

F.Ü. Mühendislik Fakültesi
Bilgisayar Mühendisliği Bölümü
Bilgisayar Sistemleri Laboratuvarı

DENEY NO: 5

WEB API KULLANIMI

Deneyin Amacı:

Farklı hizmet sağlayıcıları tarafından sunulan hizmetleri projeye dahil etmek için API'ler günümüzde sıklıkla kullanılmaktadır. Bu deneyde, örnek API kullanımı incelenerek, Java ile API servislerine sorgu gönderme işlemi gerçekleştirilecektir.

API Nedir?

API (Application Programming Interface), uygulama programlama arayüzü anlamına gelmektedir. Uygulama programlama arayüzleri, farklı programların birbirleri ile iletişimini sağlayan protokollerdir. Günümüzde sosyal ağlardan alışveriş sitelerine kadar birçok site API'lerini yazarak bunları uygulama geliştiricilerin hizmetine açmıştır. Örneğin bir e-ticaret uygulamasında, geliştirdiğiniz uygulamayı e-ticaret platformları ile entegre etmek isterseniz o platform tarafından geliştirilen API'leri kullanmanız gerekir. API'leri kullanarak, e-ticaret platformuna giriş yapmadan geliştirdiğiniz program üzerinden ürün açma, ürün silme, stok güncelleme, ürünleri listeleme gibi izin verilen işlemleri gerçekleştirebilirsiniz. Veya uygulamanızda hava durumu bilgisi gerekiyorsa, bu bilgiye hava durumu servis sağlayıcısı tarafından sunulan API'yi kullanarak ulaşabilirsiniz. Sosyal ağlar tarafından sağlanan API'leri kullanarak izin verilen gönderiler çekilebilir, mesajlaşma işlemleri gerçekleştirilebilir ve izin verilen kişisel bilgilere (cinsiyet, yaş, arkadaş listesi gibi) geliştirdiğiniz uygulama aracılığıyla erişilebilir. Bazı zamanlar kritik ve veri güvenliğinin önemli olduğu işleri yapmakla görevli olan API'lerin güvenliğinin sağlanması önemli bir konudur. Bunun için API kullanımından önce geliştiricilere özel olarak üretilen bazı kodlar verilerek kimliklerinin doğrulanması sağlanmalıdır.

REST Nedir?

REST (REpresentational State Transfer), temsili durum transferi anlamına gelmektedir. API'lerin belirli standartlara oturması için geliştirilmiş olan bir terimdir. REST API'lerin belirli özellikleri sağlaması gerekir. Bu özellikler Stateless, Uniform Interface, Cacheable, Client-Server, Layered System, Code on Demand olup API geliştirilmesi aşamasında dikkat edilmesi gereken hususlardır. Tüm bu özellikleri sağlayan API'ler RESTFUL API olarak isimlendirilir. REST API'lerde genellikle JSON (JavaScript Object Notation) formatı kullanılarak bilgi alışverişi yapılır.

JSON Nedir?

JSON (JavaScript Object Notation), objelerin veya dizilerin serileştirilmesi ve bu sayede iletiminin kolaylaştırılmasını sağlar. Bir veri transfer edilecekse öncelikle JSON formatına dönüştürülür. JSON özel olarak yapılandırılmış string ifadelerdir. JSON aracılığıyla ulaşması gereken yere ulaşan veri gerektiğinde tekrar nesne veya dizi haline dönüştürülebilir. Örnek JSON formatı aşağıda verilmiştir. JSON'da objeler { }, diziler [] ile gösterilir. Dizi içinde obje, obje içinde de dizi kullanımına izin verilir.

```
{
  "no":1,
  "isim": "Ahmet",
  "kurum": {
    "kurumAdı": "Fırat Üniversitesi",
    "birim": "Mühendislik Fakültesi",
    "bölüm": "Bilgisayar Mühendisliği",
  },
  "telefon": [
    { "tur": "ev", "no": "0424 000 00 00" },
    { "tur": "iş", "no": "0424 237 00 00" }
  ],
  "sectigiDersler": [],
  "danışman": null
}
```

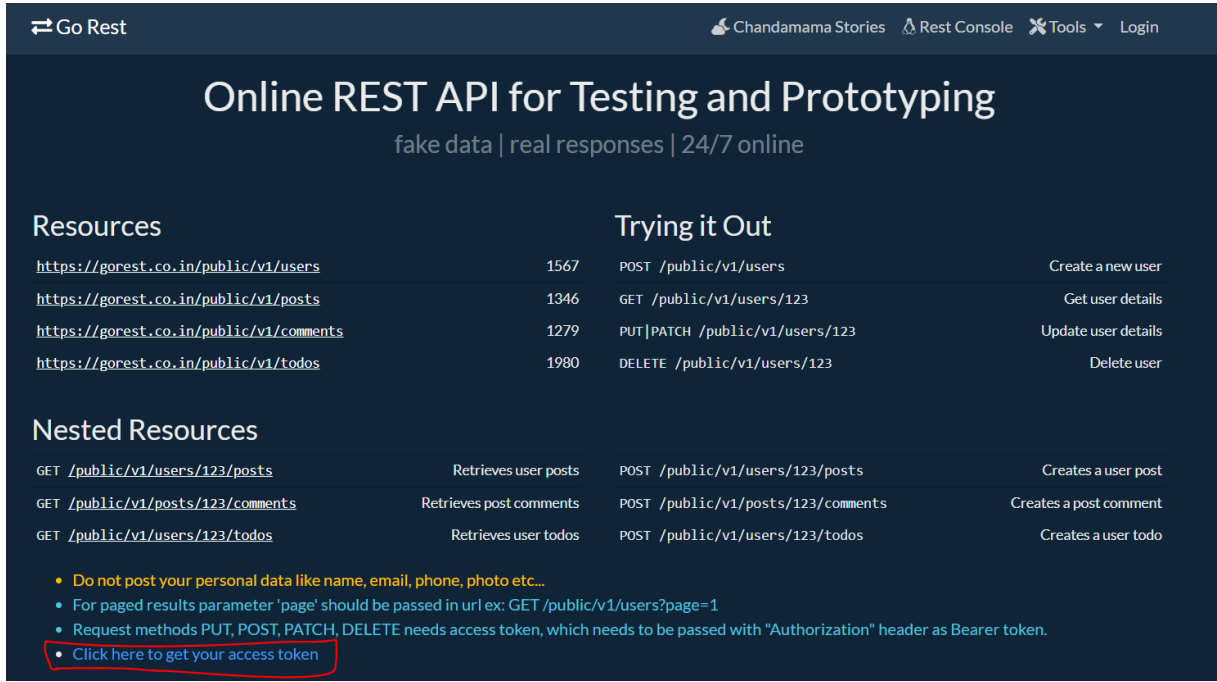
Örnek API Servisleri

Birçok büyük kuruluş API servisi vermektedir. Bunların yanı sıra geliştiricilerin sınırlı denemelerle veya örnek olarak kullanabileceği API servisleri de bulunmaktadır. Collectapi (<https://collectapi.com/tr/>) sayfasında farklı servis sağlayıcılarına ait API'ler bulunmaktadır. Bunlardan bazıları Tablo 1 ile verilmiştir. Kullanım detaylarına ve hizmet kapsamına ilgili URL'lerden ulaşılabilir. Collectapi sayfası altındaki API'leri ücretsiz olarak denemek için ücretsiz hesap açarak API Key almanız gerekmektedir.

Tablo 1. Collectapi tarafından sağlanan sınırlı ücretsiz kullanıma sahip API listesi

API Adı	URL	Ücretsiz Kullanım
Nöbetçi Eczane API	https://collectapi.com/tr/api/health/nobetci-eczane-api	100 istek / ay
Akaryakıt Fiyatları API	https://collectapi.com/tr/api/gasPrice/akaryakit-fiyatlari-api	100 istek / ay
Altın, Döviz ve Borsa API	https://collectapi.com/tr/api/economy/altin-doviz-ve-borsa-api	100 istek / ay
Haberler API	https://collectapi.com/tr/api/news/haberler-api	1000 istek / ay
Hava Durumu API	https://collectapi.com/tr/api/weather/hava-durumu-api	2500 istek / ay
Faiz Oranları API	https://collectapi.com/tr/api/credit/faiz-oranlari-api	30 istek / ay
Besin Bilgisi Analizi API	https://collectapi.com/tr/api/food/besin-bilgisi-analizi-api	10 istek / ay
Kitaplar API	https://collectapi.com/tr/api/book/kitaplar-api	100 istek / ay
Sözlük API	https://collectapi.com/tr/api/dictionary/sozluk-api	10 istek / ay

Örnek olarak kullanılabilir bir başka API servisi de <https://gorest.co.in/> adresinde sağlanmaktadır. Bir API'yi kullanmak için ilgili API dokümantasyonunun incelenmesi gerekir. Adrese giderek API kullanımı ile ilgili gerekli bilgileri edinebilirsiniz. Genel olarak REST API'ler güvenliği sağlamak için kullanıcı doğrulaması (authentication) gerektirir. Bu işlem API key veya tokenlar ile gerçekleştirilir. Gorest sayfasındaki sahte API'yi kullanmak için de Şekil 1'de gösterilen linkten token almanız gerekir.

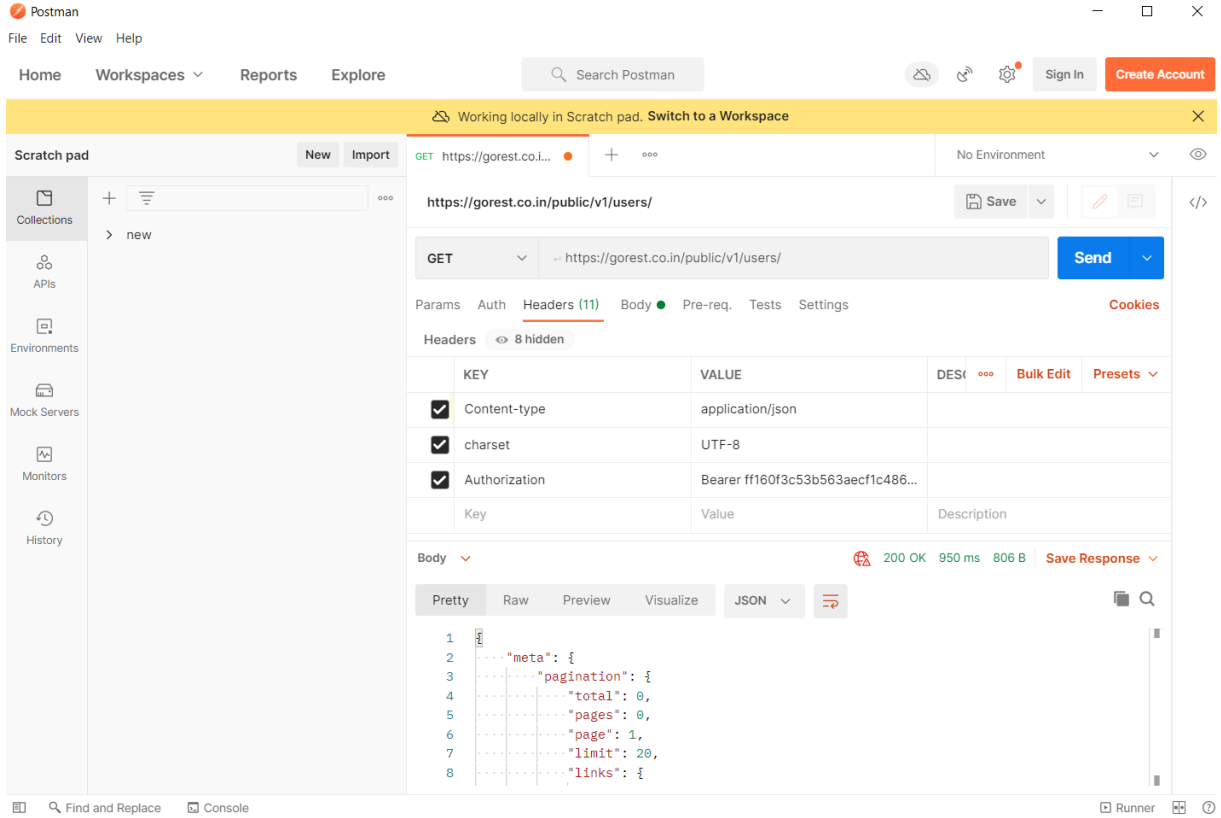


Şekil 1. <https://gorest.co.in/> adresinden token alma

Yukarda bahsedilen API örneklerini internet aramaları ile çoğaltmak mümkündür.

API Test Araçları ve POSTMAN

Özellikle büyük çaplı projelerde hatanın kaynaklanma sebebinin bulunması karmaşık bir hale gelir. API kullanılması ile programdan veya API bağlantısından kaynaklanan problemler ortaya çıkabilmektedir. Bu yüzden öncelikle API'lere sorguların düzgün bir şekilde yapılması ve sonucun alınması gerekir. API araçları sayesinde sorguların API dokümantasyonunda yer alan bilgilere göre düzgün bir şekilde gerçekleştirilip gerçekleştirilemediği test edilir. POSTMAN bu amaçla kullanılan bir API aracıdır. POSTMAN programına ait arayüz şekil 2'de verilmiştir. Bu arayüz aracılığıyla API adresine (URI) ilgili header (üst bilgi), params (parametreler) bilgileri ile sorgular gerçekleştirilebilir. POSTMAN programı ile POST, GET, PUT gibi http metotlarını kullanarak sorgular gerçekleştirebilirsiniz. Sorgu sonucunda sunucudan dönen yanıt alt kısımda görülecektir. Bu kısımda işlem başarılı ile işlem sonucu üretilmesi gereken cevap, başarısız ise de hatanın kodunu içeren mesaj görülmektedir. Sunucudan dönen JSON ifade Raw sekmesi altında olduğu gibi, Pretty sekmesi altında ise hizalanmış bir şekilde görüntülenebilir. Haberleşmede kullanılan orijinal JSON ifade Raw sekmesi altın görüldüğü gibi olmakla birlikte, okunurluğunu kolaylaştırmak için formatlanmış bir şekilde gösterilmesi tercih edilir.



Şekil 2. POSTMAN Arayüzü

Sık Kullanılan API Request Metotları

1. **GET:** Genellikle sunucudan veri almak için kullanılan http metodudur. GET metodu ile sorgu ile gönderilmesi gereken parametreler URL içinde gönderilebilir. Parametrelerin URL gibi görülebilir bir yolla gitmesi gizlilik gerektiren iletişim ihtiyacı için uygun değildir.
2. **POST:** Bu metot ile parametreler hem URL ile hem de body bilgisinde gönderilebilir. Body bilgisinde gönderilen parametreler gizlilik açısından daha etkili olmaktadır.
3. **PUT:** Genellikle veri güncellemek için kullanılan protokoldür.
4. **DELETE:** Genellikle veri silmek için kullanılan protokoldür.

HTTP Durum kodları

API'ler http protokolü üzerinden çalıştıklarından dolayı gönderdikleri hata kodları da http durum kodları ile aynı olmaktadır. http durum kodlarının sınıflandırılması aşağıda verilmiştir. POSTMAN gibi bir araç ile sorgu işlemi esnasında 4xx hata kodlarından biri meydana geliyorsa, bu sorguda (parametre, uri vs.) hata olduğu anlamına gelmektedir.

HTTP Durum Kodu Sınıfları

http durum kodları baştaki ifadeye göre sınıflandırılır. Örneğin durum kodu 1 ile başlayan kodlar aynı türevden benzer hatalar anlamına gelmektedir. http Durum kodu sınıfları aşağıda verilmiştir.

1xx: Gönderilen isteğin sunucuya başarılı bir şekilde ulaştığını ve işlemin başladığını bildiren durum kodlarını içerir (Bilgi içeren mesajlar).

2xx: Gönderilen isteğin sunucuya başarılı bir şekilde ulaştığını ve işlemin başarılı bir şekilde sonlandığını bildiren durum kodlarını içerir (Başarılı işlem mesajları).

3xx: Sunucunun başka bir adrese taşındığı durumlarda verilir (Yönlendirme mesajları).

4xx: Gönderilen sorguda eksik veya yanlış ifadelerin olması durumunda sunucudan dönen durum kodudur (İstemci hatası mesajları). Bu başlıkta gelen hata kodları genellikle sorgunun yanlış veya eksik olmasından kaynaklanır.

5xx: Sunucu tarafında bir hata olduğunu ifade eden durum mesajlarıdır (Sunucu hatası mesajları). Bu durumda servis sağlayıcısının durumu düzeltmesi beklenir.

Sık Karşılaşılan HTTP Durum Kodları:

200: İstemci ve sunucu arasındaki iletişimin başarılı olduğunu gösterir.

201: Gönderilen sorgu sonucunda sunucu tarafında bir verinin oluşturulduğu anlamına gelir.

400: İstek URI'sinin sözdiziminde bir yanlışlık olduğunu ifade eder.

401: Sunucu-istemci bağlantısı için kimlik doğrulaması gerekliliğini ifade eder.

403: Kimlik doğrulaması başarılı olup, kimliği doğrulanan kişinin ilgili URI'ye erişim hakkının olmamasını ifade eder.

404: URI adresinin sunucu tarafından bulunamadığını ifade eder.

405: Gönderilen request metodunun sunucu tarafından desteklenmediğini ifade eder.

POSTMAN ile API Sorgusu Yazma

Collectapi sitesinde yer alan (tablo 1’de verilmiştir) Besin Bilgisi Analizi API’sini kullanarak uygulamamıza entegre etmeyi deneyelim. Besin Bilgisi Analizi API açıklamasında; “Besinlerin kalorilerine ulaşabilir, girilen ayın hangi meyve ve sebzelerin mevsimi olduğunu öğrenebilirsiniz” bilgisi yer almaktadır. Bu API’de GET /calories ve GET /whenFoods isminde iki temel *end point* bulunmaktadır. Giriş yapıp API key aldıktan sonra API dokümantasyonunda yer alan bilgiler ile POSTMAN aracılığıyla sorgu gerçekleştirebiliriz. Şekil 3’te GET /calories metoduna sorgu göndermek için gereken parametreler yer almaktadır.

END POINTS

▼ GET /calories

Yiyeceklerin kalorisini getiren servis. Girilen ürün ve bu ürünle yapılan diğer ürünlerin kalorilerini getirir.

PARAMETRELER

Alan	Açıklama	Tip	Başlık	Zorunlu
query	Kalorisini öğrenmek istediğiniz yiyeceği yazmanız gerekiyor.	text		✓

ÖRNEK

Shell Go Node Javascript Java Python Ruby Csharp Swift Ocaml Php

```
curl --request GET \
  --url 'https://api.collectapi.com/food/calories?query=elma' \
  --header 'authorization: apikey 6yj4exYlI5CavIzYu512zp:5KlB26EojjaIAFVP88TmBR' \
  --header 'content-type: application/json'
```

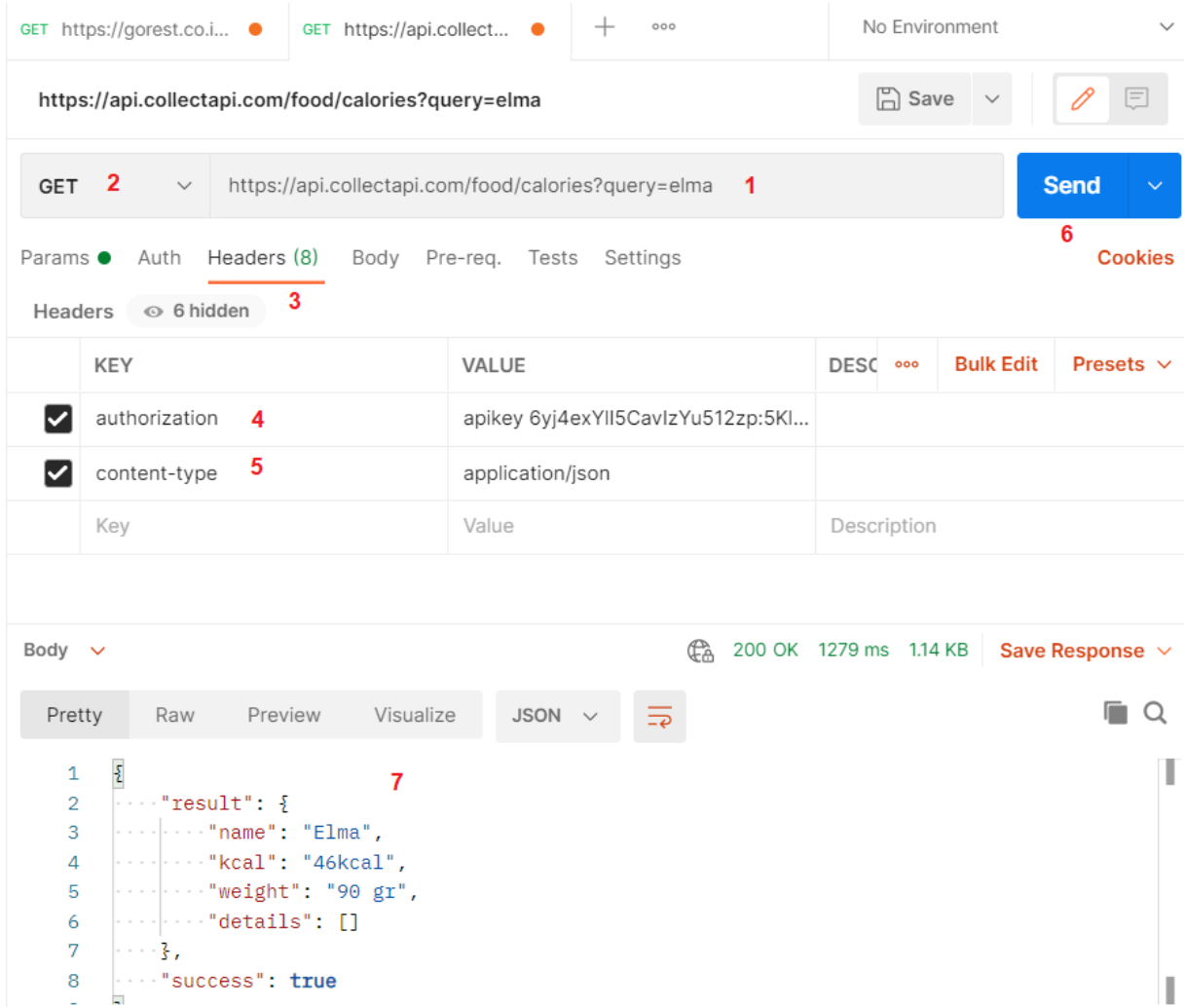
Kopyala

Şekil 3. Besin değerleri API dokümantasyonu.

POSTMAN programını açtıktan sonra file new diyerek yeni bir sorgu ekranı açılır:

- 1- Bu sorgu ekranında Enter Request Url yazan kısma API’nin erişim adresi ve URL’de göndermek istenilen parametreler yazılır.
(<https://api.collectapi.com/food/calories?query=elma>).
- 2- Metodun dokümantasyonda belirtildiği şekilde (GET) olduğu kontrol edilir.
- 3- Header sekmesi açılır.
- 4- Header sekmesi altında key kısmına “authorization”, value kısmına “apikey size_verilen_api_key” yazılır.
- 5- Yine header altında yeni satıra key yerine “content-type”, value kısmına da “application/json” yazılır.
- 6- Send butonuna tıklanır
- 7- Sonuç sekmesinde cevap gösterilir.

Verilen adımlar sırası ile şekil 4’te gösterilmiştir.



Şekil 4. POSTMAN ile API sorgusu gönderme

Görüldüğü üzere yapılan sorgu 200 durum kodu ile birlikte dönmüştür. Bu durum kodu sorgu başarılı bir şekilde yürütüldü ve yanıt döndü anlamına gelmektedir. Dolayısı ile API sorgusunun bu parametreler ile doğru bir şekilde çalıştığı doğrulanmış olur. Bu aşamadan sonra programlama dili aracılığıyla aynı parametreler ile API sorgusu gerçekleştirilebilir.

JAVA ile Örnek API Bağlantısının Gerçekleştirilmesi

API'lerin amacı yazılım geliştiricileri için hizmet sunmaktır. Dolayısı ile API'nın bir programlama dili ile kullanılması gerekir. POSTMAN aracı ile doğrulanan API sorgusunun programlama dili ile gönderilmesi için ilgili programlama dillerinde API'lerin nasıl kullanıldıklarına bakılmalıdır. Bu deneyde Java programlama dili kullanılarak sorgular gerçekleştirilecektir.

Java ile http istekte bulunmak için client ve request nesneleri oluşturmak gerekmektedir. Request nesnesi uri ve parametre bilgilerini içermektedir. Örnek bir sorgunun yazılması aşağıda verilmiştir. Dönen cevap response nesnesi olup, body() fonksiyonu ile JSON formatında dönen bilgi yazdırılabilir. Kodda yer alan ... kısmına site aracılığıyla size verilen API key yazılmalıdır. Burdaki query=armut kısmı dinamikleştirilerek bu bilgiyi kullanıcının girmesi sağlanabilir.

```
// client (istemci) oluşturma

var client = HttpClient.newHttpClient();

// request (sorgu) oluşturma

var request = HttpRequest.newBuilder()

    .uri(URI.create("https://api.collectapi.com/food/calories?query=armut"))

    .header("authorization", "apikey ...")

    .header("content-type", "application/json")

    .build();

var response = client.send(request, BodyHandlers.ofString()); //BodyHandlers.discarding()

System.out.println(response.body());
```

Java kodunun çalıştırılması ile konsolda yazılan ifade aşağıdaki gibi olmalıdır.

```
{ "result": { "name": "Armut", "kcal": "114kcal", "weight": "200 gr", "details": [] }, "success": true }
```

Dönen cevap JSON formatında olup, JSONObject nesnesi ile ayrıştırılabilir. Dönen JSON cevabı yazılımın ihtiyacına göre Java nesnelere dönüştürülebilir. JSON stringinden direkt olarak veriyi okumak için JSON formatını bilmemiz gerekmektedir. Bu örnekte result objesi altında name, kcal, weight gibi bilgiler yer aldığından, örneğin name bilgisini çekmek için aşağıdaki kod eklenmelidir. Yine daha öncesinden response nesnesi ile dönen durum kodunun incelenerek işlemin başarısı kontrol edilebilir.

```
JSONObject obj = new JSONObject(response.body()); //response.statusCode()

System.out.println(obj.getJSONObject("result").getString("name"));
```

Yukardaki işlem Java net kütüphanesi kullanılarak gerçekleştirilmektedir. İstenirse bu işlem için 3. Part uygulamalar da kullanılabilir. Collectapi sitesinde verilen kodu doğrudan çalıştırmak için ekstra kütüphaneler dahil edilmesi gerekmektedir. Unirest-java isimli kütüphaneyi kullanarak verilerin çekilmesi aşağıda gösterilmiştir. Burada dönen JSON verisini yazdırmak için `getBody()` fonksiyonu kullanılmalıdır.

```
kong.unirest.HttpResponse<String> response = Unirest.get("...?query=elma")

    .header("content-type", "application/json")

    .header("authorization", "apikey ...")

    .asString();

System.out.println(response.getBody());
```

Kütüphanenin Maven kullanarak projeye dahil etmek için pom.xml dosyası içerisinde `<dependencies></dependencies>` etiketleri arasına aşağıdaki satırlar eklenmelidir.

```
<dependency>

    <groupId>com.konghq</groupId>

    <artifactId>unirest-java</artifactId>

    <version>3.12.0</version>

</dependency>
```

Deneyin Gerçekleştirilme Aşamaları

Deney için herhangi bir API servisi kullanarak istediğiniz programlama dili ile bu servise sorgu yapmanız istenmektedir. Deneyin gerçekleştirilme aşamaları aşağıda verilmiştir:

- 1- Problemin (hava durumu, nöbetçi eczaneler gibi) ve kullanılacak API servisinin belirlenmesi, ilgili servisten API key gibi gerekli erişim izinlerinin alınması.
- 2- API dokümantasyonu incelenerek end pointlere nasıl sorgu gönderileceği, hangi parametrelerin gönderilmesi gerektiğinin belirlenmesi.
- 3- POSTMAN programı ile belirlenen parametreler ile ilgili uri'lere sorgu gerçekleştirilerek, API'nin düzgün bir şekilde çalıştığının gösterilmesi.

- 4- Seçilen programlama dilinde http/api isteklerinin nasıl gerçekleştirileceği araştırılarak ilgili kodların yazılması.
- 5- Gelen cevaptaki http durum kodunun okunarak hatalı bir durum olup olmadığının kontrol edilmesi. Hatalı bir durum varsa hatanın yazdırılması.
- 6- Hatalı bir durum olmadığı durumda gelen JSON cevabının ayrıştırılarak ekranda/konsolda ilgili yerlerde gösterilmesi. İlgili butonlara tıklandığı/konsolda ilgili komut verildiği zaman API servise sorguda bulunulması.
- 7- Kullanıcıların hatalı parametreler girmesine veya http durum kodu hatalarına karşı gerekli önlemlerin alınarak programın çökmeden çalışmasının sağlanması.

Hazırlık Soruları

- 1- API kavramı nedir? Ne için ihtiyaç duyulur? Kullanım alanlarını araştırınız.
- 2- End point, uri, istemci, sunucu kavramlarını açıklayınız.
- 3- Arama motorları, sosyal ağlar ve e-ticaret siteleri gibi kategorilerde API hizmeti sunan firmalar hangileridir?
- 4- Bu firmalardan bir tanesinin API dokümantasyonunu inceleyerek ne tür sorgulara izin verdiğini öğreniniz.
- 5- JSON ve XML nedir? Birbirlerine göre avantajları dezavantajları nelerdir?
- 6- JSON formatında diziler ve nesneler nasıl ifade edilir. Araştırınız.
- 7- API Test Aracı nedir? En çok kullanılan API Test Araçları hangileridir?
- 8- Http durum kodları kategorileri nelerdir? Bütün durum kodlarının ne anlama geldiklerini araştırınız.
- 9- JSON string, JSON obje nedir? Farkları nelerdir? Kullandığınız programlama dilinde bunların dönüşümünü sağlayan hazır sınıflar var mıdır? Araştırınız.

Kaynaklar:

1. Zhou, W., Li, L., Luo, M., & Chou, W. (2014, May). REST API design patterns for SDN northbound API. In 2014 28th international conference on advanced information networking and applications workshops (pp. 358-365). IEEE.
2. Nurseitov, N., Paulson, M., Reynolds, R., & Izurieta, C. (2009). Comparison of JSON and XML data interchange formats: a case study. Caine, 9, 157-162.

3. POSTMAN, Build APIs together, <https://www.postman.com/>, erişim tarihi: 30.09.2022
4. Amazon, API nedir?, <https://aws.amazon.com/tr/what-is/api/>, erişim tarihi: 30.09.2022
5. Mulloy, B. (2013). Web API design. <https://hashingit.com/elements/research-resources/2012-web-api-design.pdf>
6. Berners-Lee, T., Fielding, R., & Frystyk, H. (1996). Hypertext transfer protocol--HTTP/1.0 (No. rfc1945).
7. API Marketplace, <https://collectapi.com/tr>, erişim tarihi: 30.09.2022
8. GraphQL and REST API for Testing and Prototyping, <https://gorest.co.in>, erişim tarihi: 30.09.2022