



PYTHON H1 YETENEK SINAVI

Hedefler

Bölüm 1: Northwind API - Mapquest API – Epoch Time için dökümantasyon

Bölüm 2: Kodlama anı

Bölüm 3: Çalışma anı

Senaryo

Northwind DB, customers&orders tabloları için online servis olan <https://northwind.netcore.io/> sitesine bağlanarak, kullanıcı isteğine göre listeleme operasyonunu yapılacaktır. Tablolardaki her kayıt için detaylı raporlamayı görebilmek adına Arama operasyonu yapılacaktır. Tablo'da bazı kayıtların bazı hücrelerinden gelen data, output'u manipüle etmektedir. Karakter uzunluğu çok fazla olmasından dolayı UI'da bize sorunlar yaşatmaktadır. Bu sorun ile başa çıkacak kodlamayı unutmayacağız. Geriye dönen zaman bilgisi epoch zamanı olduğu için en doğru modülü ekleyip günümüz zamanına çevrimini başaracağız. Orders tablosundan arama yapılırken detaylı raporlama anında, kullanıcının girdiği çıkış noktasından siparişin hedef varış noktasına kadar olan rotayı çizeceğiz. Json datalarını text formatında görüntülediğimizde pekte insani çıktılar görünmüyor. Kolay okunabilir çıktılar olmuyor. Kolay okunabilmesi adına json beautifier konsepti öğrenilecek.

Gereksinimler

- MapQuest yazılım geliştiriciler için erişime açılan API key
- <https://jsonformatter.curiousconcept.com> json beautifier
- python-h1.py öğrenci script'i.

Talimatlar

Bölüm 1: Northwind API – MapRequest API – Epoch Time İçin Dökümantasyon

Bu bölümde Northwind DB, customers&orders tabloları için online servis olan <https://northwind.netcore.io/> sitesine bağlanacağız.

Adım 1: API request", "URL parse", "epoch time conversion" için gerekli kütüphaneleri araştırın.

- a. Login olun → <https://developer.mapquest.com/>
- b. Kişisel key'inizi oluşturun. Key ne zaman expire olacak? Sınırsız kullanım süresi - Never
- c. Müşterileri listeyeceğiniz URL'yi bulun ve aşağıya yazın.
<https://northwind.netcore.io/query/customers.json>
- d. Siparişleri listeyeceğiniz URL'yi bulun ve aşağıya yazın.
<https://northwind.netcore.io/query/orders.json>

Adım2: MapQuest API'yi detaylı inceleyelim.

- e. Login olun → <https://developer.mapquest.com/>

Kişisel key'inizi oluşturun

- f. Chrome tarayıcısına aşağıdaki URL giriniz, **kisisel_key** kısmındaki yere kendi key'inizi ekleyin:

https://www.mapquestapi.com/geocoding/v1/address?key=kisisel_key&location=istanbul

- g. Output'u inceleyin

```
{
  "info": {
    "statusCode": 0,
    "copyright": {
      "text": "\u00A9 2021 MapQuest, Inc.",
      "imageUrl": "http://api.mqcdn.com/res/mqlogo.gif",
      "imageAltText": "\u00A9 2021 MapQuest, Inc."
    },
    "messages": []
  },
  "options": {
    "maxResults": -1,
    "thumbMaps": true,
    "ignoreLatLngInput": false
  },
  "results": [
    {
      "providedLocation": {
        "location": "istanbul"
      },
      "locations": [
        {
          "street": "",
          "adminArea6": "",
          "adminArea6Type": "Neighborhood",
          "adminArea5": "Istanbul",
          "adminArea5Type": "City",
          "adminArea4": "Fatih",
          "adminArea4Type": "County",
          "adminArea3": "\u00130STANBUL",
          "adminArea3Type": "State",
          "adminArea1": "TR",
          "adminArea1Type": "Country",
          "postalCode": "",
          "geocodeQualityCode": "A5XAX",
          "geocodeQuality": "CITY",
          "dragPoint": false,
          "sideOfStreet": "N",
          "linkId": "283510263",
          "unknownInput": "",
          "type": "s",
          "latLng": {
            "lat": 41.017058,
            "lng": 28.985568
          },
          "displayLatLng": {
            "lat": 41.017058,
            "lng": 28.985568
          },
          "mapUrl": "http://www.mapquestapi.com/staticmap/v5/map?key=eod44I9Nm3s0qCIgAk31NZYQ9TloitnL&type=map&size=225,160&locations=41.017058,28.985568|marker-sm-50318A-1&scalebar=true&zoom=12&rand=1278520010",
          "street": "",
          "adminArea6": "",
          "adminArea6Type": "Neighborhood",
          "adminArea5": "",
          "adminArea5Type": "City",
          "adminArea4": "",
          "adminArea4Type": "County",
          "adminArea3": "\u00130STANBUL",
          "adminArea3Type": "State",
          "adminArea1": "TR",
          "adminArea1Type": "Country",
          "postalCode": "",
          "geocodeQualityCode": "A3XAX",
          "geocodeQuality": "STATE",
          "dragPoint": false,
          "sideOfStreet": "N",
          "linkId": "306019918",
          "unknownInput": "",
          "type": "s",
          "latLng": {
            "lat": 41.076602,
            "lng": 29.052495
          },
          "displayLatLng": {
            "lat": 41.076602,
            "lng": 29.052495
          },
          "mapUrl": "http://www.mapquestapi.com/staticmap/v5/map?key=eod44I9Nm3s0qCIgAk31NZYQ9TloitnL&type=map&size=225,160&locations=41.076602,29.052495|marker-sm-50318A-2&scalebar=true&zoom=5&rand=-1029967550"
        }
      ]
    }
  ]
}
```

- h. <https://jsonformatter.curiousconcept.com/> sitesi ile dönen json'u beautifier konsepti ile daha okunur hale getirin.

```
{
  "info": {
    "statusCode": 0,
    "copyright": {
      "text": "\u00A9 2021 MapQuest, Inc.",
      "imageUrl": "http://api.mqcdn.com/res/mqlogo.gif",
      "imageAltText": "\u00A9 2021 MapQuest, Inc."
    },
    "messages": [

    ]
  },
  "options": {
    "maxResults": -1,
    "thumbMaps": true,
```

```
        "ignoreLatLngInput":false
    },
    "results":[
        {
            "providedLocation":{"
                "location":"istanbul"
            },
            "locations":[
                {
                    "street":"","
                    "adminArea6":"","
                    "adminArea6Type":"Neighborhood",
                    "adminArea5":"Istanbul",
                    "adminArea5Type":"City",
                    "adminArea4":"Fatih",
                    "adminArea4Type":"County",
                    "adminArea3":"\u0130STANBUL",
                    "adminArea3Type":"State",
                    "adminArea1":"TR",
                    "adminArea1Type":"Count
```

Adım3: Northwind DB, customers&orders tablolarından gelen datalardaki epoch zamanını daha okunaklı hale getirelim

Kod anında, **ctime** fonksiyonunu kullanacağız. O yüzden Python **time** kütüphanesini kullanarak epoch zamanı daha okunaklı hale dönüştürmek adına çevrimini yapacaksınız.

a. "epoch" nedir?

Tarihlerin 01.01.1970 beri geçen saniye sayısı ile ifade edildiği veri tipine verilen isimdir.

b. Kendi doğum tarihinizin epoch zamanındaki karşılığını sitesi ile keşf edin. Epoch datasındaki digit'leri anlamak için keyifli bir serüven olacak.

316016811 → 6 Ocak 1980

c. Şimdi bir de, burada keşfettiğimiz epoch zamanını python shell'de deneyelim.

PYTHON H1 YETENEK SINAVI

Powershell kullanarak denemek istiyorum. (win + R → powershell)

PS C:\> python

Python 3.9.2 (tags/v3.9.2:1a79785, Feb 19 2021, 13:44:55) [MSC v.1928 64 bit (AMD64)]
on win32

Type "help", "copyright", "credits" or "license" for more information.

>>> import time

>>> print(time.ctime(316016811))

Sun Jan 6 17:26:51 1980

Bölüm 2: Kodlama anı

Not: `python-h1.py` script'i kurumdan talep edin.

Tavsiye: Belki doğru rotada olacağınız ihtimalinden dolayı orjinal dosyayı Save As diyerek saklı kalmasını sağlayın. Baştan başlamak isteyebilirsiniz.

Adım1: API request", "URL parse", "epoch time conversion" için kütüphaneleri import edin.

```
# API request",  
# "URL parse",  
# "epoch time conversion"  
import #<Gerekli Kütüphaneler İle Değiştir>
```

Adım 2: Customer API için gerekli URL girin.

```
urlCustomers = "<Gerekli URL İle Değiştir>"
```

Adım 3: Customer API için GET Request

```
rCustomers = #<Request>
```

Adım 4: Status_Code kontrolü herhangi bir kod değişikliği olmayacak.

```
if not rCustomers.status_code == 200:  
    raise Exception("API Bağlantı Sorunu. Status code: {}. Text: {}".format(  
        rCustomers.status_code, rCustomers.text))
```

Adım 5: Response doğrulayalım. Sadece referans amaçlı.

```
#print(rCustomers.text)
```

Adım 6: Response içeriği json formatına encode edelim

```
jsonCustomers = #<Gerekli Json Encode Kodu İle Değiştir>
```

Adım 7: Order API için gerekli URL girin.

```
urlOrders = "<Gerekli URL İle Değiştir>"
```

Adım 8: Order API için GET Request

```
rOrders = #<Request>
```

Adım 9: Status_Code kontrolü herhangi bir kod değişikliği olmayacak.

```
if not rOrders.status_code == 200:  
    raise Exception("API Bağlantı Sorunu. Status code: {}. Text: {}".format(  
        rOrders.status_code, rOrders.text))
```

Adım 10: Response doğrulayalım. Sadece referans amaçlı.

```
#print(rOrders.text)
```

Adım 11: Response içeriği json formatına encode edelim

```
jsonOrders = #<Gerekli Json Encode Kodu İle Değiştir>
```

Adım 12: MapQuest API için URL girelim

```
mainMapApiUrl="<Gerekli URL İle Değiştir>"
```

Adım 13: MapQuest API için kişisel Credential'larımızı girelim

```
mapApiKey="Kisisel_Key"
```

Adım 14: UI manipulasyonu. Output anında 15 karakterden uzun metinler için manipulasyon yapılacaktır

```
def metinKontrol(metin):  
    if len(metin)>15:  
        return f"<Gerekli Kod İle Değiştir>"  
    else:  
        return f"<Gerekli Kod İle Değiştir>"
```

Adım 15: Müşterileri listeleyecek

```
def musteriliste():  
    print("Müşteri Listesi")  
    print("+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+  
+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+  
|ID          |CompanyName          |ContactName          |Address  
|Country          |City          |  
+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+  
+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+  
  
    for i in jsonCustomers["<Gerekli Key İle Değiştir>"]:  
        print(f"Gerekli Output İçin Formatla")  
  
    print("+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+  
+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+)
```

Adım 16: Müşterileri Id'ye göre arama yapılacaktır

```
def musteriiAra(musteriId):
    for i in jsonCustomers[ "<Gerekli Key İle Değiştir> "]:
        if i['id']==musteriId:
            print(f" {musteriId} ID'li müşteri bulundu. Detay Listesi")
            print("=====")
            print(f"Gerekli Outputlar İçin Değiştir")
            break
        else:
            print(f" {musteriId} ID'li müşteri bulunamadı")
```

Adım 17: Siparişleri listeleyecek

```
def siparisListele():
    print("Sipariş Listesi")
    print("+-----+-----+-----+-----+")
    print("|ID          |CustomerId   |OrderDate      |ShipAddress     |")
    print("|ShipCity    |              |                |                 |")
    print("+-----+-----+-----+-----+")
    print("+-----+-----+-----+-----+")
    for i in jsonOrders["<Gerekli Key İle Değiştir>"]:
        epochSaniye = #<Gerekli Key İle Değiştir>
        gunumuzZamani = #<Time Kütüphanesi Gerekli Fonksiyon İle Değiştir>
        print(f"Gerekli Output İçin Formatla")
    print("+-----+-----+-----+-----+")
    print("+-----+-----+-----+-----+")
```

Adım 18: Sipariş Id'ye göre arama yapılacaktır

```
def siparisAra(siparisId):
    for i in jsonOrders["<Gerekli Key İle Değiştir>"]:
        if i['order']['id']==siparisId:
            epochSaniye = #<Gerekli Key İle Değiştir>
            gunumuzZamani = #<Time Kütüphanesi Gerekli Fonksiyon İle Değiştir>
            print(f"{siparisId} ID'li sipariş bulundu. Detay Listesi")
            print("=====")
            print(f"Gerekli Outputlar İçin Değiştir")
            nereye = #<Gerekli Key İle Değiştir>
            cevap = input(f"Kargo Rotasını {nereye.upper()} Şehri İçin Görmek İster misiniz? [e/E] :")
            if cevap.lower()=="e":
                while True:
                    print(f"Varış Noktası {nereye} için Rota Hesaplanacak")
                    nereden = input("Nereden Çıkacak: ")
                    #<MapQuest API'yı Kullanarak Rota Belirlenecek>
                    break
                break
            else:
                print(f"{siparisId} ID'li sipariş bulunamadı")
```

Adım 19: Menü hazırlıkları yapılacaktır

```
while True:
    for i in range(5):
        print()
    secim = int(input("""
Seçiminiz:
[1]    → MÜŞTERİ LİSTELE
[2]    → MÜŞTERİ ARA <MÜŞTERİ ID'E GÖRE>
[3]    → SİPARİŞ LİSTELE
[4]    → SİPARİŞ ARA <SİPARİŞ ID'E GÖRE>
[5]    → ÇIK
"""))
    if secim==1:
        pass
    elif secim==2:
        pass
    elif secim==3:
        pass
    elif secim==4:
        pass
    elif secim==5:
        pass
    else: pass
```


Bölüm 3: Çalışma Anı

Powershell terminal ekranında:

```
PS C:\ python python-h1.py
```

Seçiminiz:

- [1] → MÜŞTERİ LİSTELE
- [2] → MÜŞTERİ ARA <MÜŞTERİ ID'E GÖRE>
- [3] → SİPARİŞ LİSTELE
- [4] → SİPARİŞ ARA <SİPARİŞ ID'E GÖRE>
- [5] → ÇIK

6

Hatalı Seçim!

Seçiminiz:

- [1] → MÜŞTERİ LİSTELE
- [2] → MÜŞTERİ ARA <MÜŞTERİ ID'E GÖRE>
- [3] → SİPARİŞ LİSTELE
- [4] → SİPARİŞ ARA <SİPARİŞ ID'E GÖRE>
- [5] → ÇIK

5

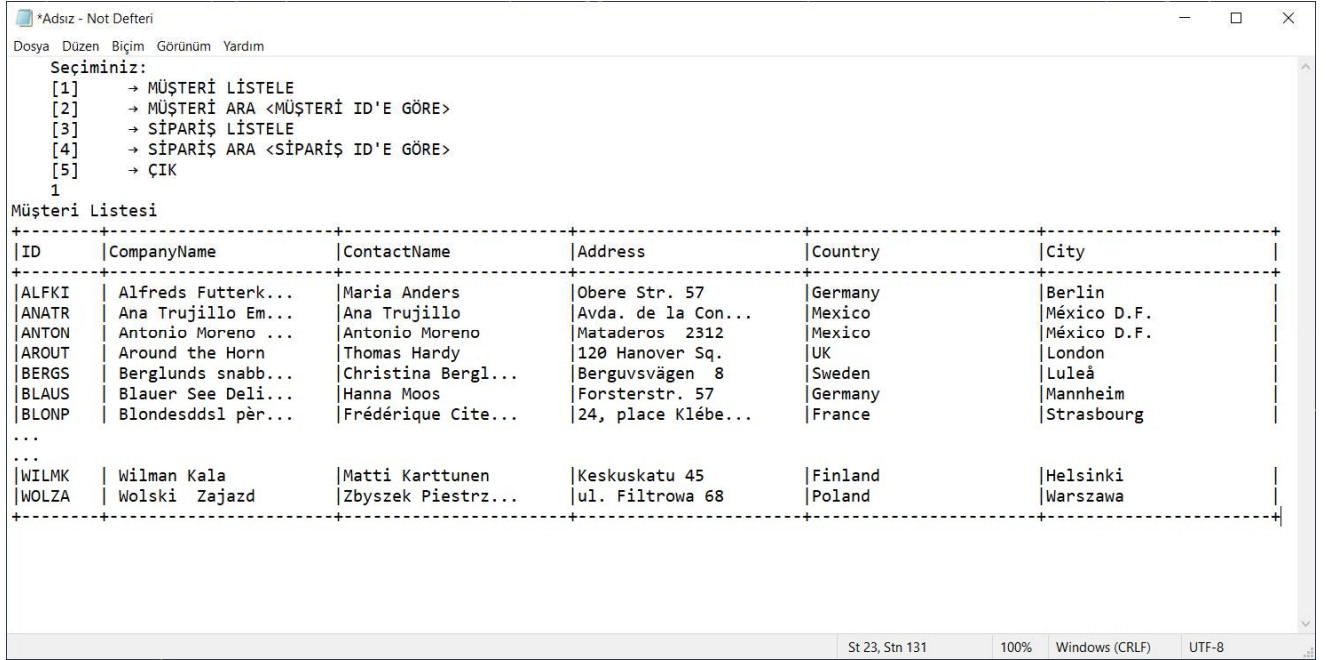
PS C:\>

PYTHON H1 YETENEK SINAVI

Seçiminiz:

- [1] → MÜŞTERİ LİSTELE
- [2] → MÜŞTERİ ARA <MÜŞTERİ ID'E GÖRE>
- [3] → SİPARİŞ LİSTELE
- [4] → SİPARİŞ ARA <SİPARİŞ ID'E GÖRE>
- [5] → ÇIK

1



The screenshot shows a Notepad window with the following content:

Seçiminiz:

- [1] → MÜŞTERİ LİSTELE
- [2] → MÜŞTERİ ARA <MÜŞTERİ ID'E GÖRE>
- [3] → SİPARİŞ LİSTELE
- [4] → SİPARİŞ ARA <SİPARİŞ ID'E GÖRE>
- [5] → ÇIK

1

Müşteri Listesi

ID	CompanyName	ContactName	Address	Country	City
ALFKI	Alfreds Futterk...	Maria Anders	Obere Str. 57	Germany	Berlin
ANATR	Ana Trujillo Em...	Ana Trujillo	Avda. de la Con...	Mexico	México D.F.
ANTON	Antonio Moreno ...	Antonio Moreno	Mataderos 2312	Mexico	México D.F.
AROUT	Around the Horn	Thomas Hardy	120 Hanover Sq.	UK	London
BERGS	Berglunds snabb...	Christina Bergl...	Berguvsvägen 8	Sweden	Luleå
BLAUS	Blauer See Deli...	Hanna Moos	Forsterstr. 57	Germany	Mannheim
BLONP	Blondesddsl për...	Frédérique Cite...	24, place Klébe...	France	Strasbourg
...					
WILMK	Wilman Kala	Matti Karttunen	Keskuskatu 45	Finland	Helsinki
WOLZA	Wolski Zajazd	Zbyszek Piestrz...	ul. Filtrowa 68	Poland	Warszawa

St 23, Stn 131 100% Windows (CRLF) UTF-8

Seçiminiz:

- [1] → MÜŞTERİ LİSTELE
- [2] → MÜŞTERİ ARA <MÜŞTERİ ID'E GÖRE>
- [3] → SİPARİŞ LİSTELE
- [4] → SİPARİŞ ARA <SİPARİŞ ID'E GÖRE>
- [5] → ÇIK

2

Aranan Müşteri Id Giriniz : JHOPS

JHOPS ID'li müşteri bulunamadı

PYTHON H1 YETENEK SINAVI

Seçiminiz:

- [1] → MÜŞTERİ LİSTELE
- [2] → MÜŞTERİ ARA <MÜŞTERİ ID'E GÖRE>
- [3] → SİPARİŞ LİSTELE
- [4] → SİPARİŞ ARA <SİPARİŞ ID'E GÖRE>
- [5] → ÇIK

2

Aranan Müşteri Id Giriniz : CHOPS
CHOPS ID'li müşteri bulundu. Detay Listesi
=====

Id	: CHOPS
Firma Adı	: Chop-suey Chinese
Müşteri Adı	: Yang Wang
İş Disiplini	: Owner
Adres	: Hauptstr. 29
Şehir	: Bern
Posta Kodu	: 3012
Ülke	: Switzerland
Telefon	: 0452-076545
Fax	: ...

PYTHON H1 YETENEK SINAVI

Seçiminiz:

- [1] → MÜŞTERİ LİSTELE
- [2] → MÜŞTERİ ARA <MÜŞTERİ ID'E GÖRE>
- [3] → SİPARİŞ LİSTELE
- [4] → SİPARİŞ ARA <SİPARİŞ ID'E GÖRE>
- [5] → ÇIK

3

*Adsız - Not Defteri						
Dosya Düzen Biçim Görünüm Yardım						
Seçiminiz:						
[1] → MÜŞTERİ LİSTELE						
[2] → MÜŞTERİ ARA <MÜŞTERİ ID'E GÖRE>						
[3] → SİPARİŞ LİSTELE						
[4] → SİPARİŞ ARA <SİPARİŞ ID'E GÖRE>						
[5] → ÇIK						
3						
Sipariş Listesi						
ID	CustomerId	OrderDate	ShipAddress	ShipCity	ShipCountry	
11074	SIMOB	Wed May 6 03:00:00 1998	Vinbæltet 34	Kobenhavn	Denmark	
11075	RICSU	Wed May 6 03:00:00 1998	Starenweg 5	Genève	Switzerland	
11076	BONAP	Wed May 6 03:00:00 1998	12, rue des Bou...	Marseille	France	
11077	RATTC	Wed May 6 03:00:00 1998	2817 Milton Dr.	Albuquerque	USA	
11070	LEHMS	Tue May 5 03:00:00 1998	Magazinweg 7	Frankfurt a.M.	Germany	
11071	LILAS	Tue May 5 03:00:00 1998	Carrera 52 con ...	Barquisimeto	Venezuela	
11072	ERNSH	Tue May 5 03:00:00 1998	Kirchgasse 6	Graz	Austria	
11073	PERIC	Tue May 5 03:00:00 1998	Calle Dr. Jorge...	México D.F.	Mexico	
11067	DRACD	Mon May 4 03:00:00 1998	Walserweg 21	Aachen	Germany	
11068	QUEE	Mon May 4 03:00:00 1998	Alameda dos Can...	Sao Paulo	Brazil	
11069	TORTU	Mon May 4 03:00:00 1998	Avda. Azteca 12...	México D.F.	Mexico	
11064	SAVEA	Fri May 1 03:00:00 1998	187 Suffolk Ln.	Boise	USA	
11065	LILAS	Fri May 1 03:00:00 1998	Carrera 52 con ...	Barquisimeto	Venezuela	
11066	WHITC	Fri May 1 03:00:00 1998	1029 - 12th Ave...	Seattle	USA	
11060	FRANS	Thu Apr 30 03:00:00 1998	Via Monte Bianc...	Torino	Italy	
11061	GREAL	Thu Apr 30 03:00:00 1998	2732 Baker Blvd...	Eugene	USA	
11062	REGGC	Thu Apr 30 03:00:00 1998	Strada Provinci...	Reggio Emilia	Italy	
11063	HUNGO	Thu Apr 30 03:00:00 1998	8 Johnstown Roa...	Cork	Ireland	
11057	NORTS	Wed Apr 29 03:00:00 1998	South House 300...	London	UK	
11058	BLAUS	Wed Apr 29 03:00:00 1998	Forsterstr. 57	Mannheim	Germany	
				St 32, Strn 131	100%	Windows (CRLF) UTF-8

Seçiminiz:

- [1] → MÜŞTERİ LİSTELE
- [2] → MÜŞTERİ ARA <MÜŞTERİ ID'E GÖRE>
- [3] → SİPARİŞ LİSTELE
- [4] → SİPARİŞ ARA <SİPARİŞ ID'E GÖRE>
- [5] → ÇIK

4

Aranan Siparis Id Giriniz : 11074

11074 ID'li sipariş bulundu. Detay Listesi

=====

Sipariş Id : 11074
Müşteri Id : SIMOB
Firma Adı : Simons bistro
Müşteri Adı : Jytte Petersen
Sipariş Tarihi : Wed May 6 03:00:00 1998
Adres : Vinbæltet 34
Şehir : Kobenhavn
Ülke : Denmark

Kargo Rotasını KOBENHAVN Şehri İçin Görmek İster misiniz? [e/E] : h

PYTHON H1 YETENEK SINAVI

Seçiminiz:

- [1] → MÜŞTERİ LİSTELE
- [2] → MÜŞTERİ ARA <MÜŞTERİ ID'E GÖRE>
- [3] → SİPARİŞ LİSTELE
- [4] → SİPARİŞ ARA <SİPARİŞ ID'E GÖRE>
- [5] → ÇIK

4

Aranan Siparis Id Giriniz : 11074
11074 ID'li sipariş bulundu. Detay Listesi

=====

Sipariş Id : 11074
Müşteri Id : SIMOB
Firma Adı : Simons bistro
Müşteri Adı : Jytte Petersen
Sipariş Tarihi : Wed May 6 03:00:00 1998
Adres : Vinbæltet 34
Şehir : Kobenhavn
Ülke : Denmark

Kargo Rotasını KOBENHAVN Şehri İçin Görmek İster misiniz? [e/E] : e

Varış Noktası KOBENHAVN için Rota Hesaplanacak

Nereden Çıkacak: Odder

=====

Kargo Rotası ODDER den/dan KOBENHAVN e/a/ye/ya

Toplam Süre: 02:53:45

Kilometre: 300.73

=====

Start out going south on 451/Rådhusgade toward Nørregade. Continue to follow 451. (0.94 km)

Enter next roundabout and take the 1st exit onto 445. (9.17 km)

Enter next roundabout and take the 2nd exit onto 445/Oddervej. (8.25 km)

Turn left onto 170/Ladegårdsbakken. Continue to follow 170. (0.78 km)

170 becomes 445/Skanderborgvej. (1.48 km)

Turn left onto Skanderborgvej. (0.01 km)

Stay straight to go onto ramp. (0.51 km)

Merge onto E45/E45. (56.81 km)

Stay straight to go onto E20/Østjyske Motorvej. Continue to follow E20 (Portions toll). (200.64 km)

Take O4/Motorring 4. (3.25 km)

Keep right at the fork to go on 21. (10.11 km)

21 becomes O2. (3.64 km)

Turn left onto O2/Vasbygade. (1.46 km)

Turn slight right onto ramp. (0.27 km)

Turn left onto Dybbølsbro. (0.31 km)

Turn right onto Ingerslevsgade. (0.77 km)

Turn slight right onto Tietgensgade. (0.57 km)

Turn left onto H.C. Andersens Boulevard. (0.88 km)

Turn right onto Nørre Farimagsgade. (0.75 km)

Nørre Farimagsgade becomes Øster Farimagsgade. (0.11 km)

Welcome to COPENHAGEN. (0.00 km)

=====

Öğrenci Script'i

```
#####  
# Bu program:  
# - Customer tablosundan gelen json'u parse ederek listeler  
# - Order tablosundan gelen json'u parse ederek listeler  
# - Kullanıcıya "/customers" istegi için aramak istediği müşteri id'sorar.  
# - Kullanıcıya "/orders" istegi için aramak istediği order id'sorar.  
# - MapQuest API için kişisel Credential alır  
# - GPS koordinatlarını MapQuest API'yı kullanarak "location" bilgisinden keşfeder.  
# - MapQuest API'yı kullanarak kargo elemanına rota çizer.  
# - Epoch time modülünü kullanarak ekrana zaman bilgisini formatlar.  
# - Runtime'da hata vermemesi için Dictionary'de "key" olup olmadığını keşfederek kod  
yazar.  
# - Dönen json'u konsepti "json beautifier" olan google araması ile daha okunaklı hale  
getirir  
# - https://jsonformatter.curiousconcept.com/ sitesini referan alır  
#  
# Öğrenci:  
# 1. API request", "URL parse", "epoch time conversion" için kütüphaneleri import edecek.  
# 2. Customers API için gerekli URL'leri girecek  
# 3. Customers API için GET request kullanacak  
# 4. if case, bu kısımda her hangi bir kod değişikliği yapmayacak, request status  
kontrol ediliyor  
# 5. Response içeriği referans amaçlı print edilecek  
# 6. Response icerigi json data formatına encode edilecek  
# 7. Orders API için gerekli URL'leri girecek  
# 8. Orders API için GET request kullanacak  
# 9. if case, bu kısımda her hangi bir kod değişikliği yapmayacak, request status  
kontrol ediliyor  
# 10. Response içeriği referans amaçlı print edilecek  
# 11. Response icerigi json data formatına encode edilecek  
# 12. Mapquest API için gerekli URL'leri girecek  
# 13. Mapquest Credential için; token, key... hazırlıkları yapılacak  
# 14. UI manipulasyonu. Output anında 15 karakterden uzun metinler için manipulasyon  
yapılacak  
# 15. Müşterileri listeleyecek  
# 16. Müşterileri Id'ye göre arama yapılacak  
# 17. Siparişleri listeleyecek  
# 18. Sipariş Id'ye göre arama yapılacak  
# 19. Menü hazırlıkları yapılacak  
#####
```

```
# 1 → Kütüphaneleri import edelim
# API request",
# "URL parse",
# "epoch time conversion"
import #<Gerekli Kütüphaneler İle Değiştir>

# 2 → Customers API için gerekli URL girelim
urlCustomers = "<Gerekli URL İle Değiştir>"

# 3 → Customers API için GET request
rCustomers = #<Request>

# 4 → if case, bu kısımda her hangi bir kod değişikliği yapmayın lütfen
if not rCustomers.status_code == 200:
    raise Exception("API Bağlantı Sorunu. Status code: {}. Text: {}".format(
        rCustomers.status_code, rCustomers.text))

# 5 → Response içeriği print et referans amaçlı
#print(rCustomers.text)

# 6 → Response icerigi json data formatına encode et
jsonCustomers = #<Gerekli Json Encode Kodu İle Değiştir>

# 7 → Orders API için gerekli URL'leri girelim
urlOrders = "<Gerekli URL İle Değiştir>"

# 8 → Orders API için GET request
rOrders = #<Request>

# 9 → if case, bu kısımda her hangi bir kod değişikliği yapmayın lütfen
if not rOrders.status_code == 200:
    raise Exception("API Bağlantı Sorunu. Status code: {}. Text: {}".format(
        rOrders.status_code, rOrders.text))

# 10 → Response içeriği print et referans amaçlı
#print(rOrders.text)

# 11 → response icerigi json data formatına encode et
```

```
# jsonOrders = #<Gerekli Json Encode Kodu İle Değiştir>

# 12 → Mapquest API için gerekli URL'leri girelim
mainMapApiUrl="#<Gerekli URL İle Değiştir>"

# 13 → Mapquest Credential için; token, key... hazırlıkları yapalım
mapApiKey="Kisisel_Key"

# 14 → UI manipülasyonu. Output anında 15 karakterden uzun metinler için manipülasyon yapalım
def metinKontrol(metin):
    if len(metin)>15:
        return f"<Gerekli Kod İle Değiştir>"
    else:
        return f"<Gerekli Kod İle Değiştir>"

# 15 → Müşterileri listeleyelim
def musteriliste():
    print("Müşteri Listesi")
    print("+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+")
    print("|ID          |CompanyName           |ContactName         |Address            |Country           |City              |")
    print("+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+")

    for i in jsonCustomers["<Gerekli Key İle Değiştir>"]:
        print(f"Gerekli Output İçin Formatla")

    print("+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+")
    print("+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+")

# 16 → Müşterileri Id'ye göre arama yapalım
def musteriar(musteriid):
    for i in jsonCustomers["<Gerekli Key İle Değiştir>"]:
        if i['id']==musteriid:
            print(f"{musteriid} ID'li müşteri bulundu. Detay Listesi")
            print("====================")
            print(f"Gerekli Outputlar İçin Değiştir")
            break
    else:
```


19 → menu

```
while True:
    for i in range(5):
        print()
    secim = int(input("""
Seçiminiz:
[1]    → MÜŞTERİ LİSTELE
[2]    → MÜŞTERİ ARA <MÜŞTERİ ID'E GÖRE>
[3]    → SİPARİŞ LİSTELE
[4]    → SİPARİŞ ARA <SİPARİŞ ID'E GÖRE>
[5]    → ÇIK
"""))
    if secim==1:
        pass
    elif secim==2:
        pass
    elif secim==3:
        pass
    elif secim==4:
        pass
    elif secim==5:
        pass
    else:
        pass
```

KAYNAKÇA/REFERANS

- 1→ <https://www.netacad.com/>
- 2→ <https://pythoninstitute.org/>