

# Custom Serialization

---



**Xavier Morera**

HELPING DEVELOPERS UNDERSTAND SEARCH & BIG DATA

@xmorera [www.xaviermorera.com](http://www.xaviermorera.com)



# Custom Serialization

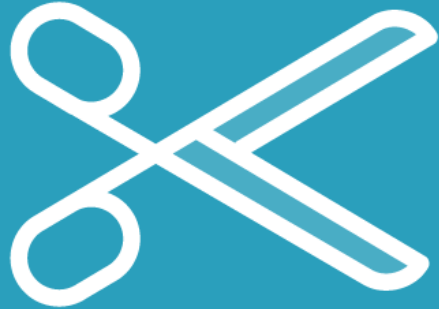


1

2

3

Conditional  
Serialization



Custom  
JsonConverter



Callbacks



ITraceWriter



# Conditional Serialization

Select dynamically what to serialize and what not based on certain conditions

**ShouldSerialize**

**IContractResolver**



```
public string[] Courses { get; set; }  
  
public bool ShouldSerializeCourses()  
{  
    return IsActive;  
}
```

## ShouldSerialize

### Conditionally serialize properties

- Placing a **ShouldSerialize** method
- Within a class



# Demo



Controlling which properties are  
serialized with `ShouldSerialize`



```
private IList<string> propertiesList = null;
```

```
public SelectiveContractResolver(IList<string> propertiesToSerialize)  
{  
    propertiesList = propertiesToSerialize;  
}
```

## IContractResolver

**Another method for conditional serialization**

- The IContractResolver

**Create a list with all properties that should be serialized**



# Demo



Controlling which properties are  
serialized with `DataContractResolver`



# Custom JsonConverter

```
{
  "name": "Xavier Morera",
  "courses": [
    "Solr",
    "dotTrace"
  ],
  "since": "2014-01-14T00:00:00",
  "happy": true,
  "issues": null,
  "car": {
    "model": "Land Rover Series III",
    "year": 1976
  }
}
```

**Create your own  
converter**

**Derived from  
JsonConvert**

**Override methods as  
needed**

```
Author xavierAuthor = new Author()
{
  name = "Xavier Morera",
  courses = new string[]
  {
    "Solr",
    "JIRA",
    "dotTrace"
  }
  happy = true,
  issues = null
};
```





```
public class RemoveNullsJsonConverter : JsonConverter
{
    ...
}
```

## Custom JSON Converters

The **JsonConverter** class is responsible for serialization and deserialization

**Possible to extend**

- Create a custom **JsonConverter**



# Demo



## Creating a custom JsonConverter



# Serialization Callbacks



**Methods Executed**

**Before & After**

**Serialization / Deserialization**



# Serialization Callbacks

**Before**

**OnSerializing**

**OnDeserializing**

**After**

**OnSerialized**

**OnDeserialized**



# Serialization Callbacks

**Before**

**OnSerializing**

**OnDeserializing**

**After**

**OnSerialized**

**OnDeserialized**



[OnSerializing]

```
internal void OnSerializingMethod(StreamingContext context)
{
```

[OnSerialized]

```
internal void OnSerializedMethod(StreamingContext context)
{
```

## Serialization Callbacks

**Manipulate objects before and after**

- Serialization and deserialization

### Callbacks

- With **OnSerializing**, **OnSerialized**, **OnDeserializing**, and **OnDeserialized**



# Demo



## Serialization callbacks



# Debugging with ITraceWriter



What if you want to debug?

Or understand better the serialization process?

Or you are running into errors but can't figure out why...

Use ITraceWriter





# Debugging with ITraceWriter

**Memory  
TraceWriter**

**Custom  
TraceWriter**



```
ITraceWriter traceWriter = new MemoryTraceWriter();
```

## TraceWriter

### Logging and debugging

- Using **ITraceWriter**

Capture serialization messages and errors



# Demo



## Debugging with Serialization Tracing using ITraceWriter



```
public class FileLogTraceWriter : ITraceWriter
{
    private string fileName;
    private TraceLevel _traceLevel;
}
```

## Custom TraceWriter

Create your own custom **TraceWriter**

- Can control **TraceLevel**



# Demo



## Creating a custom TraceWriter



# Takeaway



## Conditional Serialization

- ShouldSerialize or IContractResolver

## Custom JsonConvert

## Callbacks

## ITraceWriter

