.cs

UQP

```
760
761     // 
762     // nper.Status = providerstatus.Status;
763     // nper.MsgStatus = providerstatus.message;
764     // nper.NotifStatus = providerstatus.notification;
765     // db.SaveChanges();
766
767     return Request.CreateResponse(HttpStatusCode.OK, "Provider status has been updated successfully");
768 }
769 else
770 {
771     return Request.CreateResponse(HttpStatusCode.NotFound,
772         "Provider ID= " + providerid + " not found"
773     );
774 }
775
776 }
777 catch (Exception ex)
778 {
779     return Request.CreateErrorResponse(HttpStatusCode.InternalServerError, ex);
780 }
781
782
783 // GET: api/Star Score
```

```
MembersController.cs  ProvidersController.cs*  X  Source Control Explorer  ACO_MembersMart.cs  Notification_UserSpecific.cs
UOP  UQP.Controllers.ProvidersController  PutProviderStatus(int providerid, ModifyStatus pm)
781 }
782
783
784 // GET: api/Star_Score
785 [HttpPost]
786 [Route("api/insuranceplan")]
787 References
788 public HttpResponseMessage InsurancePlan(InputDate objdate)
789 {
790     try
791     {
792         using (UQPEntities db = new UQPEntities(HttpContext.Current.Request.Headers.GetValues("IPA").FirstOrDefault()))
793         {
794             dynamic query = null;
795             List<CustomInsurance> lstcustomins = new List<CustomInsurance>();
796             if (objdate.ipaid != 0)
797             {
798                 query = (from hp in db.insurancebyipa_membercount.AsNoTracking()
799                         where hp.Ipa_Id == objdate.ipaid && hp.timestamp == objdate.startdate
800                         && hp.totalMembership > 0
801                         select new
802                         {
803                             HealthPlanId = hp.healthplan_id,
804                             hp.HealthPlan_Name,
805                             hp.SNP_Members,
806                             hp.NonSNP_members,
807                             hp.totalMembership
808                         }).OrderBy(e => e.HealthPlan_Name).ToList();
809             }
810             if (objdate.providerid != 0)
811             {
812                 query = (from hp in db.insurancebyprovider_membercount.AsNoTracking()
813                         where hp.provider_Id == objdate.providerid && hp.timestamp == objdate.startdate
814                         && hp.totalMembership > 0
815                         select new
816                         {
817
818
819
820
821
822
823
824
825
826
827
828
829
830
831
832
833
834
835
836
837
838
839
840
841
842
843
844
845
846
847
848
849
850
851
852
853
854
855
856
857
858
859
860
861
862
863
864
865
866
867
868
869
870
871
872
873
874
875
876
877
878
879
880
881
882
883
884
885
886
887
888
889
890
891
892
893
894
895
896
897
898
899
900
901
902
903
904
905
906
907
908
909
910
911
912
913
914
915
916
917
918
919
920
921
922
923
924
925
926
927
928
929
930
931
932
933
934
935
936
937
938
939
940
941
942
943
944
945
946
947
948
949
950
951
952
953
954
955
956
957
958
959
960
961
962
963
964
965
966
967
968
969
970
971
972
973
974
975
976
977
978
979
980
981
982
983
984
985
986
987
988
989
990
991
992
993
994
995
996
997
998
999
999
```

```
SCController.cs  ProvidersController.cs*  X  Source Control Explorer  ACO_MembersMart.cs  Notification_UserSpecific.cs  InsurancePlan(InputDate objdate)
UOP.Controllers.ProvidersController
}
    hp.SNP_Members,
    hp.NonSNP_members,
    hp.totalMembership
}).OrderBy(e => e.HealthPlan_Name).ToList();

}

if (objdate.providerid != 0)
{
    query = (from hp in db.insurancebyprovider_membercount.AsNoTracking()
              where hp.provider_id == objdate.providerid && hp.timestamp == objdate.startdate
&ampamp hp.totalMembership > 0
              select new
              {
                  HealthPlanId = hp.healthplan_id,
                  hp.HealthPlan_Name,
                  hp.SNP_Members,
                  hp.NonSNP_members,
                  hp.totalMembership
              }).OrderBy(e => e.HealthPlan_Name).ToList();
}
else if (objdate.healthplanid != 0)
{
    query = (from hp in db.newHealthPlan_membercount.AsNoTracking()
              where hp.countmonth == objdate.startdate && hp.HealthPlanId == objdate.healthplanid
              select new { hp.HealthPlanId, hp.HealthPlan_Name, hp.SNP_Members, hp.NonSNP_members, hp.totalMembership }).ToList();
}
else if (objdate.healthplanid == 0 && objdate.providerid == 0 && objdate.ipaid == 0)
{
    query = (from hp in db.newHealthPlan_membercount.AsNoTracking()
              where hp.countmonth == objdate.startdate && hp.totalMembership > 0
              select new { hp.HealthPlanId, hp.HealthPlan_Name, hp.SNP_Members, hp.NonSNP_members, hp.totalMembership }).OrderBy(e =>
e.HealthPlan_Name).ToList();
}

}

```

The screenshot shows a Microsoft Visual Studio interface with the following details:

- Menu Bar:** File, View, Project, Build, Debug, Team, Tools, Test, Analyze, Window, Help.
- Toolbars:** Standard, Debug, Task List, Solution Explorer, Properties, Toolbox, Task List, Status Bar.
- Status Bar:** Scanned with CamScanner

The main code editor window displays the file `ProvidersController.cs` with the following content:

```
UQP
else if (objdate.newHealthPlanId == 0 && objdate.providerId == 0 && objdate.ipId == 0)
{
    query = (from hp in db.newHealthPlan_membercount.AsNoTracking()
              where hp.countmonth == objdate.startdate && hp.totalMembership > 0
              select new { hp.HealthPlanId, hp.HealthPlan_Name, hp.SNP_Members, hp.NonSNP_members, hp.totalMembership }).ToList();
}

foreach (var q in query)
{
    CustomInsurance objcustominsurance2 = new CustomInsurance();
    objcustominsurance2.Insurance_id = Convert.ToInt32(q.HealthPlanId);
    objcustominsurance2.Insurance_Name = q.HealthPlan_Name;
    objcustominsurance2.SNPCount = q.SNP_Members;
    objcustominsurance2.MemberCount = Convert.ToInt32(q.totalMembership);
    objcustominsurance2.NonSNPCount = q.NonSNP_members;

    lstcustomins.Add(objcustominsurance2);
}
return Request.CreateResponse(HttpStatusCode.OK, lstcustomins);
}
catch (Exception ex)
{
    return Request.CreateResponse(HttpStatusCode.BadRequest, ex.InnerException);
}
}
```

```
UQP UQP.Controllers.ProvidersController InsurancePlan(inputData objData)
861     return Request.CreateResponse(HttpStatusCode.BadRequest, ex.InnerException);
862 }
863 }
864
865 // PUT: api/Providers/5
866 [ResponseType(typeof(void))]
867 [References]
868 public async Task<IHttpActionResult> PutProvider(long id, Provider provider)
869 {
870     using (UQPEntities db = new UQPEntities(HttpContext.Current.Request.Headers.GetValues("IPA").FirstOrDefault()))
871     {
872         if (!ModelState.IsValid)
873         {
874             return BadRequest(ModelState);
875         }
876
877         if (id != provider.Provider_ID)
878         {
879             return BadRequest();
880         }
881         db.Entry(provider).State = EntityState.Modified;
882
883         try
884         {
885             await db.SaveChangesAsync();
886         }
887         catch (DbUpdateConcurrencyException)
888         {
889             if (!ProviderExists(id))
890             {
891                 return NotFound();
892             }
893             else
894             {
895                 throw;
896             }
897         }
898     }
899 }
```

```
    await db.SaveChangesAsync();
}
catch (DbUpdateConcurrencyException)
{
    if (!ProviderExists(id))
    {
        return NotFound();
    }
    else
    {
        throw;
    }
}

return StatusCode(HttpStatusCode.NoContent);
}

[HttpGet]
[Route("api/Search/{type}/{fname}")]
public HttpResponseMessage SearchProvider(string type, string fname)
{
    try
    {
        using (UQPEntities db = new UQPEntities(HttpContext.Current.Request.Headers.GetValues("IPA").FirstOrDefault()))
        {
            List<CustomProviderDetails> lstprovider = new List<CustomProviderDetails>();
            if (type == "Provider")
            {
                var result = (from p in db.ProviderDetailsTestings.AsNoTracking()
                            where p.ProviderName.StartsWith(fname))

```


MembersController.cs ProvidersController.cs* Source Control Explorer ACO_MembersV1Art.cs Notification_UserSpecific.cs

UQP UQP.Controllers.ProvidersController InsurancePlan(InputData objdate)

```
926
927
928
929
930 foreach (var lst in result)
{
    objprov = new CustomProviderDetails();
    objprov.id = lst.providerid;
    objprov.email = (lst.emailid == "" || lst.emailid == null) ? "" : lst.emailid;
    objprov.FirstName = (lst.Physician_first_name==null)? "": textInfo.ToTitleCase(lst.Physician_first_name.ToLower());
    objprov.LastName = (lst.Physician_last_name==null)? "": textInfo.ToTitleCase(lst.Physician_last_name.ToLower());
    objprov.level = lst.Title;
    objprov.providerno = Convert.ToInt64(lst.NPI);
    objprov.status = lst.status;
    objprov.count = result.Count();
    lstprovider.Add(objprov);
}
}
if (type == "ACOProvider")
{
    var result = db.ACOProviders.AsNoTracking()
        .Where(x => x.C_IND_NPI != null && (x.firstname.Trim().StartsWith(fname)
    || x.lastname.Trim().StartsWith(fname) ||
    (x.firstname.Trim() + " " + x.lastname.Trim()).StartsWith(fname)))
    .Select(x => new { x.firstname, x.lastname, x.C_IND_NPI }).Distinct().ToList();
    CustomProviderDetails objprov = null;
```

```
950
951
952
953     }
954
955     if (type == "ACOProvider")
956     {
957
958         var result = db.ACOProviders.AsNoTracking()
959             .Where(x => x.C_IND_NPI != null && (x.firstname.Trim().StartsWith(fname)
960             || x.lastname.Trim().StartsWith(fname) ||
961             (x.firstname.Trim() + " " + x.lastname.Trim()).StartsWith(fname)))
962             .Select(x => new { x.firstname, x.lastname, x.C_IND_NPI }).Distinct().ToList();
963
964         CustomProviderDetails objprov = null;
965         foreach (var lst in result)
966         {
967             objprov = new CustomProviderDetails();
968             // objprov.id = Convert.ToInt64(lst.C_IND_NPI);
969             // objprov.email =
970             //     (lst.EMAIL == "" || lst.EMAIL == null) ? "" : lst.EMAIL;
971             objprov.FirstName = textInfo.ToTitleCase(lst.firstname.ToLower());
972             objprov.LastName = textInfo.ToTitleCase(lst.lastname.ToLower());
973
974             objprov.count = result.Count();
975             lstprovider.Add(objprov);
976         }
977
978     }
979
980
981
982
983
984
985 }
```

The screenshot shows a Microsoft Visual Studio interface with the following details:

- Project Explorer:** Shows 'UQP' as the active project.
- Solution Explorer:** Shows files like 'MembersController.cs', 'ProvidersController.cs*', 'Source Control Explorer', 'ACO_MembersMart.cs', 'Notification_UserSpecific.cs', and 'InsurancePlan(InputData objdata)'.
- Code Editor:** Displays the 'ProvidersController.cs' file content. The code is a C# class for an ASP.NET Web API controller. It includes methods for handling provider search requests based on ID and first name, and a method for creating responses. Error handling is provided for exceptions.
- Status Bar:** Shows '100 %' and a small icon.

```
983     }
984     }
985     }
986     }
987     return Request.CreateResponse(HttpStatusCode.OK, lstprovider);
988 }
989 }
990 catch (Exception ex)
991 {
992     return Request.CreateResponse(HttpStatusCode.InternalServerError, ex.Message);
993 }
994 }
995
996 // Below api will return list according to the healthplan
997 // [HttpGet]
998 // [Route("api/SearchProviderByHp/{id}/{fname}")]
999 // public List<CustomProviderDetails> SearchProviderByHp(int id, string fname)
1000 //{
1001 //     List<CustomProviderDetails> lstprovider = new List<CustomProviderDetails>();
1002 //     dynamic result;
1003 //     if (id != 0)
1004 //     {
1005 //         result = (from i in db.insuranceIpabyprovider_membercount
1006 //                   join prov in db.Providers on i.provider_Id equals prov.Provider_ID
1007 //                   where
1008 //                     i.healthplan_id == id &&
1009 //                     (
1010 //                         prov.Provider_FirstName.StartsWith(fname) ||
1011 //                         prov.Provider_MidName.StartsWith(fname) ||
1012 //                         prov.Provider_LastName.StartsWith(fname) ||
1013 //                         (prov.Provider_FirstName + " " + prov.Provider_MidName + " " + prov.Provider_LastName).StartsWith(fname) ||
1014 //                         (prov.Provider_FirstName + " " + prov.Provider_MidName).StartsWith(fname)
1015 //                     );
1016 //         select prov
1017 //     }.Distinct();
1018 // }
```

Debug Any CPU IIS Express (Google Chrome) UQP

MembersController.cs ProvidersController.cs* Source Control Explorer ACO_MembersMart.cs Notification_UserSpecific.cs

UQP

```
1013 // (prov.Provider_FirstName + " " + prov.Provider_MidName + " " + prov.Provider_LastName).StartsWith(fname) ||
1014 // (prov.Provider_FirstName + " " + prov.Provider_MidName).StartsWith(fname)
1015 // )
1016 // select prov
1017 // ).Distinct();
1018 // }
1019 // else
1020 // {
1021 //     result = (db.Providers.GroupBy(x => x.UniqueProviderID).Select(x => x.FirstOrDefault()).Where
1022 //     (x1 => ((x1.Provider_FirstName.StartsWith(fname)) ||
1023 //     (x1.Provider_LastName.StartsWith(fname)) ||
1024 //     (x1.Provider_MidName.StartsWith(fname))
1025 // ))).ToList();
1026 // }
1027 // foreach (var lst in result)
1028 // {
1029 //     CustomProviderDetails objprov = new CustomProviderDetails();
1030 //     objprov.id = lst.Provider_ID;
1031 //     objprov.FirstName = textInfo.ToTitleCase(lst.Provider_FirstName.ToLower());
1032 //     objprov.MidName = textInfo.ToTitleCase(lst.Provider_MidName.ToLower());
1033 //     objprov.LastName = textInfo.ToTitleCase(lst.Provider_LastName.ToLower());
1034 //     objprov.level = lst.Provider_Level;
1035 //     objprov.uniqueproviderno = lst.UniqueProviderID;
1036 //     objprov.mobNo = lst.Provider_MobileNumber;
1037 //     lstprovider.Add(objprov);
1038 // }
1039 // return lstprovider;
1040 //}
1041 [HttpGet]
1042 [Route("api/SearchProviderByHp/{id}/{fname}")]
1043 Or references
1044 public HttpResponseMessage SearchProviderByHp(int id, string fname)
1045 {
1046     try
1047     {
```

```
34 //      objprov.level = lst.Provider_Level;
35 //      objprov.uniqueproviderno = lst.UniqueProviderID;
36 //      objprov.mobNo = lst.Provider_MobileNumber;
37 //      lstprovider.Add(objprov);
38 //  }
39 //  return lstprovider;
40 //}
41
42 [HttpGet]
43 [Route("api/SearchProviderByHp/{id}/{fname}")]
44 References
45 public HttpResponseMessage SearchProviderByHp(int id, string fname)
46 {
47     try
48     {
49         using (UQPEntities db = new UQPEntities(HttpContext.Current.Request.Headers.GetValues("IPA").FirstOrDefault()))
50         {
51             List<CustomProviderDetails> lstprovider = new List<CustomProviderDetails>();
52             dynamic result;
53             string hid = Convert.ToString(id);
54             if (id != 0)
55             {
56                 result = (from p in db.ProviderDetailsTestings.AsNoTracking()
57                           where p.healthplanid.Contains(hid) && p.visible == true &&
58                           (p.Physician_first_name.StartsWith(fname) || p.Physician_last_name.StartsWith(fname) || (p.Physician_first_name + " " +)
59                           p.Physician_last_name).StartsWith(fname))
60                 select new
61                 {
62                     p.providerid,
63                     p.Physician_first_name,
64                     p.Physician_last_name,
65                     p.Title,
66                     //p.UniqueProviderID,
67                     //p.Phone
68                 }).Distinct().ToList();
69             }
70         }
71     }
72 }
```

```
1066 } .Distinct().ToList();
1067 }
1068 }
1069 {
1070     result = (from p in db.ProviderDetailsTestings.AsNoTracking()
1071             where
1072                 (p.Physician_first_name.StartsWith(fname) ||
1073                  p.Physician_last_name.StartsWith(fname)) ||
1074                  (p.Physician_first_name + " " + p.Physician_last_name).StartsWith(fname)
1075                  && p.visible == true)
1076         select new
1077         {
1078             p.providerid,
1079             p.Physician_first_name,
1080             p.Physician_last_name,
1081             p.Title,
1082             //p.UniqueProviderID,
1083             //p.Phone
1084         }).Distinct().ToList();
1085 }
1086 }
1087 }
1088 }
1089 foreach (var lst in result)
1090 {
1091     CustomProviderDetails objprov = new CustomProviderDetails();
1092     if (lst.providerid != null || lst.providerid != 0)
1093     {
1094         objprov.id = Convert.ToInt64(lst.providerid);
1095     }
1096     objprov.FirstName = (lst.Physician_first_name==null)? "" : textInfo.ToTitleCase(lst.Physician_first_name.ToLower());
1097     objprov.LastName = (lst.Physician_last_name==null)? "" : textInfo.ToTitleCase(lst.Physician_last_name.ToLower());
1098     objprov.level = lst.Title;
1099     objprov.uniqueproviderno = null;
1100     //objprov.mobNo = lst.Phone;
1101 }
```

100 %

Output

-- Control - Team Foundation

Microsoft Visual Studio

File View Project Build Debug Team Tools Test Analyze Window Help

Debug Any CPU IIS Express (Google Chrome)

MembersController.cs ProvidersController.cs Source Control Explorer ACO_MembersMart.cs Notification_UserSpecific.cs

UQP

```
1087 }  
1088  
1089 foreach (var lst in result)  
1090 {  
1091     CustomProviderDetails objprov = new CustomProviderDetails();  
1092     if (lst.providerid != null || lst.providerid != 0)  
1093     {  
1094         objprov.id = Convert.ToInt64(lst.providerid);  
1095         objprov.FirstName =(lst.Physician_first_name==null)? "" : textInfo.ToTitleCase(lst.Physician_first_name.ToLower());  
1096         objprov.LastName =(lst.Physician_last_name==null)? "" : textInfo.ToTitleCase(lst.Physician_last_name.ToLower());  
1097         objprov.level = lst.Title;  
1098         objprov.uniqueproviderno = null;  
1099         //objprov.mobNo = lst.Phone;  
1100         lstprovider.Add(objprov);  
1101     }  
1102 }  
1103 return Request.CreateResponse(HttpStatusCode.OK, lstprovider);  
1104 }  
1105 }  
1106 catch (Exception ex)  
1107 {  
1108     return Request.CreateResponse(HttpStatusCode.InternalServerError, ex.Message);  
1109 }  
1110 }  
1111 }  
1112  
1113  
1114  
1115  
1116  
1117  
1118 [HttpGet]  
1119 [Route("api/providerlist/")]
```


The screenshot shows a Microsoft Visual Studio interface with the following details:

- Open Files:** ProvidersController.cs (highlighted in orange), Source Control Explorer, ACO_MembersMait.cs, and Notification_UserSpecific.cs.
- Current File:** ProvidersController.cs
- Code Preview:** The code is displayed with line numbers from 1146 to 1181. It includes logic for handling provider status, report generation, and pagination, followed by a foreach loop that maps database results to a CustomProviderDetails object.
- Toolbars and Status:** Standard Visual Studio toolbars are visible at the top. The status bar at the bottom left shows "100 %".
- Output Tab:** An "Output" tab is visible at the bottom left.

```
1146 //if (obj.status == "all")
1147 //{
1148 //    obj.status1 =null;
1149 //    obj.status2 = null;
1150 //}
1151 if (obj.report == true)
1152 {
1153     obj.ispagination = false;
1154 }
1155 else
1156 {
1157     obj.ispagination = true;
1158 }
1159
1160 var query1 = db.USP_Providerlist(obj.startdate, obj.enddate, obj.providerid, (obj.healthplanid.ToString()), obj.ipaid, obj.name,
1161     bbj.status, obj.status1, obj.status2, obj.ispagination,
1162     pagingparametermodel.pageNumber, pagingparametermodel.pageSize, totalcount).ToList();
1163
1164
1165 foreach (var lst in query1)
1166 {
1167     CustomProviderDetails objprova = new CustomProviderDetails();
1168
1169     objprova.id = Convert.ToInt64(lst.ProviderID);
1170     objprova.email = lst.emailid;
1171     objprova.FirstName =(lst.Physician_first_name==null)? "" : lst.Physician_first_name.ToLower();
1172     objprova.LastName =(lst.Physician_last_name == null) ? "" : lst.Physician_last_name.ToLower();
1173     objprova.level = lst.Title;
1174
1175     objprova.providerno = (lst.NPI == " " || lst.NPI == null) ? 0 : Convert.ToInt64(lst.NPI);
1176     objprova.mobNo = lst.phone;
1177     objprova.status = lst.status;
1178     objprova.notification = lst.NotifStatus;
1179     objprova.message = lst.MsgStatus;
1180
1181     objprova.address = lst.address;
```

```
8
9     objprova.id = Convert.ToInt64(lst.ProviderID);
10    objprova.email = lst.emailid;
11    objprova.FirstName = (lst.Physician_first_name==null) ? "" : lst.Physician_first_name.ToLower();
12    objprova.LastName = (lst.Physician_last_name == null) ? "" : lst.Physician_last_name.ToLower();
13    objprova.level = lst.Title;
14
15    objprova.providerno = (lst.NPI == " " || lst.NPI == null) ? 0 : Convert.ToInt64(lst.NPI);
16    objprova.mobNo = lst.phone;
17    objprova.status = lst.status;
18    objprova.notification = lst.NotifStatus;
19    objprova.message = lst.MsgStatus;
20    objprova.address = lst.address;
21
22    lstprovider.Add(objprova);
23 }
24
25
26 if (obj.report == false)
27 {
28     #region pagination
29     // Get's No of Rows Count
30     int count = Convert.ToInt32(totalcount.Value);
31     //int count = lstmembermart.Count();
32
33     // Parameter is passed from Query string if it is null then it default Value will be pageNumber:1
34     int CurrentPage = pagingparametermodel.pageNumber;
35
36     // Parameter is passed from Query string if it is null then it default Value will be pageSize:20
37     int PageSize = pagingparametermodel.pageSize;
38
39     // Display TotalCount to Records to User
40     int TotalCount = count;
41
42     // Calculating Totalpage by Dividing (No of Records / Pagesize)
43     int TotalPage = count / PageSize;
44 }
```

```
1     totalCount = Convert.ToInt32(totalCount.value);
2 //int count = lstmembermart.Count();
3
4 // Parameter is passed from Query string if it is null then it default Value will be pageNumber:1
5 int currentPage = pagingparametermodel.pageNumber;
6
7 // Parameter is passed from Query string if it is null then it default Value will be pageSize:20
8 int pageSize = pagingparametermodel.pageSize;
9
10 // Display TotalCount to Records to User
11 int TotalCount = count;
12
13 // Calculating Totalpage by Dividing (No of Records / Pagesize)
14 int TotalPages = (int)Math.Ceiling(count / (double)pageSize);
15
16 // Returns List of Customer after applying Paging
17 var items = lstprovider;//.Skip((currentPage - 1) * pageSize).Take(pageSize).ToList();
18
19 // if currentPage is greater than 1 means it has previousPage
20 var previousPage = currentPage > 1 ? "Yes" : "No";
21
22 // if TotalPages is greater than currentPage means it has nextPage
23 var nextPage = currentPage < TotalPages ? "Yes" : "No";
24
25 // Object which we are going to send in header
26 var paginationMetadata = new
27 {
28     totalCount = TotalCount,
29     pageSize = pageSize,
30     currentPage = currentPage,
31     totalPages = TotalPages,
32     previousPage,
33     nextPage
34 };
35
```


Visual Studio

File View Project Build Debug Team Tools Test Analyze Window Help

Debug Any CPU IIS Express (Google Chrome)

MembersController.cs ProvidersController.cs* Source Control Explorer ACO_MembersMart.cs Notification_UserSpecific.cs

UQP

```
1233
1234
1235
1236     if (obj.report == true)
1237     {
1238         var rprobj = new ReportsController()
1239         {
1240             Request = new System.Net.Http.HttpRequestMessage(),
1241             Configuration = new HttpConfiguration()
1242         };
1243         return rprobj.GenerateNotifReport(null, null, lstprovider, null, null, null, "Provider List", obj.reporttype, null, null,
1244                                         null, null);
1245     }
1246     //else
1247     //{
1248     //    return Request.CreateResponse(HttpStatusCode.OK, items);
1249     //}
1250
1251     return Request.CreateResponse(HttpStatusCode.OK, "ok");
1252 }
1253 catch (Exception ex)
1254 {
1255     return Request.CreateResponse(HttpStatusCode.InternalServerError, ex.InnerException);
1256 }
1257
1258
1259
1260
1261
1262 }
```

```
1260
1261
1262
1263     1reference
1264     private bool ProviderExists(long id)
1265     {
1266         using (UQPEntities db = new UQPEntities(HttpContext.Current.Request.Headers.GetValues("IPA").FirstOrDefault()))
1267         {
1268             return db.ProviderDetailsTestings.AsNoTracking().Count(e => e.providerid == id) > 0;
1269         }
1270
1271     [HttpPost]
1272     [Route("api/providerscorecard/addnote")]
1273     public HttpResponseMessage addnote(ProviderScorecard note)
1274     {
1275         try
1276         {
1277             using (UQPEntities db = new UQPEntities(HttpContext.Current.Request.Headers.GetValues("IPA").FirstOrDefault()))
1278             {
1279                 ProviderScorecard_Note providerScorecard_Note = new ProviderScorecard_Note
1280                 {
1281                     entereddate = note.dateentered,
1282                     userid = note.userid,
1283                     firstname = note.firstname,
1284                     midname = note.midname,
1285                     lastname = note.lastname,
1286                     procedure_performed = note.procedure_performed,
1287                     meeting_with = note.discussion_with,
1288                 };
1289             }
1290         }
1291     }
1292 }
```

Microsoft Visual Studio

File View Project Build Debug Team Tools Test Analyze Window Help

UQP ProvidersController.cs* ACO_MembersMart.cs Notification_UserSpecific.cs

```
1286     lastname = note.lastname,
1287     procedure_performed = note.procedure_performed,
1288     meeting_with = note.discussion_with,
1289     notes = note.notes,
1290     followupdays = note.followupdays,
1291     followupdate = note.followupdate,
1292     responsible_party = note.assignedto,
1293     min_spent = note.minutes,
1294     providerid = note.providerid,
1295     levelofimportance = note.importance
1296   };
1297   db.ProviderScorecard_Note.Add(providerScorecard_Note);
1298   db.SaveChanges();
1299   return Request.CreateResponse(HttpStatusCode.OK, "Note Inserted Successfully");
1300 }
1301 }
1302 catch (Exception ex)
1303 {
1304   return Request.CreateResponse(HttpStatusCode.InternalServerError, ex.Message);
1305 }
1306 }
1307 
```

[HttpPost]
[Route("api/addfollowupnote")]

```
UOP.Controllers.ProvidersController -> ProviderDetails(PagingParametersModel p...  
1304     }  
1305     return Request.CreateResponse(HttpStatusCode.InternalServerError, ex.Message);  
1306   }  
1307 }  
1308 [HttpPost]  
1309 [Route("api/addfollowupnote")]  
1310 0 references  
1311 public HttpResponseMessage followupnote(Followupnote notesfollowup)  
1312 {  
1313   try  
1314   {  
1315     using (UQPEntities db = new UQPEntities(HttpContext.Current.Request.Headers.GetValues("IPA").FirstOrDefault()))  
1316     {  
1317       ProviderScorecard_FollowupNote obj = new ProviderScorecard_FollowupNote  
1318       {  
1319         scorecard_noteid = notesfollowup.noteid,  
1320         followupminutes = notesfollowup.followuptime,  
1321         date_entered = notesfollowup.dateentered,  
1322         userid = notesfollowup.userid,  
1323         firstname = notesfollowup.firstname,  
1324         lastname = notesfollowup.lastname,  
1325         midname = notesfollowup.midname,  
1326         notes = notesfollowup.followupcomment  
1327       };  
1328       db.ProviderScorecard_FollowupNote.Add(obj);  
1329       db.SaveChanges();  
1330       return Request.CreateResponse(HttpStatusCode.OK, "Followup added successfully");  
1331     }  
1332   }  
1333   catch (Exception ex)  
1334   {  
1335     return Request.CreateResponse(HttpStatusCode.InternalServerError, ex);  
1336   }  
1337 }  
1338 }
```

```
UQP | UQP.Controllers.ProvidersController | ProviderDetails(Paging) [F5]
1336 } }
1337 }
1338 }
1339 [HttpPost]
1340 [Route("api/providerscorecardlist")]
1341 public HttpResponseMessage providerscorecardlist(ProviderScorecard obj)
1342 {
1343     try
1344     {
1345         using (UQPEntities db = new UQPEntities(HttpContext.Current.Request.Headers.GetValues("IPA").FirstOrDefault()))
1346         {
1347             var query = (db.ProviderScorecard_Note.Where(e => e.providerid == obj.providerid).GroupJoin
1348                         (db.ProviderScorecard_FollowupNote.DefaultIfEmpty(), p => p.id, c => c.scorecard_noteid
1349                         , (p, c1) => new
1350                         {
1351                             p.entereddate,
1352                             p.userid,
1353                             p.firstname,
1354                             p.midname,
1355                             p.lastname,
1356                             p.procedure_performed,
1357                             p.meeting_with,
1358                             p.notes,
1359                             p.followupdays,
1360                             p.followupdate,
1361                             p.responsible_party,
1362                             p.min_spent,
1363                             p.providerid,
1364                             p.id,
1365                             p.levelofimportance,
1366                             ans = c1.ToList().OrderByDescending(e => e.id).FirstOrDefault()
1367                         }).OrderByDescending(e => e.entereddate)
1368                         }).Skip(obj.pageSize * (obj.pageNumber - 1)).Take(obj.pageSize).ToList();
1369     }
1370 }
```

```
Project Build Debug Team Tools Test Analyze Window Help
Debug Any CPU IIS Express (Google Chrome) ACO_MembersList.cs Notification_UserSpecific.cs
ProvidersController.cs* Source Control Explorer UOP.Controllers.ProvidersController.cs ProviderDetails(PagingParameterModel paging)
68     ans = c1.ToList().OrderByDescending(e => e.id).FirstOrDefault()
69     }).OrderByDescending(e => e.entereddate)
70     ).Skip(obj.pageSize * (obj.pageNumber - 1)).Take(obj.pageSize).ToList();
71
72     //var query = (from p in db.ProviderScorecard_Note
73     //join f in db.ProviderScorecard_FollowupNote.DefaultIfEmpty() on p.id equals f.scorecard_noteid
74     //where p.providerid == obj.providerid && (obj.userid == null || p.userid == obj.userid)
75
76     //
77     select new
78     {
79         //
80         p.entereddate,
81         //
82         p.userid,
83         //
84         p.firstname,
85         //
86         p.midname,
87         //
88         p.lastname,
89         //
90         p.procedure_performed,
91         //
92         p.meeting_with,
93         //
94         p.notes,
95         //
96         p.followupdays,
97         //
98         p.followupdate,
99         //
100        p.responsible_party,
101        //
102        p.min_spent,
103        //
104        p.providerid,
105        //
106        p.id,
107        //
108        p.levelofimportance,
109
110    }).OrderByDescending(e => e.entereddate)
111    ).Skip(obj.pageSize * (obj.pageNumber - 1)).Take(obj.pageSize).ToList();
112
113    return ans;
114}
```

```
UOP - UOP.Controllers.ProvidersController - ProviderDetails(PagingParameters) [1]
1392
1393
1394
1395
1396
1397
1398
1399
1400
1401
1402
1403
1404
1405
1406
1407
1408
1409
1410
1411
1412
1413
1414
1415
1416
1417
1418
1419
1420
1421
1422
1423
1424
1425
1426
1427
```

```
//          p.id,
//          p.levelofimportance,
//
//          }).OrderByDescending(e => e.entereddate)
//          .Skip(obj.pageSize * (obj.pageNumber - 1)).Take(obj.pageSize).ToList();

List<ProviderScorecard> providerscorecardlist = new List<ProviderScorecard>();
foreach (var value in query)
{
    ProviderScorecard providerscorecard = new ProviderScorecard
    {
        dateentered = value.entereddate.Value,
        userid = value.userid,
        firstname = value.firstname,
        midname = value.midname,
        lastname = value.lastname,
        procedure_performed = value.procedure_performed,
        discussion_with = value.meeting_with,
        notes = value.notes,
        followupdays = value.followupdays,
        followupdate = value.followupdate.Value,
        assignedto = value.responsible_party,
        minutes = value.min_spent.Value,
        providerid = value.providerid.Value,
        id = value.id,
        importance = value.levelofimportance,
        followupfirstname = (value.ans==null)? "": value.ans.firstname,
        followuplastname = (value.ans == null) ? "" : value.ans.lastname,
        followupmidname = (value.ans == null) ? "" : value.ans.midname,
        followupnote = (value.ans == null) ? "" : value.ans.notes,
        followupminutes = (value.ans == null) ? null : value.ans.followupminutes
    };
    providerscorecardlist.Add(providerscorecard);
}
```

File Explorer Tools

MembersController.cs ProvidersController.cs* Source Control Explorer ACO_MemberslList.cs Notification_UserSpecific.cs UQP.Controllers.ProvidersController ProviderDetails(PagingParameterModel)

UQP

```
1404     dateentered = value.entereddate.Value,
1405     userid = value.userid,
1406     firstname = value.firstname,
1407     midname = value.midname,
1408     lastname = value.lastname,
1409     procedure_performed = value.procedure_performed,
1410     discussion_with = value.meeting_with,
1411     notes = value.notes,
1412     followupdays = value.followupdays,
1413     followupdate = value.followupdate.Value,
1414     assignedto = value.responsible_party,
1415     minutes = value.min_spent.Value,
1416     providerid = value.providerid.Value,
1417     id = value.id,
1418     importance = value.levelofimportance,
1419     followupfirstname = (value.ans==null)? "" : value.ans.firstname,
1420     followuplastname = (value.ans == null) ? "" : value.ans.lastname,
1421     followupmidname = (value.ans == null) ? "" : value.ans.midname,
1422     followupnote = (value.ans == null) ? "" : value.ans.notes,
1423     followupminutes = (value.ans == null) ? null : value.ans.followupminutes
1424   };
1425   providerscorecardlist.Add(providerscorecard);
1426 }
1427 return Request.CreateResponse(HttpStatusCode.OK, providerscorecardlist);
1428 }
1429 }
1430 catch (Exception ex)
1431 {
1432   return Request.CreateResponse(HttpStatusCode.OK, ex.InnerException);
1433 }
1434 }
1435 }
1436 }
```

A screenshot of a Microsoft Visual Studio IDE interface. The title bar shows "Debug Any CPU IIS Express (Google Chrome)". The tabs at the top include "MembersController.cs", "ProvidersController.cs*", "Source Control Explorer", "ACO_MembersMart.cs", "Notification_UserSpecific.cs", and "ProviderDetails(PagingParametersModel pagingparameters)". The main window displays the "ProvidersController.cs" file. The code is heavily annotated with line numbers from 1434 to 1466. It contains two main sections: a delete method and a post method. The delete method uses Entity Framework to remove a record from the database if it exists. The post method is annotated with [HttpPost] and [Route("api/provider/overallproviderscorecard")].

```
1434     }
1435 }
1436
1437 [HttpDelete]
1438 [Route("api/providerscorecard/deletenote/{id}")]
1439 public HttpResponseMessage deletenote(long id)
1440 {
1441     try
1442     {
1443         using (UQPEntities db = new UQPEntities(HttpContext.Current.Request.Headers.GetValues("IPA").FirstOrDefault()))
1444         {
1445             var query = from p in db.ProviderScorecard_Note where p.id == id select p;
1446             if (query.Count() != 0)
1447             {
1448                 db.ProviderScorecard_Note.RemoveRange(query);
1449                 db.SaveChanges();
1450                 return Request.CreateResponse(HttpStatusCode.OK, "Note Deleted Successfully");
1451             }
1452             else
1453             {
1454                 return Request.CreateResponse(HttpStatusCode.OK, "No record found with id " + id);
1455             }
1456         }
1457     catch (Exception ex)
1458     {
1459         return Request.CreateResponse(HttpStatusCode.OK, ex.InnerException);
1460     }
1461 }
1462
1463 [HttpPost]
1464 [Route("api/provider/overallproviderscorecard")]
1465 public HttpResponseMessage overallproviderscorecard (ProviderScorecard prov)
```

```
return Request.CreateResponse(HttpStatusCode.OK, "No record found with id " + id);
}

catch (Exception ex)
{
    return Request.CreateResponse(HttpStatusCode.OK, ex.InnerException);
}

[HttpPost]
[Route("api/provider/overallproviderscorecard")]
public HttpResponseMessage overallproviderscorecard (ProviderScorecard prov)
{
    try
    {
        using (UQPEntities db = new UQPEntities(HttpContext.Current.Request.Headers.GetValues("IPA").FirstOrDefault()))
        {
            int? totalgaps = db.NewReportByProviders.Where(e => e.Provider_id == prov.providerid && e.CalculatedMonth == prov.startdate)
                .Select(e => e.gap).Sum();
            var healthplanquery = db.ProviderDetailsTestings.Where(e => e.providerid == prov.providerid).Select(e => e.healthplanid).FirstOrDefault();
            int healthplancount =(healthplanquery == null)? 0 : healthplanquery.Split(',').Count();
            var member = (from dashss in db.NewDashboardSupersets.AsNoTracking()
                         where dashss.CalculatedMonth == prov.startdate && dashss.Provider_id == prov.providerid
                         select new
                         {
                             dashss.MemberCount,
                             dashss.MemberCount,
                         }).FirstOrDefault();
            ProviderScorecard provider = new ProviderScorecard();
            provider.currentthedisgaps = (totalgaps == null)? 0 : totalgaps.Value;
            provider.totalhealthplancount = healthplancount;
            provider.totalmembercount = (member.MemberCount == null)? 0 : member.MemberCount.Value;
            return Request.CreateResponse(HttpStatusCode.OK, provider);
        }
    }
}
```

```
Debug Any CPU ProvidersController.cs* Source Control Explorer ACO_MemberSmart.cs Notification_UserSpecific.cs
UQP.Controllers.ProvidersController
int healthplancount = (healthplanquery == null) ? 0 : healthplanquery.Split(',').Count();
var member = (from dashss in db.NewDashboardSupersets.AsNoTracking()
              where dashss.CalculatedMonth == prov.startdate && dashss.Provider_id == prov.providerid
              select new
              {
                  dashss.MemberCount,
              }).FirstOrDefault();
ProviderScorecard provider = new ProviderScorecard();
provider.currentthedisgaps = (totalgaps == null) ? 0 : totalgaps.Value;
provider.totalhealthplancount = healthplancount;
provider.totalmembercount = (member.MemberCount == null) ? 0 : member.MemberCount.Value;

return Request.CreateResponse(HttpStatusCode.OK, provider);
}

catch (Exception ex)
{
    return Request.CreateResponse(HttpStatusCode.InternalServerError, ex);
}
}
}
```