

OWIN's GetExternalLoginInfoAsync Always Returns null

[Ask Question](#)

▲
63
▼
★
15

I've created a new MVC5 Web Application, and when I try to login with Google or Facebook, the `ExternalLoginCallback` Action in the `AccountController` is called, but `GetExternalLoginInfoAsync()` always returns null:

```
var loginInfo = await AuthenticationManager.GetExternalLoginInfoAsync();  
if (loginInfo == null)  
{  
    return RedirectToAction("Login");  
}
```

Because it's always null, it just redirects back to the login page and the process starts over. How can I fix this?

[c#](#) [.net](#) [asp.net-mvc](#) [owin](#)
[google-openid](#)

edited Aug 17 '17 at 3:17



[Hossein Narimani Rad](#)
21.1k 11 64 97

asked Nov 4 '13 at 19:18



[VineetYadav](#)
350 1 3 6

Take a look at:

[asp.net/mvc/tutorials/mvc-5/...](#) I followed it yesterday with no problems – [Felipe Miosso](#) Nov 5 '13 at 19:32

Did you check whether correct provider is being passed to `ChallengeResult` of `ExternalLoginActionResult`? – [Santosh](#) Nov 11 '13 at 13:19

- 2 Experiencing this same problem. And yes the provider is correct. I click sign in, I'm redirected to facebook, I sign in, and then I'm redirected to the method... with a null loginInfo. –

[nVentimiglia](#) Nov 24 '13 at 21:36

- 1 Experiencing this same problem. It fixes the issue after I restarted the IIS. Is there a proper solution for this issue? – [Lee](#) Jan 5 '14 at 21:26
- 6 I've got the same issue, I forgot to turn on the Google +API – [sabotero](#) Dec 6 '14 at 21:11

14 Answers



97

To get OWIN Google login to work properly on a standard Visual Studio 2013, ASP.Net MVC5 site, I had to:



1. Setup a Google OpenId account at <https://console.developers.google.com/project>
2. Set the callback URL there to `blah/signin-google`.
Important notes on things you **don't** need to do:
 - You don't need to use HTTPS for Google to redirect back; you can even redirect back to plain <http://localhost>, no problem.
 - You don't need to setup anything for the redirect URL - no routes, Controller Actions or special permissions in Web.Config. The redirect URL is always `/signin-google` and OWIN handles this behind the scenes for you.

As an example, if your site was `me.com`, you might have these 3 callback URLs in the Google Developer Console:

`http://localhost:53859/signin-google`
`http://test.me.com/signin-google`
`https://me.com/signin-google`

The first one including whatever port number VS gave you for your project.

3. **Enable the Google+ API.** This is one hidden b**** of a gotcha and is the root cause of the problem in the question here - if you don't do this, it's easy to miss that the

Request to

/account/ExternalLoginCallback
 includes &error=access_denied ,
 and that's because Google said
 no to a permissions request
 OWIN made for the user's
 Google+ basic profile. I can't tell
 whose fault this is, Google's or
 Microsoft's.

To enable the Google+ API in the
 Developers Console, click APIs on
 the left, hunt for Google+, click that
 and hit Enable. Yes you really do
 need to do that. **You're hosed if you
 don't do that.**

4. Add the ClientId and ClientSecret
 Google gave you in the
 Developers Console to
 Startup.Auth, but improve the
 code in the process to explicitly
 use OAuth2, and explicitly ask
 for the user's email address:

```
var google = new GoogleOAuth2Auth
{
    ClientId = "123abc.apps.googleusercontent.com",
    ClientSecret = "456xyz",
    Provider = new GoogleOAuth2Auth
};
google.Scope.Add("email");
app.UseGoogleAuthentication(google);
```

That's it. That finally got it working.

Just want to reiterate one more time,
 there are a LOT of answers about this
 and issues like it where
 OWIN/Google isn't working, and
 nearly all of them are wrong for the
 current VS2013/MVC5/OWIN
 template.

You don't need to modify Web.Config
 at all.

You don't need to create any special
 Routes whatsoever.

You should not attempt to point
 /signin-google to a different place, or
 use a different callback URL, and you
 definitely shouldn't attempt to tie it
 directly to

/account/externallogincallback or
 externalloginconfirmation , because
 those are both separate from
 /signin-google and necessary steps
 in the OWIN/Google process.

edited May 30 '18 at 8:26

 Hakam Fostok



5,612 8 44 69

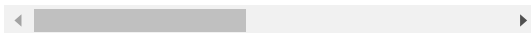
answered Apr 28 '15 at 13:50



Chris Moschini

26.5k 14 123 161

-
- 11 Number three did it for me. Sadly, I knew this to be the problem in the past but had forgotten it... and yes, totally agree this is a b****. Many thanks for the list. – [roboto1986](#) Jul 12 '15 at 12:08
-
- 4 Thank you, Thank you, Thank You! Worked, yes ENABLE Google+ API, it worked even without "code improvement". Great detailed answer! – [Pawel Cioch](#) Jul 28 '15 at 14:14 ✎
-
- 1 If I could upvote you 1000 times, I would. Thank you! – [Giovanni Galbo](#) Jan 17 '16 at 5:30
-
- 2 Doesn't work for me. It is actually impossible to create an appID and appSecret without enabling Google+ API. I have enabled it and was able to create the secret and id, but I am still being redirected back to the login page. – [tutiplain](#) May 14 '16 at 20:53
-
- 1 I have it enabled, but it still doesn't work. I am considering manually doing the Oauth flow, since this seems to be a bug in either Owin or the project templates in Visual Studio. – [tutiplain](#) May 17 '16 at 15:10
-



45 ▲ OK, I found out why it's null. You have to enable Google + API in the Google console. Also make sure the secret key is not concatenated with a space at the end after you paste it to your code. Why can't they return a normal error? I don't know.

[edited Jun 22 '16 at 22:10](#)

answered Dec 21 '14 at 1:03




Ronen Festinger

1,743 17 27

2 This immediately solved the problem

- This immediately solved the problem.
 Good find, thanks. – [James Mauldin](#)
 Jan 4 '15 at 14:52

Now that I look carefully, I see that this step is clearly indicated in the [instructions](#) on asp.net. – [James Mauldin](#) Jan 4 '15 at 16:51 

Yes, but I thought it's not obligatory. Anyway the connection between that and the error that is shown is not clear. – [Ronen Festinger](#) Jan 4 '15 at 16:57

Yes, I've seen all kinds of crazy answers but this is the correct one for me. – [Papa Burgundy](#) Apr 7 '15 at 15:09

You sir, are a god! – [Kris](#) Mar 18 '17 at 15:16

▲
19 It seems that Nuget package Microsoft.Owin.Security.Facebook version 3.0.1 no longer works with Facebook Login.
▼

Update this package to the pre-release 3.1.0 version, you can use the following:

Install-Package
Microsoft.Owin.Security.Facebook
-Pre

[edited Mar 31 '17 at 17:45](#)

answered Mar 29 '17 at 18:56



[Luke](#)

13.1k 21 82 158

Just came across this out of nowhere today as well and updating to the RC of 3.1.0 worked for me too. – [Chris Searles](#) Mar 30 '17 at 1:29

- That did it, thanks I owe you a beer :D
 – [Castaldi](#) Apr 6 '17 at 5:47
- Crikey, hope they fix this soon spent all night reconfiguring URLs trying to work out what I'd inadvertently changed since it was working last time I was playing around. Your suggestion worked. Still no updated version later than 3.0.1 to install according to NuGet
 – [Richard](#) Apr 8 '17 at 21:35
- After trying pretty much everything on stack overflow and wasting about 3

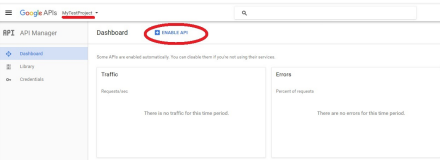
hours, this was the final solution for me. Thanks :) – Dan Cook Apr 20 '17 at 22:15

- 1
- Upgrading to 3.1 fixed my problem. Thanks for saving me many painful hours debugging this! – johnnyRose Jun 10 '17 at 3:57

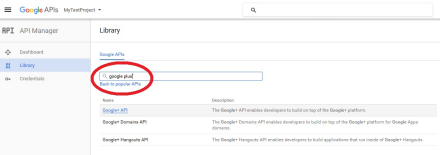
7

As others correctly mentioned, most of the time that's because you do not have permission to the Google+ API so here is how to get permission for a project in Google API Manager to Google+ API

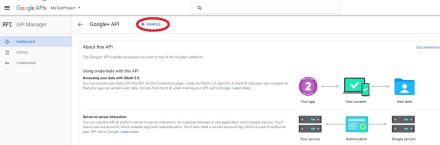
Step 1. Select You Project from the top combobox and go to Dashboard > Enable API



Step 2: Search for Google plus and select it



Step 3: Enable it!



if you return to dashboard for that project you can see the list of enabled API's for that project at the bottom



answered Aug 6 '16 at 15:10

Hossein Narimani Rad
21.1k 11 64 97

3

This solved my problem:
Enable the Google+ API. This is a gotcha and is the root cause of the problem in the question here - if you

don't do this, it's easy to miss that the Request to

/account/ExternalLoginCallback includes `&error=access_denied`, and that's because Google said no to a permissions request OWIN made for the user's Google+ basic profile. I can't tell whose fault this is, Google's or Microsoft's.

To enable the Google+ API in the Developers Console, click APIs on the left, hunt for Google+, click that and hit Enable.

edited Dec 12 '15 at 1:05



Undo ♦

23.1k 20 91 115

answered Dec 11 '15 at 23:02



Hugh Proctor

56 1

3

I got it to work by simply updating all the nugget package in the application and it worked.

answered May 31 '17 at 14:04



Salem BeeJay
Kosemani

61 6

3

I know it's silly, but after a long struggle, restarting IIS solved the issue for me.

answered Feb 8 '14 at 21:10



Omri

575 3 8 22

Same here but the problem returned after a while (days). It worked fine for some time then all of a sudden it just stops and a refresh of the app pool solves it again. Very weird. – Patrick Aug 14 '15 at 22:14

This solved the issue for me.
Hopefully it will keep working now.

stackoverflow.com/a/20948631/155758 – Patrick Aug 14 '15 at 22:18

I did the following to get it working.

- 1 Logon to the developer portal, locate your application and do the following.

**App details > App centered
Listed Platforms > Select
Yes for website**

answered Sep 12 '14 at 16:23



tony

124 6 22

I ran in to this issue today and it turned out that I defined the remote cookie after I assigned the providers.

Make sure you place...

```
app.UseExternalSignInCookie(DefaultAu
```

before...

```
app.UseFacebookAuthentication(  
    appId: "",  
    appSecret: "");
```

answered Oct 26 '16 at 20:16



Hagge

11 2

I wanted to contribute to this one also. I just recently got this working. I had the problem with the GetExternalLoginInfoAsync returning null but only in production.

After a lot of searching I finally found my answer it was simply a problem with my database. In production I had set the wrong connection string so it would not connect properly but it was basically silent about it. The only thing that happened was GetExternalLoginInfoAsync returned null. So check you database connection string if this happens!

Also on a sidenote, the only thing that was needed to get this working was:

- Set up a project in the Google console
- Enable Google+ API
- Copy your client id and client secret to the Startup.Auth.cs file.

You do not have to enable HTTPS, you do not have to create custom routes. But make sure your database is working properly!

answered Jan 19 '17 at 11:31



0 Although the answers above are all good, in my instance none of these worked - I'd checked and double checked the Google settings and agree with Chris Moschini that there's a lot of misleading info.

For me it was a 'doh moment when I realised that my Out of Process **state service was not started!** No errors (as a login was the first thing I was attempting after a reboot where the state service is set to manual start-up on the machine) just a Null from `GetExternalLoginInfoAsync`

Hope this helps someone else out.

answered Feb 15 '16 at 13:47



0 After much searching and head scratching as well as following numerous red herring answers here on Stackoverflow I eventually went through all my options on my Google dev console and discovered a little blue [Enable] button on the Google+API overview page. I clicked this and hey presto it worked. Forget all the baloney you read about callback url and route configs, OWIN overrides the google default /signin-google redirect uri in any case and sends you back to `ExternalLoginCallback`. Just stick with the default implementation all will be

good so long as you enable your Google+API.

answered May 12 '16 at 15:17



Graham Walker

11 2

▲
0

For those who are experiencing this problem for Web Api. Other solutions doesnt help

AuthenticationManager.GetExternalLoginInfoAsync(); returns always null even google plus api is enabled.

use this custom function to get logininfo. obviously Microsoft has a bug for GetExternalLoginInfoAsync when requesting over web api.

```
private async Task<ExternalLoginInfo>
AuthenticationManager_GetExternalLoginInfoAsync()
{
    ExternalLoginInfo loginInfo;

    var result = await
Authentication.AuthenticateAsync(DefaultAuthenticationTypes.BasicAuthenticationType);

    if (result != null && result.Identity.IsAuthenticated)
    {
        var idClaim = result.Identity.Claims.FirstOrDefault(c => c.Type == ClaimTypes.NameIdentifier);
        if (idClaim != null)
        {
            loginInfo = new ExternalLoginInfo
            {
                AuthenticationType = DefaultAuthenticationTypes.BasicAuthenticationType,
                IdentityName = result.Identity.Name.Replace(" ", "%20"),
                LoginProvider = "Basic",
                ProviderDisplayName = "Basic",
                UserId = idClaim.Value
            };
        }
    }
    return loginInfo;
}
```

answered May 23 '18 at 22:24



batmaci

2,411 3 30 51

◀ ▶

▲
-1

All of the other answers didn't solve this for me, so if your in the same boat then make sure your registration controller action has the RequireHttps attribute:

```
// GET: /Account/LoginRegister
[AllowAnonymous]
[RequireHttps]
public ActionResult LoginRegister
{
}
```

```
    }  
    return View(new RegisterLogin
```

answered Oct 10 '15 at 0:41



rman
7 2

