Twitter Thread by Oleg Kyrylchuk

Oleg Kyrylchuk
@okyrylchuk

A thread of System.Text.Json features in the .NET 6. ■

Retweets are appreciated ■■

#dotnet6 #coding #devcommunity #json #programming

Ignore circular references.
https://t.co/2sAx2HNZDc

> In .NET 5, you can preserve references for circular references using System.Text.Json.
>
> But you couldn't ignore them. The 'JsonException' is thrown if circular references have been detected.
>
> In .NET 6, you can ignore them \U0001f642#dotnet6 #json #coding #devcommunity pic.twitter.com/FnvR8gjwvN
>
> — Oleg Kyrylchuk (@okyrylchuk) October 6, 2021

Notifications for (de)serialization.
https://t.co/1EyOaF4p3q

> In .NET 6, System.Text.Json exposes notifications for (de)serialization.
>
> There are four new interfaces to implement according to your needs:
> \u2022 IJsonOnDeserialized
> \u2022 IJsonOnDeserializing
> \u2022 IJsonOnSerialized
> \u2022 IJsonOnSerializing#dotnet6 #json #coding #devcommunity pic.twitter.com/iYpzHsg8Hy
>
> — Oleg Kyrylchuk (@okyrylchuk) October 7, 2021

Serialization order of properties.
https://t.co/Y6XyaWLqAR

In .NET 6, the 'JsonPropertyOrderAttribute' has been added to System.Text.Json.

It allows controlling the serialization order of properties.

Previously, the serialization order was determined by reflection order.#dotnet6 #json #coding #devcommunity pic.twitter.com/5sBJRWOvcX

— Oleg Kyrylchuk (@okyrylchuk) October 8, 2021

Write raw JSON with Utf8JsonWriter.
https://t.co/nvHyJtnLPM

.NET 6 introduces the possibility to write 'raw' #json with System.Text.Json.Utf8JsonWriter.

Useful when you want:
\u2022 to enclose existing JSON in new JSON
\u2022 to format values differently from the default formatting#dotnet6 #coding #devcommunity pic.twitter.com/NguR3wEOAM

— Oleg Kyrylchuk (@okyrylchuk) October 16, 2021

IAsyncEnumerable support.
https://t.co/iBjgBXEX6U

In .NET 6, System.Text.Json supports 'IAsyncEnumerable'.

The serialization of 'IAsyncEnumerable' transforms it into an array.

For deserialization of root level JSON Arrays, the 'DeserializeAsyncEnumerable' method has been added.#dotnet6 #json #coding #devcommunity pic.twitter.com/Dnc4uU3AbJ

— Oleg Kyrylchuk (@okyrylchuk) October 19, 2021

Serialize and deserialize JSON data to/from a stream.
https://t.co/cN0M9DIEEp

You couldn't serialize and deserialize JSON data to/from a stream using System.Text.Json in .NET 5.

But in .NET 6, you can do that \U0001f642#dotnet6 #json #coding #devcommunity pic.twitter.com/FmiBY4FREL

— Oleg Kyrylchuk (@okyrylchuk) October 20, 2021

Work with JSON like a DOM.
https://t.co/bvf2Zq8un7

.NET 6 provides types for handling an in-memory writeable document object model (DOM) for random access of the JSON elements within a structured view of data.

New types:
\u2022 JsonArray
\u2022 JsonNode
\u2022 JsonObject
\u2022 JsonValue#dotnet6 #json #coding #devcommunity pic.twitter.com/pAUIAlKV1g

— Oleg Kyrylchuk (@okyrylchuk) October 21, 2021