



in/poyrazavsever



NEXT.JS CACHE

# Next.js Cache Mantiğına Derin Bakış



in/poyrazavsever



## Öncellikle Cache Nedir?

Cache, hesaplama maliyeti yüksek veya sık kullanılan verilerin geçici olarak saklanmasıdır.

Amaç, aynı işlemi tekrar tekrar yapmadan daha hızlı sonuç üretmektir.

Doğru kullanıldığında sistem yükünü azaltır, yanlış kullanıldığında ise tutarsız veri üretir.



in/poyrazavsever



## Neden Cache Kullanırız?

Bir web uygulamasında aynı sayfa, aynı veri, aynı API sonucunun her istekte yeniden hesaplanması gereksizdir.

Cache şu sorunları çözer:

- **Gereksiz server maliyetleri**
- **Yüksek response süreleri**
- **Trafik arttığında performans kaybı**

Bu nedenle cache, modern web'in görünmez ama kritik katmanıdır.



in/poyrazavsever



## Cache Sistemi Neden Bu Kadar Önemli?

Next.js, performansı artırmak için her isteği yeniden hesaplamak yerine veriyi ve sayfayı akıllıca saklar.

Doğru cache stratejisi, uygulamayı gereksiz render maliyetinden ve kaynak tüketiminden kurtarır.



in/poyrazavsever



## Üç Temel Cache Türü

Next.js üç farklı cache yaklaşımı kullanır:

- **Full Route Cache**
- **Data Cache**
- **Request Memoization**

Her biri farklı bir problemi çözer ve birlikte çalışarak uygulamayı hızlandırır.



in/poyrazavsever



## Full Route Cache

Statik olarak üretilebilen tüm sayfalar build sırasında oluşturulur ve CDN üzerinde saklanır.

Sayfa tamamen cache'ten gelir, server'a dokunmaz.

Bu sayfa türü değiştirildiğinde invalidate işlemi çalışır.



in/poyrazavsever



## Data Cache

fetch çağrılarının üzerinde durur.

Aynı parametrelerle yapılan fetch işlemleri tekrar çalıştırılmaz; sonuç cache'ten gelir.

Cache süresi revalidate değeri ile yönetilir.



in/poyrazavsever



## dynamic vs force-dynamic

**dynamic** sayfalar koşula göre server'da hesaplanabilir.

**force-dynamic** ise her istekte server'ın çalışmasını zorunlu kılar.

Bu ayar, sayfanın cache davranışını doğrudan belirler.





in/poyrazavsever



## Revalidate Mantığı

**revalidate** fetch sonucunun geçerlilik süresini tanımlar.

Süre dolunca ilk gelen istek veriyi yeniler, sonraki istekler yeni veriyi alır.

Bu yaklaşım edge tarafında oldukça verimlidir.



in/poyrazavsever



## Cache'in Yan Etkileri

Yanlış cache kullanımı:

- **Güncel olmayan veri**
- **Gereksiz server yükü**
- **Tutarsız UI durumları gibi problemlere yol açabilir.**

Davranışı doğru anlamak, doğru cache yönetimi anlamına gelir.



in/poyrazavsever



## Server Actions ile Cache Etkileşimi

Server actions, mutation işlemlerinde ilgili route ve data cache alanlarını otomatik olarak geçersiz kılar.

Bu sayede manuel “update etme” ihtiyacı azalır ve veri tutarlılığı korunur.



**Sen ne düşünüyorsun?**

Bugün projenin data akışını gözden geçir; hızlanması gereken yerler belki de cache ayarlarında saklıdır.



in/poyrazavsever



@poyraz\_avsever



@poyrazavsever