

JSON NEDİR?

JSON (*Javascript Object Notation*) (*Javascript nesne gösterimi*), bütün programlama dilinden bağımsız olan, XML'e (*Extensible Markup Language*) (*genişletilebilir işaretleme diline*) alternatif olarak üretilmiş, javascript tabanlı veri değişim formatıdır. EcmaScript dil şartnamesinde tanımlandığı gibi Javascript'te bulunan nesne değişmezlerinden esinlenilerek tasarlanmıştır. JSON'ın (*Javascript Object Notation*) (*Javascript nesne gösteriminin*) temel amacı veri alış verişi yaparken daha küçük boyutlarda veri alıp göndermektir. Temel olarak Javascript programlama dilinin standart Ecma-262 3. yayın sürümünün alt kümesi üzerine kurulmuştur fakat programlama dilinden bağımsızdır. Javascript'in alt kümesi üzerine kurulu olduğu için javascript işlemlerinde XML'e (*Extensible Markup Language*) (*genişletilebilir işaretleme diline*) göre daha hızlıdır, daha az yer kaplar ve okunabilirliği daha kolaydır.

C türevi (C, C++, C#, Java, Javascript, Perl, Python vs.) dillere benzeyen yazılımından dolayı, bu tür dillerde rahatlıkla işlenebilir. JSON (*Javascript Object Notation*) (*Javascript nesne gösterimi*) sayesinde hızlı ve küçük boyutlarda veri oluşturulabilmektedir. JSON (*Javascript Object Notation*) (*Javascript nesne gösterimi*) sahip olduğu ağaç yapısıyla DOM (*Document Object Model*) (*belge nesne modeli*) ve XML (*Extensible Markup Language*) (*genişletilebilir işaretleme dili*) ile benzerlik gösterse de bunlardan tamamen farklı bir yapıdadır. Bu yapılar, evrensel veri yapılarıdır. Günümüzde bütün modern programlama dilleri, bu yapıları, bir şekilde içlerinde barındırmaktadırlar. Programla dilleri arasında veri değişimi için kullanılan bir formatın, bu yapıları kullanarak oluşturulması da oldukça anlamlıdır.

JSON (*Javascript Object Notation*) (*Javascript nesne gösterimi*) tamamıyla Unicode (*evrensel kod*) olarak çalışır. Varsayılan olarak UTF-8 (*Unicode Transformation Format*) (*evrensel dönüşüm formatı*) karakter setini kullanır. Karakter seti problemi yaşanılmayacak veri aktarım formatlarından biridir. JSON (*Javascript Object Notation*) (*Javascript nesne gösterimi*) herhangi bir sürüm numarasına sahip değildir. Çünkü grameri kesindir, tamamıyla oturmuş ve düzgün çalışan bir yapıdadır. Ayrıca hemen hemen tüm dillerin veri aktarımları için kullanabildiği ve desteklediği bir formattır. Postel'in Kabul ederken esnek ol, gönderirken tutucu ifadesine uygun yapısı vardır. Bu yüzden JSON (*Javascript Object Notation*) (*Javascript nesne gösterimi*) yorumlayıcılar JSON (*Javascript Object Notation*) (*Javascript nesne gösterimi*) harici formatları kabul eder, ama JSON (*Javascript Object Notation*) (*Javascript nesne gösterimi*) oluşturucuların sağlıklı çalışması için geçerli bir JSON (*Javascript Object Notation*) (*Javascript nesne gösterimi*) ifadesi oluşturması şarttır. Bu JSON'ın (*Javascript Object Notation*) (*Javascript nesne gösteriminin*) genişletilmesine izin verir. Örnek vermek gerekirse Javascript, JSON'ın (*Javascript Object Notation*) (*Javascript nesne gösteriminin*) bir üst kümesidir ve doğal olarak JSON (*Javascript Object Notation*) (*Javascript nesne gösterimi*) desteğine sahiptir.

JSON (*Javascript Object Notation*) (*Javascript nesne gösterimi*) günümüzde web uygulamalarında, web servislerde, API'lerde (*Application Programing Interfaces*) (*uygulama programlama arayüzlerinde*), mobil uygulamalarda, Java uygulamalarında, .NET uygulamalarında, PHP (*Hypertext Preprocessor*) (*üstün yazı ön işlemcisi*) (*Personal Home Page*) (*kişisel ana sayfa*) uygulamalarında, ASP (*Active Server Pages*) (*aktif sunucu sayfaları*) uygulamalarında, veri transferlerinde vs. birçok ortamda kullanılmaktadır.

JSON (*Javascript Object Notation*) (*Javascript nesne gösterimi*) verileri birbirlerine { } (*süslü parantez açma - kapatma*) işareti ile bağlanır ve JSON Object (*Javascript Object Notation Object*) (*Javascript nesne gösterimi nesnesi*) olarak adlandırılır. JSON (*Javascript Object Notation*) (*Javascript nesne gösterimi*) söz dizimi kurallarında, veriler, (*virgül*) işareti ile ayrılmalı, diziler [] (*köşeli parantez açma - kapatma*) işareti ile ve nesneler ise { } (*süslü parantez açma - kapatma*) işareti ile belirtilmelidir. JSON (*Javascript Object Notation*) (*Javascript nesne gösterimi*) verileri key (*anahtar*) ve value (*değer*) çiftlerinden oluşur ve istenildiği kadar key (*anahtar*) ve value (*değer*) ikilisi tanımlanabilir.

```
{ key:value }
```

Veriler oluşturulurken key'ler (*anahtarlar*) string (*alfa numerik karakter dizisi*) veri türünde olması gerekirken, değerler string (*alfa numerik karakter dizisi*), number (*numara, rakam veya sayı*), boolean (*mantıksal*), array (*dizi*), object (*nesne*) veya null (*boş / geçersiz*) veri türünde olabilir.

String (*alfa numerik karakter dizisi*) değer kullanımı

```
{ "Adi": "Volkan", "Soyadi": "Alakent" }
```

Number (*numara, rakam veya sayı*) değer kullanımı

```
{ "DogumTarihi": "1980" }
```

Boolean (*mantıksal*) değer kullanımı

```
{ "OnayDurumu": true }
```

Array (*dizi*) değer kullanımı

```
{ "Gunler": ["Pazartesi", "Salı", "Çarşamba", "Perşembe", "Cuma", "Cumartesi", "Pazar"] }
```

Object (*nesne*) değer kullanımı

```
{ "Bilgi": { "HTML", "CSS", "PHP", "Javascript", "jQuery", "AngularJS", "NodeJS", "XML", "Java", "C", "C++", "Python", "Ruby", "vs." } }
```

Array (*dizi*) ve Object (*nesne*) değer kullanımı

```
{ "Bilgi": [ { "Adi" : "HTML", "Seviye" : "Uzmanlık"}, { "Adi" : "CSS", "Seviye" : "Uzmanlık"}, { "Adi" : "PHP", "Seviye" : "Uzmanlık" } ] }
```

Null (*boş / geçersiz*) değer kullanımı

```
{ "YasadigiUlke": "", "YasadigiSehir": null }
```