

JSON

VALTTERI VARTIAINEN

05/12/2024

Mikä on JSON?

- ▶ JSON eli JavaScript Object Notation on kevyt ja helposti luettava tiedonvälitysmuoto, joka on laajalti käytössä verkkosovelluksissa.
- ▶ JSON esittää tietoa avain-arvopareina, jotka voivat olla yksinkertaisia arvoja esim. Merkkijonoja, numeroita tai monimutkaisempia rakenteita, kuten taulukoita tai sisäkkäisiä objekteja.
- ▶ JSON on yleinen valinta API-yhteyksissä, konfiguraatiotiedostoissa ja tiedon tallennuksessa sen yksinkertaisuuden ja joustavuuden vuoksi.
- ▶ Esimerkki JSON-muodosta:
 - ▶ `{ "name": "John Doe", "age": 30, "isDeveloper": true, "languages": ["C#", "JavaScript", "Python"] }`

JSON C#:ssa

- ▶ C#:ssa JSON-tietojen käsittely on sujuvaa, koska .NET tarjoaa sisäänrakennettuja työkaluja, kuten **System.Text.Json**, joka mahdollistaa JSON-datan jäsentämisen, muodostamisen ja tiedostojen käsittelyn tehokkaasti.

JSON:n jäsentäminen

```
1  using System;
2  using System.Text.Json;
3
4  2 references
5  public class Person
6  {
7      1 reference
8      public string Name { get; set; }
9      1 reference
10     public int Age { get; set; }
11     1 reference
12     public bool IsDeveloper { get; set; }
13 }
14
15 0 references
16 class Program
17 {
18     0 references
19     static void Main()
20     {
21         string json = @"{ ""name"": ""John Doe"", ""age"": 30, ""isDeveloper"": true }";
22         Person person = JsonSerializer.Deserialize<Person>(json);
23         Console.WriteLine($"Name: {person.Name}, Age: {person.Age}, Is Developer: {person.IsDeveloper}");
24     }
25 }
```

JSON:n luominen

```
1  using System;
2  using System.Text.Json;
3
4  var person = new { Name = "Alice", Age = 25, IsDeveloper = false };
5  string json = JsonSerializer.Serialize(person, new JsonSerializerOptions { WriteIndented = true });
6  Console.WriteLine(json);
7  
```