

# Github Rehberi

bugra9

Kasım, 2016

# İçindekiler

1	Giriş . . . . .	2
2	Arayüz . . . . .	3
2.1	Code (Kod Yönetimi) . . . . .	3
2.2	Issues (Sorun Yönetimi) . . . . .	4
2.3	Pull Requests (Kod Ekleme İsteği Yönetimi) . . . . .	5
2.4	Projects (Proje Yönetimi) . . . . .	5
2.5	Wiki . . . . .	6
2.6	Pulse . . . . .	6
2.7	Graphs . . . . .	6
2.8	Settings (Ayarlar) . . . . .	6

# 1 Giriş

Github, [git sürüm kontrol sistemi](#) ile geliştirdiğiniz projelerinizi depolaması başta olmak üzere takım halinde bir proje geliştirmek için gerekli olan tüm servisleri bir arada sunar. Açık kaynaklı projeler için tüm servisleri ücretsiz olmakla beraber açık kaynaklı projeler için en çok tercih edilen platformdur. Bu rehberde tüm Github servislerini ayrıntılarıyla açıklayacağım. Anlatımda resimlere bağlantılar (R), sayfalara bağlantılar (S) isimlendirmesiyle yapılmıştır.

Giriş olarak önce temel git komutlarından bahsedelim. İlk önce github üzerinde uygulamamız için yeni bir depo oluşturalım. Bunu giriş yaptıktan sonra anasayfada sağ üst kısımda bulunan + simgesine tıklayıp açılan menüdeki "New repository" bağlantısına tıklayarak gerçekleştirebilirsiniz. (R) Depo oluşturma sayfasında uygulamanın ismini yazıp, "Initialize this repository with a README" kutucuğunu işaretleyip, "Add a license" kısmında "GNU General Public License v3.0" gibi bir lisansı seçtikten sonra "Create repository" düğmesine tıkladığınızda depo ilk verileriyle oluşmuş olacaktır.

Şimdi Github üzerindeki depoyu yerele kopyalayacağız. (R) Oluşturduğumuz deponun sayfasına gittiğinizde yeşil arka plana sahip bir düğmenin üzerinde "Clone or download" yazar. Bu düğmeye tıkladığınızda deponun git bağlantısı gösterilir. Bu bağlantıyı kullanarak bilgisayarımızda uçbirime

```
1 git clone gitBağlantısı
```

yazdığımızda uygulamanın adıyla bir dizin oluşacak ve veriler bu dosyanın içerisine kopyalanacaktır. Artık hep bu dizin üzerinde çalışacağız. Bu dizin içerisinde bir değişiklik yaptığınızda

```
1 git pull
2 git add .
3 git commit -am "Değişiklik ile ilgili açıklama"
4 git push
```

komutlarıyla değişikliği kaydedip Github üzerindeki depomuza gönderebiliriz. Başkalarının yaptığı değişikliği çekmek için ise

```
1 git pull
```

komutunu kullanabilirsiniz. En temel git komutları bunlar olup daha ayrıntılı bilgiye arama yaparak ulaşabilirsiniz. Bundan sonraki kısımda Github arayüzü incelenecektir.

## 2 Arayüz

(S) Tüm git ile geliştirdiğiniz projeyi Github üzerinde barındırabilirsiniz. Github bu dosyalarınızı ve dosyalara yaptığınız işlemleri detaylıca yönetebileceğiniz bir arayüz sunar. Bu arayüzü tek tek inceleyelim.

(R) Resimde gördüğünüz ilk kısmın sol tarafında proje sahibi ve proje ismi yazılıdır. Sağ tarafında ise üç düğme bulunur. Bunların ilki "Watch / Unwatch" düğmesi proje üzerinde yapılacak her değişikliğin size bildirilip bildirilmeyeceğini ayarlar. İkinci düğmemiz "Star / Unstar" düğmesi ile projeyi beğenip beğenmediğimizi belirtebiliriz. Size yararı dokunan projeleri beğenirseniz bu proje geliştiricilerine motivasyon sağlayacaktır. Üçüncü düğmemiz "Fork" ise projeyi çatallamamızı sağlar. Örneğin projeyi beğendiniz ve bu proje üzerinden kendiniz özel geliştirme yapmak istiyorsunuz. "Fork" düğmesi ile çatalladığınızda proje dosyaları sizin kullanıcınız üzerine kopyalanacak ve oradan devam etmenize imkan sağlayacaktır.

(R) Bir sonraki satıra geçtiğimizde resimde gördüğünüz gibi proje ile ilgili servisler arasında geçiş yapabileceğimiz bir menü görürüz. Bu menü tüm github servislerine giden bağlantıları içerir.

### 2.1 Code (Kod Yönetimi)

Proje sayfasını ilk açtığımızda "Code" sekmesi seçilidir. Burası dosyalarımızı / kodlarımızı web arayüzünden listeleyp yönetebileceğimiz alandır.

(R) Buranın ilk kısmında projemiz ile ilgili kısa açıklamayı görürsünüz. En sağda bulunan "Edit" yazısı ile de bu yazı değiştirilir.

(R) İkinci kısmında kodlarımızı yönetebileceğimiz birkaç sayfaya bağlantı vardır. Bunlar;

#### • Commit;

(S) Kodlarımız üzerinde yaptığımız değişiklik kayıt sayısını görürüz. Üzerine tıklayınca ise bu kayıtları listeleyp yönetebileceğimiz sayfa açılır.

(S) Listeleme sayfası tarihlere göre bölünmüş ve üst sırada en son değişiklikleri içerir. Bir kaydın üzerine tıklayarak değişiklikleri görebileceğimiz ve yorum yapabileceğimiz sayfa açılır. Burada kod ekranında + işareti ile birlikte yeşil arka plan ile gösterilen kodlar eklenenler, - işareti ile birlikte kırmızı arka plana sahip kodlar ise silinen kodları temsil ediyor. Bu sayfada anlamadığımız değişikliklere yorum yapabilir ve daha detaylı açıklanmasını isteyebilirsiniz. Bu tarz durumlar için yorum yapma kısmı kullanılabilir.

#### • Branch;

(S) Bu kısımda tüm dal sayımızı görürüz. Projeyi geliştirirken bazı kısımları ayrı geliştirebiliriz. Bu ayrı kısımlar için bir dal oluşturup geliştirmeyi burada yaparız. Geliştirme bittikten sonra ise ana dalımız ile birleştirip bu dalı sileriz. Ayrıca eklenti gibi ana proje üzerinde bulundurmamak istemeyip ayrı geliştirdiğiniz kodları ise farklı bir dalda tutabilirsiniz. Bu bağlantının üzerine tıklayınca tüm dallarımız listelenecektir.

#### • Releases;

Projenizin yayınlanabilecek seviyeye geldiğini düşünün. Artık sürümler çıkarmak ve bunları yayınlamak istiyorsunuz.

(S) Bu kısımda yazılıma ait tüm sürümlerin sayısını görürsünüz. Bağlantıya gittiğinizde tüm sürümler listelenir. Bu listelemede sürüm numarası, açıklama ve indirme bağlantıları bulunur.

(R) Eğer yeni bir sürüm çıkarmak istiyorsanız "Draft a new release" düğmesi ile ekleme sayfasına gidebilirsiniz. Bu sayfayı doldurup "Publish Release" düğmesi ile sürümünüz yayınlanır.

Numaralandırmayı planlı yapmak ve bir kurala uymak önemlidir. Böylece siz ve yazılımı kullanan kişiler bu numaradan tahmini olarak neler yapıldığını anlayabilir. Nasıl numaralandırma yapacağınızda özgürsünüz. Genel olarak numaralandırma iki şekilde yapılır.

#### **Belirli aralıklarla yayınlanan sürümler:**

Bu tarz sürümlerde numara olarak ay ve yıl verilir. Örn: 16.04

### **Belirsiz aralıklarla yayınlanan sürümler:**

Örneğin v1.2.4 numaralandırmasını inceleyelim. İlk kısım alt yapı değişikliği olup birçok durumda uyumsuzluk olabilecek durumlarda arttırılır. İkinci kısım uyumsuzluk yaratmayacak gelişmelerde arttırılır ve son kısım ise düzeltme yapılan durumlarda arttırılır. Eğer yazılımınız alfa ya da beta gibi kararlı sürüm öncesi dönemdeyse numaralandırmayı v0.2.4 şeklinde yapabilirsiniz. Burada 0 dışındaki ilk numara alt yapı ve yeniliklerde, ikinci kısım ise yenilik ve düzeltmelerde arttırılabilir.

### **• Contributors;**

(S) Burada projeye kaç kişinin katkıda bulunduğu gösterilir ve üzerine tıklayınca katkıda bulunanlar ile istatistik paylaşılır.

(R) Ana kısımda tüm dosyalarımız listelenir. Bu listelemede dosya ismi, son değişiklik açıklaması ve son değişiklik zamanı yer alır. Burada bir dizine tıklayıp içindekileri listeletebilir ya da bir dosyaya tıklayıp içeri görüntüleyebilirsiniz.

(R) Listeleme üst kısmında birkaç düğme bulunur.

- Bu düğmelerin ilki dal seçmemizi sağlar. Seçtiğimiz dala göre dosyaların listelenmesi değişir.
- "New pull request" düğmesi ile bir dalı ana dal veya diğer dallarla birleştirilmesi sağlanır.
- "Create new file" düğmesi ile yeni dosya oluşturulur. Github üzerinde klasör mantığı yoktur. Sadece dosyalar vardır ve bu dosyaların bulunduğu dizinlere göre gruplama yapılır. Dolayısıyla deneme diye klasör oluşturup içine benioku.txt diye dosya atmak isterseniz sadece yeni dosya oluşturup dosya yolunu deneme/benioku.txt şeklinde belirtin.
- "Upload files" düğmesi ile dışarıdan dosya yükleyebilirsiniz.
- "Find file" düğmesi dosyalarınız içerisinde arama yapmanızı sağlar.
- "Clone or download" düğmesi ile açılan kısımdaki bağlantı ile `git clone https://bağlantıAdresi klasörAdı` şeklinde bilgisayarınıza kopyalayabilirsiniz. Yine aynı kısımda bulunan "Download ZIP" bağlantısı ile dosyaları sıkıştırılmış olarak bilgisayara indirirsiniz.

En alt kısımda proje hakkında detaylar bulunur. Bu bilgiler readme ya da readme.md dosyasından çekilir.

## **2.2 Issues (Sorun Yönetimi)**

Yazılımınızı yazdınız ve yayınladınız. Şimdi kullanıcılar için geliştiricilerle, geliştiriciler için de kullanıcılarla etkileşime geçme zamanı. Hata kaydı mı bırakmak istiyorsunuz, bir öneride mi bulunacaksınız, bir özellik mi eklenmesini istiyorsunuz, hepsini bu bölümde yapabilirsiniz.

(S) "Issues" sayfasını açtığınızda proje için açık olan tüm kayıtlar listelenir. (R) Listenin hemen başında bu kayıtları süzgeçten geçirebileceğiniz seçenekler bulunur. Soldaki alan, açık ile kapatılmış kayıtlar arasında seçim yapmanızı sağlar. Sağdaki alanlar ise proje yöneticisinin kayda atadığı özelliklere göre kayıtları süzgeçten geçirmenize olanak verir. Bu özellikler ileride açıklanacak.

Kullanıcılar yeni bir kayıt oluşturmak istiyorsa sağ üstte yeşil arka plana sahip "New issue" düğmesi ile bunu yapabilirler.

(S) Yeni kayıt oluşturmak için açılan sayfada başlık ve konuyla ilgili açıklama girilir. "Submit new issue" düğmesi ile kaydın gönderilmesi sağlanır.

### **Proje Yöneticileri için;**

Belirli bir süre sonra ve proje büyüdükçe kayıtların birbirine girdiğini ve size okumada zorluk çıkardığını farkedeceksiniz. Bunun için kayıtlara özellik atayarak belirli bir düzen içinde durmasını sağlayabilirsiniz.

(R) "Issues" sayfasını açtığınızda resimde gördüğünüz gibi "Labels" ve "Milestones" düğmeleri ile karşılaşacaksınız. Bunlar en temel iki özellik olup "Labels" özelliği etiket gibi, "Milestones" özelliği ise kategori mantığında çalışır. Bu özelliklere istediğiniz değerleri ekleyebilirsiniz ama genelde "Labels" özelliğinde hata kaydı, yeni özellik isteği, soru gibi değerler girilir. "Milestones" kısmında ise yazılımın sürümleri girilir ve kayıtlar sürümlere göre kategorize edilmiş olur. Yeni bir sürüm verdiğiniz "milestones" değerine sahip tüm kayıtlar tamamlandığında ise o sürüme geçmiş olursunuz.

(S) Örnek bir "Labels" tanımlaması

(S1) (S2) Örnek bir "Milestones" tanımlaması

(S) Bu özelliklerin değerlerini belirttiyseniz şimdi mevcut kayıtlara bu özellikleri atayabilirsiniz. Bir kaydın içine girdiğinizde belirleyebileceğiniz tüm özellikler sağ tarafta listelenir. Burada "Labels" ve "Milestones" kısmında daha önce belirlediğimiz değerlerden ilgili olanlar seçilir. Eğer bu kayıt ile ilgili ilgilenmesi gereken kişiler varsa "Assignees" alanından bu kişiler seçilir.

Evet artık tüm kayıtları bir düzende tutup hızlıca projemizi geliştirebiliriz. Eğer projeyi izlemeyi/takip etmeyi seçmişseniz her kayıt açıldığında size eposta ve bildirim gelir. Seçmemişseniz sadece isminizin geçtiği ve size atanan kayıtlar için bildirim gelir.

## 2.3 Pull Requests (Kod Ekleme İsteği Yönetimi)

Açık kaynaklı yazılımların en büyük özelliği herkesin bir şey ekleyerek katkıda bulunmasıdır. Tek başınıza ya da takım halinde proje geliştirdiğinizi düşünün. Sizin kodlara erişiminiz vardır ama dışarıdaki kişiler direk kodlara erişemez. Bazı kullanıcılarınız hata kaydı açmak ya da yeni özellik istemek yerine bu özelliği kendi yazıp ya da hatayı kendi düzelterek katkıda bulunmak ister. İşte bu şekilde erişiminiz olmayan projede kod ekleme/düzeltilme yapmak istediğinizde veya proje yöneticisiyseniz sizin projenize gelen bu tarz istekleri yönetmek için bu sayfayı kullanırsınız.

Bu bölüm iki kısımda incelenecektir. İlk olarak başka bir projenin kodlarını düzeltmeyi ikinci olarak başkasının düzeltip gönderdiği kodların yönetilmesi anlatılacaktır.

### a) Erişiminizin olmadığı proje üzerinde değişiklik yapmak

İlk önce projeyi kendi üzerimize çatallamamız gerekmektedir. Proje sayfasında sağ üstte bulunan "Fork" düğmesi ile proje çatallanır. Daha sonra kendi hesabımız üzerinde oluşan bu çatallanmış projeye gidilip istenilen değişiklikler yapılır. (R) İşlemlerimiz bitince sıra ana projeye birleştirmek için dal seçiminin yanında bulunan "New pull request" düğmesine basılır. (R) Açılan sayfada yaptığımız tüm değişiklikler listelenir. Eğer bir sorun yoksa "Create pull request" düğmesi ile bu yaptığımız değişiklikler yollanır. Eğer proje yöneticisi uygun görürse bu değişiklikleri onaylar ve kodlar ana proje üstüne geçmiş olur.

Eğer tek bir dosyada değişiklik yapmak istiyorsanız tek adımda yukarıda yazanları gerçekleştirebilirsiniz. Bunun için ana proje üzerinde değiştirmek istediğiniz dosyayı açıp değiştirin. Altına yaptığımız değişikliği yorumlayarak "Propose file change" düğmesine basın. İsteğiniz proje yöneticisine ulaşacaktır.

### b) Gönderilen kod ekleme isteklerini yönetmek

(S) "Pull Requests" sayfasından proje için gönderilmiş tüm kod ekleme isteklerini herkes listeleyebilir. (S) İsteğin üzerine tıklayınca açılan sayfada ilk önce bu değişiklikler ile ilgili açıklama ve altında da bununla ilgili yorumlar görüntülenir. Burada yazılarak kodun istenilen hale getirilmesi sağlanır.

Yorumların üstünde üç sekme bulunur. (S) Bunların ikincisi "Commits" sekmesi ile değişiklik kayıtları, (S) "Files Changed" sekmesi ile de dosyalarda yapılan değişiklikler görüntülenir.

Eğer proje yöneticileri ilgili değişiklikleri uygun görürse "Merge pull request" düğmesi ile kodları ana projedeki kodlar ile birleştirilmesini sağlarlar.

## 2.4 Projects (Proje Yönetimi)

(S) Bu kısım "issues" ve "pull requests" kayıtlarını belirli sütunlara ayırarak daha kolay organize edilmesini sağlar. Sürük- le bırak şeklinde bu kayıtları oluşturduğunuz sütun altına taşıyabilirken bu sütunlara notlar da ekleyebilirsiniz. Ayrıca geliştirme planlarınızı burada yapabilir, yol haritanızı belirleyebilirsiniz. Kısaca projenizin kontrol odası gibi düşünürsek yanlış olmaz.

Örneğin bu Sudo Portal uygulamasının proje sayfasını inceleyelim. Öncelikle bu uygulamada hem makaleler hem de kodlar bulunmaktadır. Açılan hata kayıtlarını birbirinden ayırmak için etiket kullandığımızı söylemiştik. Burada maka-

leler için ayrı etiketler, yazılım için ayrı etiketler oluşturursak işin içinden çıkılmaz. Ayrıca proje büyüdükçe ve yazılımı parçalar halinde geliştirmeye başladıkça her bir parça için etiket oluşturmak yerine burayı kullanabiliriz.

(S) Örneğin burada makaleler için bir proje oluşturmuşuz. Bu proje ile makaleler için açılmış "issue" ve "pull request" kayıtlarını, güncellenmesi ve gözden geçirilmesi gerekenler şeklinde kategorize ederek daha rahat görmemizi sağlıyoruz. Ayrıca makaleler için ayrı bir etiket oluşturmak zorunda kalmadık.

(S) Burada ise yazılımın yönetimi için bir proje oluşturmuşuz. Bu şekilde açılan kayıtları belirli aşamalara bölerek daha kolay yönetmeyi amaçlamışız. Bir sorunu çözmek için geçirilecek süreç sol sütundan başlıyor ve en sağ sütunda son buluyor.

Daha önce yazdığım gibi yazılım büyüdükçe yazılım parçalara bölünerek daha kolay geliştirilmesi sağlanır. Burada bu parçalara özel projeler oluşturarak daha kolay yönetebilirsiniz.

## 2.5 Wiki

Bu bölümü yol haritanızı ve mevcut durumu göstermek, yazılımınız hakkında belge hazırlayıp sunmak amacıyla kullanabilirsiniz. Yazılar başlık ve markdown biçiminde anlatım yapabileceğiniz bir alandan oluşur.

## 2.6 Pulse

(S) Son bir hafta içinde yapılan "issue" ve "pull request" kayıtlarındaki durumu gösterir. Bu zaman dilimi için 24 saat, 3 gün ve 1 ay seçeneklerinden biri de seçilebilir.

## 2.7 Graphs

Proje üzerindeki tüm istatistiklerin grafiksel olarak gösterildiği alandır.

- (S) "Contributors" sekmesi ile projeye katkıda bulunanlar ile ne kadar katkıda buldukları gösterilir.
- (S) "Traffic" sekmesi ile proje sayfasının görüntülenme istatistikleri paylaşılır.
- (S) "Commits" sekmesi ile hangi zamanlarda ne kadar "commit" gönderildiği grafiksel olarak gösterilir.
- (S) "Code frequency" sekmesi ile hangi zamanlarda ne kadar kod ekleme/çıkarması yapıldığı grafiksel olarak gösterilir.
- (S) "Punch card" sekmesi ile gönderilen "commit" sayısının haftanın günlerine göre dağılımı gösterilir.
- (S) "Network" sekmesi ile proje ve proje çatallarının "commit" yapılan zamanları gösterilir. Burada tüm çatallamaları takip edebilirsiniz.
- (S) "Members" sekmesi ile tüm proje çatalları listelenir.

## 2.8 Settings (Ayarlar)

### 2.8.1 Options

- (R) "Settings" alanında projeyi yeniden adlandırabilirsiniz.
- (R) "Features" alanında listelenen Github servislerini aktif/pasif hale getirebilirsiniz.
- (R) "Merge button" alanında birleştirme işlemlerini ayarlayabilirsiniz. Daha sonra detaylı bir ekleme yapılacak.
- (R) "Github Pages" alanı dışarıya yayın yapmak istediğiniz durumda bu yayın ile ilgili ayarları içerir. Bu konu ileride anlatılmış olup burada sadece ayarlarından bahsedilecektir.
- "Source" alanında hangi daldaki kodların yayınlanacağı belirlenir.
- "Custom domain" ile kendi alan adınızla yayın yapabilirsiniz. Oraya bu alan adınızı yazabilirsiniz. Bu konu ileride ayrıntılı olarak anlatılmıştır.

- “Overwrite site” kısmında Github’ın otomatik sayfa oluşturucu servisini başlatabilirsiniz.
- “Enforce HTTPS” seçeneği ile de yayının sadece https üzerinden yapılmasını söyleyebilirsiniz.
- (R) “Danger Zone” alanı ile projeyi dışarıya kapatabilir, başkası üzerine aktarabilir veya silebilirsiniz.

## 2.8.2 Collaborators

(R) Buradan istediğiniz kişilere projenize erişim izni verebilirsiniz. İzin verdiğiniz kişiler kodlar dahil olmak üzere projeniz üzerindeki tüm Github servislerini (“issues”, “pull requests”, “projects”, ...) yönetebilir.

İzin vermek istediğiniz kişiyi kullanıcı adından, isminden ya da eposta adresinden yakalayıp ekleyebilirsiniz. Siz yazmaya başladıkça yazdıklarınıza uygun olan kullanıcılar aşağıda listelenecek, bunlardan birini seçince de eklenmiş olacaktır. İzin verdiğiniz kişiyi istediğiniz zaman silerek iznini iptal edebilirsiniz.

## 2.8.3 Collaborators & teams

Bu seçenek kullanıcı üzerinde değilde organizasyon üzerinde bir depo oluşturduğunuzda bu deponun ayarlarında görünür.

(R) Burada üst kısımda anlatılmış “Collaborators” alanına ilave olarak takım izinlerinin yapılacağı yer ile organizasyon üyelerinin yetki durumunu anlatan bir yazı bulunur. Takım bölümünden daha önce oluşturulmuş takımlara projeye erişim izni verilir.

## 2.8.4 Branches

Buradan projede kullanılan dallardan birisi ön tanımlı olarak seçilebilirken, istenilen dallar düzenleme yapılmasına karşı korumaya alınabilir.

## 2.8.5 Webhooks

Github içerisinde herhangi bir olay meydana geldiğinde bunu dışarıdaki bir servise bildirilmesi buradan ayarlanır. Örneğin Sudo Portal projesinde her sürüm çıkarıldığında github bunu ubuntu-tr.net servisine bildirir. Bu bildiri alınınca kodlar githubdan çekilir, jekyll aracılığıyla statik içerik oluşturulur ve bunlar Sudo Portalın alan adı altında yayınlanır.

(R) “Add webhook” düğmesine tıklayınca resimdeki gibi ekleme sayfası bizi karşılar.

- “Payload URL” alanına bildirim gönderilecek harici servisin bağlantısı yazılır.
- “Content type” kısmından gönderilecek içeriğin json veya x-www-form-urlencoded biçimlerinden hangisi olacağı söylenir.
- “Secret” alanına girilecek rasgele bir değerle servis, bildirim Github’dan geldiğini doğrulayabilir.
- (R) “Which events would you like to trigger this webhook?” kısmından ise hangi olayların bildirileceği seçilir. Buradaki olaylar daha sonra açıklanacaktır.

## 2.8.6 Integrations & services

(R) Github, harici bağlantılarla iletişim olayını bir düzene sokmak için servis olayını oluşturmuştur. Github üzerinde oluşturulmuş servislere izin verdiğinizde bu servisler burada listelenir. (R) İçine girdiğinizde ise bu servisler ile ilgili ayarlar yapabilirsiniz.

Bu konu oldukça kapsamlı olup daha sonra anlatılacaktır.