

Process: [1872] iisexpress.exe    Lifecycle Events - Thread:    Stack Frame:

User\_DetailsController.cs

UQP

UQP.Controllers.User\_DetailsController

```
1  using System;
2  using System.Collections.Generic;
3  using System.Data;
4  using System.Linq;
5  using System.Net;
6  using System.Net.Http;
7  using System.Net.Mail;
8  using System.Security.Cryptography;
9  using System.Text;
10 using System.Threading.Tasks;
11 using System.Web.Http;
12 using System.Web.Http.Description;
13 using UQP.Models;
14 using UQP.Utility;
15 using System.Security.Claims;
16 using System.Net.Mime;
17 using System.Configuration;
18 using System.Net.Configuration;
19 using System.Data.Entity.Core.Objects;
20 using System.Web;
21 using System.Net.Http.Headers;
22 using System.IO;
23
24 namespace UQP.Controllers
25 {
26
27     [Authorize]
28     public class User_DetailsController : ApiController
```

100 %

Error List

Entire Solution    0 Errors    2 Warnings    0 of 108 Messages    Build + IntelliSense

Code Description

User\_DetailsController.cs

UQP

```
0 references
public class User_DetailsController : ApiController
{
    //static string dbname = HttpContext.Current.Request.Headers.GetValues("IPA").FirstOrDefault() ==
    //    "UQPUltimate"
    //    ?
    //    "UQPUltimate" : "UQPEntities";

    // For UQP
    UQPEntities db =
        new UQPEntities(
            );
    //UQPEntities db = new UQPEntities("UQPUltimate");
    //For ultimate
    //QPEntities db = new UQPEntities("UQPUltimate");

    0 references
    public object UserManager { get; private set; }

    0 references
}
```

100 %

Error List

Entire Solution 0 Errors 2 Warnings 0 of 102 Messages

```
UQP.Controller.cs □ □ X Stack Frame
UQP.Controllers.User_DetailsController
53
54
55
56
57 [HttpGet]
58 [Route("api/IPASession/{ipaid}")]
59 References:
60 public HttpResponseMessage IPASession(int ipaid)
61 {
62     var connname = (ipaid == 2) ? "UQPEntitiesAccess" : ipaid == 5 ?
63         "UQPEntitiesAllAmerican" : ipaid == 6 ? "UQPEntitiesPrimecare" :
64         ipaid == 0 ? "UQPUltimate" : ipaid == 10 ? "UQPEntitiesMirra" : string.Empty;
65     HttpContext.Current.Response.Headers.Add("IPA", connname);
66     HttpContext.Current.Response.Headers.Add("Access-Control-Expose-Headers", "IPA");
67     DateTime? lastgaprefresh = null;
68
69     using (UQPEntities db = new UQPEntities
70         (connname))
71     {
72         lastgaprefresh = db.NewDashboardSupersets.Where(x => (x.Executive_id == 1))
73             .Select(x => x.LastGapRefreshedDate).FirstOrDefault();
74     }
75
76     return Request.CreateResponse(HttpStatusCode.OK, lastgaprefresh);
77 }
78
79
80 [HttpGet]
```





[1872] iisexpress.exe Lifecycle Events - Thread: Stack Frame:

DetailsController.cs

QP

UOP.Controllers.User\_DetailsController

db

```
127
128     foreach (var per in query)
129     {
130         string ipanameforexcel = null;
131         string healthplanforexcel = null;
132         User_Type objtyp = new User_Type();
133         User_Role objrl = new User_Role();
134         ExtendedUser_Details obj = new ExtendedUser_Details();
135         List<CustomHealthPlan> healthplanlist = new List<CustomHealthPlan>();
136         List<CustomIPA> ipalist = new List<CustomIPA>();
137         if (per.User_IPA_ID != null)
138         {
139             var stripuids = per.User_IPA_ID.Split(',');
140             int[] ipaid = stripuids.Select(int.Parse).ToArray();
141             foreach (var ip in ipaid)
142             {
143                 CustomIPA ipaobj = new CustomIPA();
144                 var ipainfo = (from ips in db.IPAs.AsNoTracking()
145                               where ips.IPA_ID == ip
146                               select new { ips.IPA_ID, ips.IPA_Name, ips.IPA_Code }).FirstOrDefault();
147                 ipaobj.IPA_ID = ipainfo.IPA_ID;
148                 ipaobj.IPA_Name = ipainfo.IPA_Name;
149                 ipalist.Add(ipaobj);
150
151                 ipanameforexcel = ipanameforexcel + ipainfo.IPA_Name + ",";
152             }
153         }
154     }
```

Solution: 0 Errors | 2 Warnings | 0 of 108 Messages | Build + IntelliSense

UQP

DetailsController.cs

UQP.Controllers.User\_DetailsController

```
151     ipanameforexcel = ipanameforexcel + ipainfo.IPA_Name + ",";
152
153     }
154 }
155
156 obj.ipa = ipalist;
157 if (per.User_HealthPlan_ID != null)
158 {
159
160     CustomHealthPlan hpobj = new CustomHealthPlan();
161     var hpinfo = (from hp in db.HealthPlans.AsNoTracking()
162                   where hp.HealthPlan_ID == per.User_HealthPlan_ID
163                   select new
164                 {
165                     hp.HealthPlan_ID,
166                     hp.HealthPlan_Name
167                 }).FirstOrDefault();
168     hpobj.healthplan_id = hpinfo.HealthPlan_ID;
169     hpobj.health_Name = hpinfo.HealthPlan_Name;
170     healthplanlist.Add(hpobj);
171     healthplanforexcel = healthplanforexcel + hpinfo.HealthPlan_Name + ",";
172
173
174 }
175
176 obj.healthplan = healthplanlist;
177 obj.DefaultIPA = per.DefaultIPA;
178 obj.HealthplanNameExcel = healthplanforexcel;
179 obj.TDANameExcel = ipanameforexcel;
```

0 Errors | 2 Warnings | 0 of 102 Messages | Build + Intelligence

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

Process: [1872] iisexpress.exe Lifecycle Events Thread: Stack Frame:

User\_DetailsController.cs □ X UQP

UQP.Controllers.User\_DetailsController

178  
179  
180  
181  
182  
183  
184  
185  
186  
187  
188  
189  
190  
191  
192  
193  
194  
195  
196  
197  
198  
199  
200  
201  
202  
203  
204  
205

```
obj.HealthplanNameExcel = healthplanforexcel;
obj.IPANameExcel = ipanameforexcel;
obj.User_ID = per.User_ID;
obj.User_FirstName = per.User_FirstName.ToLower();
obj.User_MiddleName = (per.User_MiddleName == null) ? null : per.User_MiddleName.ToLower();
obj.User_LastName = per.User_LastName.ToLower();
obj.User_Provider_ID = per.User_Provider_ID;
obj.User_IPA_ID = per.User_IPA_ID;
obj.User_HealthPlan_ID = per.User_HealthPlan_ID;
obj.User_Member_ID = per.User_Member_ID;
obj.User_Email = per.User_Email;
obj.User_Address = per.User_Address.ToLower();
obj.User_DOB = per.User_DOB;
obj.User_Gender = per.User_Gender.ToLower();
obj.User_MobileNumber = per.User_MobileNumber;
obj.User_WorkNumber = per.User_WorkNumber;
//obj.User_Password = per.User_Password;
obj.FirstTimeUser = per.FirstTimeUser;
objtyp.User_Type_Name = per.User_Type_Name.ToLower();
objrl.User_Role_Name = per.User_Role_Name.ToLower();
obj.User_Type = objtyp;
obj.User_Role = objrl;

obj.User_Role.User_Role_Name = per.User_Role_Name.ToLower();
obj.Status = per.Status;
obj.User_Role_ID = per.User_Role_ID;
obj.User_Type_ID = per.User_Type_ID;
obj.title = per.title.ToLower();
userList.Add(obj);
```

[18/2] IISExpress.exe Lifecycle Events - Thread: Stack Frames

User\_DetailsController.cs # □ X

UQP

UQP.Controllers.User\_DetailsController

db

```
208
209
210 } if (user.report == false)
211 {
212     #region Pagination
213     int count = userlist.Count();
214
215     // Parameter is passed from Query string if it is null then it default Value will be pageNumber:1
216     int currentPage = pagingParameterModel.pageNumber;
217
218     // Parameter is passed from Query string if it is null then it default Value will be pageSize:20
219     int pageSize = pagingParameterModel.pageSize;
220
221     // Display TotalCount to Records to User
222     int totalCount = count;
223
224     // Calculating Totalpage by Dividing (No of Records / Pagesize)
225     int totalPages = (int)Math.Ceiling(count / (double)pageSize);
226
227     // Returns List of Customer after applying Paging
228     var items = userlist.Skip((currentPage - 1) * pageSize).Take(pageSize).ToList();
229
230     // if currentPage is greater than 1 means it has previousPage
231     var previousPage = currentPage > 1 ? "Yes" : "No";
232
233     // if TotalPages is greater than currentPage means it has nextPage
234     var nextPage = currentPage < TotalPages ? "Yes" : "No";
235
236     // Object which we are going to send in header
```

0 Errors | 2 Warnings | 0 of 108 Messages | Build + IntelliSense

[1872] IISExpress.exe Lifecycle Events - Thread: Stack Frame:

UQP.Controllers.User\_DetailsController

```
// if TotalPages is greater than CurrentPage means it has nextPage
var nextPage = CurrentPage < TotalPages ? "Yes" : "No";

// Object which we are going to send in header
var paginationMetadata = new
{
    totalCount = TotalCount,
    pageSize = PageSize,
    currentPage = currentPage,
    totalPages = TotalPages,
    previousPage,
    nextPage
};

// Setting Header

System.Web.HttpContext.Current.Response.Headers.Add("Paging-Headers", Newtonsoft.Json.JsonConvert.SerializeObject(paginationMetadata));
//HttpContext.Current.Response.Headers.Add("Filename", zipFileName);
HttpContext.Current.Response.Headers.Add("Access-Control-Expose-Headers", "Paging-Headers");
#endregion
return Request.CreateResponse(HttpStatusCode.OK, items);
}
// Returing List of Customers Collections
else if (user.report == true)
{
    var rpoj = new ReportsController
    {
        Request = new System.Net.Http.HttpRequestMessage()
    }
    return rpoj.GetReport();
}

0 Errors | 2 Warnings | 0 of 108 Messages | Build + IntelliSense | Project
```



Source Control Explorer    User\_DetailsController.cs\* + X

UQP

```
274 }  
275  
276     1reference  
277     public ExtendedUser_Details GetUser_Details(int id)  
278     {  
279  
280         ExtendedUser_Details obj = new ExtendedUser_Details();  
281         var query = (from ab in db.User_Details.AsNoTracking()  
282             join me in db.User_Role.AsNoTracking()  
283                 on ab.User_Role_ID equals me.User_Role_ID  
284             join ty in db.User_Type.AsNoTracking() on ab.User_Type_ID equals ty.User_Type_ID  
285             where ab.User_ID == id  
286  
287             select new  
288             {  
289                 ab.User_ID,  
290                 ab.User_Email,  
291                 ab.User_MiddleName,  
292                 ab.User_LastName,  
293                 ab.User_FirstName,  
294                 ab.User_Address,  
295                 ab.User_DOB,  
296                 ab.User_Gender,  
297                 ab.User_MobileNumber,  
298                 ab.User_WorkNumber,  
299                 //ab.User_Password,  
300                 ab.FirstTimeUser,  
301                 ab.Status,  
302                 ab.User_Role_ID,  
me.User_Role_Name,
```

Output

Show output from: Debug

```
294     ab.User_DOB,
295     ab.User_Gender,
296     ab.User_MobileNumber,
297     ab.User_WorkNumber,
298     //ab.User_Password,
299     ab.FirstTimeUser,
300     ab.Status,
301     ab.User_Role_ID,
302     me.User_Role_Name,
303     ty.User_Type_Name,
304     ab.User_Type_ID,
305     ab.User_IPA_ID,
306     ab.User_HealthPlan_ID,
307     ab.User_Member_ID,
308     ab.User_Provider_ID,
309     ab.title,
310     ab.OrganisationType,
311     ab.MRATEam,
312     ab.HedisTeam,
313     ab.HedisTeamId,
314     ab.MRATEamId,
315     ab.DefaultIPA
316
317     }).ToList();
318
319
320     var assignadd = (from ofc in db.OfficeStaff_AssignedProvider.AsNoTracking()
321                     join pv in db.ProviderDetailsTestings.AsNoTracking() on ofc.User_ProviderId equals pv.providerid
322                     where ofc.User_Id == id && pv.visible == true
323                     select new
```

Microsoft Visual Studio

View Project Build Debug Team Tools Test Analyze Window Help

User\_DetailsController.cs

```
UQP.Controllers.User_DetailsController
 GetUser_Details(UserDetailsFilter user, PagingParameterModel pag
}).ToList();

var assignadd = (from ofc in db.OfficeStaff_AssignedProvider.AsNoTracking()
join pv in db.ProviderDetailsTestings.AsNoTracking() on ofc.User_ProviderId equals pv.providerid
where ofc.User_Id == id && pv.visible == true
select new
{
    pv.Physician_first_name,
    pv.Physician_last_name,
    pv.emailid,
    pv.providerid
}).ToList();

List<CustomProviderDetails> lstobj = new List<CustomProviderDetails>();
List<CustomIPA> ipalist = new List<CustomIPA>();
List<providerTesting_address> addresslist = new List<providerTesting_address>();

foreach (var per in query)
{
    User_Type objtyp = new User_Type();
    User_Role objrl = new User_Role();
    Organisation orgobj = new Organisation();
    // IPA ipaobj = new IPA();
    if (per.OrganisationType != null)
    {
        var org = per.OrganisationType.Split(',');
    }
}
```

Project Build Debug Team Tools Test Analyze Window Help

Source Control Explorer User\_DetailsController.cs\* □ X

UQP

```
345 if (per.OrganisationType != null)
346 {
347     var org = per.OrganisationType.Split(',');
348
349     foreach (var x in org)
350     {
351         if (x == "1")
352         {
353             orgobj.hedis = true;
354
355         }
356         if (x == "2")
357         {
358             orgobj.mra = true;
359         }
360         if (x == "3")
361         {
362             orgobj.aco = true;
363         }
364
365     }
366
367     obj.MRATeam = per.MRATeam;
368     obj.HedisTeam = per.HedisTeam;
369     obj.HedisTeamId = per.HedisTeamId;
370     obj.MRATeamId = per.MRATeamId;
371     obj.DefaultIPA = per.DefaultIPA;
372
373
374     obj.User_ID = per.User_ID;
```

User Control Explorer      User\_DetailsController.cs\*      IIS Express (Google Chrome)      Debug Any CPU

```
UQP.Controllers.User_DetailsController
 GetUser_Details(UserDetailsFilter user, PagingParameters parameters)
{
    var obj = new UserDetails();
    obj.User_ID = per.User_ID;
    obj.User_FirstName = per.User_FirstName.ToLower();
    obj.User_MiddleName = (per.User_MiddleName == null) ? null : per.User_MiddleName.ToLower();
    obj.User_LastName = per.User_LastName.ToLower();
    obj.User_Provider_ID = per.User_Provider_ID;
    obj.User_IPA_ID = per.User_IPA_ID;
    if (obj.User_IPA_ID != null)
    {
        var stripaids = obj.User_IPA_ID.Split(',');
        int[] ipaid = stripaids.Select(int.Parse).ToArray();
        foreach (var ip in ipaid)
        {
            CustomIPA ipaobj = new CustomIPA();
            var ipainfo = (from ips in db.IPAs.AsNoTracking()
                           where ips.IPA_ID == ip
                           select new { ips.IPA_ID, ips.IPA_Name, ips.IPA_Code }).FirstOrDefault();
            ipaobj.IPA_ID = ipainfo.IPA_ID;
            ipaobj.IPA_Name = ipainfo.IPA_Name;
            ipalist.Add(ipaobj);
        }
    }
}
```

A screenshot of the Microsoft Visual Studio IDE interface. The menu bar includes 'Edit', 'View', 'Project', 'Build', 'Debug' (set to 'Any CPU'), 'Team', 'Tools', 'Test', 'Analyze', 'Window', and 'Help'. The toolbar has icons for file operations like Open, Save, and Print. The title bar shows 'User\_DetailsController.cs\*' and 'UQP.Controllers.User\_DetailsController'. The code editor displays C# code for a controller action named ' GetUser\_Details'. The code uses reflection to map properties from a 'per' object to a 'obj' object, including 'User\_HealthPlan\_ID', 'User\_Member\_ID', 'User\_Email', 'User\_Address', 'User\_DOB', 'User\_Gender', 'User\_MobileNumber', 'User\_WorkNumber', 'FirstTimeUser', 'User\_Type\_Name', 'User\_Role\_Name', 'User\_Type', 'User\_Role', 'User\_Role\_Name', 'Status', 'User\_Role\_ID', 'User\_Type\_ID', and 'title'. It also loops through a collection 'assignadd' to create 'CustomProviderDetails' objects for each item. The code editor shows lines 402 to 431. The status bar at the bottom left indicates '100 %'.

```
402     obj.User_HealthPlan_ID = per.User_HealthPlan_ID;
403     obj.User_Member_ID = per.User_Member_ID;
404     obj.User_Email = per.User_Email;
405     obj.User_Address = per.User_Address.ToLower();
406     obj.User_DOB = per.User_DOB;
407     obj.User_Gender = per.User_Gender.ToLower();
408     obj.User_MobileNumber = per.User_MobileNumber;
409     obj.User_WorkNumber = per.User_WorkNumber;
410
411     obj.FirstTimeUser = per.FirstTimeUser;
412     objtyp.User_Type_Name = per.User_Type_Name.ToLower();
413     objrsl.User_Role_Name = per.User_Role_Name.ToLower();
414     obj.User_Type = objtyp;
415     obj.User_Role = objrsl;
416
417     obj.User_Role.User_Role_Name = per.User_Role_Name.ToLower();
418     obj.Status = per.Status;
419     obj.User_Role_ID = per.User_Role_ID;
420     obj.User_Type_ID = per.User_Type_ID;
421     obj.title = per.title;
422
423     foreach (var a in assignadd)
424     {
425         CustomProviderDetails objprovdet = new CustomProviderDetails();
426         objprovdet.FirstName = a.Physician_first_name.ToLower();
427         objprovdet.MidName = null;
428         objprovdet.LastName = a.Physician_last_name.ToLower();
429         objprovdet.email = a.emailid;
430         objprovdet.id = a.providerid;
431     }

```

UOP

UQP.Controllers.User\_DetailsController

GetUser\_Details(UserDetailsFilter user, PagingOptions options)

```
429     objprovdet.email = a.emailid;
430     objprovdet.id = a.providerid;
431
432     lstobj.Add(objprovdet);
433 }
434
435 if (per.User_Role_ID == 35)
436 {
437     var subquery = (from pf in db.providertesting_address
438                     where pf.providerid == per.User_Provider_ID
439
440                     select new
441                     {
442                         pf.address,
443                         pf.backline_phone_number,
444                         pf.fax,
445                         pf.npi,
446                         pf.phone,
447                         pf.providerid,
448                         pf.recordid,
449                         pf.isdefault
450                     }).Distinct().ToList();
451
452     foreach (var value in subquery)
453     {
454         providertesting_address address = new providertesting_address
455         {
456             address = value.address,
457             backline_phone_number = value.backline_phone_number,
458             fax = value.fax,
```

Output from: Debug

isexpress.exe' (CLR v4.0.30319: /LM/W3SVC/2/ROOT-1-13230222775672406): Loaded 'C:\Windows\Microsoft.NET\Framework\v4.0.30319\Temporary ASP.NET Files\root\12230222775672406\assembly\GAC\_MSIL\System.Windows.Forms\v4.0.4.0\_4.0.0.0\_31bf3856ad364f37\System.Windows.Forms.dll'

Source Control Explorer User\_DetailsController.cs\* X UQP UQP.Controllers.User\_DetailsController GetUser\_Details(UserDetails)

```
456     address = value.address,
457     backline_phone_number = value.backline_phone_number,
458     fax = value.fax,
459     npi = value.npi,
460     phone = value.phone,
461     providerid = value.providerid,
462     isdefault = value.isdefault,
463     recordid = value.recordid
464   };
465   addresslist.Add(address);
466 }
467 obj.addresslist = addresslist;
468 }
469
470 obj.AssignedProviders = lstobj;
471 obj.ipadetails = ipalist;
472 obj.organisations = orgobj;
473
474 }
475
476 return obj;
477 }
478
479
480
481 [HttpGet]
482 [Route("api/UserDetails/{userid}")]
483 public dynamic UserDetails(int userid)
484 {
```

100 % 4 Output

Source Control Explorer    User\_DetailsController.cs\*    X

UQP

User\_DetailsController

GetUser\_Details(UserDetailsFilter user, PageModel<UserDetails> model)

```
480
481     [HttpGet]
482     [Route("api/UserDetails/{userid}")]
483     public dynamic UserDetails(int userid)
484     {
485
486         List<CustomProviderDetails> lstobjcstmprovdetail = new List<CustomProviderDetails>();
487         ExtendedUser_Details objusrdetail = new ExtendedUser_Details();
488
489         var g = db.User_Details.AsNoTracking().FirstOrDefault(e => e.User_ID == userid);
490
491         var pid = g.User_Provider_ID;
492         if (g.User_Type_ID == 3 || g.User_Provider_ID == null || g.User_Role_ID == 1065)
493         {
494
495             objusrdetail = GetUser_Details(Convert.ToInt32(userid));
496         }
497         else
498         {
499             var query = (from st in db.User_Details.AsNoTracking()
500                         join ft in db.providertesting_address.AsNoTracking() on st.User_Provider_ID equals ft.providerid
501                         join ty in db.User_Type.AsNoTracking() on st.User_Type_ID equals ty.User_Type_ID
502                         join ur in db.User_Role.AsNoTracking() on st.User_Role_ID equals ur.User_Role_ID
503                         where st.User_Provider_ID == pid && st.User_ID == userid
504                         && ft.isdefault == true
505                         select new
506                         {
507                             st.User_Email,
508                             st.User_MiddleName,
```

100 %

Output

Show output from: Debug

```
User_DetailsController.cs*  ▾ X  UQP.Controllers.User_DetailsController  ▾ GetUser_Details(UserDetailsFilter user, PagingOptions options) {  
    var providerlist = db.ProviderDetailsTestings.FirstOrDefault(e => e.providerid == pid);  
  
    string sdate = "01/" + DateTime.Now.AddMonths(-1).ToString("MM") + "/" + DateTime.Now.Year.ToString();  
    DateTime startdate = DateTime.ParseExact(sdate, "dd/MM/yyyy", null);  
  
    var hp = (from hps in db.insurancebyprovider_membercount.AsNoTracking()  
              where hps.provider_Id == pid & hps.timestamp == startdate & hps.totalMembership > 0  
              select new {
```

Source Control Explorer    User\_DetailsController.cs\*  X

UQP

```
527 string sdate = "01/" + DateTime.Now.AddMonths(-1).ToString("MM") + "/" + DateTime.Now.Year.ToString();
528 DateTime startdate = DateTime.ParseExact(sdate, "dd/MM/yyyy", null);
529
530
531 var hp = (from hps in db.insurancebyprovider_membercount.AsNoTracking()
532           where hps.provider_Id == pid & hps.timestamp == startdate & hps.totalMembership > 0
533           select new
534           {
535
536               hps.HealthPlan_Name
537
538           }).ToList();
539
540
541 var hpsubquery = hp.Distinct();
542
543
544 List<CustomInsurance> lstci = new List<CustomInsurance>();
545 List<CustomIPA> lstIPA = new List<CustomIPA>();
546 List<ProvidersGroups> lstgroup = new List<ProvidersGroups>();
547 CustomProviderDetails objdet = new CustomProviderDetails();
548 List<CustomIPA> ipalist = new List<CustomIPA>();
549 foreach (var per in query)
550 {
551     User_Type objtyp = new User_Type();
552     User_Role objrle = new User_Role();
553     objusrdetail.User_FirstName = per.User_FirstName.ToLower();
554     //objusrdetail.User_MiddleName = per.User_MiddleName.ToLower();
555     objusrdetail.User_LastName = per.User_LastName.ToLower();
556     objusrdetail.User_Email = per.User_Email;
```

Output

Show output from: Debug

The screenshot shows a Microsoft Visual Studio interface with the code editor open. The title bar says "User\_DetailsController.cs\*". The code itself is a C# file containing logic to map user details from a parameter object to a database entity and then associate it with a list of IPAs.

```
//objusrdetail.User_MiddleName = per.User_MiddleName.ToLower();
objusrdetail.User_LastName = per.User_LastName.ToLower();
objusrdetail.User_Email = per.User_Email;
objusrdetail.PrimaryFacilityId = per.PrimaryFacilityId;
objusrdetail.User_DOB = per.User_DOB;
objusrdetail.User_Gender = per.User_Gender.ToLower();
objusrdetail.User_Address = per.address;

objtyp.User_Type_Name = per.User_Type_Name.ToLower();
objr1.User_Role_Name = per.User_Role_Name.ToLower();
objusrdetail.User_Type = objtyp;
objusrdetail.User_Role = objr1;
objusrdetail.User_IPA_ID = per.User_IPA_ID;

var stripaids = objusrdetail.User_IPA_ID.Split(',');
int[] ipaid = stripaids.Select(int.Parse).ToArray();
foreach (var ip in ipaid)
{
    CustomIPA ipaobj = new CustomIPA();
    var ipainfo = (from ips in db.IPAs.AsNoTracking()
                  where ips.IPA_ID == ip
                  select new { ips.IPA_ID, ips.IPA_Name, ips.IPA_Code }).FirstOrDefault();
    ipaobj.IPA_ID = ipainfo.IPA_ID;
    ipaobj.IPA_Name = ipainfo.IPA_Name;
    ipalist.Add(ipaobj);
}
```

```
UQP
User_DetailsController.cs*  ↗ X  UQP.Controllers.User_DetailsController  GetUser_Details(UserDetails)
581
582
583
584     foreach (var b in hpsubquery)
585     {
586         CustomInsurance cis = new CustomInsurance();
587         cis.Insurance_Name = b.HealthPlan_Name.ToLower();
588         // cis.Insurance_Code = b.HealthPlan_Code;
589         lstci.Add(cis);
590     }
591
592
593     objusrdetail.ipa = ipalist;
594     objusrdetail.insurance = lstci;
595     //objusrdetail.healthplan = lsthp;
596     objusrdetail.group = lstgroup;
597     objusrdetail.AssignedProviders = null;
598     objusrdetail.ipadetails = ipalist;
599
600 }
601
602
603     return objusrdetail;
604 }
605
606 [Route("api/User_Details/GetUserlist/{firstname}")]
607 public List<User_Details> GetUserlist(string firstname)
```

```
[Route("api/User_Details/ GetUserlist/{firstname}")]
public List<User_Details> GetUserlist(string firstname)
{
    //using (var db = new UQPEntities("UQPEntities"))
    //{
        var results = (from c in db.User_Details.AsNoTracking()
                      join me in db.User_Role.AsNoTracking()
                      on c.User_Role_ID equals me.User_Role_ID
                      join ty in db.User_Type.AsNoTracking() on c.User_Type_ID equals ty.User_Type_ID
                      where (c.User_FirstName.StartsWith(firstname) || c.User_MiddleName.StartsWith(firstname) || c.User_LastName.StartsWith(firstname))
                            select new
                            {
                                c.User_ID,
                                c.User_Email,
                                c.User_MiddleName,
                                c.User_LastName,
                                c.User_FirstName,
                                c.User_Address,
                                c.User_DOB,
                                c.User_Gender,
                                c.User_MobileNumber,
                                c.User_WorkNumber,
                                //c.User_Password,
                                c.FirstTimeUser,
                                c.Status,
                                c.User_Role_ID,
                                " "
                            });
    //}
}
```

```
User_Control Explorer User_DetailsController.cs* ▾ X UQP.UQP.Controllers.User_DetailsController GetUser_Details(UserDetailsFilter) 631 c.Status, 632 633 c.User_Role_ID, 634 me.User_Role_Name, 635 ty.User_Type_Name, 636 c.User_Type_ID, 637 c.User_IPA_ID, 638 c.User_HealthPlan_ID, 639 c.User_Member_ID, 640 c.User_Provider_ID, 641 c.title 642 643 }).ToList(); 644 List<User_Details> userlist = new List<User_Details>(); 645 646 647 648 foreach (var per in results) 649 { 650 651 652 653 User_Type objtyp = new User_Type(); 654 User_Role objr1 = new User_Role(); 655 User_Details obj = new User_Details(); 656 obj.User_ID = per.User_ID; 657 obj.User_FirstName = per.User_FirstName.ToLower(); 658 obj.User_MiddleName = per.User_MiddleName.ToLower(); 659 obj.User_LastName = per.User_LastName.ToLower(); 660 obj.User_Provider_ID = per.User_Provider_ID;
```

Source Control Explorer    User\_DetailsController.cs\*    X    UQP.Controllers.User\_DetailsController    GetUser\_Details(UserDetailsFilter)

```
UQP
652
653     User_Type objtyp = new User_Type();
654     User_Role objrl = new User_Role();
655     User_Details obj = new User_Details();
656     obj.User_ID = per.User_ID;
657     obj.User_FirstName = per.User_FirstName.ToLower();
658     obj.User_MiddleName = per.User_MiddleName.ToLower();
659     obj.User_LastName = per.User_LastName.ToLower();
660     obj.User_Provider_ID = per.User_Provider_ID;
661     obj.User_IPA_ID = per.User_IPA_ID;
662     obj.User_HealthPlan_ID = per.User_HealthPlan_ID;
663     obj.User_Member_ID = per.User_Member_ID;
664     obj.User_Email = per.User_Email;
665     obj.User_Address = per.User_Address.ToLower();
666     obj.User_DOB = per.User_DOB;
667     obj.User_Gender = per.User_Gender.ToLower();
668     obj.User_MobileNumber = per.User_MobileNumber;
669     obj.User_WorkNumber = per.User_WorkNumber;
670     //obj.User_Password = per.User_Password;
671     obj.FirstTimeUser = per.FirstTimeUser;
672     objtyp.User_Type_Name = per.User_Type_Name.ToLower();
673     objrl.User_Role_Name = per.User_Role_Name.ToLower();
674     obj.User_Type = objtyp;
675     obj.User_Role = objrl;
676
677     obj.User_Role.User_Role_Name = per.User_Role_Name.ToLower();
678     obj.Status = per.Status;
679     obj.User_Role_ID = per.User_Role_ID;
680     obj.User_Type_ID = per.User_Type_ID;
681     obj.title = per.title.ToLower();
```

100 %

Output

Source Control Explorer      User\_DetailsController.cs\*    X      UOP      UOP.Controllers.User\_DetailsController      GetUser\_Details(UserDetailsFilter)

```
676      obj.User_Role.User_Role_Name = per.User_Role_Name.ToLower();
677      obj.Status = per.Status;
678      obj.User_Role_ID = per.User_Role_ID;
679      obj.User_Type_ID = per.User_Type_ID;
680      obj.title = per.title.ToLower();
681      userlist.Add(obj);
682 
683 
684 
685 }
686 
687 return userlist;
688 // }
689 
690 }
691 
692 
693 
694 [HttpPost]
695 [Route("api/User_Details/PutUserDetailStatus/{id}")]
696 public HttpResponseMessage PutUserDetailStatus(int id, ModifyStatus user_status)
697 {
698     if (!ModelState.IsValid)
699     {
700         return Request.CreateResponse(HttpStatusCode.BadRequest, "Inputs are invalid ");
701     }
702 
703     try
704     {
705         var address = db.User_Details.FirstOrDefault(u => u.User_ID == id); //Checking whether the passed user id is present in database
706         if (address != null)
707         {
708             address.Modify_Status = user_status;
709             db.SaveChanges();
710         }
711     }
712     catch (Exception ex)
713     {
714         return Request.CreateResponse(HttpStatusCode.InternalServerError, "An error occurred while updating the record");
715     }
716 }
```

```
91
92
93
94
95
96
97
98
99
100
101
102
103
104
105
106
107
108
109
110
111
112
113
114
115
116
117
118
119
120
121
122
123
124
125
126
127
128
129
130
131
132
133
134
135
136
137
138
139
140
141
142
143
144
145
146
147
148
149
150
151
152
153
154
155
156
157
158
159
160
161
162
163
164
165
166
167
168
169
170
171
172
173
174
175
176
177
178
179
180
181
182
183
184
185
186
187
188
189
190
191
192
193
194
195
196
197
198
199
200
201
202
203
204
205
206
207
208
209
210
211
212
213
214
215
216
217
218
219
220
221
222
223
224
225
226
227
228
229
230
231
232
233
234
235
236
237
238
239
240
241
242
243
244
245
246
247
248
249
250
251
252
253
254
255
256
257
258
259
260
261
262
263
264
265
266
267
268
269
270
271
272
273
274
275
276
277
278
279
280
281
282
283
284
285
286
287
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
322
323
324
325
326
327
328
329
330
331
332
333
334
335
336
337
338
339
340
341
342
343
344
345
346
347
348
349
350
351
352
353
354
355
356
357
358
359
360
361
362
363
364
365
366
367
368
369
370
371
372
373
374
375
376
377
378
379
380
381
382
383
384
385
386
387
388
389
390
391
392
393
394
395
396
397
398
399
400
401
402
403
404
405
406
407
408
409
410
411
412
413
414
415
416
417
418
419
420
421
422
423
424
425
426
427
428
429
430
431
432
433
434
435
436
437
438
439
440
441
442
443
444
445
446
447
448
449
450
451
452
453
454
455
456
457
458
459
459
460
461
462
463
464
465
466
467
468
469
470
471
472
473
474
475
476
477
478
479
479
480
481
482
483
484
485
486
487
488
489
489
490
491
492
493
494
495
496
497
498
499
500
501
502
503
504
505
506
507
508
509
509
510
511
512
513
514
515
516
517
518
519
519
520
521
522
523
524
525
526
527
528
529
529
530
531
532
533
534
535
536
537
538
539
539
540
541
542
543
544
545
546
547
548
549
549
550
551
552
553
554
555
556
557
558
559
559
560
561
562
563
564
565
566
567
568
569
569
570
571
572
573
574
575
576
577
578
579
579
580
581
582
583
584
585
586
587
588
589
589
590
591
592
593
594
595
596
597
598
599
599
600
601
602
603
604
605
606
607
608
609
609
610
611
612
613
614
615
616
617
618
619
619
620
621
622
623
624
625
626
627
628
629
629
630
631
632
633
634
635
636
637
638
639
639
640
641
642
643
644
645
646
647
648
649
649
650
651
652
653
654
655
656
657
658
659
659
660
661
662
663
664
665
666
667
668
669
669
670
671
672
673
674
675
676
677
678
679
679
680
681
682
683
684
685
686
687
688
689
689
690
691
692
693
694
695
696
697
698
699
699
700
701
702
703
704
705
706
707
708
709
709
710
711
712
713
714
715
716
717
718
719
719
720
721
722
723
724
725
726
727
728
729
729
730
731
732
733
734
735
736
737
738
739
739
740
741
742
743
744
745
746
747
748
749
749
750
751
752
753
754
755
756
757
758
759
759
760
761
762
763
764
765
766
767
768
769
769
770
771
772
773
774
775
776
777
778
779
779
780
781
782
783
784
785
786
787
788
789
789
790
791
792
793
794
795
796
797
797
798
799
799
800
801
802
803
804
805
806
807
808
809
809
810
811
812
813
814
815
816
817
818
819
819
820
821
822
823
824
825
826
827
828
829
829
830
831
832
833
834
835
836
837
838
839
839
840
841
842
843
844
845
846
847
848
849
849
850
851
852
853
854
855
856
857
858
859
859
860
861
862
863
864
865
866
867
868
869
869
870
871
872
873
874
875
876
877
878
879
879
880
881
882
883
884
885
886
887
888
889
889
890
891
892
893
894
895
896
897
898
899
899
900
901
902
903
904
905
906
907
908
909
909
910
911
912
913
914
915
916
917
917
918
919
919
920
921
922
923
924
925
926
927
927
928
929
929
930
931
932
933
934
935
936
937
938
939
939
940
941
942
943
944
945
946
947
947
948
949
949
950
951
952
953
954
955
956
957
958
959
959
960
961
962
963
964
965
966
967
968
969
969
970
971
972
973
974
975
976
977
978
979
979
980
981
982
983
984
985
986
987
987
988
989
989
990
991
992
993
994
995
996
997
998
999
```

```
    });
    if (entity.Count() != 0)
    {
        foreach (var value in entity)
        {
            if (value.Status == true)
            {
                return Request.CreateResponse(HttpStatusCode.BadRequest, "Can't change status as one of the account associated with " +
                    query.User_Email + " is already active");
            }
        }
        query.Status = user_status.Status;
        db.SaveChanges();
        return Request.CreateResponse(HttpStatusCode.OK, "User status has been updated successfully");
    }
    else
    {
        query.Status = user_status.Status;
        db.SaveChanges();
        return Request.CreateResponse(HttpStatusCode.OK, "User status has been updated successfully");
    }
}
else
{
    return Request.CreateResponse(HttpStatusCode.NotFound,
        "User ID= " + id + " not found");
}
```

```
  m: Debug
  exe: (CLR v4.0.30319: /LM/W3SVC/2/ROOT-1-132302222775672406): Loaded 'C:\Windows\Microsoft.NET\Framework\v4.0.30319\Temporary ASP.NET Files\root\12330222775672406\1\123302227756724061': Loaded 'C:\Windows\Microsoft.NET\assembly\GAC_MSIL\System.Windows.Forms\v4.0.0.0_b77a5c5619
  exe: (CLR v4.0.30319: /LM/W3SVC/2/ROOT-1-132302222775672406): Loaded 'C:\Windows\Microsoft.NET\assembly\GAC_MSIL\System.v4.0.0.0_b77a5c5619
  exe: (CLR v4.0.30319: /LM/W3SVC/2/ROOT-1-132302222775672406): Loaded 'C:\Windows\Microsoft.NET\assembly\GAC_32\System.Data.OracleClient\v4.0.0.0_b77a5c5619
```

```
741
742
743
744
745
746
747
748
749
750
751
752
753
754
755
756
757
758
759
760
761
762
763
764
765
766
767
768
769
770
771
772
773
774
775
776
777
778
779
```

UQP.Controllers.User\_DetailsController

```
    return Request.CreateResponse(HttpStatusCode.NotFound,
        "User ID= " + id + " not found"
    );
}

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

// PUT: api/User_Details/5
[ResponseType(typeof(void))]
0 references
public HttpResponseMessage PutUser_Details(int id, User_Details user_Details)
{
    if (!ModelState.IsValid)
    {
        return Request.CreateResponse(HttpStatusCode.BadRequest, "Inputs are invalid ");
    }
    try
    {
        //using (var db = new UQPEntities("UQPEntities"))
        //{
        var ab = db.User_Details.FirstOrDefault(e => e.User_ID == id);
        if (ab != null)
        {
            List<OfficeStaff_AssignedProvider> ofcadd =
                db.OfficeStaff_AssignedProvider.Where(e => e.User_Id == user_Details.User_ID).ToList();
            ab.Offic...
```

Debug Any CPU IIS Express (Google Chrome) UQP.Controllers.User\_DetailsController PutUserDetailStatus(int id, ModifyStatus user\_status)

```
try
{
    //using (var db = new UQPEntities("UQPEntities"))
    //{
        var ab = db.User_Details.FirstOrDefault(e => e.User_ID == id);
        if (ab != null)
        {
            List<OfficeStaff_AssignedProvider> ofcadd =
                db.OfficeStaff_AssignedProvider.Where(e => e.User_Id == user_Details.User_ID).ToList();
            db.OfficeStaff_AssignedProvider.RemoveRange(ofcadd);
            db.SaveChanges();

            if (user_Details.User_Role_ID == 35)
            {
                var query = (from p in db.ProviderDetailsTestings
                            join a in db.providertesting_address
                            on p.providerid equals a.providerid
                            where p.providerid == user_Details.User_Provider_ID && a.isdefault == true && p.visible == true
                            select a).FirstOrDefault();

                if (query.recordid != user_Details.PrimaryFacilityId)
                {
                    query.isdefault = false;
                    var newaddress = db.providertesting_address.Where(e => e.recordid == user_Details.PrimaryFacilityId).FirstOrDefault();
                    if (newaddress != null)
                    {
                        newaddress.isdefault = true;
                    }
                }
            }
        }
    //}
}
```

```
792
793
794
795
796
797
798
799
800
801
802
803
804
805
806
807
808
809
810
811
812
813
814
815
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
```

```
UQP.Controllers.User_DetailsController
PutUserDetailStatus(int id, ModifyStatus user_status)

ab.HedisTeam = user_Details.HedisTeam;
ab.MRATeamId = user_Details.MRATeamId;
ab.HedisTeamId = user_Details.HedisTeamId;
ab.OrganisationType = user_Details.OrganisationType;
ab.User_IPA_ID = user_Details.User_IPA_ID;
ab.DefaultIPA = user_Details.DefaultIPA;
foreach (var q in user_Details.OfficeStaff_AssignedProvider)
{
    OfficeStaff_AssignedProvider officestaffobj = new OfficeStaff_AssignedProvider();
    officestaffobj.User_ProviderId = q.Id;
    officestaffobj.User_Id = id;
    db.OfficeStaff_AssignedProvider.Add(officestaffobj);
}

db.SaveChanges();
}

//var s = (User_Details)ab;
user_Details = ab;

db.SaveChanges();
return Request.CreateResponse(HttpStatusCode.OK, "User details has been updated successfully");
}
else
{
    return Request.CreateResponse(HttpStatusCode.NotFound,
        "User ID= " + id + " not found"
    );
}
//}
```

Project Build Debug Team Tools Test Analyze Window Help

User\_Control Explorer User\_DetailsController.cs\* IIS Express (Google Chrome)

```
UQP
834     return Request.CreateResponse(HttpStatusCode.OK, "User details has been updated successfully");
835 }
836 else
837 {
838     return Request.CreateResponse(HttpStatusCode.NotFound,
839         "User ID= " + id + " not found"
840         );
841 }
842 //}
843 }
844 catch (Exception ex)
845 {
846     return Request.CreateErrorResponse(HttpStatusCode.InternalServerError, ex);
847 }
848 }
849
850 [Route("api/User_Details/PostUser_EmailVerify")]
851 public bool PostUser_EmailVerify(EmailString useremail)
852 {
853     //using (var db = new UQPEntities("UQPEntities"))
854     //{
855     if (useremail.EmailID != string.Empty || useremail.EmailID.Contains("@"))
856     {
857         var result = (from eml in db.User_Details
858                     where eml.User_Email.Contains(useremail.EmailID) && eml.Status == true
859                     select eml.User_Email).ToList();
860         //var result = (from eml in db.User_Details
861
862
0 %
```

View Project Build Debug Team Tools Test Analyze Window Help

User\_DetailsController.cs

```
57
58
59     {
60         var result = (from eml in db.User_Details
61                       where eml.User_Email.Contains(useremail.EmailID) && eml.Status == true
62                       select eml.User_Email).ToList();
63
64         //var result = (from eml in db.User_Details
65         //              where eml.User_Email.Contains(useremail.EmailID)
66         //              select eml.User_Email).ToList();
67         if (result.Count > 0)
68         {
69             return true;
70         }
71         else
72         {
73             return false;
74         }
75     }
76
77     return false;
78 }
79 //}
80
81 [Route("api/User_Détails/ChangePassword")]
82 public ResponseMessage ChangePassword(FirtUserLoginInPut firstuser)
83 {
84     using (var db = new UQPEntities())
85     {
86         var result = (from eml in db.User_Details
87                       where eml.User_Email.Contains(firstuser.EmailID) && eml.Status == true
88                       select eml.User_Email).ToList();
89
90         //var result = (from eml in db.User_Details
91         //              where eml.User_Email.Contains(firstuser.EmailID)
92         //              select eml.User_Email).ToList();
93         if (result.Count > 0)
94         {
95             return true;
96         }
97         else
98         {
99             return false;
100        }
101    }
102 }
```

```
        }
    //}
}

[Route("api/User_Details/ChangePassword")]
public HttpResponseMessage ChangePassword(FirtUserLoginInput firstuser)
{
    using (var db = new UQPEntities())
    {
        HttpResponseMessage objresponse = new HttpResponseMessage();
        SHA1CryptoServiceProvider encryptFirstUserPassword = new SHA1CryptoServiceProvider();
        byte[] hashValueFirstUser = encryptFirstUserPassword.ComputeHash(Encoding.Default.GetBytes(firstuser.oldpassword));
        string encryptedValueFirstUser = BitConverter.ToString(hashValueFirstUser);
        var entity = db.User_Details.FirstOrDefault(e => e.User_Email == firstuser.email && e.User_Password == encryptedValueFirstUser && firstuser.userid);

        var encryptedMessage = "";
        if (entity != null)
        {
            encryptedMessage = firstuser.newpassword;
            SHA1CryptoServiceProvider encrypt = new SHA1CryptoServiceProvider();
            byte[] hashValueMessage = encrypt.ComputeHash(Encoding.Default.GetBytes(encryptedMessage));
            string hexOfValueMessage = BitConverter.ToString(hashValueMessage);
            entity.User_Password = hexOfValueMessage;
            //To update modifiedpassword in user_details table,which further,used to check when the password is last changed
            entity.ModifiedPasswordDate = DateTime.Now;
            entity.Message = null;
        }
    }
}
```

Output from: Debug

Debug Any CPU IIS Express (Google Chrome)

User\_DetailsController.cs\* ▾ X

UQP.Controllers.User\_DetailsController ChangePassword(FirtUserLoginInput firstuser)

```
byte[] hashValueMessage = encrypt.ComputeHash(Encoding.Default.GetBytes(encryptedMessage));
string hexOfValueMessage = BitConverter.ToString(hashValueMessage);
entity.User_Password = hexOfValueMessage;
//To update modifiedpassword in user_details table,which further,used to check when the password is last changed
entity.ModifiedPasswordDate = DateTime.Now;
entity.Message = null;
entity.FirstTimeUser = false;

db.SaveChanges();
if (ConfigSettings.email.Contains("true"))
{
    processAuthEmail obj2 = new processAuthEmail();
    var EmailContent1 = (from a in db.NewEmailContents
                         where a.Id == 2
                         select new
                         {
                             a.EmailSubject,
                             a.EmailBody,
                             a.EmailSignature
                         }).ToList();
    foreach (var per in EmailContent1)
    {
        obj2.body = "Hello " + entity.User_FirstName + per.EmailBody;
        obj2.subject = per.EmailSubject;
        obj2.signature = per.EmailSignature;
    }
    obj2.replyto = firstuser.email;
    using (var ProcessAuthController = new ProcessAuthController())
}
```

User Control Explorer    User\_DetailsController.cs\*    X    UQP.Controllers.User\_DetailsController    ChangePassword(FirtUserLoginInput firstuser)

```
19
20
21     foreach (var per in EmailContent1)
22     {
23
24         obj2.body = "Hello " + entity.User_FirstName + per.EmailBody;
25         obj2.subject = per.EmailSubject;
26         obj2.signature = per.EmailSignature;
27
28     }
29     obj2.replyto = firstuser.email;
30     using (var ProcessAuthController = new ProcessAuthController())
31     {
32         ProcessAuthController.processAuthEmail(obj2);
33     }
34
35     objresponse.status = true;
36     objresponse.responsemessage = "Password successfully updated";
37     return objresponse;
38
39 }
40 else
41 {
42     objresponse.status = false;
43     objresponse.responsemessage = "Password updatation failed";
44     return objresponse;
45 }
46
47 }
```

Output from: Debug

User Control Explorer User\_DetailsController.cs\* UQP

UQP.Controllers.User\_DetailsController

ChangePassword(FirtUserLoginInput firstuser)

```
943     objresponse.responsemessage = "Password updatation failed";
944     return objresponse;
945 }
946 }
947 }
948 }
949
950 [HttpPost]
951 [Route("api/User_Details/LoginUser")]
952 //public CustomUser ValidateUser(string email, string password)
953 //references
954 public CustomUser ValidateUser(LoginInput logininput)
955 {
956     //using (var db = new UQPEntities("UQPEntities"))
957     //{
958         CustomUser objuser = new CustomUser();
959         var entity = db.User_Details.FirstOrDefault(e => e.User_Email == logininput.email & e.User_Password == logininput.password & e.Status == 1);
960
961         if (entity != null)
962         {
963             objuser.email = entity.User_Email;
964             objuser.userid = entity.User_ID;
965             objuser.Isfirsttime = entity.FirstTimeUser;
966             objuser.name = entity.User_FirstName;
967             objuser.isuservalid = true;
968             if (entity.User_Provider_ID != null)
969             {
970                 objuser.providerid = entity.User_Provider_ID;
971             }
972         }
973     }
974 }
```

Visual Studio

User\_DetailsController.cs\* - UQP.Controllers.User\_DetailsController - ChangePassword(FirUserLoginInput firstuser)

```
192.168.99.105
```

```
59
60     {
61         objuser.providerid = entity.User_Provider_ID;
62     }
63     if (entity.User_HealthPlan_ID != null)
64     {
65         objuser.healthplanid = entity.User_HealthPlan_ID;
66     }
67     if (entity.User_IPA_ID != null)
68     {
69         objuser.ipaid = null;//entity.User_IPA_ID;
70     }
71     if (entity.User_Member_ID != null)
72     {
73         objuser.memberid = entity.User_Member_ID;
74     }
75
76     return (objuser);
77 }
78 else
79 {
80     objuser.isuservalid = false;
81     return objuser;
82 }
83 //}
84 }
85
86 [HttpGet]
87 [Route("api/userdetails/profile")]
88 [References]
89 public IHttpActionResult GetProfile()
90 {
91 }
```

```
    //}
}

[HttpGet]
[Route("api/userdetails/profile")]
public HttpResponseMessage GetProfile()
{
    try
    {
        //using (var db = new UQPEntities("UQPEntities"))
        //{
        ClaimsPrincipal principal = Request.GetRequestContext().Principal as ClaimsPrincipal;
        var userName = principal.Claims.Where(c => c.Type == "useremail").Single().Value;
        var loginid = principal.Claims.Where(c => c.Type == "id").Single().Value;
        int userid = Convert.ToInt32(principal.Claims.Where(c => c.Type == "userid").Single().Value);
        CustomUser objuser = new CustomUser();
        var entity = db.User_Details.AsNoTracking().FirstOrDefault(e => e.User_ID == userid);
        Datetime? lastrefresheddate = db.NewDashboardSuperset.Where(x => x.Executive_id == 1)
            .Select(x => x.LastGapRefreshedDate).FirstOrDefault();
        List<ACLDetails> acclist = new List<ACLDetails>();
        List<CustomIPA> ipalist = new List<CustomIPA>();

        if (entity != null)
        {
            objuser.lastgrefresh = lastrefresheddate;
            if (entity.User_IPA_ID != null)
            {
                user_ipaids = entity.User_IPA_ID.Split(',');
            }
        }
    }
}
```

Source Control Explorer    User\_DetailsController.cs\*    X

UQP

```
1016
1017     if (entity == null)
1018     {
1019         objuser.lastgrefresh = lastrefresheddate;
1020         if (entity.User_IPA_ID != null)
1021         {
1022             var stripaids = entity.User_IPA_ID.Split(',');
1023             int[] ipaid = stripaids.Select(int.Parse).ToArray();
1024             foreach (var ip in ipaid)
1025             {
1026                 CustomIPA ipaobj = new CustomIPA();
1027                 var ipainfo = (from ips in db.IPAs.AsNoTracking()
1028                               where ips.IPA_ID == ip
1029                               select new { ips.IPA_ID, ips.IPA_Name, ips.IPA_Code }).FirstOrDefault();
1030                 ipaobj.IPA_ID = ipainfo.IPA_ID;
1031                 ipaobj.IPA_Name = ipainfo.IPA_Name;
1032                 ipalist.Add(ipaobj);
1033
1034             }
1035         }
1036     }
1037
1038     CustomIPA ipadefobj = new CustomIPA();
1039     var ipadefvar =
1040
1041     (from ips in db.IPAs.AsNoTracking()
1042       where ips.IPA_ID == entity.DefaultIPA
1043       select new { ips.IPA_ID, ips.IPA_Name, ips.IPA_Code }).FirstOrDefault();
1044     if (ipadefvar != null)
1045     {
1046
1047 }
```

Output

Source Control Explorer    User\_DetailsController.cs\*    X    UQP.Controllers.User\_DetailsController

```
UQP
1043     where ips.IPA_ID == entity.defaultIPA
1044     select new { ips.IPA_ID, ips.IPA_Name, ips.IPA_Code }).FirstOrDefault();
1045     if (ipadefvar != null)
1046     {
1047         ipadefobj.IPA_ID = ipadefvar.IPA_ID;
1048         ipadefobj.IPA_Name = ipadefvar.IPA_Name;
1049         ipadefobj.ipaicode = ipadefvar.IPA_Code;
1050     }
1051     objuser.defaultIPAOBJ = ipadefobj;
1052
1053     Organisation orgobj = new Organisation();
1054     if (entity.OrganisationType != null)
1055     {
1056         var org = entity.OrganisationType.Split(',');
1057
1058         foreach (var per in org)
1059         {
1060             if (per == "1")
1061             {
1062                 orgobj.hedis = true;
1063             }
1064             if (per == "2")
1065             {
1066                 orgobj.mra = true;
1067             }
1068             if (per == "3")
1069             {
1070                 orgobj.aco = true;
1071             }
1072         }
1073     }
1074
1075     if (entity.OrganisationType != null)
1076     {
1077         var org = entity.OrganisationType.Split(',');
1078
1079         foreach (var per in org)
1080         {
1081             if (per == "1")
1082             {
1083                 orgobj.hedis = true;
1084             }
1085             if (per == "2")
1086             {
1087                 orgobj.mra = true;
1088             }
1089             if (per == "3")
1090             {
1091                 orgobj.aco = true;
1092             }
1093         }
1094     }
1095
1096     if (entity.OrganisationType != null)
1097     {
1098         var org = entity.OrganisationType.Split(',');
1099
1100         foreach (var per in org)
1101         {
1102             if (per == "1")
1103             {
1104                 orgobj.hedis = true;
1105             }
1106             if (per == "2")
1107             {
1108                 orgobj.mra = true;
1109             }
1110             if (per == "3")
1111             {
1112                 orgobj.aco = true;
1113             }
1114         }
1115     }
1116
1117     if (entity.OrganisationType != null)
1118     {
1119         var org = entity.OrganisationType.Split(',');
1120
1121         foreach (var per in org)
1122         {
1123             if (per == "1")
1124             {
1125                 orgobj.hedis = true;
1126             }
1127             if (per == "2")
1128             {
1129                 orgobj.mra = true;
1130             }
1131             if (per == "3")
1132             {
1133                 orgobj.aco = true;
1134             }
1135         }
1136     }
1137
1138     if (entity.OrganisationType != null)
1139     {
1140         var org = entity.OrganisationType.Split(',');
1141
1142         foreach (var per in org)
1143         {
1144             if (per == "1")
1145             {
1146                 orgobj.hedis = true;
1147             }
1148             if (per == "2")
1149             {
1150                 orgobj.mra = true;
1151             }
1152             if (per == "3")
1153             {
1154                 orgobj.aco = true;
1155             }
1156         }
1157     }
1158
1159     if (entity.OrganisationType != null)
1160     {
1161         var org = entity.OrganisationType.Split(',');
1162
1163         foreach (var per in org)
1164         {
1165             if (per == "1")
1166             {
1167                 orgobj.hedis = true;
1168             }
1169             if (per == "2")
1170             {
1171                 orgobj.mra = true;
1172             }
1173             if (per == "3")
1174             {
1175                 orgobj.aco = true;
1176             }
1177         }
1178     }
1179
1180     if (entity.OrganisationType != null)
1181     {
1182         var org = entity.OrganisationType.Split(',');
1183
1184         foreach (var per in org)
1185         {
1186             if (per == "1")
1187             {
1188                 orgobj.hedis = true;
1189             }
1190             if (per == "2")
1191             {
1192                 orgobj.mra = true;
1193             }
1194             if (per == "3")
1195             {
1196                 orgobj.aco = true;
1197             }
1198         }
1199     }
1200
1201     if (entity.OrganisationType != null)
1202     {
1203         var org = entity.OrganisationType.Split(',');
1204
1205         foreach (var per in org)
1206         {
1207             if (per == "1")
1208             {
1209                 orgobj.hedis = true;
1210             }
1211             if (per == "2")
1212             {
1213                 orgobj.mra = true;
1214             }
1215             if (per == "3")
1216             {
1217                 orgobj.aco = true;
1218             }
1219         }
1220     }
1221
1222     if (entity.OrganisationType != null)
1223     {
1224         var org = entity.OrganisationType.Split(',');
1225
1226         foreach (var per in org)
1227         {
1228             if (per == "1")
1229             {
1230                 orgobj.hedis = true;
1231             }
1232             if (per == "2")
1233             {
1234                 orgobj.mra = true;
1235             }
1236             if (per == "3")
1237             {
1238                 orgobj.aco = true;
1239             }
1240         }
1241     }
1242
1243     if (entity.OrganisationType != null)
1244     {
1245         var org = entity.OrganisationType.Split(',');
1246
1247         foreach (var per in org)
1248         {
1249             if (per == "1")
1250             {
1251                 orgobj.hedis = true;
1252             }
1253             if (per == "2")
1254             {
1255                 orgobj.mra = true;
1256             }
1257             if (per == "3")
1258             {
1259                 orgobj.aco = true;
1260             }
1261         }
1262     }
1263
1264     if (entity.OrganisationType != null)
1265     {
1266         var org = entity.OrganisationType.Split(',');
1267
1268         foreach (var per in org)
1269         {
1270             if (per == "1")
1271             {
1272                 orgobj.hedis = true;
1273             }
1274             if (per == "2")
1275             {
1276                 orgobj.mra = true;
1277             }
1278             if (per == "3")
1279             {
1280                 orgobj.aco = true;
1281             }
1282         }
1283     }
1284
1285     if (entity.OrganisationType != null)
1286     {
1287         var org = entity.OrganisationType.Split(',');
1288
1289         foreach (var per in org)
1290         {
1291             if (per == "1")
1292             {
1293                 orgobj.hedis = true;
1294             }
1295             if (per == "2")
1296             {
1297                 orgobj.mra = true;
1298             }
1299             if (per == "3")
1300             {
1301                 orgobj.aco = true;
1302             }
1303         }
1304     }
1305
1306     if (entity.OrganisationType != null)
1307     {
1308         var org = entity.OrganisationType.Split(',');
1309
1310         foreach (var per in org)
1311         {
1312             if (per == "1")
1313             {
1314                 orgobj.hedis = true;
1315             }
1316             if (per == "2")
1317             {
1318                 orgobj.mra = true;
1319             }
1320             if (per == "3")
1321             {
1322                 orgobj.aco = true;
1323             }
1324         }
1325     }
1326
1327     if (entity.OrganisationType != null)
1328     {
1329         var org = entity.OrganisationType.Split(',');
1330
1331         foreach (var per in org)
1332         {
1333             if (per == "1")
1334             {
1335                 orgobj.hedis = true;
1336             }
1337             if (per == "2")
1338             {
1339                 orgobj.mra = true;
1340             }
1341             if (per == "3")
1342             {
1343                 orgobj.aco = true;
1344             }
1345         }
1346     }
1347
1348     if (entity.OrganisationType != null)
1349     {
1350         var org = entity.OrganisationType.Split(',');
1351
1352         foreach (var per in org)
1353         {
1354             if (per == "1")
1355             {
1356                 orgobj.hedis = true;
1357             }
1358             if (per == "2")
1359             {
1360                 orgobj.mra = true;
1361             }
1362             if (per == "3")
1363             {
1364                 orgobj.aco = true;
1365             }
1366         }
1367     }
1368
1369     if (entity.OrganisationType != null)
1370     {
1371         var org = entity.OrganisationType.Split(',');
1372
1373         foreach (var per in org)
1374         {
1375             if (per == "1")
1376             {
1377                 orgobj.hedis = true;
1378             }
1379             if (per == "2")
1380             {
1381                 orgobj.mra = true;
1382             }
1383             if (per == "3")
1384             {
1385                 orgobj.aco = true;
1386             }
1387         }
1388     }
1389
1390     if (entity.OrganisationType != null)
1391     {
1392         var org = entity.OrganisationType.Split(',');
1393
1394         foreach (var per in org)
1395         {
1396             if (per == "1")
1397             {
1398                 orgobj.hedis = true;
1399             }
1400             if (per == "2")
1401             {
1402                 orgobj.mra = true;
1403             }
1404             if (per == "3")
1405             {
1406                 orgobj.aco = true;
1407             }
1408         }
1409     }
1410
1411     if (entity.OrganisationType != null)
1412     {
1413         var org = entity.OrganisationType.Split(',');
1414
1415         foreach (var per in org)
1416         {
1417             if (per == "1")
1418             {
1419                 orgobj.hedis = true;
1420             }
1421             if (per == "2")
1422             {
1423                 orgobj.mra = true;
1424             }
1425             if (per == "3")
1426             {
1427                 orgobj.aco = true;
1428             }
1429         }
1430     }
1431
1432     if (entity.OrganisationType != null)
1433     {
1434         var org = entity.OrganisationType.Split(',');
1435
1436         foreach (var per in org)
1437         {
1438             if (per == "1")
1439             {
1440                 orgobj.hedis = true;
1441             }
1442             if (per == "2")
1443             {
1444                 orgobj.mra = true;
1445             }
1446             if (per == "3")
1447             {
1448                 orgobj.aco = true;
1449             }
1450         }
1451     }
1452
1453     if (entity.OrganisationType != null)
1454     {
1455         var org = entity.OrganisationType.Split(',');
1456
1457         foreach (var per in org)
1458         {
1459             if (per == "1")
1460             {
1461                 orgobj.hedis = true;
1462             }
1463             if (per == "2")
1464             {
1465                 orgobj.mra = true;
1466             }
1467             if (per == "3")
1468             {
1469                 orgobj.aco = true;
1470             }
1471         }
1472     }
1473
1474     if (entity.OrganisationType != null)
1475     {
1476         var org = entity.OrganisationType.Split(',');
1477
1478         foreach (var per in org)
1479         {
1480             if (per == "1")
1481             {
1482                 orgobj.hedis = true;
1483             }
1484             if (per == "2")
1485             {
1486                 orgobj.mra = true;
1487             }
1488             if (per == "3")
1489             {
1490                 orgobj.aco = true;
1491             }
1492         }
1493     }
1494
1495     if (entity.OrganisationType != null)
1496     {
1497         var org = entity.OrganisationType.Split(',');
1498
1499         foreach (var per in org)
1500         {
1501             if (per == "1")
1502             {
1503                 orgobj.hedis = true;
1504             }
1505             if (per == "2")
1506             {
1507                 orgobj.mra = true;
1508             }
1509             if (per == "3")
1510             {
1511                 orgobj.aco = true;
1512             }
1513         }
1514     }
1515
1516     if (entity.OrganisationType != null)
1517     {
1518         var org = entity.OrganisationType.Split(',');
1519
1520         foreach (var per in org)
1521         {
1522             if (per == "1")
1523             {
1524                 orgobj.hedis = true;
1525             }
1526             if (per == "2")
1527             {
1528                 orgobj.mra = true;
1529             }
1530             if (per == "3")
1531             {
1532                 orgobj.aco = true;
1533             }
1534         }
1535     }
1536
1537     if (entity.OrganisationType != null)
1538     {
1539         var org = entity.OrganisationType.Split(',');
1540
1541         foreach (var per in org)
1542         {
1543             if (per == "1")
1544             {
1545                 orgobj.hedis = true;
1546             }
1547             if (per == "2")
1548             {
1549                 orgobj.mra = true;
1550             }
1551             if (per == "3")
1552             {
1553                 orgobj.aco = true;
1554             }
1555         }
1556     }
1557
1558     if (entity.OrganisationType != null)
1559     {
1560         var org = entity.OrganisationType.Split(',');
1561
1562         foreach (var per in org)
1563         {
1564             if (per == "1")
1565             {
1566                 orgobj.hedis = true;
1567             }
1568             if (per == "2")
1569             {
1570                 orgobj.mra = true;
1571             }
1572             if (per == "3")
1573             {
1574                 orgobj.aco = true;
1575             }
1576         }
1577     }
1578
1579     if (entity.OrganisationType != null)
1580     {
1581         var org = entity.OrganisationType.Split(',');
1582
1583         foreach (var per in org)
1584         {
1585             if (per == "1")
1586             {
1587                 orgobj.hedis = true;
1588             }
1589             if (per == "2")
1590             {
1591                 orgobj.mra = true;
1592             }
1593             if (per == "3")
1594             {
1595                 orgobj.aco = true;
1596             }
1597         }
1598     }
1599
1600     if (entity.OrganisationType != null)
1601     {
1602         var org = entity.OrganisationType.Split(',');
1603
1604         foreach (var per in org)
1605         {
1606             if (per == "1")
1607             {
1608                 orgobj.hedis = true;
1609             }
1610             if (per == "2")
1611             {
1612                 orgobj.mra = true;
1613             }
1614             if (per == "3")
1615             {
1616                 orgobj.aco = true;
1617             }
1618         }
1619     }
1620
1621     if (entity.OrganisationType != null)
1622     {
1623         var org = entity.OrganisationType.Split(',');
1624
1625         foreach (var per in org)
1626         {
1627             if (per == "1")
1628             {
1629                 orgobj.hedis = true;
1630             }
1631             if (per == "2")
1632             {
1633                 orgobj.mra = true;
1634             }
1635             if (per == "3")
1636             {
1637                 orgobj.aco = true;
1638             }
1639         }
1640     }
1641
1642     if (entity.OrganisationType != null)
1643     {
1644         var org = entity.OrganisationType.Split(',');
1645
1646         foreach (var per in org)
1647         {
1648             if (per == "1")
1649             {
1650                 orgobj.hedis = true;
1651             }
1652             if (per == "2")
1653             {
1654                 orgobj.mra = true;
1655             }
1656             if (per == "3")
1657             {
1658                 orgobj.aco = true;
1659             }
1660         }
1661     }
1662
1663     if (entity.OrganisationType != null)
1664     {
1665         var org = entity.OrganisationType.Split(',');
1666
1667         foreach (var per in org)
1668         {
1669             if (per == "1")
1670             {
1671                 orgobj.hedis = true;
1672             }
1673             if (per == "2")
1674             {
1675                 orgobj.mra = true;
1676             }
1677             if (per == "3")
1678             {
1679                 orgobj.aco = true;
1680             }
1681         }
1682     }
1683
1684     if (entity.OrganisationType != null)
1685     {
1686         var org = entity.OrganisationType.Split(',');
1687
1688         foreach (var per in org)
1689         {
1690             if (per == "1")
1691             {
1692                 orgobj.hedis = true;
1693             }
1694             if (per == "2")
1695             {
1696                 orgobj.mra = true;
1697             }
1698             if (per == "3")
1699             {
1700                 orgobj.aco = true;
1701             }
1702         }
1703     }
1704
1705     if (entity.OrganisationType != null)
1706     {
1707         var org = entity.OrganisationType.Split(',');
1708
1709         foreach (var per in org)
1710         {
1711             if (per == "1")
1712             {
1713                 orgobj.hedis = true;
1714             }
1715             if (per == "2")
1716             {
1717                 orgobj.mra = true;
1718             }
1719             if (per == "3")
1720             {
1721                 orgobj.aco = true;
1722             }
1723         }
1724     }
1725
1726     if (entity.OrganisationType != null)
1727     {
1728         var org = entity.OrganisationType.Split(',');
1729
1730         foreach (var per in org)
1731         {
1732             if (per == "1")
1733             {
1734                 orgobj.hedis = true;
1735             }
1736             if (per == "2")
1737             {
1738                 orgobj.mra = true;
1739             }
1740             if (per == "3")
1741             {
1742                 orgobj.aco = true;
1743             }
1744         }
1745     }
1746
1747     if (entity.OrganisationType != null)
1748     {
1749         var org = entity.OrganisationType.Split(',');
1750
1751         foreach (var per in org)
1752         {
1753             if (per == "1")
1754             {
1755                 orgobj.hedis = true;
1756             }
1757             if (per == "2")
1758             {
1759                 orgobj.mra = true;
1760             }
1761             if (per == "3")
1762             {
1763                 orgobj.aco = true;
1764             }
1765         }
1766     }
1767
1768     if (entity.OrganisationType != null)
1769     {
1770         var org = entity.OrganisationType.Split(',');
1771
1772         foreach (var per in org)
1773         {
1774             if (per == "1")
1775             {
1776                 orgobj.hedis = true;
1777             }
1778             if (per == "2")
1779             {
1780                 orgobj.mra = true;
1781             }
1782             if (per == "3")
1783             {
1784                 orgobj.aco = true;
1785             }
1786         }
1787     }
1788
1789     if (entity.OrganisationType != null)
1790     {
1791         var org = entity.OrganisationType.Split(',');
1792
1793         foreach (var per in org)
1794         {
1795             if (per == "1")
1796             {
1797                 orgobj.hedis = true;
1798             }
1799             if (per == "2")
1800             {
1801                 orgobj.mra = true;
1802             }
1803             if (per == "3")
1804             {
1805                 orgobj.aco = true;
1806             }
1807         }
1808     }
1809
1810     if (entity.OrganisationType != null)
1811     {
1812         var org = entity.OrganisationType.Split(',');
1813
1814         foreach (var per in org)
1815         {
1816             if (per == "1")
1817             {
1818                 orgobj.hedis = true;
1819             }
1820             if (per == "2")
1821             {
1822                 orgobj.mra = true;
1823             }
1824             if (per == "3")
1825             {
1826                 orgobj.aco = true;
1827             }
1828         }
1829     }
1830
1831     if (entity.OrganisationType != null)
1832     {
1833         var org = entity.OrganisationType.Split(',');
1834
1835         foreach (var per in org)
1836         {
1837             if (per == "1")
1838             {
1839                 orgobj.hedis = true;
1840             }
1841             if (per == "2")
1842             {
1843                 orgobj.mra = true;
1844             }
1845             if (per == "3")
1846             {
1847                 orgobj.aco = true;
1848             }
1849         }
1850     }
1851
1852     if (entity.OrganisationType != null)
1853     {
1854         var org = entity.OrganisationType.Split(',');
1855
1856         foreach (var per in org)
1857         {
1858             if (per == "1")
1859             {
1860                 orgobj.hedis = true;
1861             }
1862             if (per == "2")
1863             {
1864                 orgobj.mra = true;
1865             }
1866             if (per == "3")
1867             {
1868                 orgobj.aco = true;
1869             }
1870         }
1871     }
1872
1873     if (entity.OrganisationType != null)
1874     {
1875         var org = entity.OrganisationType.Split(',');
1876
1877         foreach (var per in org)
1878         {
1879             if (per == "1")
1880             {
1881                 orgobj.hedis = true;
1882             }
1883             if (per == "2")
1884             {
1885                 orgobj.mra = true;
1886             }
1887             if (per == "3")
1888             {
1889                 orgobj.aco = true;
1890             }
1891         }
1892     }
1893
1894     if (entity.OrganisationType != null)
1895     {
1896         var org = entity.OrganisationType.Split(',');
1897
1898         foreach (var per in org)
1899         {
1900             if (per == "1")
1901             {
1902                 orgobj.hedis = true;
1903             }
1904             if (per == "2")
1905             {
1906                 orgobj.mra = true;
1907             }
1908             if (per == "3")
1909             {
1910                 orgobj.aco = true;
1911             }
1912         }
1913     }
1914
1915     if (entity.OrganisationType != null)
1916     {
1917         var org = entity.OrganisationType.Split(',');
1918
1919         foreach (var per in org)
1920         {
1921             if (per == "1")
1922             {
1923                 orgobj.hedis = true;
1924             }
1925             if (per == "2")
1926             {
1927                 orgobj.mra = true;
1928             }
1929             if (per == "3")
1930             {
1931                 orgobj.aco = true;
1932             }
1933         }
1934     }
1935
1936     if (entity.OrganisationType != null)
1937     {
1938         var org = entity.OrganisationType.Split(',');
1939
1940         foreach (var per in org)
1941         {
1942             if (per == "1")
1943             {
1944                 orgobj.hedis = true;
1945             }
1946             if (per == "2")
1947             {
1948                 orgobj.mra = true;
1949             }
1950             if (per == "3")
1951             {
1952                 orgobj.aco = true;
1953             }
1954         }
1955     }
1956
1957     if (entity.OrganisationType != null)
1958     {
1959         var org = entity.OrganisationType.Split(',');
1960
1961         foreach (var per in org)
1962         {
1963             if (per == "1")
1964             {
1965                 orgobj.hedis = true;
1966             }
1967             if (per == "2")
1968             {
1969                 orgobj.mra = true;
1970             }
1971             if (per == "3")
1972             {
1973                 orgobj.aco = true;
1974             }
1975         }
1976     }
1977
1978     if (entity.OrganisationType != null)
1979     {
1980         var org = entity.OrganisationType.Split(',');
1981
1982         foreach (var per in org)
1983         {
1984             if (per == "1")
1985             {
1986                 orgobj.hedis = true;
1987             }
1988             if (per == "2")
1989             {
1990                 orgobj.mra = true;
1991             }
1992             if (per == "3")
1993             {
1994                 orgobj.aco = true;
1995             }
1996         }
1997     }
1998
1999     if (entity.OrganisationType != null)
2000     {
2001         var org = entity.OrganisationType.Split(',');
2002
2003         foreach (var per in org)
2004         {
2005             if (per == "1")
2006             {
2007                 orgobj.hedis = true;
2008             }
2009             if (per == "2")
2010             {
2011                 orgobj.mra = true;
2012             }
2013             if (per == "3")
2014             {
2015                 orgobj.aco = true;
2016             }
2017         }
2018     }
2019
2020     if (entity.OrganisationType != null)
2021     {
2022         var org = entity.OrganisationType.Split(',');
2023
2024         foreach (var per in org)
2025         {
2026             if (per == "1")
2027             {
2028                 orgobj.hedis = true;
2029             }
2030             if (per == "2")
2031             {
2032                 orgobj.mra = true;
2033             }
2034             if (per == "3")
2035             {
2036                 orgobj.aco = true;
2037             }
2038         }
2039     }
2040
2041     if (entity.OrganisationType != null)
2042     {
2043         var org = entity.OrganisationType.Split(',');
2044
2045         foreach (var per in org)
2046         {
2047             if (per == "1")
2048             {
2049                 orgobj.hedis = true;
2050             }
2051             if (per == "2")
2052             {
2053                 orgobj.mra = true;
2054             }
2055             if (per == "3")
2056             {
2057                 orgobj.aco = true;
2058             }
2059         }
2060     }
2061
2062     if (entity.OrganisationType != null)
2063     {
2064         var org = entity.OrganisationType.Split(',');
2065
2066         foreach (var per in org)
2067         {
2068             if (per == "1")
2069             {
2070                 orgobj.hedis = true;
2071             }
2072             if (per == "2")
2073             {
2074                 orgobj.mra = true;
2075             }
2076             if (per == "3")
2077             {
2078                 orgobj.aco = true;
2079             }
2080         }
2081     }
2082
2083     if (entity.OrganisationType != null)
2084     {
2085         var org = entity.OrganisationType.Split(',');
2086
2087         foreach (var per in org)
2088         {
2089             if (per == "1")
2090             {
2091                 orgobj.hedis = true;
2092             }
2093             if (per == "2")
2094             {
2095                 orgobj.mra = true;
2096             }
2097             if (per == "3")
2098             {
2099                 orgobj.aco = true;
2100             }
2101         }
2102     }
2103
2104     if (entity.OrganisationType != null)
2105     {
2106         var org = entity.OrganisationType.Split(',');
2107
2108         foreach (var per in org)
2109         {
2110             if (per == "1")
2111             {
2112                 orgobj.hedis = true;
2113             }
2114             if (per == "2")
2115             {
2116                 orgobj.mra = true;
2117             }
2118             if (per == "3")
2119             {
2120                 orgobj.aco = true;
2121             }
2122         }
2123     }
2124
2125     if (entity.OrganisationType != null)
2126     {
2127         var org = entity.OrganisationType.Split(',');
2128
2129         foreach (var per in org)
2130         {
2131             if (per == "1")
2132             {
2133                 orgobj.hedis = true;
2134             }
2135             if (per == "2")
2136             {
2137                 orgobj.mra = true;
2138             }
2139             if (per == "3")
2140             {
2141                 orgobj.aco = true;
2142             }
2143         }
2144     }
2145
2146     if (entity.OrganisationType != null)
2147     {
2148         var org = entity.OrganisationType.Split(',');
2149
2150         foreach (var per in org)
2151         {
2152             if (per == "1")
2153             {
2154                 orgobj.hedis = true;
2155             }
2156             if (per == "2")
2157             {
2158                 orgobj.mra = true;
2159             }
2160             if (per == "3")
2161             {
2162                 orgobj.aco = true;
2163             }
2164         }
2165     }
2166
2167     if (entity.OrganisationType != null)
2168     {
2169         var org = entity.OrganisationType.Split(',');
2170
2171         foreach (var per in org)
2172         {
2173             if (per == "1")
2174             {
2175                 orgobj.hedis = true;
2176             }
2177             if (per == "2")
2178             {
2179                 orgobj.mra = true;
2180             }
2181             if (per == "3")
2182             {
2183                 orgobj.aco = true;
2184             }
2185         }
2186     }
2187
2188     if (entity.OrganisationType != null)
2189     {
2190         var org = entity.OrganisationType.Split(',');
2191
2192         foreach (var per in org)
2193         {
2194             if (per == "1")
2195             {
2196                 orgobj.hedis = true;
2197             }
2198             if (per == "2")
2199             {
2200                 orgobj.mra = true;
2201             }
2202             if (per == "3")
2203             {
2204                 orgobj.aco = true;
2205             }
2206         }
2207     }
2208
2209     if (entity.OrganisationType != null)
2210     {
2211         var org = entity.OrganisationType.Split(',');
2212
2213         foreach (var per in org)
2214         {
2215             if (per == "1")
2216             {
2217                 orgobj.hedis = true;
2218             }
2219             if (per == "2")
2220             {
2221                 orgobj.mra = true;
2222             }
2223             if (per == "3")
2224             {
2225                 orgobj.aco = true;
2226             }
2227         }
2228     }
2229
2230     if (entity.OrganisationType != null)
2231     {
2232         var org = entity.OrganisationType.Split(',');
2233
2234         foreach (var per in org)
2235         {
2236             if (per == "1")
2237             {
2238                 orgobj.hedis = true;
2239             }
2240             if (per == "2")
2241             {
2242                 orgobj.mra = true;
2243             }
2244             if (per == "3")
2245             {
2246                 orgobj.aco = true;
2247             }
2248         }
2249     }
2250
2251     if (entity.OrganisationType != null)
2252     {
2253         var org = entity.OrganisationType.Split(',');
2254
2255         foreach (var per in org)
2256         {
2257             if (per == "1")
2258             {
2259                 orgobj.hedis = true;
2260             }
2261             if (per == "2")
2262             {
2263                 orgobj.mra = true;
2264             }
2265             if (per == "3")
2266             {
2267                 orgobj.aco = true;
2268             }
2269         }
2270     }
2271
2272     if (entity.OrganisationType != null)
2273     {
2274         var org = entity.OrganisationType.Split(',');
2275
2276         foreach (var per in org)
2277         {
2278             if (per == "1")
2279             {
2280                 orgobj.hedis = true;
2281             }
2282             if (per == "2")
2283             {
2284                 orgobj.mra = true;
2285             }
2286             if (per == "3")
2287             {
2288                 orgobj.aco = true;
2289             }
2290         }
2291     }
2292
2293     if (entity.OrganisationType != null)
2294     {
2295         var org = entity.OrganisationType.Split(',');
2296
2297         foreach (var per in org)
2298         {
2299             if (per == "1")
2300             {
2301                 orgobj.hedis = true;
2302             }
2303             if (per == "2")
23
```

```
UQP.Controllers.User_DetailsController
ChangePassword(FirtUserLoginInput firstuser)

        orgobj.aco = true;
    }

}

objuser.organisations = orgobj;

var acl = (from a in db.ACLs.AsNoTracking()
join ar in db.AC_L_Role on a.id equals ar.AC_ID
where ar.Role_ID == entity.User_Role_ID
select new { a.id, ar.AddACL, ar.EditACL, ar.ViewACL, a.Functions }).ToList();
var role = from a in db.User_Role.AsNoTracking()
where a.User_Role_ID == entity.User_Role_ID
select new { a.User_Role_Name };

objuser.email = entity.User_Email;
objuser.DefaultIPA = entity.DefaultIPA;

objuser.userid = entity.User_ID;
objuser.Isfirsttime = entity.FirstTimeUser;
objuser.name = entity.User_FirstName + " " + (entity.User_MiddleName == "" ? "" : entity.User_MiddleName + " ") + entity.User_LastName;
objuser.isuservalid = true;
objuser.roleid = entity.User_Role_ID;
objuser.passwordmessage = entity.Message;

if (entity.User_Provider_ID != null)
{
    objuser.providerid = entity.User_Provider_ID;
}
```

```
from: Debug
Assembly (CLR v4.0.30319; /LM/W3SVC/2/ROOT-1-13230222775672406): Loaded 'C:\Windows\Microsoft.NET\assembly\GAC_MSIL\System.Windows.Forms\v4.0_4.0.0.0_b77a5c561de5bb08\System.Windows.Forms.dll' (Version: 4.0.0.0, Culture: neutral, PublicKeyToken: b77a5c561de5bb08)
Assembly (CLR v4.0.30319; /LM/W3SVC/2/ROOT-1-13230222775672406): Loaded 'C:\Windows\Microsoft.NET\assembly\GAC_MSIL\System.Data.OracleClient\v4.0_4.0.0.0_b77a5c561de5bb08\System.Data.OracleClient.dll' (Version: 4.0.0.0, Culture: neutral, PublicKeyToken: b77a5c561de5bb08)
```

- Microsoft Visual Studio

View Project Build Debug Team Tools Test Analyze Window Help

Debug Any CPU IIS Express (Google Chrome)

User\_DetailsController.cs\* GetProfile()

```
objuser.email = entity.user_email;
objuser.DefaultIPA = entity.DefaultIPA;

objuser.userid = entity.User_ID;
objuser.Isfirsttime = entity.FirstTimeUser;
objuser.name = entity.User_FirstName + " " + (entity.User_MiddleName == "" ? "" : entity.User_MiddleName + " ") + entity.User_LastName;
objuser.isuservalid = true;
objuser.roleid = entity.User_Role_ID;
objuser.passwordmessage = entity.Message;

if (entity.User_Provider_ID != null)
{
    objuser.providerid = entity.User_Provider_ID;
    //objuser.officestaff_providername =
    var x = (from p in db.ProviderDetailsTestings.AsNoTracking() where p.providerid == objuser.providerid && p.visible == true select new
    {
        p.Physician_first_name, p.Physician_last_name
    }).FirstOrDefault();

    if (x != null)
    {
        if (x.Physician_first_name != null || x.Physician_last_name != null)
        {
            objuser.officestaff_providername = x.Physician_last_name + " " + x.Physician_first_name;
        }
    }
    else
    {
        errormessage msg = new errormessage();
    }
}
```

ft Visual Studio

Project Build Debug Team Tools Test Analyze Window Help

Debug Any CPU IIS Express (Google Chrome) GetProfile()

User\_DetailsController.cs\* UQP.Controllers.User\_DetailsController

```
objuser.providerid = entity.user_provider_id;
//objuser.officestaff_providername =
var x = (from p in db.ProviderDetailsTestings.AsNoTracking() where p.providerid == objuser.providerid && p.visible == true select new
{ p.Physician_first_name, p.Physician_last_name }).FirstOrDefault();

if (x != null)
{
    if (x.Physician_first_name != null || x.Physician_last_name != null)
    {
        objuser.officestaff_providername = x.Physician_last_name + " " + x.Physician_first_name;
    }
}
else
{
    errormessage msg = new errormessage();
    msg.error = "invalid_grant";
    msg.error_description = "The associated provider with this account: " + entity.User_Email + " is no longer active. Please contact" +
        "| administrator";
    return Request.CreateResponse(HttpStatusCode.BadRequest, msg);
}

if (entity.User_HealthPlan_ID != null)
{
    objuser.healthplanid = entity.User_HealthPlan_ID;
    objuser.Healthplan = db.HealthPlans.Where(x => x.HealthPlan_ID == entity.User_HealthPlan_ID)
        .Select(x => x.HealthPlan_Name).FirstOrDefault();
    ipalist.Add(new CustomIPA { IPA_ID = 0, IPA_Name = null, ipacode = null });
}

if (entity.User_IPA_ID != null)
{
```

The screenshot shows the Microsoft Visual Studio IDE interface. The title bar indicates the project is 'Debug Any CPU' and the application is running in 'IIS Express (Google Chrome)'. The current file is 'User\_DetailsController.cs'. The code is as follows:

```
1  if (entity.User_HealthPlan_ID != null)
2  {
3      objuser.healthplanid = entity.User_HealthPlan_ID;
4      objuser.Healthplan = db.HealthPlans.Where(x => x.HealthPlan_ID == entity.User_HealthPlan_ID)
5          .Select(x => x.HealthPlan_Name).FirstOrDefault();
6      ipalist.Add(new CustomIPA { IPA_ID = 0, IPA_Name = null, ipacode = null });
7  }
8
9  if (entity.User_IPA_ID != null)
10 {
11     objuser.ipaid = null; //entity.User_IPA_ID;
12 }
13
14 if (entity.Subscriber_id != null)
15 {
16     objuser.subscriberid = entity.Subscriber_id;
17     DateTime now = DateTime.Now;
18     var startDate = new DateTime(now.Year, now.Month, 1);
19     var endDate = startDate.AddMonths(1).AddDays(-1);
20
21     var fromyear1 = endDate.Year;
22     var fromyear2 = fromyear1 - 1;
23     var fromyear3 = fromyear1 - 2;
24     int[] ids2 = { 1, 3 };
25     int[] ids1 = { 5, 9, 11, 10, 12, 13, 14, 15, 7, 4, 6, 18, 19 };
26     int[] id3 = { 2 };
27     int[] id4 = { 8 };
28
29
30     List<UVisitReport> lvisit = new List<UVisitReport>();
31 }
```

```
List<LVisitReport> lvisit = new List<LVisitReport>();  
  
//var startdate = objdate.startdate;  
//var enddate = objdate.enddate;  
  
  
if (endDate.Month < 7)  
{  
    ids2 = ids2.Concat(id4).ToArray();  
}  
else  
{  
    ids1 = ids1.Concat(id4).ToArray();  
}  
  
var result = db.FinalDStats.AsNoTracking().Where(p => (p.RosterSubscriberID == entity.Subscriber_id) &&  
    (p.DateOfServiceFrom) <= endDate  
);  
  
var query2 = (result.Where(category => ids2.Contains((int)category.MEASURE_ID)  
    && category.FromYear.Year == fromyear2).GroupBy(category => category.Measure_Name
```

Microsoft Visual Studio

View Project Build Debug Team Tools Test Analyze Window Help

192.61.99.225

User\_DetailsController.cs

```
var query2 = (result.Where(category => ids2.Contains((int)category.MEASURE_ID)
    && category.FromYear.Year == fromyear2).GroupBy(category => category.Measure_Name
    )
    .Select(grouped => new
    {
        totalgap = grouped.Select(x => x.RosterSubscriberID).Distinct().Count(),
        measname = grouped.Key,
        measureid = grouped.Select(x => x.MEASURE_ID).FirstOrDefault(),
    }));
}

var query1 = (result.Where(category => ids1.Contains((int)category.MEASURE_ID)
    && category.FromYear.Year == fromyear1).GroupBy(category => category.Measure_Name
    )
    .Select(grouped => new
    {
        totalgap = grouped.Select(x => x.RosterSubscriberID).Distinct().Count(),
        measname = grouped.Key,
        measureid = grouped.Select(x => x.MEASURE_ID).FirstOrDefault(),
    }));
}

var query3 = (result.Where(category => category.MEASURE_ID == 2
    && category.FromYear.Year == fromyear3).GroupBy(category => category.Measure_Name
    )
    .Select(grouped => new
    {
        totalgap = grouped.Select(x => x.RosterSubscriberID).Distinct().Count(),
        measname = grouped.Key,
        measureid = grouped.Select(x => x.MEASURE_ID).FirstOrDefault(),
    }));
}
```

UQP - Microsoft Visual Studio

Edit View Project Build Debug Team Tools Test Analyze Window Help

Source Control Explorer User\_DetailsController.cs\*

UQP

```
1199     && Category.FromYear.Year == FromYear).GroupBy(category => category.Measure_Name)
1200     )
1201     .Select(grouped => new
1202     {
1203         totalgap = grouped.Select(x => x.RosterSubscriberID).Distinct().Count(),
1204         measname = grouped.Key,
1205         measureid = grouped.Select(x => x.MEASURE_ID).FirstOrDefault(),
1206     });
1207
1208
1209
1210
1211
1212     var query = query1.Concat(query2).Concat(query3);
1213
1214
1215
1216
1217     List<long> gaps = new List<long>();
1218     foreach (var q in query)
1219     {
1220         long l = q.measureid.Value;
1221         gaps.Add(l);
1222     }
1223     objuser.gaps = gaps.ToArray();
1224
1225
1226
1227
1228     if (entity.User_Type_ID == 3)
```

Output

```
Show output from: Debug
```

IISExpress.exe' (CLR v4.0.30319: /LM/W3SVC/2/ROOT-1-13230222775672406): Loaded 'C:\Windows\Microsoft.NET\Framework\assembly\GAC\_MSIL\System.Windows.Forms\v4.0\_4.0.0.0\_\_b77a5c561934e08'

Scanned with CamScanner

```
1223
1224
1225
1226
1227
1228     if (entity.User_Type_ID == 3)
1229     {
1230         var officetype = (from of in db.OfficeStaff_AssignedProvider.AsNoTracking()
1231                             where of.User_Id == entity.User_ID
1232                             select new { of.User_ProviderId });
1233         List<ExtendedOfficeStaff> ofclist = new List<ExtendedOfficeStaff>();
1234         foreach (var per in officetype)
1235         {
1236             ExtendedOfficeStaff of = new ExtendedOfficeStaff();
1237             of.User_ProviderId = per.User_ProviderId;
1238             var x = (from p in db.ProviderDetailsTestings.AsNoTracking()
1239                         where p.providerid == of.User_ProviderId && p.visible == true
1240                         select new { p.Physician_first_name, p.Physician_last_name }).FirstOrDefault();
1241             if (x.Physician_first_name != null || x.Physician_last_name != null)
1242             {
1243                 of.ProviderName = x.Physician_last_name + " " + x.Physician_first_name;
1244             }
1245             ofclist.Add(of);
1246         }
1247         objuser.officetype = ofclist;
1248         objuser.roleid = 35;
1249     }
1250
1251     List<long> mraids = new List<long>();
1252     List<long> hedisids = new List<long>();
```



Source Control Explorer      User\_DetailsController.cs\*      UOP

```
1262 // aclobj.acl10 = per.lu;
1263 long abcd = per.id;
1264 if (objuser.organisations.hedis == null && hedisid.Contains(per.id))
1265 {
1266     aclobj.add = false;
1267     aclobj.edit = false;
1268     aclobj.view = false;
1269     aclobj.func = per.Functions;
1270 }
1271 if (objuser.organisations.mra == null && mraid.Contains(per.id))
1272 {
1273     aclobj.add = false;
1274     aclobj.edit = false;
1275     aclobj.view = false;
1276     aclobj.func = per.Functions;
1277 }
1278 if ((objuser.organisations.mra == null && !mraid.Contains(per.id)) ||
1279     (objuser.organisations.hedis == null && !hedisid.Contains(per.id)) ||
1280     ((objuser.organisations.mra == true && objuser.organisations.hedis == true)))
1281 {
1282
1283     aclobj.add = per.AddACL;
1284     aclobj.edit = per.EditACL;
1285     aclobj.view = per.ViewACL;
1286     aclobj.func = per.Functions;
1287 }
1288 var x = db.ACOLUMN_By_Role.AsNoTracking().Where(e => e.AC_Id == per.id && e.Role_id == objuser.roleid);
1289 int count = x.Count();
1290 if (count == 0)
1291 {
1292 }
```

Output

Show output from: Debug



```
UQP
```

```
1301 | } UQP.Controllers.User_DetailsController
1302 |     columns.aclid = value.ACL_id.Value;
1303 |     columns.column_name = value.Column_name;
1304 |     columns.viewstatus = value.View_Status.Value;
1305 |     Columnlist.Add(columns);
1306 | }
1307 | aclobj.Columnlist = Columnlist;
1308 | }
1309 | else if (count != 0)
1310 | {
1311 |     foreach (var value in x)
1312 |     {
1313 |         acl_columns columns = new acl_columns();
1314 |         columns.aclid = value.ACL_id.Value;
1315 |         columns.column_name = value.Column_name.Column_name;
1316 |         columns.viewstatus = value.View_Status.Value;
1317 |         Columnlist.Add(columns);
1318 |     }
1319 |     aclobj.Columnlist = Columnlist;
1320 | }
1321 | acclist.Add(aclobj);
1322 | }
1323 | foreach (var per in role)
1324 | {
1325 |     objuser.rolename = per.User_Role_Name;
1326 |     objuser.loginrole = per.User_Role_Name;
1327 | }
1328 |
1329 | objuser.acl = acclist;
1330 | objuser.loginid = Common.TATOTSA/LogintIDN;
```

Output

```
Show output from: Debug
```

```
Ilisexpress.exe' (CLR v4.0.30319: /LM/W3SVC/2/ROOT-1-13230222775672406): Loaded 'C:\Windows\Microsoft.NET\Assembly\GAC_MSIL\System.Windows.Forms\v4.0_4.0.0.0__b77a5c561934e089';
'Ilisexpress.exe' (CLR v4.0.30319: /LM/W3SVC/2/ROOT-1-13230222775672406): Loaded 'C:\Windows\Microsoft.NET\Assembly\GAC_32\System.Data.OracleClient\v4.0_4.0.0.0__b77a5c561934e089';
'Ilisexpress.exe' (CLR v4.0.30319: /LM/W3SVC/2/ROOT-1-13230222775672406): Loaded 'EntityframeworkDynamicProxies-UQP';
'Ilisexpress.exe' (CLR v4.0.30319: /LM/W3SVC/2/ROOT-1-13230222775672406): Loaded 'C:\Windows\Microsoft.NET\Assembly\GAC_MSIL\System.Dynamic\v4.0_4.0.0.0__b03f5f7f15d804e0';
```

```
1325
1326     objuser.rolename = per.User_Role_Name;
1327     objuser.loginrole = per.User_Role_Name;
1328 }
1329
1330     objuser.acl = acclist;
1331     objuser.loginid = Convert.ToInt64(loginid);
1332     objuser.HedisTeamId = entity.HedisTeamId;
1333     objuser.MRATeamId = entity.MRATeamId;
1334     objuser.MRATeam = entity.MRATeam;
1335     objuser.HedisTeam = entity.HedisTeam;
1336
1337     Theme tobj = new Theme
1338     {
1339         colourcode = entity.Colour_Code,
1340         theme = entity.Theme,
1341         user_id = entity.User_ID
1342     };
1343     if (entity.User_Role_ID == 1065)
1344     {
1345         objuser.roleid = 35;
1346     }
1347     objuser.themeobj = tobj;
1348     objuser.ipadetails = ipalist;
1349     return Request.CreateResponse(HttpStatusCode.OK, objuser);
1350
1351 }
1352
1353
1354 }
```

```
User_DetailsController.cs*  ▾ X UQP.Controllers.User_DetailsController GetProfile()
1346
1347
1348
1349
1350
1351
1352
1353
1354
1355
1356
1357     else
1358     {
1359         objuser.isuservalid = false;
1360         return Request.CreateResponse(HttpStatusCode.BadRequest, objuser);
1361     }
1362     //}
1363 }
1364 catch (Exception e)
1365 {
1366     return Request.CreateResponse(HttpStatusCode.InternalServerError, e.InnerException);
1367 }
1368 }
1369
1370
1371
1372
1373
1374
1375
1376
1377
1378
1379
1380
1381
1382
1383
1384
1385
1386
1387
1388
1389
1390
1391
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
1428
1429
1430
1431
1432
1433
1434
1435
1436
1437
1438
1439
1440
1441
1442
1443
1444
1445
1446
1447
1448
1449
1450
1451
1452
1453
1454
1455
1456
1457
1458
1459
1460
1461
1462
1463
1464
1465
1466
1467
1468
1469
1470
1471
1472
1473
1474
1475
1476
1477
1478
1479
1480
1481
1482
1483
1484
1485
1486
1487
1488
1489
1490
1491
1492
1493
1494
1495
1496
1497
1498
1499
1500
1501
1502
1503
1504
1505
1506
1507
1508
1509
1510
1511
1512
1513
1514
1515
1516
1517
1518
1519
1520
1521
1522
1523
1524
1525
1526
1527
1528
1529
1530
1531
1532
1533
1534
1535
1536
1537
1538
1539
1540
1541
1542
1543
1544
1545
1546
1547
1548
1549
1550
1551
1552
1553
1554
1555
1556
1557
1558
1559
1560
1561
1562
1563
1564
1565
1566
1567
1568
1569
1570
1571
1572
1573
1574
1575
1576
1577
1578
1579
1580
1581
1582
1583
1584
1585
1586
1587
1588
1589
1590
1591
1592
1593
1594
1595
1596
1597
1598
1599
1600
1601
1602
1603
1604
1605
1606
1607
1608
1609
1610
1611
1612
1613
1614
1615
1616
1617
1618
1619
1620
1621
1622
1623
1624
1625
1626
1627
1628
1629
1630
1631
1632
1633
1634
1635
1636
1637
1638
1639
1640
1641
1642
1643
1644
1645
1646
1647
1648
1649
1650
1651
1652
1653
1654
1655
1656
1657
1658
1659
1660
1661
1662
1663
1664
1665
1666
1667
1668
1669
1670
1671
1672
1673
1674
1675
1676
1677
1678
1679
1680
1681
1682
1683
1684
1685
1686
1687
1688
1689
1690
1691
1692
1693
1694
1695
1696
1697
1698
1699
1700
1701
1702
1703
1704
1705
1706
1707
1708
1709
1710
1711
1712
1713
1714
1715
1716
1717
1718
1719
1720
1721
1722
1723
1724
1725
1726
1727
1728
1729
1730
1731
1732
1733
1734
1735
1736
1737
1738
1739
1740
1741
1742
1743
1744
1745
1746
1747
1748
1749
1750
1751
1752
1753
1754
1755
1756
1757
1758
1759
1760
1761
1762
1763
1764
1765
1766
1767
1768
1769
1770
1771
1772
1773
1774
1775
1776
1777
1778
1779
1780
1781
1782
1783
1784
1785
1786
1787
1788
1789
1790
1791
1792
1793
1794
1795
1796
1797
1798
1799
1800
1801
1802
1803
1804
1805
1806
1807
1808
1809
1810
1811
1812
1813
1814
1815
1816
1817
1818
1819
1820
1821
1822
1823
1824
1825
1826
1827
1828
1829
1830
1831
1832
1833
1834
1835
1836
1837
1838
1839
1840
1841
1842
1843
1844
1845
1846
1847
1848
1849
1850
1851
1852
1853
1854
1855
1856
1857
1858
1859
1860
1861
1862
1863
1864
1865
1866
1867
1868
1869
1870
1871
1872
1873
1874
1875
1876
1877
1878
1879
1880
1881
1882
1883
1884
1885
1886
1887
1888
1889
1890
1891
1892
1893
1894
1895
1896
1897
1898
1899
1900
1901
1902
1903
1904
1905
1906
1907
1908
1909
1910
1911
1912
1913
1914
1915
1916
1917
1918
1919
1920
1921
1922
1923
1924
1925
1926
1927
1928
1929
1930
1931
1932
1933
1934
1935
1936
1937
1938
1939
1940
1941
1942
1943
1944
1945
1946
1947
1948
1949
1950
1951
1952
1953
1954
1955
1956
1957
1958
1959
1960
1961
1962
1963
1964
1965
1966
1967
1968
1969
1970
1971
1972
1973
1974
1975
1976
1977
1978
1979
1980
1981
1982
1983
1984
1985
1986
1987
1988
1989
1990
1991
1992
1993
1994
1995
1996
1997
1998
1999
1999
2000
2000
```

Source Control Explorer    User\_DetailsController.cs\*  X  UOP.Controllers.User\_DetailsController

Debug Any CPU  ▶  VS Express (Google Chrome)  C  M  B  D  E  F  G  H  I  J  K  L  N  O  P  R  S  T  U  V  W  X  Y  Z  GetProfile()

```
1439 //     objuser.isUserValid = false;
1440 //     return objuser;
1441 // }
1442 //}
1443
1444
1445 2 references
1446 private string GenerateRandomPWD(int length)
1447 {
1448
1449     const string valid = "abcdefghijklmnopqrstuvwxyzABCDEFGHIJKLMNOPQRSTUVWXYZ1234567890";
1450     StringBuilder res = new StringBuilder();
1451     using (RNGCryptoServiceProvider rng = new RNGCryptoServiceProvider())
1452     {
1453         byte[] uintBuffer = new byte[sizeof(uint)];
1454
1455         while (length-- > 0)
1456         {
1457             rng.GetBytes(uintBuffer);
1458             uint num = BitConverter.ToInt32(uintBuffer, 0);
1459             res.Append(valid[(int)(num % (uint)valid.Length)]);
1460         }
1461     }
1462
1463     return res.ToString();
1464
1465
1466 }
1467
1468 }
```

Output

Source Control Explorer    User\_DetailsController.cs\*    X    UQP    UQP.Controllers.User\_DetailsController    GetProfile()

```
1463     return res.ToString();
1464 
1465 }
1466 
1467 [HttpPost]
1468 [Route("api/User_Details/ProviderEmail")]
1469 Preferences
1470 public List<CustomProviderDetails> PoivderEmail(EmailString useremail)
1471 {
1472     List<CustomProviderDetails> pvrdrlist = new List<CustomProviderDetails>();
1473     var query = (from ab in db.ProviderDetailsTestings.AsNoTracking()
1474                 where ab.emailid.Equals(useremail.EmailID) && ab.visible == true
1475                 select new
1476                 {
1477                     ab.Physician_first_name,
1478                     //ab.Provider_MidName,
1479                     ab.Physician_last_name,
1480                     // ab.p,
1481                     //ab.Provider_Description,
1482                     ab.emailid,
1483                     ab.providerid,
1484                     ab.Specialty
1485 
1486                 });
1487     pvrdrlist = query.ToList();
1488     foreach (var per in query)
1489     {
1490 
1491     }
```

Output

Show output from: Debug

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

- Menu Bar:** File, Edit, View, Project, Build, Debug, Team, Tools, Test, Analyze, Window, Help.
- Toolbar:** Standard toolbar with icons for New, Open, Save, Print, etc.
- Status Bar:** Shows "Scanned with CamScanner".

The code editor displays the `User_DetailsController.cs` file under the project `UQP`. The current tab is `UQP.Controllers.User_DetailsController`. The code is as follows:

```
1486
1487
1488
1489     }
1490     ).ToList();
1491     foreach (var per in query)
1492     {
1493         CustomProviderDetails obj = new CustomProviderDetails();
1494         obj.FirstName = per.Physician_first_name;
1495         obj.MidName = null;
1496         obj.LastName = per.Physician_last_name;
1497         obj.email = per.emailid;
1498         obj.mobNo = null;
1499         obj.providerno = per.providerid;
1500         obj.level = per.Specialty;
1501
1502         pvrdrlist.Add(obj);
1503     }
1504
1505     return pvrdrlist;
1506     //return db.User_Role;
1507 }
1508
1509 // POST: api/User_Details
1510 // [ResponseType(typeof(User_Details))]
1511 [HttpPost]
1512 [Route("api/user_details/createuser")]
1513 public HttpResponseMessage PostUser_Details(User_Details user_Details)
1514 {
```

Source Control Explorer      User\_DetailsController.cs\*      Debug    Any CPU    IIS Express (Google Chrome)    GetProfile()

UQP

```
1510 // [ResponseType(typeof(User_Details))]  
1511 [HttpPost]  
1512 [Route("api/user_details/createuser")]  
1513 public HttpResponseMessage PostUser_Details(User_Details user_Details)  
{  
    try  
    {  
        //using (var db = new UQPEntities("UQPEntities"))  
        //{
        if (!ModelState.IsValid)  
        {  
            return Request.CreateResponse(HttpStatusCode.BadRequest, "");  
        }  
        var pwd = GenerateRandomPWD(7);  
        SHA1CryptoServiceProvider encrypt = new SHA1CryptoServiceProvider();  
        byte[] hashValueMessage = encrypt.ComputeHash(Encoding.Default.GetBytes(pwd));  
        string hexOfValueMessage = BitConverter.ToString(hashValueMessage);  
        User_Details user = new User_Details();  
        //{
        user.OfType = user_Details.OfType;  
        //user.PrimaryFacilityId = user_Details.PrimaryFacilityId;  
        user.User_Address = user_Details.User_Address;  
        user.User_DOB = user_Details.User_DOB;  
        user.User_Email = user_Details.User_Email;  
        user.User_FirstName = user_Details.User_FirstName;  
        user.User_Gender = user_Details.User_Gender;  
        user.User_HealthPlan_ID = user_Details.User_HealthPlan_ID;  
        user.User_IPA_ID = user_Details.User_IPA_ID;  
        user.DefaultIPA = user_Details.DefaultIPA;  
    }  
}
```

100 %

Output

Source Control Explorer      User\_DetailsController.cs\*      UOP

User\_DetailsController.cs

UOP.Controllers.User\_DetailsController

GetProfile()

```
1533     user.User_Email = user_Details.User_Email;
1534     user.User_FirstName = user_Details.User_FirstName;
1535     user.User_Gender = user_Details.User_Gender;
1536     user.User_HealthPlan_ID = user_Details.User_HealthPlan_ID;
1537     user.User_IPA_ID = user_Details.User_IPA_ID;
1538     user.DefaultIPA = user_Details.DefaultIPA;
1539     if ((user_Details.User_IPA_ID == ""))
1540     {
1541         user.User_IPA_ID = null;
1542     }
1543     if ((user_Details.User_Role_ID == 35))
1544     {
1545         var query = (from p in db.ProviderDetailTesting
1546                      join a in db.ProviderTesting_Address
1547                      on p.providerid equals a.providerid
1548                      where p.providerid == user_Details.Provider_ID && a.isdefault == true && p.visible == true
1549                      select a).FirstOrDefault();
1550
1551     if (query.recordid != user_Details.PrimaryFacilityId)
1552     {
1553         query.isdefault = false;
1554         var newaddress = db.ProviderTesting_Address.Where(e => e.recordid == user_Details.PrimaryFacilityId).FirstOrDefault();
1555         if (newaddress != null)
1556         {
1557             newaddress.isdefault = true;
1558         }
1559         db.SaveChanges();
1560     }
1561 }
1562 }
```

Output

Show output from: Debug

A screenshot of Microsoft Visual Studio showing the code for `User_DetailsController.cs`. The code is part of a method named `PostUser_Details`, which takes a parameter `User_Details user_Details`. The code handles updating a user's details, including their address, provider, and other personal information. It uses Entity Framework to interact with a database named `providerTesting`.

```
1551
1552     if (query.recordid != user_Details.PrimaryFacilityId)
1553     {
1554         query.isdefault = false;
1555         var newaddress = db.providerTesting.address.Where(e => e.recordid == user_Details.PrimaryFacilityId).FirstOrDefault();
1556         if (newaddress != null)
1557         {
1558             newaddress.isdefault = true;
1559         }
1560         db.SaveChanges();
1561     }
1562
1563
1564
1565     user.User_LastName = user_Details.User_LastName;
1566     user.User_MiddleName = user_Details.User_MiddleName;
1567     user.User_MobileNumber = user_Details.User_MobileNumber;
1568     user.User_Provider_ID = (user_Details.OfficeStaff_AssignedProvider.Count == 0) ? user_Details.User_Provider_ID : [
1569         user_Details.OfficeStaff_AssignedProvider.FirstOrDefault().Id];
1570     user.User_Role_ID = user_Details.User_Role_ID;
1571     user.User_Type_ID = user_Details.User_Type_ID;
1572     user.User_WorkNumber = user_Details.User_WorkNumber;
1573     //CustomFacility= user_Details.
1574     user.User_Member_ID = user_Details.User_Member_ID;
1575     user.Status = user_Details.Status;
1576     user.title = user_Details.title;
1577     user.User_Password = hexOfValueMessage;
1578     user.FirstTimeUser = true;
1579     user.OrganisationType = user_Details.OrganisationType;
1580     user.HedisTeam = user_Details.HedisTeam;
```

Debug Any CPU

User\_DetailsController.cs\* UOP.Controllers.User\_DetailsController PostUser\_Details(User\_Details user\_Detal...

```
1569     user_Details.OfficeStaff.AssignedProvider.FirstOrDefault().Id;
1570     user.User_Role_ID = user_Details.User_Role_ID;
1571     user.User_Type_ID = user_Details.User_Type_ID;
1572     user.User_WorkNumber = user_Details.User_WorkNumber;
1573     //CustomFacility= user_Details.
1574     user.User_Member_ID = user_Details.User_Member_ID;
1575     user.Status = user_Details.Status;
1576     user.title = user_Details.title;
1577     user.User_Password = hexOfValueMessage;
1578     user.FirstTimeUser = true;
1579     user.OrganisationType = user_Details.OrganisationType;
1580     user.HedisTeam = user_Details.HedisTeam;
1581     user.MRATeam = user_Details.MRATeam;
1582     user.ModifiedPasswordDate = DateTime.Now;
1583     user.CreatedDate = DateTime.Now;
1584 
1585     var name = user_Details.User_LastName;
1586     var title = user_Details.title;
1587     user.HedisTeamId = user_Details.HedisTeamId;
1588     user.MRATeamId = user_Details.MRATeamId;
1589     db.User_Details.Add(user);
1590     foreach (var q in user_Details.OfficeStaff_AssignedProvider.ToList())
1591     {
1592         OfficeStaff_AssignedProvider ofc = new OfficeStaff_AssignedProvider();
1593         ofc.User_Id = user_Details.User_ID;
1594         ofc.User_ProviderId = q.Id;
1595         db.OfficeStaff_AssignedProvider.Add(ofc);
1596         //
1597     }
}
```

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

- Title Bar:** Shows "User\_DetailsController.cs" as the active file.
- Code Editor:** Displays C# code for the "User\_DetailsController" class. The code handles saving user details to a database and sending an email if the configuration setting "email" contains "true".
- Status Bar:** Shows "100 %".
- Output Window:** Shows the message: "The thread 0x008 has exited with code 0 (0x0). The program 'f18721\_iisexpress.exe' has exited with code -1 (0xffffffff)."



Source Control Explorer    User\_DetailsController.cs\*    X

UQP    UQP.Controllers.User\_DetailsController

PostUser\_Details(User\_Details user\_Det)

```
1641
1642
1643
1644
1645
1646     [AllowAnonymous]
1647     [HttpPost]
1648     [Route("api/User_Details/ResetPasswordToken")]
1649     public HttpResponseMessage ResetPasswordToken(UrlString url)
1650     {
1651         //using (var db = new UQPEntities("UQPEntities"))
1652         //{
1653             string email_message;
1654             var entity = db.User_Details.FirstOrDefault(e => e.User_Email == url.email);
1655             if (entity == null)
1656             {
1657                 return Request.CreateResponse(HttpStatusCode.BadRequest, "You are not registered with us !");
1658             }
1659             else
1660             {
1661                 ResetPassword objpwd = new ResetPassword
1662                 {
1663                     Email = url.email
1664                 };
1665                 Guid obj = Guid.NewGuid();
1666                 objpwd.Email = url.email;
1667                 objpwd.Token = obj.ToString();
1668                 objpwd.PasswordReset = false;
1669                 db.ResetPasswords.Add(objpwd);
1670                 db.SaveChanges();
1671             }
1672         //}
1673     }
```

File Edit View Build Debug Team Tools Test Analyze Window Help

User\_DetailsController.cs

UOP.Controllers.User\_DetailsController

ResetPasswordToken(UrlString url)

```
Guid obj = Guid.NewGuid();
objpwd.Email = url.Email;
objpwd.Token = obj.ToString();
objpwd.PasswordReset = false;
db.ResetPasswords.Add(objpwd);
db.SaveChanges();

var linkpath = url.Url + "" + obj.ToString();
processAuthEmail obj2 = new processAuthEmail();
var EmailContent1 = (from a in db.NewEmailContents
                     where a.Id == 4
                     select new
                     {
                         a.EmailSubject,
                         a.EmailBody,
                         a.EmailSignature
                     }).ToList();
foreach (var per in EmailContent1)
{
    string test = "<span style='background-color:#293b6e; height:45px; width:150px; border-color:#293b6e; padding:10px;'>" +
        "<a href=" + linkpath + ">Reset Password</a>";
    //obj2.body = "Hello " + entity.User.FirstName + per.EmailBody + "<table><tbody><tr><td><br><a href=" + linkpath + ">" + linkpath + [</a><br></td><td>&ampnbsp</td></tr></tbody></table>" + per.EmailSignature;
    obj2.body = "Hello " + entity.User.FirstName + per.EmailBody + "<a href=" + linkpath + ">" + test + per.EmailSignature;
    obj2.subject = per.EmailSubject;
}

obj2.replyto = url.Email + ",";
```

Debug Any CPU → IIS Express (Google Chrome) ↗

User\_Controler.cs

```
UOP
1685     +> Reset_Password</span></a>;
1686     //obj2.body = "Hello " + entity.User_FirstName + per.EmailBody + "<table ><tbody><tr><td><br><a href =" + lnkpath + ">" + lnkpath +
1687     //</a><br></td><td>&nbsp;</td></tr></tbody></table>" + per.EmailSignature;
1688     obj2.body = "Hello " + entity.User_FirstName + per.EmailBody + "<a href =" + lnkpath + ">" + test + per.EmailSignature;
1689     obj2.subject = per.EmailSubject;
1690
1691
1692     }
1693     obj2.replyto = url.email + ",";
1694
1695     using (var ProcessAuthController = new ProcessAuthController())
1696     {
1697         email_message = ProcessAuthController.processAuthEmail(obj2);
1698     }
1699
1700
1701
1702     return Request.CreateResponse(HttpStatusCode.OK, email_message);
1703 }
1704 //}
1705
1706
1707 }
1708
1709 [AllowAnonymous]
1710 [HttpGet]
1711 [Route("api/User_Details/ResetPassword/{id}")]
1712
1713 public TokenEmailStatus Resetpassword(Guid? id)
1714 {
1715     //using (var db = new UQPEntities("UQPEntities"))
1716     //
```

```
1706 }  
1707  
1708 [AllowAnonymous]  
1709 [HttpGet]  
1710 [Route("api/User_Details/ResetPassword/{id}")]  
1711 references  
1712 public TokenEmailStatus Resetpassword(Guid? id)  
1713 {  
1714     //using (var db = new UQPEntities("UQPEntities"))  
1715     //{{  
1716     TokenEmailStatus objstatus = new TokenEmailStatus();  
1717     string response = (from ab in db.ResetPasswords  
1718                         where ab.Token == id.ToString()  
1719                         select ab.Email).First().ToString();  
1720     if (response != null)  
1721     {  
1722         objstatus.tokenstatus = "true";  
1723         objstatus.email = response;  
1724         return objstatus;  
1725     }  
1726     else  
1727     {  
1728         objstatus.tokenstatus = "false";  
1729         objstatus.email = string.Empty;  
1730         return objstatus;  
1731     }  
1732 } //}  
1733 }  
1734  
1735 }
```

Debug Any CPU

User\_DetailsController.cs

UQP.Controllers.User\_DetailsController

ResetPasswordToken(UriString url)

```
1729     return objStatus;
1730 }
1731 }
1732 }
1733 }
1734 
```

[AllowAnonymous]

```
1735 [HttpPost]
1736 [Route("api/User_Details/ModifyCredentials")]
1737 
```

public string ModifyCredentials(TokenEmailStatus pwd)

```
1738 {
1739     try
1740     {
1741         //using (var db = new UQPEntities("UQPEntities"))
1742         //{
1743             var entity = db.User_Details.FirstOrDefault(e => e.User_Email == pwd.email);
1744             if (entity != null)
1745             {
1746                 SHA1CryptoServiceProvider encrypt = new SHA1CryptoServiceProvider();
1747                 byte[] hashValueMessage = encrypt.ComputeHash(Encoding.Default.GetBytes(pwd.password));
1748                 string hexOfValueMessage = BitConverter.ToString(hashValueMessage);
1749                 entity.User_Password = hexOfValueMessage;
1750                 //To update modifiedpassword in user_details table,which further,used to check when the password is last changed
1751                 entity.ModifiedPasswordDate = DateTime.Now;
1752                 entity.Message = null;
1753                 entity.FirstTimeUser = false;
1754                 db.SaveChanges();
1755                 if (ConfigSettings.email.Contains("true"))
1756                 {
1757                     processAuthEmail obj1 = new processAuthEmail();
1758                 }
1759             }
1760         }
1761     }
1762 }
```

User DetailsController.cs\* - P X

UQP.Controllers.User.DetailsController

ResetPasswordToken(UriString url)

```
56     db.ProcessEmail();
57     if (ConfigSettings.email.Contains("true"))
58     {
59         processAuthEmail obj1 = new processAuthEmail();
60         var EmailContent = (from a in db.NewEmailContents
61                             where a.Id == 2
62                             select new
63                             {
64                                 a.EmailSubject,
65                                 a.EmailBody,
66                                 a.EmailSignature
67                             }).ToList();
68         foreach (var per in EmailContent)
69         {
70             obj1.body = per.EmailBody;
71             obj1.subject = per.EmailSubject;
72             obj1.signature = per.EmailSignature;
73         }
74         obj1.replyto = pwd.email;
75         using (var ProcessAuthController = new ProcessAuthController())
76         {
77             ProcessAuthController.processAuthEmail(obj1);
78         }
79     }
80     return ("True");
81 }
else
{
}
```

Microsoft Visual Studio

File View Project Build Debug Team Tools Test Analyze Window Help

UQP

User\_DetailsController.cs

UQP.Controllers.User\_DetailsController

ResetPasswordToken

```
1779     }
1780     }
1781     return ("True");
1782 }
1783 else
1784 {
1785
1786     return ("False");
1787 }
1788 }
1789 //}
1790 }
1791 catch (Exception ex)
1792 {
1793     return (ex.Message);
1794 }
1795
1796 }
1797
1798 //}
1799 //protected override void Dispose(bool disposing)
1800 //{
1801 //    if (disposing)
```

UOP

User\_DetailsController.cs

UOP.Controllers.User\_DetailsController

ResetPasswordToken(UrlString url)

```
1804 //protected override void Dispose(bool disposing)
1805 //{
1806     // if (disposing)
1807     //{
1808         db.Dispose();
1809     //}
1810     base.Dispose(disposing);
1811 }
1812
1813 private bool User_DetailsExists(int id)
1814 {
1815     return db.User_Details.Count(e => e.User_ID == id) > 0;
1816 }
1817
1818
1819
1820
1821
1822 [HttpPost]
1823 [Route("api/User_Details/ProviderEmails")]
1824 public List<CustomProviderDetails> ProivderEmails(EmailString useremail)
1825 {
1826     List<CustomProviderDetails> pvrdrlist = new List<CustomProviderDetails>();
1827     dynamic phone = null;
1828     var query = db.ProviderDetailsTestings.Where
1829             (x => x.NPI.Contains(useremail.providerid.ToString()))).Select(ab => new
1830             {
1831
1832
1833
1834
1835
1836
1837
1838
1839
1840
1841
1842
1843
1844
1845
1846
1847
1848
1849
1850
1851
1852
1853
1854
1855
1856
1857
1858
1859
1860
1861
1862
1863
1864
1865
1866
1867
1868
1869
1870
1871
1872
1873
1874
1875
1876
1877
1878
1879
1880
1881
1882
1883
1884
1885
1886
1887
1888
1889
1890
1891
1892
1893
1894
1895
1896
1897
1898
1899
1900
1901
1902
1903
1904
1905
1906
1907
1908
1909
1910
1911
1912
1913
1914
1915
1916
1917
1918
1919
1920
1921
1922
1923
1924
1925
1926
1927
1928
1929
1930
1931
1932
1933
1934
1935
1936
1937
1938
1939
1940
1941
1942
1943
1944
1945
1946
1947
1948
1949
1950
1951
1952
1953
1954
1955
1956
1957
1958
1959
1960
1961
1962
1963
1964
1965
1966
1967
1968
1969
1970
1971
1972
1973
1974
1975
1976
1977
1978
1979
1980
1981
1982
1983
1984
1985
1986
1987
1988
1989
1990
1991
1992
1993
1994
1995
1996
1997
1998
1999
2000
2001
2002
2003
2004
2005
2006
2007
2008
2009
2010
2011
2012
2013
2014
2015
2016
2017
2018
2019
2020
2021
2022
2023
2024
2025
2026
2027
2028
2029
2030
2031
2032
2033
2034
2035
2036
2037
2038
2039
2040
2041
2042
2043
2044
2045
2046
2047
2048
2049
2050
2051
2052
2053
2054
2055
2056
2057
2058
2059
2060
2061
2062
2063
2064
2065
2066
2067
2068
2069
2070
2071
2072
2073
2074
2075
2076
2077
2078
2079
2080
2081
2082
2083
2084
2085
2086
2087
2088
2089
2090
2091
2092
2093
2094
2095
2096
2097
2098
2099
2100
2101
2102
2103
2104
2105
2106
2107
2108
2109
2110
2111
2112
2113
2114
2115
2116
2117
2118
2119
2120
2121
2122
2123
2124
2125
2126
2127
2128
2129
2130
2131
2132
2133
2134
2135
2136
2137
2138
2139
2140
2141
2142
2143
2144
2145
2146
2147
2148
2149
2150
2151
2152
2153
2154
2155
2156
2157
2158
2159
2160
2161
2162
2163
2164
2165
2166
2167
2168
2169
2170
2171
2172
2173
2174
2175
2176
2177
2178
2179
2180
2181
2182
2183
2184
2185
2186
2187
2188
2189
2190
2191
2192
2193
2194
2195
2196
2197
2198
2199
2200
2201
2202
2203
2204
2205
2206
2207
2208
2209
2210
2211
2212
2213
2214
2215
2216
2217
2218
2219
2220
2221
2222
2223
2224
2225
2226
2227
2228
2229
2230
2231
2232
2233
2234
2235
2236
2237
2238
2239
2240
2241
2242
2243
2244
2245
2246
2247
2248
2249
2250
2251
2252
2253
2254
2255
2256
2257
2258
2259
2260
2261
2262
2263
2264
2265
2266
2267
2268
2269
2270
2271
2272
2273
2274
2275
2276
2277
2278
2279
2280
2281
2282
2283
2284
2285
2286
2287
2288
2289
2290
2291
2292
2293
2294
2295
2296
2297
2298
2299
2300
2301
2302
2303
2304
2305
2306
2307
2308
2309
2310
2311
2312
2313
2314
2315
2316
2317
2318
2319
2320
2321
2322
2323
2324
2325
2326
2327
2328
2329
2330
2331
2332
2333
2334
2335
2336
2337
2338
2339
2340
2341
2342
2343
2344
2345
2346
2347
2348
2349
2350
2351
2352
2353
2354
2355
2356
2357
2358
2359
2360
2361
2362
2363
2364
2365
2366
2367
2368
2369
2370
2371
2372
2373
2374
2375
2376
2377
2378
2379
2380
2381
2382
2383
2384
2385
2386
2387
2388
2389
2390
2391
2392
2393
2394
2395
2396
2397
2398
2399
2400
2401
2402
2403
2404
2405
2406
2407
2408
2409
2410
2411
2412
2413
2414
2415
2416
2417
2418
2419
2420
2421
2422
2423
2424
2425
2426
2427
2428
2429
2430
2431
2432
2433
2434
2435
2436
2437
2438
2439
2440
2441
2442
2443
2444
2445
2446
2447
2448
2449
2450
2451
2452
2453
2454
2455
2456
2457
2458
2459
2460
2461
2462
2463
2464
2465
2466
2467
2468
2469
2470
2471
2472
2473
2474
2475
2476
2477
2478
2479
2480
2481
2482
2483
2484
2485
2486
2487
2488
2489
2490
2491
2492
2493
2494
2495
2496
2497
2498
2499
2500
2501
2502
2503
2504
2505
2506
2507
2508
2509
2510
2511
2512
2513
2514
2515
2516
2517
2518
2519
2520
2521
2522
2523
2524
2525
2526
2527
2528
2529
2530
2531
2532
2533
2534
2535
2536
2537
2538
2539
2540
2541
2542
2543
2544
2545
2546
2547
2548
2549
2550
2551
2552
2553
2554
2555
2556
2557
2558
2559
2560
2561
2562
2563
2564
2565
2566
2567
2568
2569
2570
2571
2572
2573
2574
2575
2576
2577
2578
2579
2580
2581
2582
2583
2584
2585
2586
2587
2588
2589
2589
2590
2591
2592
2593
2594
2595
2596
2597
2598
2599
2600
2601
2602
2603
2604
2605
2606
2607
2608
2609
2610
2611
2612
2613
2614
2615
2616
2617
2618
2619
2620
2621
2622
2623
2624
2625
2626
2627
2628
2629
2630
2631
2632
2633
2634
2635
2636
2637
2638
2639
2640
2641
2642
2643
2644
2645
2646
2647
2648
2649
2650
2651
2652
2653
2654
2655
2656
2657
2658
2659
2660
2661
2662
2663
2664
2665
2666
2667
2668
2669
2670
2671
2672
2673
2674
2675
2676
2677
2678
2679
2680
2681
2682
2683
2684
2685
2686
2687
2688
2689
2689
2690
2691
2692
2693
2694
2695
2696
2697
2698
2699
2700
2701
2702
2703
2704
2705
2706
2707
2708
2709
27010
27011
27012
27013
27014
27015
27016
27017
27018
27019
27020
27021
27022
27023
27024
27025
27026
27027
27028
27029
27030
27031
27032
27033
27034
27035
27036
27037
27038
27039
270310
270311
270312
270313
270314
270315
270316
270317
270318
270319
270320
270321
270322
270323
270324
270325
270326
270327
270328
270329
270330
270331
270332
270333
270334
270335
270336
270337
270338
270339
270340
270341
270342
270343
270344
270345
270346
270347
270348
270349
270350
270351
270352
270353
270354
270355
270356
270357
270358
270359
270360
270361
270362
270363
270364
270365
270366
270367
270368
270369
270370
270371
270372
270373
270374
270375
270376
270377
270378
270379
270380
270381
270382
270383
270384
270385
270386
270387
270388
270389
270390
270391
270392
270393
270394
270395
270396
270397
270398
270399
2703100
2703111
2703122
2703133
2703144
2703155
2703166
2703177
2703188
2703199
2703200
2703211
2703222
2703233
2703244
2703255
2703266
2703277
2703288
2703299
2703300
2703311
2703322
2703333
2703344
2703355
2703366
2703377
2703388
2703399
2703400
2703411
2703422
2703433
2703444
2703455
2703466
2703477
2703488
2703499
2703500
2703511
2703522
2703533
2703544
2703555
2703566
2703577
2703588
2703599
2703600
2703611
2703622
2703633
2703644
2703655
2703666
2703677
2703688
2703699
2703700
2703711
2703722
2703733
2703744
2703755
2703766
2703777
2703788
2703799
2703800
2703811
2703822
2703833
2703844
2703855
2703866
2703877
2703888
2703899
2703900
2703911
2703922
2703933
2703944
2703955
2703966
2703977
2703988
2703999
27031000
27031111
27031222
27031333
27031444
27031555
27031666
27031777
27031888
27031999
27032000
27032111
27032222
27032333
27032444
27032555
27032666
27032777
27032888
27032999
27033000
27033111
27033222
27033333
27033444
27033555
27033666
27033777
27033888
27033999
27034000
27034111
27034222
27034333
27034444
27034555
27034666
27034777
27034888
27034999
27035000
27035111
27035222
27035333
27035444
27035555
27035666
27035777
27035888
27035999
27036000
27036111
27036222
27036333
27036444
27036555
27036666
27036777
27036888
27036999
27037000
27037111
27037222
27037333
27037444
27037555
27037666
27037777
27037888
27037999
27038000
27038111
27038222
27038333
27038444
27038555
27038666
27038777
27038888
27038999
27039000
27039111
27039222
27039333
27039444
27039555
27039666
27039777
27039888
27039999
270310000
270311111
270312222
270313333
270314444
270315555
270316666
270317777
270318888
270319999
270320000
270321111
270322222
270323333
270324444
270325555
270326666
270327777
270328888
270329999
270330000
270331111
270332222
270333333
270334444
270335555
270336666
270337777
270338888
270339999
270340000
270341111
270342222
270343333
270344444
270345555
270346666
270347777
270348888
270349999
270350000
270351111
270352222
270353333
270354444
270355555
270356666
270357777
270358888
270359999
270360000
270361111
270362222
270363333
270364444
270365555
270366666
270367777
270368888
270369999
270370000
270371111
270372222
270373333
270374444
270375555
270376666
270377777
270378888
270379999
270380000
270381111
270382222
270383333
270384444
270385555
270386666
270387777
270388888
270389999
270390000
270391111
270392222
270393333
270394444
270395555
270396666
270397777
270398888
270399999
2703100000
2703111111
2703122222
2703133333
2703144444
2703155555
2703166666
2703177777
2703188888
2703199999
2703200000
2703211111
2703222222
2703233333
2703244444
2703255555
2703266666
2703277777
2703288888
2703299999
2703300000
2703311111
2703322222
2703333333
2703344444
2703355555
2703366666
2703377777
2703388888
2703399999
2703400000
2703411111
2703422222
2703433333
2703444444
2703455555
2703466666
2703477777
2703488888
2703499999
2703500000
2703511111
2703522222
2703533333
2703544444
2703555555
2703566666
2703577777
2703588888
2703599999
2703600000
2703611111
2703622222
2703633333
2703644444
2703655555
2703666666
2703677777
2703688888
2703699999
2703700000
2703711111
2703722222
2703733333
2703744444
2703755555
2703766666
2703777777
2703788888
2703799999
2703800000
2703811111
2703822222
2703833333
2703844444
2703855555
2703866666
2703877777
2703888888
2703899999
2703900000
2703911111
2703922222
2703933333
2703944444
2703955555
2703966666
2703977777
2703988888
2703999999
27031000000
27031111111
27031222222
27031333333
27031444444
27031555555
27031666666
27031777777
27031888888
27031999999
27032000000
27032111111
27032222222
27032333333
27032444444
27032555555
27032666666
27032777777
27032888888
27032999999
27033000000
27033111111
27033222222
27033333333
27033444444
27033555555
27033666666
27033777777
27033888888
27033999999
27034000000
27034111111
27034222222
27034333333
27034444444
27034555555
27034666666
27034777777
27034888888
27034999999
27035000000
27035111111
27035222222
27035333333
27035444444
27035555555
27035666666
27035777777
27035888888
27035999999
27036000000
27036111111
27036222222
27036333333
27036444444
27036555555
27036666666
27036777777
27036888888
27036999999
27037000000
27037111111
27037222222
27037333333
27037444444
27037555555
27037666666
27037777777
27037888888
27037999999
27038000000
27038111111
27038222222
27038333333
27038444444
27038555555
27038666666
27038777777
27038888888
27038999999
27039000000
27039111111
27039222222
27039333333
27039444444
27039555555
27039666666
27039777777
27039888888
27039999999
270310000000
270311111111
270312222222
270313333333
270314444444
270315555555
270316666666
270317777777
270318888888
270319999999
270320000000
270321111111
270322222222
270323333333
270324444444
270325555555
270326666666
270327777777
270328888888
270329999999
270330000000
270331111111
270332222222
270333333333
270334444444
270335555555
270336666666
270337777777
270338888888
270339999999
270340000000
270341111111
270342222222
270343333333
270344444444
270345555555
270346666666
270347777777
270348888888
270349999999
270350000000
270351111111
270352222222
270353333333
270354444444
270355555555
270356666666
270357777777
270358888888
270359999999
270360000000
270361111111
270362222222
270363333333
270364444444
270365555555
270366666666
270367777777
270368888888
270369999999
270370000000
270371111111
270372222222
270373333333
270374444444
270375555555
270376666666
270377777777
270378888888
270379999999
270380000000
270381111111
270382222222
270383333333
270384444444
270385555555
270386666666
270387777777
270388888888
270389999999
270390000000
270391111111
270392222222
270393333333
270394444444
270395555555
270396
```

```
User_DetailsController.cs*  ▾ X UQP.UQP.Controllers.User_DetailsController ResetPasswordToken(UNString
1829
1830
1831
1832
1833
1834
1835
1836
1837
1838
1839
1840
1841
1842
1843
1844
1845
1846
1847
1848
1849
1850
1851
1852
1853
1854
1855
1856
1857
1858
1859
1860
1861
1862
1863
1864
1865
1866
1867
1868
1869
1870
1871
1872
1873
1874
1875
1876
1877
1878
1879
1880
1881
1882
1883
1884
1885
1886
1887
1888
1889
1890
1891
1892
1893
1894
1895
1896
1897
1898
1899
1900
1901
1902
1903
1904
1905
1906
1907
1908
1909
1910
1911
1912
1913
1914
1915
1916
1917
1918
1919
1920
1921
1922
1923
1924
1925
1926
1927
1928
1929
1930
1931
1932
1933
1934
1935
1936
1937
1938
1939
1940
1941
1942
1943
1944
1945
1946
1947
1948
1949
1950
1951
1952
1953
1954
1955
1956
1957
1958
1959
1960
1961
1962
1963
1964
1965
1966
1967
1968
1969
1970
1971
1972
1973
1974
1975
1976
1977
1978
1979
1980
1981
1982
1983
1984
1985
1986
1987
1988
1989
1990
1991
1992
1993
1994
1995
1996
1997
1998
1999
2000
2001
2002
2003
2004
2005
2006
2007
2008
2009
2010
2011
2012
2013
2014
2015
2016
2017
2018
2019
2020
2021
2022
2023
2024
2025
2026
2027
2028
2029
2030
2031
2032
2033
2034
2035
2036
2037
2038
2039
2040
2041
2042
2043
2044
2045
2046
2047
2048
2049
2050
2051
2052
2053
2054
2055
2056
2057
2058
2059
2060
2061
2062
2063
2064
2065
2066
2067
2068
2069
2070
2071
2072
2073
2074
2075
2076
2077
2078
2079
2080
2081
2082
2083
2084
2085
2086
2087
2088
2089
2090
2091
2092
2093
2094
2095
2096
2097
2098
2099
2099
2100
2101
2102
2103
2104
2105
2106
2107
2108
2109
2110
2111
2112
2113
2114
2115
2116
2117
2118
2119
2120
2121
2122
2123
2124
2125
2126
2127
2128
2129
2130
2131
2132
2133
2134
2135
2136
2137
2138
2139
2140
2141
2142
2143
2144
2145
2146
2147
2148
2149
2150
2151
2152
2153
2154
2155
2156
2157
2158
2159
2160
2161
2162
2163
2164
2165
2166
2167
2168
2169
2170
2171
2172
2173
2174
2175
2176
2177
2178
2179
2180
2181
2182
2183
2184
2185
2186
2187
2188
2189
2190
2191
2192
2193
2194
2195
2196
2197
2198
2199
2199
2200
2201
2202
2203
2204
2205
2206
2207
2208
2209
2210
2211
2212
2213
2214
2215
2216
2217
2218
2219
2219
2220
2221
2222
2223
2224
2225
2226
2227
2228
2229
2229
2230
2231
2232
2233
2234
2235
2236
2237
2238
2239
2239
2240
2241
2242
2243
2244
2245
2246
2247
2248
2249
2249
2250
2251
2252
2253
2254
2255
2256
2257
2258
2259
2259
2260
2261
2262
2263
2264
2265
2266
2267
2268
2269
2269
2270
2271
2272
2273
2274
2275
2276
2277
2278
2279
2279
2280
2281
2282
2283
2284
2285
2286
2287
2288
2289
2289
2290
2291
2292
2293
2294
2295
2296
2297
2298
2299
2299
2300
2301
2302
2303
2304
2305
2306
2307
2308
2309
2309
2310
2311
2312
2313
2314
2315
2316
2317
2318
2319
2319
2320
2321
2322
2323
2324
2325
2326
2327
2328
2329
2329
2330
2331
2332
2333
2334
2335
2336
2337
2338
2339
2339
2340
2341
2342
2343
2344
2345
2346
2347
2348
2349
2349
2350
2351
2352
2353
2354
2355
2356
2357
2358
2359
2359
2360
2361
2362
2363
2364
2365
2366
2367
2368
2369
2369
2370
2371
2372
2373
2374
2375
2376
2377
2378
2379
2379
2380
2381
2382
2383
2384
2385
2386
2387
2388
2389
2389
2390
2391
2392
2393
2394
2395
2396
2397
2398
2399
2399
2400
2401
2402
2403
2404
2405
2406
2407
2408
2409
2409
2410
2411
2412
2413
2414
2415
2416
2417
2418
2419
2419
2420
2421
2422
2423
2424
2425
2426
2427
2428
2429
2429
2430
2431
2432
2433
2434
2435
2436
2437
2438
2439
2439
2440
2441
2442
2443
2444
2445
2446
2447
2448
2449
2449
2450
2451
2452
2453
2454
2455
2456
2457
2458
2459
2459
2460
2461
2462
2463
2464
2465
2466
2467
2468
2469
2469
2470
2471
2472
2473
2474
2475
2476
2477
2478
2479
2479
2480
2481
2482
2483
2484
2485
2486
2487
2488
2489
2489
2490
2491
2492
2493
2494
2495
2496
2497
2498
2498
2499
2499
2500
2501
2502
2503
2504
2505
2506
2507
2508
2509
2509
2510
2511
2512
2513
2514
2515
2516
2517
2518
2519
2519
2520
2521
2522
2523
2524
2525
2526
2527
2528
2529
2529
2530
2531
2532
2533
2534
2535
2536
2537
2538
2539
2539
2540
2541
2542
2543
2544
2545
2546
2547
2548
2549
2549
2550
2551
2552
2553
2554
2555
2556
2557
2558
2559
2559
2560
2561
2562
2563
2564
2565
2566
2567
2568
2569
2569
2570
2571
2572
2573
2574
2575
2576
2577
2578
2579
2579
2580
2581
2582
2583
2584
2585
2586
2587
2588
2589
2589
2590
2591
2592
2593
2594
2595
2596
2597
2598
2598
2599
2599
2600
2601
2602
2603
2604
2605
2606
2607
2608
2609
2609
2610
2611
2612
2613
2614
2615
2616
2617
2618
2619
2619
2620
2621
2622
2623
2624
2625
2626
2627
2628
2629
2629
2630
2631
2632
2633
2634
2635
2636
2637
2638
2639
2639
2640
2641
2642
2643
2644
2645
2646
2647
2648
2649
2649
2650
2651
2652
2653
2654
2655
2656
2657
2658
2659
2659
2660
2661
2662
2663
2664
2665
2666
2667
2668
2669
2669
2670
2671
2672
2673
2674
2675
2676
2677
2678
2679
2679
2680
2681
2682
2683
2684
2685
2686
2687
2688
2689
2689
2690
2691
2692
2693
2694
2695
2696
2697
2698
2698
2699
2699
2700
2701
2702
2703
2704
2705
2706
2707
2708
2709
2709
2710
2711
2712
2713
2714
2715
2716
2717
2718
2719
2719
2720
2721
2722
2723
2724
2725
2726
2727
2728
2729
2729
2730
2731
2732
2733
2734
2735
2736
2737
2738
2739
2739
2740
2741
2742
2743
2744
2745
2746
2747
2748
2749
2749
2750
2751
2752
2753
2754
2755
2756
2757
2758
2759
2759
2760
2761
2762
2763
2764
2765
2766
2767
2768
2769
2769
2770
2771
2772
2773
2774
2775
2776
2777
2778
2779
2779
2780
2781
2782
2783
2784
2785
2786
2787
2788
2789
2789
2790
2791
2792
2793
2794
2795
2796
2797
2798
2798
2799
2799
2800
2801
2802
2803
2804
2805
2806
2807
2808
2809
2809
2810
2811
2812
2813
2814
2815
2816
2817
2818
2819
2819
2820
2821
2822
2823
2824
2825
2826
2827
2828
2829
2829
2830
2831
2832
2833
2834
2835
2836
2837
2838
2839
2839
2840
2841
2842
2843
2844
2845
2846
2847
2848
2849
2849
2850
2851
2852
2853
2854
2855
2856
2857
2858
2859
2859
2860
2861
2862
2863
2864
2865
2866
2867
2868
2869
2869
2870
2871
2872
2873
2874
2875
2876
2877
2878
2879
2879
2880
2881
2882
2883
2884
2885
2886
2887
2888
2889
2889
2890
2891
2892
2893
2894
2895
2896
2897
2898
2898
2899
2899
2900
2901
2902
2903
2904
2905
2906
2907
2908
2909
2909
2910
2911
2912
2913
2914
2915
2916
2917
2918
2919
2919
2920
2921
2922
2923
2924
2925
2926
2927
2928
2929
2929
2930
2931
2932
2933
2934
2935
2936
2937
2938
2939
2939
2940
2941
2942
2943
2944
2945
2946
2947
2948
2949
2949
2950
2951
2952
2953
2954
2955
2956
2957
2958
2959
2959
2960
2961
2962
2963
2964
2965
2966
2967
2968
2969
2969
2970
2971
2972
2973
2974
2975
2976
2977
2978
2979
2979
2980
2981
2982
2983
2984
2985
2986
2987
2988
2989
2989
2990
2991
2992
2993
2994
2995
2996
2997
2998
2998
2999
2999
3000
3001
3002
3003
3004
3005
3006
3007
3008
3009
3009
3010
3011
3012
3013
3014
3015
3016
3017
3018
3019
3019
3020
3021
3022
3023
3024
3025
3026
3027
3028
3029
3029
3030
3031
3032
3033
3034
3035
3036
3037
3038
3039
3039
3040
3041
3042
3043
3044
3045
3046
3047
3048
3049
3049
3050
3051
3052
3053
3054
3055
3056
3057
3058
3059
3059
3060
3061
3062
3063
3064
3065
3066
3067
3068
3069
3069
3070
3071
3072
3073
3074
3075
3076
3077
3078
3079
3079
3080
3081
3082
3083
3084
3085
3086
3087
3088
3089
3089
3090
3091
3092
3093
3094
3095
3096
3097
3098
3098
3099
3099
3100
3101
3102
3103
3104
3105
3106
3107
3108
3109
3109
3110
3111
3112
3113
3114
3115
3116
3117
3118
3119
3119
3120
3121
3122
3123
3124
3125
3126
3127
3128
3129
3129
3130
3131
3132
3133
3134
3135
3136
3137
3138
3139
3139
3140
3141
3142
3143
3144
3145
3146
3147
3148
3149
3149
3150
3151
3152
3153
3154
3155
3156
3157
3158
3159
3159
3160
3161
3162
3163
3164
3165
3166
3167
3168
3169
3169
3170
3171
3172
3173
3174
3175
3176
3177
3178
3179
3179
3180
3181
3182
3183
3184
3185
3186
3187
3188
3189
3189
3190
3191
3192
3193
3194
3195
3196
3197
3198
3198
3199
3199
3200
3201
3202
3203
3204
3205
3206
3207
3208
3209
3209
3210
3211
3212
3213
3214
3215
3216
3217
3218
3219
3219
3220
3221
3222
3223
3224
3225
3226
3227
3228
3229
3229
3230
3231
3232
3233
3234
3235
3236
3237
3238
3239
3239
3240
3241
3242
3243
3244
3245
3246
3247
3248
3249
3249
3250
3251
3252
3253
3254
3255
3256
3257
3258
3259
3259
3260
3261
3262
3263
3264
3265
3266
3267
3268
3269
3269
3270
3271
3272
3273
3274
3275
3276
3277
3278
3279
3279
3280
3281
3282
3283
3284
3285
3286
3287
3288
3289
3289
3290
3291
3292
3293
3294
3295
3296
3297
3298
3298
3299
3299
3300
3301
3302
3303
3304
3305
3306
3307
3308
3309
3309
3310
3311
3312
3313
3314
3315
3316
3317
3318
3319
3319
3320
3321
3322
3323
3324
3325
3326
3327
3328
3329
3329
3330
3331
3332
3333
3334
3335
3336
3337
3338
3339
3339
3340
3341
3342
3343
3344
3345
3346
3347
3348
3349
3349
3350
3351
3352
3353
3354
3355
3356
3357
3358
3359
3359
3360
3361
3362
3363
3364
3365
3366
3367
3368
3369
3369
3370
3371
3372
3373
3374
3375
3376
3377
3378
3379
3379
3380
3381
3382
3383
3384
3385
3386
3387
3388
3389
3389
3390
3391
3392
3393
3394
3395
3396
3397
3398
3398
3399
3399
3400
3401
3402
3403
3404
3405
3406
3407
3408
3409
3409
3410
3411
3412
3413
3414
3415
3416
3417
3418
3419
3419
3420
3421
3422
3423
3424
3425
3426
3427
3428
3429
3429
3430
3431
3432
3433
3434
3435
3436
3437
3438
3439
3439
3440
3441
3442
3443
3444
3445
3446
3447
3448
3449
3449
3450
3451
3452
3453
3454
3455
3456
3457
3458
3459
3459
3460
3461
3462
3463
3464
3465
3466
3467
3468
3469
3469
3470
3471
3472
3473
3474
3475
3476
3477
3478
3479
3479
3480
3481
3482
3483
3484
3485
3486
3487
3488
3489
3489
3490
3491
3492
3493
3494
3495
3496
3497
3498
3498
3499
3499
3500
3501
3502
3503
3504
3505
3506
3507
3508
3509
3509
3510
3511
3512
3513
3514
3515
3516
3517
3518
3519
3519
3520
3521
3522
3523
3524
3525
3526
3527
3528
3529
3529
3530
3531
3532
3533
3534
3535
3536
3537
3538
3539
3539
3540
3541
3542
3543
3544
3545
3546
3547
3548
3549
3549
3550
3551
3552
3553
3554
3555
3556
3557
3558
3559
3559
3560
3561
3562
3563
3564
3565
3566
3567
3568
3569
3569
3570
3571
3572
3573
3574
3575
3576
3577
3578
3579
3579
3580
3581
3582
3583
3584
3585
3586
3587
3588
3589
3589
3590
3591
3592
3593
3594
3595
3596
3597
3598
3598
3599
3599
3600
3601
3602
3603
3604
3605
3606
3607
3608
3609
3609
3610
3611
3612
3613
3614
3615
3616
3617
3618
3619
3619
3620
3621
3622
3623
3624
3625
3626
3627
3628
3629
3629
3630
3631
3632
3633
3634
3635
3636
3637
3638
3639
3639
3640
3641
3642
3643
3644
3645
3646
3647
3648
3649
3649
3650
3651
3652
3653
3654
3655
3656
3657
3658
3659
3659
3660
3661
3662
3663
3664
3665
3666
3667
3668
3669
3669
3670
3671
3672
3673
3674
3675
3676
3677
3678
3679
3679
3680
3681
3682
3683
3684
3685
3686
3687
3688
3689
3689
3690
3691
3692
3693
3694
3695
3696
3697
3698
3698
3699
3699
3700
3701
3702
3703
3704
3705
3706
3707
3708
3709
3709
3710
3711
3712
3713
3714
3715
3716
3717
3718
3719
3719
3720
3721
3722
3723
3724
3725
3726
3727
3728
3729
3729
3730
3731
3732
3733
3734
3735
3736
3737
3738
3739
3739
3740
3741
3742
3743
3744
3745
3746
3747
3748
3749
3749
3750
3751
3752
3753
3754
3755
3756
3757
3758
3759
3759
3760
3761
3762
3763
3764
3765
3766
3767
3768
3769
3769
3770
3771
3772
3773
3774
3775
3776
3777
3778
3779
3779
3780
3781
3782
3783
3784
3785
3786
3787
3788
3789
3789
3790
3791
3792
3793
3794
3795
3796
3797
3798
3798
3799
3799
3800
3801
3802
3803
3804
3805
3806
3807
3808
3809
3809
3810
3811
3812
3813
3814
3815
3816
3817
3818
3819
3819
3820
3821
3822
3823
3824
3825
3826
3827
3828
3829
3829
3830
3831
3832
3833
3834
3835
3836
3837
3838
3839
3839
3840
3841
3842
3843
3844
3845
3846
3847
3848
3849
3849
3850
3851
3852
3853
3854
3855
3856
3857
3858
3859
3859
3860
3861
3862
3863
3864
3865
3866
3867
3868
3869
3869
3870
3871
3872
3873
3874
3875
3876
3877
3878
3879
3879
3880
3881
3882
3883
3884
3885
3886
3887
3888
3889
3889
3890
3891
3892
3893
3894
3895
3896
3897
3898
3898
3899
3899
3900
3901
3902
3903
3904
3905
3906
3907
3908
3909
3909
3910
3911
3912
3913
3914
3915
3916
3917
3918
3919
3919
3920
3921
3922
3923
3924
3925
3926
3927
3928
3929
3929
3930
3931
3932
3933
3934
3935
3936
3937
3938
3939
3939
3940
3941
3942
3943
3944
3945
3946
3947
3948
3949
3949
3950
3951
3952
3953
3954
3955
3956
3957
3958
3959
3959
3960
3961
3962
3963
3964
3965
3966
3967
3968
3969
3969
3970
3971
3972
3973
3974
3975
3976
3977
3978
3979
3979
3980
3981
3982
3983
3984
3985
3986
3987
3988
3989
3989
3990
3991
3992
3993
3994
3995
3996
3997
3998
3998
3999
3999
4000
4001
4002
4003
4004
4005
4006
4007
4008
4009
4009
4010
4011
4012
4013
4014
4015
4016
4017
4018
4019
4019
4020
4021
4022
4023
4024
4025
4026
4027
4028
4029
4029
4030
4031
4032
4033
4034
4035
4036
4037
4038
4039
4039
4040
4041
4042
4043
4044
4045
4046
4047
4048
4049
4049
4050
4051
4052
4053
4054
4055
4056
4057
4058
4059
4059
4060
4061
4062
4063
4064
4065
4066
4067
4068
4069
4069
4070
4071
4072
4073
4074
4075
4076
4077
4078
4079
4079
4080
4081
4082
4083
4084
4085
4086
4087
4088
4089
4
```

User\_DetailsController.cs\* X

UQP.Controllers.User\_DetailsController

ResetPasswordToken(UriString url)

```
1853 if (query.Count != 0)
1854 {
1855     foreach (var per in query)
1856     {
1857         CustomProviderDetails objcustom = new CustomProviderDetails();
1858         objcustom.FirstName = per.Physician_first_name.ToLower();
1859         //objcustom.MidName = per.Provider_MidName.ToLower();
1860         objcustom.LastName = per.Physician_last_name.ToLower();
1861         objcustom.email = per.emailid;
1862         objcustom.level = per.Specialty.ToLower();
1863
1864         //objcustom.uniqueproviderno = per.UniqueProviderID;
1865         objcustom.providerno = per.providerid;
1866         objcustom.IsPcp = null;
1867         objcustom.title = "Dr.";
1868         objcustom.speciality = per.Specialty.ToLower();
1869
1870         var subquery = (from pf in db.providertesting_address
1871                         where pf.providerid == per.providerid
1872
1873                         select new
1874                         {
1875                             pf.address,
1876                             pf.backline_phone_number,
1877                             pf.fax,
1878                             pf.npi,
1879                             pf.phone,
1880                             pf.providerid,
1881                             pf.recordid,
1882                         });
1883
1884         objcustom.address = subquery.FirstOrDefault().address;
1885         objcustom.backline_phone_number = subquery.FirstOrDefault().backline_phone_number;
1886         objcustom.fax = subquery.FirstOrDefault().fax;
1887         objcustom.npi = subquery.FirstOrDefault().npi;
1888         objcustom.phone = subquery.FirstOrDefault().phone;
1889         objcustom.providerid = subquery.FirstOrDefault().providerid;
1890         objcustom.recordid = subquery.FirstOrDefault().recordid;
1891
1892         db.SaveChanges();
1893     }
1894 }
```

User Details Controller.cs\*    X    UQP.Controllers.User\_DetailsController    ResetPasswordToken(UriString url)

```
1874 select new
1875 {
1876     pf.address,
1877     pf.backline_phone_number,
1878     pf.fax,
1879     pf.npi,
1880     pf.phone,
1881     pf.providerid,
1882     pf.recordid,
1883     pf.isdefault
1884 }).Distinct().ToList();
1885 List<providerTesting_address> addressList = new List<providerTesting_address>();
1886 foreach (var value in subquery)
1887 {
1888     if (value.isdefault == true)
1889     {
1890         phone = value.phone;
1891         objCustom.mobNo = phone;
1892     }
1893     providerTesting_address address = new providerTesting_address
1894     {
1895         address = value.address,
1896         backline_phone_number = value.backline_phone_number,
1897         fax = value.fax,
1898         npi = value.npi,
1899         phone = value.phone,
1900         providerid = value.providerid,
1901         isdefault = value.isdefault,
1902         recordid = value.recordid
1903     };
}
```

Output

```
Show output from: Debug
iisexpress.exe (CLR v4.0.30319; /LM/W3SVC/2/ROOT-1-13230222775672406): Loaded 'C:\Windows\Microsoft.NET\assembly\GAC_MSIL\System.Windows.Forms\v4.0_4.0.0.0_c1d30d94d0d641e5\System.Windows.Forms.dll'
```

Solution Explorer    Tools    Build    Debug    Any CPU    Run    JI Express (Google Chrome)    Help

User\_DetailsController.cs\*    UOP.Controllers.User\_DetailsController    ResetPasswordToken(UriString url)

```
1892
1893     providertesting_address address = new providertesting_address
1894     {
1895         address = value.address,
1896         backline_phone_number = value.backline_phone_number,
1897         fax = value.fax,
1898         npi = value.npi,
1899         phone = value.phone,
1900         providerid = value.providerid,
1901         isdefault = value.isdefault,
1902         recordid = value.recordid
1903     };
1904     addresslist.Add(address);
1905 }
1906 objcustom.addresslist = addresslist;
1907 pvrdrlist.Add(objcustom);
1908 }
1909
1910 }
1911
1912 }
1913
1914 return pvrdrlist;
1915 }
1916
1917
1918 [HttpPost]
1919 [Route("api/User_Details/logout/{id}/{logoutreason?}")]
1920 public HttpResponseMessage logout_user(long id, [FromUri]string logoutreason = null)
```

The screenshot shows the Microsoft Visual Studio IDE interface. The top menu bar includes 'File', 'Edit', 'View', 'Team', 'Tools', 'Test', 'Analyze', 'Window', and 'Help'. A dropdown menu 'Debug' is open, showing 'Any CPU'. Below the menu bar, the title bar displays 'User\_DetailsController.cs\*' and 'UQP.Controllers.User\_DetailsController'. The status bar at the bottom left shows '100 %'. The main code editor area contains the following C# code:

```
1913
1914     return pvrdrlist;
1915 }
1916
1917
1918 [HttpPut]
1919 [Route("api/User_Details/logout/{id}/{logoutreason?}")]
1920 References
1921 public HttpResponseMessage logout_user(long id, [FromUri]string logoutreason = null)
1922 {
1923     //using (var db = new UQPEntities("UQPEntities"))
1924     //{
1925         var result = db.LoginDetails.FirstOrDefault(b => b.Id == id);
1926         if (result != null)
1927         {
1928             result.LogoutTime = DateTime.Now;
1929             result.LogoutReason = logoutreason == "true" ? "Logout due to inactivity" : "Manual Signout";
1930             db.SaveChanges();
1931             return Request.CreateResponse(HttpStatusCode.OK, "Signed Out Successfully");
1932         }
1933         else
1934         {
1935             return Request.CreateResponse(HttpStatusCode.NotFound, "");
1936         }
1937     //}
1938 }
1939
1940 [HttpGet]
1941 [Route("api/User_Details/loginlist")]
```

A screenshot of the Microsoft Visual Studio IDE interface. The main window shows the code editor with a file named "User\_DetailsController.cs". The code is a C# class for a controller, specifically for handling login lists. The editor includes line numbers, syntax highlighting, and a vertical scroll bar. To the left of the code editor is the "Solution Explorer" pane, which displays the project structure with a single item named "UOP". To the right is the "Task List" pane, which is currently empty. The top menu bar shows "Debug" and "Any CPU" settings. The status bar at the bottom indicates the file path "C:\Windows\Microsoft.NET\Framework\v4.0.30319\Temporary ASP.NET Files\root\13230222775672406\1\13230222775672406.cs" and the build number "100.0.0.0\_13230222775672406".

```
1939
1940
1941
1942     [HttpGet]
1943     [Route("api/User_Details/loginlist")]
1944     public HttpResponseMessage loginlist([FromUri]PagingParameterModel pagingparametermodel, bool report)
1945     {
1946         try
1947         {
1948             //using (var db = new UQPEntities("UQPEntities"))
1949             //{
1950                 dynamic query = null;
1951                 ObjectParameter totalcount = new ObjectParameter("totalcount", typeof(Int32));
1952                 List<LogDetails> LogList = new List<LogDetails>();
1953                 int tcount = 0;
1954                 if (report == false)
1955                 {
1956                     query = db.USP_LoginDetailsProcedure("abc", DateTime.Now, DateTime.Now, "", "", pagingparametermodel.pageSize,
1957                     pagingparametermodel.pageNumber, totalcount).ToList();
1958                     tcount = Convert.ToInt32(totalcount.Value);
1959                 }
1960                 else
1961                 {
1962                     query = db.USP_LoginDetailsProcedure("abc", DateTime.Now, DateTime.Now, "", "", pagingparametermodel.pageSize, 0, totalcount).ToList();
1963                     // tcount = Convert.ToInt32(totalcount.Value);
1964                 }
1965
1966                 foreach (var q in query)
1967                 {
1968                     LogDetails logDetails = new LogDetails();
1969                     logDetails.Username = q.username;
1970                     logDetails.Date = q.date;
1971                     LogList.Add(logDetails);
1972                 }
1973             //}
1974         }
1975     }
```

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

User\_DetailsController.cs\* -> X

UQP.Controllers.User\_DetailsController

```
1962
1963
1964     foreach (var q in query)
1965     {
1966         LogDetails logDetails = new LogDetails();
1967         logDetails.Username = q.username;
1968         logDetails.loginid = q.Login_ID;
1969         logDetails.logoutime = q.LoginTime.ToString();
1970         if (q.LogoutTime != null && q.LogoutReason != null)
1971         {
1972             if (q.LogoutTime > DateTime.Now)
1973             {
1974                 logDetails.logoutime = "";
1975                 logDetails.logoutreason = "Member is currently sign-in";
1976             }
1977             else
1978             {
1979                 if ((q.LogoutTime < DateTime.Now && q.LogoutReason == "Tentative Logout Time"))
1980                 {
1981                     logDetails.logoutime = q.LogoutTime.ToString();
1982                     logDetails.logoutreason = "System Sign Out";
1983                 }
1984                 else
1985                 {
1986                     logDetails.logoutime = q.LogoutTime.ToString();
1987                     logDetails.logoutreason = q.LogoutReason;
1988                 }
1989             }
1990         }
1991     }
1992 //if (q.LogoutReason != null)
```

Output

Show output from: Debug

UQP - Microsoft Visual Studio

Edit View Project Build Debug Team Tools Test Analyze Window Help

Source Control Explorer User\_DetailsController.cs\* UQP.UQP

```
//  
1992 //  
1993 //logDetails.logoutreason = q.LogoutReason== "Tentative Logout Time"?":q.LogoutReason;  
1994 //}  
1995 if (q.IpAddress != null)  
1996 {  
1997 logDetails.ipaddress = q.IpAddress;  
1998 }  
1999 if (q.GroupName != null)  
2000 {  
2001 logDetails.group = q.GroupName;  
2002 }  
2003 LogList.Add(logDetails);  
2004 }  
2005  
2006 if (report == false)  
2007 {  
2008 #region pagination  
2009 // Get's No of Rows Count  
2010 int count = tcount;  
2011 //int count = lstmemberart.Count();  
2012  
2013 // Parameter is passed from Query string if it is null then it default Value will be pageNumber:1  
2014 int CurrentPage = pagingparametermodel.pageNumber;  
2015  
2016 // Parameter is passed from Query string if it is null then it default Value will be pageSize:20  
2017 int PageSize = pagingparametermodel.pageSize;  
2018  
2019 // Display TotalCount to Records to User  
2020 int TotalCount = count;  
2021
```

The screenshot shows a Microsoft Visual Studio interface. The menu bar includes File, Build, Debug, Team, Tools, Test, Analyze, Window, and Help. The toolbar has icons for New, Open, Save, Cut, Copy, Paste, Find, and Replace. The status bar at the bottom shows "Scanned with CamScanner".

The code editor displays the file `User_DetailsController.cs` under the namespace `UQP.Controllers`. The code implements a `User_DetailsController` class with methods for handling user details.

```
2010 // Get s No of Rows Count
2011 int count = tcount;
2012 //int count = lstmembermart.Count();
2013
2014 // Parameter is passed from Query string if it is null then it default Value will be pageNumber:1
2015 int currentPage = pagingparametermodel.pageNumber;
2016
2017 // Parameter is passed from Query string if it is null then it default Value will be pageSize:20
2018 int pageSize = pagingparametermodel.pageSize;
2019
2020 // Display TotalCount to Records to User
2021 int totalCount = count;
2022
2023 // Calculating Totalpage by Dividing (No of Records / Pagesize)
2024 int totalPages = (int)Math.Ceiling(count / (double)pageSize);
2025
2026 // Returns List of Customer after applying Paging
2027 var items = Loglist/*.Skip((currentPage - 1) * pageSize).Take(pageSize).ToList()*/;
2028
2029 // if currentPage is greater than 1 means it has previousPage
2030 var previousPage = currentPage > 1 ? "Yes" : "No";
2031
2032 // if totalPages is greater than currentPage means it has nextPage
2033 var nextPage = currentPage < totalPages ? "Yes" : "No";
2034
2035 // Object which we are going to send in header
2036 var paginationMetadata = new
2037 {
2038     totalCount = totalCount,
2039     pageSize = pageSize,
```

```
    // if CurrentPage is greater than 1 means it has previousPage
    var previousPage = currentPage > 1 ? "Yes" : "No"; . . .

    // if TotalPages is greater than CurrentPage means it has nextPage
    var nextPage = currentPage < totalPages ? "Yes" : "No"; . . .

    // Object which we are going to send in header
    var paginationMetadata = new
    {
        totalCount = TotalCount,
        pageSize = PageSize,
        currentPage = currentPage,
        totalPages = TotalPages,
        previousPage,
        nextPage
    };

    // Setting Header
    #endregion
    System.Web.HttpContext.Current.Response.Headers.Add("Paging-Headers", Newtonsoft.Json.JsonConvert.SerializeObject(paginationMetadata));
    //HttpContext.Current.Response.Headers.Add("filename", zipFileName);
    HttpContext.Current.Response.Headers.Add("Access-Control-Expose-Headers", "Paging-Headers");
    return Request.CreateResponse(HttpStatusCode.OK, items);
}

else if (report == true)
{
    var rpoj = new ReportsController();
    {
```



User Details Controller

User\_DetailsController.cs

```
UQP.UQP.Controllers.User_DetailsController
2064     return Request.CreateResponse(HttpStatusCode.OK, "OK");
2065 }
2066 }
2067 catch (Exception e)
2068 {
2069     return Request.CreateResponse(HttpStatusCode.InternalServerError, e.InnerException);
2070 }
2071 }
2072
2073 [HttpPut]
2074 [Route("api/User_Details/theme")]
2075 public HttpResponseMessage theme(Theme themeobj)
2076 {
2077     try
2078     {
2079         //using (var db = new UQPEntities("UQPEntities"))
2080         //{
2081             var result = db.User_Details.FirstOrDefault(u => u.User_ID == themeobj.user_id);
2082             if (result != null)
2083             {
2084                 result.Theme = themeobj.theme;
2085                 result.Colour_Code = themeobj.colourcode;
2086                 db.SaveChanges();
2087                 return Request.CreateResponse(HttpStatusCode.OK, "Theme updated successfully");
2088             }
2089             else
2090             {
2091                 return Request.CreateResponse(HttpStatusCode.OK, "User id passed is wrong");
2092             }
2093         //}
2094     }
2095 }
```

Output

Show output from: Debug

```
Windows Client Configuration Assembly loaded C:\Windows\Microsoft.NET\Assembly\GAC_MSIL\System.Windows.Forms\1.0.29.5727\WindowsClientConfig.dll
```

Source Control Explorer    User\_DetailsController.cs\*    X

UQP    UQP.Controllers.User\_DetailsController    loginlist(PagingParameterModel paging)

```
2090
2091
2092
2093
2094
2095     catch (Exception e)
2096     {
2097         return Request.CreateResponse(HttpStatusCode.BadRequest, e.Message);
2098     }
2099
2100
2101 [HttpPost]
2102 [Route("api/User_Details/SendOTP")]
2103 public HttpResponseMessage SendOTP(UrlString objemail)
2104 {
2105     try
2106     {
2107         using (UQPEntities db = new UQPEntities(HttpContext.Current.Request.Headers.GetValues("IPA").FirstOrDefault()))
2108         {
2109
2110             var emaildetails = (from n in db.NewEmailContents
2111                                 where n.Id == 5
2112                                 select n).FirstOrDefault();
2113             string OTP = GenerateRandomPWD(4);
2114             ZipFileOtp zipFileOtp =
2115             {
2116                 email = objemail.email,
2117                 otp = OTP,
2118                 createddate = DateTime.Now,
2119                 expiredstatus = false
2120             };
2121             db.NewEmailContents.Add(zipFileOtp);
2122             db.SaveChanges();
2123             return Request.CreateResponse(HttpStatusCode.OK, "User id passed is wrong");
2124         }
2125     }
2126     catch (Exception e)
2127     {
2128         return Request.CreateResponse(HttpStatusCode.BadRequest, e.Message);
2129     }
2130 }
```

Output

```
Show output from: Debug
iisexpress.exe (CLR v4.0.30319: /LM/W3SVC/2/ROOT-1-13230222775672406): Loaded 'C:\Windows\Microsoft.NET\assembly\GAC_MSIL\System.Windows.Forms\v4.0_4.0.0.0__b77adcf6bb6a3c8a'. Loaded 'C:\Windows\Microsoft.NET\assembly\GAC_MSIL\System.Data.OracleClient\v4.0_4.0.0.0__b77adcf6bb6a3c8a'.
'iisexpress.exe' (CLR v4.0.30319: /LM/W3SVC/2/ROOT-1-13230222775672406): Loaded 'C:\Windows\Microsoft.NET\assembly\GAC_32\System.Data.EntityFrameworkDynamicProxies-UQP'.
```

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

- Menu Bar:** File, Project, Build, Debug, Team, Tools, Test, Analyze, Window, Help.
- Toolbars:** Standard, Debug, Task List, Solution Explorer, Properties, Toolbox, Task List, Status Bar.
- Code Editor:** The main window displays the code for `User_DetailsController.cs`. The code is as follows:

```
2117 otp = OTP,
2118 createddate = DateTime.Now,
2119 expiredstatus = false
2120 };
2121 db.ZipfileOtps.Add(zipFile);
2122 db.SaveChanges();
2123 if (ConfigSettings.email.Contains("true"))
2124 {
2125     processAuthEmail emailobj = new processAuthEmail
2126     {
2127         replyto = objemail.email,
2128         subject = emaildetails.EmailSubject,
2129         body = emaildetails.EmailBody + OTP,
2130         signature = emaildetails.EmailSignature
2131     };
2132     using (var ProcessAuthController = new ProcessAuthController())
2133     {
2134         ProcessAuthController.processAuthEmail(emailobj);
2135     }
2136 }
2137 return Request.CreateResponse(HttpStatusCode.OK, "");
2138 }
2139 }
2140 catch (Exception e)
2141 {
2142     return Request.CreateResponse(HttpStatusCode.BadRequest, e);
2143 }
2144 }
2145 }
```

The code handles the creation of a ZIP file containing OTPs and sends an email with the OTP using a `processAuthEmail` object. It uses a `ProcessAuthController` to perform the actual email sending.

A screenshot of the Microsoft Visual Studio IDE interface. The title bar shows "User\_DetailsController.cs" and "UQP.Controllers.User\_DetailsController". The code editor displays the following C# code:

```
2141     catch (Exception e)
2142     {
2143         return Request.CreateResponse(HttpStatusCode.BadRequest, e);
2144     }
2145 }
2146
2147 //protected override void Dispose(bool disposing)
2148 //{
2149 //    if (disposing)
2150 //    {
2151 //        db.Dispose();
2152 //    }
2153 //    base.Dispose(disposing);
2154 //}
2155
2156 [HttpPost]
2157 [Route("api/User_Details/ResetPasswordByAdmin/{userid}")]
2158 public HttpResponseMessage ResetPasswordByAdmin(int userid, TokenEmailStatus token)
2159 {
2160     try
2161     {
2162         var query = db.User_Details.Where(e => e.User_ID == userid).FirstOrDefault();
2163         if (query != null)
2164         {
2165             SHA1CryptoServiceProvider encrypt = new SHA1CryptoServiceProvider();
2166             byte[] hashValueMessage = encrypt.ComputeHash(Encoding.Default.GetBytes(token.password));
2167             string hexOfValueMessage = BitConverter.ToString(hashValueMessage);
2168             query.User_Password = hexOfValueMessage;
2169             db.SaveChanges();
2170         }
2171     }
2172 }
```

```
2158     public HttpResponseMessage ResetPasswordByAdmin(int userid, TokenEmailStatus token)
2159     {
2160         try
2161         {
2162             var query = db.User_Details.Where(e => e.User_ID == userid).FirstOrDefault();
2163             if (query != null)
2164             {
2165                 SHA1CryptoServiceProvider encrypt = new SHA1CryptoServiceProvider();
2166                 byte[] hashValueMessage = encrypt.ComputeHash(Encoding.Default.GetBytes(token.password));
2167                 string hexOfValueMessage = BitConverter.ToString(hashValueMessage);
2168                 query.User_Password = hexOfValueMessage;
2169                 db.SaveChanges();
2170                 return Request.CreateResponse(HttpStatusCode.OK, "Password changed successfully");
2171             }
2172             else
2173             {
2174                 return Request.CreateResponse(HttpStatusCode.OK, "User Id specified is wrong");
2175             }
2176         }
2177         catch (Exception ex)
2178         {
2179             return Request.CreateResponse(HttpStatusCode.InternalServerError, ex.InnerException);
2180         }
2181     }
2182 
2183     [HttpGet]
2184     [Route("api/User_Details/GetTeam/{id}")]
2185     [Reference]
2186     public HttpResponseMessage GetTeam(int id)
```

User\_Control Explorer    User\_DetailsController.cs\*    X

Debug    Any CPU    IIS Express (Google Chrome)    loginlist(PagingParameterModel pagingparametermodel)

UOP

```
2176 }  
2177 catch (Exception ex)  
2178 {  
2179     return Request.CreateResponse(HttpStatusCode.InternalServerError, ex.InnerException);  
2180 }  
2181  
2182  
2183  
2184 [HttpGet]  
2185 [Route("api/User_Details/GetTeam/{id}")]  
2186 public HttpResponseMessage GetTeam(int id)  
2187 {  
2188     try  
2189     {  
2190         var team = from tm in db.OrganisationTeams  
2191                 where tm.OrganisationTypeId == id  
2192                 select tm;  
2193         List<OrganisationTeam> lstteam = new List<OrganisationTeam>();  
2194         foreach (var per in team)  
2195         {  
2196  
2197             lstteam.Add(new OrganisationTeam { Id = per.Id, Name = per.Name });  
2198         }  
2199         return Request.CreateResponse(HttpStatusCode.OK, lstteam);  
2200     }  
2201     catch (Exception e)
```

Output

Show output from: Debug

User DetailsController.cs

```
2199
2200
2201
2202     return Request.CreateResponse(HttpStatusCode.OK, 1stteam);
2203 }
2204 catch (Exception e)
2205 {
2206     return Request.CreateErrorResponse(HttpStatusCode.NotFound, e);
2207 }
2208
2209
2210
2211
2212
2213
2214 [HttpGet]
2215 [Route("api/userdetails/rolewiseacl/{roleid}")]
2216 public HttpResponseMessage rolewiseacl(int roleid)
2217 {
2218     try
2219     {
2220         ////using (var db = new UQPEntities("UQPEntities"))
2221         //////
2222         //ClaimsPrincipal principal = Request.GetRequestContext().Principal as ClaimsPrincipal;
2223         //var userName = principal.Claims.Where(c => c.Type == "useremail").Single().Value;
2224         //var loginid = principal.Claims.Where(c => c.Type == "id").Single().Value;
2225         //int userid = Convert.ToInt32(principal.Claims.Where(c => c.Type == "userid").Single().Value);
2226         CustomUser objuser = new CustomUser();
2227         //    var entity = db.User_Details.AsNoTracking().FirstOrDefault(e => e.User_ID == userid);
```

Output

```
Show output from: Debug
'iisexpress.exe' (CLR v4.0.30319: /LM/W3SVC/2/ROOT-1-13230222775672406): Loaded 'C:\Windows\Microsoft.NET\Framework\v4.0.30319\Temporary ASP.NET Files\root\13230222775672406\1\{assembly}.dll' from 'C:\Windows\Microsoft.NET\assembly\GAC_MSIL\System.Web\4.0.0.0__b03f5f7f15d71a0a\System.Web.dll'.
'iisexpress.exe' (CLR v4.0.30319: /LM/W3SVC/2/ROOT-1-13230222775672406): Loaded 'C:\Windows\Microsoft.NET\assembly\GAC_MSIL\System.Windows.Forms\v4.0_4.0.0_0__b03f5f7f15d71a0a\System.Windows.Forms.dll'.
'iisexpress.exe' (CLR v4.0.30319: /LM/W3SVC/2/ROOT-1-13230222775672406): Loaded 'C:\Windows\Microsoft.NET\assembly\GAC_32\System.Data.OracleClient\v4.0_4.0.0_0__b77acca47984e78c\System.Data.OracleClient.dll'.
```

```
//int userid = Convert.ToInt32(principal.Claims.Where(c => c.Type == "userid").Single().Value);
CustomUser objuser = new CustomUser();
// var entity = db.User_Details.AsNoTracking().FirstOrDefault(e => e.User_ID == userid);
// DateTime? lastrefresheddate = db.NewDashboardSuperset.Where(x => x.Executive_id == 1)
// .Select(x => x.LastGapRefreshedDate).FirstOrDefault();
List<ACLDetails> acclist = new List<ACLDetails>();
//List<CustomIPA> ipalist = new List<CustomIPA>();

// if (entity != null)

// {
// objuser.lastgrefresh = lastrefresheddate;
// if (entity.User_IPA_ID != null)
//{
//     var stripaids = entity.User_IPA_ID.Split(',');
//     int[] ipaid = stripaids.Select(int.Parse).ToArray();
//     foreach (var ip in ipaid)
//     {
//         CustomIPA ipaobj = new CustomIPA();
//         var ipainfo = (from ips in db.IPAs.AsNoTracking()
//                     where ips.IPA_ID == ip
//                     select new { ips.IPA_ID, ips.IPA_Name, ips.IPA_Code }).FirstOrDefault();
//         ipaobj.IPA_ID = ipainfo.IPA_ID;
//         ipaobj.IPA_Name = ipainfo.IPA_Name;
//         ipalist.Add(ipaobj);
//     }
// }
```

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

- Title Bar:** Debug - Any CPU
- File List:** UQP
- Code Editor:** User\_DetailsController.cs\*  
The code is part of the UQP.Controllers.User\_DetailsController class. It handles the addition of a new IPA object to a list and retrieves a default IPA object based on entity settings. It also iterates through organisation types and checks if a specific value is present.

```
2249
2250     // ipaobj.IPA_Name = ipainfo.IPA_Name;
2251     ipalist.Add(ipaobj);
2252
2253
2254     //}
2255 }
2256
2257 // CustomIPA ipadefobj = new CustomIPA();
2258 // var ipadefvar =
2259
2260 //((from ips in db.IPAs.AsNoTracking()
2261 // where ips.IPA_ID == entity.DefaultIPA
2262 // select new { ips.IPA_ID, ips.IPA_Name, ips.IPA_Code }).FirstOrDefault();
2263 // if (ipadefvar != null)
2264 //{
2265 //     ipadefobj.IPA_ID = ipadefvar.IPA_ID;
2266 //     ipadefobj.IPA_Name = ipadefvar.IPA_Name;
2267 //     ipadefobj.ipacode = ipadefvar.IPA_Code;
2268 //}
2269 // objuser.defaultIPAobj = ipadefobj;
2270
2271 Organisation orgobj = new Organisation();
2272 //if (entity.OrganisationType != null)
2273 //{
2274 //    var org = entity.OrganisationType.Split(',');
2275
2276 //    foreach (var per in org)
2277 //{
2278 //        if (per == "1")
```

- Status Bar:** 00 %
- Output Tab:** Output
- Show Output From:** Debug
- Task List:** ilseexpress.exe (CLR v4.0.30319; /LM/W3SVC/2/ROOT-1-132302222775672406) loaded 'C:\Windows\Microsoft.NET\assembly\GAC\_MSIL\System.Windows.Forms\v4.0\_4.0.0.0\_b77a5c561934e089'; ilseexpress.exe (CLR v4.0.30319; /LM/W3SVC/2/ROOT-1-132302222775672406) loaded 'C:\Windows\Microsoft.NET\assembly\GAC\_32\System.Data.OracleClient\v4.0\_4.0.0.0\_b'

```
UOP
2270
2271
2272     Organisation orgobj = new Organisation();
2273     //if (entity.OrganisationType != null)
2274     //{
2275     //    var org = entity.OrganisationType.Split(',');
2276
2277     //    foreach (var per in org)
2278     //{
2279     //        if (per == "1")
2280     //{
2281         orgobj.hedis = true;
2282
2283     //}
2284     //    if (per == "2")
2285     //{
2286         orgobj.mra = true;
2287
2288     //}
2289     //    if (per == "3")
2290     //{
2291         orgobj.aco = true;
2292
2293     //}
2294
2295
2296     objuser.organisations = orgobj;
2297
2298     var acl = (from a in db.ACls.AsNoTracking()
2299               join ar in db.ACList_Role on a.id equals ar.ACList_ID
100 %
```

UQP.UQP.Controllers.User\_DetailsController loginlist(PagingParameterModel paging, [FromServices] IUnitOfWork unitOfWork)

```
2294
2295
2296     objuser.organisations = orgobj;
2297
2298     var acl = (from a in db.ACls.AsNoTracking()
2299                 join ar in db.ACList_Role on a.id equals ar.ACList_ID
2300                 where ar.Role_ID == roleid
2301                 select new { a.id, ar.AddACL, ar.EditACL, ar.ViewACL, a.Functions }).ToList();
2302
2303     comment
2304
2305     List<long> mraids = new List<long>();
2306     List<long> hedisids = new List<long>();
2307
2308     long[] abc = mraids.ToArray();
2309
2310     mraids.InsertRange(mraids.Count, new long[] { 29, 32, 34 });
2311     hedisids.InsertRange(hedisids.Count, new long[] { 28, 31 });
2312     foreach (var per in acl.ToList())
2313     {
2314         ACLDetails aclobj = new ACLDetails();
2315         List<acl_columns> Columnlst = new List<acl_columns>();
2316         // aclobj.aclid = per.id;
2317         long abcd = per.id;
2318         if (objuser.organisations.hedis == null && hedisids.Contains(per.id))
2319         {
2320             aclobj.add = false;
2321             aclobj.edit = false;
2322             aclobj.view = false;
2323             aclobj.func = per.Functions;
2324         }
2325     }
2326
2327 }
```

User Control Explorer    User\_DetailsController.cs\*    X

```
UQP
2463 List<acl_columns> ColumnList = new List<acl_columns>();
2464 // aclobj.aclid = per.id;
2465 long abcd = per.id;
2466 if (objuser.organisations.hedis == null && hedisid.Contains(per.id))
2467 {
2468     aclobj.add = false;
2469     aclobj.edit = false;
2470     aclobj.view = false;
2471     aclobj.func = per.Functions;
2472 }
2473 if (objuser.organisations.mra == null && mraid.Contains(per.id))
2474 {
2475     aclobj.add = false;
2476     aclobj.edit = false;
2477     aclobj.view = false;
2478     aclobj.func = per.Functions;
2479 }
2480 if ((objuser.organisations.mra == null && !mraid.Contains(per.id)) ||
2481     (objuser.organisations.hedis == null && !hedisid.Contains(per.id)) ||
2482     (objuser.organisations.mra == true && objuser.organisations.hedis == true))
2483 {
2484     aclobj.add = per.AddACL;
2485     aclobj.edit = per.EditACL;
2486     aclobj.view = per.ViewACL;
2487     aclobj.func = per.Functions;
2488 }
2489 var x = db.ACL_Column_By_Role.AsNoTracking().Where(e => e.ACL_id == per.id && e.Role_id == objuser.roleid);
```



```
2507
2508
2509
2510
2511
2512
2513
2514
2515
2516
2517
2518
2519
2520
2521
2522
2523
2524
2525
2526
2527
2528
2529
2530
2531
2532
2533
2534
    {
        acl_columns columns = new acl_columns();
        columns.aclid = value.AC1_id.Value;
        columns.column_name = value.Column_name;
        columns.viewstatus = value.View_Status.Value;
        Columnlst.Add(columns);
    }
    aclobj.Columnlist = Columnlst;
}
}
else if (count != 0)
{
    foreach (var value in x)
    {
        acl_columns columns = new acl_columns();
        columns.aclid = value.AC1_id.Value;
        columns.column_name = value.AC1_Column.Column_name;
        columns.viewstatus = value.View_Status.Value;
        Columnlst.Add(columns);
    }
    aclobj.Columnlist = Columnlst;
}
acclist.Add(aclobj);
}
//foreach (var per in role)
//{
//    objuser.rolename = per.User_Role_Name;
//}
```

0 %

Output

Show output from: Debug

```
iisexpress.exe' (CLR v4.0.30319; /LM/W3SVC/2/ROOT-1-132302222775672406): Loaded 'C:\Windows\Microsoft.Net\assembly\GAC_MSIL\System.Windows.Forms\v4.0_4.0.0.0__b77adcf6bbc1deae\System.Windows.Forms.dll'
iisexpress.exe' (CLR v4.0.30319; /LM/W3SVC/2/ROOT-1-132302222775672406): Loaded 'C:\Windows\Microsoft.Net\assembly\GAC_32\System.Data.OracleClient\v4.0_4.0.0.0__b77adcf6bbc1deae\System.Data.OracleClient.dll'
iisexpress.exe' (CLR v4.0.30319; /LM/W3SVC/2/ROOT-1-132302222775672406): Loaded 'EntityFrameworkDynamicProxies-UQP'.
```

```
2532
2533
2534         //      objuser.rolename = per.User_Role_Name;
2535         //}
2536
2537         objuser.acl = acllist;
2538         //      objuser.loginid = Convert.ToInt64(loginid);
2539         //      objuser.HedisTeamId = entity.HedisTeamId;
2540         //      objuser.MRATeamId = entity.MRATeamId;
2541         //      objuser.MRATeam = entity.MRATeam;
2542         //      objuser.HedisTeam = entity.HedisTeam;
2543
2544         //Theme tobj = new Theme
2545         //{
2546             //      colourcode = entity.Colour_Code,
2547             //      theme = entity.Theme,
2548             //      user_id = entity.User_ID
2549
2550         //};
2551         //if (entity.User_Role_ID == 1065)
2552         //{
2553             //      objuser.roleid = 35;
2554         //}
2555         //objuser.themeobj = tobj;
2556         //objuser.ipadetails = ipalist;
2557         return Request.CreateResponse(HttpStatusCode.OK, objuser);
2558
2559     }
2560
2561 }
```

## Output

```
Show output from: Debug
'iisexpress.exe' (CLR v4.0.30319: /LM/W3SVC/2/ROOT-1-13230222275672406): Loaded 'C:\Windows\Microsoft.NET\assembly\GAC_MSIL\System.Windows.Forms\v4.0_4.0.0.0__b77a5c561934e089\System.Windows.Forms.dll'
'iisexpress.exe' (CLR v4.0.30319: /LM/W3SVC/2/ROOT-1-13230222275672406): Loaded 'C:\Windows\Microsoft.NET\assembly\GAC_MSIL\System.Data.OracleClient\v4.0_4.0.0.0__b77a5c561934e089\System.Data.OracleClient.dll'
'iisexpress.exe' (CLR v4.0.30319: /LM/W3SVC/2/ROOT-1-13230222275672406): Loaded 'EntityFrameworkDynamicProxies-UQP'.
'iisexpress.exe' (CLR v4.0.30319: /LM/W3SVC/2/ROOT-1-13230222275672406): Loaded 'C:\Windows\Microsoft.NET\assembly\GAC_MSIL\System.Dynamic\v4.0_4.0.0.0__b77a5c561934e089\System.Dynamic.dll'
```

```
2552 //if (entity.User_Role_ID == 1065)
2553 //{
2554     //    objuser.roleid = 35;
2555 //}
2556 //objuser.themeobj = tobj;
2557 //objuser.ipadetails = ipalist;
2558 return Request.CreateResponse(HttpStatusCode.OK, objuser);
2559 }
2560
2561
2562
2563
2564
2565
2566
2567 catch (Exception e)
2568 {
2569     return Request.CreateResponse(HttpStatusCode.InternalServerError, e.InnerException);
2570 }
2571
2572
2573
2574
2575 }
2576 }
2577
2578
2579 }
```

100 %

Output

Show output from: Debug

```
iisexpress.exe (CLR v4.0.30319; /LM/W3SVC/2/ROOT-1-132302222775672406): Loaded C:\Windows\Microsoft.NET\Framework\v4.0.30319\Temporary ASP.NET Files\root\132302222775672406\1\{assemblyname}.dll
'iisexpress.exe' (CLR v4.0.30319; /LM/W3SVC/2/ROOT-1-132302222775672406): Loaded 'C:\Windows\Microsoft.NET\assembly\GAC_MSIL\System.Windows.Forms\v4.0_4.0.0.0__b77a5c561934e089'
'iisexpress.exe' (CLR v4.0.30319; /LM/W3SVC/2/ROOT-1-132302222775672406): Loaded 'C:\Windows\Microsoft.NET\assembly\GAC_32\System.Data.OracleClient\v4.0_4.0.0.0__b77a5c561934e089'
'iisexpress.exe' (CLR v4.0.30319; /LM/W3SVC/2/ROOT-1-132302222775672406): Loaded 'EntityFrameworkDynamicProxies-UQP'.
'iisexpress.exe' (CLR v4.0.30319; /LM/W3SVC/2/ROOT-1-132302222775672406): Loaded 'C:\Windows\Microsoft.NET\assembly\GAC_MSIL\System.Dynamic\v4.0_4.0.0.0__b77a5c561934e089'
The thread 0xc08 has exited with code 0 (0x0).
```