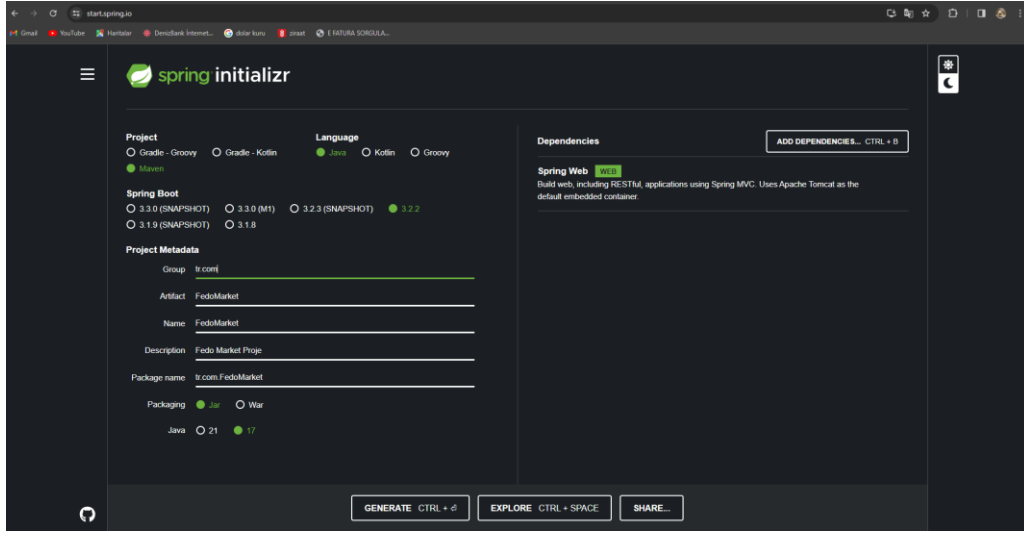


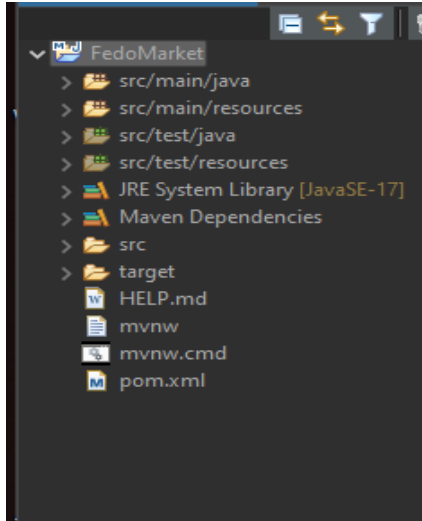
YAZILIM GELİŞTİRME VE ORTAM ARAÇLARI DERSİ FİNAL PROJESİ

FEDO MARKET WEB SERVİS PROJESİ



1.) SPRİNG BOT SİTESİNDEN PROJE OLUŞTURMA

Öncelikle “star.spring.io” linkinden yukarıda belirttiğim resimdeki seçenekleri seçerek bir proje oluşturdum. Spring bot kullanacağımızdan dolayı Spring Web seçeneğini seçip “GENERATE” diyerek zip dosyasını indiriyoruz.



Masaüstüne ayıkladıktan sonra ECLIPSE İDE ye import ettim.

Soldaki görüntü ile karşılaştıysam problem yok demektir.

Kodlarımı yazmaya başlayabilirim demektir.

```
UrunController.java  UrunServis.java  Urun.java  FedoMarketApplication.java X
1 package tr.com.FedoMarket;
2 import org.springframework.boot.SpringApplication;
3
4
5 @SpringBootApplication
6 public class FedoMarketApplication {
7     public static void main(String[] args) {
8         SpringApplication.run(FedoMarketApplication.class, args);
9     }
10 }
```

2) @SPRINGAPPLICATION

Spring Boot uygulamasını başlatmak için Application.java klasörüne bu komutu yazıyorum.

Bu örnekte,

@SpringBootApplication annotasyonu, genellikle kullanılan birkaç diğer annotasyonu bir araya getirir ve bir Spring Boot uygulamasının temel yapılandırmasını sağlar.

```

1 package tr.com.FedoMarket;
2
3 import org.springframework.web.bind.annotation.*;
4
5
6
7 @RestController
8 @RequestMapping("/urun")
9 public class UrunWebservisi {
10
11     public record Urun(String ad, String numara, String fiyat) {};
12
13
14     private static final List<Urun> URUN_LISTESI = new ArrayList<>();
15

```

3) Package komutum ile Java sınıfının belirtilen pakete ait olduğunu belirttim.

İmport satırı ile Spring Web ve koleksiyonlar için gerekli olan sınıfları içe aktardım.

Public record komutuyla (ad, numara, fiyat) olan Urun adında bir kayıt sınıfını tanımladım.

Private static final komutu ile , Urun kayıtlarının örneklerini içeren URUN_LISTESI adında bir statik liste tanımladım. Başlangıçta boş bir ArrayList olarak başlattım.

```

16
17 @RestController
18 @RequestMapping("/urun")
19 public class UrunController {
20
21
22     @Autowired
23     public UrunServis urunServis;
24
25
26     @PostMapping("/urunEkle")
27     public ResponseEntity<String> stokArtir(@RequestBody Urun urun) {
28
29         return ResponseEntity.ok(urunServis.urunStokArtir(urun));
30
31     }
32
33
34     @GetMapping("/urunleriListele")
35     public List<Urun> stokListele () {
36
37         return urunServis.urunStokListele();
38
39     }
40
41
42     @DeleteMapping("/urunAzalt")
43     public ResponseEntity<String> stokAzalt(@RequestParam String urunAdi) {
44
45         return ResponseEntity.ok(urunServis.urunStokAzalt(urunAdi));
46
47     }
48
49 }

```

4) Üst tarafta görülen kodlarla **LİSTELEME, ÜRÜN BULMA, ÜRÜN SİLME, ÜRÜN EKLEME** metodlarını kodladım

- **Listeleme Metodu:**

Endpoint: @GetMapping("/")

Mevcut ürün listesini döndürür.

- **Ürün Bulma Metodu:**

Endpoint: @GetMapping("/{no}")

Belirtilen numaraya sahip ürünü bulup döndürür.

- **Ürün Silme Metodu:**

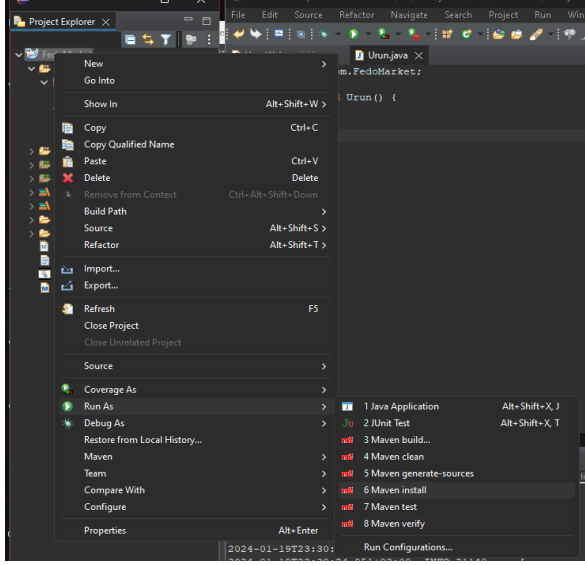
Endpoint: @DeleteMapping("/{no}")

Belirtilen numaraya sahip ürünü listeden siler.

- **Ürün Ekleme Metodu:**

Endpoint: @PostMapping("/")

Gelen ürünü listeye ekler.

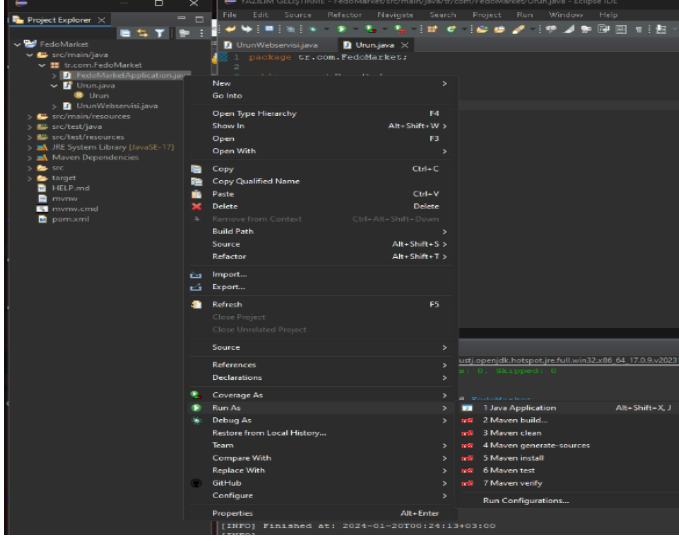


5) Kodlarımızı yazdıktan sonra sağ click tuşu ile projemize tıklayıp “Run As” , ardından “Maven Install” ı seçerek projemize hata testi yapıyoruz

```
<terminated> C:\Users\slckc\p2\pool\plugins\org.eclipse.justi.openjdk.hotspot.jre.full.win32.x86_64_17.0.9.v20231028-0838\re\bin\javaw.exe (20 Oca 2024 01:22:17) [pid: 5632]
[INFO]
[INFO]
[INFO] --- jar:3.3.0:jar (default-jar) @ FedoMarket ---
[INFO] Building jar: C:\Users\slckc\git\yazilimgelistirme-final-projesi\FedoMarket\target\FedoMarket-0.0.1-SNAPSHOT.jar
[INFO]
[INFO] --- spring-boot:3.2.1:repackage (repackage) @ FedoMarket ---
[INFO] Replacing main artifact C:\Users\slckc\git\yazilimgelistirme-final-projesi\FedoMarket\target\FedoMarket-0.0.1-SNAPSHOT.jar with repackaged archive
[INFO] The original artifact has been renamed to C:\Users\slckc\git\yazilimgelistirme-final-projesi\FedoMarket\target\FedoMarket-0.0.1-SNAPSHOT.jar.original
[INFO]
[INFO] --- install:3.1.1:install (default-install) @ FedoMarket ---
[INFO] Installing C:\Users\slckc\git\yazilimgelistirme-final-projesi\FedoMarket\pom.xml to C:\Users\slckc\.m2\repository\tr\com\FedoMarket\0.0.1-SNAPSHOT
[INFO] Installing C:\Users\slckc\git\yazilimgelistirme-final-projesi\FedoMarket\target\FedoMarket-0.0.1-SNAPSHOT.jar to C:\Users\slckc\.m2\repository\tr\com\FedoMarket\0.0.1-SNAPSHOT
[INFO]
[INFO] BUILD SUCCESS
[INFO]
[INFO]
[INFO] Total time: 5.671 s
[INFO] Finished at: 2024-01-20T01:22:24+03:00
[INFO]
```

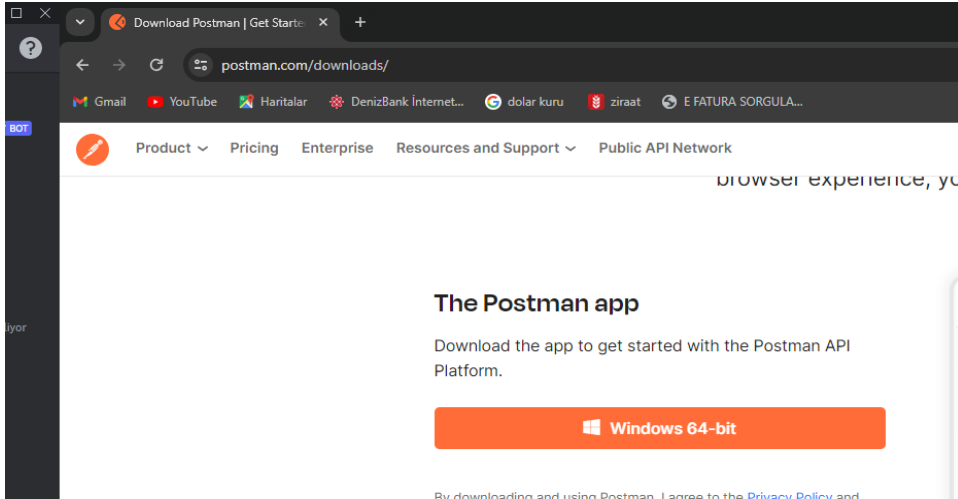
6) **BUILD SUCCESS** Bu yazıyı gördüysek şu ana kadar harika bir yazılımcıyız demektir.

Çünkü bu yazı komutlarımızda herhangi bir hata olmadığı anlamına gelir.



7) Build Succes yazısını görüp hata olmadığını anladıktan sonra “src/main/java” klasörünün altında olan “tr.com.FedoMarket” klasörünün içerisindeki “Application.java” dosyasına sağ click ile

“Runn As” ardından “Java Application” seçeneğini seçerek programı java ile çalıştırıyoruz.

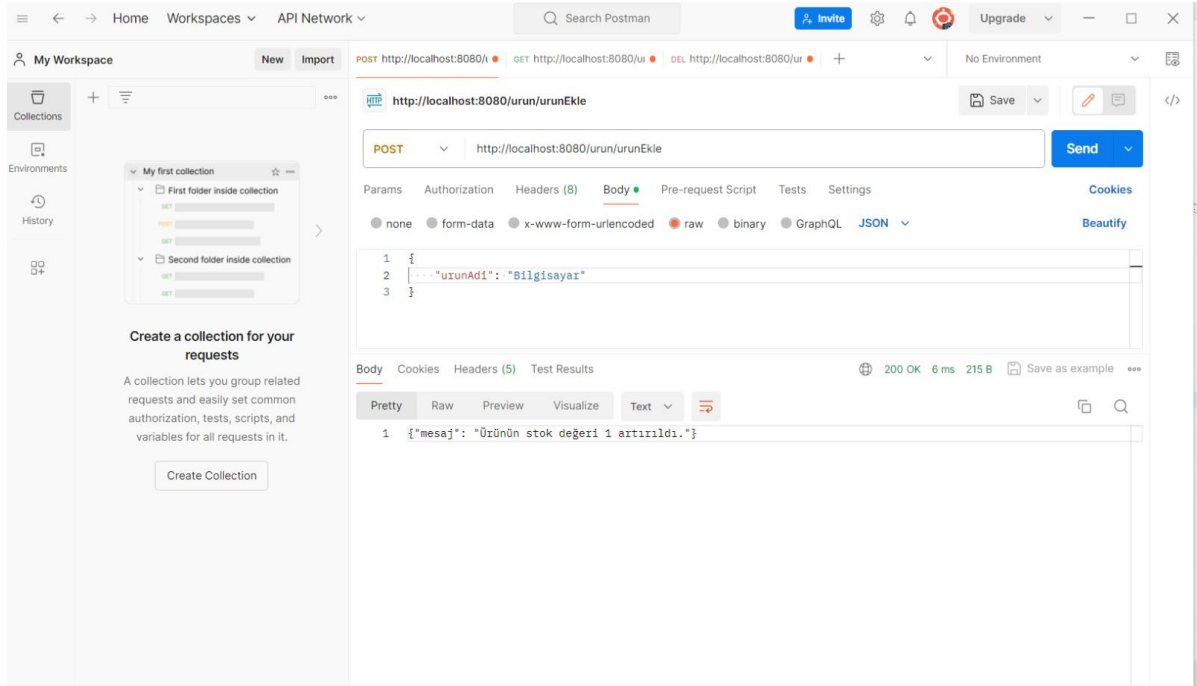


8)PROGRAM DENEME AŞAMASI

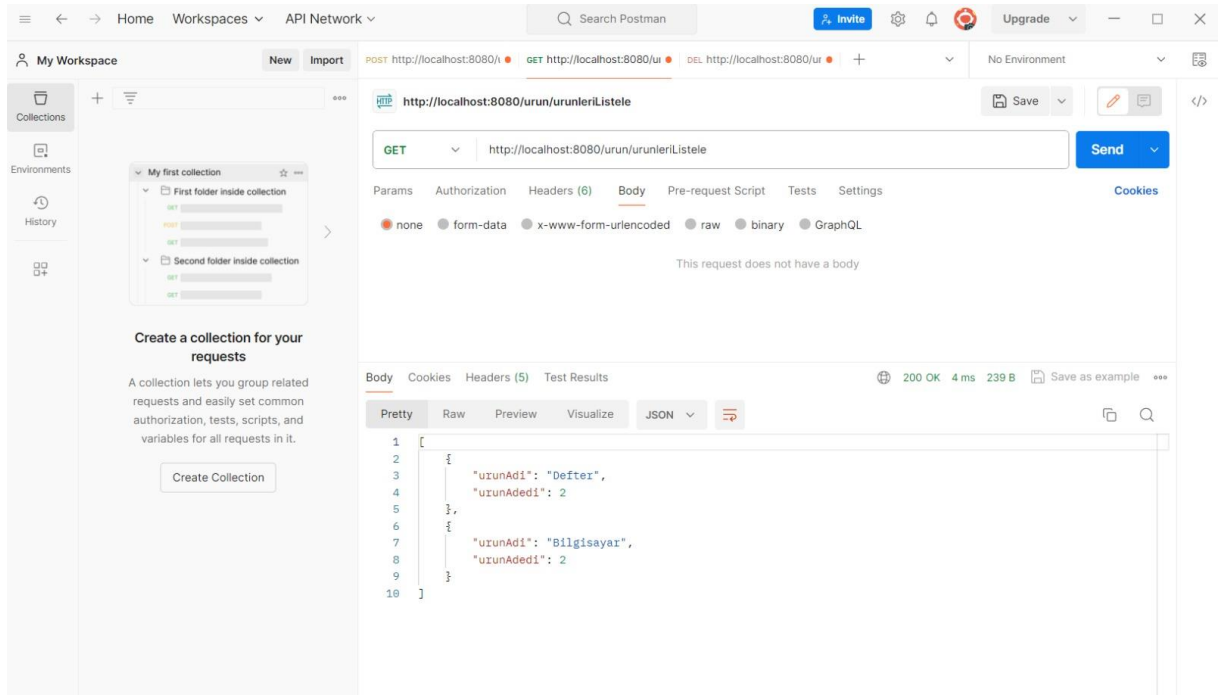
Yazdığımız kodları “Postman” programıyla çalıştıracağız. “postman.com/downloads/” bağlantısına gidip bilgisayarımız için uygun sürümü indirip çalıştırıyoruz.

POST http://localhost:8080/urun GET http://localhost:8080/urun DEL http://localhost:8080/urun

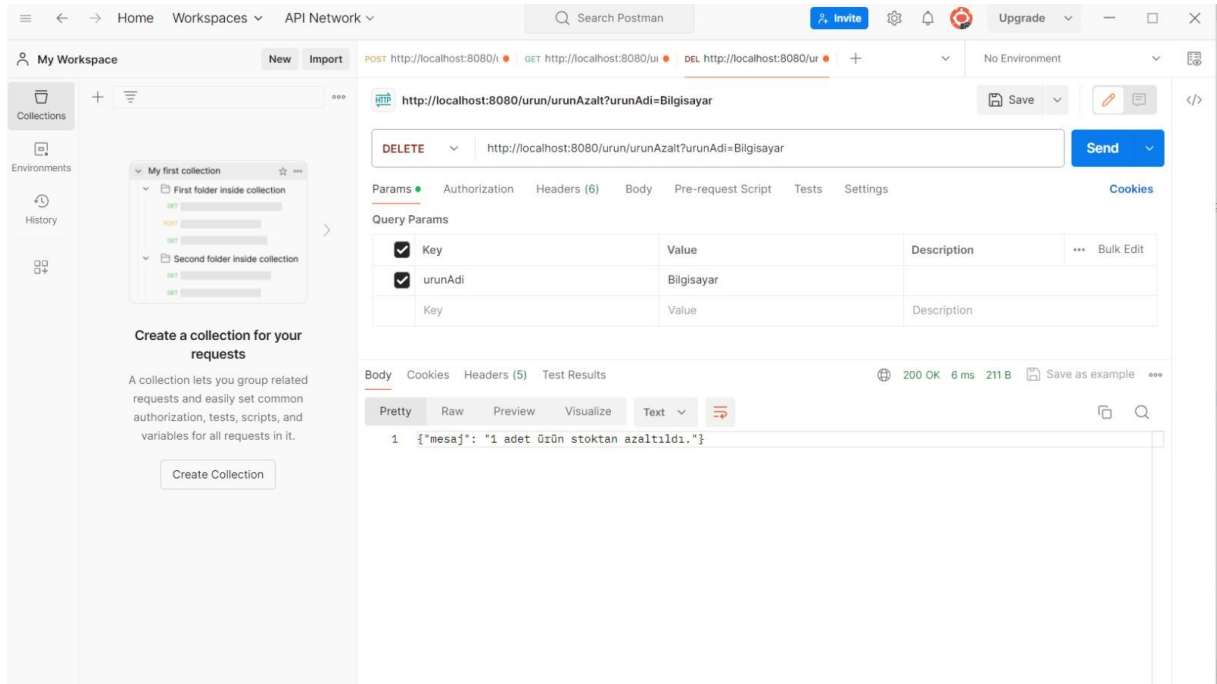
- New kısmından **ÜRÜN SİLME, ÜRÜN EKLEME ve LİSTELEME** olacak şekilde adet dosya “New Project” oluşturuyoruz.



- POST (ÜRÜN EKLEME) :
- Sekmemizin “Headers” kısmına “Content-Type:application/json” komutunu yazdıktan sonra “Body” kısmından “raw>JSON” seçeneğini seçiyoruz. Ve ardından web servisimizin URL sini yazıyoruz. Ardından bir Ürün stringlerini (bilgileri) giriyoruz . ardından “ Send” butonuna tıklıyoruz.

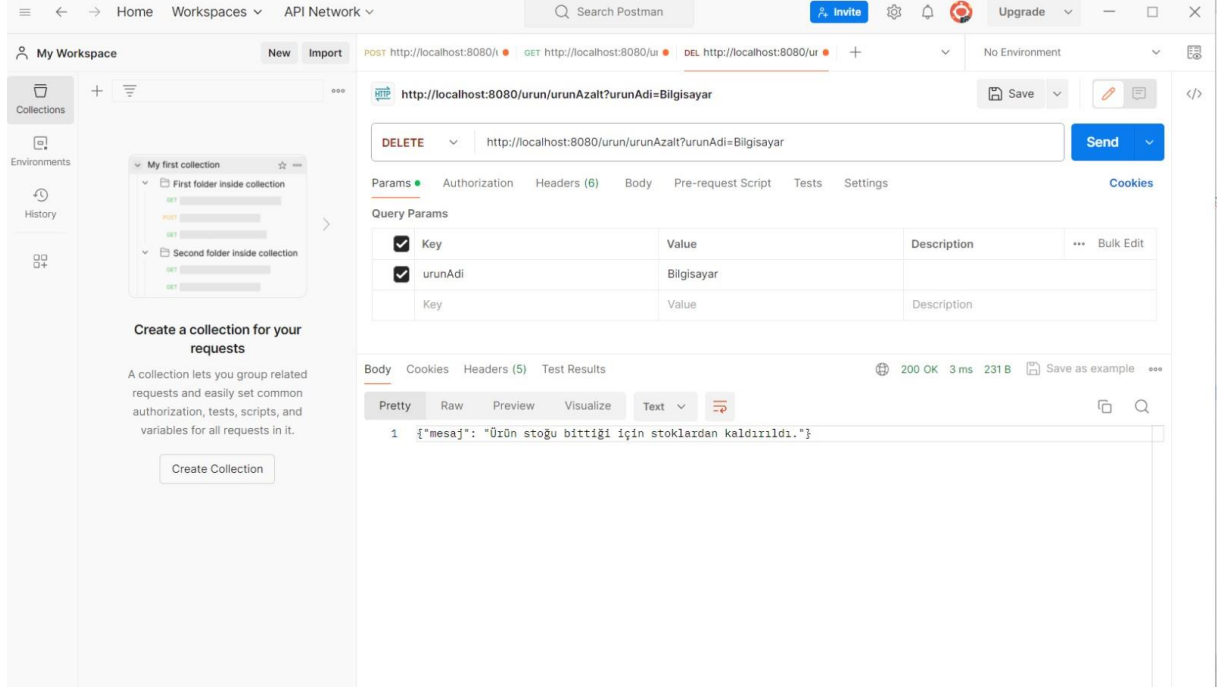


- GET (ÜRÜN LİSTELEME) :
- Bu sekmemiz ise “POST” sekmesinde girilen ürünleri listelememizi sağlar. GET butonunun yanında bulunan boşuğa Web Servis URL mizi girdikten sonra “Send” butonuna clickliyoruz. Ve POST sekmesinde eklenen tüm ürünler listelenmiş şekilde karşımıza çıkıyor.



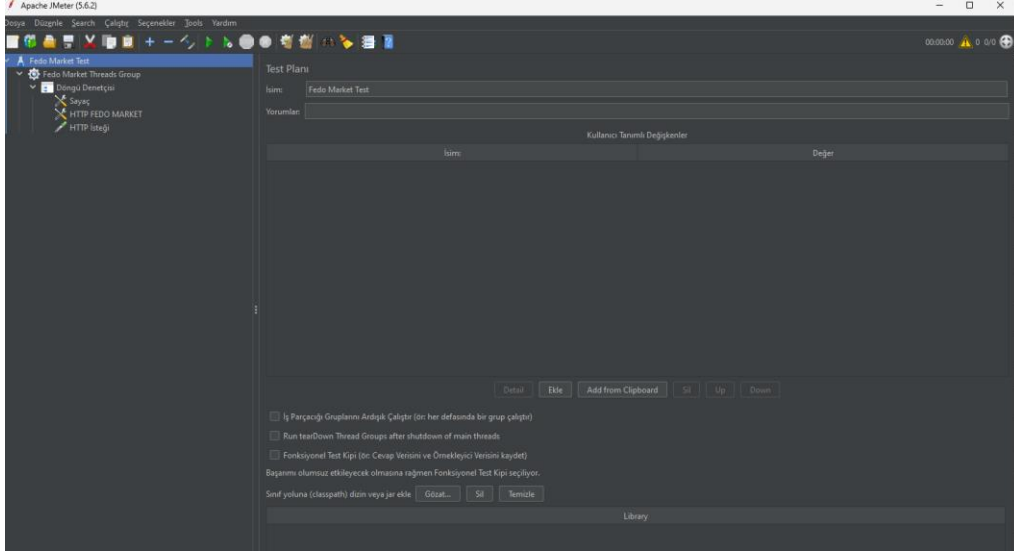
- **DELETE (ÜRÜN SİLME)** : Postta girilen Ürünlerden Stokta kalmayanları silmemizi sağlar. Burada ise Web Servis linkimize ek olarak silmek istediğimiz ürünün numarasını belirtmemiz gerekmektedir.
Örneğin yukarıdaki gibi
“http://localhost:8080/urun/1” şeklindedir. Ardından
“SEND” butonuna clickliyoruz.
- Eğer girdiğimiz ürün numarası stokta bulunan bir ürünün numarası ile uyuşuyorsa yukarıdaki gibi
“{\"mesaj\": \"1 adet ürün stoktan azaltıldı.\"}” şeklinde geri dönüş alırız,

- Deęil ise aŗaęıdaki gibi “{"mesaj": "Ürün stoęu bittięi için stoklardan kaldırılmıŗtır"}” dönüşünü alırız.

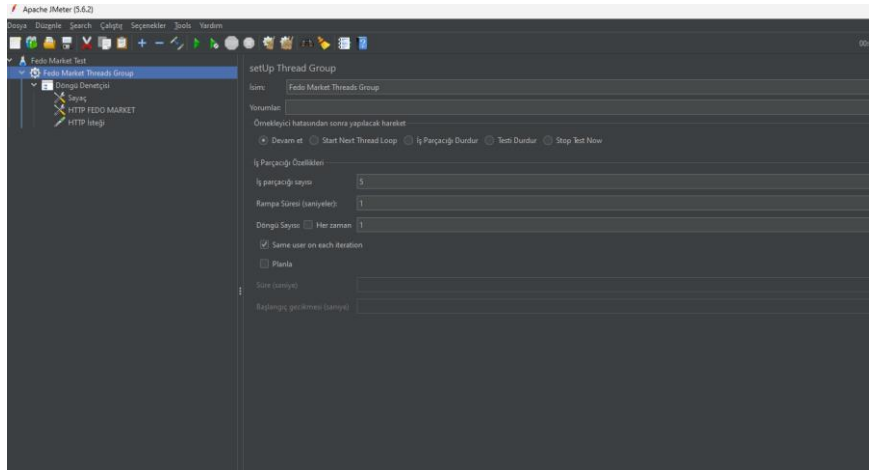


9) PROGRAM TEST AŗAMASI

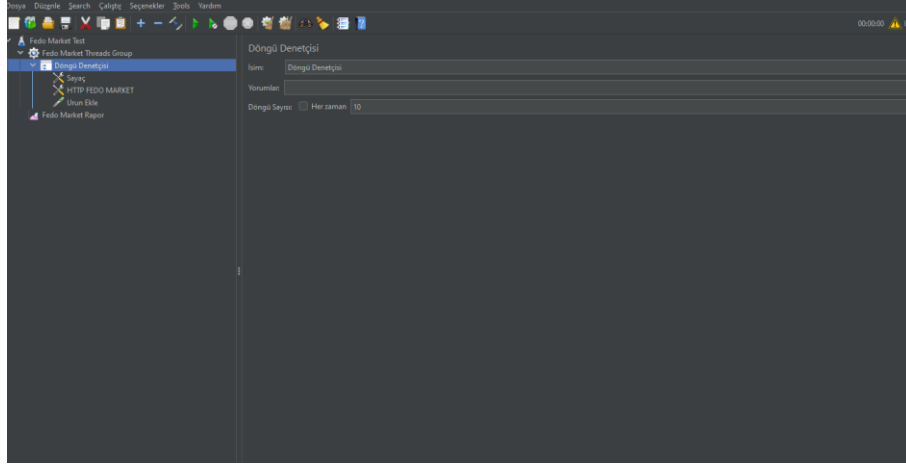
Program testimizi “Apache JMeter” uygulamasıyla yapacaęız. “jmeter.apache.org/download_jmeter.cgi” baęlantısına gidip “Apache-jmeter-5.6.2.zip” dosyasını indiriyoruz. Masaüstüne ayıkladıktan sonra Bilgisayarımızdan “jdk.11>bin” sekmesinden cmd komutu ile çalıştırıyoruz.



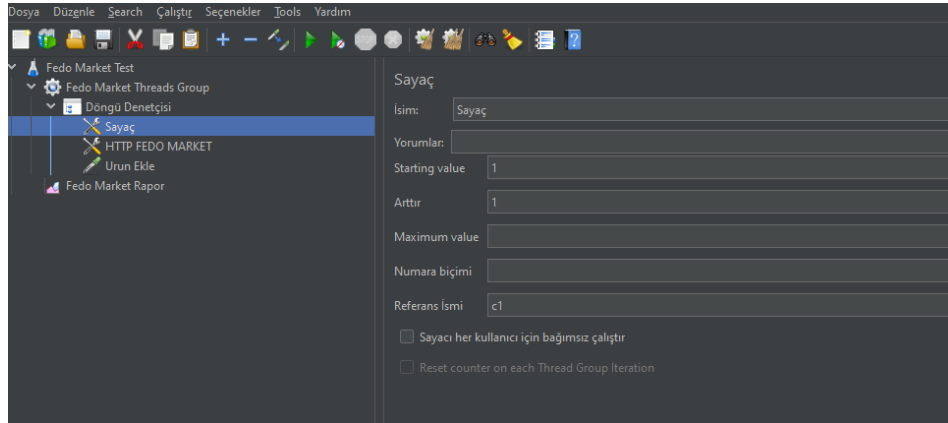
- Test planımıza isim vererek başlıyoruz.



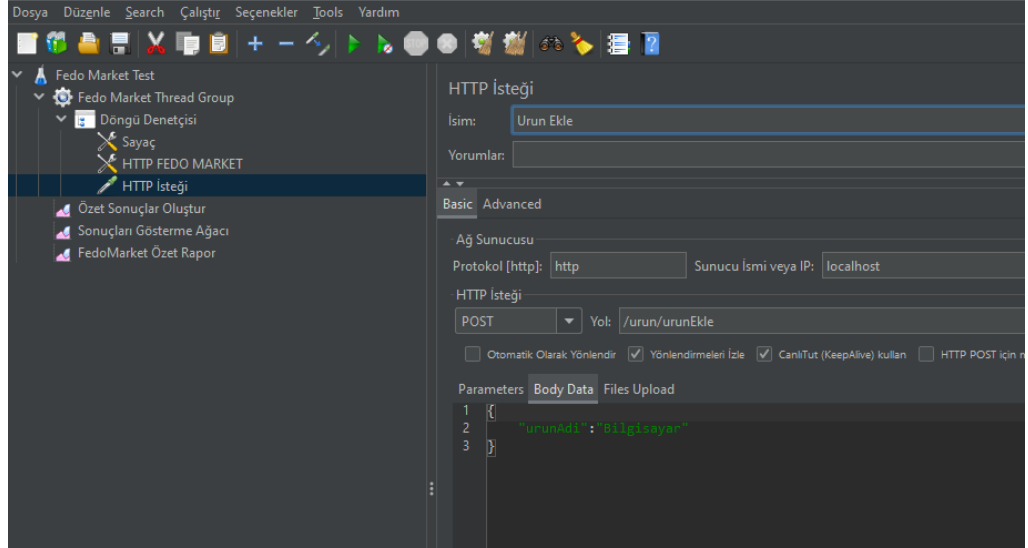
- Ardından başlığımıza sağ click liyip “Add” sekmesinden “Threads Group” ekliyoruz ve test sırasında kaç kullanıcı tarafından ne kadar Döngü (İşlem) göndereceğimizi seçiyoruz.



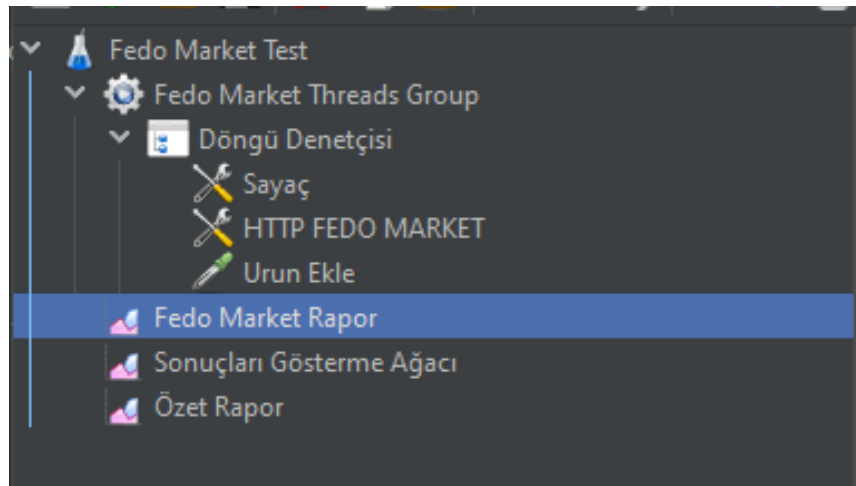
- Threads grubumuza da aynı şekilde “Add>denetleyici” sekmesinden “Döngü Denetçisi” ekliyoruz. Ve Döngü sayısını giriyoruz.



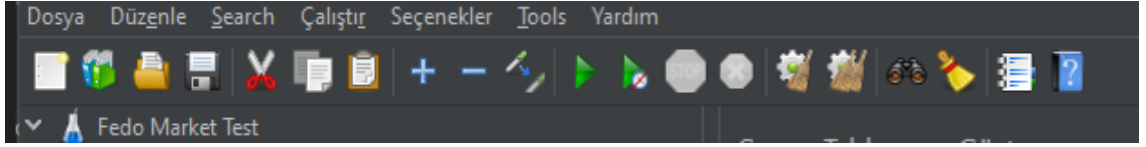
- Döngü Denetçisinin içerisine ise “Sayaç” , “HTTP HEADSET” ve “HTTP İSTEĞİ” sekmelerini açıyoruz.
- Sayaç in içerisine arttırmayı 1 olarak bıraktıktan sonra referans ismimizi test amaçlı “c1” yapıyoruz.



- HTTP HEAD SET seçeneğini kendi marketimize göre ayarladıktan sonra HTTP İsteği' ni :
 - İSİM : Urun ekle
 - Protokol : http
 - Sunucu ismi: localhost
 - IP : 8080
- Yaptıktan sonra isteğimiz ürün ekleme testi olduğu için “POST” Seçeneğini seçip kodlarımızda yazdığımız “/urun/urunEkle” yolunu yazıyoruz.
- Ardından “Body Data” sekmesine gelip örnek ürün bilgilerini “\${referans ismi}” olacak şekilde yazıyoruz.



- Test Sonuçlarımızı ve raporlarımızı görmek için, “Threads Group” sekmesine sağ click yapıp “Add>Dinleyici” sekmesinden “Market Rapor”, “Sonuçları Gösterme Ağacı” ve “Özet Rapor” seçeneğini açıyoruz.

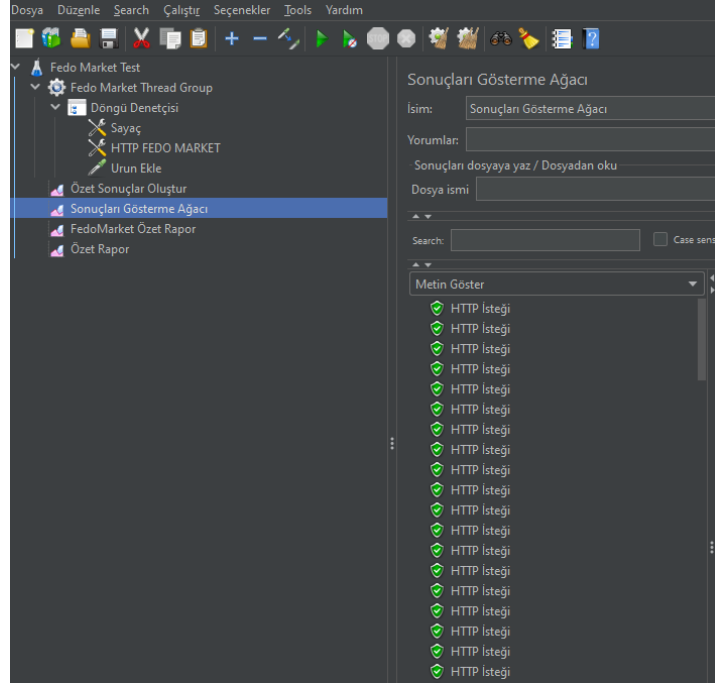


- Ardından üstte bulunan kısayollar sekmesinden “Tools”un altında bulunan Yeşil renkli “Play” Butonuna tıklıyoruz.

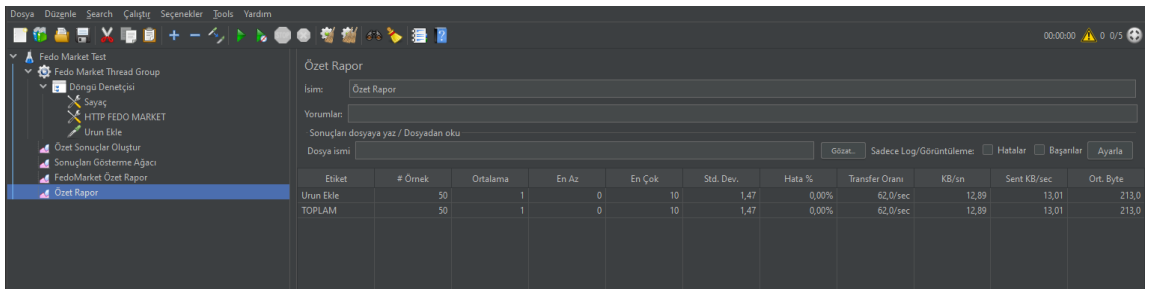
A screenshot of the application interface. The 'Sonuç Tablosunu Göster' window is open, displaying a table of test results. The table has columns for 'Örnek #', 'Başlama Zamanı', 'İş Parçası İsmi', 'Etiket', 'Örnek Zamanı(ms)', 'Durum', 'Bayt', 'Sent Bytes', 'Latency', and 'Connect Time(ms)'. The table contains 15 rows of data, all with a green checkmark in the 'Durum' column, indicating successful test results. The 'Başlama Zamanı' column shows times starting from 00:30:52.415. The 'Örnek Zamanı(ms)' column shows values ranging from 47 to 3. The 'Durum' column shows green checkmarks. The 'Bayt' column shows values ranging from 221 to 224. The 'Sent Bytes' column shows values ranging from 242 to 245. The 'Latency' column shows values ranging from 42 to 3. The 'Connect Time(ms)' column shows values ranging from 34 to 0.

Örnek #	Başlama Zamanı	İş Parçası İsmi	Etiket	Örnek Zamanı(ms)	Durum	Bayt	Sent Bytes	Latency	Connect Time(ms)
1	00:30:52.415	Fedo Market Thre...	Urun Ekle	47	✓	221	242	42	34
2	00:30:52.462	Fedo Market Thre...	Urun Ekle	3	✓	221	242	3	0
3	00:30:52.465	Fedo Market Thre...	Urun Ekle	3	✓	221	242	2	0
4	00:30:52.468	Fedo Market Thre...	Urun Ekle	2	✓	221	242	2	0
5	00:30:52.471	Fedo Market Thre...	Urun Ekle	5	✓	221	242	5	0
6	00:30:52.476	Fedo Market Thre...	Urun Ekle	6	✓	221	242	6	0
7	00:30:52.482	Fedo Market Thre...	Urun Ekle	2	✓	221	242	2	0
8	00:30:52.484	Fedo Market Thre...	Urun Ekle	2	✓	221	242	2	0
9	00:30:52.487	Fedo Market Thre...	Urun Ekle	1	✓	221	242	1	0
10	00:30:52.489	Fedo Market Thre...	Urun Ekle	2	✓	224	245	2	0
11	00:30:52.527	Fedo Market Thre...	Urun Ekle	3	✓	224	245	3	1
12	00:30:52.531	Fedo Market Thre...	Urun Ekle	3	✓	224	245	3	0
13	00:30:52.534	Fedo Market Thre...	Urun Ekle	4	✓	224	245	3	0
14	00:30:52.538	Fedo Market Thre...	Urun Ekle	2	✓	224	245	2	0
15	00:30:52.541	Fedo Market Thre...	Urun Ekle	3	✓	224	245	3	0

- Raporumuzda görüldüğü üzere kodumuz herhangi bir hata vermeden sıkıntısız çalışmakta.



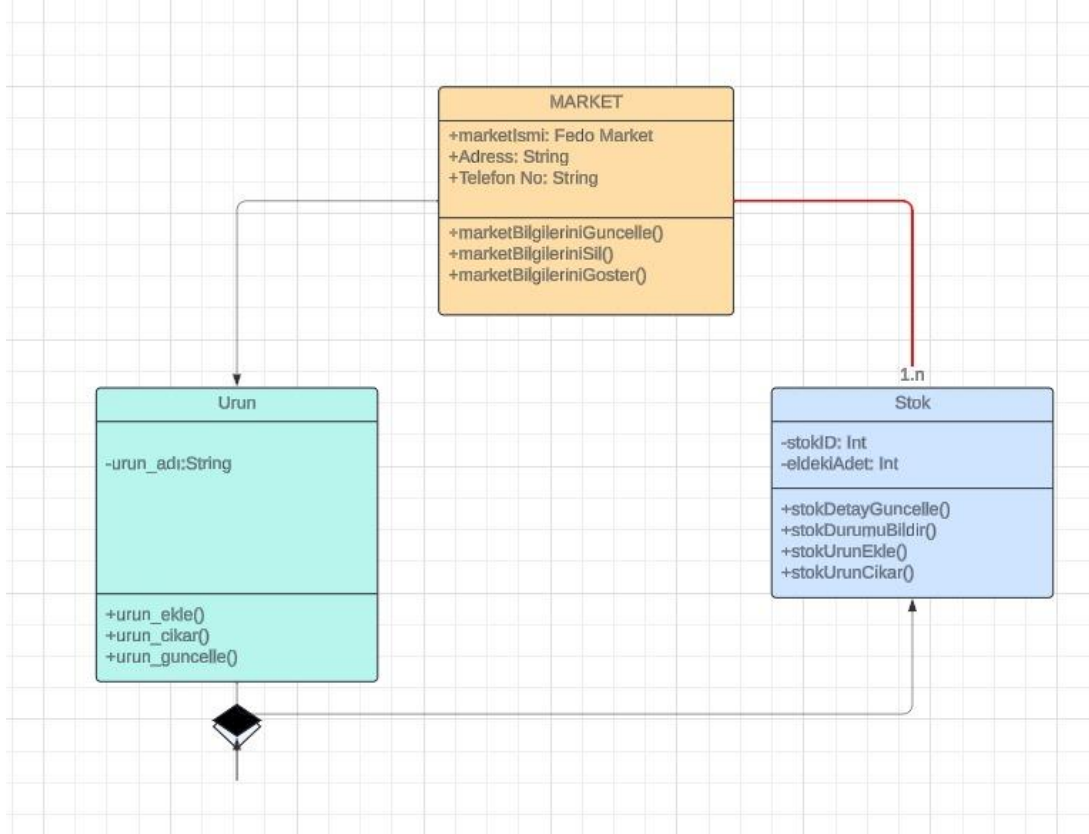
- Sonuç Gösterge Ağacında ürünlerimizin hata vermeden eklendiğini görmekteyiz



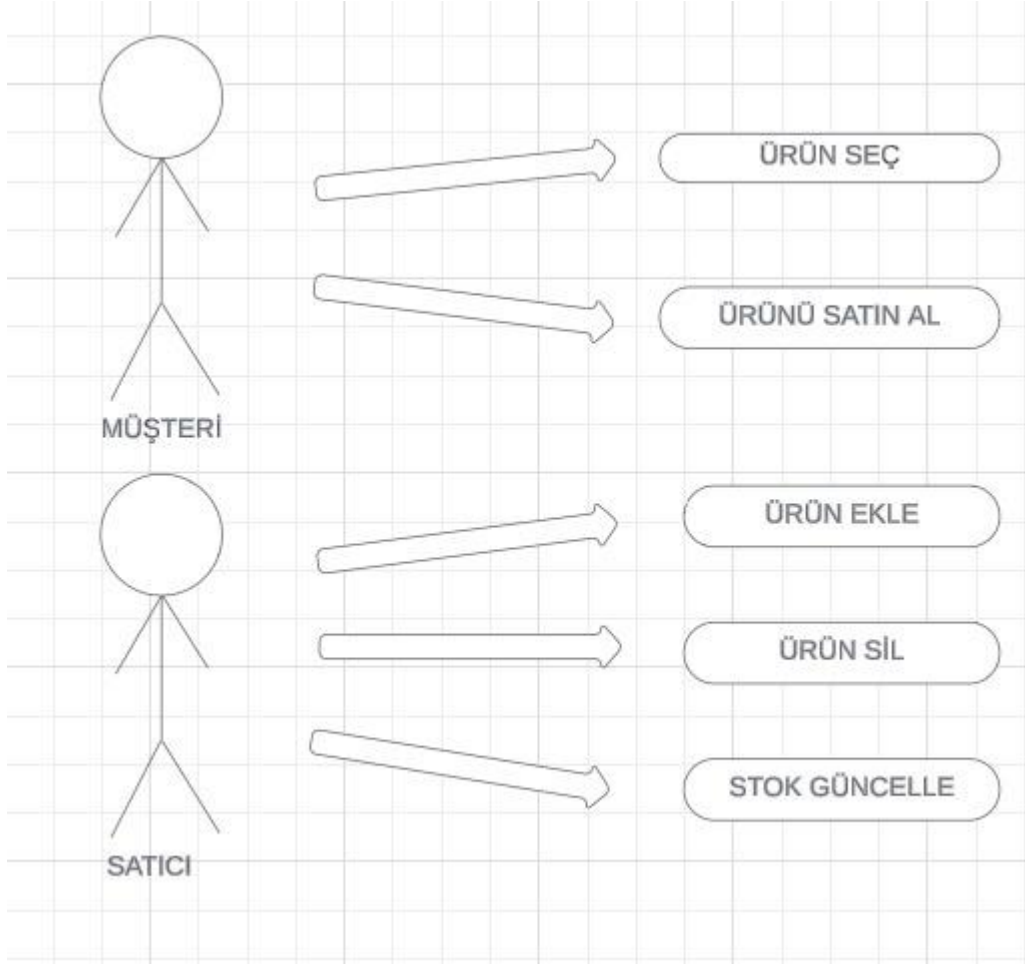
Etiket	# Örnek	Ortalama	En Az	En Çok	Std. Dev.	Hata %	Transfer Oranı	KB/sn	Send KB/sec	Ort. Byte
Urun Ekle	50	1	0	10	1,47	0,00%	62,0/sec	12,89	13,01	213,0
TOPLAM	50	1	0	10	1,47	0,00%	62,0/sec	12,89	13,01	213,0

- Özet Raporda ise %0.00 hata ile döngümüzde girdiğimiz 50 adedin hatasız eklendiğini görmekteyiz.

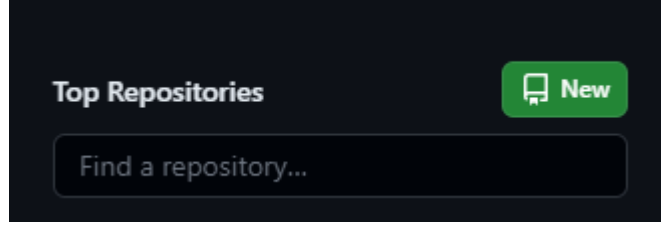
FEDO MARKET SINIF DİYAGRAMI



FEDO MARKET USE-CASE DİYAGRAMI




**MAVEN PROJEMİZİ ECLİPSE ÜZERİNDEN
GİTHUB' A EKLEME**



- Yeni bir repostory oluřturmak için new butonuna tıkladım.

Required fields are marked with an asterisk ().*

Owner * **Repository name ***

 Feder4l / yazilimgelistirmebutproje

✓ yazilimgelistirmebutproje is available.

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

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.

Initialize this repository with:

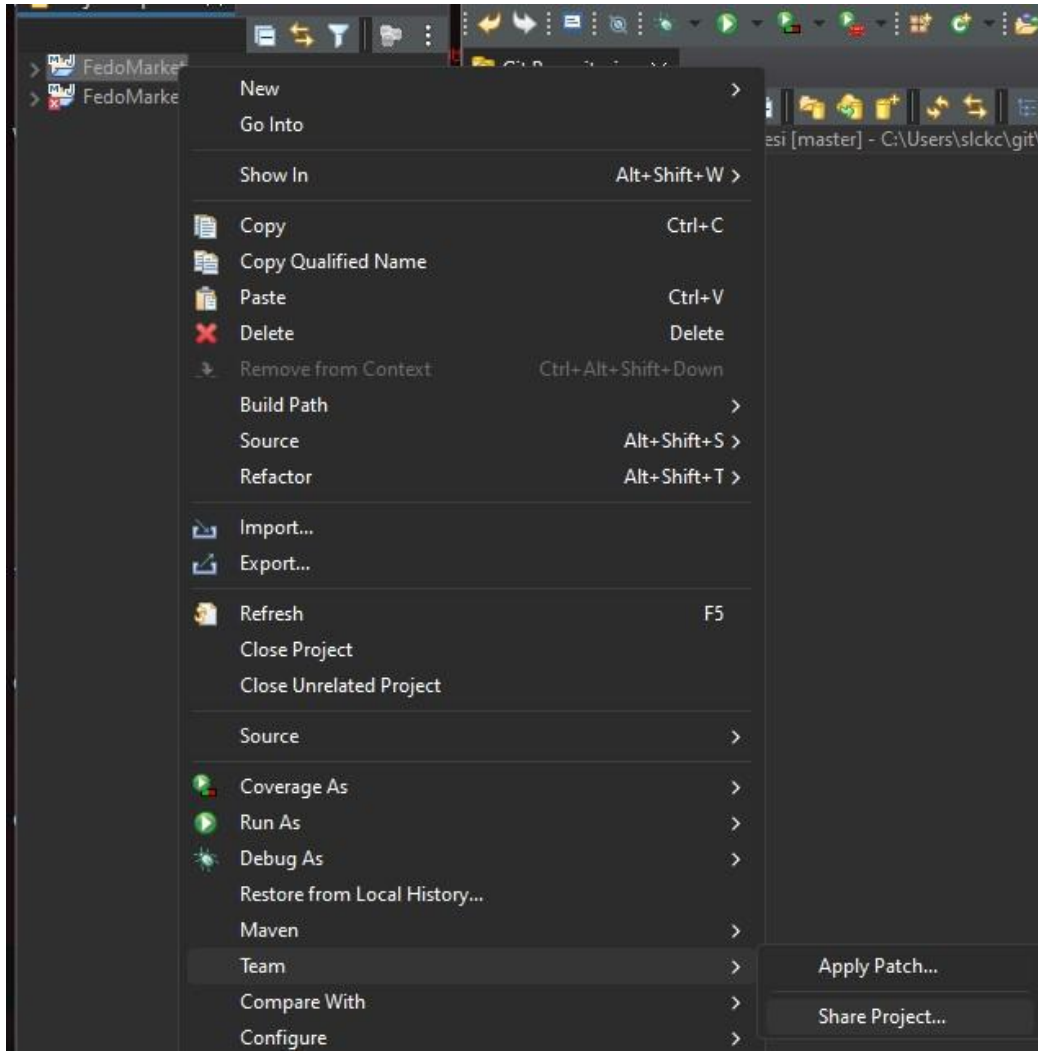
☐ **Add a README file**
This is where you can write a long description for your project. [Learn more about READMEs.](#)

Add .gitignore

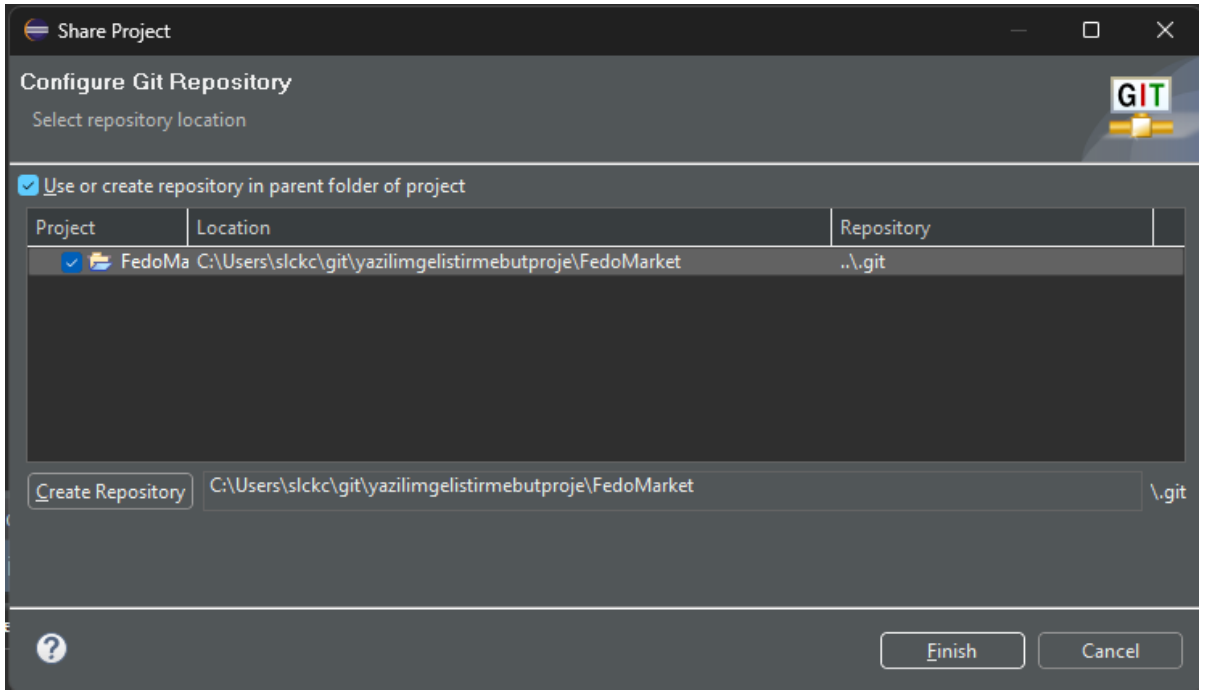
.gitignore template: **None**

Choose which files not to track from a list of templates. [Learn more about ignoring files.](#)

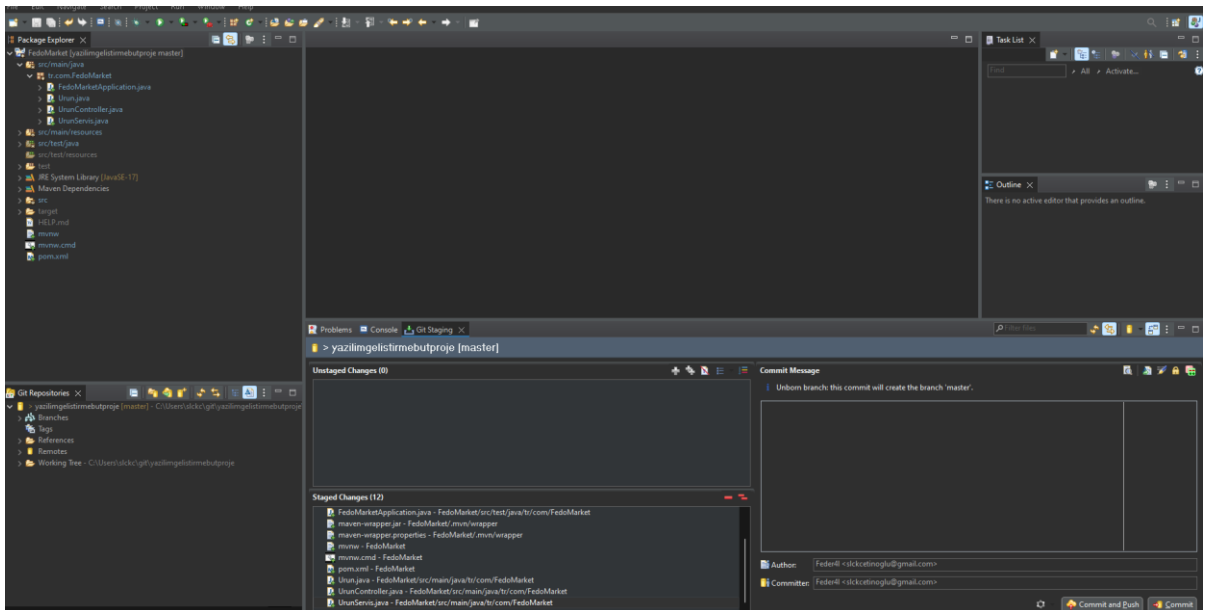
- Repostory ismini ve private özelliğini seçerek creat butınuna tıklayıp oluřturdum



- Eclipse' e girip Project Explorer üzerinden Market projemize sağ clik'leyip "Team" seçeneği üzerinden "Share Project" i seçiyoruz.



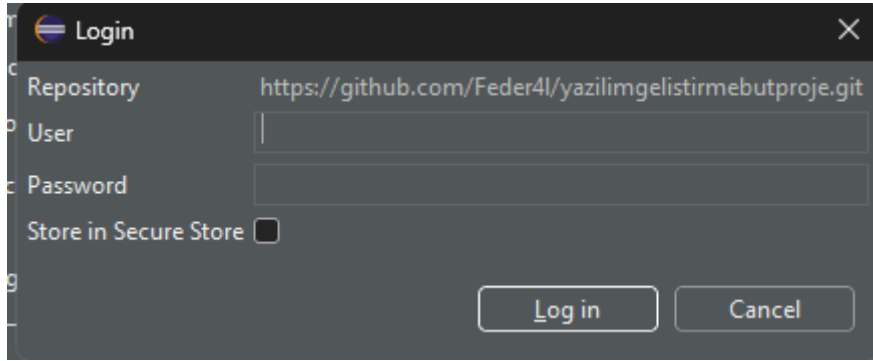
- Açılan pencerede yükleme yapmak istediğimiz Repository i seçiyoruz ve göndermek istediğimiz dosyayı click liyoruz.



- Açılan sekmede yüklemek istediğimiz dosyaları “Unstaged Changes” sekmesinden “Staged

Changes” sekmesine taşıyoruz ve ardından sağ alt kısımda bulunan “Commit and Push” ibaresine tıklıyoruz.

-



- Açılan pencereye Github kullanıcı adımızı ve Github içerisinden aldığımız Token’i yazıyoruz ve “Log in” butonuna basıyoruz . Daha sonra Finish butonuna basıp yükleme işlemini gerçekleştirmiş oluyoruz.

SELÇUK GÜNDÜÇ

H5230086

YAZILIM GELİŞTİRME VE ORTAM ARAÇLARI

DERSİ BÜT PROJESİ

