

**TECHNO**  
**STUDY** +

**REST ASSURED**

## REST ASSURED NEDİR?

- **REST** (**R**epresentational **S**tate **T**ransfer = Temsili Durum Transferi) **Assured**'ı bir dil olarak düşünebiliriz çünkü kullanıcılara RESTful API'leri test etmek için bir yol sağlar. Ancak, REST Assured'ı bir kütüphane olarak da düşünebiliriz çünkü kullanıcılara RESTful API'leri test etmek için araçlar sağlar.
- REST Assured, RESTful API'leri test etmek için kullanılan açık kaynaklı bir Java kütüphanesidir. Bu kütüphane, kullanıcıların **güçlü**, **okunabilir** ve **sürdürülebilir** testler yazmasını sağlayan bir DSL (Domain-Specific Language = Alana Özel bir Dil) sağlar.
- REST Assured, **SOAP** (**S**imple **O**bject **A**ccess **P**rotocol) tabanlı API'leri desteklemez.

## REST ASSURED'IN FAYDALARI

- Bir HTTP bağlantısı kurmak, bir istek göndermek ve bir yanıt almak ve ayrıştırmak için gereken çok sayıda sabit kod yazma ihtiyacını ortadan kaldırır.
- Testlerinizi, anında herkes tarafından okunabilir hale getiren Given / When / Then test notasyonunu destekler.
- REST Assured bir Java kütüphanesi olduğundan, özellikle JUnit veya TestNG gibi bir Java test çerçevesi ile birlikte kullanıldığında, onu sürekli entegrasyon / sürekli teslimat (CI/CD) kurulumuna entegre etmek çok kolaydır.

## GIVEN-WHEN-THEN PATERNI

- Bu desen, testleri daha okunabilir ve sürdürülebilir hale getirmenin etkili bir yoludur. Testlerin amacını ve neyin test edildiğini açıkça ortaya koyar. Ayrıca, testleri tekrarlanabilir ve tutarlı hale getirmeye yardımcı olur.

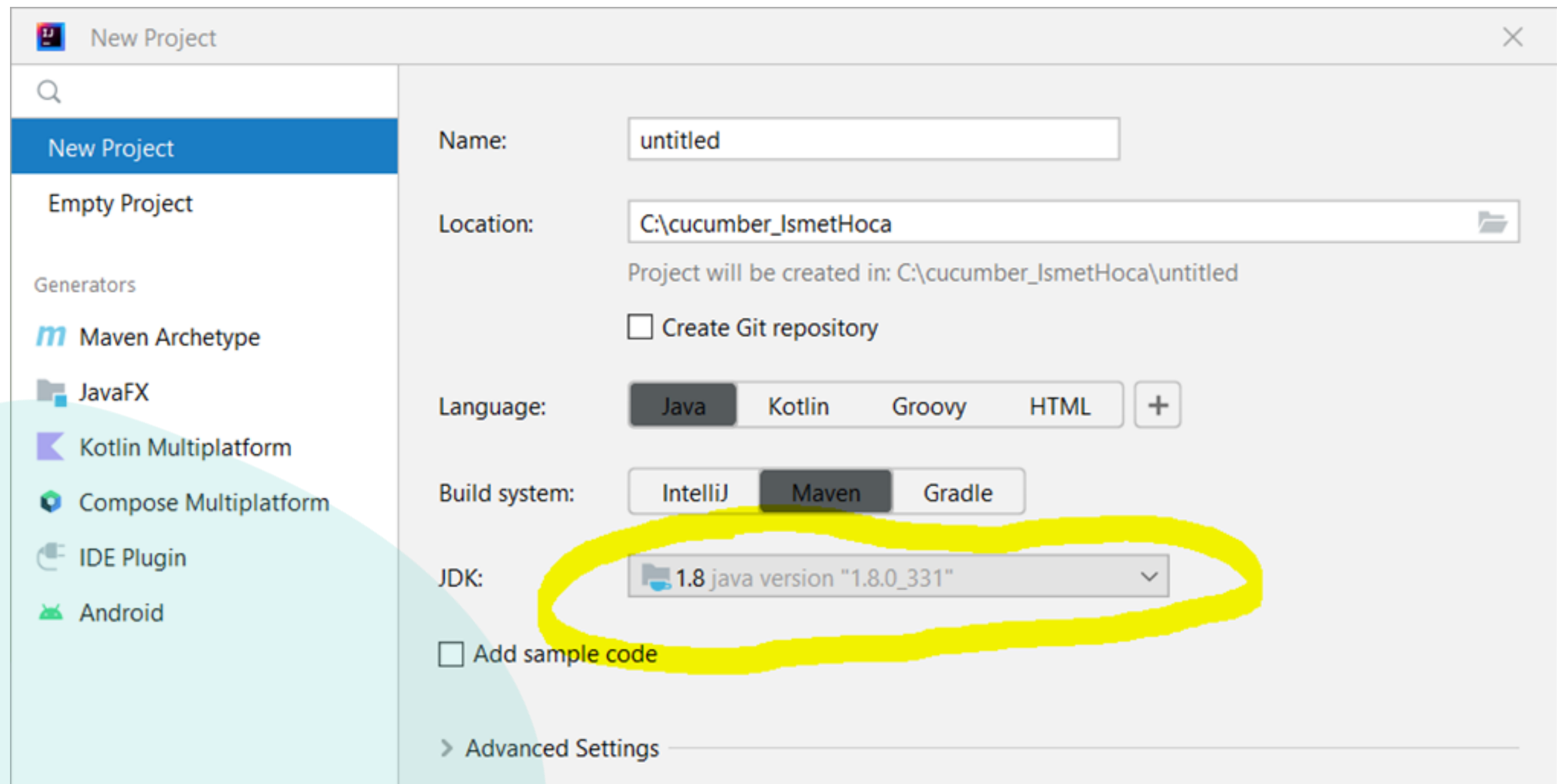
@Test

```
public void test(){  
    //Given: set up the input for the action under test (test data) . Setup: Hazırlık bölümü,  
başlangıç ayar kısmı  
    //When: execute the action you want to test. Action: linki, uygun yöntem ile çağırma  
    //Then: check the output with assertions. Test : Sonucu test etme kısmı  
}
```

## NEW PROJECT : SELECT SDK

- <https://blog.anurut.com/setting-up-restassured-with-testng-in-intellijidea>  
<https://semaphoreci.com/community/tutorials/testing-rest-endpoints-using-rest-assured#h-prerequisites>

TECHNO  
STUDY +



## POM (POM.XML) AYARLARI

- `<properties>`
- `<maven.compiler.source>1.8</maven.compiler.source>`
- `<maven.compiler.target>1.8</maven.compiler.target>`
- `<project.build.sourceEncoding>UTF-8</project.build.sourceEncoding>`
- `<argLine>-Duser.language=en</argLine>`
- `</properties>`

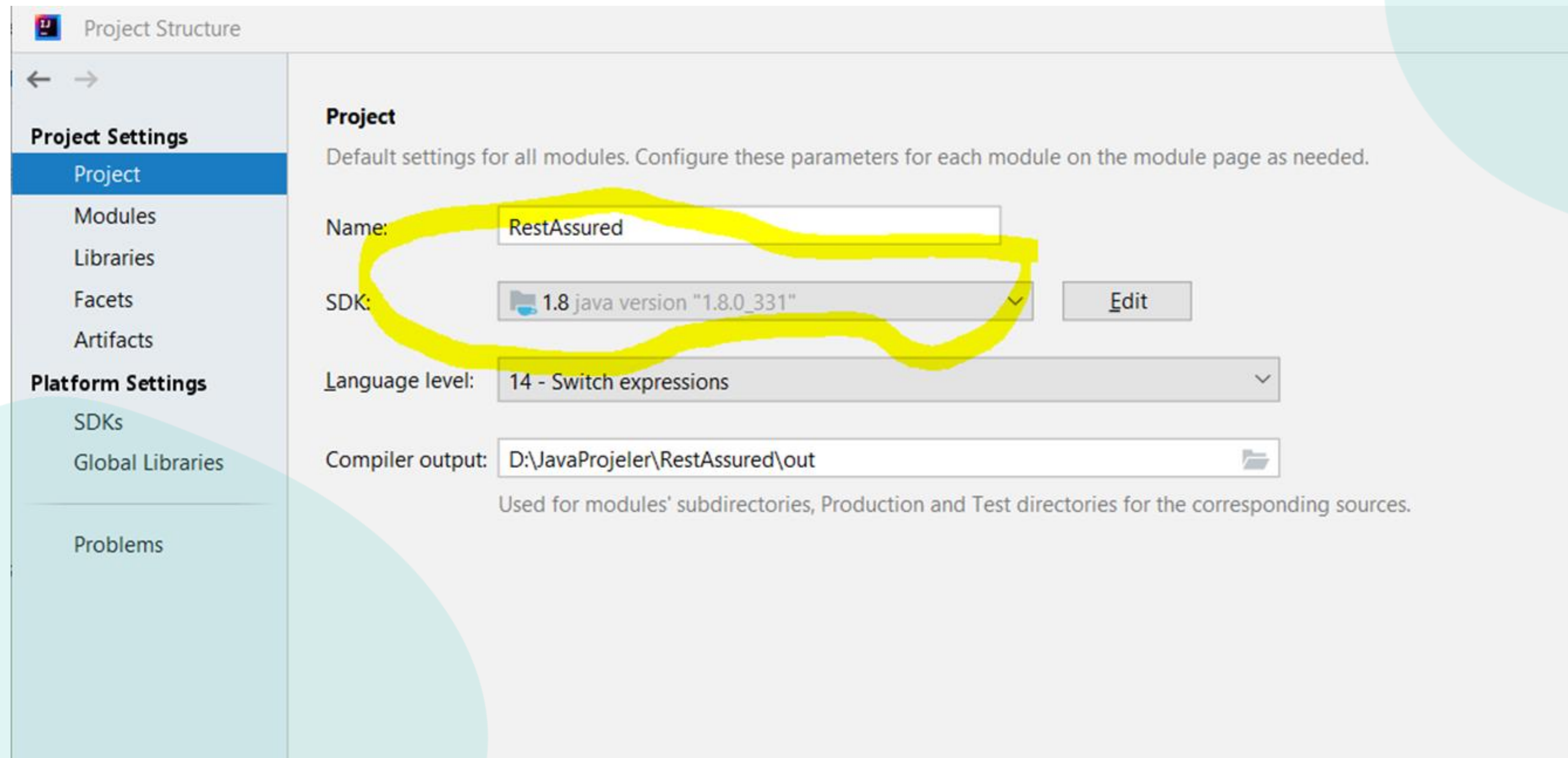
## POM (POM.XML) BAĞIMLILIKLAR

- <dependency>
  - <groupId>org.hamcrest</groupId>
  - <artifactId>**hamcrest-all**</artifactId>
  - <version>1.3</version>
  - <scope>test</scope>
  - </dependency>
- <!-- test yazmak için ->
  - <dependency>
  - <groupId>org.testng</groupId>
  - <artifactId>testng</artifactId>
  - <version>7.4.0</version>
  - </dependency>
- <dependency>
  - <groupId>io.rest-assured</groupId>
  - <artifactId>**rest-assured**</artifactId>
  - <version>5.2.0</version>
  - <scope>test</scope>
  - </dependency>




# PROJECT STRUCTURE' I KONTROL EDIN

TECHNO  
STUDY +






## YOL HARITAMIZ

- 1- Zippo site test
  - 2- gorest.co.in test burada POJO bölümü göreceğiz.
  - 3- Campus ı test edeceğiz.
- 
- There are two large, light teal abstract shapes on the slide. One is in the top right corner, containing the logo. The other is in the bottom left corner, partially cut off by the edge of the slide.

## STANDART “STATIC IMPORT” LARIMIZ:

- `import static io.restassured.RestAssured.*;`
  - `import static org.hamcrest.Matchers.*;`
- 
- In the bottom left corner, there are several overlapping teal shapes: a large light blue shape, a medium teal shape, and a small dark teal circle.

# PROJE TEST

TECHNO  
STUDY +

The image shows a screenshot of an IDE (IntelliJ IDEA) with a project named 'RestAssured' located at 'D:\JavaProjeler\RestAssured'. The project structure includes a 'test' directory with a 'java' subdirectory containing the 'ZippoTest' class. The 'pom.xml' file is also visible.

The 'ZippoTest.java' file is open, showing the following code:

```
1 import org.testng.annotations.Test;
2
3 import static io.restassured.RestAssured.*;
4 import static org.hamcrest.Matchers.*;
5
6 public class ZippoTest {
7
8     @Test
9     public void test()
10    {
11        given()
12            // hazırlık işlemlerini yapacağız
13
14        .when()
15            // link ve aksiyon işlemleri
16
17        .then()
18            // test ve extract işlemleri
19
20    }
```

The 'Run' tab shows the test results for 'ZippoTest.test'. The test passed successfully, with a duration of 774 ms. The output console shows the following message:

```
====
Default Suite
Total tests run: 1, Passes: 1, Failures: 0, Skips: 0
=====
Process finished with exit code 0
```

The 'Tests passed: 1' status is highlighted in green at the bottom left.

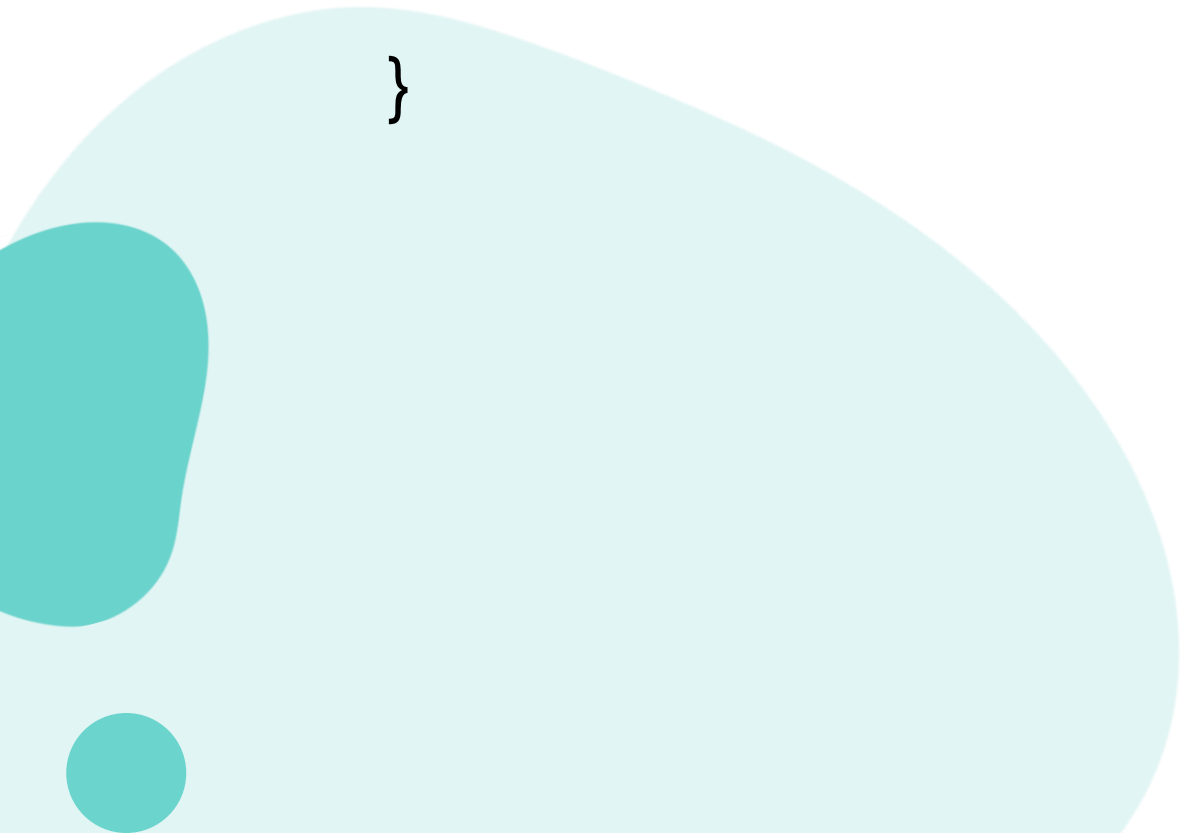
## İLK TEST SITEMİZ:

<http://api.zippopotam.us/us/90210>

**TECHNO**  
**STUDY** +

## GELEN CEVABIN (RESPONSE) DURUM KONTROLÜ:

```
@Test
public void statusCodeTest() {
    given().
    when().
    get("http://zippopotam.us/us/90210").
    then().
        statusCode(200);
}
```

In the bottom left corner, there are several overlapping teal and light blue abstract shapes, including a large light blue circle and a smaller dark teal circle.

## YANITIN (RESPONSE) İÇERİK TÜRÜNÜ KONTROL ETMEK:

```
@Test
public void contentTypeTest() {
    given().
    when().
    get("http://zippopotam.us/us/90210").
    then().
        contentType(ContentType.JSON);
}
```

## YAPILAN İSTEK (REQUEST) VE DÖNEN CEVABIN (RESPONSE) AYRINTILARINI “LOG” LAMAK:

```
@Test
public void logRequestAndResponseDetails() {
    given().
        log().all().
    when().
        get("http://zippopotam.us/us/90210").
    then().
        log().body();
}
```



## HAMCREST NEDİR?

Hamcrest eşleştiricileri, otomatize edilmiş testlerinizde gerçekleştirmek istediğiniz beklentileri, kontrolleri veya doğrulamaları okunabilir bir dilde ifade etmenizi sağlar.

Bazı örnekler:

- **equalTo(X)** - gerçek bir öge değerinin önceden belirlenmiş beklenen bir öge değerine eşit olup olmadığını kontrol etmek için kullanılır.
- **hasItem("value")** - bir öge koleksiyonunun belirli bir önceden belirlenmiş öge değerini içerip içermediğini görmek için kullanılır.
- **hasSize(3)** - bir koleksiyondaki öğelerin gerçek sayısını doğrulamak için kullanılır.
- **not(equalTo(X))** - Hamcrest kitaplığında bulunan herhangi bir eşleştiriciyi ters çevirir.

## DÖNEN CEVAP GÖVDESİNİN (BODY) KONTROLÜ 1:

```
@Test
public void checkStateInResponseBody() {
    given().
    when().
        get("http://zippopotam.us/us/90210").
    then().
        body("places[0].state",
equalTo("California"));
}
```

## DÖNEN CEVAP GÖVDESİNİN (BODY) KONTROLÜ 2:

```
@Test
public void checkListOfPlaceNamesInResponseBody() {
    given().
    when().
        get("http://zippopotam.us/us/90210").
    then().
        body("places.'place name'",
hasItem("Beverly Hills"));
}
```

## DÖNEN CEVAP GÖVDESİNİN (BODY) KONTROLÜ 3:

```
@Test
public void checkNumberOfPlaceNamesInResponseBody() {
    given().
    when().
        get("http://zippopotam.us/us/90210").
    then().
        body("places.'place name'",
hasSize(1));
}
```

## BIRDEN FAZLA DOĞRULAMAYI BİRLEŞTİRMEK (COMBINING)

```
@Test
public void combiningTest() {
    given().
    when().
        get("http://zippopotam.us/us/90210").
    then().
        statusCode(200)
        .contentType(ContentType.JSON)
        .body("places[0].state", equalTo("California"))
        .body("places.'place name'", hasItem("Beverly Hills"))
        .body("places.'place name'", hasSize(1));
}
```

## BASE URI (UNIFORM RESOURCE IDENTIFIER)

URI, internet üzerinde bir kaynağın tam yerine işaret eder. Base URI ise kaynağın tam adresinin kök dizinine işaret eder. Base URI ile URL benzer kavramlardır. Çünkü URL (Uniform Resource Locator) de internette bir kaynağın ana dizinine işaret eder. URL ana klasör yolu, URI ise tam dosya yoluna benzetilebilir. URI ile URL farklı kavramlar olsa da, çoğu zaman birbirini yerine kullanılır.

```
RestAssured.baseURI = "http://zippopotam.us";
```

## PATH PARAMETRELERİ

```
@Test
public void pathParametersTest() {
    given()
        .pathParam("countryCode", "us").pathParam("zipCode", "90210").
    when()
        .get("/{countryCode}/{zipCode}").
    then()
        .body("places[0].place name", equalTo("Beverly Hills"));
}
```



## SORGULAMA (QUERY) PARAMETRELERİ

[https://gorest.co.in/public-api/users?\\_format=json&access-token=j6XoJSutZrv-ikB-4X4\\_Zndi54\\_iqSZES-Ap](https://gorest.co.in/public-api/users?_format=json&access-token=j6XoJSutZrv-ikB-4X4_Zndi54_iqSZES-Ap)

@Test

```
public void queryParametersTest() {
    given()
        .param("_format", "json").param("access-token", "j6XoJSutZrv-ikB-4X4_Zndi54_iqSZES-Ap").
    when()
        .get("https://gorest.co.in/public-api/users").
    then()
        .body("result", not(empty()));
}
```

## İSTEKLERİMİZİ (REQUEST) TEKRAR KULLANILABİLİR YAPMAK 1 :

```
private RequestSpecification requestSpec;  
@BeforeClass  
public void createRequestSpecification() {  
    requestSpec = new RequestSpecBuilder().  
        setBaseUrl("http://zippopotam.us").  
        setAccept(ContentType.JSON).  
        build();  
}
```

## İSTEKLERİMİZİ (REQUEST) TEKRAR KULLANILABİLİR YAPMAK 2 :

```
@Test
public void statusCodeTestWithRequestSpec() {
    given()
        .spec(requestSpec).
    when()
        .get("us/90210").
    then()
        .statusCode(200);
}
```

## YANITTAKI (RESPONSE) ÖĞELERİ YENIDEN KULLANILABİLİR YAPMAK 1 :

```
private ResponseSpecification responseSpec;
@BeforeClass
public void createResponseSpecification() {
    responseSpec = new ResponseSpecBuilder().
        expectStatusCode(200).
        expectContentType(ContentType.JSON).
        build();
}
```

## YANITTAKI (RESPONSE) ÖĞELERİ YENIDEN KULLANILABİLİR YAPMAK 2 :

```
@Test
public void checkPlaceNameInResponseBodyWithResponseSpec() {
    given().
        spec(requestSpec).
    when().
        get("us/90210").
    then().
        spec(responseSpec).
        body("places[0].place name", equalTo("Beverly Hills"));
}
```

## YANIT GÖVDESİNDEN (RESPONSE BODY) İSTENİLEN DEĞERLERİN ÇIKARILMASI :

@Test

```
public void extractPlaceNameFromResponseBody() {  
    String placeName =  
        given().  
            spec(requestSpec).  
            when().  
                get("us/90210").  
            then().  
                extract().  
                    path("places[0].place name");  
    Assert.assertEquals(placeName, "Beverly Hills");  
}
```

## KAYNAKLAR :

- REST Assured home page : <http://rest-assured.io/>
- REST Assured usage guide : <https://github.com/rest-assured/rest-assured/wiki/Usage>
- <https://techbeacon.com/app-dev-testing/efficient-api-testing-how-get-started-rest-assured>
- <https://testautomationu.applitools.com/automating-your-api-tests-with-rest-assured/>