

F10MembersController.cs Source Control Explorer ACO_MembersMart.cs BulkEmailController.cs X Notification_UserSpecific.cs

UQP

```
1  using System;
2  using System.Collections.Generic;
3  using System.Data;
4  using System.Data.Entity;
5  using System.Data.Entity.Infrastructure;
6  using System.Linq;
7  using System.Net;
8  using System.Net.Http;
9  using System.Web;
10 using System.Web.Http;
11 using System.Web.Http.Description;
12 using UQP.Models;
13 using UQP.Utility;
14
15 namespace UQP.Controllers
16 {
17     [Authorize]
18     public class BulkEmailController : ApiController
19     {
20
21         [HttpGet]
22         [Route("api/providers/{type}/{id}")]
23         public HttpResponseMessage Providers(string type, int id)
24         {
25             try
26             {
27                 using (UQPEntities db = new UQPEntities())
28                 {
29                     db.Database.CommandTimeout = 240;
30                     //int year = 2018;
31                     //int month = 05;
32                     DateTime date = DateTime.Today.AddMonths(-1);
33                     //foreach (var item in db.providers)
34                     //    item.providers = db.providers
35                     //    .Where(p => p.provider_id == item.provider_id)
36                     //    .Select(p => p.provider_name).ToList();
37                     var provider = db.providers
38                     .Where(p => p.provider_id == id)
39                     .Select(p => p.provider_name).ToList();
40                     return Request.CreateResponse(HttpStatusCode.OK, provider);
41                 }
42             }
43             catch (Exception ex)
44             {
45                 return Request.CreateErrorResponse(HttpStatusCode.InternalServerError, ex.Message);
46             }
47         }
48     }
49 }
```

Output

Show output from: Source Control - Team Foundation

Project Build Debug Team Tools Test Analyze Window Help

Debug Any CPU IIS Express (Google Chrome)

BulkEmailController.cs* Source Control Explorer ACO_MembersMart.cs Notification_UserSpecific.cs

UQP.Controllers.BulkEmailController Providers(string type, int id)

```
{  
    db.Database.CommandTimeout = 240;  
    //int year = 2018;  
    //int month = 05;  
    DateTime date = DateTime.Today.AddMonths(-1);  
    List<Provider_bulkemail> lstprovider = new List<Provider_bulkemail>();  
  
    if (type == "healthplan")  
    {  
  
        //var xh = (id != 0) ? ((from m in db.Members_Mart.AsNoTracking()  
        //where m.Healthplan_id == id && m.TimeStamp.Year == date.Year && m.TimeStamp.Month == date.Month && m.provider_id  
        //!= null  
        //select new  
        //{
/>        //    m.ProviderFirstName,  
        //    m.provider_id  
        //}).Distinct()) : ((from m in db.Members_Mart  
        //where m.TimeStamp.Year == date.Year && m.TimeStamp.Month == date.Month && m.provider_id != null  
        //select new  
        //{
/>        //    m.provider_id  
        //}).Distinct());  
  
        var xh = (from p in db.ProviderDetailsTestings where p.visible==true  
        select new  
        {  
            p.providerid,  
            p.Physician_first_name,  
            p.Physician_last_name,  
            p.emailid  
        }).ToList().OrderBy(e=>e.Physician_last_name);  
  
        foreach (var value in xh)  
    }  
}
```

```
ier.cs          BulkEmailController.cs  Source Control Explorer      ACO_MembersMart.cs      Notification_UserSpecific.cs
                ↴ UQP.Controllers.BulkEmailController
                ↴ Providers(string type, int id)
})).ToList().OrderBy(e=>e.Physician_last_name);

foreach (var value in xh)
{
    //int temp = Convert.ToInt32(value.provider_id);
    //var x1 = (from p in db.ProviderDetailsTestings
    //           where p.providerid == temp
    //           select new
    //           {
    //               p.providerid,
    //               p.Physician_first_name,
    //               p.Physician_last_name,
    //               p.emailid
    //           }).First();

    // lstprovider.Add(x1);
    Provider_bulkemail provider = new Provider_bulkemail();
    provider.provideremail = value.emailid;
    provider.providerid = value.providerid;
    if (value.Physician_last_name != null)
    {
        provider.providername = value.Physician_last_name + " " + value.Physician_first_name;
    }
    else
    {
        provider.providername = value.Physician_first_name;
    }
    lstprovider.Add(provider);
}

}
else if (type == "ipa")
{
    var xp = (from m in db.Members_Mart.AsNoTracking()
              where m.TPA_id == id & m.Timestamp_Year == date.Year & m.Timestamp_Month == date.Month & m.provider_id != null)
```


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

- Title Bar:** Displays "UQP.Controllers.BulkEmailController" as the active file.
- Code Editor:** Shows C# code for the `BulkEmailController`. The code handles two cases: "confirmed" and "suspected". For each case, it queries the database for provider details, creates a `Provider_bulkemail` object, and adds it to a list. The "confirmed" case is straightforward, while the "suspected" case includes a condition to exclude providers who have already received notifications.
- Status Bar:** Shows "100 %".
- Output Window:** Shows the message "Show output from: Source Control - Team Foundation".

```
125     provider.providername = x1.Physician_first_name;
126 }
127
128         lstprovider.Add(provider);
129     }
130 }
131
132 else if (type == "suspected")
133 {
134     var x1 = (from p in db.ProviderDetailsTestings.AsNoTracking()
135               where p.status == true && p.visible==true && !(from g in db.Chroniccondition_provider_email select g.providerid).Contains(p.providerid)
136               select new
137                 {
138                     p.providerid,
139                     p.Physician_first_name,
140                     p.Physician_last_name,
141                     p.emailid
142                 }).ToList();
143
144     foreach (var value in x1)
145     {
146         Provider_bulkemail provider = new Provider_bulkemail();
147         provider.provideremail = value.emailid;
148         provider.providerid = value.providerid;
149         if (value.Physician_last_name != null)
150         {
151             provider.providername = value.Physician_first_name + " " + value.Physician_last_name;
152         }
153         else
154         {
155             provider.providername = value.Physician_first_name;
156         }
157         lstprovider.Add(provider);
158     }
159 }
```

```
155
156     provider.providername = value.Physician_first_name;
157     }
158     lstprovider.Add(provider);
159   }
160 }
161 return Request.CreateResponse(HttpStatusCode.OK, lstprovider);
162 }
163 catch (Exception e)
164 {
165   return Request.CreateResponse(HttpStatusCode.BadRequest, e);
166 }
167 }
168
169
170
171 [HttpGet]
172 [Route("api/healthplans")]
173 References:
174 public HttpResponseMessage healthplans()
175 {
176   try
177   {
178     using (UQPEntities db = new UQPEntities(HttpContext.Current.Request.Headers.GetValues("IPA").FirstOrDefault()))
179     {
180       List<CustomHealthPlan> healthplanlst = new List<CustomHealthPlan>();
181       var x = (from h in db.HealthPlans.AsNoTracking()
182               select new
183               {
184                 h.HealthPlan_ID,
185                 h.HealthPlan_Name
186               }).Distinct();
187       foreach (var value in x)
188       {
189         CustomHealthPlan healthPlan = new CustomHealthPlan();
190         healthPlan.health_Name = value.HealthPlan_Name;
191         healthPlan.healthPlan_id = value.HealthPlan_ID;
192       }
193     }
194   }
195 }
```

Input
Now output from: Source Control - Team Foundation

- Microsoft Visual Studio 192.61.99.225

View Project Build Debug Team Tools Test Analyze Window Help

Debug Any CPU IIS Express (Google Chrome)

ersController.cs BulkEmailController.cs Source Control Explorer ACO_MembersMart.cs Notification_UserSpecific.cs

UQP.Controllers.BulkEmailController Providers(string type, int id)

```
88     CustomHealthPlan healthPlan = new CustomHealthPlan();
89     healthPlan.health_Name = value.HealthPlan_Name;
90     healthPlan.healthplan_id = value.HealthPlan_ID;
91     healthplanlst.Add(healthPlan);
92 }
93 return Request.CreateResponse(HttpStatusCode.OK, healthplanlst);
94 }
95 }
96 catch (Exception e)
97 {
98     return Request.CreateResponse(HttpStatusCode.InternalServerError, e);
99 }
100 //protected override void Dispose(bool disposing)
101 //{
102 //    if (disposing)
103 //    {
104 //        db.Dispose();
105 //    }
106 //    base.Dispose(disposing);
107 //}
108
109 //Chronic condition email
110
111 [HttpPost]
112 [Route("api/suspectedemail")]
113 }
```

The screenshot shows a Microsoft Visual Studio interface. The main window displays a C# code editor for a file named `BulkEmailController.cs`. The code is a `[HttpPost]` method for handling suspected emails. It uses Entity Framework to interact with a database named `UQPEntities`. The code checks if a group with the same name already exists. If it does, it returns an error response. If it doesn't, it creates a new `Chroniccondition_group` object with the provided email details and adds it to the database. It then retrieves the group's ID and performs some additional database operations involving `EmailDomains` and `Chroniccondition_provider_email` tables.

```
//Chronic condition email
[HttpPost]
[Route("api/suspectedemail")]
public HttpResponseMessage suspectedemail(Suspected_email email)
{
    try
    {
        using (UQPEntities db = new UQPEntities(HttpContext.Current.Request.Headers.GetValues("IPA").FirstOrDefault()))
        {
            var query = (from c in db.Chroniccondition_group where c.group_name == email.groupname select c.group_name).ToList();
            if (query.Count() != 0)
            {
                return Request.CreateResponse(HttpStatusCode.InternalServerError, "A group already exist with the " + email.groupname + " name");
            }
            else
            {
                Chroniccondition_group group = new Chroniccondition_group
                {
                    group_name = email.groupname,
                    email_subject = email.subject,
                    email_body = email.body,
                    from_email = email.fromemail
                };
                db.Chroniccondition_group.Add(group);
                db.SaveChanges();
                long groupid = group.group_id;

                //var statuschange = (from e in db.EmailDomains where e.emailid == email.fromemail select e).FirstOrDefault();
                //statuschange.status = true;
                //db.SaveChanges();

                List<Chroniccondition_provider_email> provideremaillist = new List<Chroniccondition_provider_email>();
                foreach (var value in email.providerid)
                {
                    Chroniccondition_provider_email provideremail = new Chroniccondition_provider_email
                    {
                        providerid = value
                    };
                    provideremaillist.Add(provideremail);
                }
                db.Chroniccondition_provider_email.AddRange(provideremaillist);
                db.SaveChanges();
            }
        }
    }
}
```

Microsoft Visual Studio 192.61.99.225

File View Project Build Debug Team Tools Test Analyze Window Help

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

UQP

```
238 //var statuschange = (from e in db.EmailDomains where e.emailid == email.fromemail select e).FirstOrDefault();
239 //statuschange.status = true;
240 //db.SaveChanges();
241 
242     List<Chroniccondition_provider_email> provideremaillist = new List<Chroniccondition_provider_email>();
243     foreach (var value in email.providerid)
244     {
245         Chroniccondition_provider_email providerdetails = new Chroniccondition_provider_email
246         {
247             group_id = groupid,
248             providerid = Convert.ToInt32(value.providerid),
249             provideremail = value.provideremail,
250             providername = value.providername
251         };
252         provideremaillist.Add(providerdetails);
253     }
254     db.Chroniccondition_provider_email.AddRange(provideremaillist);
255     db.SaveChanges();
256     return Request.CreateResponse(HttpStatusCode.OK, "Group added successfully");
257 }
258 }
259 }
260 catch (Exception ex)
261 {
262     return Request.CreateResponse(HttpStatusCode.InternalServerError, ex);
263 }
264 }
265 }
```

Microsoft Visual Studio

192.61.99.225

Edit View Project Build Debug Team Tools Test Analyze Window Help

UQP

BulkEmailController.cs* | Source Control Explorer | ACO_MembersMart.cs | Notification_UserSpecific.cs

UQP.Controllers.BulkEmailController

192.61.99.225

```
259 }  
260 catch (Exception ex)  
261 {  
262     return Request.CreateResponse(HttpStatusCode.InternalServerError, ex);  
263 }  
264  
265 [HttpGet]  
266 [Route("api/emaildomain")]  
267 [References]  
268 public HttpResponseMessage suspectedfromemail()  
269 {  
270     try  
271     {  
272         using (UQPEntities db = new UQPEntities(HttpContext.Current.Request.Headers.GetValues("IPA").FirstOrDefault()))  
273         {  
274             List<string> emaildomain = new List<string>();  
275             var query = (from e in db.EmailDomains where e.status == true && !(from g in db.Chroniccondition_group select g.from_email).  
276                         Contains(e.emailid) select e.emailid).ToList();  
277             foreach (var value in query)  
278             {  
279                 emaildomain.Add(value);  
280             }  
281             return Request.CreateResponse(HttpStatusCode.OK, emaildomain);  
282         }  
283     }  
284     catch (Exception ex)  
285     {  
286         return Request.CreateResponse(HttpStatusCode.InternalServerError, ex);  
287     }  
288 }  
289  
290 [HttpGet]
```

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

UQP

UQP.Controllers.BulkEmailController

suspectedfromemail()

```
288
289
290
291     [HttpGet]
292     [Route("api/reports/grouplisting")]
293     public HttpResponseMessage grouplisting([FromUri]Suspected_email emailobj)
294     {
295         try
296         {
297             using (UQPEntities db = new UQPEntities(HttpContext.Current.Request.Headers.GetValues("IPA").FirstOrDefault()))
298             {
299                 List<Suspected_email> emailist = new List<Suspected_email>();
300                 var query = (from g in db.Chroniccondition_group.AsNoTracking()
301                             where (emailobj.groupname == null || g.group_name.StartsWith(emailobj.groupname))
302                             select new
303                             {
304                                 g.from_email,
305                                 g.group_id,
306                                 g.group_name
307                             }).ToList();
308                 if (query.Count != 0)
309                 {
310                     foreach (var value in query)
311                     {
312                         Suspected_email obj = new Suspected_email
313                         {
314                             fromemail = value.from_email,
315                             groupname = value.group_name,
316                             groupid = value.group_id
317                         };
318                         emailist.Add(obj);
319                     }
320                     return Request.CreateResponse(HttpStatusCode.OK, emailist);
321                 }
322             }
323         }
324     }
```

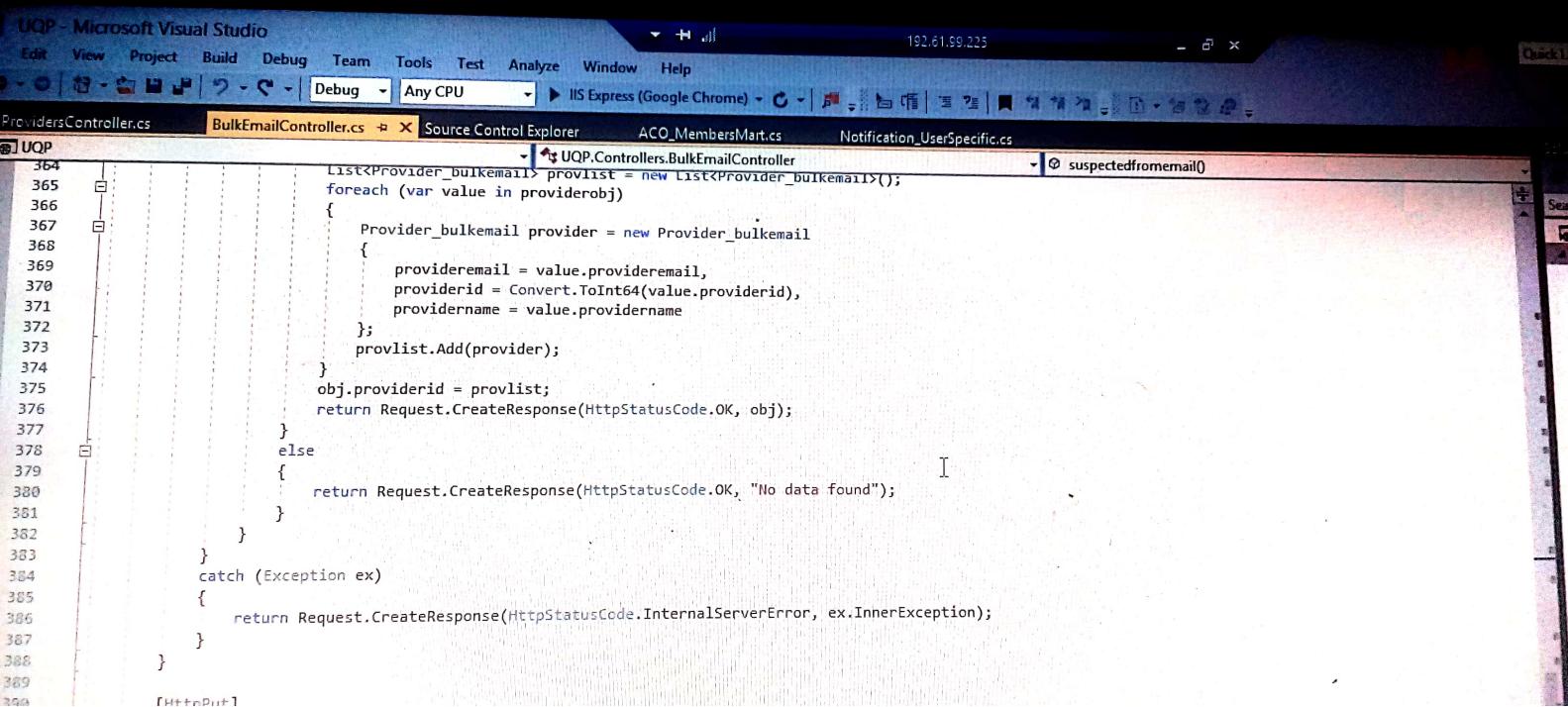
Scanned with CamScanner

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

- Menu Bar:** File, View, Project, Build, Debug, Team, Tools, Test, Analyze, Window, Help.
- Toolbars:** Standard, Debug, Task List, Solution Explorer, Properties, Task List, Status Bar.
- Status Bar:** IS2101.SS.220
- Code Editor:** Displays the file `BulkEmailController.cs`. The code is as follows:

```
302         g.from_email,
303         g.group_id,
304         g.group_name
305     }) .ToList();
306     if (query.Count != 0)
307     {
308         foreach (var value in query)
309         {
310             Suspected_email obj = new Suspected_email
311             {
312                 fromemail = value.from_email,
313                 groupname = value.group_name,
314                 groupid = value.group_id
315             };
316             emailist.Add(obj);
317         }
318         return Request.CreateResponse(HttpStatusCode.OK, emailist);
319     }
320     else
321     {
322         return Request.CreateResponse(HttpStatusCode.OK, emailist);
323     }
324 }
325 }
326 }
327 }
328 catch (Exception ex)
329 {
330     return Request.CreateResponse(HttpStatusCode.InternalServerError, ex.InnerException);
331 }
332 }
```

The code implements a controller action that retrieves a list of suspected emails from a database and returns them as a response. It handles both successful and empty queries and catches any exceptions that might occur during the process.



```
385         }
386         return Request.CreateResponse(HttpStatusCode.InternalServerError, ex.InnerException);
387     }
388 }
389
390 [HttpPost]
391 [Route("api/reports/editgroup")]
392 public HttpResponseMessage editgroup(Suspected_email email)
393 {
394     try
395     {
396         using (UQPEntities db = new UQPEntities(HttpContext.Current.Request.Headers.GetValues("IPA").FirstOrDefault()))
397     {
398         var query = (from g in db.Chroniccondition_group where g.group_id == email.groupid select g).FirstOrDefault();
399         if (query!=null)
400         {
401             if(query.from_email!=email.fromemail)
402             {
403                 var domainquery = db.EmailDomains.Where(a => a.emailid == query.from_email).FirstOrDefault();
404                 domainquery.status = false;
405                 var newdomainquery = db.EmailDomains.Where(a => a.emailid == email.fromemail).FirstOrDefault();
406                 newdomainquery.status = true;
407                 db.SaveChanges();
408             }
409             query.email_body = email.body;
410             query.email_subject = email.subject;
411             query.from_email = email.fromemail;
412             query.group_name = email.groupname;
413             db.SaveChanges();
414
415             var providerquery = db.Chroniccondition_provider_email.Where(e => e.group_id == email.groupid).ToList();
416             db.Chroniccondition_provider_email.RemoveRange(providerquery);
417             db.SaveChanges();
418
419             List<Chroniccondition_provider_email> providermaillist = new List<Chroniccondition_provider_email>();
420             foreach (var value in email.providerid)
```

Microsoft Visual Studio

File View Project Build Debug Team Tools Test Analyze Window Help

Debug Any CPU IIS Express (Google Chrome)

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

UOP UOP.Controllers.BulkEmailController

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

```
var providerquery = db.Chroniccondition_provider_email.Where(e => e.group_id == email.groupid).ToList();
db.Chroniccondition_provider_email.RemoveRange(providerquery);
db.SaveChanges();

List<Chroniccondition_provider_email> provideremaillist = new List<Chroniccondition_provider_email>();
foreach (var value in email.providerid)
{
    Chroniccondition_provider_email providerdetails = new Chroniccondition_provider_email
    {
        group_id = email.groupid,
        providerid = Convert.ToInt32(value.providerid),
        providermail = value.provideremail,
        providername = value.providername
    };
    provideremaillist.Add(providerdetails);
}
db.Chroniccondition_provider_email.AddRange(provideremaillist);
db.SaveChanges();
return Request.CreateResponse(HttpStatusCode.OK, "Status updated successfully");

}

else
{
    return Request.CreateResponse(HttpStatusCode.InternalServerError, "No group found with the requested group id :" + email.groupid);
}
}

}

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

Suspected_email.cs

```
442     catch(Exception ex)
443     {
444         return Request.CreateResponse(HttpStatusCode.InternalServerError, ex.InnerException);
445     }
446 }
447
448 [HttpPut]
449 [Route("api/reports/deletegroup")]
450 References
451 public HttpResponseMessage deletegroup(Suspected_email obj)
452 {
453     try
454     {
455         using (UQPEntities db = new UQPEntities(HttpContext.Current.Request.Headers.GetValues("IPA").FirstOrDefault()))
456         {
457             var query = db.Chroniccondition_group.Where(e => e.group_id == obj.groupid).ToList();
458             if (query.Count() != 0)
459             {
460                 db.Chroniccondition_group.RemoveRange(query);
461                 db.SaveChanges();
462
463                 var providerquery = db.Chroniccondition_provider_email.Where(e => e.group_id == obj.groupid).ToList();
464                 db.Chroniccondition_provider_email.RemoveRange(providerquery);
465                 db.SaveChanges();
466                 return Request.CreateResponse(HttpStatusCode.OK, "Group deleted successfully");
467             }
468             else
469             {
470                 return Request.CreateResponse(HttpStatusCode.OK, "No group found with the given group id");
471             }
472         }
473     catch (Exception ex)
474     {
475         return Request.CreateResponse(HttpStatusCode.InternalServerError, ex.InnerException);
476     }
477 }
```

```
454
455
456
457     using (UQPEntities db = new UQPEntities(HttpContext.Current.Request.Headers.GetValues("IPA").FirstOrDefault()))
458     {
459         var query = db.Chroniccondition_group.Where(e => e.group_id == obj.groupid).ToList();
460         if (query.Count() != 0)
461         {
462             db.Chroniccondition_group.RemoveRange(query);
463             db.SaveChanges();
464
465             var providerquery = db.Chroniccondition_provider_email.Where(e => e.group_id == obj.groupid).ToList();
466             db.Chroniccondition_provider_email.RemoveRange(providerquery);
467             db.SaveChanges();
468             return Request.CreateResponse(HttpStatusCode.OK, "Group deleted successfully");
469         }
470         else
471         {
472             return Request.CreateResponse(HttpStatusCode.OK, "No group found with the given group id");
473         }
474     }
475     catch (Exception ex)
476     {
477         return Request.CreateResponse(HttpStatusCode.InternalServerError, ex.InnerException);
478     }
479
480 }
481 }
```

