Error - is not marked as serializable

Asked 6 years, 5 months ago   Active 10 months ago   Viewed 110k times

▲

46

▼

★

6

The error I'm getting is:

```
Type 'OrgPermission' in Assembly 'App_Code.ptjvczom, Version=0.0.0.0, Culture=neutral,
PublicKeyToken=null' is not marked as serializable.
```

here is my code:

I have a gridview, that uses the following DataSource:

```
<asp:ObjectDataSource ID="ObjectDataSource1" runat="server" SelectMethod="GetOrgList"
        TypeName="Org">
    <SelectParameters>
      <asp:SessionParameter Name="orgCodes" SessionField="UserOrgs" Type="Object" />
       <asp:Parameter DefaultValue="Y" Name="active" Type="String" />
    </SelectParameters>
</asp:ObjectDataSource>
```

I set the session variable in my page load like so:

```
User cUser = new User(userid);
//make sure the user is an Admin
List<OrgPermission> orgs = new List<OrgPermission>();
foreach(OrgPermission org in cUser.orgs)
   {
     if (org.type=='admin')
     {
        orgs.Add(org);
     }
   }
Session["UserOrgs"] = orgs;
```

My user class looks like this:

```csharp
        public string Org { get; set; }
        public List<string> type { get; set; }

        public OrgPermission()
        { }
    }
    public class cUser
    {
        public string userid { get; set; }
        public List<OrgPermission> orgs { get; set; }

        public clsUser(string username)
        {
          //i set everything here
        }
    }
```

I can't understand why it's breaking, can I use it without making it serializable?

I tried to debug, and the session variable sets just fine, it then goes into the GetOrgList and returned correct results, but the page does not load and I get the error above.

Here is a snippet of my GetOrgList function:

```csharp
    public DataTable GetOrgList(List<OrgPermission> orgCodes, string active)
        {

            string orgList = null;

            //code to set OrgList using the parameter is here.

            DataSet ds = new DataSet();
            SqlConnection conn = new SqlConnection(cCon.getConn());
            SqlCommand cmd = new SqlCommand("sp_GetOrgList", conn);
            cmd.CommandType = CommandType.StoredProcedure;
            cmd.Parameters.Add(new SqlParameter("@orgList", orgList));
            cmd.Parameters.Add(new SqlParameter("@active", active));

                conn.Open();
                SqlDataAdapter sqlDA = new SqlDataAdapter();

                sqlDA.SelectCommand = cmd;
                sqlDA.Fill(ds);
```

```
        return ds.Tables[0];
    }
```

c#   asp.net   .net   serialization

edited Mar 20 '17 at 11:16   asked Mar 29 '13 at 17:01

Raghavendra   Madam Zu Zu
687   2   13   20   2,705   15   63   115

3 Answers

```
[Serializable]
public class OrgPermission
```

136

answered Mar 29 '13 at 17:03

burning_LEGION
11.5k   8   33   47

1   that did it. thank you! i guess i still have ways to learn. – Madam Zu Zu Mar 29 '13 at 17:07

20   Just as important is *why* you must add that serializable tag: any object put into a session variable (other than base objects like ints and bools) must be marked as serializable. Note that some .NET classes are not by default - i.e. DataView. – Paul Oct 31 '14 at 17:07

     I have stored my List type object type in Viewstate and it showed me the same error when I tried to use it as DataSource for a repeater control. Does it also apply for the Viewstate along with Session State ? – sohaiby May 28 '15 at 8:47

1   Didn't work for me; I think this is being caused by something else for me.This error doesn't always appear. But when it does, there's nothing I can do.Can't rebuild or save. I'm left to re-start VS and loose work. – n00dles Jul 23 '17 at 20:54

If you store an object in session state, that object must be serializable.

http://www.hpenterprisesecurity.com/vulncat/en/vulncat/dotnet/asp_dotnet_bad_practices_non_serializable_object_stored_in_session.ht

edit:

In order for the session to be serialized correctly, all objects the application stores as session attributes must declare the [Serializable] attribute. Additionally, if the object requires custom serialization methods, it must also implement the ISerializable interface.

https://vulncat.hpefod.com/en/detail?
id=desc.structural.dotnet.asp_dotnet_bad_practices_non_serializable_object_stored_in_session#C%23%2fVB.NET%2fASP.NET

edited Aug 24 '17 at 17:11

answered Mar 29 '13 at 17:19

nimeshjm
1,507   8   13

1   Not for mode = InProc – Roland Jan 6 '16 at 19:28

2   The page that you requested could not be found. – JoshYates1980 Aug 24 '17 at 16:24

Neither page you requested can be found – Hoppe Oct 17 '18 at 19:14

Leaving my specific solution of this for prosperity, as it's a tricky version of this problem:

1

```
Type 'System.Linq.Enumerable+WhereSelectArrayIterator[T...] was not marked as serializable
```

Due to a class with an attribute `IEnumerable<int>` eg:

```
[Serializable]
class MySessionData{
    public int ID;
    public IEnumerable<int> RelatedIDs; //This can be an issue
}
```

Originally the problem instance of `MySessionData` was set from a non-serializable list:

```
MySessionData instance = new MySessionData(){
    ID = 123,
    RelatedIDs = nonSerizableList.Select<int>(item => item.ID)
};
```

The cause here is the concrete class that the `Select<int>(...)` returns, **has type data that's not serializable**, and you need to copy the id's to a fresh `List<int>` to resolve it.

```
RelatedIDs = nonSerizableList.Select<int>(item => item.ID).ToList();
```

answered Oct 26 '18 at 10:19

Jono
3,051 3 21 44