



Title

Rest API Versioning

. . .

Merhaba, bu yazımda Rest API versiyonlama nedir? Neden yapılır ? Gibi konulardan bahsetmeye çalışacağım. Hadi başlayalım.

. . .

API Versiyonlama Nedir ?

API'ları kısaca biz ve istemci arasında yapılmış olan bir sözleşmeye benzetebiliriz. Bu sözleşme iki tarafıda belirli bir süre bağlayıcı niteliktedir ve çıkışı pek mümkün değildir.

Peki ya sözleşmede bir değişiklik yapmak istersek ?

API'lar üzerinde bir değişiklik yapmak istediğimizde iki tarafada zarar vermeyecek şekilde bunu gerçekleştirmenin yolu mevcut API sürümüne dokunmadan yeni bir sürüm çıkartmak yani API Versioning (API Versiyonlama) yapmaktır.

Neden İhtiyaç Duyarız ?

En basitinden örnek vermek gerekirse bir property isminde değişiklik yapmak istediğimizi düşünelim. (productId → productID)

Böyle basit bir değişiklik bile istemci tarafına giden veri akışını kırmaya yetecektir. Fakat burada şöyle bir durum var. İhtiyaçlar günden güne değişebilir ve bizde API'larımız üzerinde bu ihtiyaçlara cevap vermek durumunda kalabiliriz. İşte bu gibi zorunlu hallerde “sözleşmemizi” ihlal etmemek adına versiyonlama yaparak işi herkes için kotarabiliriz.

Nasıl Yapılır ?

API Versiyonlamanın birkaç yolu vardır ve her tekniğin değişiklikleri ele almada kendi güçlü ve zayıf yönleri vardır.

URI Path Versioning:

URI tekniğini kullanmak, bir URI'nin benzersiz bir kaynağa referans göstermesi ilkesini ihlal etse de, en basit ve aynı zamanda en yaygın kullanılan yaklaşımdır. İstemci ile aramızdaki entegrasyonun bozulacağı garantidir çünkü path'i değiştirmiş oluyoruz.

```
http://www.example.com/api/v1/products
http://api.example.com/v1/products
```

Query Params Versioning:

Bu teknik, gelen isteğe (request) versiyonu belirten bir sorgu parametresi ekler. Kaynağın versiyonunu istemek açısından çok esnektir, ancak genel API'nin versiyonu hakkında kapalı kutudur. Yine, istemci ile aramızdaki entegrasyonun bozulması garantidir.

```
http://www.example.com/api/products?version=1
```

Header Versioning:

Versiyon farklılığını **Header** altındaki **Accept** alanını kullanarak sağlarız.

```
Accept: version=1.0
```

URI de versiyon belirtilmediği için farklı versiyonlarda dahi aynı URI yapısını korur ve URI değiştirilen versiyonlama tekniklerine göre çok daha temiz bir çözüm sunar.

. . .

Umarım bu makale yazılım geliştiricilerine bir nebze de olsa katkı sağlamıştır.

Başka bir yazıda görüşmek üzere.

İstek ve önerileriniz için bana batuhankiltac@gmail.com adresinden ulaşabilirsiniz.