**Backbone Mimarisi**

Backbone mimarisi, yazılım geliştirmede uygulamanın farklı işlevlerini düzenli katmanlara ayıran, modüler, bakımı kolay bir yapı sağlayan bir tasarım mimarisidir. Modüler tasarım prensiplerine dayanır,katmanlı mimari yapısını benimser,bileşenler arası gevşek bağlılık sağlar ve yeniden kullanılabilirlik,sürdürülebilirlik odaklıdır.

**🔹** Backbone (omurga) ifadesi, uygulamanın taşıyıcı iskeletini kurduğu anlamına gelir.

**🔹** Çoğunlukla MVC (Model-View-Controller) prensiplerine dayanır, kod karmaşasını azaltır ve sürdürülebilirliği artırır.

**İhtiyaç duyulan durumlar:**

* Büyük projelerde kodu düzenli tutma
* Ekiplerin paralel ve hızlı geliştirme yapabilmesi
* Veri (iş mantığı) ve arayüzün ayrılması
* Tekrarlanan kodun azaltılması
* Test edilebilirlik ve sürdürülebilirlik

**Backbone Mimarisi Bileşenleri:**

**Model**

* Veri ve iş mantığını barındırır (ör: kullanıcı verileri, hesaplama işlemleri).
* Veriyi saklar, işler ve gerektiğinde sunucu ile senkronize eder.

### View (Görünüm)

* Kullanıcıya gösterilen arayüzdür.
* Modelde bir değişiklik olduğunda kendini günceller.
* Kullanıcıdan gelen olayları kontrol edebilir.

**Collection:**

* Birden fazla modelin bir arada tutulduğu yapıdır.
* Collection, model gruplarını yönetmek için kullanılır ve genellikle bir dizi model üzerinde toplu işlemler yapma yeteneği sağlar.

**Router(Yönlendirici):**

* URL yönetimini sağlar.
* Kullanıcının uygulama içi URL değişikliklerini algılar ve gerekli işlemleri başlatır.
* SPA mimaride URL’ye göre hangi görünümün açılacağını belirler.

**Events:**

* Backbone, model ve view'lar arasında iletişim sağlamak için bir olay sistemi kullanır. Bu, uygulamanın farklı bileşenleri arasında gevşek bir bağlılık sağlar.
* Olay tabanlı bir yapı sunar.
* Modeller, görünümler ve koleksiyonlar arası iletişimi sağlar.

**Nasıl Çalışır? :**

1. Kullanıcı bir butona tıklar (View tetiklenir)
2. View, modeli günceller (örneğin formdan gelen veriyi modele aktarır).
3. Model güncellenince event tetiklenir.
4. View bu olayı dinleyip arayüzü günceller.
5. Router sayesinde URL değişikliği varsa uygun View yüklenir.
6. Veriler API üzerinden çekilecekse Collection üzerinden sunucu ile iletişim kurulur.

**Avantajları:**

Kodun modüler olmasını sağlar.

Bakım ve test süreçlerini kolaylaştırır.

SPA (tek sayfa uygulama) mimarisi için uygundur.

Kod tekrarını azaltır.

## **Nerelerde Kullanılır?**

📌 **Single Page Application (SPA)** geliştirmede.  
📌 Karmaşık frontend uygulamalarda.  
📌 RESTful API ile çalışan frontend mimarilerde.  
📌 Backbone.js ile geliştirilen uygulamalarda.  
📌 **Veri ve arayüz ayrışması gerektiren** büyük projelerde.

**Ozan Yılmaz**