



# **İSTANBUL MEDİPOL ÜNİVERSİTESİ**

## **YAZILIM GELİŞTİRME VE ORTAM ARAÇLARI Final Proje**

**H5150061 KERİM BİBER**

**ŞUBAT 2018**

# MAVEN

**Maven**, proje geliştirirken proje içerisinde bir standart oluşturmamızı, geliştirme sürecini basitleştirmemizi, dökümantasyonumuzu etkili bir şekilde oluşturmamızı sağlayan bir araçtır.

# KODLAR

```
Haber.java
1 package kerimbiber;
2
3
4 public class Haber {
5
6     public static i
7     private int id;
8     private String
9     private String
10    public Haber(St
11        this.id = C
12        this.habera
13        this.icerik
14    }
15    public String g
16        return habe
17    }
18    public String g
19        return icerik;
20    }
21
22 }
23
```

```
KerimovicApplication.java
1 package kerimbiber.kerimovic;
2
3 import org.springframework.boot.SpringApplication;
4
5 @SpringBootApplication
6 public class KerimovicApplication {
7
8     public static void main(String[] args) {
9         SpringApplication.run(KerimovicApplication.class, args);
10    }
11
12 }
```

```
RESTServis.java
1 package kerimbiber.kerimovic;
2
3 import java.util.ArrayList;
4
5 @RestController
6 public class RESTServis {
7     private static final List<Haber> HABER = new ArrayList<Haber>();
8
9     private Object kilit = new Object();
10
11     @RequestMapping("/haber/olustur")
12     public synchronized Haber ogrenciOlustur(String haberiadi, String icerik){
13         synchronized (kilit) {
14             Haber haber = new Haber(haberadi, icerik);
15             HABER.add(haber);
16             return haber;
17         }
18     }
19
20     @RequestMapping("/haber/listele")
21     public synchronized List<Haber> ogrenciListele(){
22         synchronized (kilit) {
23             if (HABER.size()>10) {
24                 return HABER.subList(0, 10);
25             } else {
26                 return HABER;
27             }
28         }
29     }
30
31 }
32
33
34
```

```
@RequestMapping("/haber/temizle")
public String haberTemizle(){
    synchronized (kilit) {
        HABER.clear();
        return "Haberler silindi.";
    }
}
```

# SPRING BOOT

**Spring Boot**, Spring tabanlı uygulama geliřtirmenin en hızlı ve kolay yolu olması amacıyla geliřtirilmiř bir frameworktür.

# JAVADOC

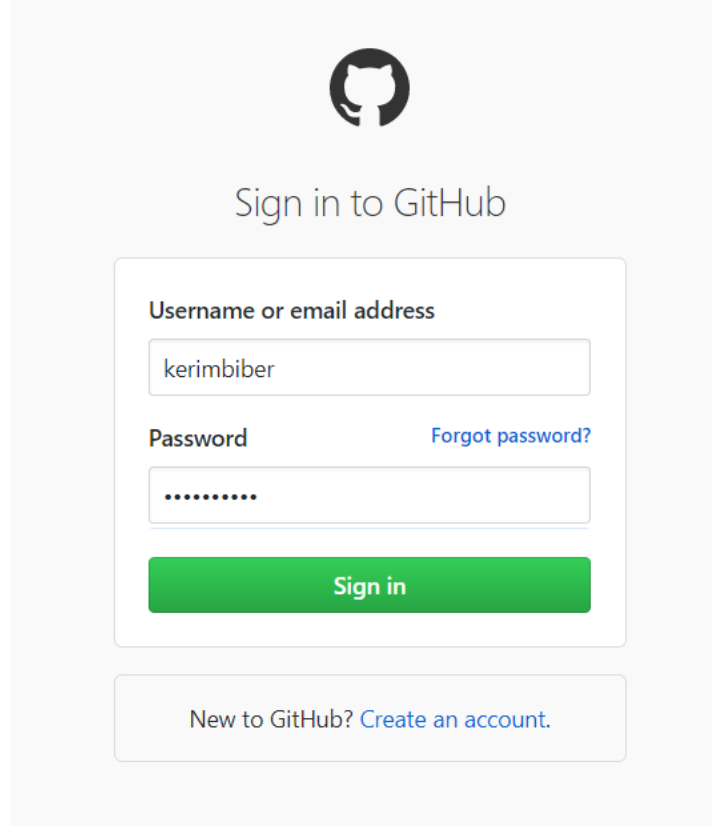
**Javadoc**, Java yazılım projelerinde yazılım dökümantasyonu yazmak için belirli bir format sunar ve bu format html olarak görüntülenir kolaylıkla içerisinde gezinebilir bir yapı olur.

Javadoc'u Eclipse'den oluşturmak için, üst menüden Project > Generate Javadoc yolunu izleyerek javadoc oluşturabilirsiniz.

# PROJİYİ GITHUB'A YÜKLEME

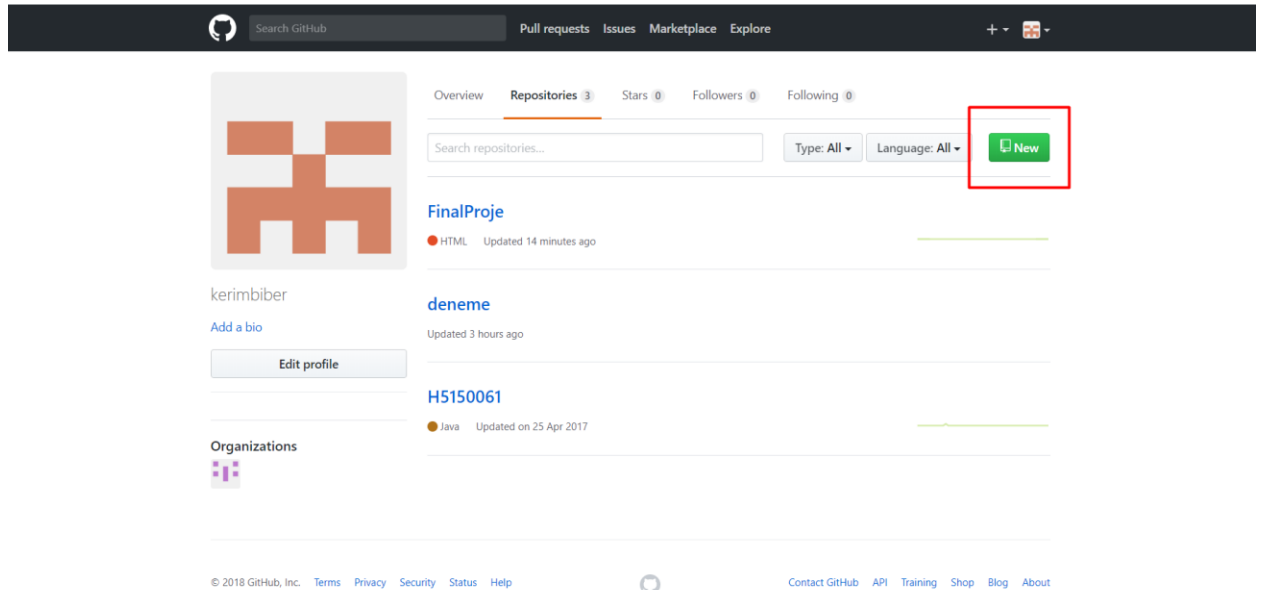
Bir projeyi githuba yüklemek için yapmamız gereken adımlar:

Öncelikle bir github hesabımızın olması gerekiyor.  
Hesabımıza giriş yapıyoruz.



The image shows the GitHub sign-in page. At the top is the GitHub logo. Below it, the text "Sign in to GitHub" is centered. There are two input fields: "Username or email address" with the text "kerimbiber" and "Password" with masked dots. To the right of the password field is a link "Forgot password?". Below the fields is a green "Sign in" button. At the bottom, there is a link "New to GitHub? Create an account."

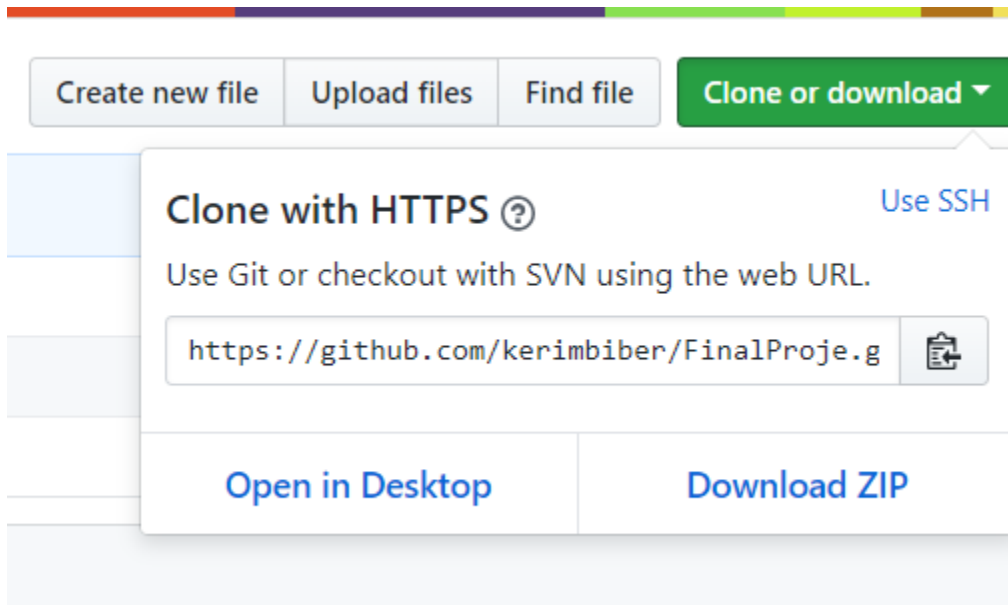
Ardından yeni bir repository oluşturuyoruz.



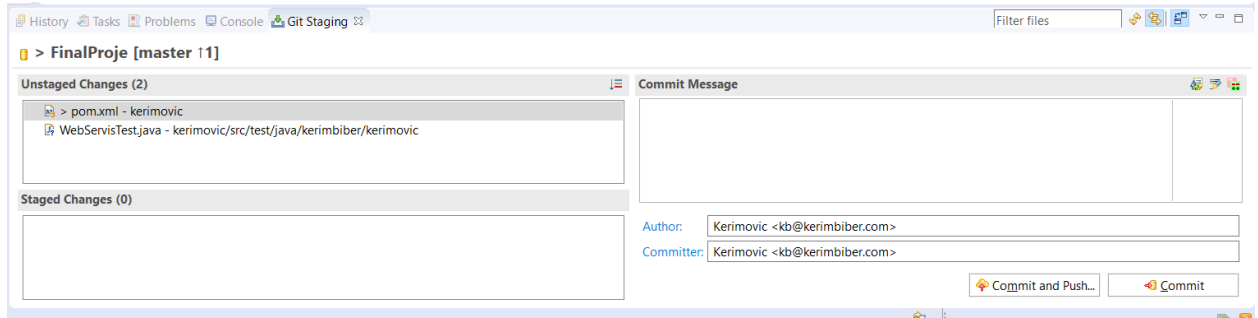
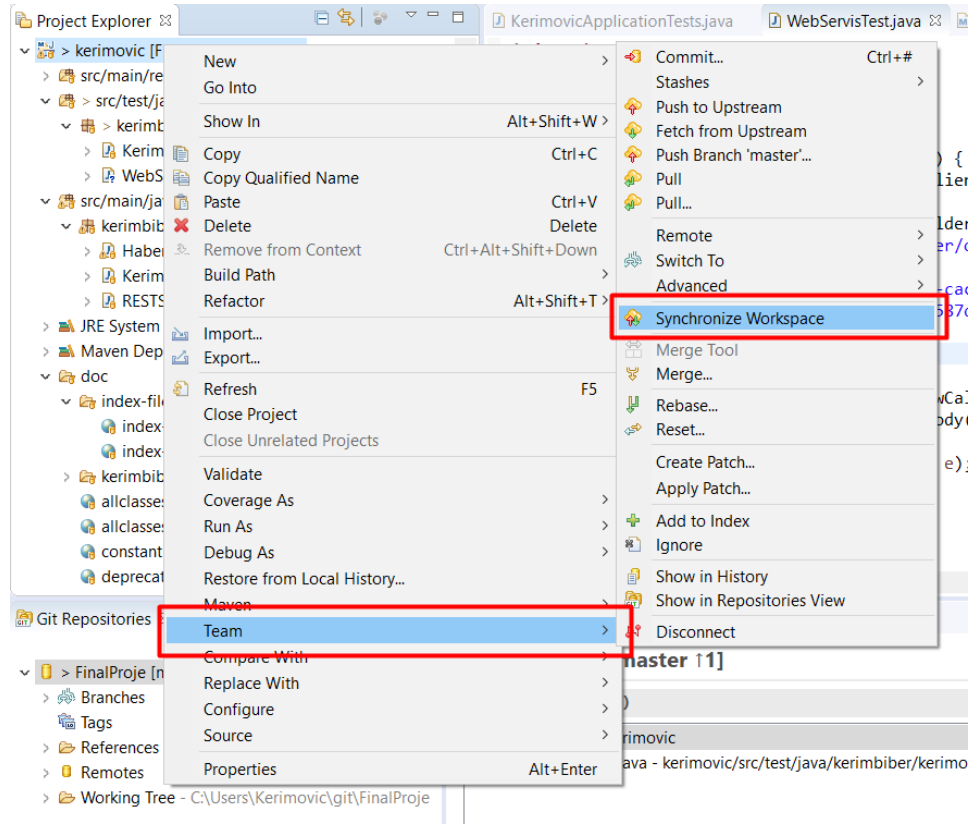
The image shows a GitHub user profile page for "kerimbiber". The page has a dark header with the GitHub logo, a search bar, and navigation links: "Pull requests", "Issues", "Marketplace", and "Explore". The user's profile is on the left, showing a repository icon and the name "kerimbiber". The main content area shows the "Repositories" tab with a search bar, filters for "Type: All" and "Language: All", and a green "New" button highlighted with a red box. Below the filters, there are three repositories listed: "FinalProje" (HTML, updated 14 minutes ago), "deneme" (updated 3 hours ago), and "H5150061" (Java, updated on 25 Apr 2017). The footer contains copyright information and links to "Terms", "Privacy", "Security", "Status", "Help", "Contact GitHub", "API", "Training", "Shop", "Blog", and "About".

Daha sonra IDE(Eclipse)'mizden projemizi oluřturuyoruz. Projemiz bittikten sonra githuba yklemek iin ise Window > Show View > Other kısmından Git Repositories ve Git Stagingi aıyoruz.

Sonra githubda ilgili repositoryyi aıyoruz. Buradan Clone or download kısmından ilgili linki kopayalayarak Git Repositoriese yapıřtırıyoruz ve github adresimiz ile giriř yapıyoruz.



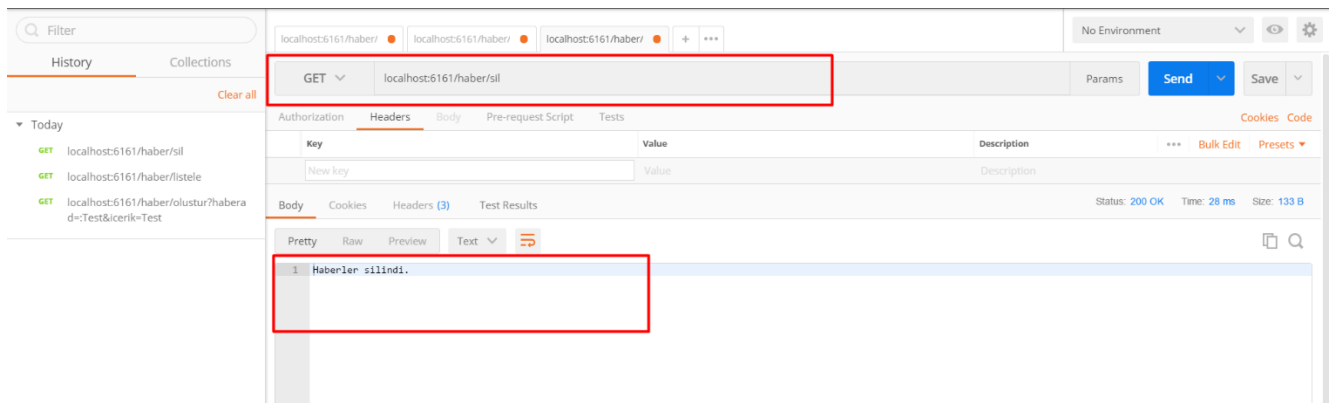
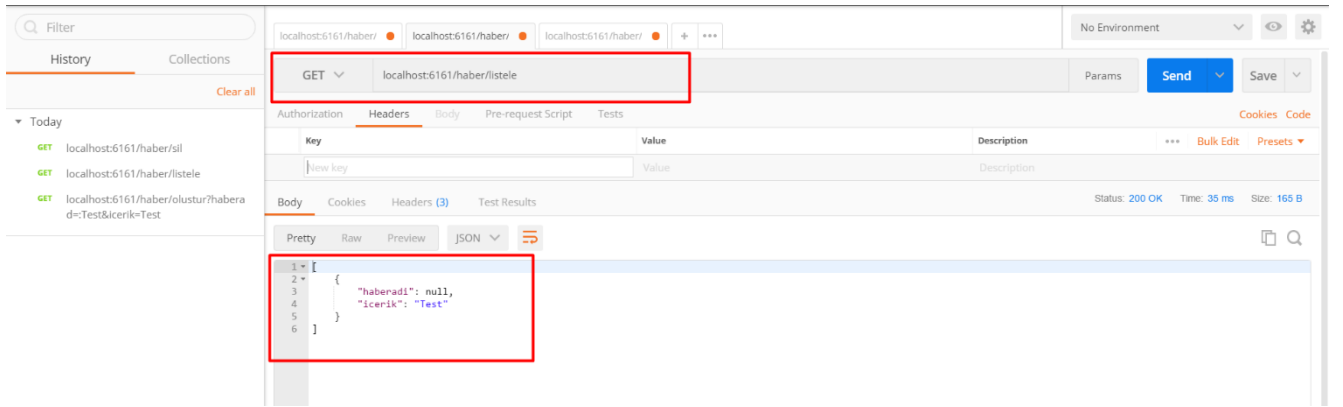
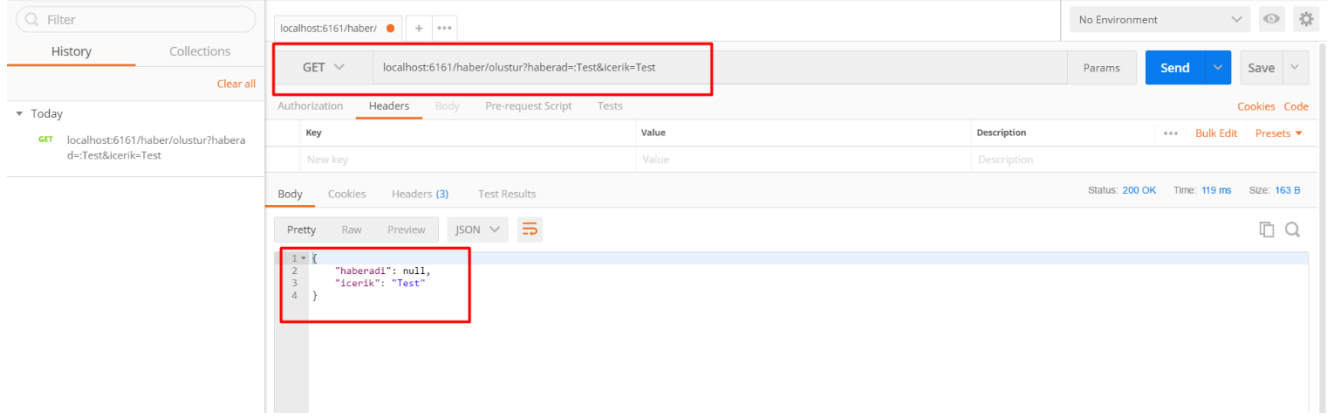
Daha sonra Eclipsten Projemize saė tıklayarak Team > Synchronize Workspace' e tıkladıktan sonra Git Staging kısmına yklenecek dosyalar geliyor.



Ardından bu dosyalar arasında githuba yüklemek istediklerimizi seçip Staged Changes kısmına sürüklüyoruz. Açıklama kısmını doldurduktan sonra Commit and Push ile github' yükleme yapabiliyoruz.

# POSTMAN TEST

Postman ile yapılan test aşağıdaki gibidir:



Proje Link: <https://github.com/kerimbiber/FinalProje.git>