

Open in app ↗

Sign up

Sign in

Medium

Search



3 min read · Jul 12, 2026



Talhakurnaz

Follow



Share

## Tarayıcıdan Sunucuya Köprü: jQuery ile Backend İletişimi

### 1. Giriş

jQuery bir backend aracı değil aslında. Tarayıcıda çalışan bir JavaScript kütüphanesi, hepsi bu. Ama AJAX sayesinde uzun yıllar boyunca frontend ile backend'i birbirine bağlayan en pratik araçlardan biri oldu. Gelin bakalım, jQuery sunucuyla nasıl konuşuyor, ne işe yarıyor, hâlâ kullanmaya değer mi.

### 2. AJAX Ne İş Yapar?

Basit aslında. Sayfa yenilenmiyor, arkada bir istek gidiyor, cevap geliyor, sadece gereken kısım güncelleniyor. Sepete ürün atarken, yorum yazarken, arama kutusuna bir şey yazdıkça sonuçlar değişirken hep bu oluyor.

Eskiden bunu yapmak baya zahmetliydi. Her tarayıcı kendi bildiğini okuyordu, kod uzayıp gidiyordu, bir yerde bir hata çıkıyordu. jQuery bu işi çok basitleştirdi. Tek bir komutla isteği gönderip cevabı yakalayabiliyordunuz.

### 3. İnsanlar jQuery'yi Neden Sevdi?

jQuery'nin web geliştiricileri tarafından bu kadar çok sevilmesinin arkasında birkaç önemli sebep yatıyor:

- **Basitlik:** İsteği nereye göndereceğinizi, ne taşıyacağını, cevap gelince ne yapacağınızı birkaç satırda yazabiliyordunuz.
- **Tarayıcı Uyumu:** O dönem tarayıcılar birbirinden çok farklı davranıyordu, jQuery aradaki farkları kapattı, geliştirici bununla uğraşmak zorunda kalmadı.

- **Kod Okunabilirliği:** İstek başarılıysa ne olacak, olmazsa ne olacak, hepsi ayrı ayrı ve net yazılabiliyordu.
- **Hazır JSON Desteği:** Backend sistemler genelde bu formatta konuştuğu için veri alışverişi doğrudan ve kolayca gerçekleşiyordu.



#### 4. Backend'e Veri Gönderirken Nelere Takılıyorsunuz?

Get Talhakurnaz's stories in your inbox

Join Medium for free to get updates from this writer.

Enter your email

Subscribe

☐ Remember me for faster sign in

Veri genelde JSON oluyor, jQuery bunu rahat hazırlayıp gönderiyor. Ama güvenlik konusu devreye girince birkaç şeye dikkat etmek lazım:

- **Yetkilendirme (Authorization):** Bir API'ye girebilmek için genelde bir anahtar veya token gerekiyor, jQuery bunu isteğe eklemene izin veriyor.

- **CSRF (Cross-Site Request Forgery):** Formlar üzerinden yapılan işlemlerde sahte isteklere karşı sunucular bir güvenlik anahtarı istiyor, jQuery onu da isteğin içine koyabiliyor.
- **CORS (Cross-Origin Resource Sharing):** Backend başka bir alan adında çalışıyorsa tarayıcı isteği durdurabiliyor. Bunun çözümü aslında backend tarafında, jQuery sadece o ayarlara uyumlu çalışıyor.

## 5. Küçük Ama İşe Yarayan Detaylar

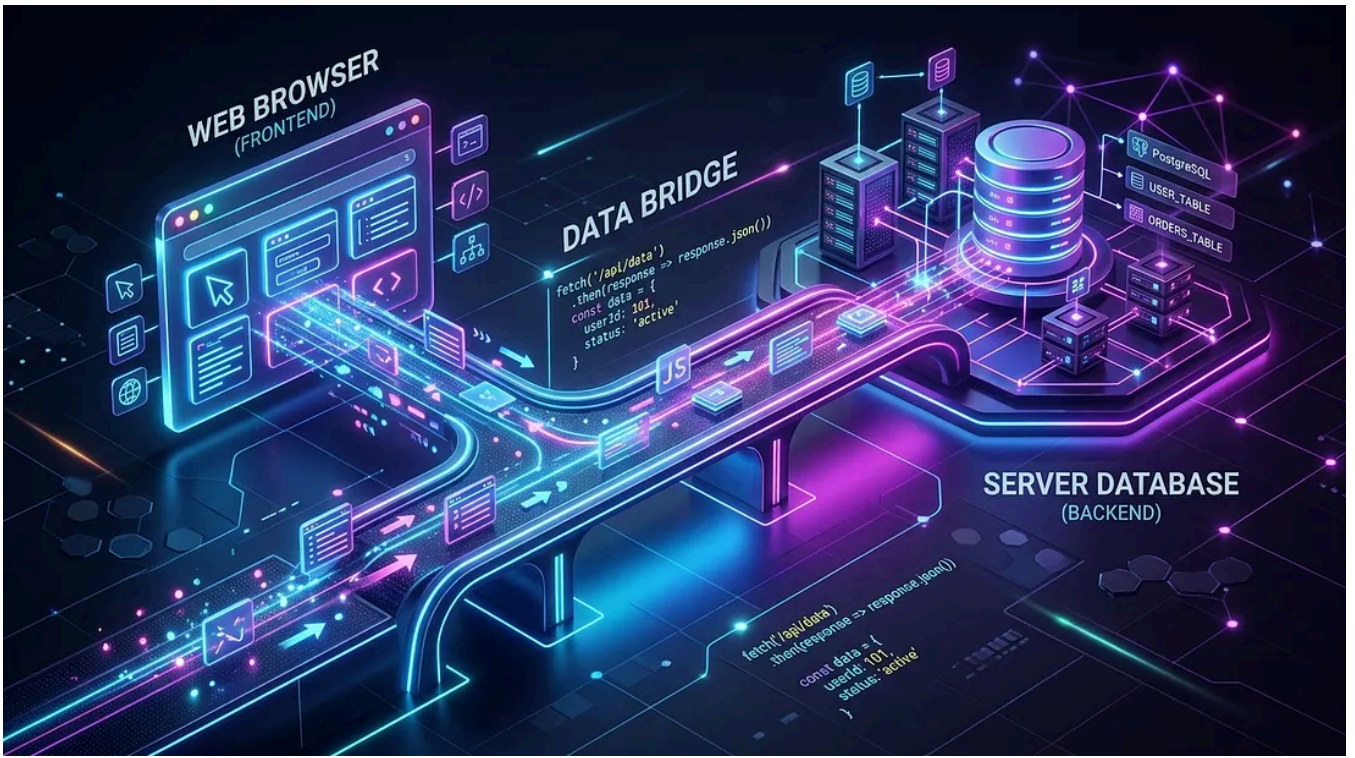
jQuery tek istekle uğraşmıyor, sayfadaki bütün AJAX trafiğini de yönetebiliyor. Mesela her isteğe aynı zaman aşımını ya da aynı kimlik bilgisini vermek istiyorsan bunu tek yerden ayarlayabiliyorsun. Sayfada bir istek başladığında yükleniyor yazısı gösterip bittiğinde kaldırmak da fazla uğraşmadan yapılabilir.

## 6. Peki Bugün Hâlâ Gerekli mi?

Açıkçası çoğu yerde artık değil. Tarayıcılar kendi başına benzer şeyleri yapabiliyor, axios gibi seçenekler de var. Modern JavaScript zaten zincirleme yapıyı kendinde barındırıyor. jQuery'nin sayfa öğelerini yönetme kısmı da React, Vue, Angular gibi araçlar yanında eskisi kadar öne çıkmıyor.

Ama bitti demek de doğru olmaz. Eski projelerde hâlâ karşına çıkar. WordPress varsayılan olarak getirir. Küçük bir iş yapacaksan ya da hızlıca bir şey deneyeceksen jQuery bugün bile işini görür.

## 7. Sonuç



jQuery kendisi backend değildi ama yıllarca frontend'i backend'e bağlayan en önemli araçlardan biriydi. Karmaşık işleri basitleştirdi, daha az kodla daha sağlam işler yapmayı mümkün kıldı. Bugün yerini başka araçlara bıraksa da, eski sistemlerde hâlâ işe yarıyor.

[Follow](#)

**Written by Talhakurnaz**

0 followers · 1 following



