



Emircan KOÇ

HR210057

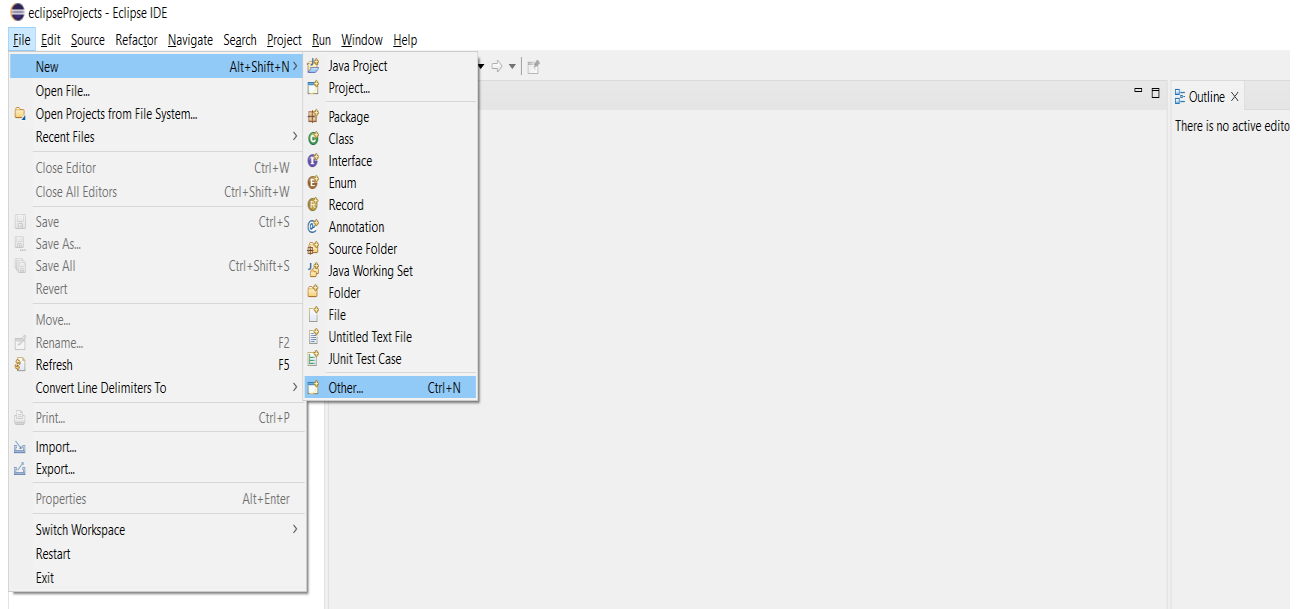
Yazılım Geliştirme Ortam ve
Araçları Final Raporu

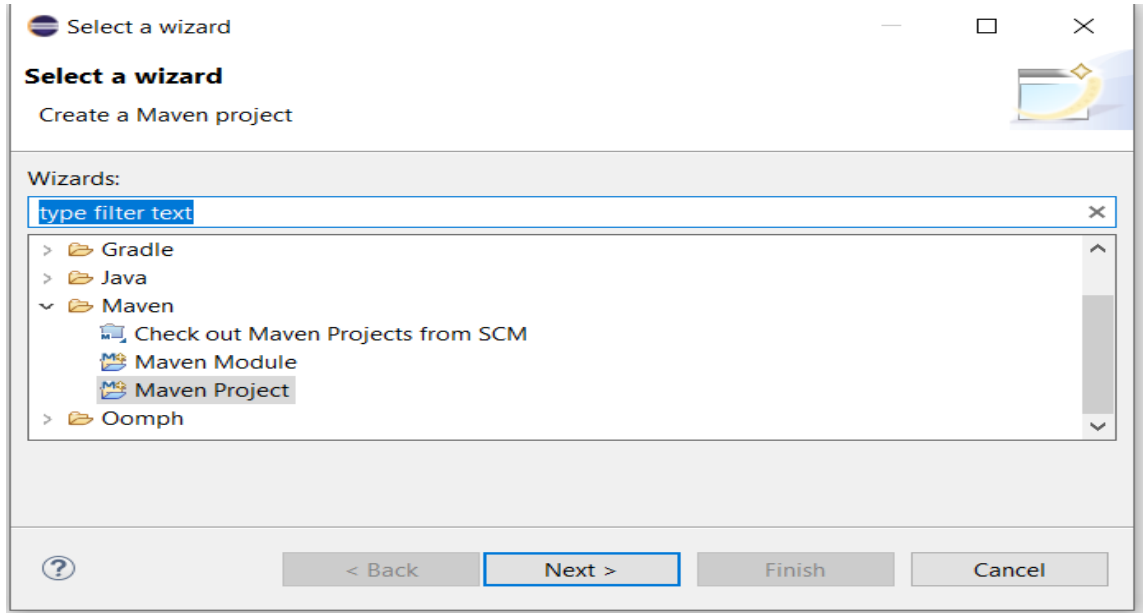
Özet

Proje sandık görevlerinin partilere ait oy bilgilerini değiştirmelerine olanak sağlayacak , görevliler partilerin oylarını arttırma , azaltma , gösterme ,listeleme gibi işlemler yapabilecekler.

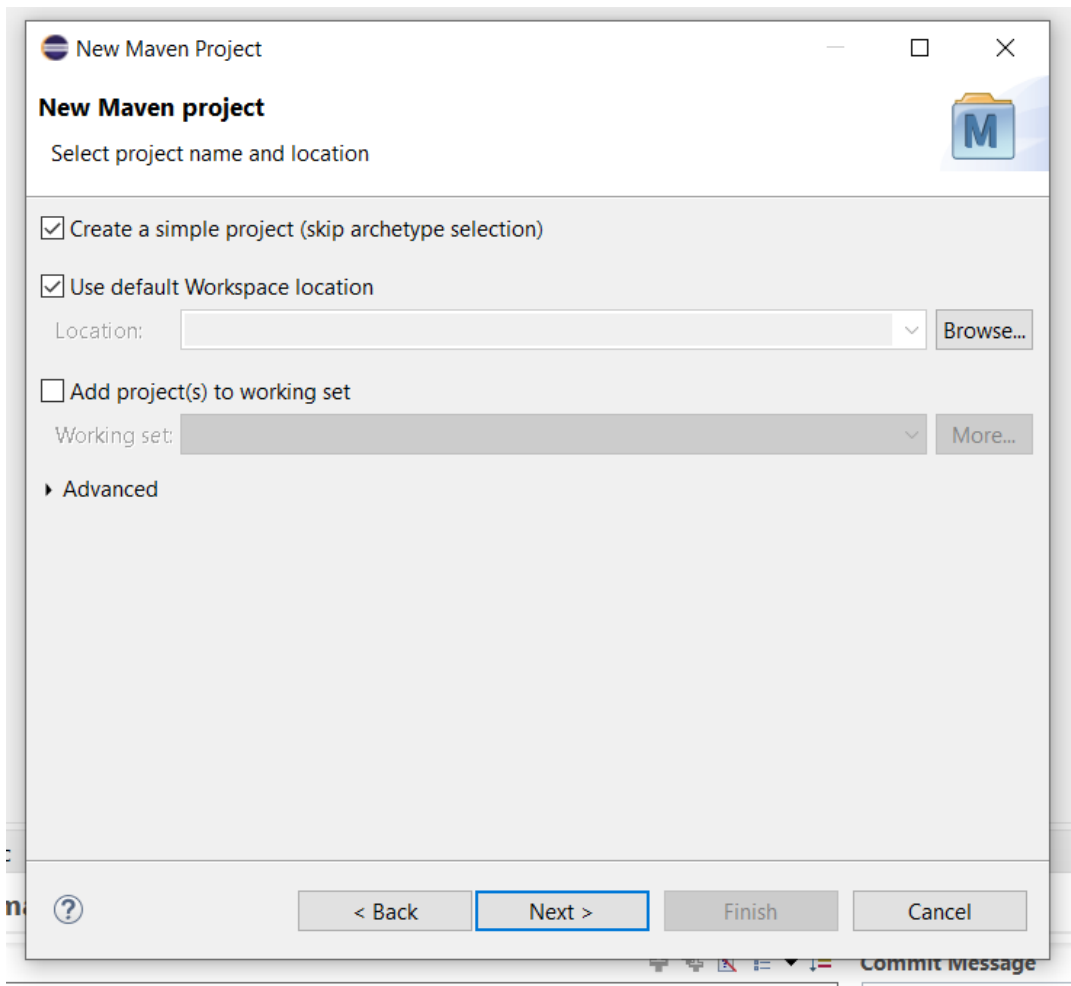
Kullandığım teknolojilerden bahsetmemiz gerekirse ArdahanSecimApi adındaki bu projede java sürümü olarak jdk 19 u tercih ettim. Bunda yeni olması ve düşük sürümleri çalıştırırken de bir problem olmamasından dolayı tercih ettim. IDE olarak eclipse ‘ i ve Dosya yönetim sistemi olarakta Maven i tercih ettim. Ayrıca JUnit ile tüm metotlar için gerekli birim tesleri yapıp , Jacoco ile de Code Coverage detayları hesaplayıp - inceleyip Github için de gerekli kısımda gösterdim. Bu kütüphaneleri kullanmak için gerekli dependency leri pom.xml içerisinde ekledim. Ayrıca compile olunacak sürümü ve target sürümü de belirttik. Burada belirlediğimiz sürümleri githubda gerekli ayarlamalar yaparken kullanacağız. Şimdi adım adım projenin yapımına geçelim .

Öncelikle IDE mizi açıyoruz , açıldıktan sonra sol üst kısmından File ‘ a tıklayıp new daha sonra Others... yı seçiyoruz.

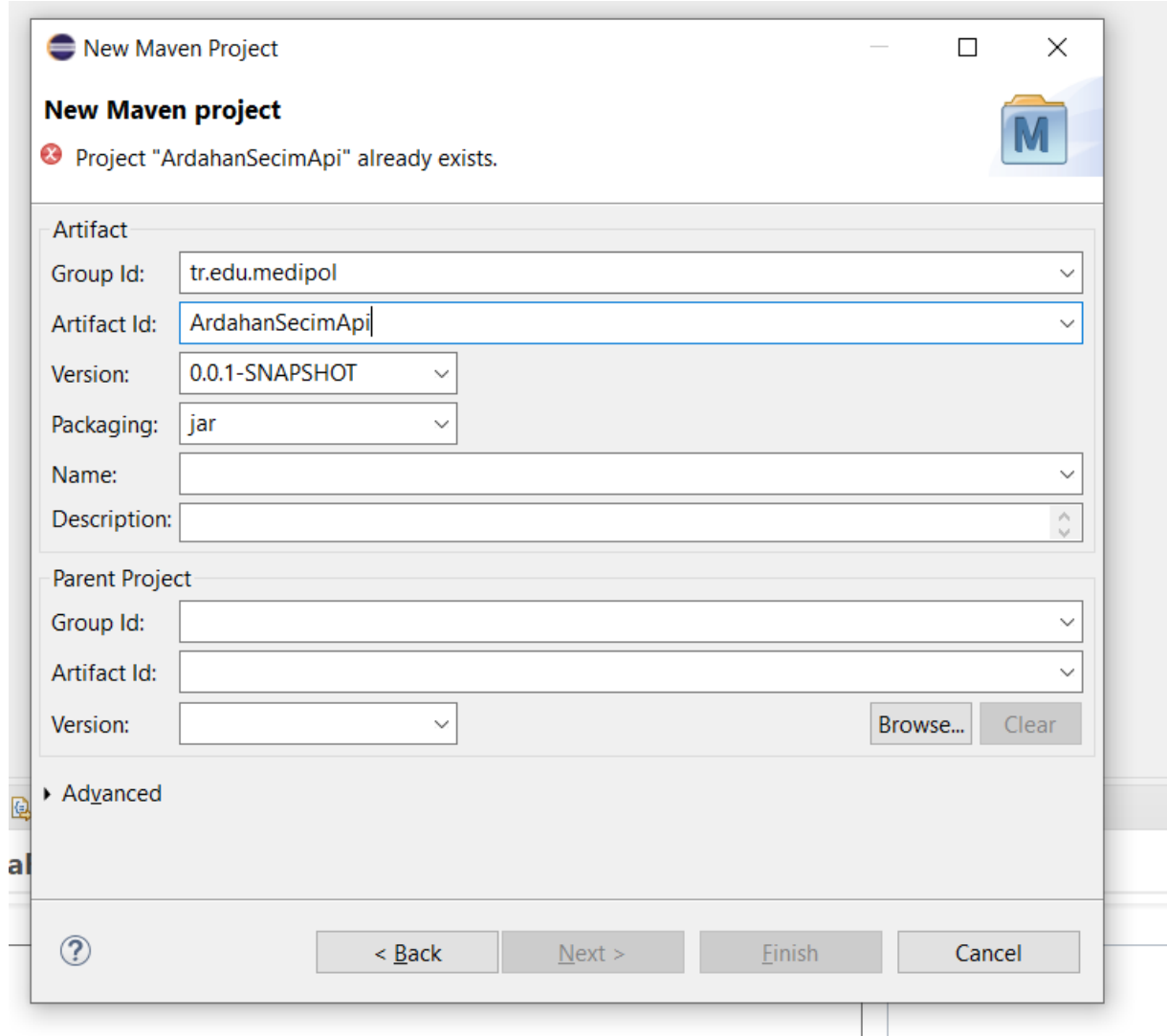




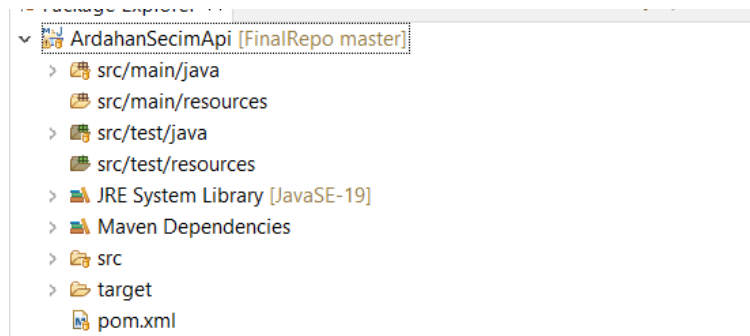
Tıkladığımız zaman açılan penceren Maven klasörüne tıklayıp , içerisinde Maven Project i seçiyoruz ve Next butonuna tıklıyoruz.



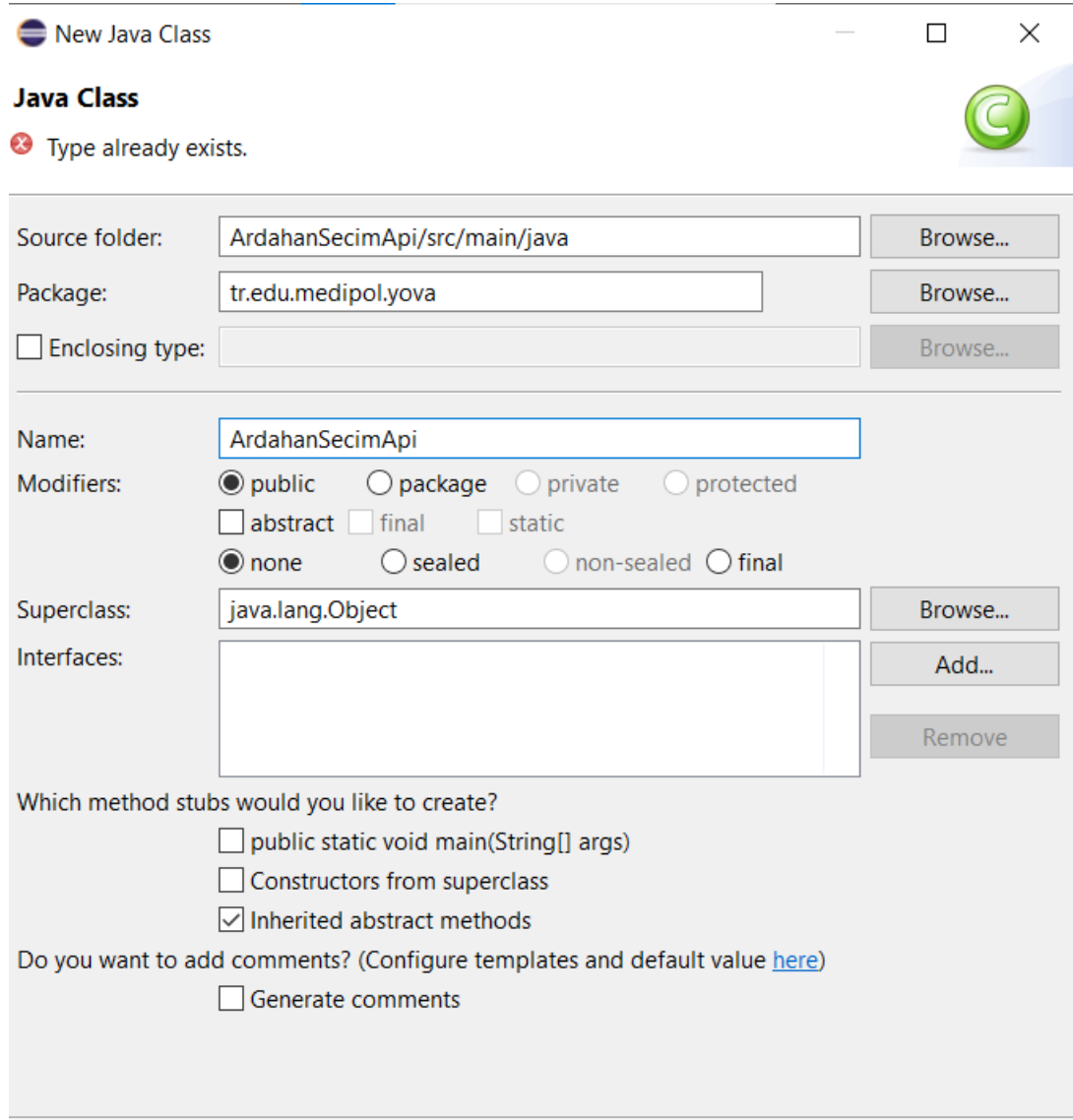
Tıkladıktan sonra açılan pencereden Create a simple project yazısına ait checkbox u işaretliyoruz. İşaretledikten sonra Next butonuna tıklıyoruz. Ardından 2 kez daha Next Next yapıp devam ediyoruz. Yaptıktan sonra karşımıza gelen pencereden Group Id ve Artifact Id kısımlarını dolduruyoruz ve Finish e basıyoruz. (Ben daha önce oluşturduğum için görselde finish kısmı kapalı görünüyor)



Butona tıkladıktan sonra Projemiz oluşmuş oluyor. Bundan sonraki adımda src/main/java ya sağ tıklayıp yeni bir class oluşturuyoruz. Package yi de bu sayede hızlıca oluşturuyoruz.



ArdahanSecimApi adında class ismimizi ve tr.edu.medipol.yova adında paket ismimizi girdikten sonra butona tıklıyoruz.



New Java Class

Java Class

Type already exists.

Source folder: Browse...

Package: Browse...

☐ Enclosing type: Browse...

Name:

Modifiers: ☒ public ☐ package ☐ private ☐ protected
☐ abstract ☐ final ☐ static
☒ none ☐ sealed ☐ non-sealed ☐ final

Superclass: Browse...

Interfaces: Add... Remove

Which method stubs would you like to create?

☐ public static void main(String[] args)

☐ Constructors from superclass

☒ Inherited abstract methods

Do you want to add comments? (Configure templates and default value [here](#))

☐ Generate comments

Şimdi gerekli diğer classlar ' ımızı ve interface ' mizi de oluşturuyoruz.

IArdahanSecimApi adında bir interfacemiz olacak ve metotları buraya yazıp , ilgili classa implemente edeceğiz. Interfacemiz bu şekilde :

```
1 package tr.edu.medipol.yova;
2
3 import java.util.ArrayList;
4
5 public interface IArdahanSecimApi {
6
7
8     public boolean partiOyArttir(String partiAdi);
9
10    public boolean partiOyAzalt(String partiAdi) ;
11
12    public int partiOyGetir(String partiAdi) ;
13
14    public ArrayList<Parti> partiOyListele();
15 }
16
```

Parti adında classımız şu şekilde :

```
1 package tr.edu.medipol.yova;
2
3 public class Parti {
4
5     public String partiAdi = "" ;
6     public int partiOySayisi = 0;
7
8
9 }
10
11
12
```

Gerekli oy işlemlerini ve parti işlemlerini bu class ile yapacağız.

Son olarak ArdahanSecimApi Adındaki classımızı oluşturuyoruz. Gerekli tüm metotları ve değişkenleri bu class içerisinde tanımlacağız.

```
1 package tr.edu.medipol.yova;
2
3 import java.util.ArrayList;
4
5 public class ArdahanSecimApi implements IArdahanSecimApi {
6
7     private Parti partiA ;
8     private Parti partiB ;
9     private Parti partiC ;
10
11+ public ArdahanSecimApi(){
12
13
14
15
16
17
18
19
20
21
22
23
24+ public boolean partiOyArttir(String partiAdi) {
25
26
27
28
29
30
31
32
33
34
35
36
37
38
39
40
41
42
43
44
45
46
47
48+ public boolean partiOyAzalt(String partiAdi) {
49
50
51
52
53
54
55
56
57
58
59
60
61
62
63
64
65
66
67
68
69
70
71
72
73
74
75
76+ public int partiOyGetir(String partiAdi) {
77
78
79
80
81
82
83
84
85
86
87
88
89
90
91
92
93
94
95+ public ArrayList<Parti> partiOyListele(){
96
97
98
99
100
101
102
103
104
105
106 }
107
```

İlgili metotların içlerini de doldurduktan sonra test aşamasına geçeceğiz ama bunun için önce pom.xml de gerekli ayarlamaları yapmalıyız.

Pom.xml e JUnit için gerekli dependenciesi ve jacoco plugini ekledikten sonra gerekli compiler ayarlarını da yapıyoruz. Hangi java sürümünde derleneceği , çalıştırılacağı gibi ayarlamaları da yaptık.

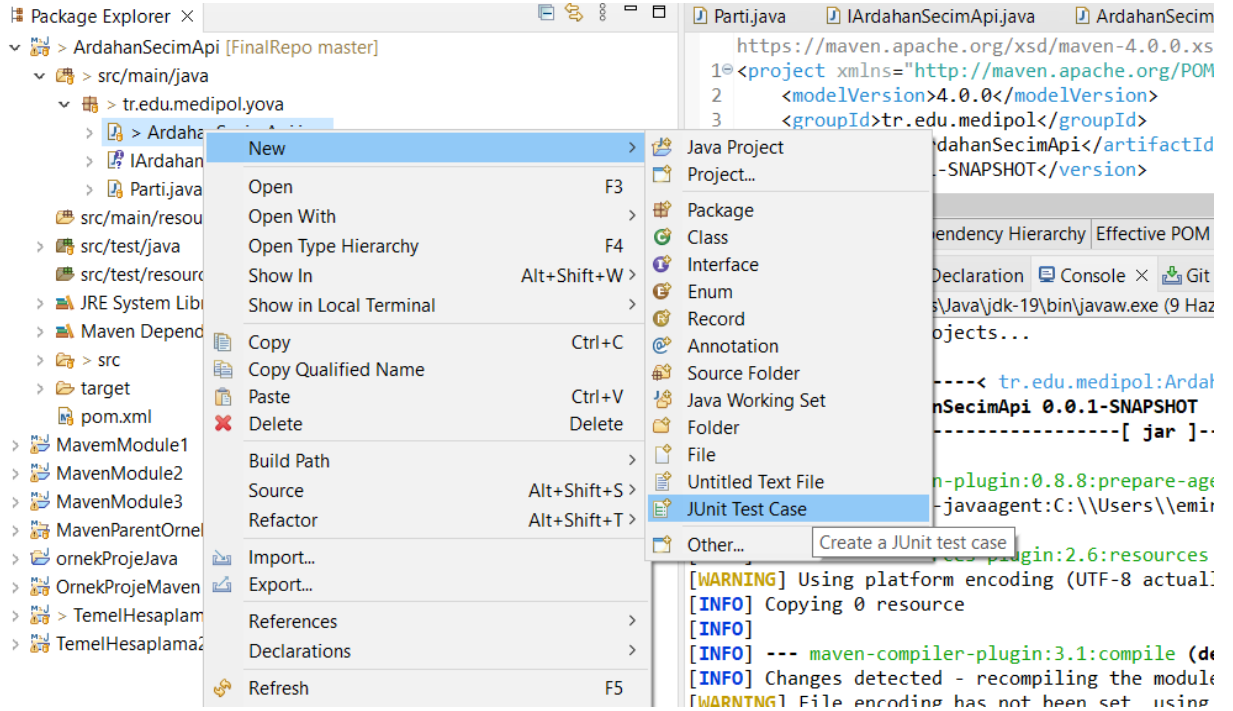
```
https://maven.apache.org/xsd/maven-4.0.0.xsd (xsi:schemaLocation)
1<project xmlns="http://maven.apache.org/POM/4.0.0" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
2  <modelVersion>4.0.0</modelVersion>
3  <groupId>tr.edu.medipol</groupId>
4  <artifactId>ArdahanSecimApi</artifactId>
5  <version>0.0.1-SNAPSHOT</version>
6  <properties>
7    <maven.compiler.source>19</maven.compiler.source>
8    <maven.compiler.target>19</maven.compiler.target>
9  </properties>
10 <dependencies>
11   <dependency>
12     <groupId>junit</groupId>
13     <artifactId>junit</artifactId>
14     <version>4.13.2</version>
15     <scope>test</scope>
16   </dependency>
17 </dependencies>
18 <build>
19   <plugins>
20     <plugin>
21       <groupId>org.jacoco</groupId>
22       <artifactId>jacoco-maven-plugin</artifactId>
23       <version>0.8.8</version>
24       <executions>
38     </plugin>
39   </plugins>
40 </build>
41 </project>
```

Bunları ekledikten sonra sağ tıklayarak Run as > maven install diyoruz ve gerekli bağımlılıkların ayarlanmasını-indirilmesini bekliyoruz.

```
[INFO] Scanning for projects...
[INFO]
[INFO] -----< tr.edu.medipol:ArdahanSecimApi >-----
[INFO] Building ArdahanSecimApi 0.0.1-SNAPSHOT
[INFO] -----[ jar ]-----
[INFO]
[INFO] --- jacoco-maven-plugin:0.8.8:prepare-agent (default) @ ArdahanSecimApi ---
[INFO] argLine set to -javaagent:C:\\Users\\emirc\\.m2\\repository\\org\\jacoco\\org.jacoco.agent\\0.8.8\\org.jacoco.agent-0.8.8-runtime.jar=destf
[INFO]
[INFO] --- maven-resources-plugin:2.6:resources (default-resources) @ ArdahanSecimApi ---
[WARNING] Using platform encoding (UTF-8 actually) to copy filtered resources, i.e. build is platform dependent!
[INFO] Copying 0 resource
[INFO]
[INFO] --- maven-compiler-plugin:3.1:compile (default-compile) @ ArdahanSecimApi ---
[INFO] Changes detected - recompiling the module!
[WARNING] File encoding has not been set, using platform encoding UTF-8, i.e. build is platform dependent!
[INFO] Compiling 3 source files to C:\\Users\\emirc\\git\\FinalRepo\\ArdahanSecimApi\\target\\classes
[INFO]
[INFO] --- maven-resources-plugin:2.6:testResources (default-testResources) @ ArdahanSecimApi ---
[WARNING] Using platform encoding (UTF-8 actually) to copy filtered resources, i.e. build is platform dependent!
[INFO] Copying 0 resource
[INFO]
[INFO] --- maven-compiler-plugin:3.1:testCompile (default-testCompile) @ ArdahanSecimApi ---
[INFO] Nothing to compile - all classes are up to date
[INFO]
[INFO] --- maven-surefire-plugin:2.12.4:test (default-test) @ ArdahanSecimApi ---
[INFO] Surefire report directory: C:\\Users\\emirc\\git\\FinalRepo\\ArdahanSecimApi\\target\\surefire-reports

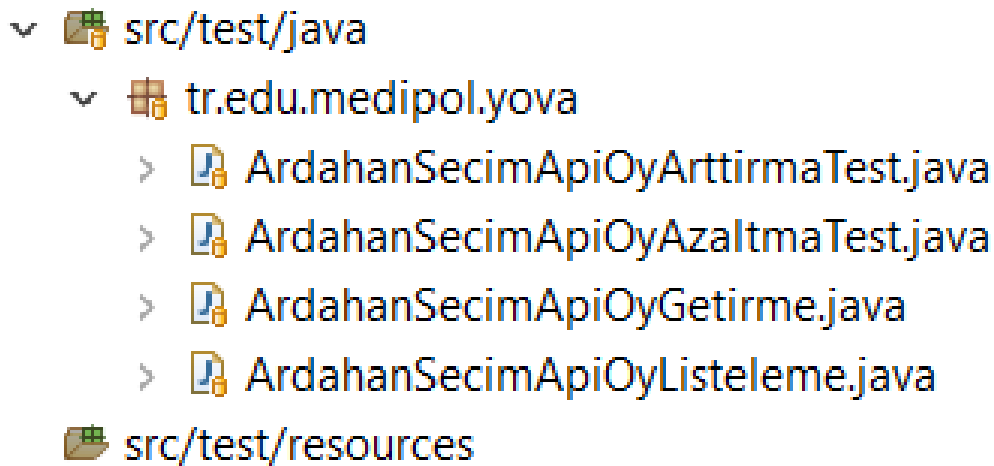
-----
T E S T S
-----
Running tr.edu.medipol.yuva.ArdahanSecimApiQuartTimeTest
```

Gerekli kısımların hallolduğunu gördük şimdi test yazımına başlayacağız. Ben ilgili classa sağ tıklayarak



Yolunu takip ediyorum. Test amaçlı class lar oluşturulduktan sonra içerini dolduruyoruz.

Test Classlarımızı resimde görülen dosya yolunun altına oluşturduk ve her metot için ayrı bir class oluşturduk bu sayede daha rahat yönetebileceğiz.



Oy arttırma işlemi için yazdığımız testler :

```
1 package tr.edu.medipol.yova;
2
3 import static org.junit.Assert.*;
4
5 import org.junit.Test;
6
7 public class ArdahanSecimApiOyArttirmaTest {
8
9
10 public void testAPartiOyArttirma() {
11
12
13
14
15
16
17
18
19
20
21
22
23 public void testBPartiOyArttirma() {
24
25
26
27
28
29
30
31
32
33
34
35
36
37
38 public void testCPartiOyArttirma() {
39
40
41
42
43
44
45
46
47
48
49
50
51
52
53 public void testOlmayanPartiOyArttirma() {
54
55
56
57
58
59
60
61
62
63
64
65
66
67
68 }
69 }
```

Oy azaltma işlemleri için yazdığımız testler :

```
1 package tr.edu.medipol.yova;
2
3 import static org.junit.Assert.*;
4
5
6
7 public class ArdahanSecimApiOyAzaltmaTest {
8
9
10 public void testAPartiOyAzaltmaSifirOy() {
11
12
13
14
15
16
17
18
19
20
21
22
23
24
25 public void testAPartiOyAzaltmaBirOy() {
26
27
28
29
30
31
32
33
34
35
36
37
38
39
40
41 public void testBPartiOyAzaltmaSifirOy() {
42
43
44
45
46
47
48
49
50
51
52
53
54
55
56 public void testBPartiOyAzaltmaBirOy() {
57
58
59
60
61
62
63
64
65
66
67
68
69
70
71
72 public void testCPartiOyAzaltmaSifirOy() {
73
74
75
76
77
78
79
80
81
82
83
84
85
86
87 public void testCPartiOyAzaltmaBirOy() {
88
89
90
91
92
93
94
95
96
97
98
99
100
101
102
103 public void testOlmayanPartiOyAzaltmaSifirOy() {
104
105
106
107
108
109
110
111
112
113
114
115
116
117
118 public void testOlmayanPartiOyAzaltmaBirOy() {
119
120
121
122
123
124
125
126
127
128
129
130
131
132
133
134 }
```

Oy getirme işlemi için yaptığımız testler :

```
1 package tr.edu.medipol.yova;
2
3 import static org.junit.Assert.*;
4
5
6
7 public class ArdahanSecimApiOyGetirme {
8
9
10 public void testAPartiOyGetirmeSifirken() {}
11
12
13
14
15
16
17
18
19
20
21 public void testAPartiOyGetirmeSifirdanBuyukken() {}
22
23
24
25
26
27
28
29
30
31
32
33 public void testBPartiOyGetirmeSifirken() {}
34
35
36
37
38
39
40
41
42
43
44 public void testBPartiOyGetirmeSifirdanBuyukken() {}
45
46
47
48
49
50
51
52
53
54
55
56 public void testCPartiOyGetirmeSifirken() {}
57
58
59
60
61
62
63
64
65
66
67 public void testCPartiOyGetirmeSifirdanBuyukken() {}
68
69
70
71
72
73
74
75
76
77
78
79 public void testOlmayanPartiOyGetirmeSifirken() {}
80
81
82
83
84
85
86
87
88
89
90 public void testOlmayanPartiOyGetirmeSifirdanBuyukken() {}
91
92
93
94
95
96
97
98
99
100
101
102
103
104
```

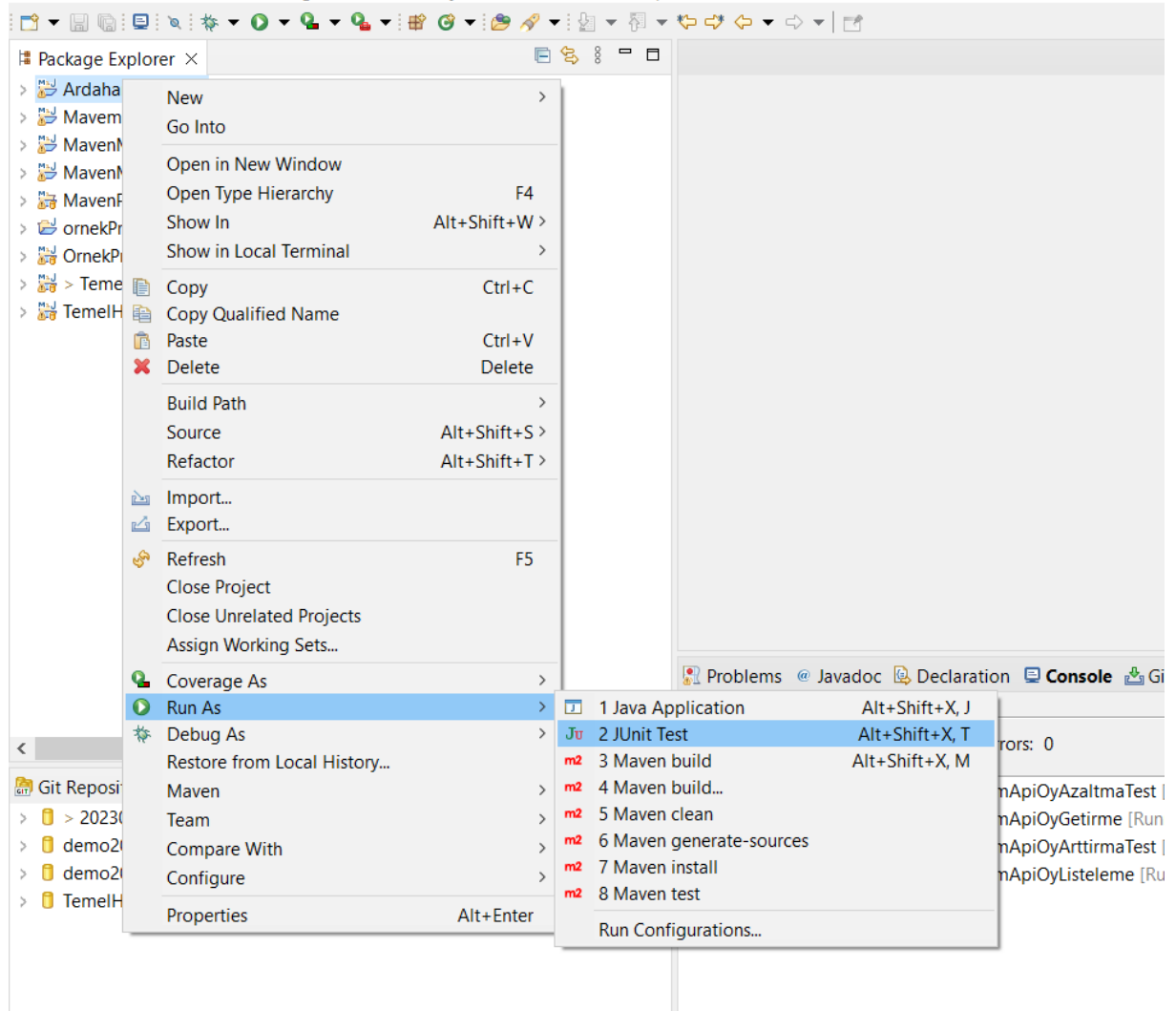
Oyları listeleme işlemi için yaptığımız testler :

```
1 package tr.edu.medipol.yova;
2
3 import static org.junit.Assert.*;
4
5
6
7
8
9 public class ArdahanSecimApiOyListeleme {
10
11
12 public void testOylariListelemeOylarSifirken() {}
13
14
15
16
17
18
19
20
21
22
23
24
25
26
27
28
29
30
31 public void testOylariListelemeOylarSifirdanBuyukken() {}
32
33
34
35
36
37
38
39
40
41
42
43
44
45
46
47
48
49
50
51
52 public void testOylariListelemeASifirdanBuyukken() {}
53
54
55
56
57
58
59
60
61
62
63
64
65
66
67
68
69
70
71
72 public void testOylariListelemeBSifirdanBuyukken() {}
73
74
75
76
77
78
79
80
81
82
83
84
85
86
87
88
89
90
91
92 public void testOylariListelemeCSifirdanBuyukken() {}
93
94
95
96
97
98
99
100
101
102
103
104
105
106
107
108
109
110
111
112 public void testOylariListelemeAveBSifirdanBuyukken() {}
113
114
115
116
117
118
119
120
121
122
123
124
125
126
127
128
129
130
131
132
133 public void testOylariListelemeAveCSifirdanBuyukken() {}
134
135
136
137
138
139
140
141
142
143
144
145
146
147
148
149
150
151
152
153
154 public void testOylariListelemeBveCSifirdanBuyukken() {}
155
156
157
158
159
160
161
162
163
164
165
166
167
168
169
170
171
172
173
174
175
176
177
```

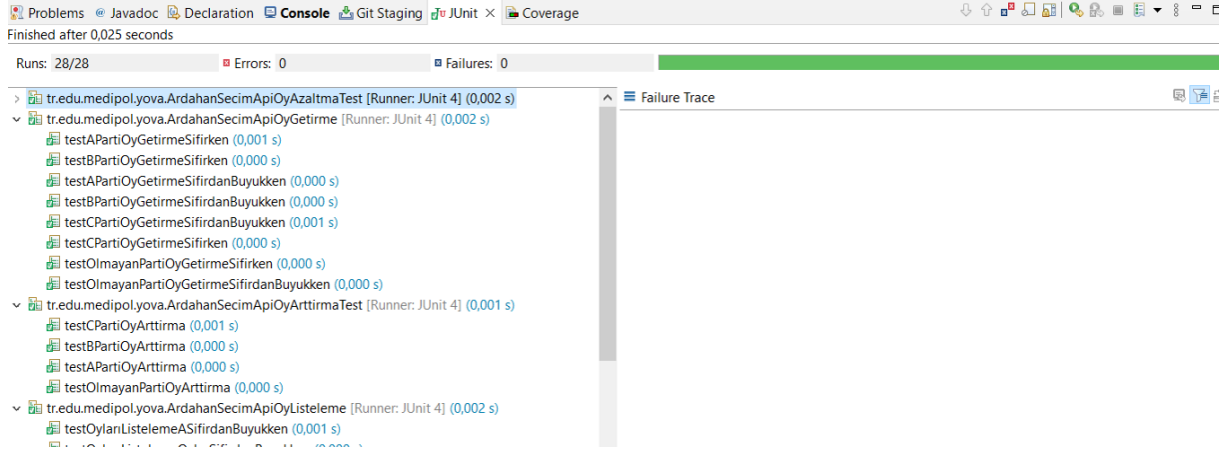
Yazdığımız testler bu şekildeydi. Her metodun içinde test için gerekli kodlar mevcut lakin raporu çok doldurmamak adına ve githubda bulunduğundan yalnızca metodlar ile anlatımda bulundum.

Şimdi testleri nasıl çalıştıracamıza geçelim.

Testleri birden fazla şekilde çalıştırabiliriz , bunlardan ilki şu şekilde :

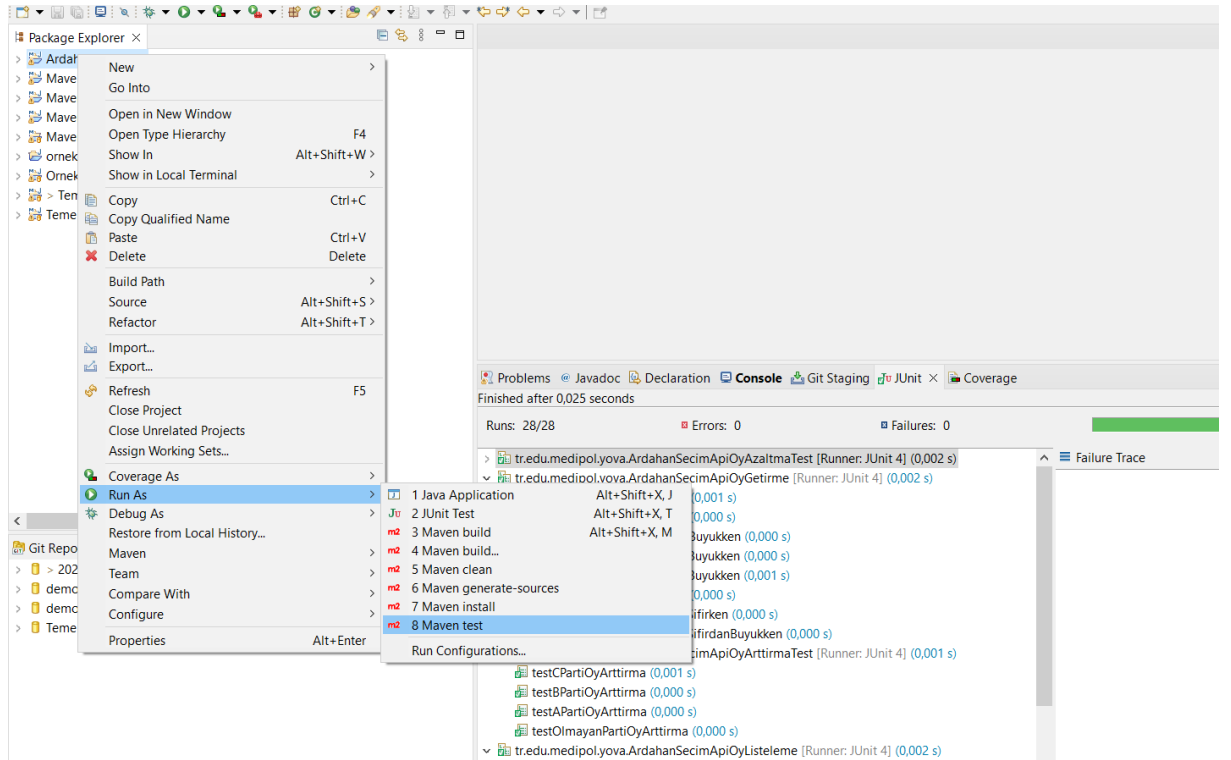


Projeye sağ tıkladık ve ilgili yolu izledik , dilersek çalıştığımız alanda da sağ tıklayarak ilgili alanda testleri çalıştırabiliriz. Çalıştığında JUnit penceresine baktığımızda şöyle bir ekranla karşılaşırız :



Burada yapılan test sayısı , kaç testin başarılı olduğu gibi bilgiler yer almakta.

Şimdi maven kısmında test etmeyi gösterelim :



Tıkladığımızda console kısmından kaç test olduğunu kaçından geçtiğimizi , kaç hata aldığımız gibi bilgileri görebiliyoruz.

```
terminated> C:\Program Files\Java\jdk-17\bin\java.exe (9 Haz 2023 00:34:57) [pid: 2000]
[WARNING] File encoding has not been set, using platform encoding UTF-8, i.e. build is platform dependent!
[INFO] Compiling 2 source files to C:\Users\emirc\ eclipseProjects\ArdahanSecimApi\target\classes
[INFO]
[INFO] --- maven-resources-plugin:2.6:testResources (default-testResources) @ ArdahanSecimApi ---
[WARNING] Using platform encoding (UTF-8 actually) to copy filtered resources, i.e. build is platform dependent!
[INFO] Copying 0 resource
[INFO]
[INFO] --- maven-compiler-plugin:3.1:testCompile (default-testCompile) @ ArdahanSecimApi ---
[INFO] Changes detected - recompiling the module!
[WARNING] File encoding has not been set, using platform encoding UTF-8, i.e. build is platform dependent!
[INFO] Compiling 4 source files to C:\Users\emirc\ eclipseProjects\ArdahanSecimApi\target\test-classes
[INFO]
[INFO] --- maven-surefire-plugin:2.12.4:test (default-test) @ ArdahanSecimApi ---
[INFO] Surefire report directory: C:\Users\emirc\ eclipseProjects\ArdahanSecimApi\target\surefire-reports

-----
T E S T S
-----
Running tr.edu.medipol.yova.ArdahanSecimApiOyArttirmaTest
Tests run: 4, Failures: 0, Errors: 0, Skipped: 0, Time elapsed: 0.065 sec
Running tr.edu.medipol.yova.ArdahanSecimApiOyAzaltmaTest
Tests run: 8, Failures: 0, Errors: 0, Skipped: 0, Time elapsed: 0.001 sec

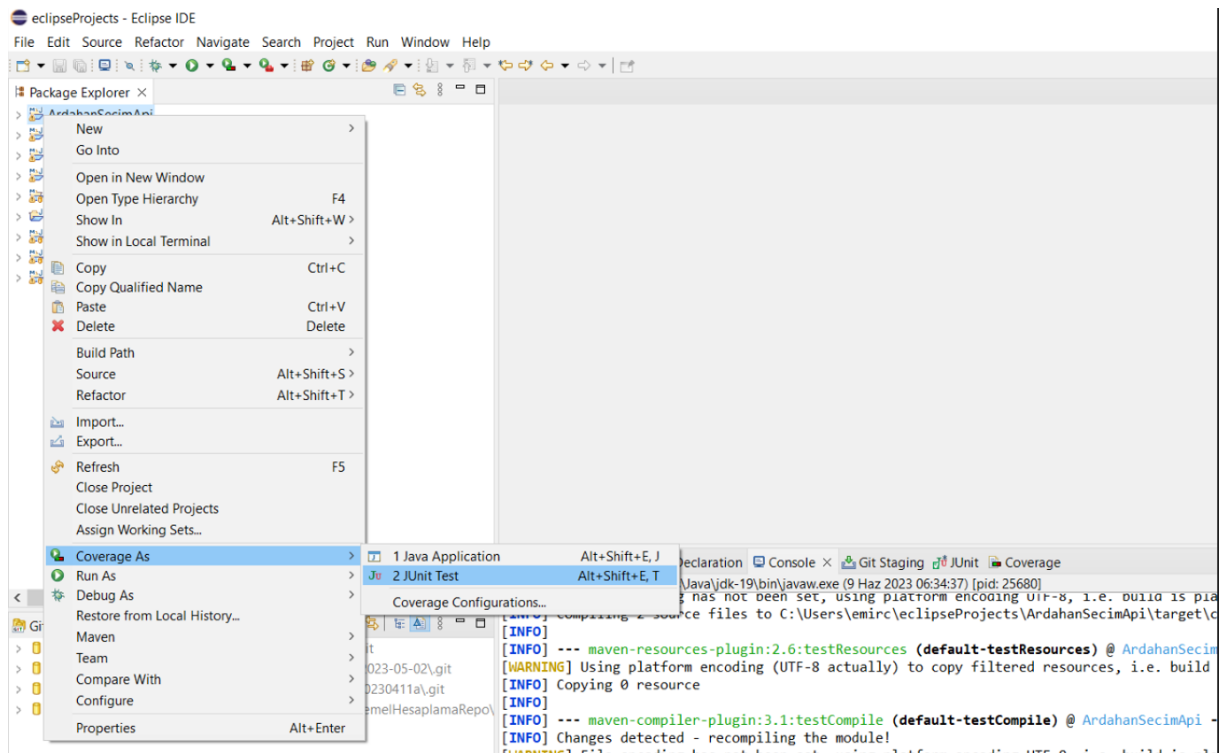
Results :

Tests run: 12, Failures: 0, Errors: 0, Skipped: 0



















[INFO]
[INFO] --- jacoco-maven-plugin:0.8.8:report (report) @ ArdahanSecimApi ---
[INFO] Loading execution data file C:\Users\emirc\ eclipseProjects\ArdahanSecimApi\target\jacoco.exec
[INFO] Analyzed bundle 'ArdahanSecimApi' with 2 classes
[INFO]
[INFO] BUILD SUCCESS
[INFO]
[INFO] Total time: 4.543 s
[INFO] Finished at: 2023-06-09T06:34:42+03:00
```

Şimdi Coverage kısmından bahsedelim . coverage kısmında yüzdeler olarak kaç testten geçtiğimiz gibi bilgileri daha detaylı bir şekilde görebiliyoruz. Ayrıca jacoco sayesinde bunu github kısmında PR yaparken gösterebileceğiz.

Bunu da şöyle yapıyoruz :




Tıkladıktan sonra Coverage pencersinde ilgili test oranlarını görmekteyiz

ArdahanSecimApi (9 Haz 2023 06:35:34)				
Element	Coverage	Covered Instru...	Missed Instru...	Total Instructio...
▼  ArdahanSecimApi	 100,0 %	1.011	0	1.011
▼  src/main/java	 100,0 %	191	0	191
>  tr.edu.medipol.yova	 100,0 %	191	0	191
▼  src/test/java	 100,0 %	820	0	820
▼  tr.edu.medipol.yova	 100,0 %	820	0	820
>  ArdahanSecimApiOyArtirma	 100,0 %	59	0	59
>  ArdahanSecimApiOyAzaltma	 100,0 %	131	0	131
>  ArdahanSecimApiOyGetirme	 100,0 %	123	0	123
>  ArdahanSecimApiOyListeleme	 100,0 %	507	0	507

Şimdi gelelim projemizi github repomuza eklemeye , öncelikle bir github repositorysi oluşturuyoruz. Repository min ismine FinalRepo dedim. Public seçerek de herkese açık olmasını sağlıyorum.

Create a new repository



A repository contains all project files, including the revision history. Already have a project repository elsewhere? [Import a repository.](#)

Owner *  EmircanKoc1 / Repository name * FinalRepo

✓ FinalRepo is available.

Great repository names are short and memorable. Need inspiration? How about [shiny-octo-journey](#)?

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.](#)

Choose a license

Github repomuzu oluşturduk şimdi sıra bunu eclipse ile bağlamaya .

Quick setup — if you've done this kind of thing before

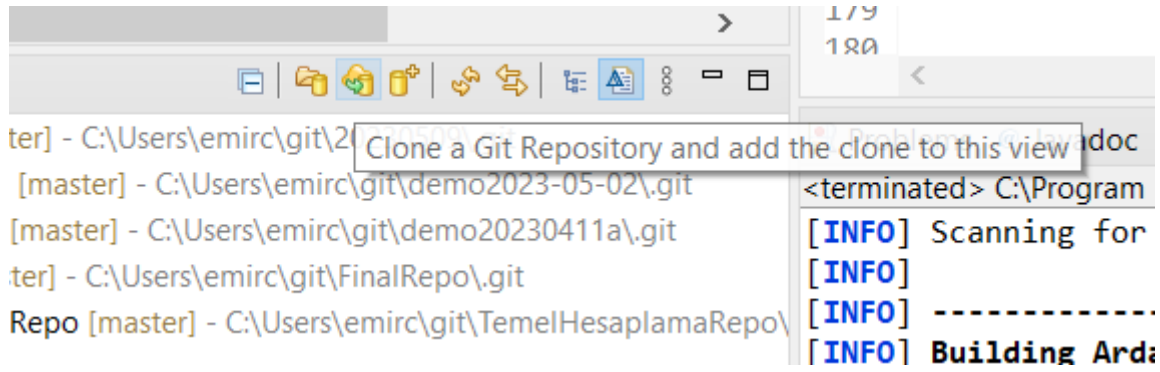
 Set up in Desktop or  HTTPS  SSH

Get started by [creating a new file](#) or [uploading an existing file](#). We recommend every repository include a [README](#), [LICENSE](#), and [.gitignore](#).

...or create a new repository on the command line

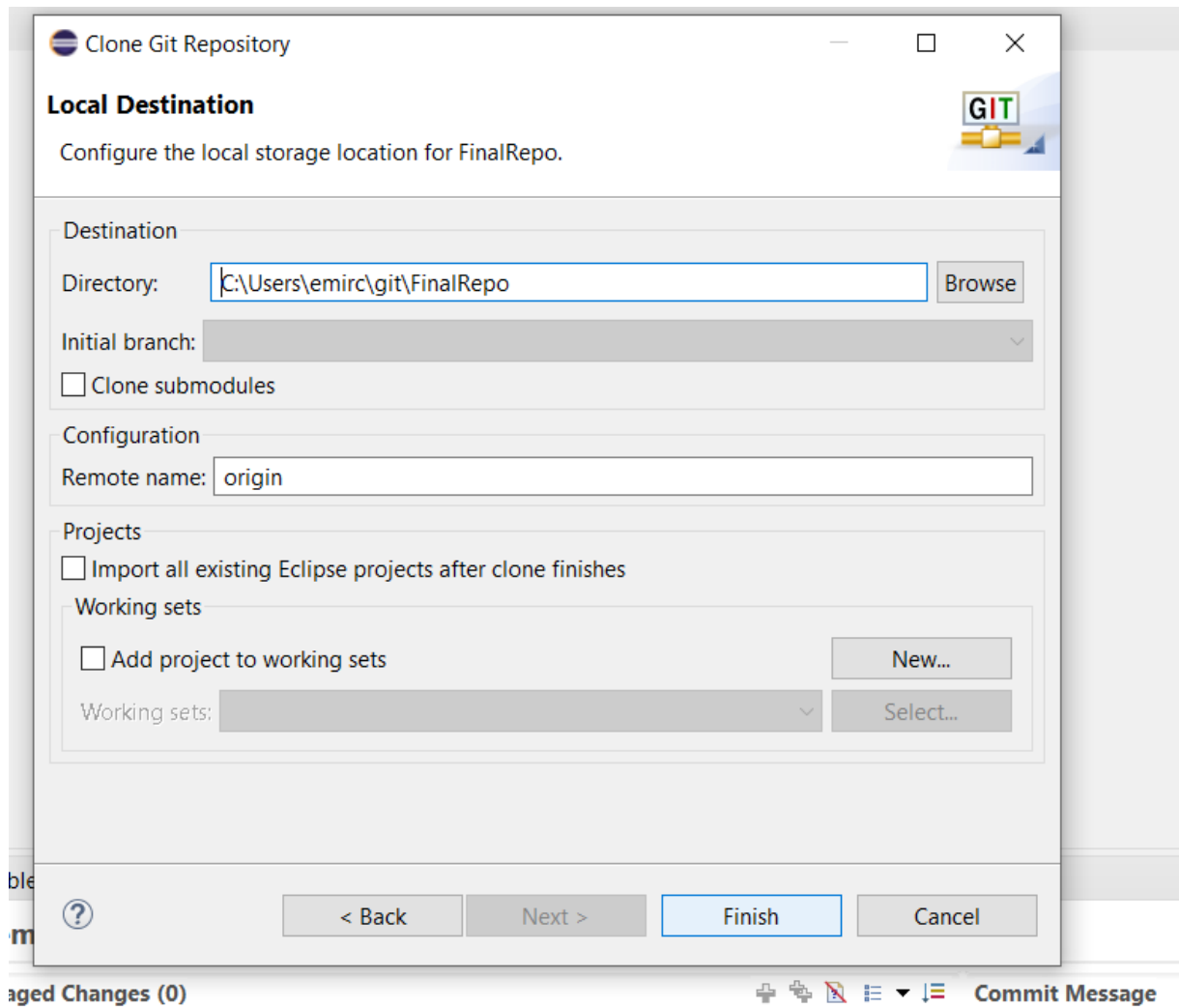
```
echo "# FinalRepo" >> README.md
git init
git add README.md
git commit -m "first commit"
git branch -M main
git remote add origin https://github.com/EmircanKoc1/FinalRepo.git
```

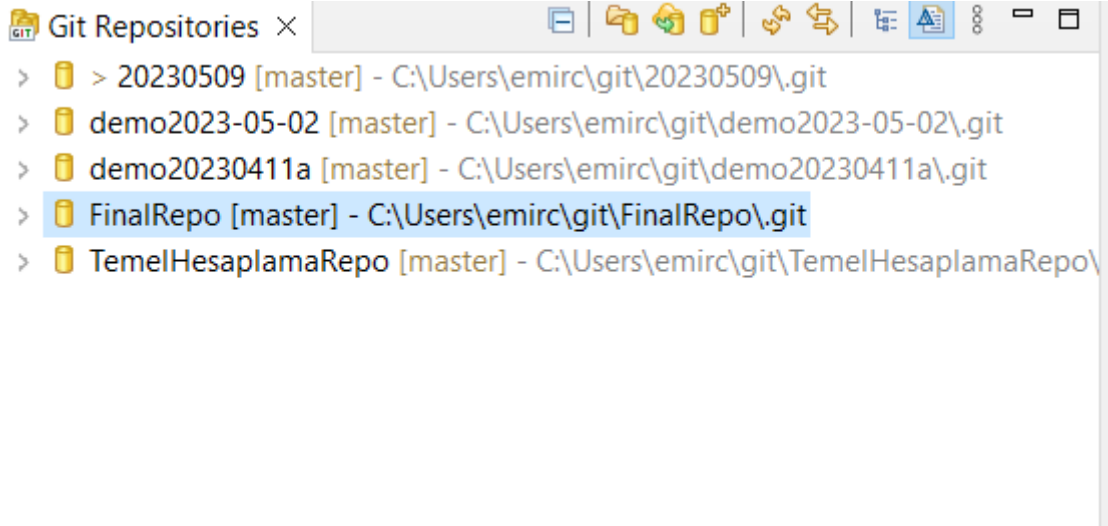
Oluşturulan repo adresimizi kopyalıyoruz. Sonra eclipse üzerinden Git Repositories penceresinden



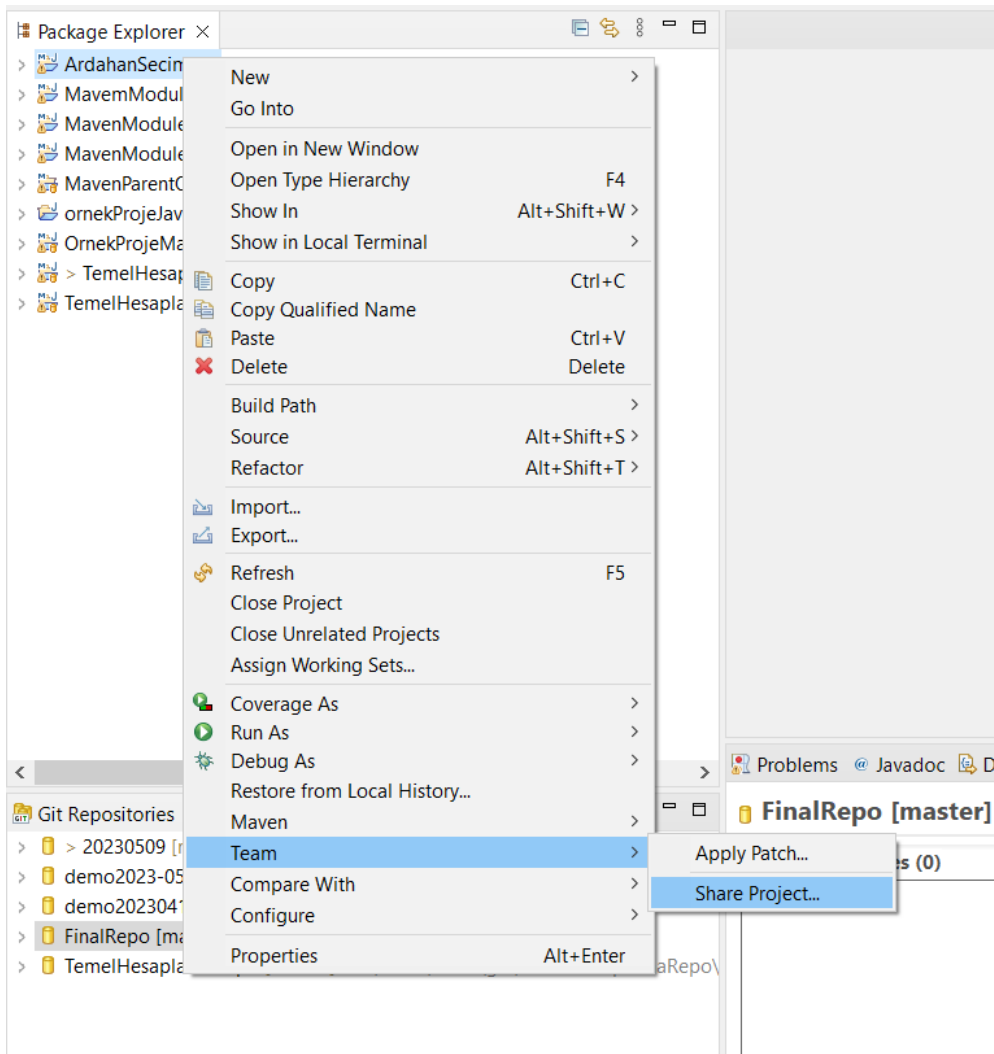
Clone a Git Repository and add the clone to this view e tıklıyoruz.

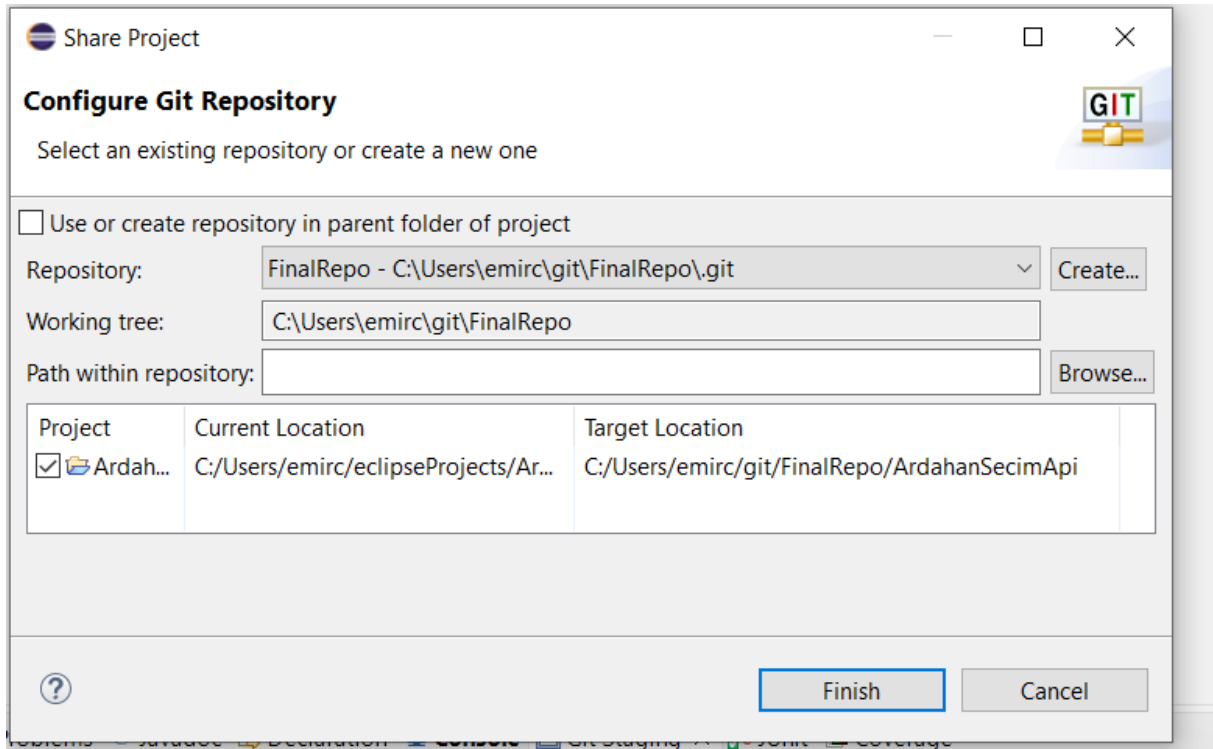
Tıkladıktan sonra açılan pencereye ilgili bilgileri giriyoruz ve Nexte basıyoruz. Sonrasında açılan pencere den finishe tıklayarak github repomuzu local repomuz ile bağliyoruz.





Görüldüğü gibi ilgili repo oluşturulmuş. Üzerinde çalıştığımız maven projesini repomuz eklemek için resimdeki yolu takip ediyoruz :





Buradan ilgili repomuzu seçtikten sonra finish e basıyoruz.

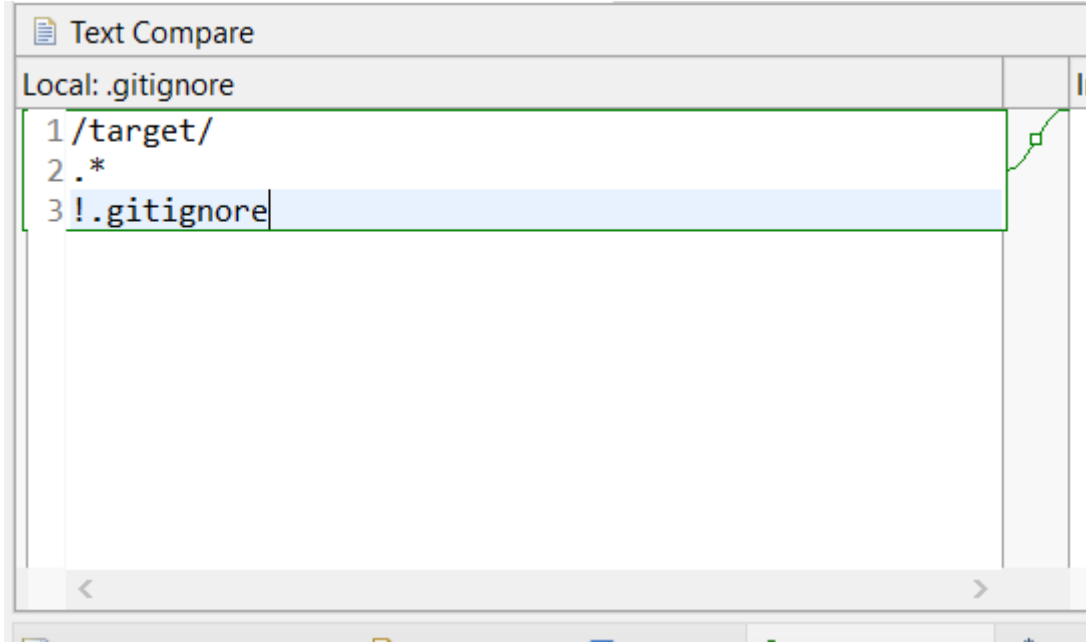
Şimdi Git Staging penceresinden githuba push etmek istediğimiz kısımları ayarlayacağız.

> FinalRepo [master]

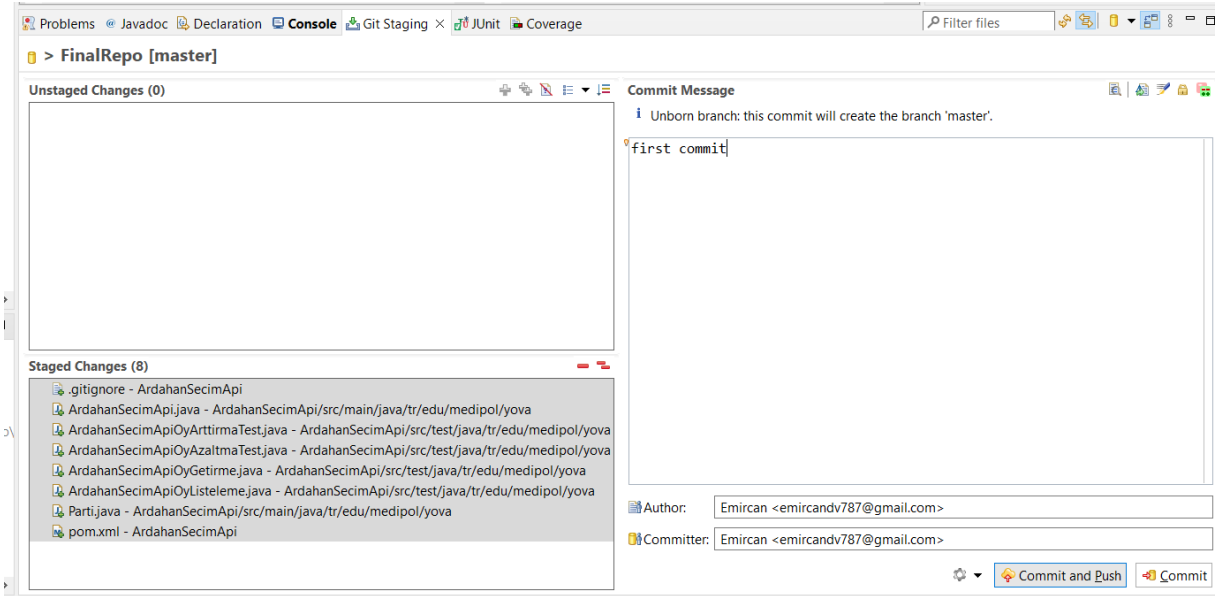
Unstaged Changes (12)

- .classpath - ArdahanSecimApi
- .gitignore - ArdahanSecimApi
- .project - ArdahanSecimApi
- ArdahanSecimApi.java - ArdahanSecimApi/src/main/java/tr/edu/medipol/yova
- ArdahanSecimApiOyArtirmaTest.java - ArdahanSecimApi/src/test/java/tr/edu/medipol/yova
- ArdahanSecimApiOyAzaltmaTest.java - ArdahanSecimApi/src/test/java/tr/edu/medipol/yova
- ArdahanSecimApiOyGetirme.java - ArdahanSecimApi/src/test/java/tr/edu/medipol/yova
- ArdahanSecimApiOyListeleme.java - ArdahanSecimApi/src/test/java/tr/edu/medipol/yova
- org.eclipse.jdt.core.prefs - ArdahanSecimApi/.settings
- org.eclipse.m2e.core.prefs - ArdahanSecimApi/.settings
- Parti.java - ArdahanSecimApi/src/main/java/tr/edu/medipol/yova
- pom.xml - ArdahanSecimApi

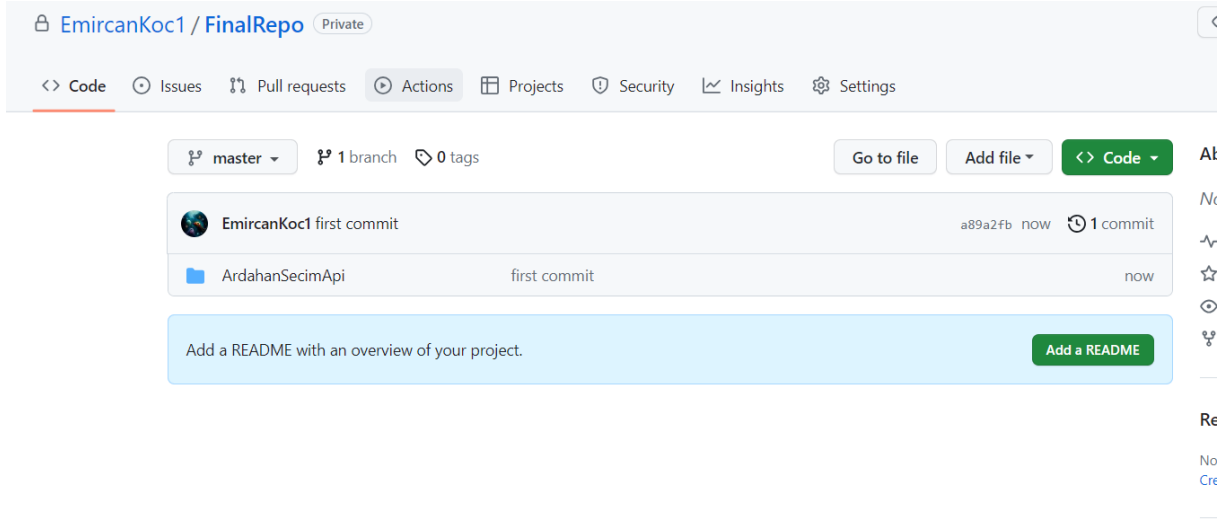
Görüldüğü gibi alakalı alakasız çok fazla dosya var göndermek istediğimizi ayarlamak için .gitignore dosyasına tıklayarak şu kodları yazıyoruz.



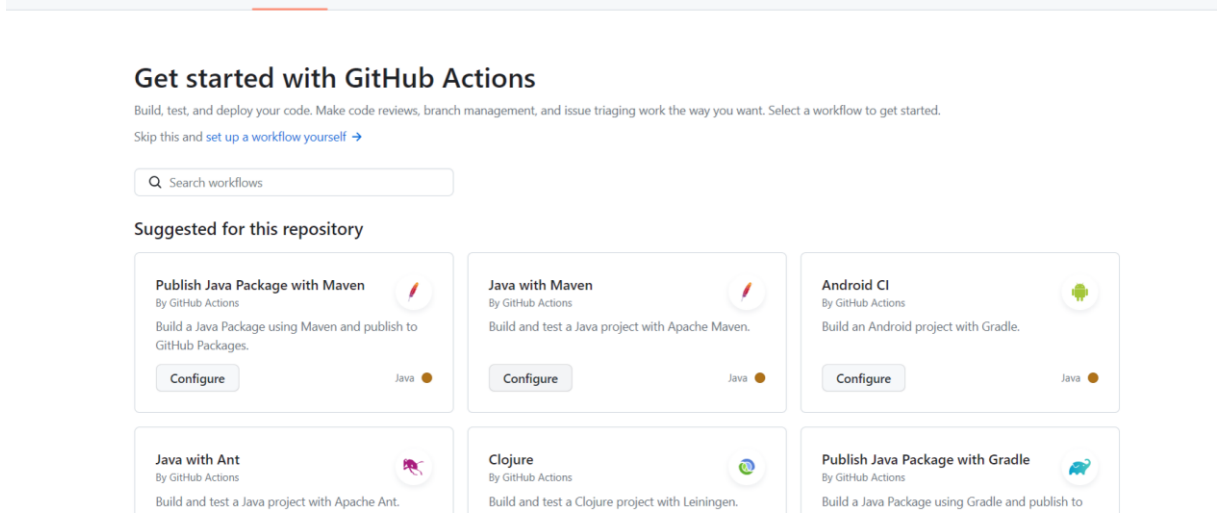
```
1 /target/  
2 .  
3 !.gitignore
```



Kalan dosyaları sürükleyip Staged Changes alanına bıraktıktan sonra sonra Commit and Push butonuna tıklıyoruz.

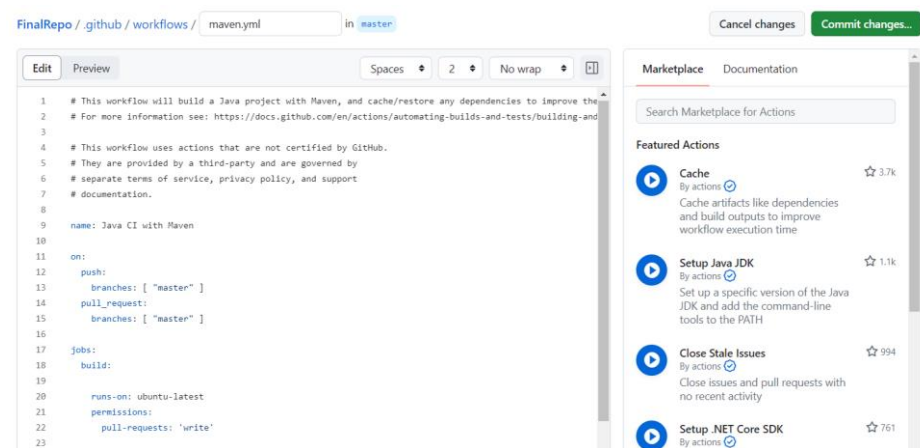


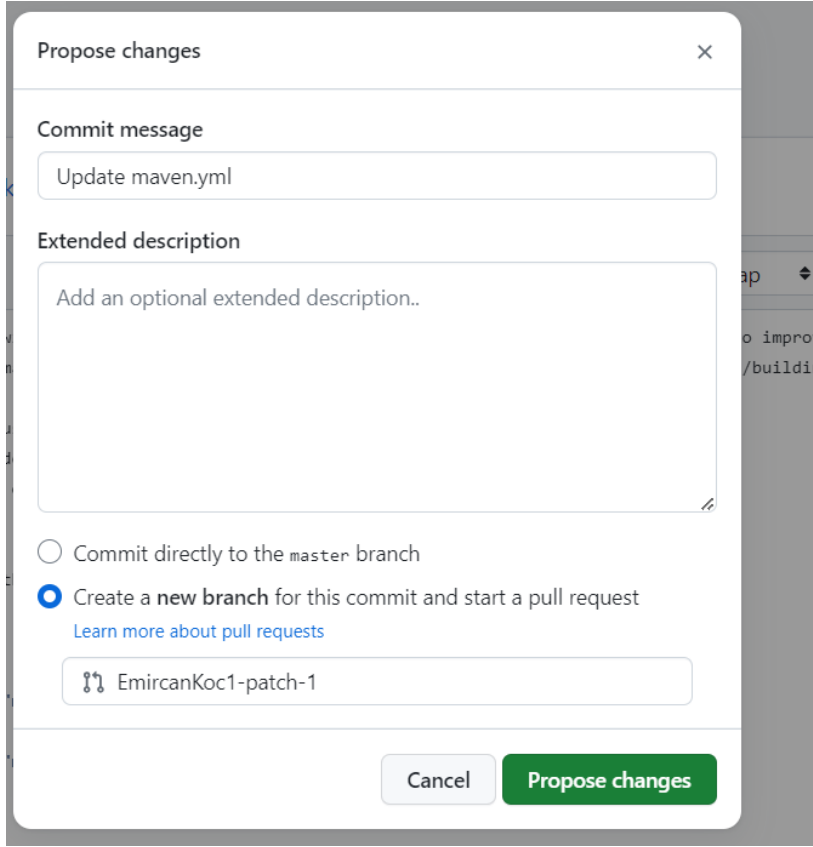
Görüldüğü gibi projemiz gelmiş. Şimdi workflow ekleme işlemi için Actions kısmına tıklıyoruz.



Açılan Pencereden Java with Maven i seçiyoruz be Configure butonuna tıklıyoruz.

Burada açılan maven.yml dosyasında gerekli ayarlamaları yaptıktan sonra sağ üstteki commit changes butonuna tıklıyoruz.



A dialog box titled "Propose changes" with a close button (X) in the top right corner. It contains a "Commit message" field with the text "Update maven.yml". Below it is an "Extended description" field with the placeholder text "Add an optional extended description..". There are two radio buttons: "Commit directly to the master branch" (unselected) and "Create a new branch for this commit and start a pull request" (selected). Below the selected option is a link "Learn more about pull requests". At the bottom, there is a branch name field containing "EmircanKoc1-patch-1" and two buttons: "Cancel" and "Propose changes".

Propose changes

Commit message

Update maven.yml

Extended description

Add an optional extended description..

☐ Commit directly to the master branch

☒ Create a new branch for this commit and start a pull request

[Learn more about pull requests](#)

EmircanKoc1-patch-1

Cancel Propose changes

Create new branch for this commit and start pull request e tıkladıktan sonra Propose changes butonuna tıklıyoruz.

Sonra actions kısmına tekrar tıklayarak build sürecine bakıyoruz.

All workflows

Showing runs from all workflows

1 workflow run

Create maven.yml

Java CI with Maven #1: Commit 22451dd pushed by EmircanKoc1

master

now

In progress

Şimdi dilersek buradan PR dilersek Readme dosyası oluşturarak PR kısmında Coverage bilgisini gösterebiliriz.

Ben yeni bir Readme.md dosyası oluşturarak göstereceğim

Open a pull request

The change you just made was written to a new branch named `EmircanKoc1-patch-3`. Create a pull request below to propose these changes.

base: master

compare: EmircanKoc1-patch-3

✓ Able to merge. These branches can be automatically merged.

Create README.md

Write Preview

H B I < > @

Leave a comment

Attach files by dragging & dropping, selecting or pasting them.

Create pull request

Remember, contributions to this repository should follow our [GitHub Community Guidelines](#).

Oluşturma kısmında create pull request butonuna tıkladıktan sonra , pull request kısmında ilgili coverage bilgisinin elde edildiğini görüyoruz.

github-actions bot commented now

There is no coverage information present for the Files changed

Total Project Coverage 74.87%

EmircanKoc1 merged commit 3956003 into master now

1 check passed

View details Revert

Pull request successfully merged and closed

You're all set—the EmircanKoc1-patch-3 branch can be safely deleted.

Delete branch

Write Preview

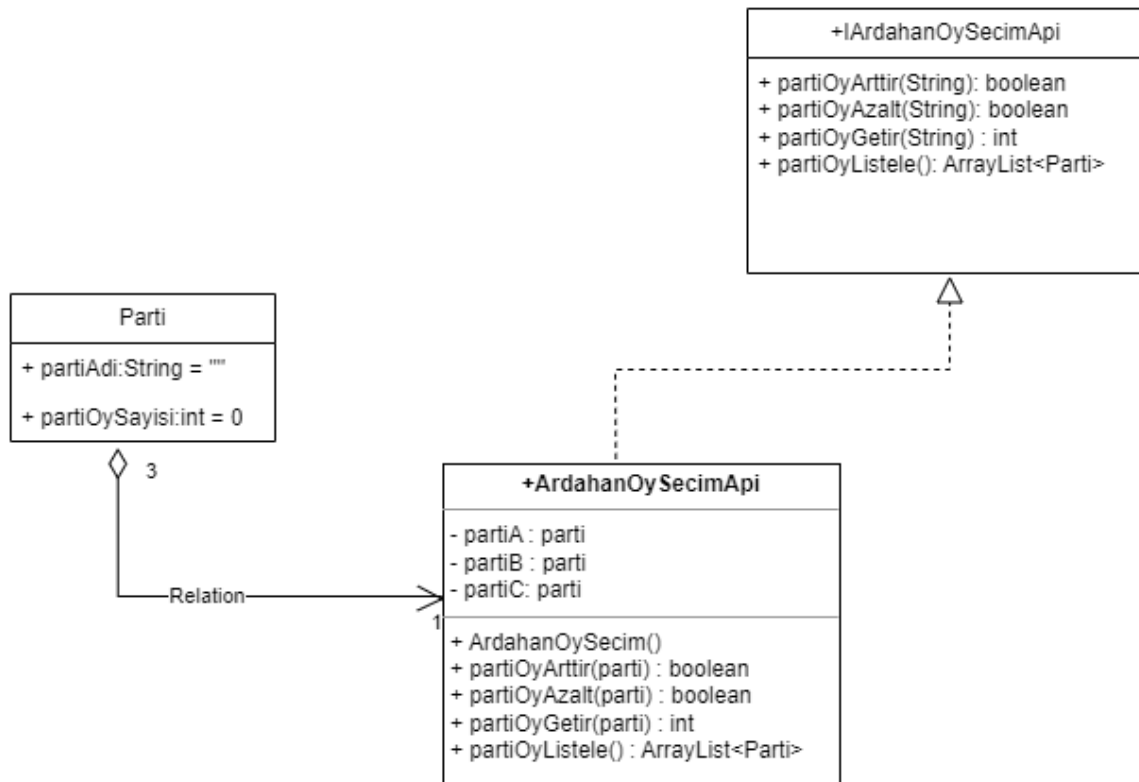
H B I < > @

Leave a comment

Bu kısımlarda sıkça hata alınabildiğinden görsel sırasında farklılıklar olmuş olabilir.

UML Diyagramları

UML Class Diyagramı



Use Case Diyagramı

