

MEDİPOL ÜNİVERSİTESİ

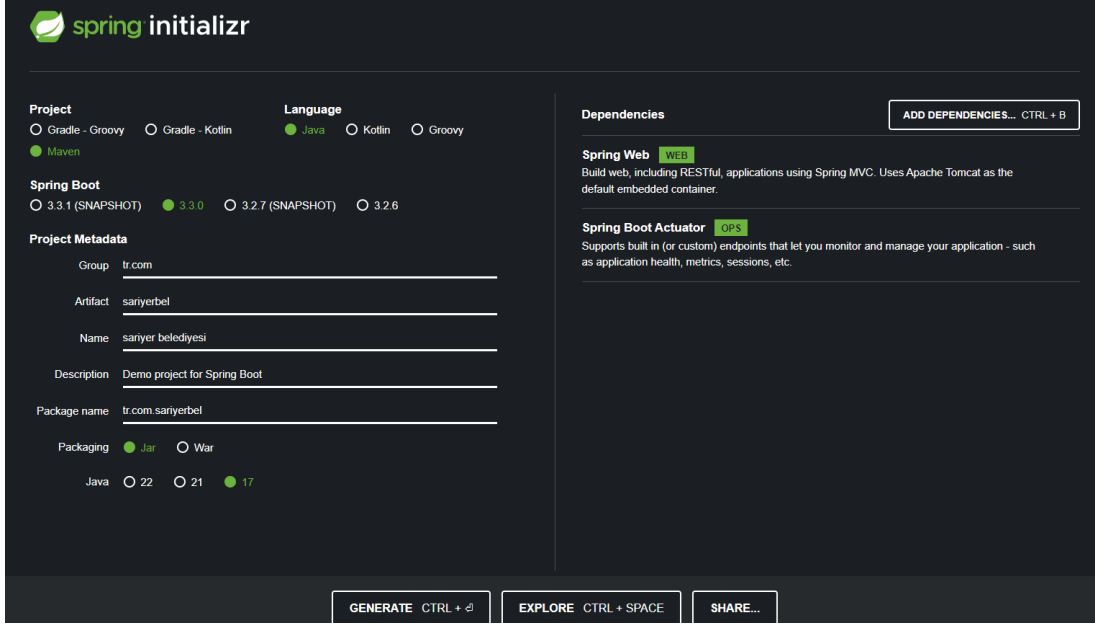
YAZILIM GELİŞTİRME VE ORTAM ARAÇLARI DERSİ

FİNAL PROJESİ

Selçuk GÜNDÜÇ
H5230086

YAZILIM GELİŞTİRME VE ORTAM ARAÇLARI DERSİ FİNAL PROJESİ

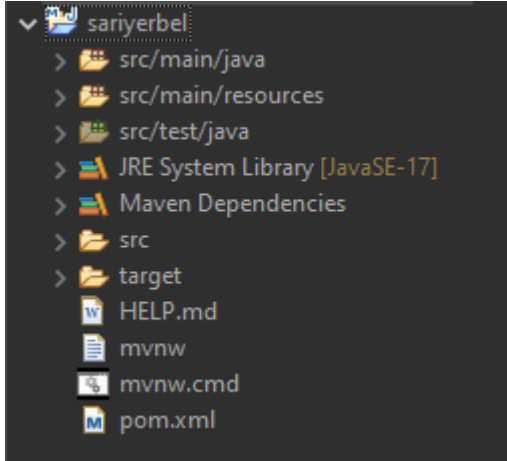
SARIYER BELEDİYESİ WEB SERVİS PROJESİ



The image shows the Spring Initializr web form. It is a dark-themed interface with a green Spring logo and the text 'spring initializr'. The form is divided into several sections: 'Project' with radio buttons for 'Gradle - Groovy' (selected), 'Gradle - Kotlin', and 'Maven'; 'Language' with radio buttons for 'Java' (selected), 'Kotlin', and 'Groovy'; 'Spring Boot' with radio buttons for '3.3.1 (SNAPSHOT)', '3.3.0' (selected), '3.2.7 (SNAPSHOT)', and '3.2.6'; 'Project Metadata' with input fields for 'Group' (tr.com), 'Artifact' (sariyerbel), 'Name' (sariyer belediyesi), 'Description' (Demo project for Spring Boot), and 'Package name' (tr.com.sariyerbel); 'Packaging' with radio buttons for 'Jar' (selected) and 'War'; and 'Java' with radio buttons for '22', '21', and '17' (selected). On the right, there is a 'Dependencies' section with a button 'ADD DEPENDENCIES... CTRL + B'. Below this, there are two dependency cards: 'Spring Web' with a 'WEB' tag and a description, and 'Spring Boot Actuator' with an 'OPS' tag and a description. At the bottom, there are three buttons: 'GENERATE CTRL + G', 'EXPLORE CTRL + SPACE', and 'SHARE...'.

SPRING BOT SİTESİNDEN PROJE OLUŞTURMA

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



Masaüstüne ayıkladıktan sonra ECLİPSE İDE ye import ettim. Yukarıdaki görüntü ile karşılaştıysam problem yok demektir. Kodlarımı yazmaya başlayabilirim demektir.

SariyerBelediyesiApplication.java sayfası kodu ve açıklamaları

```

1 package tr.com.sariyerbel;
2
3 import org.springframework.boot.SpringApplication;
4
5
6 @SpringBootApplication
7 public class SariyerBelediyesiApplication {
8
9     public static void main(String[] args) {
10         SpringApplication.run(SariyerBelediyesiApplication.class, args);
11     }
12
13 }
14

```

- + @SpringBootApplication: Bu sınıfın bir Spring Boot uygulaması olduğunu belirtir.
- + public class SariyerBelediyesiApplication: Ana uygulama sınıfı.
- + public static void main(String[] args): Uygulamanın başlangıç noktası.
- + SpringApplication.run(SariyerBelediyesiApplication.class, args): Spring Boot uygulamasını başlatır.

GorevEkle.java sayfası kodu ve açıklamaları

```

1 package tr.com.sariyerbel;
2
3 import org.springframework.web.bind.annotation.*;
4
5
6 @RestController
7 @RequestMapping("/gorev")
8 public class GorevEkle {
9
10     public record Gorev(String ad, String numara) {};
11
12     static final List<Gorev> GOREV_LISTESI = new ArrayList<>();
13
14
15     @GetMapping("/")
16     public List<Gorev> listele() {
17         return GOREV_LISTESI;
18     }
19
20     @GetMapping("/{no}")
21     public Gorev bul(@PathVariable String no) {
22         for(Gorev gorev: GOREV_LISTESI) {
23             if(gorev.numara().equals(no)) {
24                 return gorev;
25             }
26         }
27         return null;
28     }
29
30
31     @PostMapping("/")
32     public Gorev ekle(@RequestBody Gorev gorev) {
33         GOREV_LISTESI.add(gorev);
34         return gorev;
35     }
36 }

```

- @RestController: Bu sınıfın bir REST denetleyicisi olduğunu belirtir.

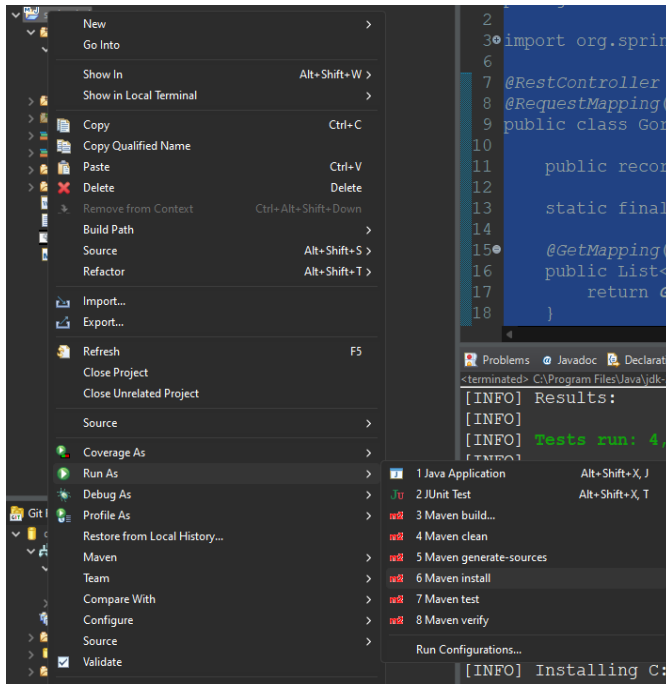
- `@RequestMapping("/gorev")`: Bu denetleyicinin /gorev yolunda erişilebilir olduğunu belirtir.
- `public record Gorev(String ad, String numara) {};`: Bir kayıt (record) tipi olan Gorev sınıfı, ad ve numara alanları içerir.
- `static final List<Gorev> GOREV_LISTESI = new ArrayList<>();`: Bir liste oluşturarak görevlerin depolanacağı yer hazırlanır.

Tanımladığım HTTP istekleri ve açıklamaları:

(EKLEME,BULMA,LİSTELEME)

- `@GetMapping("/")`: Tüm görevleri listelemek için kullanılır.
- `@GetMapping("/{no}")`: Belirli bir görevi bulmak için kullanılır, görev numarasıyla çalışır.
- `@PostMapping("/")`: Yeni bir görev eklemek için kullanılır, istek gövdesinden (`@RequestBody`) gelen verilere dayanarak bir görev ekler.

MAVEN İNSTALL TEST



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

```
[INFO] Installing C:\Users\selcu\Desktop\sariyerbel\s
[INFO] Installing C:\Users\selcu\Desktop\sariyerbel\s
[INFO] -----
[INFO] BUILD SUCCESS
[INFO] -----
[INFO] Total time: 5.903 s
[INFO] Finished at: 2024-06-12T22:11:58+03:00
[INFO] -----
```

BUILD SUCCESS bu yazıyı gördüysek kodumuzda hata yok demektir.

JUNIT VE COVERAGE TESTLER

The screenshot shows an IDE with a Java file containing a JUnit test. The test class is `GorevEkleTest` and it contains a test method `testBul()`. The test method creates a `Gorev` object, adds it to a list, and then calls the `bul()` method. The test runner output shows that the test passed successfully.

```
30 List<Gorev> gorevList = gorevEkle.listele();
31 assertEquals(2, gorevList.size());
32 assertEquals(gorev1, gorevList.get(0));
33 assertEquals(gorev2, gorevList.get(1));
34 }
35
36 @Test
37 public void testBul() {
38     Gorev gorev1 = new Gorev("Gorev1", "1");
39     GorevEkle.GOREV_LISTESI.add(gorev1);
40
41     Gorev result = gorevEkle.bul("1");
42     assertEquals(gorev1, result);
43
44     Gorev resultNotFound = gorevEkle.bul("2");
45 }
```

JUnit Test Results:

Test Method	Duration (s)
testBul()	0.034
testEkle()	0.001
testListele()	0.001

The screenshot shows the Coverage tab in the IDE, displaying a table of coverage data for the project. The table includes columns for Element, Coverage, Covered Instructions, Missed Instructions, and Total Instructions.

Element	Coverage	Covered Instructions	Missed Instructions	Total Instructions
sariyerbel	92,4 %	146	12	158
src/main/java	84,9 %	45	8	53
tr.com.sariyerbel	84,9 %	45	8	53
SariyerBelediyesiApplication.java	0,0 %	0	8	8
GorevEkle.java	100,0 %	45	0	45
src/test/java	96,2 %	101	4	105
tr.com.sariyerbel	96,2 %	101	4	105
SariyerBelediyesiApplicationTests.java	0,0 %	0	4	4
GorevEkleTest.java	100,0 %	101	0	101

JUnit ve Covverage testlerimin sonuçlarını yukarıda görmekteyiz. Aşağıda ise bu testlerin HTML görüntüsünü görmekteyiz.

← → ↻ Dosya C:/Users/selcu/Desktop/sariyerbel/sariyerbel/target/site/jacoco/index.html

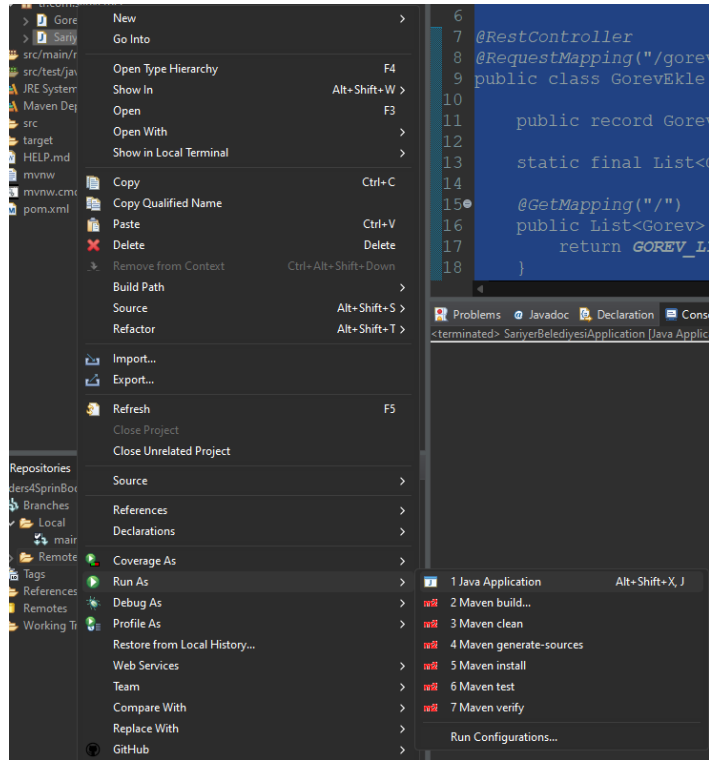
Gmail YouTube Haritalar DenizBank İnternet... ziraat E FATURA SORGULA... WhatsApp BTK Akademi

sariyer belediyesi

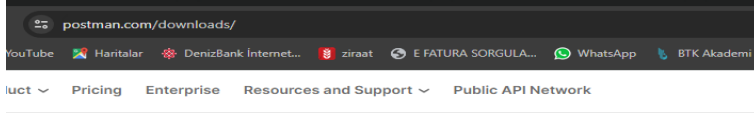
sariyer belediyesi

Element	Missed Instructions	Cov.	Missed Branches	Cov.	Missed	Cxty	Missed	Lines	Missed	Methods	Missed	Classes
tr.com.sariyerbel	<div><div></div></div>	%86	<div><div></div></div>	%100	2	12	2	13	2	10	0	3
Total	8 of 59	%86	0 of 4	%100	2	12	2	13	2	10	0	3

POSTMAN TEST



“src/main/java” klasörünün altında olan “tr.com.SariyerBelediyesiApplication.java” klasörünün içerisindeki “Application.java” dosyasına sağ click ile “Runn As” ardından “Java Application” seçeneğini seçerek programı java ile çalıştırıyoruz.



Bu siteden bilgisayarımıza uygun Postman sürümünü indirip çalıştırıyoruz.

Do

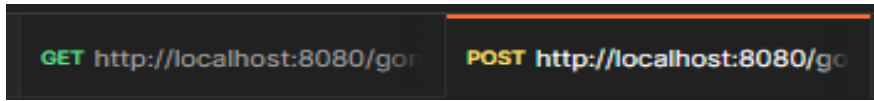
postman.com/downloads/

Download the app to get started with the Postman API Platform.

The Postman app

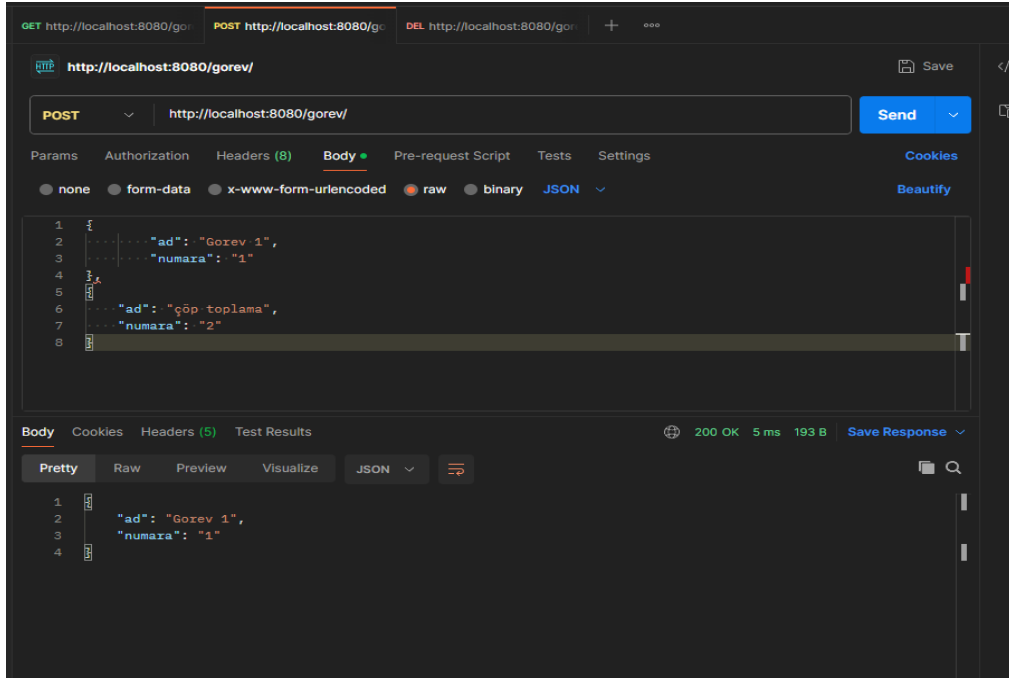
Download the app to get started with the Postman API Platform.

Windows 64-bit



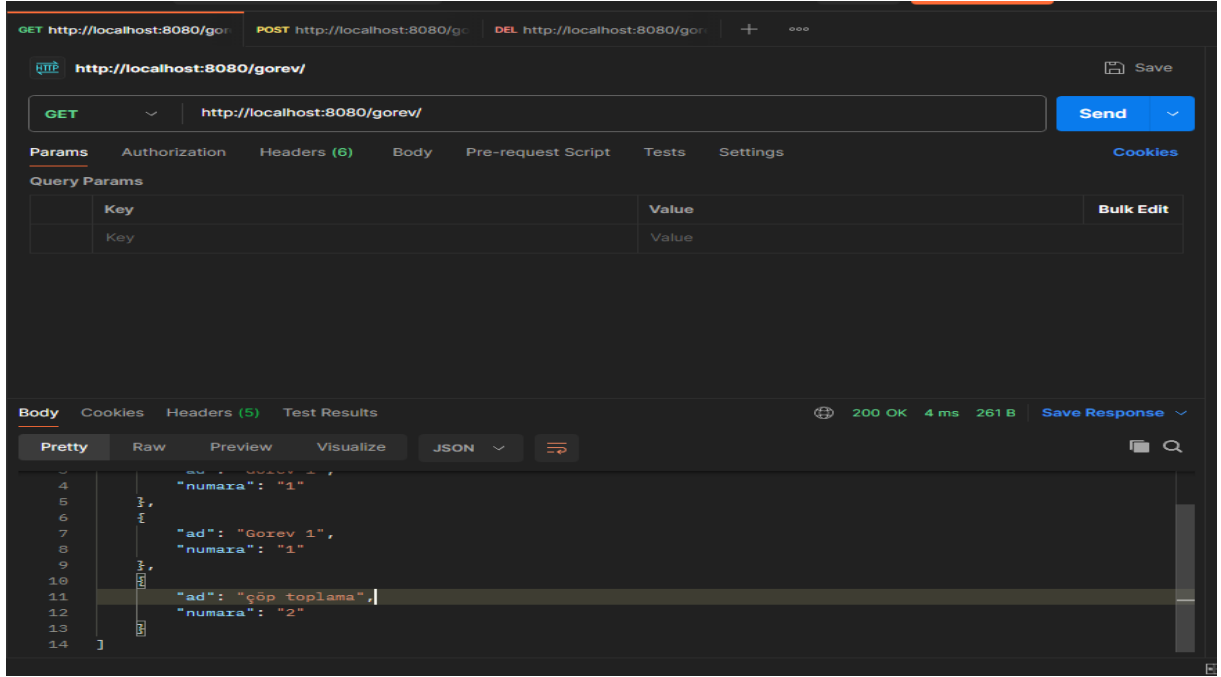
- New kısmından ÜRÜN EKLEME ve LİSTELEME olacak şekilde adet dosya “New Project” oluşturuyoruz.

POST (ÜRÜN EKLEME) SEKMEİ :



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

GET (ÜRÜN LİSTELEME) SEKMEİ :

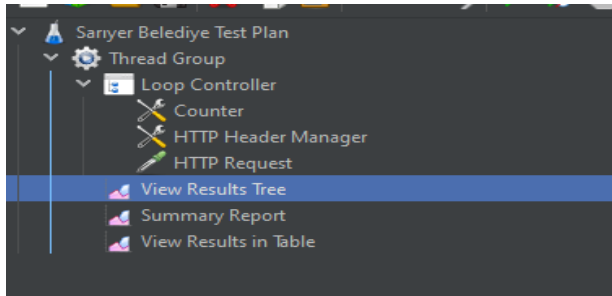


Bu sekmemiz ise “POST” sekmesinde girilen ürünleri listelememizi sağlar. GET butonunun yanında bulunan boşuğa Web Servis URL mizi girdikten sonra “Send” butonuna clickliyoruz. Ve POST sekmesinde eklenen tüm ürünler listelenmiş şekilde karşımıza çıkıyor.

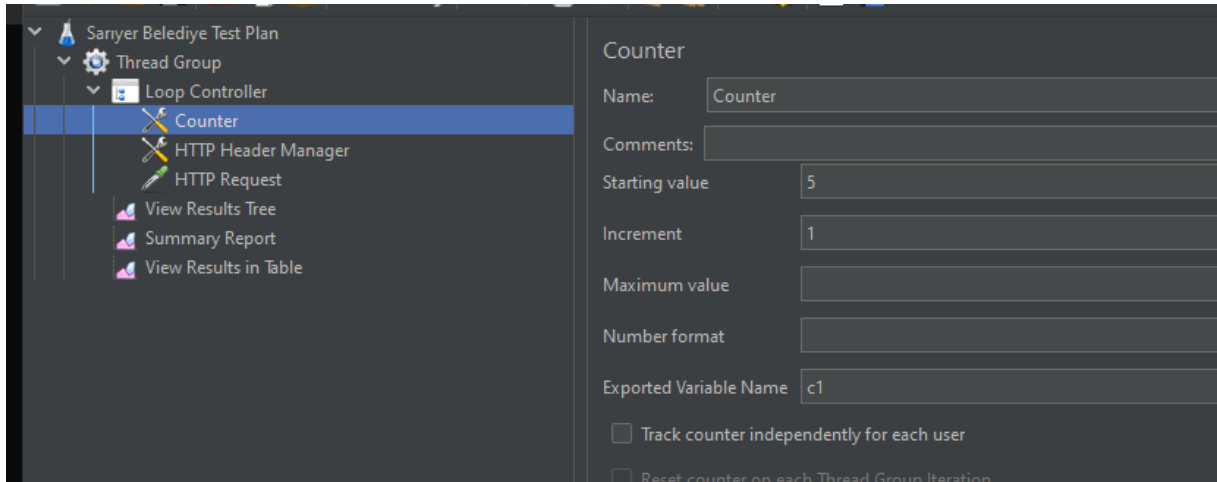
APACHE JMETER TEST

Program yük testimizi “Apache JMeter” uygulamasıyla yapacağız.

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

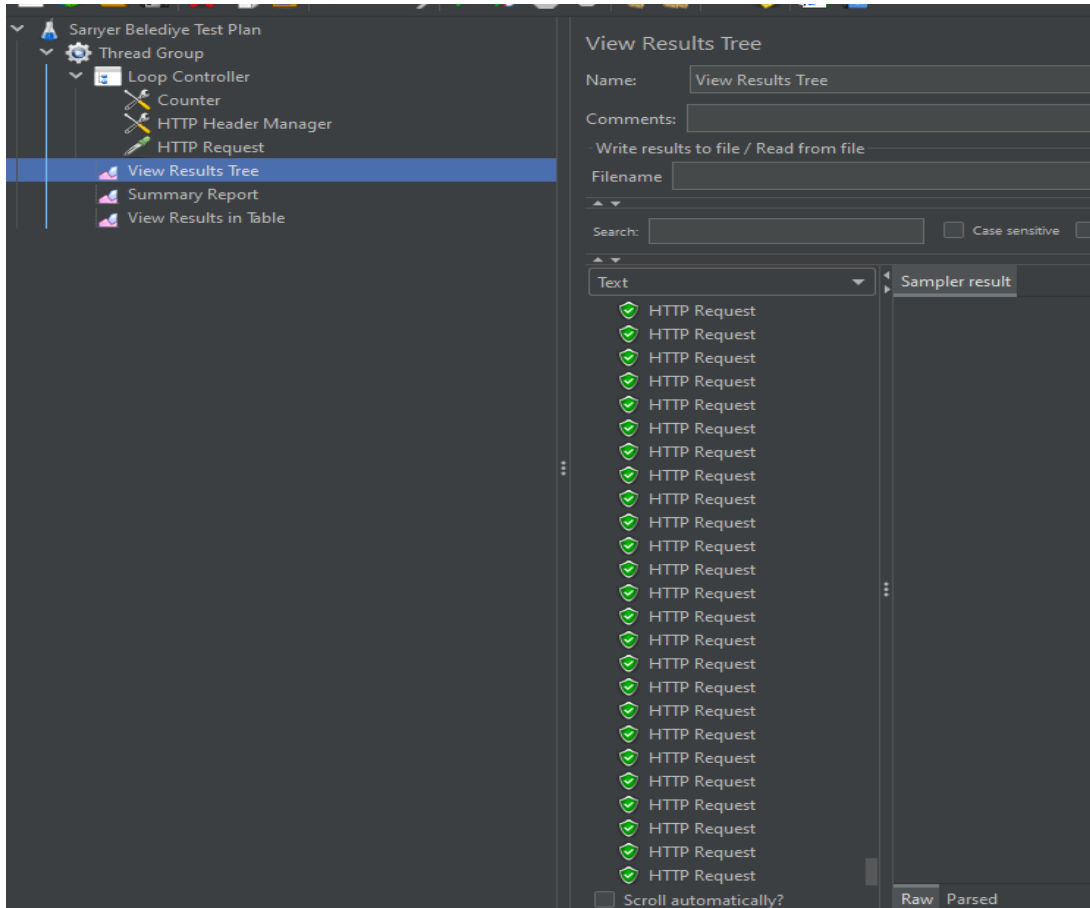


Test için gerekli ekleme ve rapor sekmelerimizi ekliyoruz.



Thread Group, Loop Controller ve Counter a test değerlerini girdikten sonra start veriyoruz.

Test Sonuçları



- Sonuç Gösterge Ağacında ürünlerimizin hata vermeden eklendiğini görmekteyiz

Summary Report

Name: Summary Report

Comments:

Write results to file / Read from file

Filename: Browse... Log/Display Only: ☐ Errors ☐ Successes

Label	# Samples	Average	Min	Max	Std. Dev.	Error %	Throughput	Received KB/sec	Sent KB/sec	Avg. Bytes
HTTP Request	500	1	0	5	0.70	0.00%	504.0/sec	101.20	109.08	205.6
TOTAL	500	1	0	5	0.70	0.00%	504.0/sec	101.20	109.08	205.6

- Özet Raporda ise %0.00 hata ile döngümüzde girdiğimiz 500 adedin hatasız eklendiğini görmekteyiz.

View Results in Table

Name: View Results in Table

Comments:

Write results to file / Read from file

Filename: Browse... Log/Display Only: ☐ Errors ☐ Successes

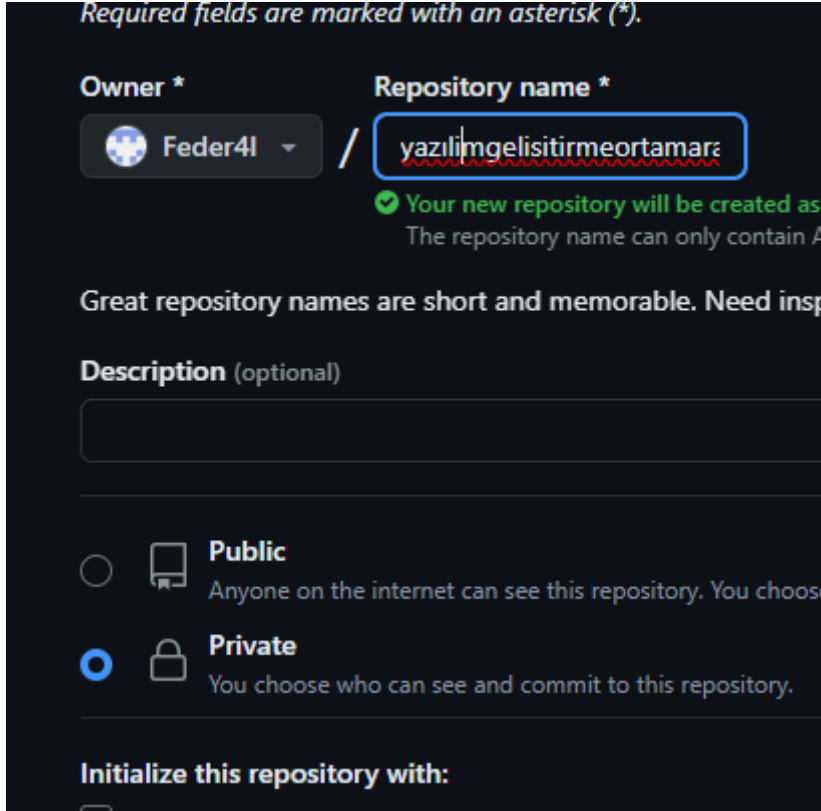
Sample #	Start Time	Thread Name	Label	Sample Time(ms)	Status	Bytes	Sent Bytes	Latency	Connect Time(m
1	21:03:05.492	Thread Group 1-1	HTTP Request	5	✓	202	218	5	
2	21:03:05.498	Thread Group 1-1	HTTP Request	3	✓	202	218	2	
3	21:03:05.501	Thread Group 1-1	HTTP Request	2	✓	202	218	2	
4	21:03:05.503	Thread Group 1-1	HTTP Request	2	✓	202	218	1	
5	21:03:05.505	Thread Group 1-1	HTTP Request	1	✓	202	218	1	
6	21:03:05.506	Thread Group 1-1	HTTP Request	1	✓	204	220	1	
7	21:03:05.507	Thread Group 1-1	HTTP Request	1	✓	204	220	1	
8	21:03:05.509	Thread Group 1-1	HTTP Request	1	✓	204	220	1	
9	21:03:05.510	Thread Group 1-1	HTTP Request	1	✓	204	220	1	
10	21:03:05.511	Thread Group 1-1	HTTP Request	1	✓	204	220	1	
11	21:03:05.511	Thread Group 1-2	HTTP Request	2	✓	204	220	1	
12	21:03:05.513	Thread Group 1-2	HTTP Request	1	✓	204	220	1	
13	21:03:05.515	Thread Group 1-2	HTTP Request	1	✓	204	220	1	
14	21:03:05.517	Thread Group 1-2	HTTP Request	1	✓	204	220	1	
15	21:03:05.519	Thread Group 1-2	HTTP Request	1	✓	204	220	1	
16	21:03:05.520	Thread Group 1-2	HTTP Request	1	✓	204	220	1	
17	21:03:05.521	Thread Group 1-2	HTTP Request	2	✓	204	220	2	

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

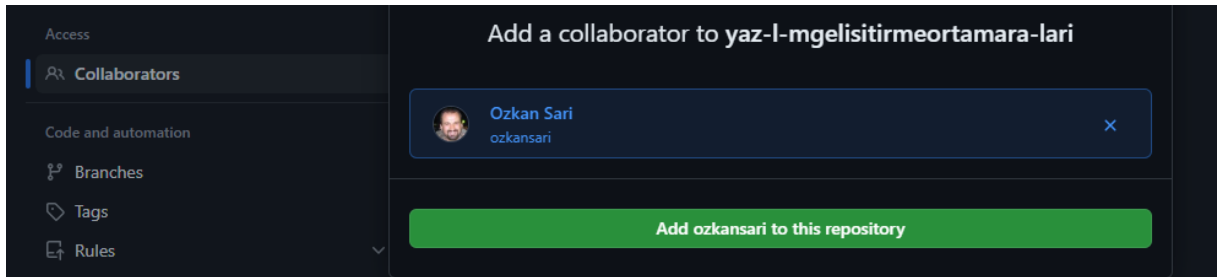
MAVEN PROJEMİZİ ECLİPSE ÜZERİNDEN GİTHUB' A EKLEME GİT KOMUTLARIYLA DOSYA YÜKLEME



New butonuna tıklayıp yeni bi repository oluşturuyoruz.

A screenshot of the GitHub 'Create new repository' form. The 'Owner' is set to 'Feder4l'. The 'Repository name' is 'yazilimgelisitirmeortamara-lari', which is highlighted with a red dashed border. A green checkmark indicates the repository will be created as public. The 'Description' field is empty. The 'Public' radio button is selected. The 'Initialize this repository with:' section is partially visible at the bottom.

Repositoryye isim verip Private seçeneğini seçiyoruz ve create repository butonuna basıyoruz.



Ayarlardan Collobrators sekmesinden projeyi sunucağımız kişiyi seçiyoruz.

```
MINGW64/c/Users/selcu/Desktop/yazılımgelistirmeortamaraçlarifinal
selcu@Federal MINGW64 ~/Desktop/yazılımgelistirmeortamaraçlarifinal (master)
$ git init
initialized empty Git repository in C:/Users/selcu/Desktop/yazılımgelistirmeortamaraçlarifinal/.git/
selcu@Federal MINGW64 ~/Desktop/yazılımgelistirmeortamaraçlarifinal (master)
|
```

Ardından masaüstünde projemisin bulunduğu klasörün içine git komu dizinini açıp git init komutunu yazıyoruz.

```
selcu@Federal MINGW64 ~/Desktop/yazılımgelistirmeortamaraçlarifinal (master)
$ git add *

selcu@Federal MINGW64 ~/Desktop/yazılımgelistirmeortamaraçlarifinal (master)
$ git status
On branch master

No commits yet

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

selcu@Federal MINGW64 ~/Desktop/yazılımgelistirmeortamaraçlarifinal (master)
$ git commit -m "Final Projesi Gönderiliyor"
[master (root-commit) 35eb824] Final Projesi Gönderiliyor
 1 file changed, 1 insertion(+)
 create mode 100644 README.md.txt

selcu@Federal MINGW64 ~/Desktop/yazılımgelistirmeortamaraçlarifinal (master)
$ |
```

Git add komutuyla dosyayı ekliyoruz , git statüs ile kontrol ediyoruz ve commit komutuyla gönderime hazır hale getiriyoruz.

```
selcu@Federal MINGW64 ~/Desktop/yazılımgelistirmeortamaraçlarifinal (master)
$ git remote
origin

selcu@Federal MINGW64 ~/Desktop/yazılımgelistirmeortamaraçlarifinal (master)
$ git push
fatal: The current branch master has no upstream branch.
To push the current branch and set the remote as upstream, use

    git push --set-upstream origin master

To have this happen automatically for branches without a tracking
upstream, see 'push.autoSetupRemote' in 'git help config'.

selcu@Federal MINGW64 ~/Desktop/yazılımgelistirmeortamaraçlarifinal (master)
$ git push --set-upstream origin master
Enumerating objects: 3, done.
Counting objects: 100% (3/3), done.
Writing objects: 100% (3/3), 267 bytes | 267.00 KiB/s, done.
Total 3 (delta 0), reused 0 (delta 0), pack-reused 0 (from 0)
To https://github.com/Feder41/yaz-1-mgelisitirmeortamara-lari.git
 * [new branch]      master -> master
```

Git remote ile oluşturduğumuz repository yi ranumlıyoruz ve git push komutuyla gönderiyoruz.

Ardından ECLİPSE içindeki projeyi de göndermek için repositori linkimizi eclipse üzerinde bulunan git repositories bölümüne yapııştırıyoruz.

```
> > yaz-l-mgelisitimeortamara-lari [master] - C:\Users\selcu\git\yaz-l-mgelisitimeortamara-lari\git
```

Gerekli ayarlamalarımızı yaptıktan sonra gönderim mesajı yazıp sağ altta bulunan commit and push butonuna basıp gönderiyoruz.

