

Web Service & Postman Öğrenme Notları (Learning Web Service & Postman Notes)

1. Web Service (What is Web Service?)

Tanım

Web service, internet üzerinden farklı sistemlerin **birbirleriyle konuşmasını** sağlayan bir yazılım bileşenidir. Yani bir uygulamanın başka bir uygulamadan veri alması veya işlem yapması için kullanılan köprü (bridge) gibi düşünebilirsiniz.

Amaç

- Uygulamaların **birbirinden bağımsız ama entegre** çalışmasını sağlamak.
- Veri paylaşımını standartlaştırmak (JSON, XML, HTTP gibi).
- Platform bağımsızlığı oluşturmak → Java, .NET, Python fark etmez, web service olduğu sürece haberleşebilirler.

Nerelerde Kullanılır?

- Banka uygulamalarında havale/EFT işlemlerinde.
- E-ticaret sitelerinde ödeme sistemleri (PayPal, lyzico, Stripe API'leri).
- Harita servisleri (Google Maps API).
- Sosyal medya entegrasyonları (Twitter API, Instagram API).

1.1 API (What is API?)

API'nin Açılımı

Application Programming Interface → Uygulamaların konuşma dilini tanımlar.

Kullanım Amacı

- Bir yazılımın diğerine "şu bilgiyi ver" veya "şunu yap" diye **talimat göndermesi**.
- Örneğin mobil uygulama → backend API → database şeklinde zincir oluşur.

Günlük Yazılım Dünyasındaki Örnekler

- Yemeksepeti uygulaması restoran verilerini API üzerinden çeker.
- Hava durumu uygulaması meteoroloji API'sinden veri alır.
- Spotify, YouTube gibi servisler üçüncü parti uygulamalara API ile erişim verir.

1.2 SOAP vs REST API (What are SOAP and REST API?)

SOAP (Simple Object Access Protocol)

- **Tanım:** Eski tip, XML tabanlı mesajlaşma protokolü.
- **Özellikler:** Katı kurallar, XML zorunluluğu, güvenlik ve transaction desteği yüksek.
- **Avantajları:** Yüksek güvenlik (WS-Security), ACID transaction desteği.
- **Dezavantajları:** Aşırı verbose (fazla XML), yavaş çalışır.

REST (Representational State Transfer)

- **Tanım:** HTTP protokolü üzerinde çalışan, JSON/XML kullanan, günümüzde en popüler API mimarisi.
- **Özellikler:** Basit, hızlı, cache kullanılabilir, HTTP metodlarını kullanır.
- **Avantajları:** JSON desteği sayesinde hızlı, mobil uyumlu.
- **Dezavantajları:** Transaction ve güvenlik SOAP kadar güçlü değil.

Karşılaştırma Tablosu

Özellik	SOAP	REST
Veri Formatı	XML	JSON, XML, Text, HTML
Hız	Daha yavaş	Daha hızlı
Güvenlik	Yüksek (WS-Security)	Orta (HTTPS ile destek)
Esneklik	Düşük	Yüksek
Kullanım Alanı	Finans, Bankacılık	Web & Mobil uygulamalar

1.3 HTTP Methods (Learning HTTP Methods)

- **GET:** Sunucudan veri çekmek için (örn: `/users` → kullanıcı listesini getir).
- **POST:** Sunucuya yeni veri eklemek için (örn: yeni kullanıcı kaydet).
- **PUT:** Var olan veriyi güncellemek için (tam güncelleme).
- **PATCH:** Var olan veriyi kısmi güncellemek için (örneğin sadece e-mail güncelle).
- **DELETE:** Sunucudaki veriyi silmek için.
- **OPTIONS:** API'nin desteklediği HTTP metodlarını öğrenmek için.

2. Postman Tool (Postman Aracı)

2.1 Postman Kullanımı (Using of Postman)

- **Arayüzü:** Sol tarafta collection, ortada request, sağda response alanı var.
- **Collection Mantığı:** API isteklerini klasörler halinde saklarsın (örneğin: User API, Product API).
- **Environment Değişkenleri:** URL, token, kullanıcı bilgilerini değişken yaparak test ortamı ↔ canlı ortam arasında kolay geçiş sağlarsın.

2.2 API Testlerinde Postman Kullanımı (How to use Postman in API testing?)

- **Request Oluşturma:** URL girilir, metod (GET/POST/PUT/DELETE) seçilir.
- **Parametre Gönderme:** Query Params (URL sonuna eklenen), Body (JSON/XML payload).
- **Header Ekleme:** Authorization, Content-Type, Accept gibi.
- **Response İnceleme:** JSON çıktısı okunur, status code (200, 400, 500) kontrol edilir.

2.3 Postman Test Script Yazımı (How to write test scripts in Postman tool?)

- **Amaç:** API'nin döndürdüğü cevapları otomatik test etmek.
- **JavaScript Tabanlı Örnek:**

```
pm.test("Status code is 200", function () {  
    pm.response.to.have.status(200);  
});
```

- **Assertion Örnekleri:**
 - Response süresi 500 ms'den kısa mı?
 - Body içinde belirli bir alan var mı?
 - Login sonrası token döndü mü?
- **Response Body Kontrolü:** JSON parse edilip belirli alanlar assert edilir.

3. Gerçek Hayat Senaryoları

Senaryo 1: Yemeksepeti Üzerinden Sipariş Akışı

1. Kullanıcı mobil uygulamadan yemeği seçer ve "Sipariş Ver" butonuna tıklar.
2. Uygulama, **REST API** üzerinden backend'e **POST /orders** isteği gönderir.
3. Backend bu bilgiyi veritabanına kaydeder ve restorana iletir.
4. Restoran siparişi onayladığında API'den **status: confirmed** yanıtı döner.
5. Kullanıcı ekranında "Siparişiniz alındı" mesajını görür.

Senaryo 2: Bankacılıkta Para Transferi (EFT/Havale)

1. Kullanıcı mobil bankacılık uygulamasından EFT yapmak ister.
2. Uygulama, bankanın **SOAP API** servisine **TransferRequest** gönderir.
3. XML formatında bilgiler (tutar, alıcı IBAN, açıklama) iletilir.
4. Banka sistemi bu isteği merkez bankasıyla paylaşır.
5. İşlem başarılı olursa kullanıcıya "Transfer başarılı" mesajı gelir.

Senaryo 3: Hava Durumu Uygulaması

1. Kullanıcı sabah telefonundaki hava durumu uygulamasını açar.
2. Uygulama, Meteoroloji'nin **REST API**'sine **GET /weather?city=Istanbul** isteği gönderir.
3. JSON formatında yanıt döner: **{ "temp": 28, "condition": "Sunny" }**.
4. Uygulama bu bilgiyi görsel olarak kullanıcıya sunar.

Senaryo 4: Postman ile E-ticaret API Testi

1. Postman açılır, **GET /products** isteği hazırlanır.

2. **Authorization** kısmına API token eklenir.
3. **Send** butonuna basıldığında **200 OK** yanıtı gelir.
4. Response body'de JSON formatında ürün listesi döner.
5. Test script eklenir:

```
pm.test("Ürün listesi boş değil", function () {  
    var jsonData = pm.response.json();  
    pm.expect(jsonData.length).to.be.above(0);  
});
```

Bugün Öğrendiklerimin Özeti

Bugün teorik olarak web service'in ne olduğunu, API'nin yazılım dünyasındaki önemini, SOAP ve REST arasındaki farkları ve HTTP metodlarının mantığını öğrendim.

Pratikte ise Postman üzerinde GET/POST istekleri denedim, header eklemeyi ve basit test scriptleri yazmayı tekrar ettim.

Şunu fark ettim: API'ler sadece sistemler arası veri alışverişini sağlamıyor, aynı zamanda projelerin dış dünyaya açılan kapısı oluyor. Yemeksepeti örneğinde REST'in pratikliğini, bankacılık alanında ise SOAP'un güvenlik ve transaction odaklı gücünü kavradım. Hava durumu API'si gibi günlük hayattaki örnekler ise REST kullanımının ne kadar yaygın olduğunu net bir şekilde gösterdi.

Postman senaryoları üzerinde çalışırken API test sürecinin otomasyonunun önemini daha iyi anladım. Basit bir test scripti bile, uzun vadede hataları önlemede ciddi bir kolaylık sağlıyor.

Artık biliyorum ki staj veya projelerimde, sadece backend geliştirmek değil; API testlerini Postman üzerinden doğru şekilde yapmak, gerektiğinde CI/CD pipeline içine bu testleri entegre etmek yazılım kalitesini belirleyen kritik adımlardan biri olacak.