

Backbone.js

Javascript MVC Framework

Fatih Kadir AKIN



İlkel Web Programlama

- ✦ Görsellik için HTML
- ✦ Tüm yükün serverda olduğu zamanlar
- ✦ Her request sonrası tüm veri geri geliyor
- ✦ Response boyutlarının çok büyük olması

Ve AJAX Tekniđi Keşfedilir...

- ✦ Response boyutlarında eskiye göre küçölme
- ✦ Yenilenmeyen daha efektif sayfalar
- ✦ Kullanılabilirlik yüksek

Ama...

- ✦ Arama motorları bu sayfaları göremiyor.
- ✦ Linkler paylaşılabılır değil.
- ✦ Veri boyutları ne kadar küçülse de yeteri kadar küçük değil.
- ✦ AJAX ile gelen veriyi işlemek çok zor, server'da işleyip client'a yollamak en basit seçenek.

Yeni çözümler...

- ✦ JSON'un gücü keşfedildi. Javascript çok rahat parse edebiliyor.
- ✦ JSON Veri boyutu çok düşük.
- ✦ jQuery/JS Ajax üzerinden gelen JSON verileri çok rahat işleyebilir halde.

Hashbang (#!/route)

- ✦ Paylaşılabilir linkler oluşturabiliyor.
- ✦ onhashchange event handler ile tetiklenebiliyor.

Hashbang yeterli mi?...

- Hashbang'ler paylaşılabilirlik sorununu çözsse de hala arama motorları tarafından çalıştırılmıyor.
- Server hashbang URL'leri göremiyor.

JSON Veri ve HTML sorunları

- Veri çok rahat işlenebiliyor, fakat gelen veriyi HTML'e çevirmek adeta işkence.
- Sayfalar dolusu HTML gömülü JS dosyaları.
- Okunabilirliği düşük Javascript element creatorler.
- HTML okunabilirliği ve düzenlenebilirlik neredeyse imkansız gibi...

Server-side MVC'ler...

- MVC Framework'ler popüler olmaya başladı.
- Okunabilirliği ve paylaşılabilirliği yüksek URL'ler. (/controller/action/data/)
- Yönetilebilir veri
- Tekrar kullanılabilirlik oranı yüksek.

Javascript MVC?

- ✦ HTML5 History PushState geliştirildi.
- ✦ Server side MVC'yi taklit ediyor.
- ✦ View'lar bir kere yükleniyor, düzenlenebilir ve parse edilebilir.
- ✦ MVVM tasarım şablonu çok kullanışlı.

Nasıl?

- ✦ Veriler hala JSON. Çünkü server taraftan gelen en ideal veri elde edildi.
- ✦ Pushstate ile server'ın parse edeceği URL'i Javascript üstlenip servera düşük boyutlu veri gönderebiliyor ve alabiliyor.
- ✦ Model + Collection mantığı kodu düzenliyor.
- ✦ Model ve Collection JSON veri ile çok uyumlu.

Adapte olmak...

- ✦ Front-end'de Model alışılmamış bir kavram.
- ✦ Tasarımda neyin birim model, neyin birim view olduğunu doğru saptamak.
- ✦ “Tekrar eden verilerin tümü Collection, her biri ise bir Model” gibi...

Model, Collection Events

- ✦ Eventlar yönetilebiliyor.
- ✦ “Yeni model eklendiğinde” (collection event: add)
- ✦ “Bir model silindiğinde” (collection event: destroy)
- ✦ “Property değiştiğinde” (model event: change)
- ✦ gibi eventlar ile yönetim sağlanabiliyor.

Backbone.sync

- ✦ Server ile iletişimi eventlar ile kuruyor.
- ✦ Server'a gerekli olduğunda veri gönderip veri çekebiliyor.
- ✦ Collection'lar serverda muhattap olacakları yeri belirleyebiliyor.
- ✦ RESTful bir davranış sergiliyor.
- ✦ Her collection'un server'a nasıl davranacağını belirleyebiliyor.

Okunabilir View Template'lar

- ✦ `<%= degisken %>` gibi kendisine has bir yöntemle veriler okunabiliyor.
- ✦ Düzenlenebilirlik çok iyi.
- ✦ Okunabilirlik çok iyi.

MVM Şablonu

- ✦ Model - View - View - Model
- ✦ Bir model ve kendi view'inin arasındaki iletişim.
- ✦ Template View'a göre otomatik düzenlenebiliyor.

Özet ve Uygulama Örneği

- ✦ v1: İlkel web programlama (Server: ~99%, Client: ~1%)
- ✦ v2: AJAX'ın keşfi (Server: ~80%, Client: ~20%)
- ✦ v3: JSON'un keşfi (Server: ~50%, Client: ~50%)
- ✦ v4: Front-end ve Back-end MVC'ler. (Server: ~40%, Client: ~60%)