

خیلی وقت‌ها لازم است تا نتیجه کوئری حاصله را بصورت Json به ویوی مورد نظر ارسال نمایید. برای اینکار کافیتست مانند زیر عمل کنیم

```
[HttpGet]
public JsonResult Get(int id)
{
    return Json(repository.Find(id), JsonRequestBehavior.AllowGet);
}
```

اما اگر کوئری پیچیده و یا یک [مدل سلسله مراتبی](#) داشته باشید که با خودش کلید خارجی داشته باشد، هنگام تبدیل نتایج به خروجی Json، با خطای Circular References مواجه می‌شوید.

A circular reference was detected while serializing an object of type
'System.Data.Entity.DynamicProxies.ItemCategory_A79...'

علت این مشکل این است که Json Serialization پیش فرض ASP.NET MVC فقط یک سطح پایین‌تر را لود می‌کند و در مدل‌های که خاصیتی از نوع خودشان داشته باشند خطای Circular References را فرا می‌خواند. کلاس نمونه در زیر آورده شده است.

```
public class Item
{
    public int Id { get; set; }
    [ForeignKey]
    public int ItemId { get; set; }
    public string Name { get; set; }
    public ICollection<Item> Items { get; set; }
}
```

راه حل:

چندین راه حل برای رفع این خطا وجود دارد؛ یکی استفاده از [Automapper](#) و راه حل دیگر استفاده از کتابخانه‌های قوی‌تر کار بار Json مثل [Json.net](#) است. اما راه حل ساده‌تر تبدیل خروجی کوئری به یک شی بی نام و سپس تبدیل به Json می‌باشد

```
[HttpGet]
public JsonResult List()
{
    var data = repository.AllIncluding(itemcategory => itemcategory.Items);
    var collection = data.Select(x => new
    {
        id = x.Id,
        name = x.Name,
        items = x.Items.Select(item => new
        {
            id=item.Id,
            name = item.Name
        })
    });
    return Json(collection, JsonRequestBehavior.AllowGet);
}
```

همین طور که در مثال بالا مشاهده می‌نمایید ابتدا همه رکوردها در متغیر data ریخته شده و سپس با یک کوئری دیگر که در آن دوباره از پروپرتی items که از نوع کلاس item می‌باشد شی بی نامی ایجاد نموده ایم. با این کار براحتی این خطا رفع می‌گردد.

نظرات خوانندگان

نویسنده: محسن خان
تاریخ: ۱۳۹۲/۰۹/۱۸ ۰:۴۹

با تشکر. یک سؤال: آیا تنظیم `context.Configuration.ProxyCreationEnabled = false` قبل از نوشتن کوئری Find (بلافاصله پس از ایجاد context) مشکل را حل می‌کند؟

نویسنده: مجتبی کاویانی
تاریخ: ۱۳۹۲/۰۹/۱۸ ۱۶:۲

خیر؛ این خطا مربوط به Json Serialization می‌باشد. ProxyCreation برای مباحث Lazy Loading و Change Tracking کاربرد دارد.

نویسنده: Ara
تاریخ: ۱۳۹۲/۰۹/۲۷ ۲۳:۹

سلام؛ راست می‌گند. اگه شما یک ابجکت رو مستقیم از dbcontext بگیرید و بدون اون که lazyloading غیر فعال باشه بدین به serializer تمام روابط اون ابجکت هم سریالایز می‌شوند که خیلی مشکل زاست حتی با json دات نت و اگر اون شی با شی دیگه که اون هم با این شی رابطه داشته باشه تو Cycle می‌افته و بهترین روش همونی بود که دوستمون گفتند یا استفاده از viewModel یا DTO هاست.

نویسنده: محمد
تاریخ: ۱۳۹۳/۰۶/۱۲ ۱۸:۵

سلام وخسته نباشید . من تو اینترنت سرچ کردم توی stack گفته بودند که اگه به صورت عمومی غیر فعالش کنی هم میشه. این کد رو هم گفته بودند تو قسمت Application_Start بزارید درست میشه

```
GlobalConfiguration.Configuration.Formatters.JsonFormatter.SerializerSettings.ReferenceLoopHandling =
    Newtonsoft.Json.ReferenceLoopHandling.Serialize;
GlobalConfiguration.Configuration.Formatters.JsonFormatter.SerializerSettings.PreserveReferencesHandling =
    Newtonsoft.Json.PreserveReferencesHandling.Objects;
```

ولی برای من نشد. من میخوام به طور عمومی طوری تنظیمش کنم که اگه جایی به circular برخورد کرد بیخیالش بشه و ارور نده. آیا راهی وجود داره؟

نویسنده: محسن خان
تاریخ: ۱۳۹۳/۰۶/۱۲ ۱۸:۱۸

GlobalConfiguration.Configuration.Formatters مربوط به Web API هست. برای MVC باید return Json توکار رو با نمونه Newtonsoft.Json در همه جا تعویض کنید.

نویسنده: محمد
تاریخ: ۱۳۹۳/۰۶/۱۲ ۱۸:۲۸

خیلی ممنون . میشه به نمونه کد یا سایت یا چیزی برام بزارید که من دقیقا بدونم چیرو کجا و چجوری تغییر بدم؟ کاری که من خودم کرده بودم این بود که از کلاس JsonResult یک کلاس دیگه ساخته بودم که ازش ارث می‌برد و بعد با تنظیمات ReferenceLoopHandling.Ignore متدExecute اون رو override کردم . جواب هم داد . فقط یه گیری داشت .اونم اینکه من تو مدلم یک فیلدی دارم که از نوع Byte[] هستش . و توش فایل هامو نگه میدارم . تو حالتی که اولیه خودش من بالای این فیلد [ScriptIgnore] گذاشته بودم و خوب کار میکرد . اما وقتی با این کلاس جدیدم اونو serelize میکنم همه چیزو serelize میکنه و

این باعث شده خیلی کند بشه . یه راهنمایی بکنید که یا حالت اول باشه ولی ارور circular نده یا حالت دوم باشه ولی فیلدهای باینری رو serelize نکنه . ممنون میشم کمک کنید .

نویسنده: محسن خان
تاریخ: ۱۸:۳۵ ۱۳۹۳/۰۶/۱۲

در Newtonsoft.Json برای صرفنظر کردن از یک خاصیت، یا از ویژگی IgnoreDataMember استفاده کنید یا از ویژگی JsonIgnore آن.