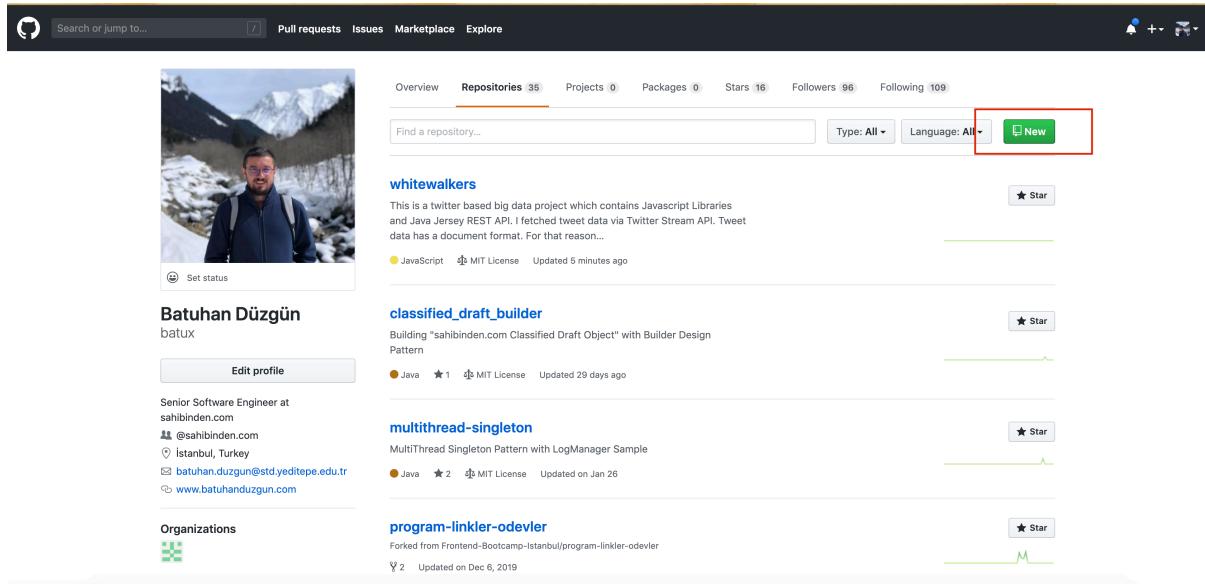


# “Github, Eclipse, Terminal ve Git” Dörtgeni ☺

## Github nedir?

Github aslında “git” kod versiyonlama sistemini kullanan ve yazılımcılara projelerini online olarak yönetmelerini sağlayan web tabanlı bir uygulamadır. Github üzerinden yeni bir repository yaratıp siz de projelerini Github üzerinde tutabilirsiniz. Bunun için bir Github hesabınızın olması yeterlidir.



The screenshot shows a GitHub user profile for 'batuhan.duzgun'. The profile includes a photo of a man with glasses and a blue jacket, a bio about a Twitter big data project, and links to his LinkedIn and email. Below the profile, there are four repository cards:

- whitewalkers**: A JavaScript and Java Jersey REST API project. It has 35 stars and was updated 5 minutes ago.
- classified\_draft\_builder**: A Java project using the Builder design pattern. It has 1 star and was updated 29 days ago.
- multithread-singleton**: A Java project demonstrating the Singleton pattern with LogManager. It has 2 stars and was updated on Jan 26.
- program-linkier-odevler**: Forked from Frontend-Bootcamp-Istanbul. It has 2 stars and was updated on Dec 6, 2019.

A green rectangular box highlights the 'New' button at the top right of the repository search bar.

Yukarıdaki örnekte “New” yeşil renkte butona tıklayarak bir repository oluşturabilirsiniz.

## Git Kod Versiyonlama Sistemi?

Git aslında bir kod versiyonlama sistemidir. Versiyonlama sisteminden kasıt şudur. Gönderilen her değişiklik bir commit olarak değerlendirilir. Değişiklikler kümülatif bir şekilde ve tarihsel bir biçimde tutulur. Böylece, projenin tüm geçmişi boyunca kodun evrimini izleyebilirsiniz. Hatta geri dönük kim ne değişiklik yapmış ne zaman yapmış görebilirsiniz. Git sistemi commit denen kod değişikliklerini bir yığın (Stack) yapısında tutar.

COMMENT	DATE
CREATED MAIN LOOP & TIMING CONTROL	14 HOURS AGO
ENABLED CONFIG FILE PARSING	9 HOURS AGO
MISC BUGFIXES	5 HOURS AGO
CODE ADDITIONS/EDITS	4 HOURS AGO
MORE CODE	4 HOURS AGO
HERE HAVE CODE	4 HOURS AGO
AAAAAAA	3 HOURS AGO
ADKFJSLKDFJSOKLFJ	3 HOURS AGO
MY HANDS ARE TYPING WORDS	2 HOURS AGO
HAAAAAAAAANDS	2 HOURS AGO

AS A PROJECT DRAGS ON, MY GIT COMMIT MESSAGES GET LESS AND LESS INFORMATIVE.

Yukarıda da bu şematize edilmiştir. Değişiklik listesini görebiliyoruz. Şimdi komut satırından bir projemize ait git commit'lerini görelim.

```
X bash
batuhan.duzgun@test-MacBook-Pro ~/git/objects-classes - (master) $
batuhan.duzgun@test-MacBook-Pro ~/git/objects-classes - (master) $
batuhan.duzgun@test-MacBook-Pro ~/git/objects-classes - (master) $ git log
commit 6d16bc9f9f7324f39dc17d4547dd1ad96e13134e (HEAD -> master, origin/master)
Author: Batuhan Düzgün <batuhan.duzgun@std.yeditepe.edu.tr>
Date:   Sun Mar 1 16:19:00 2020 +0300

    textAutomobile function moved to another class!

commit 13bdff0b108cae4869f3742a65b5bbfad04031b3d
Author: Batuhan Düzgün <batuhan.duzgun@std.yeditepe.edu.tr>
Date:   Sun Mar 1 16:15:59 2020 +0300

    textAutomobile function moved to another class!

commit f7e25dec7b86efd98912757048aa6fe6fa2f73f5
Author: Batuhan Düzgün <batuhan.duzgun@std.yeditepe.edu.tr>
Date:   Sun Mar 1 12:50:42 2020 +0300

    Gear inc/dec functions developed, textAutomobile function added!

commit 87ca46aa3cfb9147fa240931b923b7b80083fa52
Author: Batuhan Düzgün <batuhan.duzgun@std.yeditepe.edu.tr>
Date:   Sun Mar 1 12:03:08 2020 +0300

    Dummy change for trial!

commit 20553573c30de226e5881808e8a9475dc0635501
Author: Batuhan Düzgün <batuhan.duzgun@std.yeditepe.edu.tr>
Date:   Sun Mar 1 11:15:19 2020 +0300

    Automobile and GPS classes developed!

    Change-Id: I462cf2c697aac34db245d847fa8463dae89f3bfc

commit da5f468c98df78fd1ec5c0ba44aba7e4c153ee29
Author: Batuhan Düzgün <batuhan.duzgun@std.yeditepe.edu.tr>
Date:   Sun Mar 1 09:50:45 2020 +0300

    initial commit!

    Change-Id: I1070b1942223f8ab5863401da44faec4e40acf19

commit b4bf1b30b974d4daf7c262925ea607d6a4640954
Author: Batuhan Düzgün <batuhan.duzgun@std.yeditepe.edu.tr>
Date:   Sun Mar 1 09:45:46 2020 +0300

    Initial commit
batuhan.duzgun@test-MacBook-Pro ~/git/objects-classes - (master) $ █
```

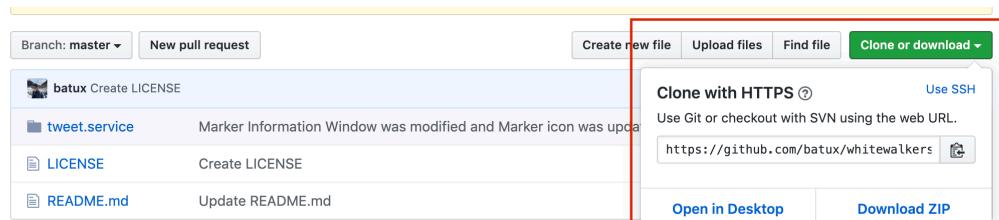
Yukarıdaki gibi projeye ait tüm commitleri görebiliyorum. Git sisteminde her commit için bir tane tekil hashcode değeri oluşturulur. Sarı renkteki yazı hashcode'u ifade eder.

## Terminal üzerinden Git clone ile Repository Entegrasyonu

Github üzerinden açığınız repository aslında uzakta başka bir sunucuda durmaktadır. Siz ise bilgisayarındaki değişiklikleri bu uzak sunucudaki git sistemine yollamak istiyorsunuz diyelim.

Not: Bunun için git sistemini bilgisayarınıza kurmanız gerekmektedir. Google üzerinden kurulum detaylarına bakabilirsiniz.

Diyelim ki git sistemi kurdunuz. Github üzerinde yeni repository oluşturduğunuz. Github üzerinden repository'nin clone path'ini almalısınız. Örneğin aşağıdaki gibi.



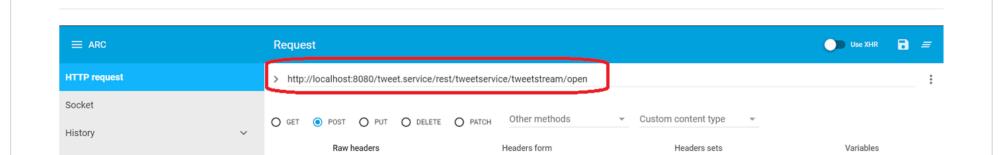
The screenshot shows a GitHub repository page for 'whitewalkers'. At the top right, there is a green 'Clone or download' button. A red box highlights this button and the surrounding area, which includes options for 'Clone with HTTPS' (with a link to <https://github.com/batux/whitewalkers>) and 'Download ZIP'. Below this, there are links for 'Open in Desktop' and 'Download ZIP'. The main content area displays the repository's README.md file, which contains the title 'whitewalkers' and a section titled 'The Description of Project'.

## whitewalkers

### The Description of Project

This project uses Twitter Stream API to store tweet data in MongoDB database. I used Java Jersey REST API and Javascript Libraries. Tweet data has a document format. For that reason, I preferred MongoDB to store big data. You can find a zip file which named as "backup.zip" in project folder. You can use to import this backup to your MongoDB. With this way, You can load tweet data that has geo location information.

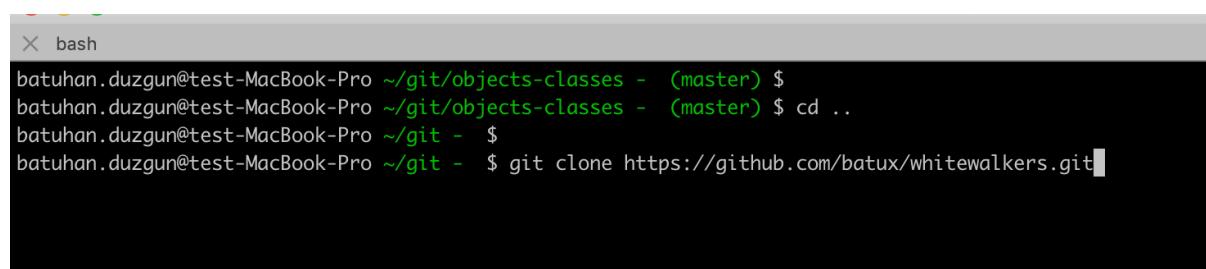
### Open Twitter Stream via REST Call



The screenshot shows a browser-based REST client interface. In the 'Request' tab, there is a red box highlighting the URL input field which contains the value '<http://localhost:8080/tweet.service/rest/tweetservice/tweetstream/open>'. Below the URL, there are buttons for selecting HTTP methods: GET, POST, PUT, DELETE, PATCH, and Other methods. The 'POST' method is selected. There are also tabs for 'Raw headers', 'Headers form', 'Headers sets', and 'Variables'.

Masaüstünüzde boş bir klasör açın. Terminalinizi açın ve oluşturduğunuz klasöre gidin.

Terminale:



```
batuhan.duzgun@test-MacBook-Pro ~/git/objects-classes - (master) $ batuhan.duzgun@test-MacBook-Pro ~/git/objects-classes - (master) $ cd .. batuhan.duzgun@test-MacBook-Pro ~/git - $ batuhan.duzgun@test-MacBook-Pro ~/git - $ git clone https://github.com/batux/whitewalkers.git
```

"git clone" komutu ile Github üzerindeki repository'i kendi bilgisayarımıza çekiyoruz.

Sonra bu projeyi Eclipse üzerinde Java projesi olarak import edin. Nasıl import edilir detaylarına girmiyorum. Sınıfta açıklamıştım. Bunu yapmak için Google araması yapınız 😊

## “git commit” ile değişiklikleri göndermek

“git commit” komutu projede yaptığınız değişiklikleri kendi bilgisayarınızdaki bir alan kalıcı olarak saklamayı sağlar. Böylece değişiklikleriniz git sisteminde kaybolmaz. Ama değişiklikleri halen uzaktaki Github repository’e yollamadık. Sadece kendi bilgisayarımızda saklamaya yarar.

```
× bash
batuhan.duzgun@test-MacBook-Pro ~/git/whitewalkers - (master) $ 
batuhan.duzgun@test-MacBook-Pro ~/git/whitewalkers - (master) $ 
batuhan.duzgun@test-MacBook-Pro ~/git/whitewalkers - (master) $ git commit -am 'Twitter geo location service development for web service!' █
```

“git commit -am” komutu yanına commit açıklamasını tek tırnaklar arasına yazıp kod commit yapabilirsiniz.

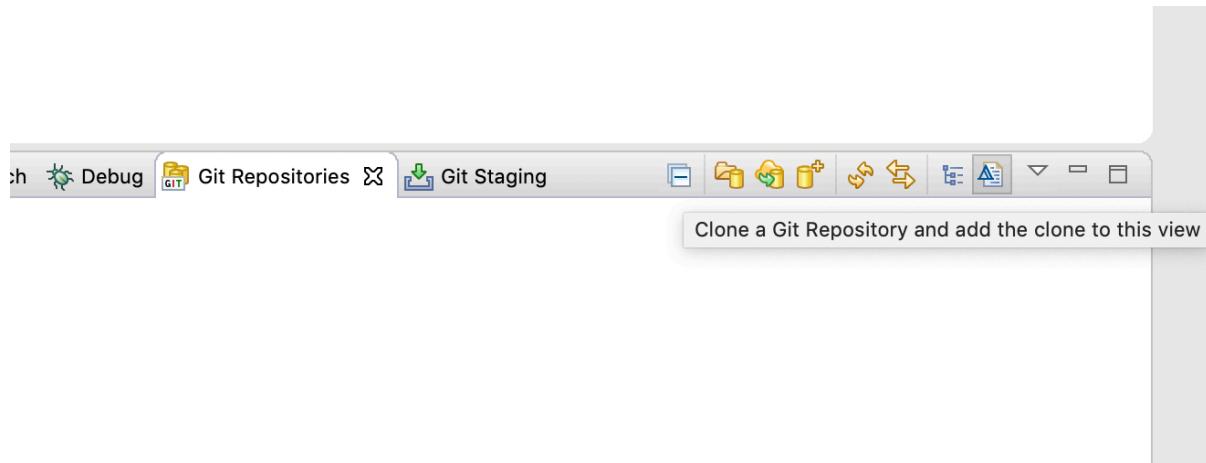
## “git push” ile değişiklikleri uzaktaki Github Repository’e göndermek

Değişiklikleri uzaktaki “master” adını verdigimiz branch’e yollamak için “git push” komutunu kullanırız. Böylece, değişiklikler kalıcı olarak uzak sunucuya gönderilir.

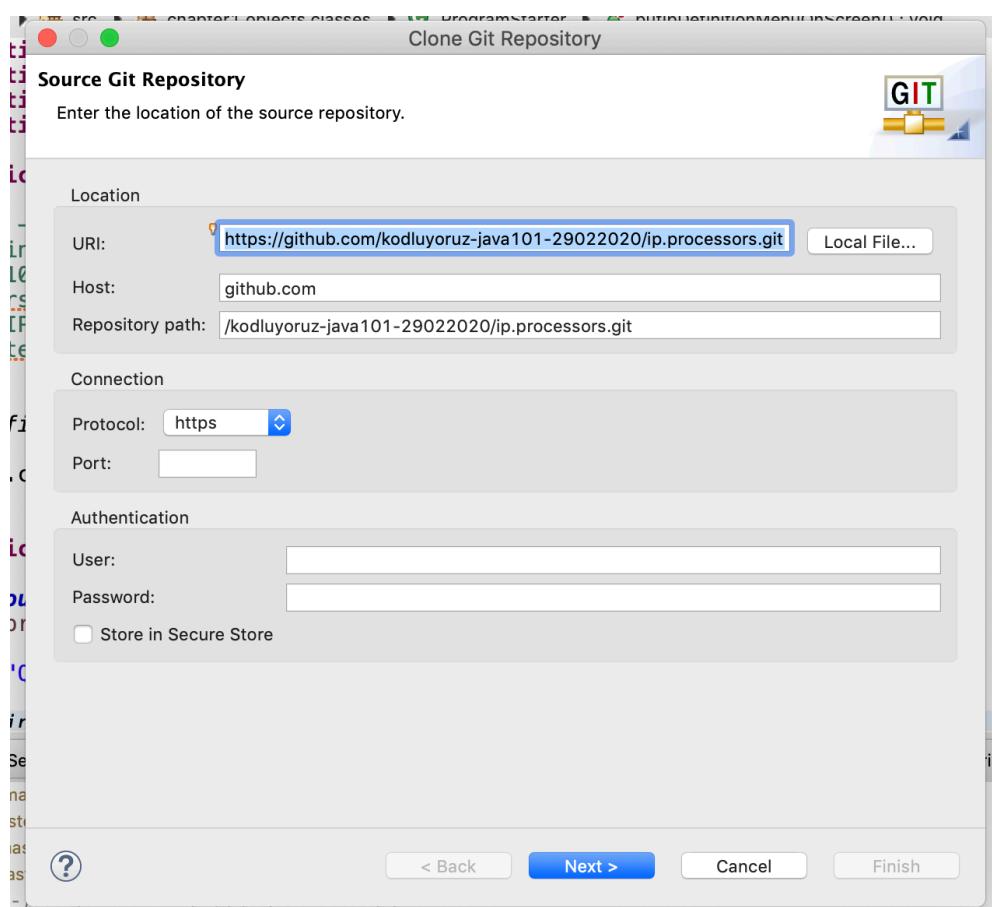
```
● ● ●
× bash
batuhan.duzgun@test-MacBook-Pro ~/git/whitewalkers - (master) $ 
batuhan.duzgun@test-MacBook-Pro ~/git/whitewalkers - (master) $ 
batuhan.duzgun@test-MacBook-Pro ~/git/whitewalkers - (master) $ 
batuhan.duzgun@test-MacBook-Pro ~/git/whitewalkers - (master) $ git push █
```

# Eclipse üzerinden Git İşlemleri

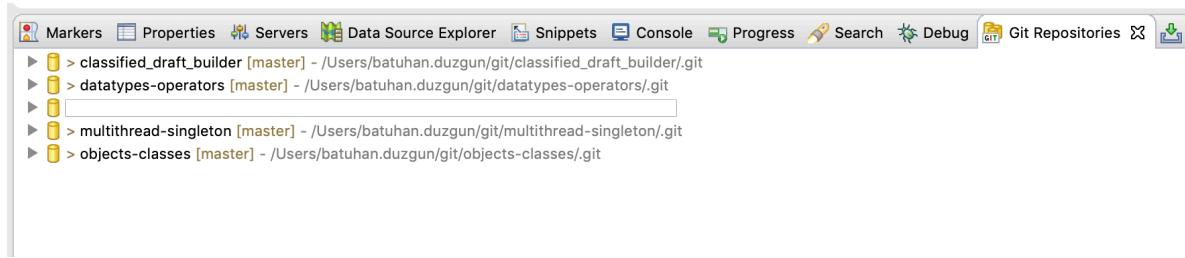
Eclipse üzerinden “git clone” yapabilmek için aşağıdaki gibi yol izlenebilir.



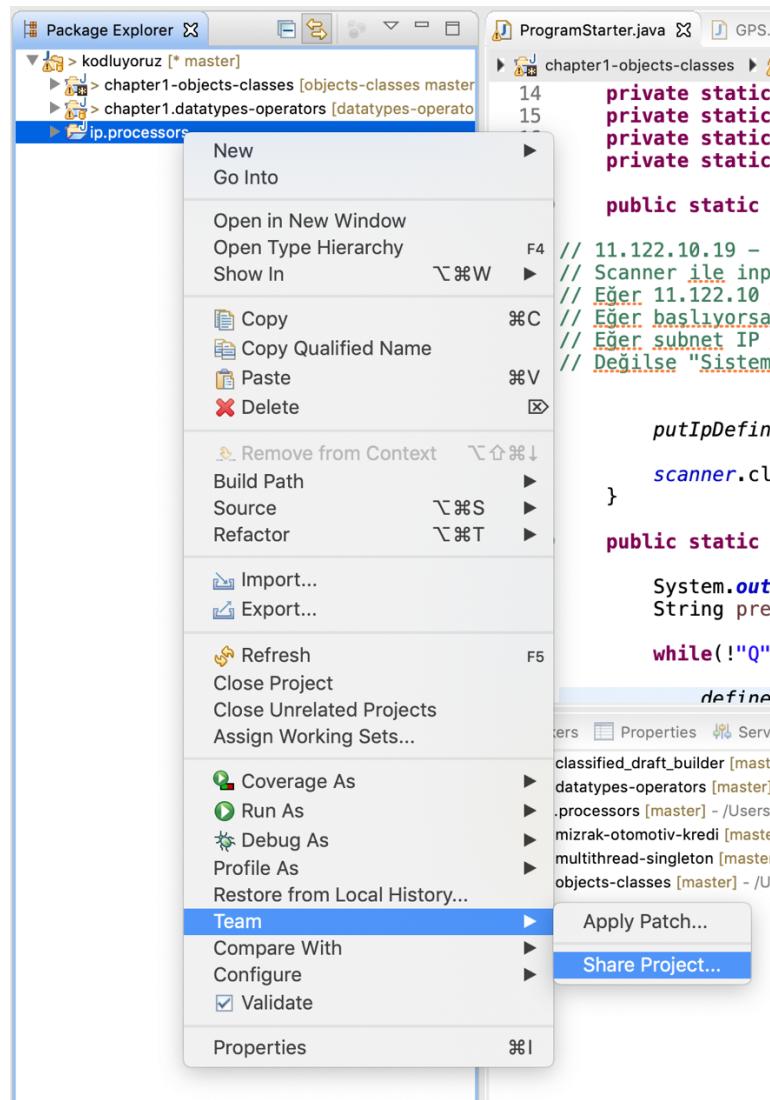
Yeşil ok işaretini olan tuşa tıklayıp git clone yapabilirsiniz. Yukarıdaki “Git Repositories” sekmesi gelmeyenler Eclipse üzerinde “Window >> Show Views >> Other ” yolunu takip ederek ekleyebilirler.

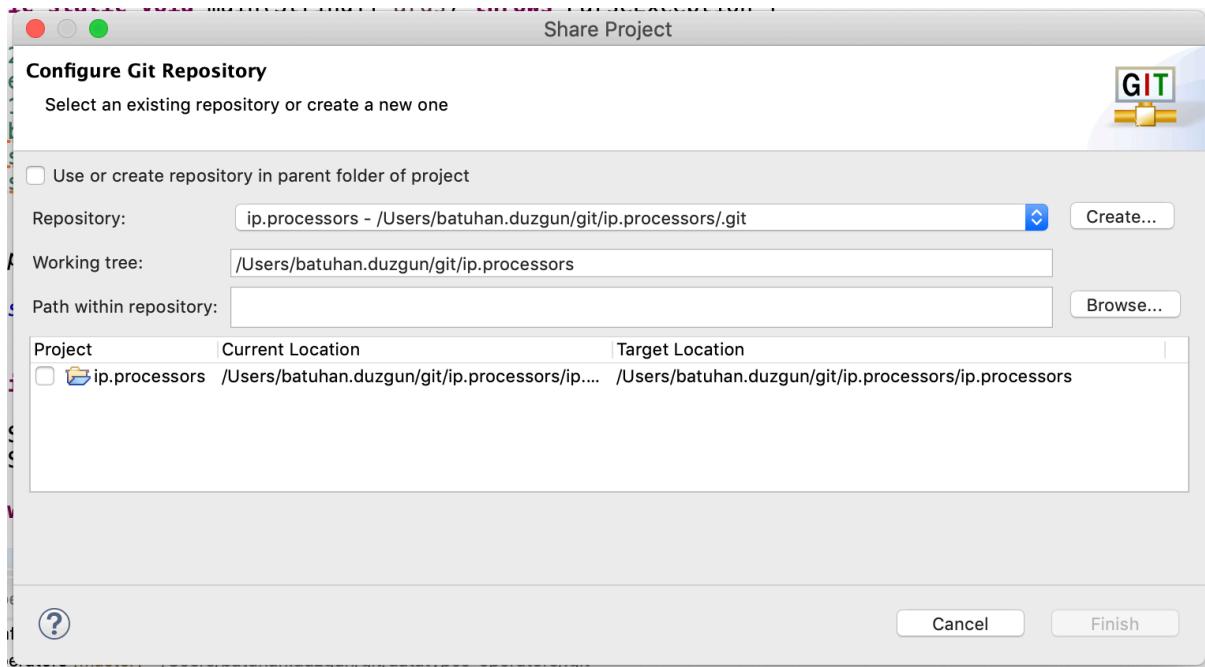


Bu işlem sonrasında tab'ın içinde repository'ler listelenir.



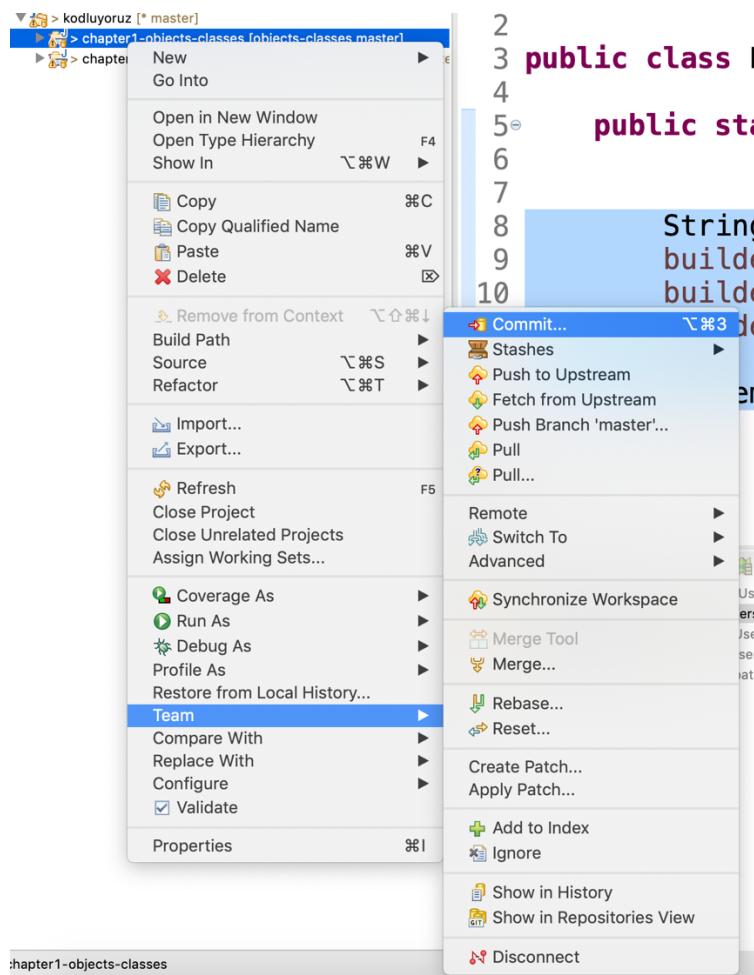
Eclipse'te bir tane Java Project oluşturarak değişikliklerinizi ve kodlarınızı geliştirin. Ardından projeye sağ tıklayıp "Team >> Share Project" diyerek proje ile git repository'si arasında bağ kurulur.



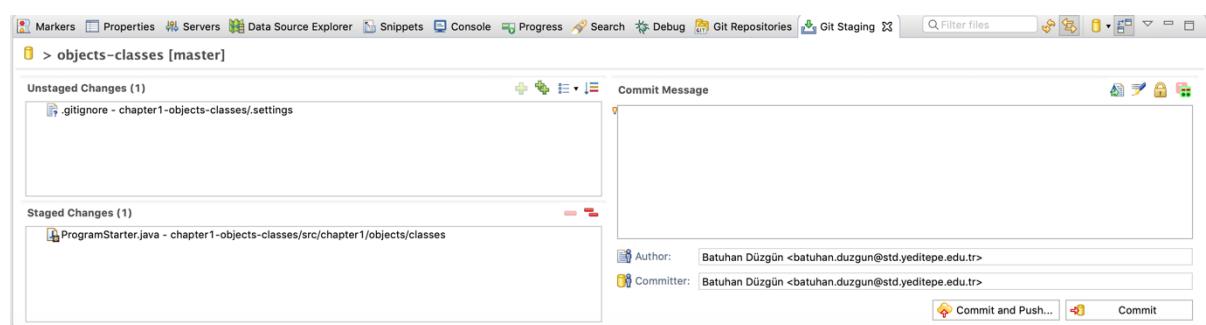


Ardından git clone yaptığınız repository'i seçip "Finish" tuşuna basabilirsiniz.

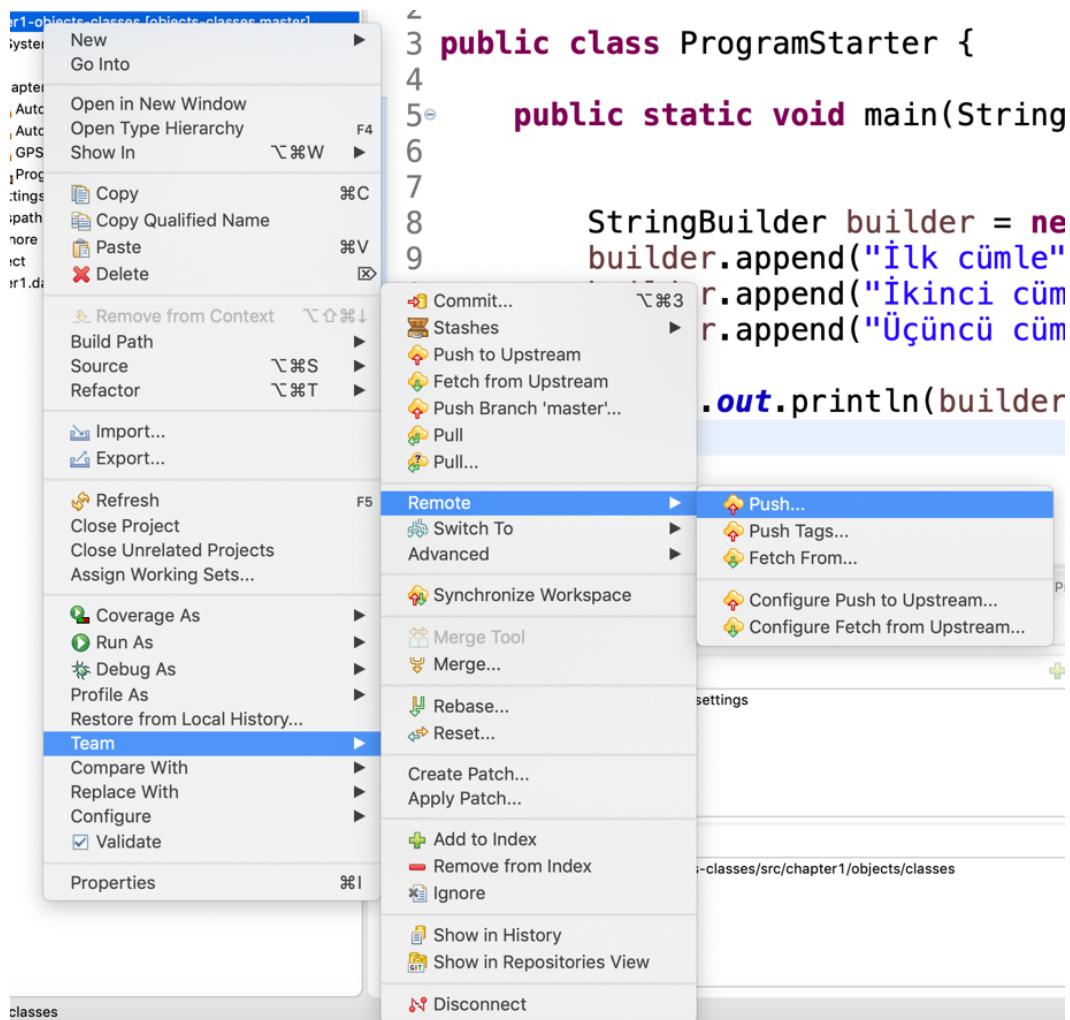
Projenize sağ tıklayıp değişiklikleri commitleyebilirsiniz.



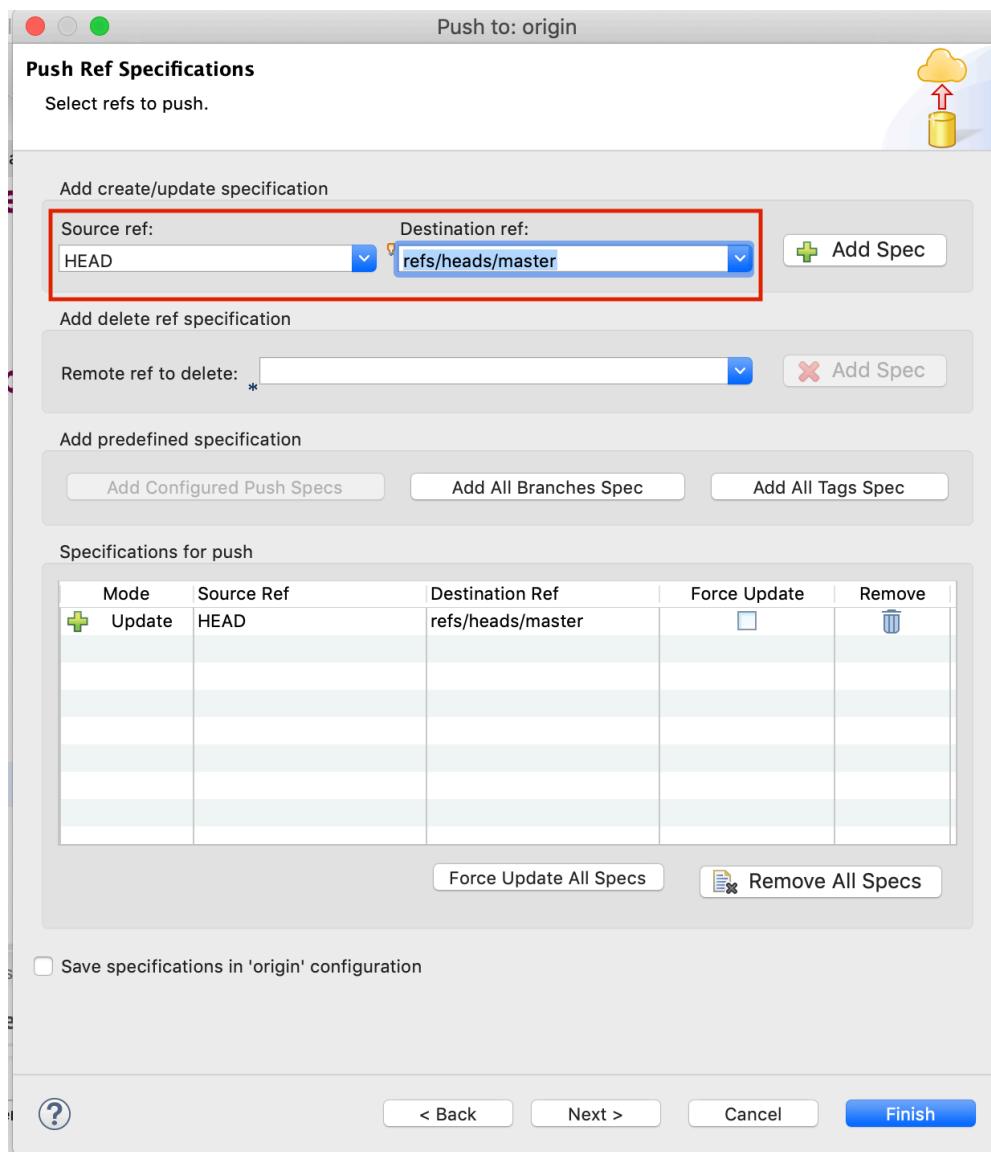
Commitleme işlemini aşağıdaki panelden yapabilirsiniz.



Projeye sağ tıklayarak git push yaparak kodları uzak sunucuya gönderebiliriz.



Push atarken sizden aşağıdaki konfigürasyonu isteyebilir.



Yukarıdaki kırmızı dikdörtgen içindeki ayarları yapın ve “Add Spec” tuşuna basın. Ardından sizden Github kullanıcı adınızı ve şifrenizi isterse giriniz.