

Related data and serialization

Article • 01/06/2023

Because EF Core automatically does fix-up of navigation properties, you can end up with cycles in your object graph. For example, loading a blog and its related posts will result in a blog object that references a collection of posts. Each of those posts will have a reference back to the blog.

Some serialization frameworks don't allow such cycles. For example, Json.NET will throw the following exception if a cycle is found.

```
Newtonsoft.Json.JsonSerializationException: Self referencing loop detected for property 'Blog' with type 'MyApplication.Models.Blog'.
```

System.Text.Json will throw a similar exception if a cycle is found.

```
System.Text.Json.JsonException: A possible object cycle was detected. This can either be due to a cycle or if the object depth is larger than the maximum allowed depth of 32. Consider using ReferenceHandler.Preserve on JsonSerializerOptions to support cycles.
```

If you're using Json.NET in ASP.NET Core, you can configure Json.NET to ignore cycles that it finds in the object graph. This configuration is done in the `ConfigureServices(...)` method in `Startup.cs`.

C#

```
public void ConfigureServices(IServiceCollection services)
{
    ...

    services.AddMvc()
        .AddJsonOptions(
            options => options.SerializerSettings.ReferenceLoopHandling =
            Newtonsoft.Json.ReferenceLoopHandling.Ignore
        );

    ...
}
```

If you're using System.Text.Json, you can configure it like this.

C#

```
public void ConfigureServices(IServiceCollection services)
{
    ...

    services.AddControllers()
        .AddJsonOptions(options =>
        {
            options.JsonSerializerOptions.ReferenceHandler =
            ReferenceHandler.IgnoreCycles;
        });

    ...
}
```

Another alternative is to ignore the navigation properties that cause the cycle for JSON serialization. If you're using Json.NET, you can decorate one of the navigation properties with the `[JsonIgnore]` attribute, which instructs Json.NET to not traverse that navigation property while serializing. For System.Text.Json, you can use the `[JsonIgnore]` attribute in the `System.Text.Json.Serialization` namespace to achieve the same effect.

Collaborate with us on GitHub

The source for this content can be found on GitHub, where you can also create and review issues and pull requests. For more information, see [our contributor guide](#).



.NET feedback

.NET is an open source project. Select a link to provide feedback:

 [Open a documentation issue](#)

 [Provide product feedback](#)