

İNCİ GIDA STOK TAKİP SİSTEMİ

Maven projesi Spring Boot teknolojisi ile oluşturularak IntelliJ IDEA 2023.2.2'e import edilerek başlandı.

Import işlemi gerçekleştikten sonra beraberinde gelen RestServislerApplication.java uzantılı classımızda main metodu içeren çalışabilir hazır durumda anotasyonlarımız da hazır gelmektedir.

m pom.xml (RestServisler) ×

```
1 <?xml version="1.0" encoding="UTF-8"?>
2 <project xmlns="http://maven.apache.org/POM/4.0.0" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
3   xsi:schemaLocation="http://maven.apache.org/POM/4.0.0 https://maven.apache.org/xsd/maven-4.0.0.xsd">
4   <modelVersion>4.0.0</modelVersion>
5   <parent>
6     <groupId>org.springframework.boot</groupId>
7     <artifactId>spring-boot-starter-parent</artifactId>
8     <version>3.2.1</version>
9     <relativePath/> <!-- lookup parent from repository -->
10  </parent>
11  <groupId>tr.edu.medipol.yazilimgelistirme</groupId>
12  <artifactId>RestServisler</artifactId>
13  <version>0.0.1-SNAPSHOT</version>
14  <name>RestServisler</name>
15  <description>Final Projesi Market Stok Takip Sistemi</description>
16  <properties>
17    <java.version>17</java.version>
18  </properties>
19  <dependencies>
20    <dependency>
21      <groupId>org.springframework.boot</groupId>
22      <artifactId>spring-boot-starter-web</artifactId>
23    </dependency>
24
25    <dependency>
26      <groupId>org.springframework.boot</groupId>
27      <artifactId>spring-boot-starter-test</artifactId>
28      <scope>test</scope>
```

RestServislerApplication.java ×

```
1 package tr.edu.medipol.yazilimgelistirme.RestServisler;
2
3 import org.springframework.boot.SpringApplication;
4 import org.springframework.boot.autoconfigure.SpringBootApplication;
5 import org.springframework.context.annotation.ComponentScan;
6
7 @SpringBootApplication
8 @ComponentScan(basePackages = "tr.edu.medipol.yazilimgelistirme.RestServisler")
9 public class RestServislerApplication {
10
11     public static void main(String[] args) {
12         SpringApplication.run(RestServislerApplication.class, args);
13     }
14 }
15
```

Rest web servislere geçmeden önce src/main/resources altında bulunan application.properties kısmında http localhostumuzu 8082 olarak değiştiriyoruz. (Bilgisayarımda oluşan hatayı düzeltemememden kaynaklıdır.)

```
application.properties x
1 server.port=8082
2 |
```

Sonraki adımda ise urunlerwebservis.java uzantılı yeni bir class oluşturduk ve burada ürün ekleme, ürün silme ve ürün listeleme olarak üç tane Rest web servisleri oluşturmaya başlıyoruz.

```
urunlerwebservis.java
```

```
1 package tr.edu.medipol.yazilimgelistirme.RestServisler;
2
3 import org.springframework.core.NativeDetector;
4 import org.springframework.web.bind.annotation.*;
5 import java.util.*;
6
```

Yukarıda yaptığımız tanımlamalar ile özellikle * kullanılarak paket alındaki tüm sınıfları import edebilmemiz için yapılan işlemidir.

```
@RestController
@RequestMapping("/Urunler")
public class urunlerwebservis {
```

@RestController ile artık oluşumu başlatmış oluyoruz. @RequestMapping yanında "Urunler" ile localhost:8082 portu üzerinde hangi URL ile sorgular yapacağımızı da belirtmiş oluruz.

```
9 usages
public record Urunler (String cesit, String adet){};

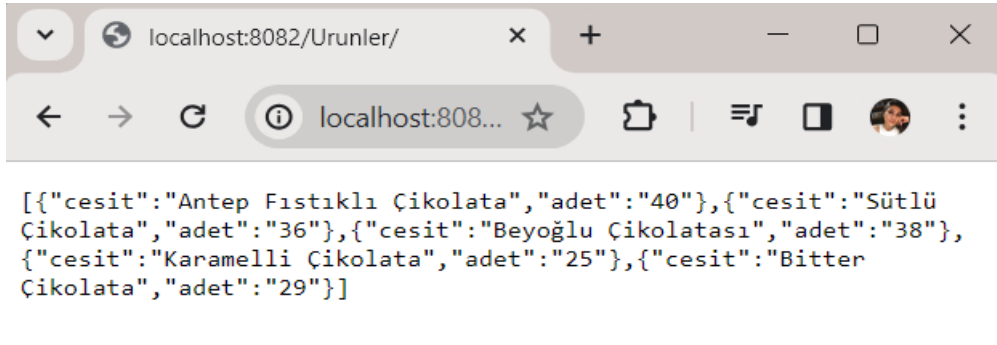
7 usages
private static final List<Urunler> URUN_LISTESI = new ArrayList<>();
static{
    URUN_LISTESI.add(new Urunler( cesit: "Antep Fıstıklı Çikolata", adet: "40"));
    URUN_LISTESI.add(new Urunler( cesit: "Sütlü Çikolata", adet: "36"));
    URUN_LISTESI.add(new Urunler( cesit: "Beyoğlu Çikolatası", adet: "38"));
    URUN_LISTESI.add(new Urunler( cesit: "Karamelli Çikolata", adet: "25"));
    URUN_LISTESI.add(new Urunler( cesit: "Bitter Çikolata", adet: "29"));
}
```

Herhangi bir veri tabanı üzerinde çalışmayacağımız için kodlama ile kendi ürün çeşit ve adetlerimizin bulunduğu bir "ürün listesi" tanımladığımıza göre artık web servislerimizi oluşturmaya başlayabiliriz.

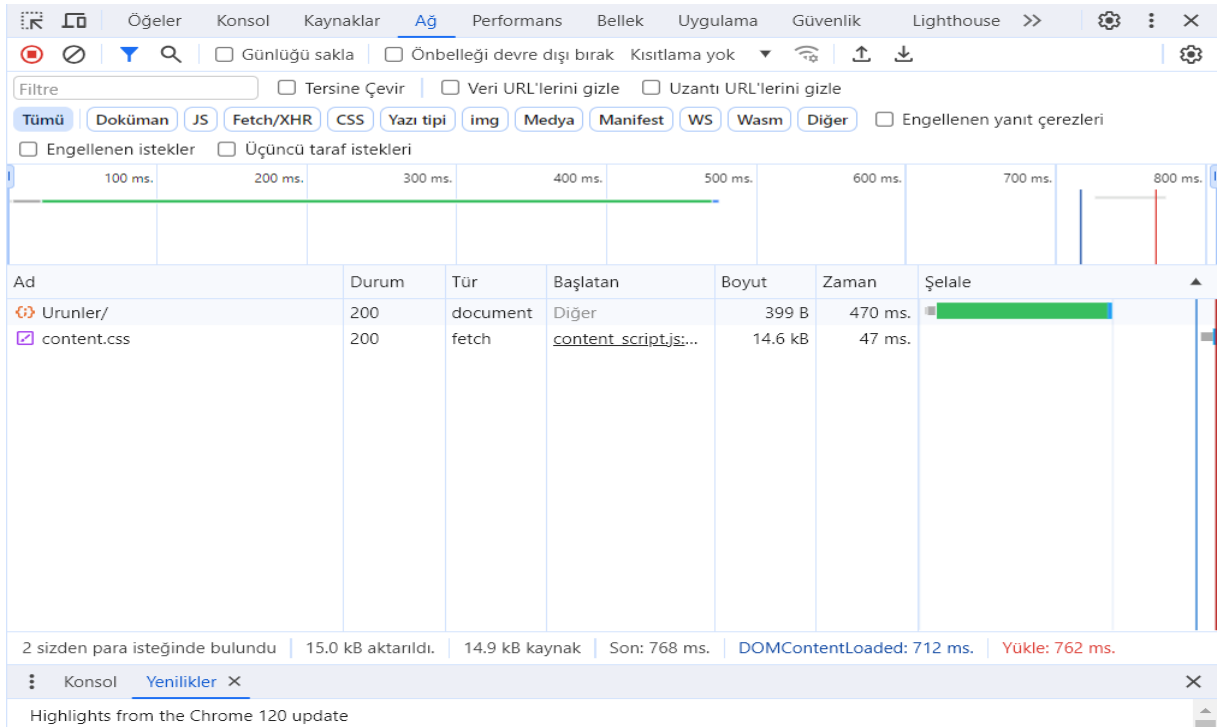
1. ÜRÜN LİSTELEME

```
@GetMapping("/")
public List<Urunler> Listele(){
    return URUN_LISTESI;
}
```

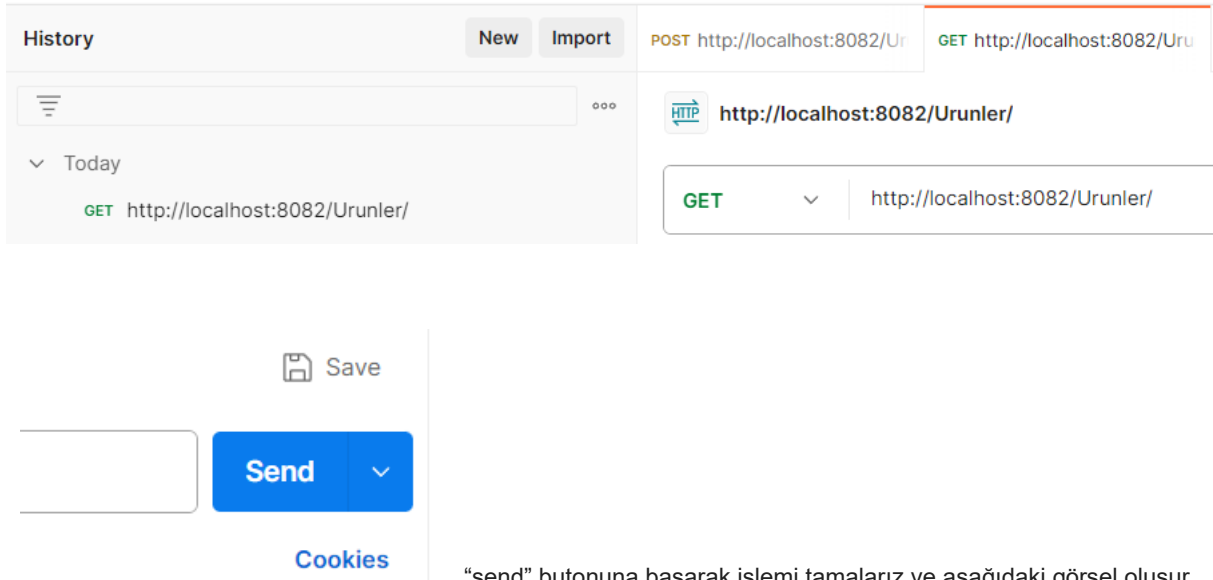
Yukarıdaki kod bloğunu çalıştırdığımızda oluşturduğumuz ürün listesini dilersek tarayıcımızın arama çubuğuna <http://localhost:8082/Urunler/> yazıp uygulamayı ayağı kaldırdığımızda sayfayı yenilediğimizde aşağıdaki gibi görebiliriz. Dağınık bir görünüm olduğundan Postmana geçiş yapacağız.



Aşağıdaki görselde ise google'ın "Developer Tools" denen aracı ile gelen ve giden istekleri daha net ve anlaşılır görebileceğimiz ekranı inceleyebilirsiniz. Ctrl+shift+I veya tarayıcınızın sağ üstü kısmındaki üç noktaya tıklayarak oradan da Diğer Araçlar/Geliştirici araçlar'ı tıklayarak da inceleyebilirsiniz.



Şimdi ise ürün listeleme rest web servis işlemimizi Postman ile nasıl gözüktüğüne bakalım. Veri tabanı kullanmadan kodlarla oluşturduğumuz ürün listemizdeki çeşitleri ve adetleri görmek için Postman adresine giderek sol taraftaki “new” butonuna tıklayıp sonrasında çıkan pencerede “http” seçiyoruz. “GET” seçeneğini seçtikten sonra <http://localhost:8082/Urunler/> linkini yapııştırıyoruz



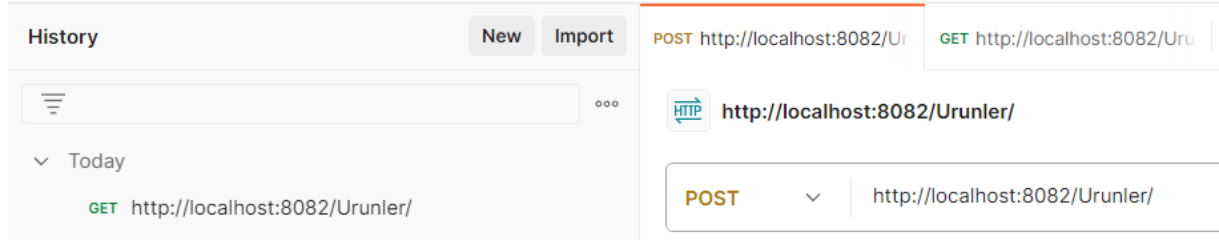
“send” butonuna basarak işlemi tamamlarız ve aşağıdaki görsel oluşur.

```
1  [
2  {
3    "cesit": "Antep Fıstıklı Çikolata",
4    "adet": "40"
5  },
6  {
7    "cesit": "Sütlü Çikolata",
8    "adet": "36"
9  },
10 {
11   "cesit": "Beyoğlu Çikolatası",
12   "adet": "38"
13 },
14 {
15   "cesit": "Karamelli Çikolata",
16   "adet": "25"
17 },
18 {
19   "cesit": "Bitter Çikolata",
20   "adet": "29"
21 },
22 ]
```

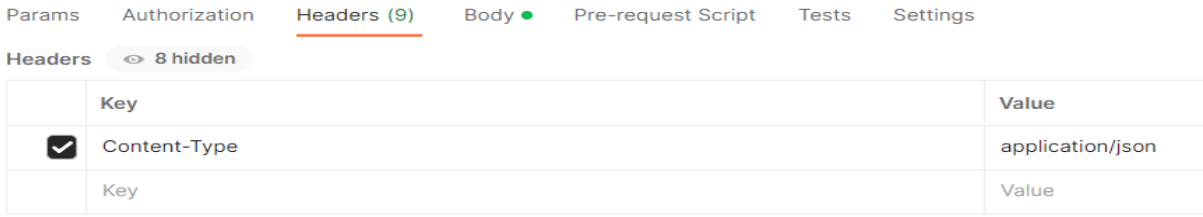
2.ÜRÜN EKLEME

```
@PostMapping("/")
public Urunler ekle(@RequestBody Urunler urunler){
    URUN_LISTESI.add(urunler);
    return urunler;
}
```

Yukarıdaki kod bloğunu çalıştırıp aynı zamanda Postman üzerinden istek attığımızda ise listeye yeni ürünün eklendiğini görmüş olacağız. Postman üzerinde isteği öncelikle yine “new” butonuna tıklayıp “http” seçeneğini seçtikten sonra “<http://localhost:8082/Urunler/>” ile birlikte “POST” seçeneği seçilir.

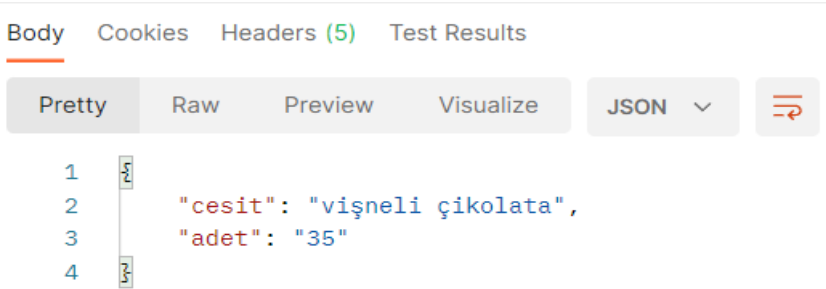


Sırasıyla “header” kısmından aşağıdaki gibi seçenekler işaretlenir ve “body” kısmında da “raw” seçeneği seçilir

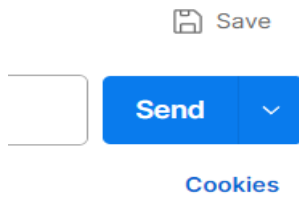


| Key | Value |
|--------------|------------------|
| Content-Type | application/json |
| Key | Value |

Ve açılan boşluktan da aşağıdaki kod satırı yazılır



Body altında “JSON” seçilmesi de önemli detaylardandır.



“send” butonuna tıklanıp istek atıldığında ise aşağıdaki gibi yeni ürün çeşidi hazır olmuş olunur.

Ancak yeni halini görmek için bir adım önce olan “Ürün Listeleme” kısmındaki gibi istek atılması gerekir. Sonuç olarak da aşağıdaki çıktı oluşur.

```
1  [
2      {
3          "cesit": "Antep Fıstıklı Çikolata",
4          "adet": "40"
5      },
6      {
7          "cesit": "Sütlü Çikolata",
8          "adet": "36"
9      },
10     {
11         "cesit": "Beyoğlu Çikolatası",
12         "adet": "38"
13     },
14     {
15         "cesit": "Karamelli Çikolata",
16         "adet": "25"
17     },
18     {
19         "cesit": "Bitter Çikolata",
20         "adet": "29"
21     },
22     {
23         "cesit": "vişneli çikolata",
24         "adet": "35"
25     }
26 ]
```

3.ÜRÜN SİLME

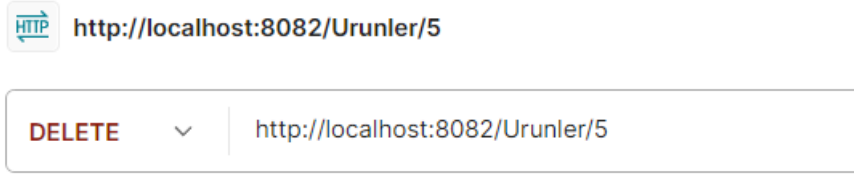
```
}

@DeleteMapping("/{id}")
public String sil(@PathVariable String id) {

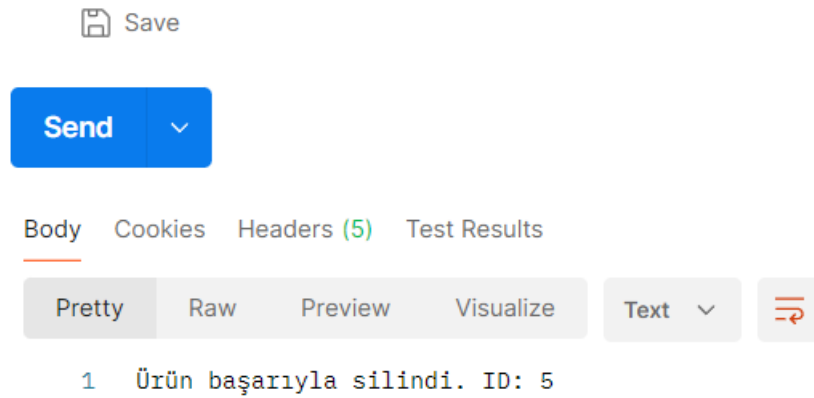
    Urunler silinecekUrun = URUN_LISTESI.stream() Stream<Urunler>
        .filter(urun -> urun.id().equals(id))
        .findFirst() Optional<Urunler>
        .orElse( other: null);

    if (silinecekUrun != null) {
        URUN_LISTESI.remove(silinecekUrun);
        return "Ürün başarıyla silindi. ID: " + id;
    } else {
        return "Ürün bulunamadı. ID: " + id;
    }
}
```

Yukarıdaki kod bulağı ile öncelikle silme anotasyonunu tanımlamış oluyoruz Rest web servisimize. Sonrasında Postman aracımıza giderek buradan gereken ayarları aşağıdaki gibi yapıyoruz.



Burada önemli olan detay @DeleteMapping anotasyonunu tanımladıktan sonra id ile işlem yapmamış bize kolaylık sağlamaktadır.



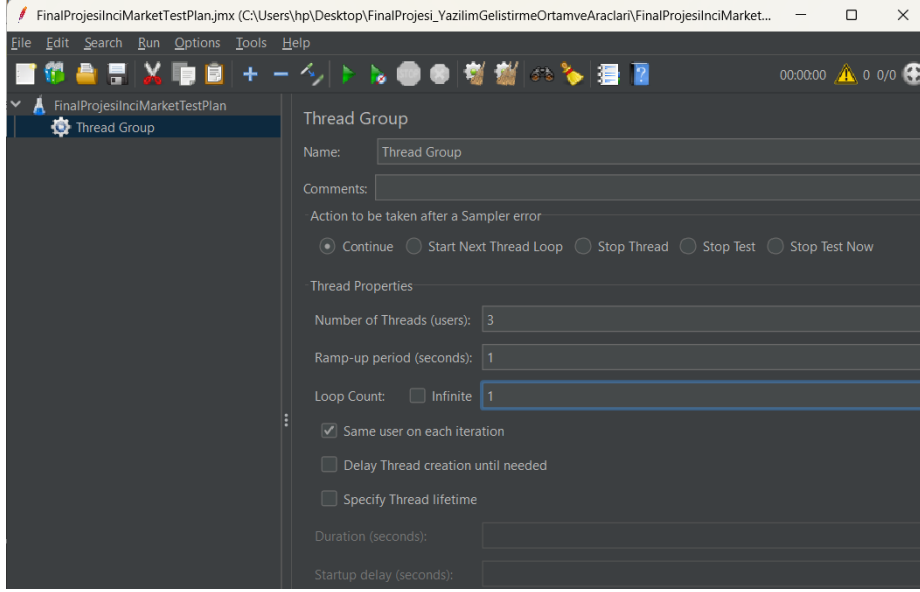
Ve yeniden dönüp "GET metodu ile ürün listeleme isteği attığımızda ise Postman üzerinden aşağıdaki görselde olduğu gibi 5 numaralı id'si olan ürünün listeden silindiğini görmüş oluruz.



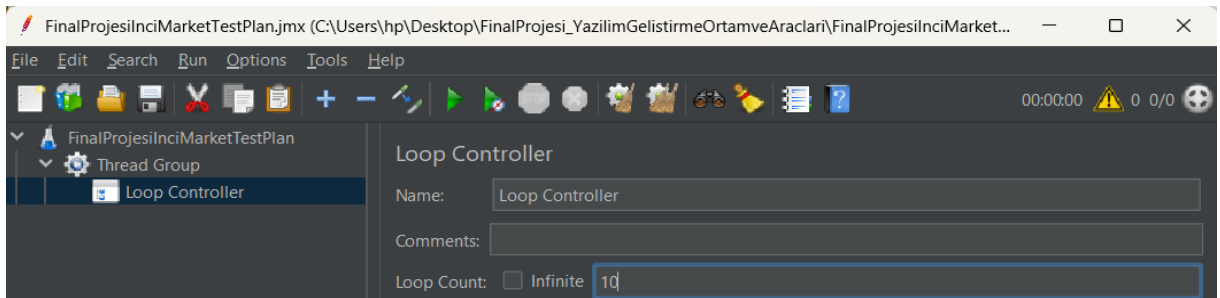
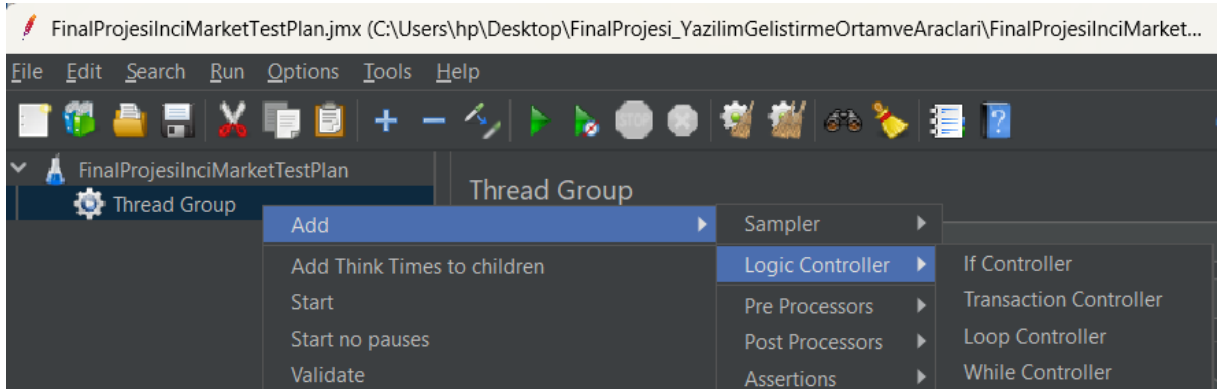
POSTMAN & JMETER TESTLERİ

Şimdi ise sıracı Market Stok takip sistemimiz için yazdığımız web servislerimizin JMeter ile test edip yine Postman üzerinden istek atarak çıktılarımızı elde edeceğimiz aşamaya gelmiş bulunuyoruz.

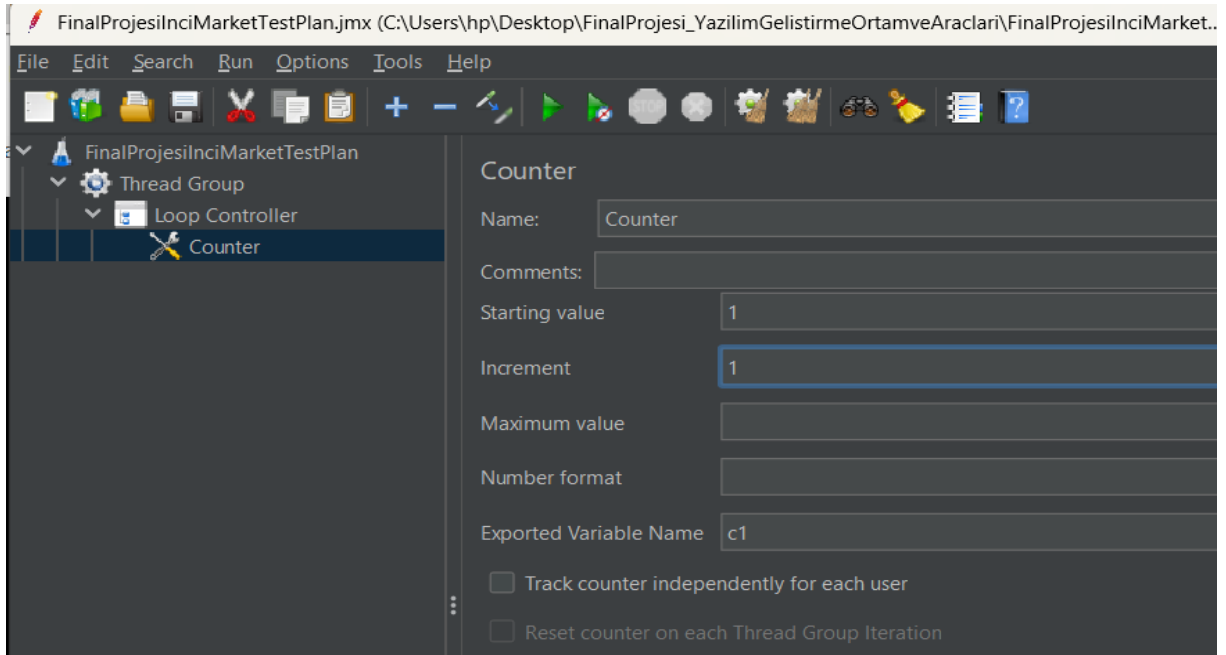
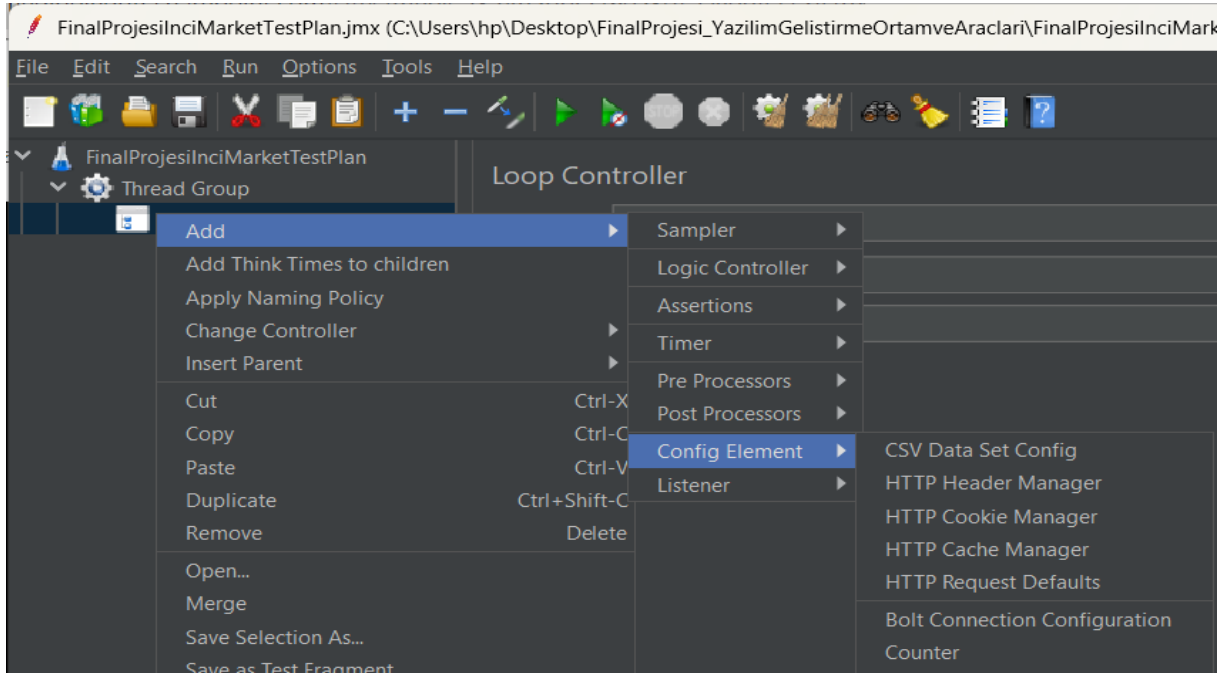
JMeter Test Planımız için öncelikle masaüstünde “FinalProjesi_YazilimGelistirmeOrtamveAraclari” adında bir klasör oluşturup ardından “FinalProjesiInciMarketTestPlan.jmx” uzantılı dosyamızı buraya kaydederek başlıyoruz. FinalProjesiInciMarketTestPlan’a sağ tıklayıp sırasıyla Add/Threads/Thread Group seçerek aşağıdaki gibi açılan kısımları uygun kullanıcı sayısı ve kaç defa döngü kurmasını istiyorsak buna göre ayarlamalar yapıyoruz. (Final projesi olduğundan mütevellit çok fazla kullanıcı ve döngü girmeden, bilgisayarımızın kaldırıp kaldırmayacağından emin olunmadığından risk almaktan kaçınacak düzeyde veriler girilmiştir.)



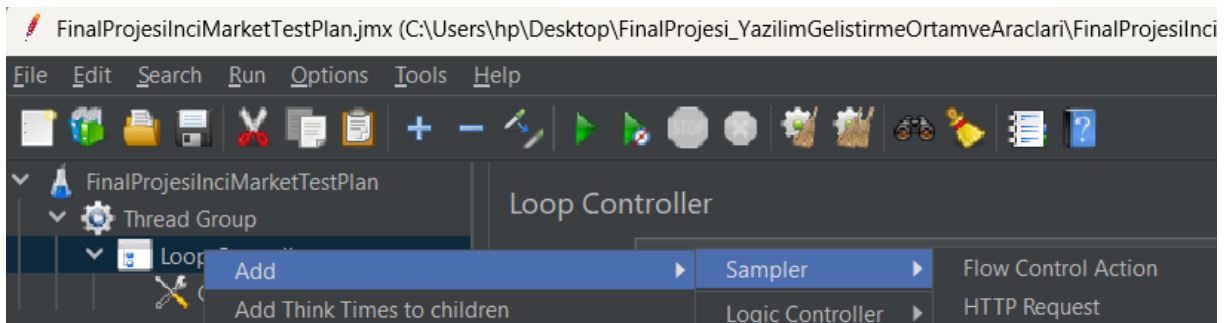
Ardından yine sol taraftaki sütunda “Thread Group”a sağ tıklayıp sırasıyla add/LogicController/LoopController tıklayarak döngü sayımızı belirtiyoruz.



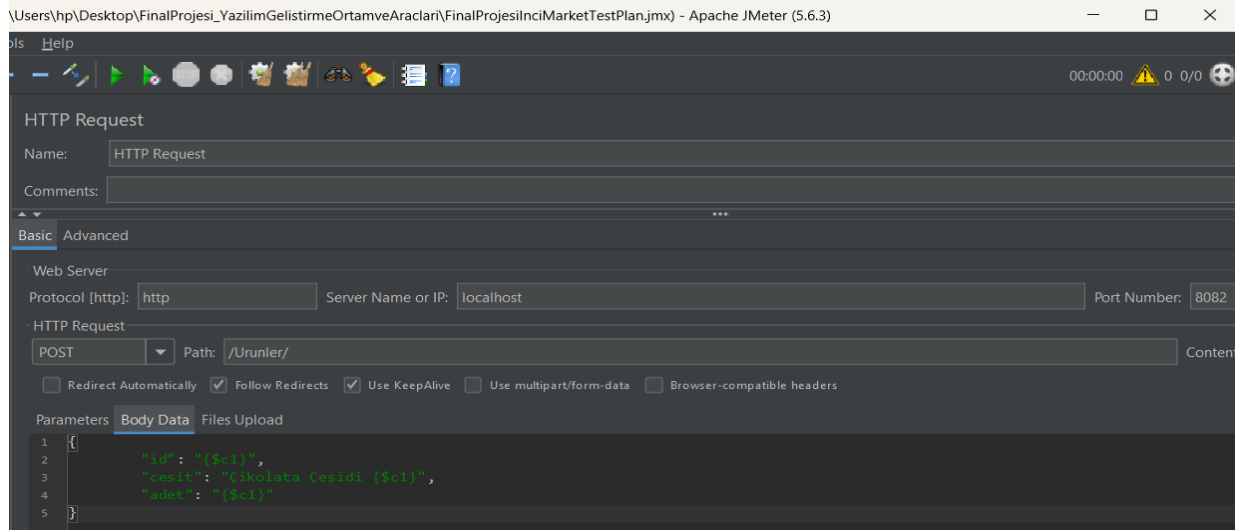
Bu defa da “LoopController”a sağ tıklayarak sırasıyla Add/Config Element/Counter seçiyoruz ve bir sonraki aşamada yine uygun şekilde verileri dolduruyoruz.



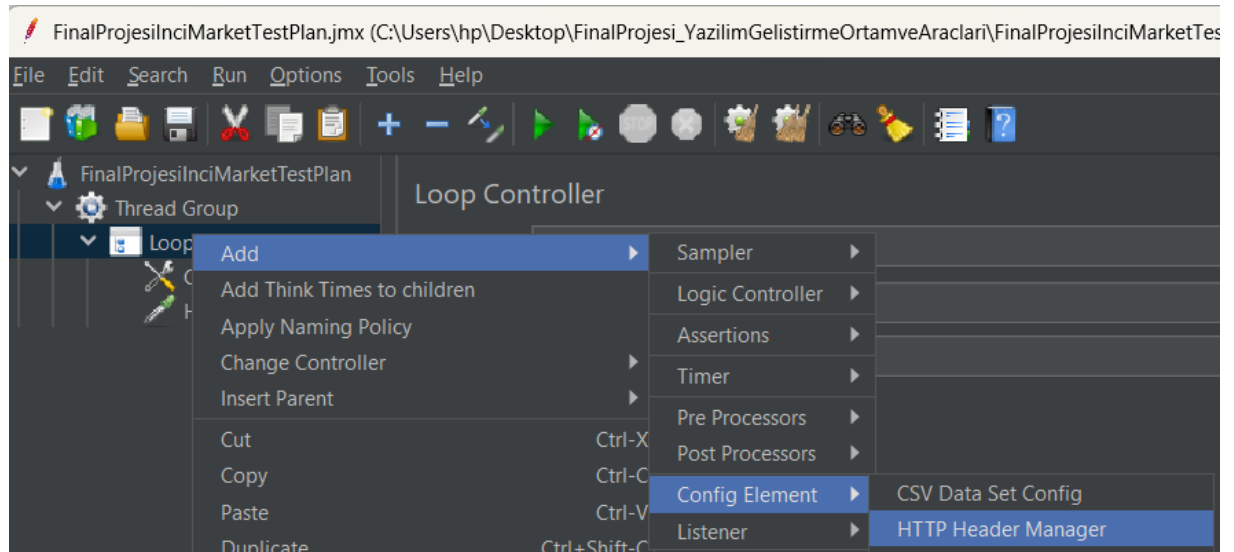
Ve yine de “Loop Controller”a gelerek sağ tıklayıp sırasıyla Add/Sampler/HTTP Request seçiyoruz.



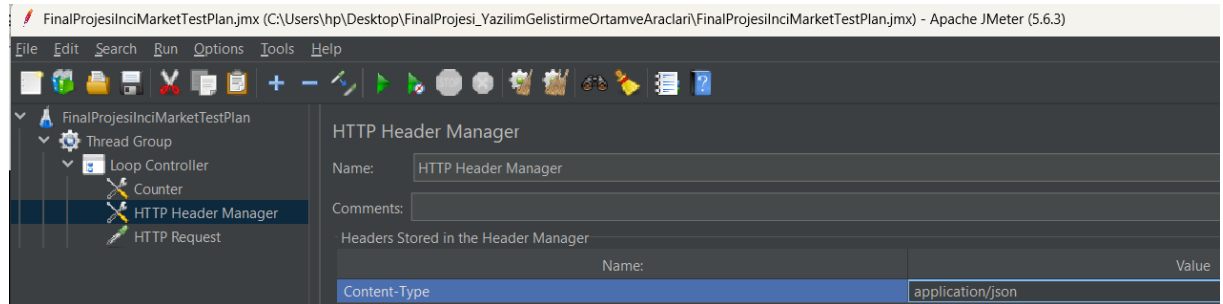
Burada gereken kısımları doldururken ilk isteğimizi POST olarak seçmemizin sebebi sistemin boş olarak gelmesinden kaynaklıdır. “Body Data” kısmında ise Postman’de istek atarken kullandığımız aynı biçimde olmasına özen göstererek yanına “\${sc1}” ifadesini eklemeyi ihmal etmemeliyiz. Bu bir adım öncesinde oluşturduğumuzu “Exported Variable Name”den gelmekte, dışa aktarılan değişken adı.



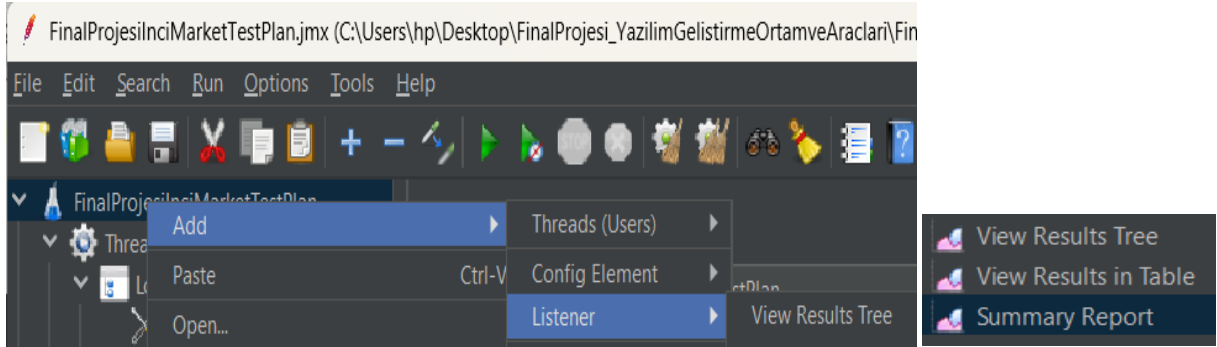
İsteğimizin JSON olduğunu belirten yeni bir Loop Controller’a sağ tıklayıp Add/Config Element/http Header Manager seçiyoruz ve http Requestin önüne sol sütunda sürükleyip bırak yöntemiyle taşıyoruz.



Ardından açılan sayfada “Add” butonuna tıklayıp, Name:Content-Type Value:application/json olarak dolduruyoruz.



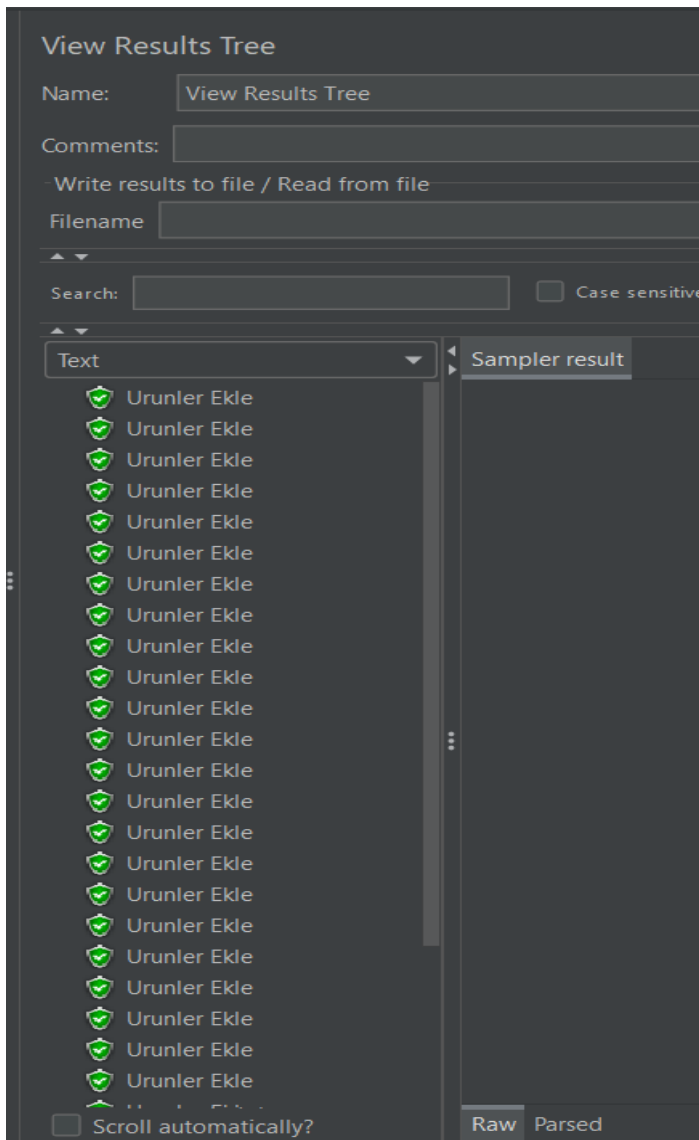
“FinalProjesilnciMarketTestPlan” a sağ tıklayarak Add/Listener/View Result Tree ,View Results Table, Summary Report diyerek aynı sırada 3 farklı test sonuçlarımızı görebileceğimiz şemaları oluşturuyoruz.



Son olarak idemizde programın çalışır durumda olduğunu kontrol edip en tepedeki “FinalProjesilnciMarketTestPlan” üzerine gelip araç çubuğundaki yeşil renkli “play” tuşuna basıyoruz.

Sırasıyla üç şemada da testlerimizin başarılı olduğu ekran görüntülerini aşağıya iletiyoruz.

1-View Results Tree’de isteklerimizin başarılı olduğunu görmüş oluyoruz



2-View Results in Table

View Results in Table

Name: View Results in Table

Comments:

Write results to file / Read from file

Filename: Log/Display Only: ☐ Errors ☐ Successes

| Sample # | Start Time | Thread Name | Label | Sample Time(ms) | Status | Bytes | Sent Bytes | Latency | Connect Time(ms) |
|----------|--------------|------------------|--------------|-----------------|--------|-------|------------|---------|------------------|
| 1 | 22:40:31.529 | Thread Group 1-1 | Urunler Ekle | 11 | ✓ | 239 | 275 | 11 | 4 |
| 2 | 22:40:31.541 | Thread Group 1-1 | Urunler Ekle | 5 | ✓ | 239 | 275 | 5 | 0 |
| 3 | 22:40:31.546 | Thread Group 1-1 | Urunler Ekle | 4 | ✓ | 239 | 275 | 4 | 0 |
| 4 | 22:40:31.551 | Thread Group 1-1 | Urunler Ekle | 6 | ✓ | 239 | 275 | 6 | 0 |
| 5 | 22:40:31.557 | Thread Group 1-1 | Urunler Ekle | 5 | ✓ | 239 | 275 | 5 | 0 |
| 6 | 22:40:31.563 | Thread Group 1-1 | Urunler Ekle | 6 | ✓ | 239 | 275 | 6 | 0 |
| 7 | 22:40:31.569 | Thread Group 1-1 | Urunler Ekle | 9 | ✓ | 239 | 275 | 9 | 0 |
| 8 | 22:40:31.579 | Thread Group 1-1 | Urunler Ekle | 8 | ✓ | 239 | 275 | 7 | 0 |
| 9 | 22:40:31.588 | Thread Group 1-1 | Urunler Ekle | 6 | ✓ | 239 | 275 | 6 | 0 |
| 10 | 22:40:31.595 | Thread Group 1-1 | Urunler Ekle | 5 | ✓ | 239 | 275 | 5 | 0 |
| 11 | 22:40:31.862 | Thread Group 1-2 | Urunler Ekle | 10 | ✓ | 239 | 275 | 9 | 3 |
| 12 | 22:40:31.873 | Thread Group 1-2 | Urunler Ekle | 9 | ✓ | 239 | 275 | 8 | 0 |
| 13 | 22:40:31.883 | Thread Group 1-2 | Urunler Ekle | 9 | ✓ | 239 | 275 | 8 | 0 |
| 14 | 22:40:31.893 | Thread Group 1-2 | Urunler Ekle | 7 | ✓ | 239 | 275 | 7 | 0 |
| 15 | 22:40:31.901 | Thread Group 1-2 | Urunler Ekle | 8 | ✓ | 239 | 275 | 8 | 0 |
| 16 | 22:40:31.910 | Thread Group 1-2 | Urunler Ekle | 8 | ✓ | 239 | 275 | 8 | 0 |
| 17 | 22:40:31.919 | Thread Group 1-2 | Urunler Ekle | 8 | ✓ | 239 | 275 | 8 | 0 |
| 18 | 22:40:31.928 | Thread Group 1-2 | Urunler Ekle | 8 | ✓ | 239 | 275 | 8 | 0 |
| 19 | 22:40:31.937 | Thread Group 1-2 | Urunler Ekle | 8 | ✓ | 239 | 275 | 8 | 0 |
| 20 | 22:40:31.946 | Thread Group 1-2 | Urunler Ekle | 8 | ✓ | 239 | 275 | 8 | 0 |
| 21 | 22:40:32.196 | Thread Group 1-3 | Urunler Ekle | 12 | ✓ | 239 | 275 | 12 | 4 |
| 22 | 22:40:32.209 | Thread Group 1-3 | Urunler Ekle | 9 | ✓ | 239 | 275 | 9 | 0 |
| 23 | 22:40:32.219 | Thread Group 1-3 | Urunler Ekle | 9 | ✓ | 239 | 275 | 8 | 0 |
| 24 | 22:40:32.229 | Thread Group 1-3 | Urunler Ekle | 9 | ✓ | 239 | 275 | 8 | 0 |
| 25 | 22:40:32.230 | Thread Group 1-3 | Urunler Ekle | 8 | ✓ | 239 | 275 | 8 | 0 |

☐ Scroll automatically? ☐ Child samples? No of Samples: 30 Latest Sample: 7 Average: 7 Deviation: 1

3-Summary Report

Summary Report

Name: Summary Report

Comments:

Write results to file / Read from file

Filename: Log/Display Only: ☐ Errors ☐ Successes

| Label | # Samples | Average | Min | Max | Std. Dev. | Error % | Throughput | Received KB/sec | Sent KB/sec | Avg. Bytes |
|--------------|-----------|---------|-----|-----|-----------|---------|------------|-----------------|-------------|------------|
| Urunler Ekle | 30 | 7 | 4 | 12 | 1.79 | 0.00% | 39.3/sec | 9.18 | 10.56 | 239.0 |
| TOTAL | 30 | 7 | 4 | 12 | 1.79 | 0.00% | 39.3/sec | 9.18 | 10.56 | 239.0 |

Ve yine kontrolünü Postman'a gidip GET anotasyonu ile istek atarak yeni ürünleri listeleyip listelemediğini görmek için kontrol ediyoruz.

GET http://localhost:8082/Uru

+

...

HTTP

http://localhost:8082/Urunler/

GET

http://localhost:8082/Urunler/

Params

Authorization

Headers (6)

Body

Pre-request Script

Tests

Settings

Query Params

| Key |
|-----|
|-----|

Body

Cookies

Headers (5)

Test Results

Pretty

Raw

Preview

Visualize

JSON

↻

311. Satıra kadar “Çikolata Çeşidi” eklendiğini görmüş oluyoruz.

The screenshot displays the JMeter GUI. On the left, the 'Test Plan' tree is visible, showing a hierarchy: 'FinalProjesiInciMarketTestPlan' (Test Plan icon) -> 'Thread Group' (Gear icon) -> 'Loop Controller' (Loop icon) -> 'Counter' (Counter icon), 'HTTP Header Manager' (Wrench icon), and 'Urunler Ekle' (Bottle icon). Below this, there are links to 'View Results Tree', 'View Results in Table', and 'Summary Report'. The 'View Results Tree' link is highlighted. On the right, the 'Results Tree' is shown, displaying a list of test results. Each result is preceded by a status icon (red X for failure, green checkmark for success) and the text 'Urunler Sil' or 'Urunler Ekle'. The 'Text' field at the top of the Results Tree is empty.

BİRİM TESTLER

Aşağıda Maven kullanılarak oluşturulmuş Junit birim testlerinden oluşan .java uzantılı classların ve aynı classların test kodlarının ekran resimleri mevcuttur.

```
m pom.xml (InciMarketBirimTest) x © RestWebServisler.java © RestWebServislerTest.java
2  <xsi:schemaLocation="http://maven.apache.org/POM/4.0.0 http://maven.apache.org/xsd/maven-4.0.0.xsd">
3  <modelVersion>4.0.0</modelVersion>
4
5  <groupId>tr.edu.medipol.FinalProjesi</groupId>
6  <artifactId>InciMarketBirimTest</artifactId>
7  <version>1.0-SNAPSHOT</version>
8  <packaging>jar</packaging>
9
10 <name>InciMarketBirimTest</name>
11 <url>http://maven.apache.org</url>
12
13 <properties>
14   <project.build.sourceEncoding>UTF-8</project.build.sourceEncoding>
15   <maven.compiler.source>1.8</maven.compiler.source>
16   <maven.compiler.target>1.8</maven.compiler.target>
17 </properties>
18
19 <dependencies>
20   <dependency>
21     <groupId>junit</groupId>
22     <artifactId>junit</artifactId>
23     <version>3.8.1</version>
24     <scope>test</scope>
25   </dependency>
26   <dependency>
27     <groupId>org.jetbrains</groupId>
28     <artifactId>annotations</artifactId>
29     <version>20.1.0</version>
30   </dependency>
31 </dependencies>
32 </project>
```

```
m pom.xml (InciMarketBirimTest) © RestWebServisler.java x © RestWebS
1  package tr.edu.medipol.FinalProjesi;
2
3  import org.jetbrains.annotations.Contract;
4  import org.jetbrains.annotations.NotNull;
5
6  import java.util.List;
7
8  @SuppressWarnings("new")
9  public class RestWebServisler {
10     @SuppressWarnings("new")
11     @Contract(pure = true)
12     public static @NotNull String urunListele (String id){
13         String hata;
14         if(id==null);
15         hata="Ürün kodu giriniz";
16
17         return hata;
18     }
19
20     @SuppressWarnings("new")
21     @Contract(pure = true)
22     public static @NotNull String urunSil (String id){
23         String hata;
24         if(id==null);
25         hata="Silinecek ürün kodunu giriniz";
26
27         return hata;
28     }
29 }
```

```
27
28     @org.jetbrains.annotations.NotNull
29     @org.jetbrains.annotations.Contract(pure = true)
30     public static String urunEkle (String id){
31         String hata;
32         if(id==null);
33         hata="Ürün kodu olusturunuz";
34         return hata;
35     }
36
37
38 }
```

```

1 package tr.edu.medipol.FinalProjesi;
2
3 import junit.framework.TestCase;
4
5 new *
6 public class RestWebServislerTest extends TestCase {
7
8     new *
9     @Test
10     public void testUrunListele() {
11         RestWebServisler nesne = new RestWebServisler();
12         String hata = new String();
13
14         String urunKodu = "Doğru Ürün kodu";
15
16         assertEquals( expected: "", hata);
17     }
18
19     new *
20     @Test
21     public void testUrunSil() {
22         RestWebServisler nesne2 = new RestWebServisler();
23         String hata2 = new String();
24
25         String urunKodu = "Doğru Ürün kodu";
26
27         assertEquals( expected: "", hata2);
28     }
29 }

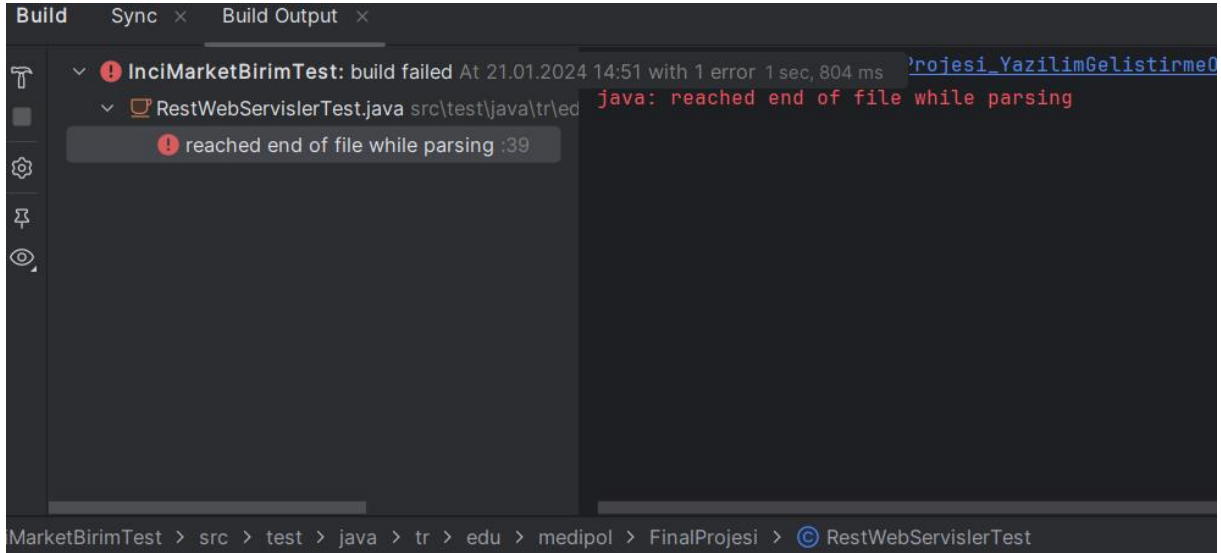
```

```

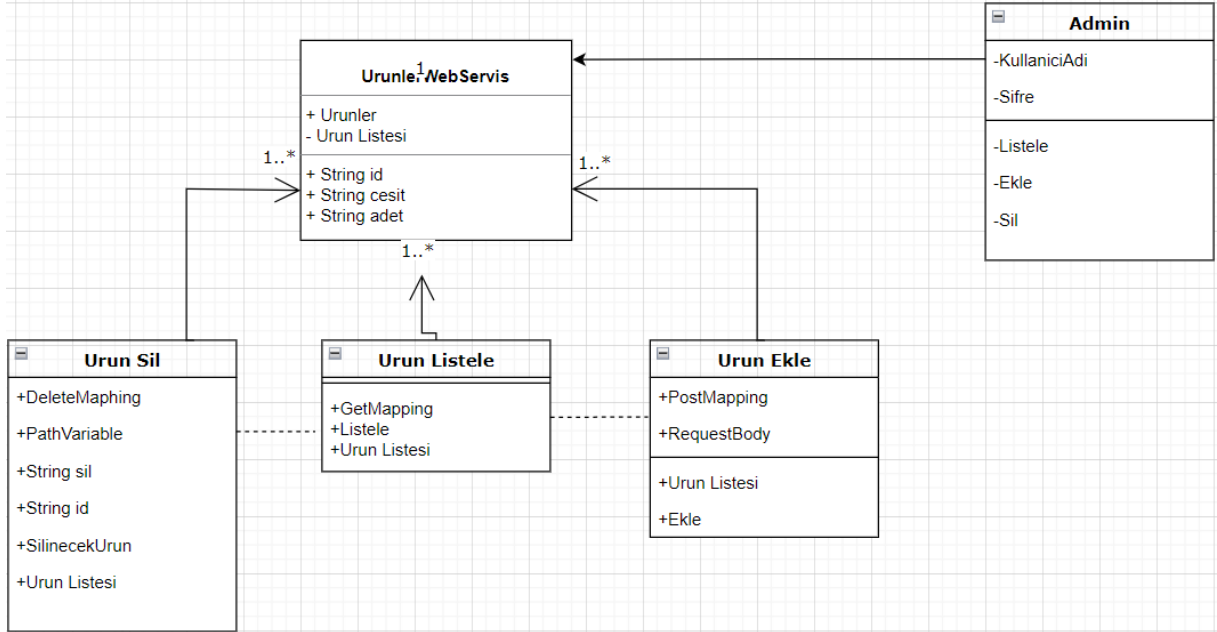
27 public class RestWebServislerTest {
28
29     no usages new *
30     @Test
31     public void testUrunEkle() {
32         RestWebServisler nesne3 = new RestWebServisler();
33         String hata3 = new String();
34
35         String urunKodu = "Doğru Ürün kodu";
36         nesne3.urunEkle(urunKodu);
37
38         assertEquals( expected: "", hata3);
39     }
40 }

```

Proje için oluşturulan birim testlere karşılık karşılaşılan buglara karşılık yeterli zaman olmamasından kaynaklı çözüm oluturulamadığından “Maven build Successé ve oluşması gereken HTML raporunun ekranı paylaşılammıştır. Yerine alınan hatanın ekran görüntüsü eklenmiştir.



UML DİYAGRAM



Oluşturulan UML diyagramda tek kullanıcı girişi sağlanmaktadır ve bu kullanıcı da tüm sisteme hakimdir. Sistemdeki sınıflar birbiriyle çoklu ilişki içerisindedir. Yani birinde olan nesne veya sınıf özelliklerinden herhangi biri diğerinde de mevcuttur.

Örneğin “Urun Listesi” tüm birimlerde mevcut çünkü yapılan her işlemten sonra yani atılan her istekten sonra yine aynı liste içerisindeki verilerden yapılmaktadır. Ve sonuçlardaki değişiklikler de yine aynı mevcut liste üzerinden kontrol edilmektedir. Ayrıca her hangi bir gizlilik şu anlık öncelik tutulmadığından, bir simülasyon olduğundan ve test aşamasında kolay ulaşılabilirlik açısından da çoğunlukla public (+) statüde tanım yapılmıştır.

GİT, SÜREKLİ ENTEGRASTON (GITHUB ACTIONS)

Bu adımda hem Github repository oluşturup hem de Git komutlarıyla oluşturduğumuz bu repositoryye Github Actions konfigürasyonu ile sürekli entegrasyon ayarını nasıl yaptığımızı rapor haline getirmiş olacağız.

Top Repositories



Find a repository...



kubraaykol/IleriJavaVizeProjesi

Required fields are marked with an asterisk (*).

Owner *



kubraaykol

Repository name *

FinalProjesiYazilimGelistirr

✓ FinalProjesiYazilimGelistirmeOrtamveAraclari1 is available.

Great repository names are short and memorable. Need inspiration? How about [reimagined-tribble](#) ?

Description (optional)



Public

Anyone on the internet can see this repository. You choose who can commit.



Private

You choose who can see and commit to this repository.



FinalProjesiYazilimGelistirmeOrtamveAraclari

Private

Unwatch 1

main



1 Branch



0 Tags

Go to file



Add file

Code



kubraaykol Initial commit

2cf9f0d · 14 minutes ago

1 Commits



README.md

Initial commit

14 minutes ago

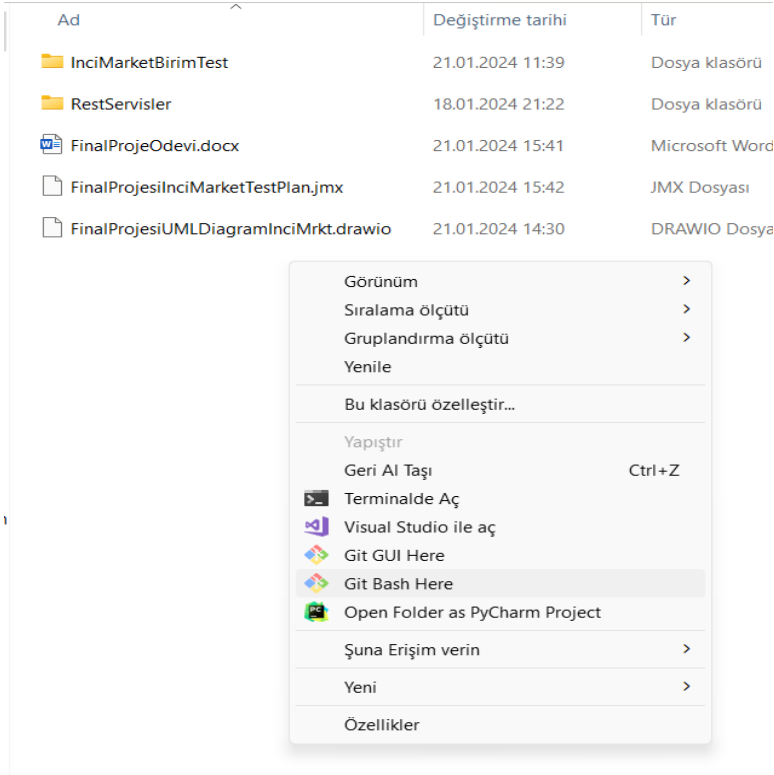


README

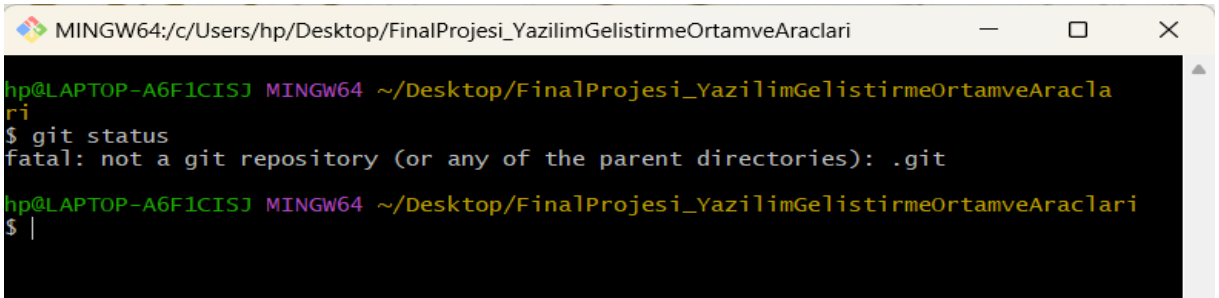


FinalProjesiYazilimGelistirmeOrtamveAraclari

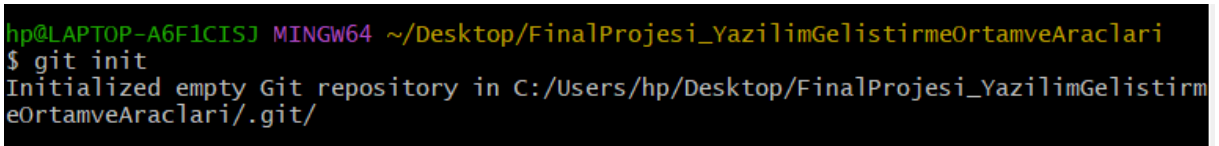
“Git Bash Here” tıklayıp terminal ekranına geçmeden önce klasörün boş olmasını ihmal etmemeliyiz.



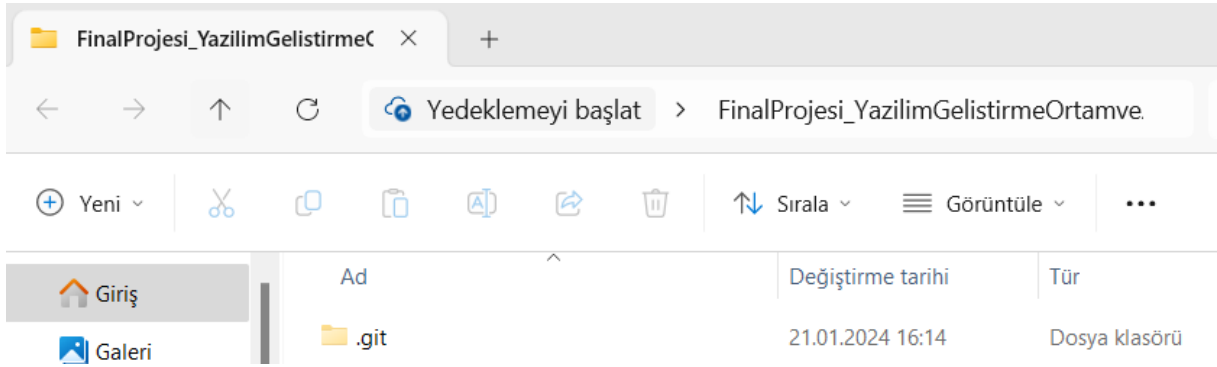
Açılan terminalde “git status” yazıp enter yaptığımızda aşağıdaki görselde görüldüğü üzere git repository olmadığını görmüş oluyoruz.



“git init” yazıp boş repositoryimizi oluşturuyoruz.



Ve bu işlem sonrasında projemizin kayıtlı olduğu klasörde .git uzantılı klasörü görüyoruz.



“git status” yaptıktan sonra şu an ki durumumuz aşağıdaki gibi olduğunu görüyoruz.

```
hp@LAPTOP-A6F1CISJ MINGW64 ~/Desktop/FinalProjesi_YazilimGelistirmeOrtamveAracLari (master)
$ git status
On branch master

No commits yet

nothing to commit (create/copy files and use "git add" to track)

hp@LAPTOP-A6F1CISJ MINGW64 ~/Desktop/FinalProjesi_YazilimGelistirmeOrtamveAracLari (master)
$
```

Şimdiden itibaren tüm git komutlarımızı aşağıdaki sırayla iletiyoruz.

```
MINGW64:/c/Users/hp/Desktop/FinalProjesiYazilimGelistirmeOrtamveAracLari

hp@LAPTOP-A6F1CISJ MINGW64 ~/Desktop/FinalProjesiYazilimGelistirmeOrtamveAracLari
$ git init
Initialized empty Git repository in C:/Users/hp/Desktop/FinalProjesiYazilimGelistirmeOrtamveAracLari/.git/

hp@LAPTOP-A6F1CISJ MINGW64 ~/Desktop/FinalProjesiYazilimGelistirmeOrtamveAracLari (master)
$ git status
On branch master

No commits yet

nothing to commit (create/copy files and use "git add" to track)

hp@LAPTOP-A6F1CISJ MINGW64 ~/Desktop/FinalProjesiYazilimGelistirmeOrtamveAracLari (master)
$ git config --local user.name "kubraaykol"

hp@LAPTOP-A6F1CISJ MINGW64 ~/Desktop/FinalProjesiYazilimGelistirmeOrtamveAracLari (master)
$ git config --local user.mail "kubraaykol<outlook.com>"
> AC

hp@LAPTOP-A6F1CISJ MINGW64 ~/Desktop/FinalProjesiYazilimGelistirmeOrtamveAracLari (master)
$ git config --local user.mail "kubraaykol@outlook.com"

hp@LAPTOP-A6F1CISJ MINGW64 ~/Desktop/FinalProjesiYazilimGelistirmeOrtamveAracLari (master)
$ git config --local user.email "kubraaykol@outlook.com"
```

```

hp@LAPTOP-A6F1CISJ MINGW64 ~/Desktop/FinalProjesiYazilimGelistirmeOrtamveAraclar
i (master)
$ git status
On branch master

No commits yet

Untracked files:
  (use "git add <file>..." to include in what will be committed)
      FINALPROJESIAZILIMGELISTIRME.txt

nothing added to commit but untracked files present (use "git add" to track)

hp@LAPTOP-A6F1CISJ MINGW64 ~/Desktop/FinalProjesiYazilimGelistirmeOrtamveAraclar
i (master)
$ git add FINALPROJESIAZILIMGELISTIRME.txt

hp@LAPTOP-A6F1CISJ MINGW64 ~/Desktop/FinalProjesiYazilimGelistirmeOrtamveAraclar
i (master)
$ git status
On branch master

No commits yet

Changes to be committed:
  (use "git rm --cached <file>..." to unstage)
      new file:   FINALPROJESIAZILIMGELISTIRME.txt

hp@LAPTOP-A6F1CISJ MINGW64 ~/Desktop/FinalProjesiYazilimGelistirmeOrtamveAraclar
i (master)
$ git commit -m "Final projesi raporu önce .txt olarak yüklendi."
[master (root-commit) 51be57c] Final projesi raporu önce .txt olarak yüklendi.
 1 file changed, 163 insertions(+)
 create mode 100644 FINALPROJESIAZILIMGELISTIRME.txt

hp@LAPTOP-A6F1CISJ MINGW64 ~/Desktop/FinalProjesiYazilimGelistirmeOrtamveAraclar
i (master)
$ git log
commit 51be57c626b1bdcd3b2da445bebb421fd2c70489 (HEAD -> master)
Author: kubraaykol <kubraaykol@outlook.com>
Date:   Sun Jan 21 17:05:05 2024 +0300

    Final projesi raporu önce .txt olarak yüklendi.

```

```

hp@LAPTOP-A6F1CISJ MINGW64 ~/Desktop/FinalProjesiYazilimGelistirmeOrtamveAraclar
i (master)
$ git remote add origin https://github.com/kubraaykol/FinalProjesiYazilimGelistirmeOrtamveAracлари

hp@LAPTOP-A6F1CISJ MINGW64 ~/Desktop/FinalProjesiYazilimGelistirmeOrtamveAracлари (master)
$ git config --global push.autoSetupRemote true

hp@LAPTOP-A6F1CISJ MINGW64 ~/Desktop/FinalProjesiYazilimGelistirmeOrtamveAracлари (master)
$ git push
Enumerating objects: 3, done.
Counting objects: 100% (3/3), done.
Delta compression using up to 8 threads
Compressing objects: 100% (3/3), done.
Writing objects: 100% (3/3), 4.08 KiB | 695.00 KiB/s, done.
Total 3 (delta 0), reused 0 (delta 0), pack-reused 0
remote:
remote: Create a pull request for 'master' on GitHub by visiting:
remote:   https://github.com/kubraaykol/FinalProjesiYazilimGelistirmeOrtamveAracлари/pull/new/master
remote:
To https://github.com/kubraaykol/FinalProjesiYazilimGelistirmeOrtamveAracлари
 * [new branch]      master -> master
branch 'master' set up to track 'origin/master'.

hp@LAPTOP-A6F1CISJ MINGW64 ~/Desktop/FinalProjesiYazilimGelistirmeOrtamveAracлари (master)
$

```

Projemizin raporunu burada öncelikle .txt olarak yüklüyoruz. Daha sonra raporumuzun güncel halini PDF uzantısına çevirip diğer tüm dosyalarla beraber Github hesabımıza yüklemiş olacağız.