.NET CORE 3.1（好像是）已集成System.Text.Json，不需要额外安装nuget包

**基本序列化和反序列化**

如下代码

TestClass testClass = new TestClass()

{

    Name = "Ice",

    Age = 18,

    Sex = true

};

// 序列化

var json = System.Text.Json.JsonSerializer.Serialize(testClass);

Console.WriteLine(json);

// 反序列化

var obj = System.Text.Json.JsonSerializer.Deserialize<TestClass>(json);

Console.WriteLine(obj.Name);

**UTF8序列化和反序列化**

如下代码

// UTF8 序列化和反序列化

byte[] jsonUtf8Bytes = JsonSerializer.SerializeToUtf8Bytes(testClass);

TestClass objUtf8 = JsonSerializer.Deserialize<TestClass>(jsonUtf8Bytes);

Console.WriteLine(objUtf8.Name);

**行为控制**

Json主要通过JsonSerializerOptions控制行为，如下：

var options = new JsonSerializerOptions

{

    // 无视大小写

    PropertyNameCaseInsensitive = true,

    // 小驼峰命名

    PropertyNamingPolicy = JsonNamingPolicy.CamelCase,

    // 忽略只读属性

    IgnoreReadOnlyProperties = true,

    // 忽略循环引用

    ReferenceHandler = ReferenceHandler.Preserve,

};

// 序列化

var json = System.Text.Json.JsonSerializer.Serialize(testClass, options);

Console.WriteLine(json);

// 反序列化

var obj = System.Text.Json.JsonSerializer.Deserialize<TestClass>(json, options);

Console.WriteLine(obj.Name);

忽略某个属性

public class TestClass

{

    ...

    // 忽略 NOField

    [JsonIgnore]

    public string NOField { get; set; }

}