



BATCH : **BATCH - 81**

LESSON : **API**

DATE : **24.10.2022**

SUBJECT : **API Tanımları ve Postman
Kullanımı**

Ali Cengiz KASAR



techproeducation



techproeducation



techproeducation



techproeducation



techproedu

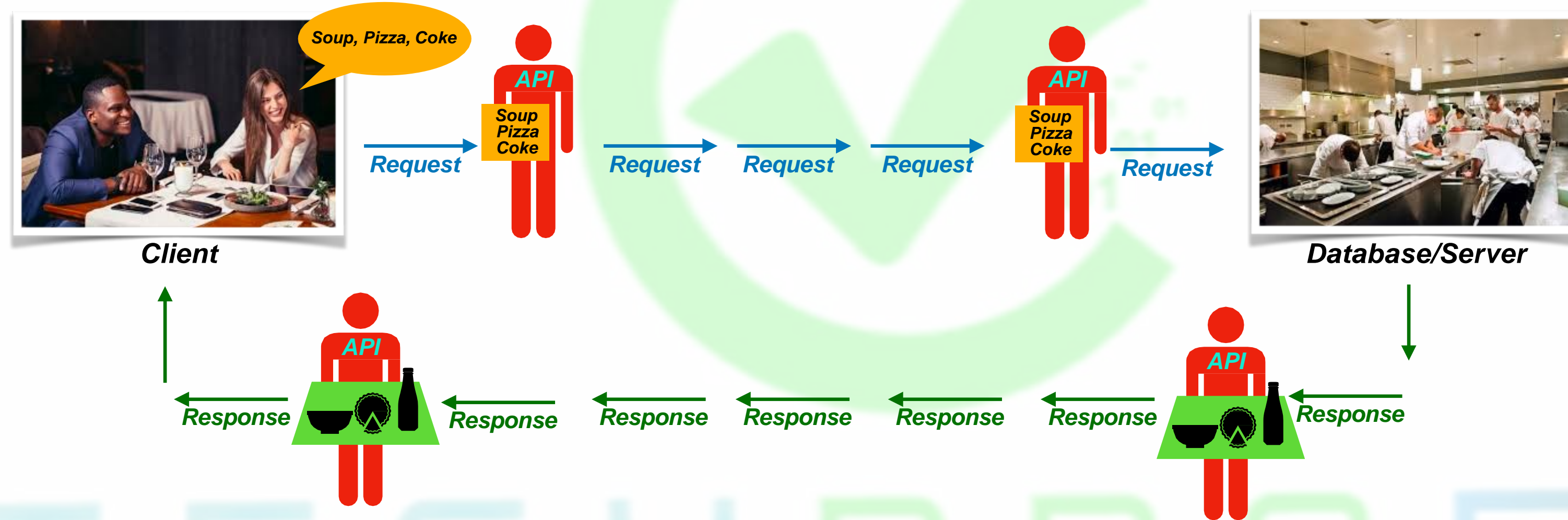
***A**pplication **P**rogramming **I**nterface*

API

(Uygulama Programlama Arayüzü)

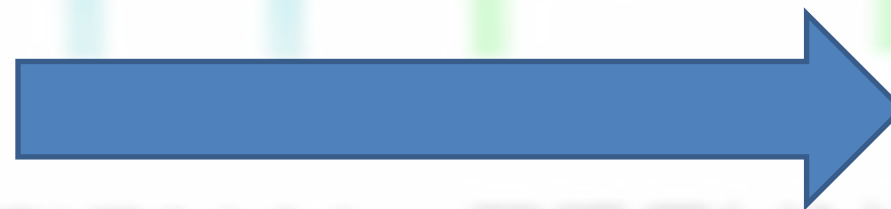
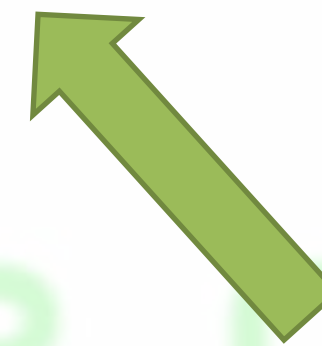
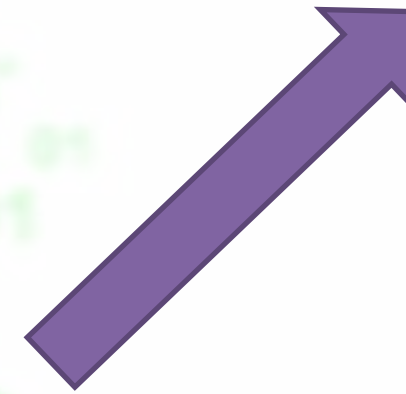
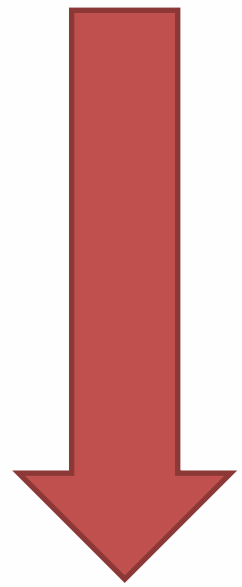
API Nedir?

- . **API** ==> **A**pplication **P**rogramming **I**nterface.
- . **API**' nin UI (User Interface - Kullanıcı Arayüzü) yoktur.

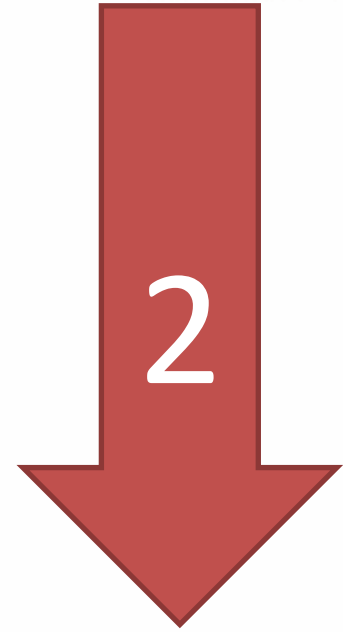


API' ı bir restoranda garson olarak düşünelim.

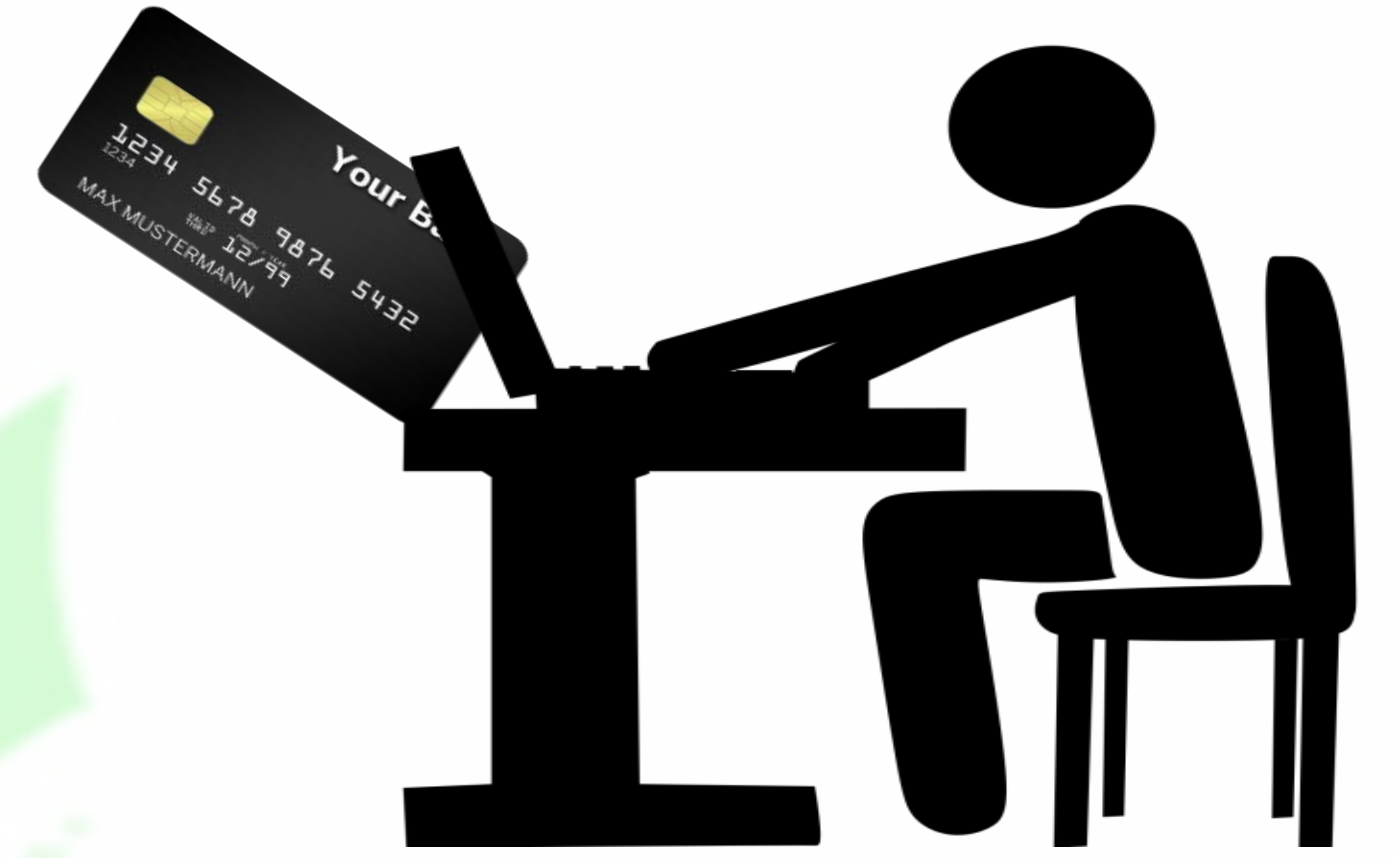
Bir garson olarak, **API bir istemciden bir istek alır (Request)**, bu isteği veri tabanına alır, istenen verileri veri tabanından alır ve **istemciye bir yanıt (Response) döndürür**.



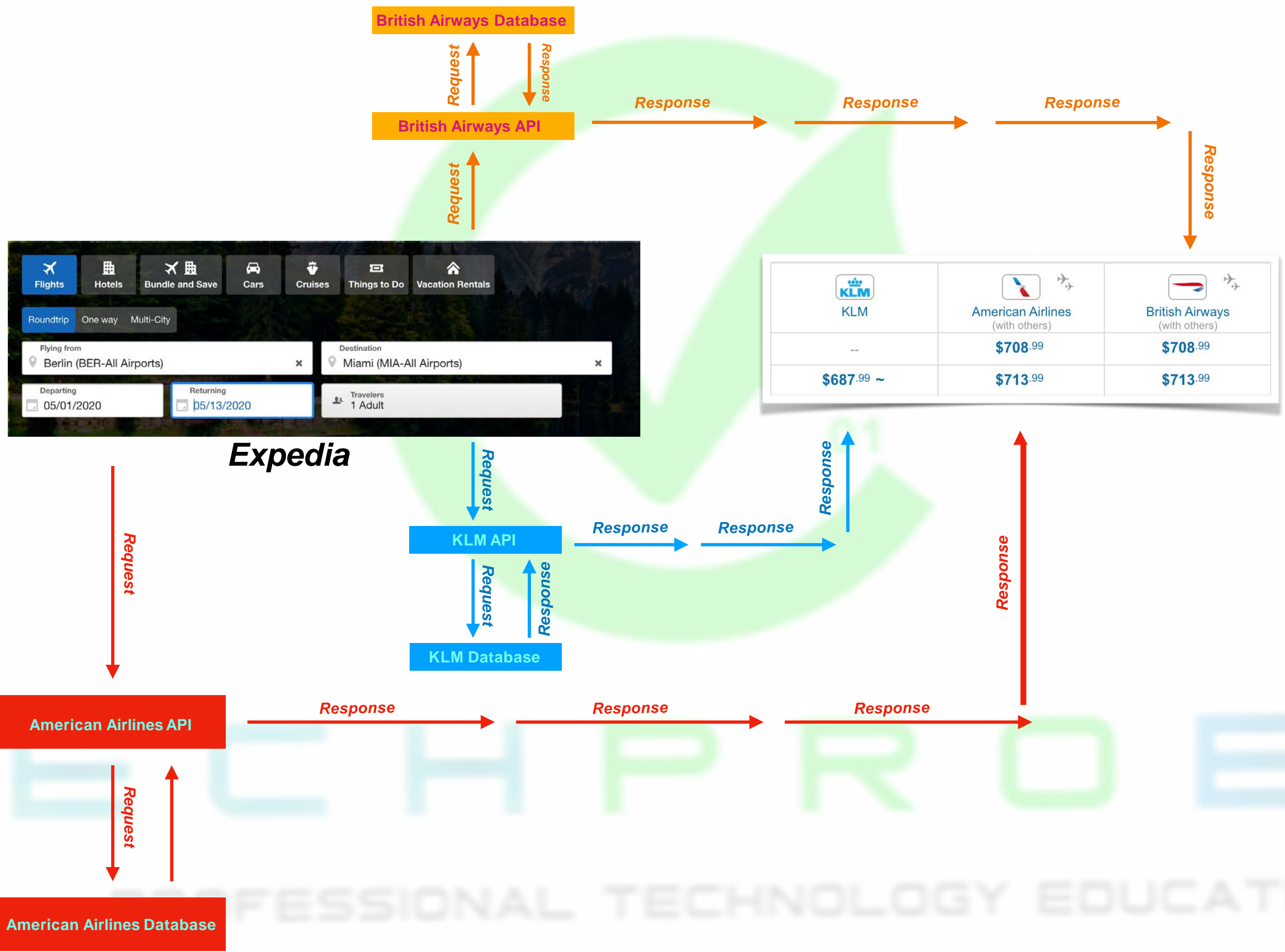
CHPRO
PROFESSIONAL TECHNOLOGY ED



BANKALARARASI
KART MERKEZI



API Nasıl Çalışır?



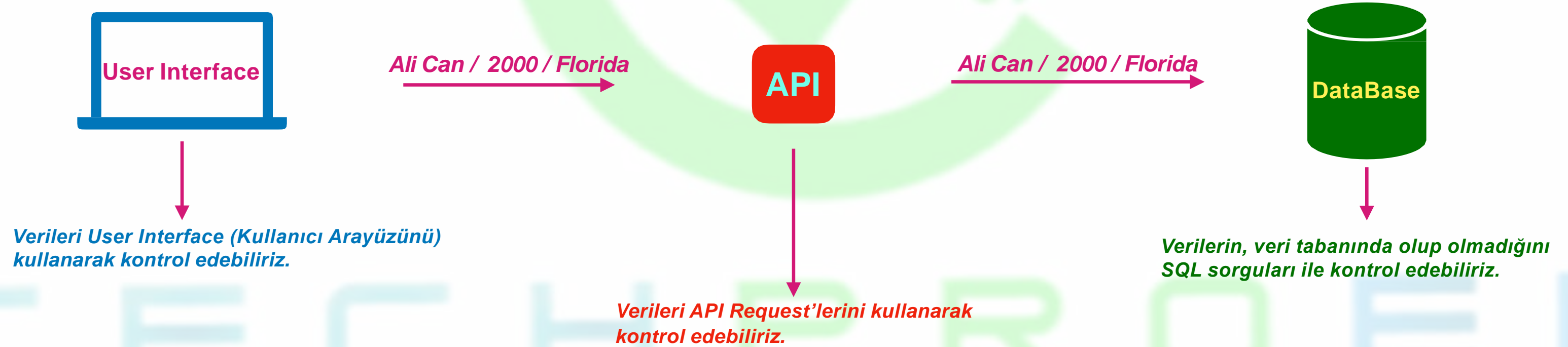
API Testi Nedir?

API testi, API' ın bir uygulamanın işlevselliği, güvenilirliği, performansı ve güvenliği açısından beklentileri karşılayıp karşılamadığını kontrol etmek için yapılır.

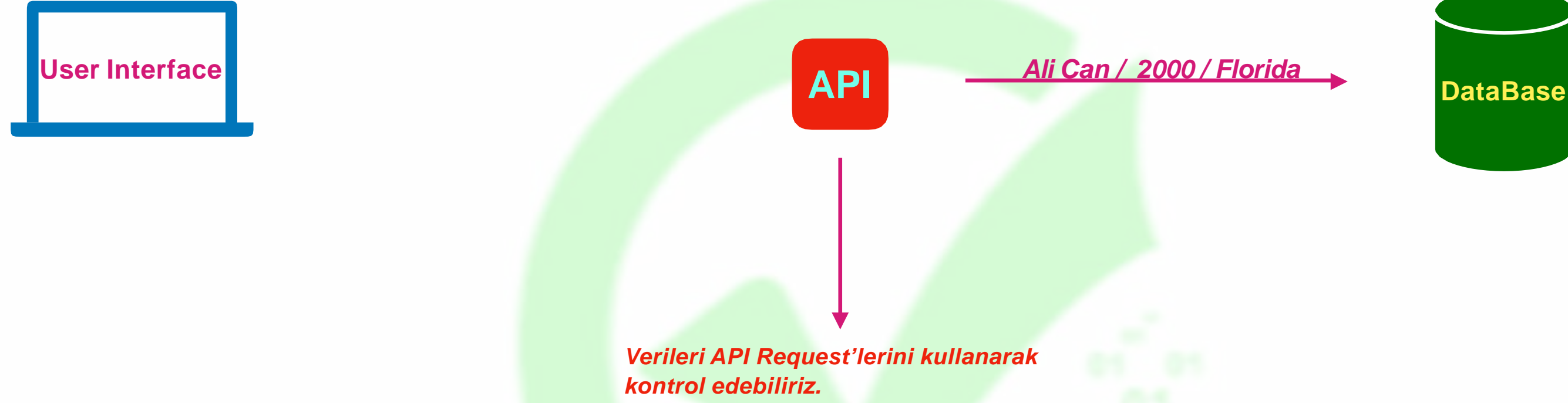
API testi uygulamanın görünümü ile ilgilenmez.

End to End (E2E) Senaryosu:

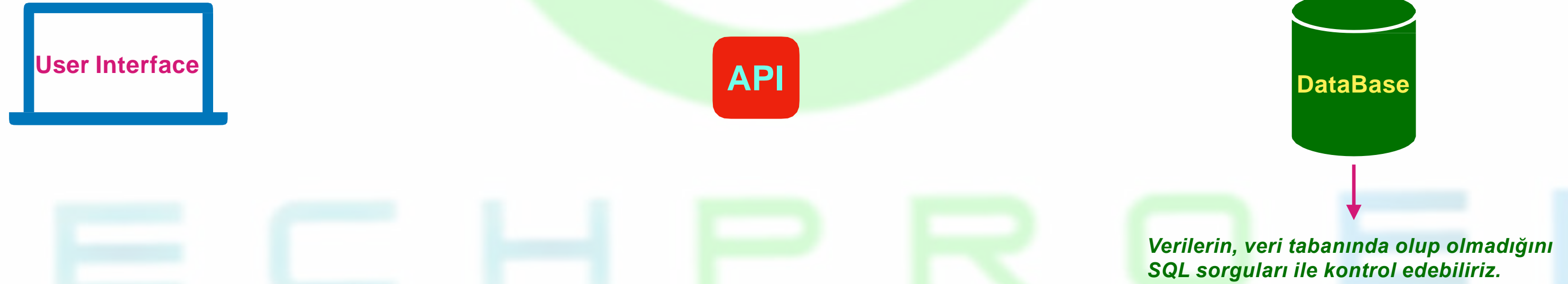
1) Create/Update/Delete kullanıcı arayüzü kullanarak verileri oluşturabilir, silebilir veya güncelleyebiliriz.



2) Create/Update/Delete API kullanarak verileri oluşturabilir, silebilir veya güncelleyebiliriz.



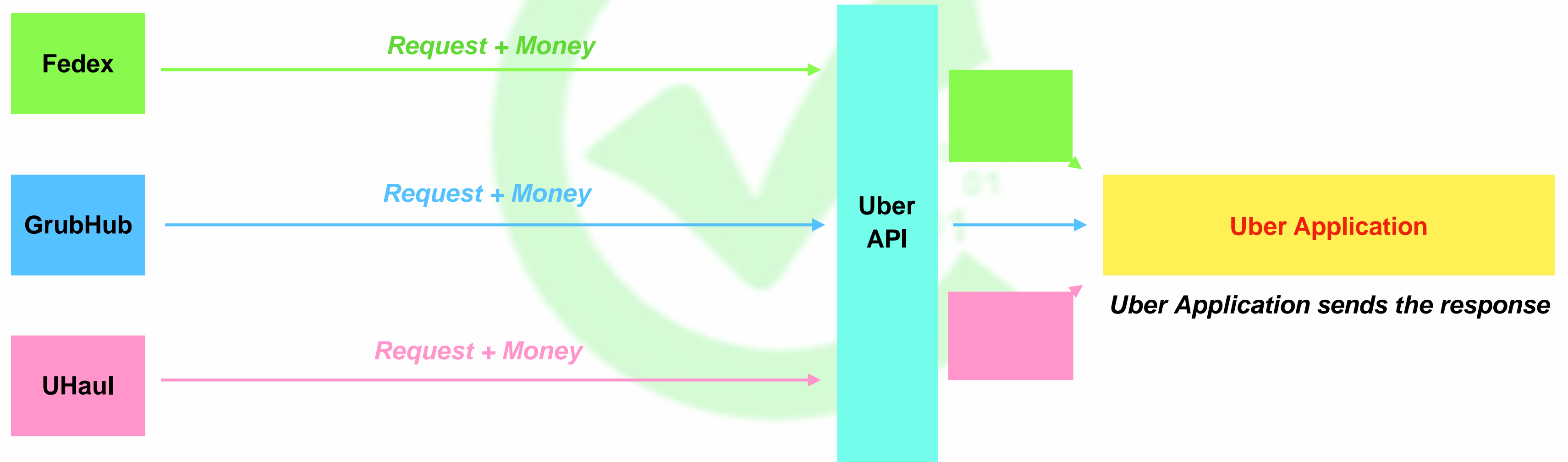
3) Create/Update/Delete SQL komutlarını kullanarak verileri oluşturabilir, silebilir veya güncelleyebiliriz.



TECHPRO

PROFESSIONAL TECHNOLOGY EDUCATION

Not: Her şirket kendi API'larını oluşturur ve bu hizmetleri satarak para kazanabilirler.



TECHPROED

PROFESSIONAL TECHNOLOGY EDUCATION

API ve Web Service Arasındaki Fark?

API ve Web Service uygulamalar arasında iletişim oluştururlar.

Web Service

Aralarındaki tek fark **Web Service** iki uygulama arasındaki iletişimi **internet** aracılığı ile yapar.

API

Microsoft Word gibi masaüstü uygulamaları, **internete ihtiyaç duymayan** kendi API' larını kullanır. Bu yüzden Microsoft Word kullandığımızda Microsoft Word API kullanıyoruz.

Not: Bütün Web Service'ler API'dır. Ancak bütün API'lar Web Service değildir.

TECHPRO ED

PROFESSIONAL TECHNOLOGY EDUCATION

HTTP Nedir?

HTTP → **H**yper **T**ext **T**ransfer **P**rotocol

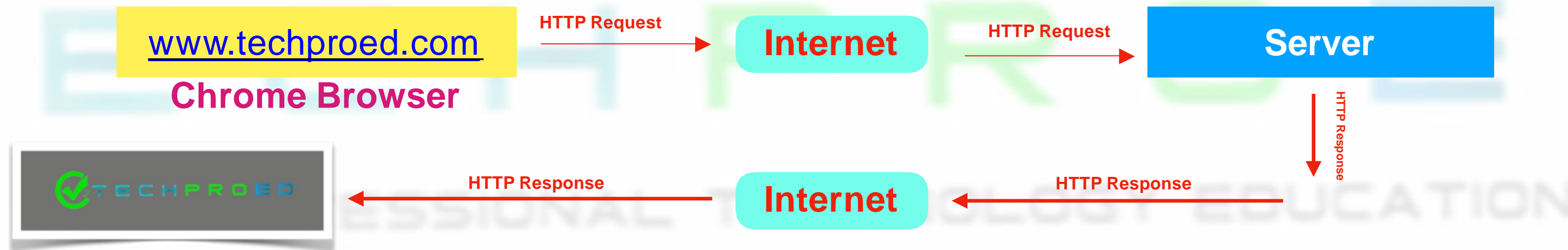
İstemci (**Client**) bilgisayarlar ve web sunucuları arasındaki iletişim, **HTTP Request** ve **HTTP Response** göndererek yapılır.

İstemciler (**Client**) genellikle tarayıcılardır (Chrome, Edge, Safari), ancak herhangi bir program veya cihaz türü olabilirler.

Sunucular (**Server**) genellikle buluttaki bilgisayarlardır.

Client ve Server arasındaki iletişim, request ve response lar ile yapılır::

1. Bir Client (bir tarayıcı) web'e bir HTTP Request gönderir.
2. Bir Web Server Request'i alır.
3. Server, Request'i işlemek için bir uygulama çalıştırır.
4. Server tarayıcıya bir **HTTP Response** (çıkıtı) gönderir.
5. Client (Browser) Response'u alır.



API Testinde Tam Olarak Neyin Doğrulanması Gerekir?

API testinde, bizler API'a bilinen verilerle bir istek göndeririz ve cevabı test ederiz.

1) HTTP Status Kodları

2) Data Doğruluğu

HTTP Status Codes



Params Authorization Headers Body Pre-request Script Tests Cookies Code Comments (0)

Query Params

KEY	VALUE	DESCRIPTION	...	Bulk Edit
Key	Value	Description		

Body Cookies Headers (22) Test Results Status: 200 OK Time: 227 ms Size: 1.3 KB Download

Pretty Raw Preview JSON

```
1 {
2   "type": "FeatureCollection",
3   "metadata": {
4     "generated": 1555900401000,
5     "url": "https://earthquake.usgs.gov/earthquakes/feed/v1.0/summary/4.5_hour.geojson",
6     "title": "USGS Magnitude 4.5+ Earthquakes, Past Hour",
7     "status": 200,
8     "api": "1.8.1",
9     "count": 1
10  },
11  "features": [
12    {
13      "type": "Feature",
14      "properties": {
15        "mag": 4.6,
16        "place": "92km WNW of Neiafu, Tonga",
17        "time": 1555898797799,
18        "updated": 1555900126040,
19        "tz": -720,
20        "url": "https://earthquake.usgs.gov/earthquakes/eventpage/us70003a3x",
```

Ortak HTTP Status Kodları

- 1) 200 (OK) ==> Talep karşılandı. Request gerçekleşti.
- 2) 201 (CREATED) ==> Database' e gönderilen datanın oluşturulduğunu söyler. (200 olarak da gelebilir)
- 3) 204 (NO CONTENT) ==> Talep yerine getirildi ancak size cevap (response) dönemiyoruz.
- 4) 400 (BAD REQUEST) ==> Syntax yanlış, yüksek boyutlu data ya da diğer kullanıcı hatası...
- 5) 403 (FORBIDDEN) ==> Bu dataya erişim izniniz yok. (400 olarak da gelebilir)
- 6) 404 (NOT FOUND) ==> Aranılan data bulunamadı.
- 7) 500 (INTERNAL SERVER ERROR) ==> Server kaynaklı bir hatadır, server'ın düzelmesini bekleriz.

TECHPROED

PROFESSIONAL TECHNOLOGY EDUCATION

API Testlerinde Hani Programları Kullanırsınız?

1) Postman

Postman, geliştiricilerin API'ları **oluşturmasını**, **paylaşmasını** ve **test etmesini** ve **belgelemesini** sağlayan popüler bir uygulamadır.



2) SOAP UI

SOAP → **S**imple **O**bject **A**ccess **P**rotocol. (Basit Nesne Erişim Protokolü)

Web uygulamalarının internet üzerinden haberleşebilmesi önemlidir.

HTTP, tüm internet tarayıcıları ve sunucuları tarafından desteklendiğinden, uygulamalar arasında iletişim kurmanın **en iyi yolu HTTP** üzerindendir. **SOAP** bunu başarmak için oluşturulmuştur.

SOAP, farklı teknolojiler ve programlama dilleri ile farklı işletim sistemlerinde çalışan uygulamalar arasında iletişim kurmak için bir **yol veya biçim** sağlar. **SOAP** platformlardan bağımsızdır.

SOAP, XML formatında çalışır.

```
<customer>
  <customer_id> 1001 </customer_id>
  <customer_name> Mark Star </customer_name>
</customer>
```


3) REST / RestFul

REST → **RE**presentational **S**tate **T**ransfer

REST sistemlerin birbirleri ile iletişim kurmalarını kolaylaştırır.

REST mimarisinde, client' lar dataları **almak (retrive)** veya **değiştirmek (modify)** için request gönderir ve server'lar bu request'lere response gönderir.

REST , XML ve JSON (Java Script Object Notation) formatlarının ikisini de kullanabilir. Bu sebeple SOAP dan daha yaygın bir kullanımı vardır.

```
<customer>
  <customer_id> 1001 </customer_id>
  <customer_name> Mark Star </customer_name>
</customer>
```

XML Format

```
{
  "customer_id": 1001,
  "customer_name": "Mark Star"
}
```

JSON Format (key- value)
(Map'lerdeki gibi)

Bir REST Request'i Nelerden Oluşur?

1) Bir **HTTP** yöntemi, ne tür bir işlemin gerçekleştirileceğini tanımlar.

- A) GET , dataları **okumak** içindir. ==> Bu sitede hotel arayalım www.hotels.com
- B) POST , yeni data **oluşturmak** içindir. ==> Sitede hotel rezervasyonu yapalım www.hotels.com
- C) PUT dataları tamamen **güncellemek** içindir. ==> Tüm rezervasyon datasını değiştirelim www.hotels.com
- D) PATCH, dataları **kısmen güncellemek** içindir. ==> Sadece rezervasyon tarihini değiştirelim www.hotels.com
- E) DELETE, dataları **silmek** içindir. ==> Rezervasyonu iptal edelim. www.hotels.com

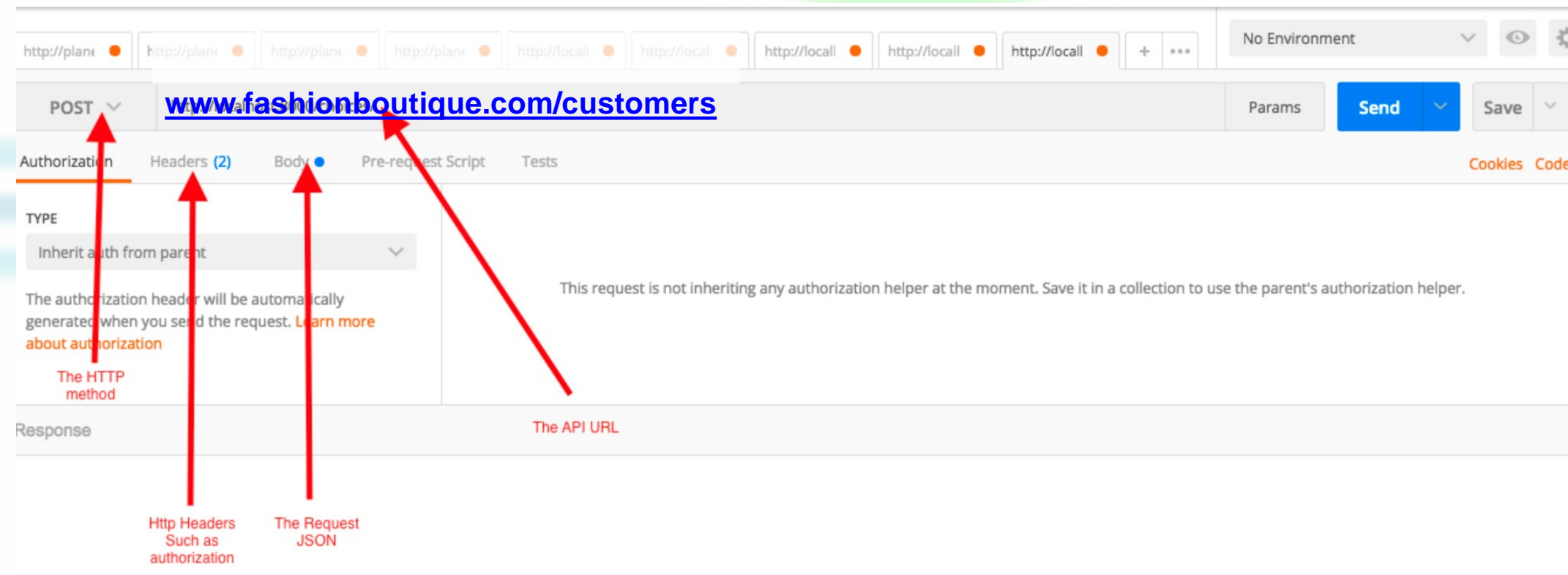
2) **HEADER (Başlık)**'da, istemci (Client) sunucudan alınacak **içeriğin türünü** gönderir.

Header (Başlık), sunucunun istemci tarafından anlaşılamayan veya işlenemeyen verileri göndermemesini sağlar.

Bir kaynağa **PATH (Yol)** (**Endpoint/URL/URI**) oluşturma :

İstekler (**Request**), işlemin gerçekleştirileceği bir kaynağa giden yolu içermelidir.

Bitiş Noktası (Endpoint) bu şekilde olacaktır, www.fashionboutique.com/customers



API Endpoint Nedir?

API'nin istek gönderdiği ve kaynağın yaşadığı yer endpoint'tir. API developer'lar bir API oluşturduğunda genel endpoint'ler (URL) oluştururlar ve hangi endpoint'in hangi HTTP Request yöntemleriyle çalışacağına karar verirler.

<http://petstore.swagger.io/>

URL / Method Adı

POST	/pet	Add a new pet to the store
PUT	/pet	Update an existing pet

GET	/pet/{petId}	Find pet by ID
POST	/pet/{petId}	Updates a pet in the store with form data
DELETE	/pet/{petId}	Deletes a pet

Testerlar aşağıdaki komutları kullanarak, beklendiği gibi çalışırsa, tüm endpoint'leri test ederler.

GET, POST, PUT, DELETE HTTP Request

TECHPROED

PROFESSIONAL TECHNOLOGY EDUCATION

Swagger Nedir?

Swagger, API' nızın, zayıf kaynak kodu bilgisine sahip olsalar bile geliştiriciler ve testerlar tarafından okunması ve anlaşılması kolay ortak bir dil kullanarak tanımlamak için bir framework'tür.

Bunu bir ev planı gibi düşünebilirsiniz.

İstediğiniz yapı malzemesini kullanabilirsiniz ancak evin planının dışına çıkamazsınız.

API belgeleri için SWAGGER kullandım.

SWAGGER bana API endpoint'leri veriyor ve bunların nasıl kullanılacağı hakkında beni bilgilendiriyor.

TECHPROED

PROFESSIONAL TECHNOLOGY EDUCATION

Postman Paneli

Düzenlemek ve tüm request'leri saklamak için koleksiyon oluşturabilirsiniz.

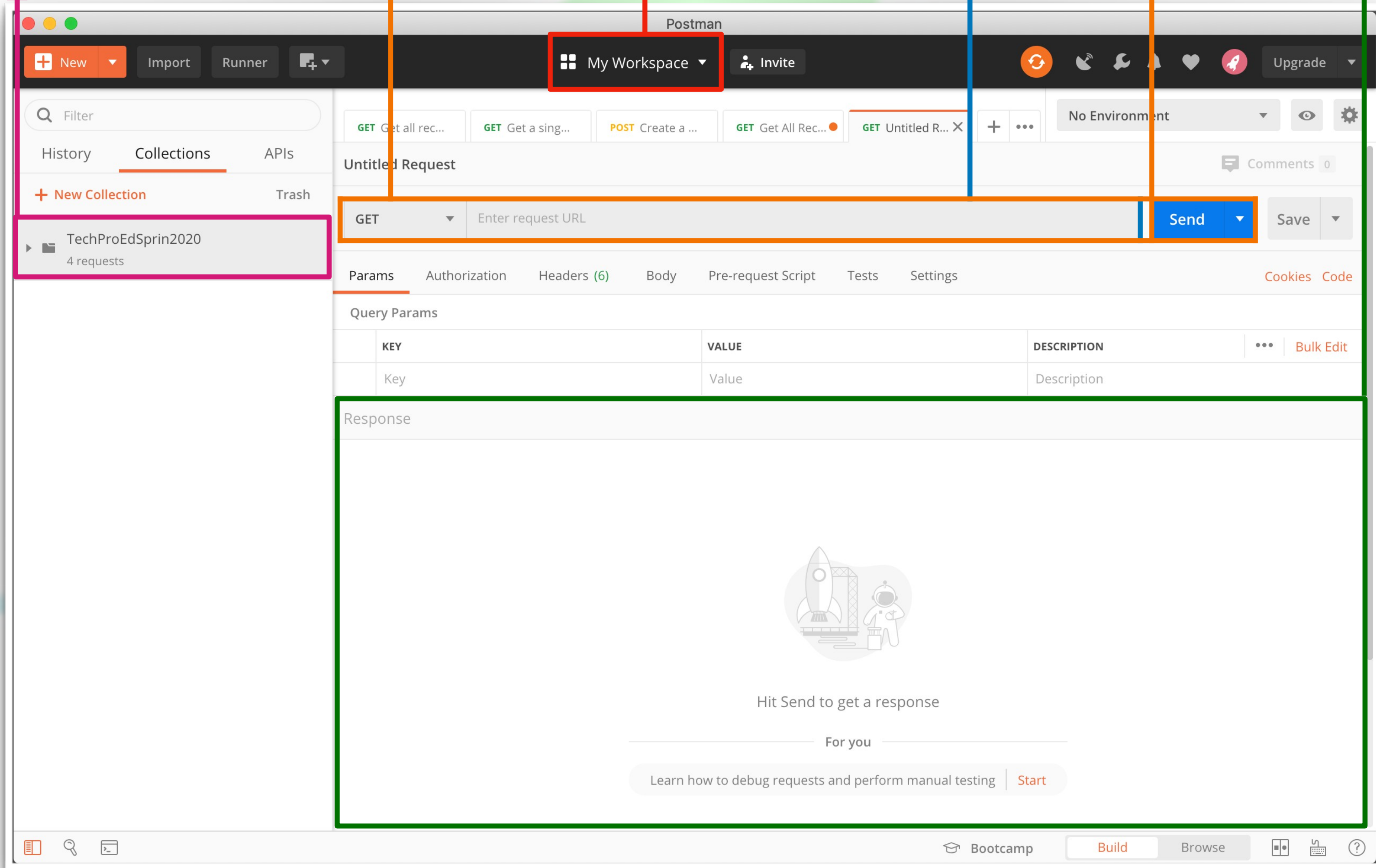
Request Yöntemi

Çalışma Alanı Adı

Request URL bar

Gönder Düğmesi

Response penceresi



GET Request Nasıl Çalışır?

Postman, end-point URL'ini girdiğimiz backend testi için kullanılır. Request'i sunucuya gönderir ve sunucudan yanıtı alır.

Base URL	Base URL	Method	Endpoints	Description
https://restful-booker.herokuapp.com	JSON	GET	/booking	Tüm çalışan verileri alır
		GET	/booking/1	Tek bir çalışan veri alır
		POST	/booking? firstname=Suleyman&lastname=Alptekin&totalprice=123&depositpaid=true&additionalneeds=Wifi	Veri tabanında yeni kayıt oluşturur
		PATCH	/booking/12	Veriyi kısmen günceller
		PUT	/booking/11	Veriyi tamamen günceller
		DELETE	/booking/11	Tüm çalışan veriyi siler

TECHPROED

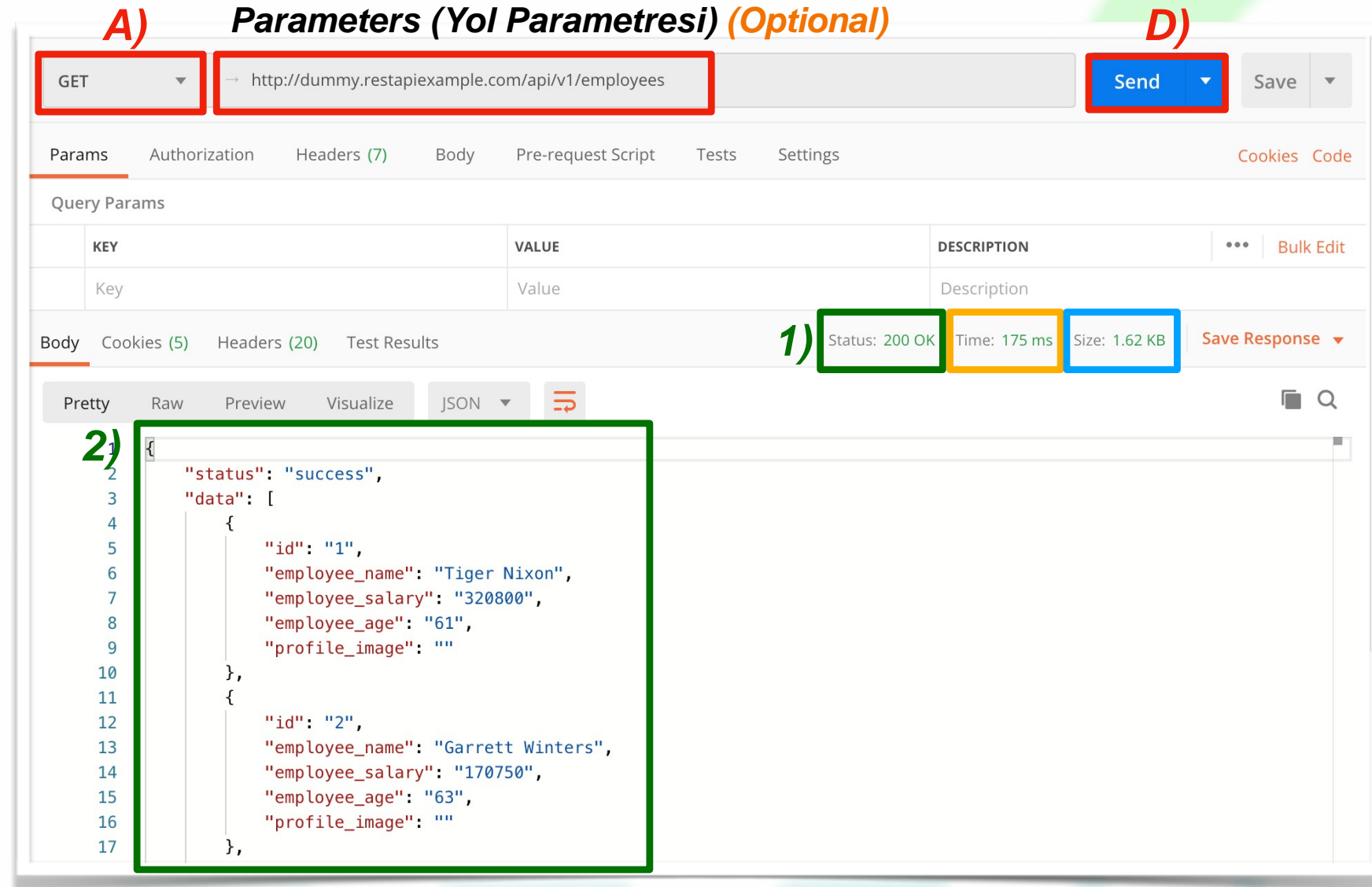
PROFESSIONAL TECHNOLOGY EDUCATION

GET Request Nasıl Çalışır?

C) Headers'da kullanılacak cevap formatını seçmek.
XML ya da JSON (Optional) olabilir.

B) REST API URL/Endpoint (Mandatory)

B) Query Parameters (Sorgu Parametresi) ya da Path
Parameters (Yol Parametresi) (Optional)



A=>B=>C=>D GET Request

1) Status Kodu

2) JSON Formatında Data

3) Header (Başlık)

API

DataBase

Note: Yanıt aldıktan sonra öncelikle **status code** kontrol ederiz.

Note: **Accept** istemcinin (Client) sunucudan (server) ne tür bir yanıtı kabul edebileceğini belirtir. Ancak API beklenen biçimi desteklemelidir, aksi takdirde API yanıtı varsayılan biçimde gönderir.

Note: GET Request yapabilmeniz için yetkilendirilmiş olmanız gerekmektedir.

POST Request Nasıl Çalışır?

The screenshot shows a REST client interface with a POST request configured. The request is to the URL `https://jsonplaceholder.typicode.com/users`. The body is in raw JSON format, containing a user object. The response status is 201 Created, and the response body is also in JSON format, showing the created user data.

Labels in the image:

- A) Method: POST
- B) URL: `https://jsonplaceholder.typicode.com/users`
- C) Body Type: raw
- D) Body Format: JSON
- E) Request Body (JSON):

```
{  "id": 11,  "name": "Suleyman Alptekin",  "username": "SAlp",  "email": "selam73@yahoo.com",  "address": {    "street": "Walk street",    "suite": "Apt. 123",    "city": "Hialeah",    "zipcode": "33018-3874",    "geo": {      "lat": "-37.3159",      "lng": "81.1496"    }  },  "phone": "1-770-123-8031 x56442",  "website": "slptkn73.org",  "company": {    "name": "TechProEd LLC",    "catchPhrase": "Full Stack Automation Tester Course",    "bs": "Real-time e-school"  }}
```
- F) Send button
- 1) Status Code: 201 Created
- 2) Response Body (JSON):

```
{  "id": 11,  "name": "Suleyman Alptekin",  "username": "SAlp",  "email": "selam73@yahoo.com",  "address": {    "street": "Walk street",    "suite": "Apt. 123",    "city": "Hialeah",    "zipcode": "33018-3874",    "geo": {      "lat": "-37.3159",      "lng": "81.1496"    }  },  "phone": "1-770-123-8031 x56442",  "website": "slptkn73.org",  "company": {    "name": "TechProEd LLC",    "catchPhrase": "Full Stack Automation Tester Course",    "bs": "Real-time e-school"  }}
```

A=>B=>C=>D=>E=>F POST Request

1) Status Code

2) Data in JSON Format

Note: Yanıt aldıktan sonra öncelikle **status code** kontrol ederiz.
Note: POST Request yapabilmeniz için yetkilendirilmiş olmanız gerekmektedir.

API

DataBase

PUT Request Nasıl Çalışır?

The screenshot shows a REST client interface with a PUT request configured. The URL is `https://restful-booker.herokuapp.com/booking/11`. The request body is a JSON object: `{ "firstname": "Suleyman", "lastname": "Alptekin", "totalprice": 321, "depositpaid": true, "bookingdates": { "checkin": "2020-05-02", "checkout": "2020-05-05" }, "additionalneeds": "Breakfast" }`. The response status is 200 OK, and the response body is the same JSON object.

Labels in the image:

- A) URL
- B) Method (PUT)
- C) Body tab
- D) JSON format
- E) Request body
- F) Send button
- 1) Status: 200 OK
- 2) Response body

A=>B=>C=>D=>E=>F PUT Request

1) Status Code

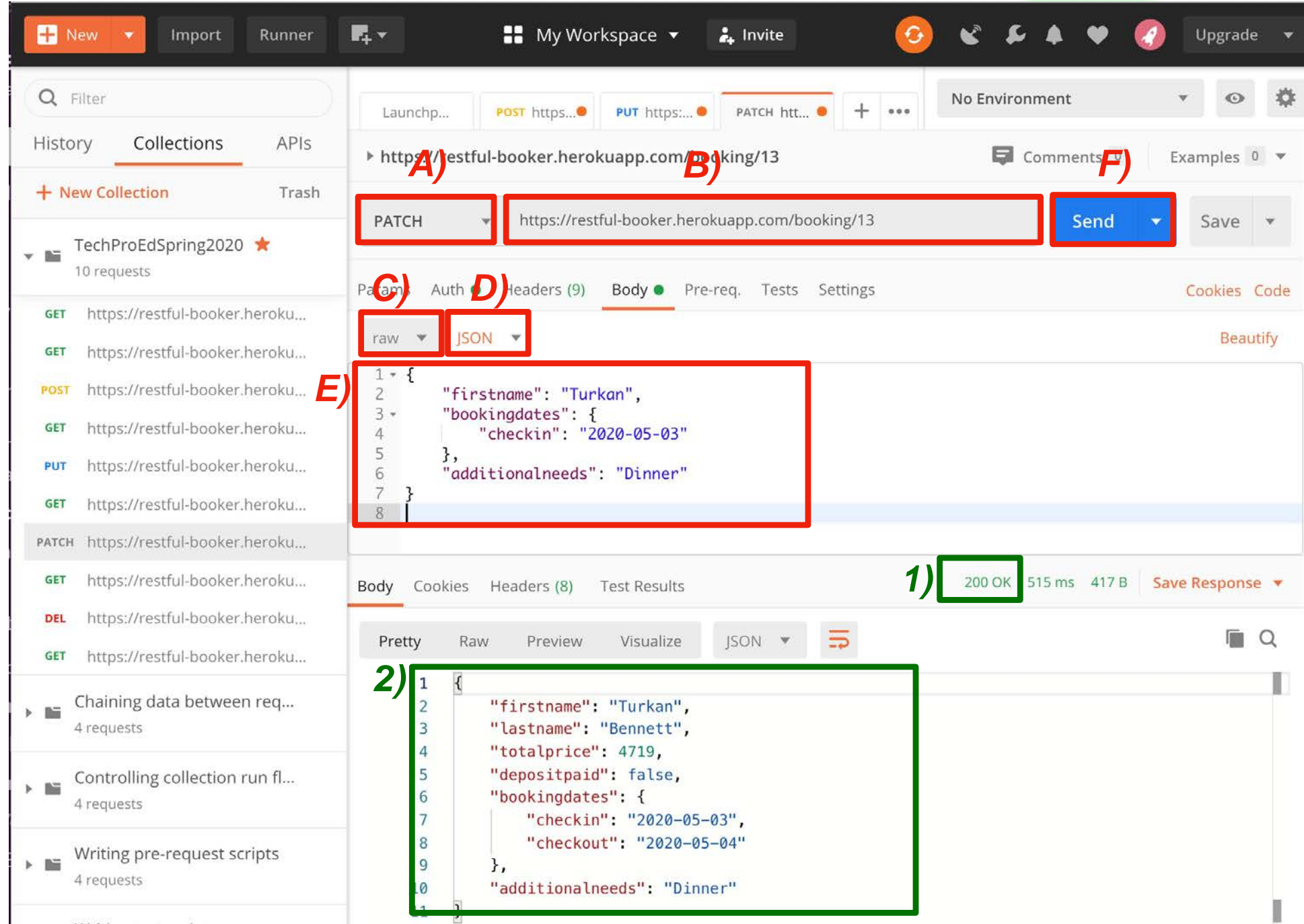
2) Data in JSON Format

API

DataBase

Note: Yanıt aldıktan sonra öncelikle **status code** kontrol ederiz.
Note: PUT Request yapabilmeniz için yetkilendirilmiş olmanız gerekmektedir.

PATCH Request Nasıl Çalışır?



A=>B=>C=>D=>E=>F PATCH Request

1) Status Code

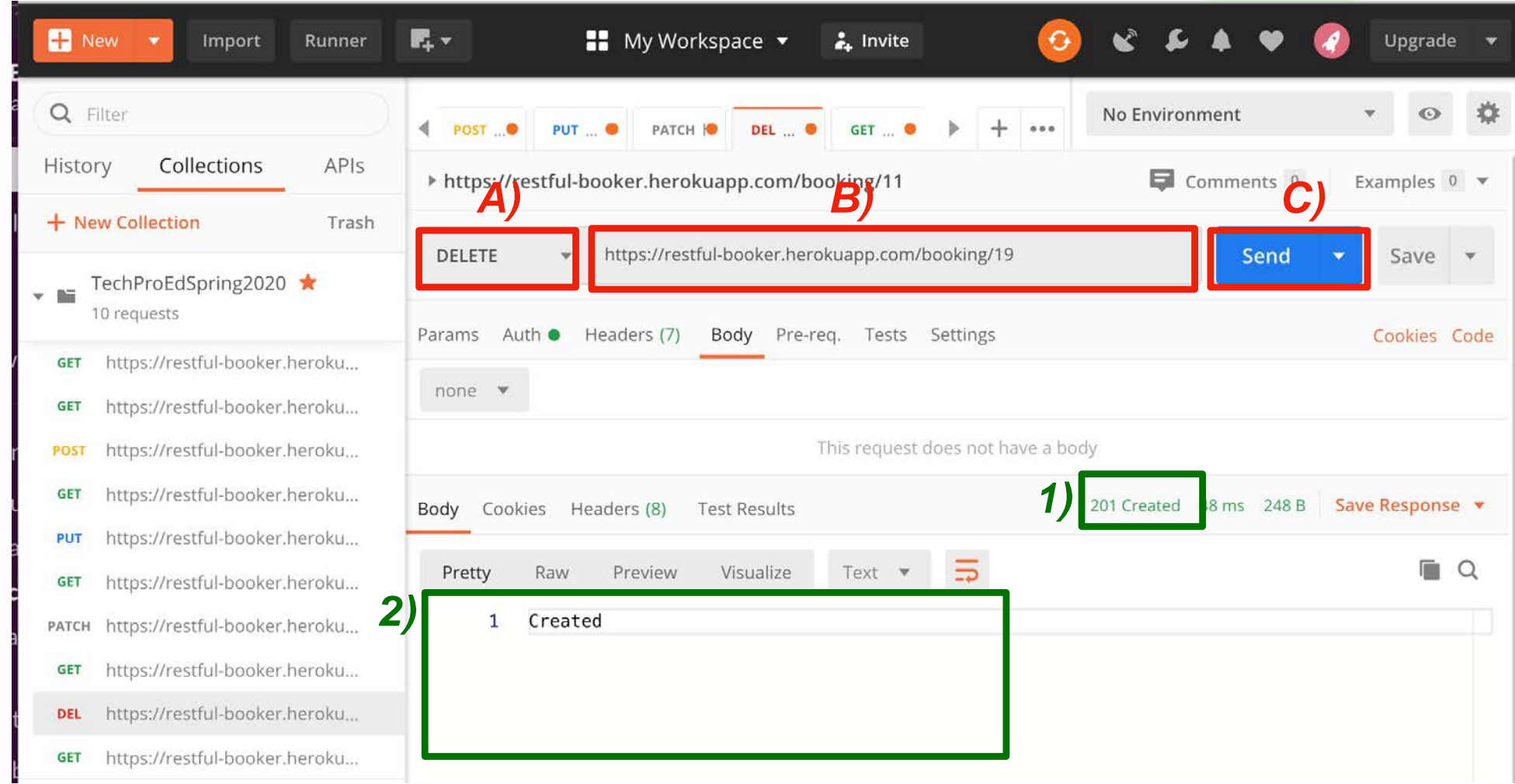
2) Data in JSON Format

API

DataBase

Note: Yanıt aldıktan sonra öncelikle **status code** kontrol ederiz.
Note: PATCH Request yapabilmeniz için yetkilendirilmiş olmanız gerekmektedir.

DELETE Request Nasıl Çalışır?



A=>B=>C DELETE Request

1) Status Code

2) Message

API

DataBase

Note: Yanıt aldıktan sonra öncelikle **status code** kontrol ederiz.

Note: DELETE Request yapabilmeniz için yetkilendirilmiş olmanız gerekmektedir.

Postman'de Testler Nasıl Yapılır ?

The screenshot shows the Postman interface with a REST client request. The request is a GET method to the URL `https://jsonplaceholder.typicode.com/users/10`. The **Tests** tab is selected, and a test script is written in the editor:

```
1 pm.test("Status code is 200", function () {
2   pm.response.to.have.status(200);
3 });
4
```

The **Send** button is highlighted with a red box, and the **Save** button is highlighted with a green box. The **Tests** tab is highlighted with a red box. The **Test Results** tab is selected, and the status is **200 OK**. The response body is shown in the **Body** tab, displaying a JSON object:

```
1 {
2   "id": 10,
3   "name": "Clementina DuBuque",
4   "username": "Moriah.Stanton",
5   "email": "Rey.Padberg@karina.biz",
6   "address": {
7     "street": "Kattie Turnpike",
8     "suite": "Suite 198",
9     "city": "Lebsackbury",
10    "zipcode": "31428-2261",
11    "geo": {
12      "lat": "-38.2386",
13      "lng": "57.2232"
14    }
15  }
16 }
```

Annotations and steps are provided for each part of the interface:

- 1) Tests' e tıklanır**: Points to the **Tests** tab.
- 2) Seçim**: Points to the **Status code: Code is 200** option in the **Test Results** tab.
- 3) Güncellemeye ihtiyacınız varsa**: Points to the **Test Results** tab.
- 4) Gönder**: Points to the **Send** button.
- 5) İsterseniz kayıt edebilirsiniz**: Points to the **Save** button.

API Test Case'leri Nasıl Yapılır ?

Aşağıdaki testleri **Postman Requestlerini** kullanarak yapalım (**Manually**)

1) **When** REST API ye **GET** sorgusu gönderdiğimizde

URL <https://restful-booker.herokuapp.com/booking>

And Accept tipi "application/JSON" olmalıdır

Then HTTP Status Code 200 olmalıdır

And Response formatı "application/JSON"

2) **When** REST API ye **GET** sorgusu gönderdiğimizde

URL <https://restful-booker.herokuapp.com/booking/5>

And Accept formatı "application/JSON" olmalıdır

Then HTTP Status Code 200 olmalıdır

And first name "Mary" olmalıdır

And total price 814 olmalıdır

3) **When** REST API ye **GET** sorgusu gönderdiğimizde

URL <http://dummy.restapiexample.com/api/v1/employees>

And Accept formatı "application/JSON" olmalıdır

Then HTTP Status Code 200 olmalıdır

And "Gerret Winters" data da görünmelidir.

4) **When** REST API ye **GET** sorgusu gönderdiğimizde

URL <http://dummy.restapiexample.com/api/v1/employee/1>

And Accept formatı "application/JSON" olmalıdır

Then HTTP Status Code 200 olmalıdır

And employee name "Tiger Nixon" olmalıdır

And employee salary "320800" olmalıdır

And employee age "61" olmalıdır