**Soru 3 : Test Otomasyonu – API**

Aşağıdaki linki kullanarak otel rezervasyon için postman kullanarak api testlerini gerçekleştiriniz. http://restful-booker.herokuapp.com/apidoc/index.html

* Dokümantasyondaki tüm çağrıları gerçekleştiriniz.
* Her çağrı için en az 1 adet test yazınız.

**Postman**

Collections – New Collection – Restful Booker API Test

1. **Auth – CreateToken**

“New Request” – Method “POST” - URL “https://restful-booker.herokuapp.com/auth” - Send’e bas

**Body sekmesinde**, raw ve JSON formatını seç ve aşağıdakini yaz;

{

"username" : "admin",

"password" : "password123"

}

**Header sekmesinde,** API'ye JSON formatında veri gönderdiğimizi belirtmemiz gerekir ;

Key: Content-Type

Value: application/json

**Test sekmesinde ;** Status Code Kontrolü : İstek başarılı olmalı ve HTTP status kodu 200 OK olmalıdır.

pm.test("Status code is 200", function () {

pm.response.to.have.status(200);

});

**Test sekmesinde ;** Response'da token Bulunduğunu Kontrol Etme : Yanıtın içinde **token** alanının olduğunu kontrol etmelisiniz.

pm.test("Response contains token", function () {

pm.response.to.have.jsonBody("token");

});

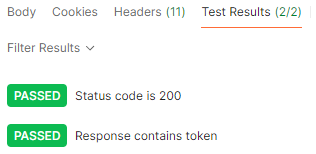
**Test Results sekmesinde** ; **Body** kısmında aşağıdaki gibi bir yanıt almalıyız ;

{

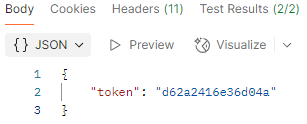
"token": "abc123"

}

**Test Results**



**Body**



**Token :** d62a2416e36d04a

1. **Booking – GetBookingIds**

“New Request” – Method “GET” - URL “http://restful-booker.herokuapp.com/booking” - Send’e bas

**Test sekmesinde ;** Status Code Kontrolü : İstek başarılı olmalı ve HTTP status kodu 200 OK olmalıdır.

**Test sekmesinde ;** Yanıtın bir dizi (array) olup olmadığını kontrol etmek;

**Test sekmesinde ;** Response'da Booking ID'lerinin Olup Olmadığını Kontrol Etmek;

pm.test("Status code is 200", function () {

pm.response.to.have.status(200);

});

pm.test("Response is an array", function () {

var responseJson = pm.response.json();

pm.expect(responseJson).to.be.an("array");

});

pm.test("Response contains bookingid", function () {

var responseJson = pm.response.json();

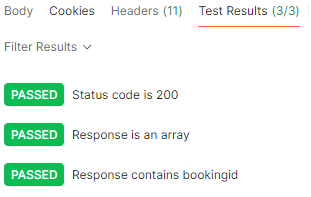
responseJson.forEach(function(booking) {

pm.expect(booking).to.have.property("bookingid");

});

});

**Test Results**



**Body**



1. **Booking – GetBooking**

“New Request” – Method “GET” – URL “https://restful-booker.herokuapp.com/booking/1” - Send’e bas

Örnek olarak, id'si 1 olan bir rezervasyonu almak için şu URL'yi kullanabilirsiniz:

* URL: https://restful-booker.herokuapp.com/booking/1

Buradaki :id kısmı, sorgulamak istediğiniz booking id ile değiştirilmelidir.

**Header sekmesinde** :

Key kısmına Accept yazın.

Value kısmına application/json yazın.

**Test sekmesinde ;**

* Status code 200 kontrolü.
* Yanıtın içinde gerekli alanların (firstname, lastname, totalprice, depositpaid, bookingdates) olup olmadığını kontrol etme.
* bookingdates objesinin içindeki checkin ve checkout tarihlerini kontrol etme.

// Status code'un 200 olup olmadığını kontrol et

pm.test("Status code is 200", function () {

pm.response.to.have.status(200);

});

// Yanıtın JSON formatında olup olmadığını kontrol et

pm.test("Response is in JSON format", function () {

pm.response.to.be.json;

});

// Yanıtın içinde gerekli alanların olup olmadığını kontrol et

pm.test("Response contains required fields", function () {

var jsonResponse = pm.response.json();

pm.expect(jsonResponse).to.have.property('firstname');

pm.expect(jsonResponse).to.have.property('lastname');

pm.expect(jsonResponse).to.have.property('totalprice');

pm.expect(jsonResponse).to.have.property('depositpaid');

pm.expect(jsonResponse).to.have.property('bookingdates');

});

// Booking dates objesinin içinde checkin ve checkout olup olmadığını kontrol et

pm.test("Booking dates are correct", function () {

var jsonResponse = pm.response.json();

pm.expect(jsonResponse.bookingdates).to.have.property('checkin');

pm.expect(jsonResponse.bookingdates).to.have.property('checkout');

});

// depositpaid değerinin boolean olup olmadığını kontrol et

pm.test("Deposit paid is boolean", function () {

var jsonResponse = pm.response.json();

pm.expect(jsonResponse.depositpaid).to.be.a('boolean');

});

// Total price'ın sayı olup olmadığını kontrol et

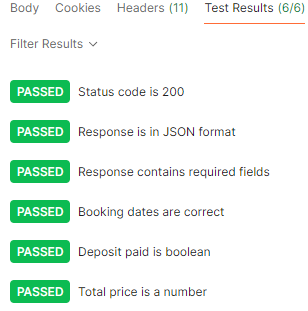
pm.test("Total price is a number", function () {

var jsonResponse = pm.response.json();

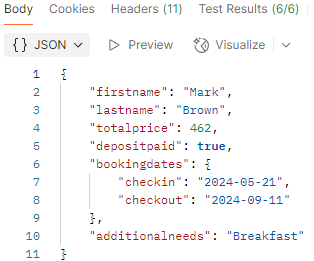
pm.expect(jsonResponse.totalprice).to.be.a('number');

});

**Test Results**



**Body**



1. **Booking – CreateBooking API**

“New Request” – Method – “POST” - URL “https://restful-booker.herokuapp.com/booking” - Send’e bas

**Header sekmesinde** :

ContentType – application/json

Accept - application/json

**Body sekmesinde**, raw ve JSON formatını seç ve aşağıdakini yaz;

{

"firstname": "Jim",

"lastname": "Brown",

"totalprice": 111,

"depositpaid": true,

"bookingdates": {

"checkin": "2018-01-01",

"checkout": "2019-01-01"

},

"additionalneeds": "Breakfast"

}

**Test sekmesinde ;**

* Status code 200 mü?
* Yanıt içinde bookingid ve booking objesi var mı?
* Rezervasyon detayları doğru mu?
* Booking ID bir sayı mı?
* Yanıt süresi 2000ms altında mı?

// API yanıtının 200 olup olmadığını kontrol et

pm.test("Status code is 200", function () {

pm.response.to.have.status(200);

});

// JSON formatında yanıt döndüğünü doğrula

pm.test("Response is valid JSON", function () {

pm.response.to.be.json;

});

// Yanıtı JSON olarak al

var jsonResponse = pm.response.json();

// Eğer yanıt bir listeyse (Tüm rezervasyonların ID'leri dönmüşse)

if (Array.isArray(jsonResponse)) {

pm.test("Response is an array of objects with bookingid", function () {

pm.expect(jsonResponse.length).to.be.above(0); // Liste boş değil

jsonResponse.forEach(booking => {

pm.expect(booking).to.have.property('bookingid'); // Her nesnede bookingid olmalı

pm.expect(booking.bookingid).to.be.a("number"); // bookingid sayısal olmalı

});

});

}

// Eğer yanıt tek bir rezervasyonun detaylarıysa

if (jsonResponse.hasOwnProperty("bookingid")) {

pm.test("Response has bookingid and booking object", function () {

pm.expect(jsonResponse).to.have.property('bookingid');

pm.expect(jsonResponse).to.have.property('booking');

});

pm.test("Booking details are correct", function () {

pm.expect(jsonResponse.booking.firstname).to.eql("Jim");

pm.expect(jsonResponse.booking.lastname).to.eql("Brown");

pm.expect(jsonResponse.booking.totalprice).to.eql(111);

pm.expect(jsonResponse.booking.depositpaid).to.eql(true);

pm.expect(jsonResponse.booking.bookingdates.checkin).to.eql("2018-01-01");

pm.expect(jsonResponse.booking.bookingdates.checkout).to.eql("2019-01-01");

pm.expect(jsonResponse.booking.additionalneeds).to.eql("Breakfast");

});

pm.test("Booking ID is a number", function () {

pm.expect(jsonResponse.bookingid).to.be.a("number");

});

}

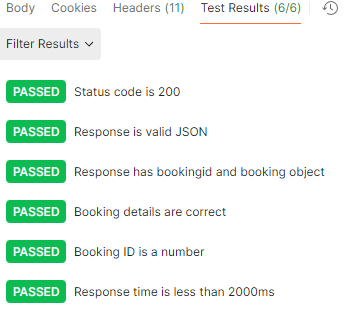
// Yanıt süresi testi

pm.test("Response time is less than 2000ms", function () {

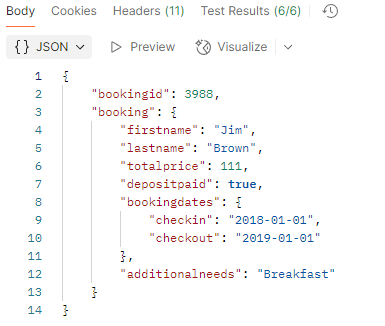
pm.expect(pm.response.responseTime).to.be.below(2000);

});

**Test Results**



**Body**



1. **Booking – CreateBooking API**

“New Request” – Method – “PUT” - URL “https://restful-booker.herokuapp.com/booking/1” - Send’e bas

**Header sekmesinde** **:**

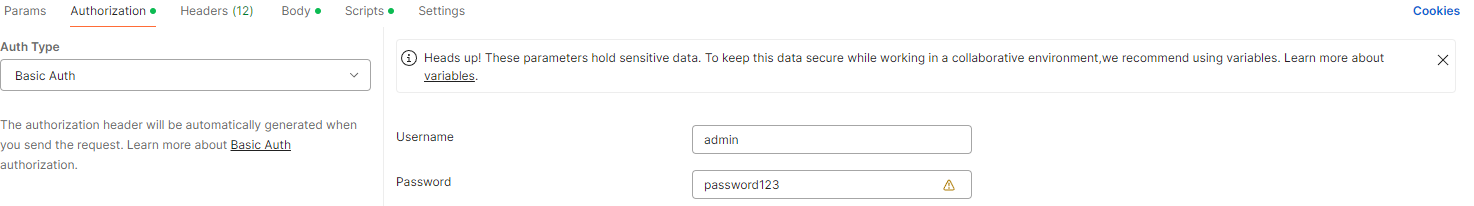
ContentType – application/json

Accept - application/json

Cookie – token = d62a2416e36d04a

Token’ı ilk testteki sonuçtan çektik.

**Authorization sekmesinde ;**



**Body sekmesinde**, raw ve JSON formatını seç ve aşağıdakini yaz;

{

"firstname": "James",

"lastname": "Brown",

"totalprice": 111,

"depositpaid": true,

"bookingdates": {

"checkin": "2018-01-01",

"checkout": "2019-01-01"

},

"additionalneeds": "Breakfast"

}

**Test sekmesinde ;**

**Status Code** → Yanıtın **200 OK** olup olmadığını doğrular.

**Yanıt Formatı** → Yanıtın **JSON** formatında olup olmadığını test eder.  
**Veri Doğrulama** → Yanıtın beklenen güncellenmiş bilgileri içerdiğini test eder.  
**Yanıt Süresi** → API’nin **2 saniyeden kısa sürede yanıt verip vermediğini** kontrol eder.  
**Header Doğrulama** → Yanıtın Content-Type header'ını test eder.  
**Boş Değer Kontrolü** → Yanıt içinde **boş veya eksik veri olup olmadığını** kontrol eder.

// 1. Status Code 200 mü?

pm.test("Status code is 200", function () {

pm.response.to.have.status(200);

});

// 2. Yanıt JSON formatında mı?

pm.test("Response is in JSON format", function () {

pm.response.to.be.json;

});

// 3. Güncellenen bilgileri doğrula

pm.test("Response has updated booking details", function () {

var jsonResponse = pm.response.json();

pm.expect(jsonResponse).to.have.property("firstname", "James");

pm.expect(jsonResponse).to.have.property("lastname", "Brown");

pm.expect(jsonResponse).to.have.property("totalprice", 111);

pm.expect(jsonResponse).to.have.property("depositpaid", true);

pm.expect(jsonResponse).to.have.property("additionalneeds", "Breakfast");

pm.expect(jsonResponse).to.have.property("bookingdates");

pm.expect(jsonResponse.bookingdates).to.have.property("checkin", "2018-01-01");

pm.expect(jsonResponse.bookingdates).to.have.property("checkout", "2019-01-01");

});

// 4. Yanıt süresi (Response Time) 2 saniyeden kısa mı?

pm.test("Response time is less than 2000ms", function () {

pm.expect(pm.response.responseTime).to.be.below(2000);

});

// 5. Header içinde Content-Type application/json olarak mı dönüyor?

pm.test("Content-Type is application/json", function () {

pm.response.to.have.header("Content-Type", "application/json; charset=utf-8");

});

// 6. Yanıt içinde boş değer var mı? (Negatif test)

pm.test("No empty fields in response", function () {

var jsonResponse = pm.response.json();

for (var key in jsonResponse) {

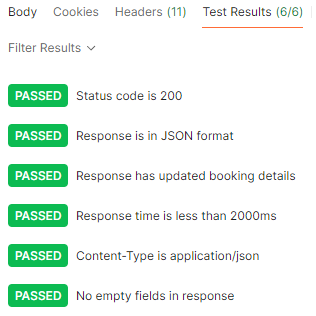
pm.expect(jsonResponse[key]).to.not.be.null;

pm.expect(jsonResponse[key]).to.not.be.undefined;

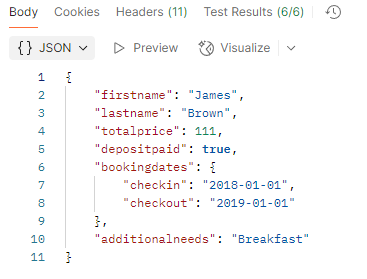
}

});

**Test Results**



**Body**



1. **Booking – PartialUpdateBooking**

“New Request” – Method – “PATCH” - URL “https://restful-booker.herokuapp.com/booking/1” - Send’e bas

**Header sekmesinde** **:**

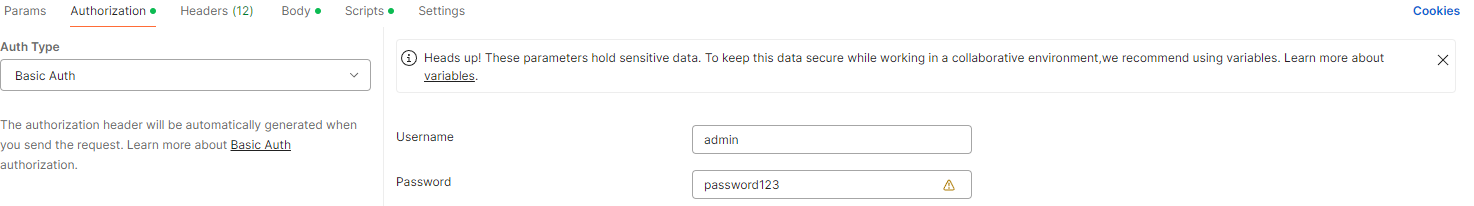
ContentType – application/json

Accept - application/json

Cookie – token = d62a2416e36d04a

Token’ı ilk testteki sonuçtan çektik.

**Authorization sekmesinde ;**



**Body sekmesinde**, raw ve JSON formatını seç ve aşağıdakini yaz;

{

"firstname": "James",

"lastname": "Brown"

}

**Test sekmesinde ;**

**Status Code 200 mü?**  
**Yanıt JSON formatında mı?**  
**Güncellenen veriler doğrulandı mı?**  
**Total price bir sayı mı?**  
**Deposit paid bir boolean mı?**  
**Booking dates nesne mi?**  
**Check-in ve Check-out tarih formatları doğru mu?**  
**Ekstra ihtiyaçlar (additionalneeds) metin mi?**

// 1️

Status Code Testi

pm.test("Status code is 200", function () {

pm.response.to.have.status(200);

});

// 2️

JSON Format Kontrolü (Daha Esnek)

pm.test("Response is in JSON format", function () {

pm.response.to.have.header("Content-Type");

pm.expect(pm.response.headers.get("Content-Type")).to.include("application/json");

});

// 3️

Güncellenmiş Rezervasyon Bilgileri Kontrolü

pm.test("Response contains updated booking details", function () {

var jsonData = pm.response.json();

pm.expect(jsonData).to.have.property("firstname");

pm.expect(jsonData).to.have.property("lastname");

pm.expect(jsonData).to.have.property("totalprice");

pm.expect(jsonData).to.have.property("depositpaid");

pm.expect(jsonData).to.have.property("bookingdates");

pm.expect(jsonData.bookingdates).to.have.property("checkin");

pm.expect(jsonData.bookingdates).to.have.property("checkout");

});

// 4️

Fiyatın Sayı Olup Olmadığını Kontrol Et

pm.test("Total price is a number", function () {

var jsonData = pm.response.json();

pm.expect(jsonData.totalprice).to.be.a("number");

});

// 5️

Depozito Ödeme Durumu Boolean mı?

pm.test("Deposit paid is a boolean", function () {

var jsonData = pm.response.json();

pm.expect(jsonData.depositpaid).to.be.a("boolean");

});

// 6️

Rezervasyon Tarihleri Nesne mi?

pm.test("Booking dates is an object", function () {

var jsonData = pm.response.json();

pm.expect(jsonData.bookingdates).to.be.an("object");

});

// 7️ Check-in Tarih Formatı Doğru mu?

pm.test("Check-in date format is correct", function () {

var jsonData = pm.response.json();

pm.expect(jsonData.bookingdates.checkin).to.match(/^\d{4}-\d{2}-\d{2}$/);

});

// 8️ Check-out Tarih Formatı Doğru mu?

pm.test("Check-out date format is correct", function () {

var jsonData = pm.response.json();

pm.expect(jsonData.bookingdates.checkout).to.match(/^\d{4}-\d{2}-\d{2}$/);

});

// 9️ Ekstra İhtiyaçlar Metin mi?

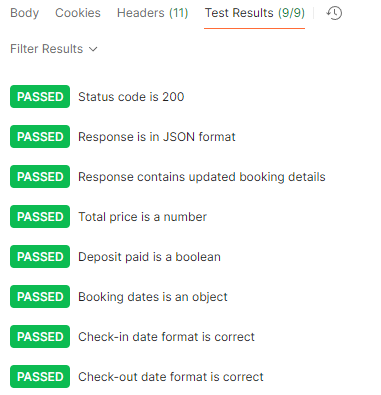
pm.test("Additional needs is a string", function () {

var jsonData = pm.response.json();

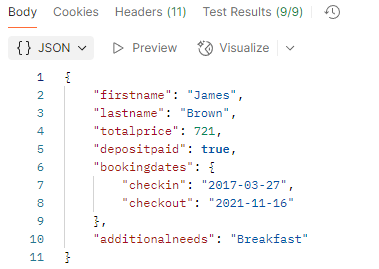
pm.expect(jsonData.additionalneeds).to.be.a("string");

});

**Test Results**



**Body**



1. **Booking – DeleteBooking**

“New Request” – Method – “DELETE” - URL “https://restful-booker.herokuapp.com/booking/1” - Send’e bas

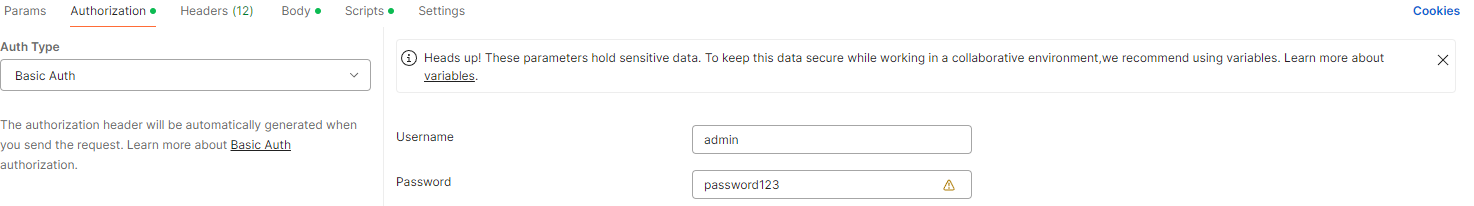
**Header sekmesinde** **:**

ContentType – application/json

Cookie – token = d62a2416e36d04a

Token’ı ilk testteki sonuçtan çektik.

**Authorization sekmesinde ;**



**Body sekmesi**, boş bırakılır.

**Test sekmesinde ;**

**Status Code 200 mü?**  
**Yanıtın boş olup olmadığının kontrolü**

// Status Code 200 veya 201 kontrolü

pm.test("Status code is 200 or 201", function () {

pm.expect(pm.response.code).to.be.oneOf([200, 201]);

});

// Yanıtın boş olup olmadığını veya "Created" mesajı içerip içermediğini kontrol etme

pm.test("Response body is empty or contains success message", function () {

pm.expect(pm.response.text()).to.satisfy(text => text === "" || text.includes("Created") || text.includes("Deleted"));

});

// Content-Type başlığının JSON veya text/plain olup olmadığını kontrol etme

pm.test("Content-Type is JSON or plain text", function () {

pm.expect(pm.response.headers.get("Content-Type")).to.match(/application\/json|text\/plain/);

});

// Yanıt süresinin 1000ms'den küçük olup olmadığını kontrol etme (Performans testi)

pm.test("Response time is less than 1000ms", function () {

pm.expect(pm.response.responseTime).to.be.below(1000);

});

// Doğru yetkilendirme (Authentication) ile çağrının yapıldığını kontrol etme

pm.test("Check if request was authorized", function () {

pm.expect(pm.response.code).to.not.equal(403);

});

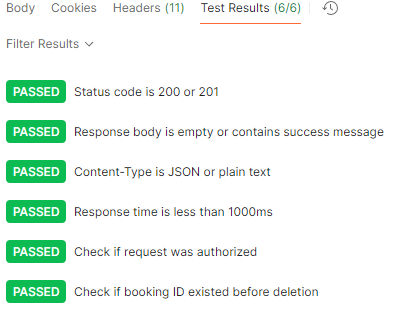
// Eğer 404 hatası alınırsa, bunun silinmiş veya mevcut olmayan bir rezervasyon ID'si nedeniyle olduğunu belirleme

pm.test("Check if booking ID existed before deletion", function () {

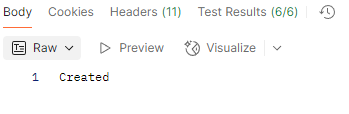
pm.expect(pm.response.code).to.not.equal(404);

});

**Test Results**



**Body**



1. **Ping – HealthCheck**

“New Request” – Method – “GET” - URL “https://restful-booker.herokuapp.com/ping” - Send’e bas

**Header sekmesinde** **:**

ContentType – application/json

**Test sekmesinde ;**

Status Code Kontrolü (201)

Yanıtın "Created" İçermesi

pm.test("Status code is 201", function () {

    pm.response.to.have.status(201);

});

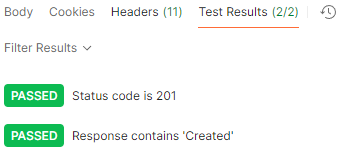
pm.test("Response contains 'Created'", function () {

    const responseBody = pm.response.text(); // Yanıtı ham metin olarak al

    pm.expect(responseBody).to.include("Created"); // İçeriğin 'Created' olduğunu kontrol et

});

**Test Results**



**Body**

