# Giriş

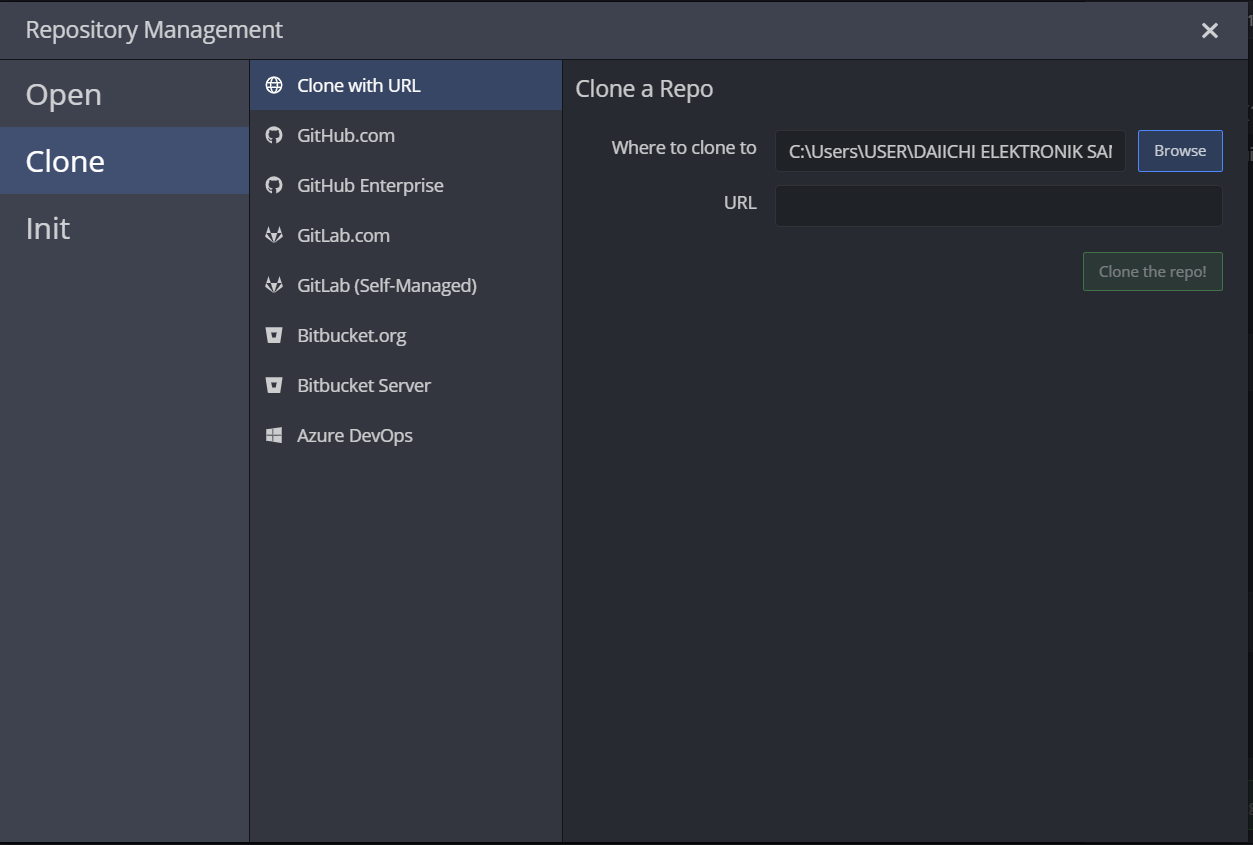
* <https://git-scm.com/downloads> adresinden Git programını işletim sisteminize uygun olanını kurun.
* <https://github.com> adresinden hesabınız yok ise hesap oluşturun.

--İleride Bitbucket olarak değiştirilebilir.

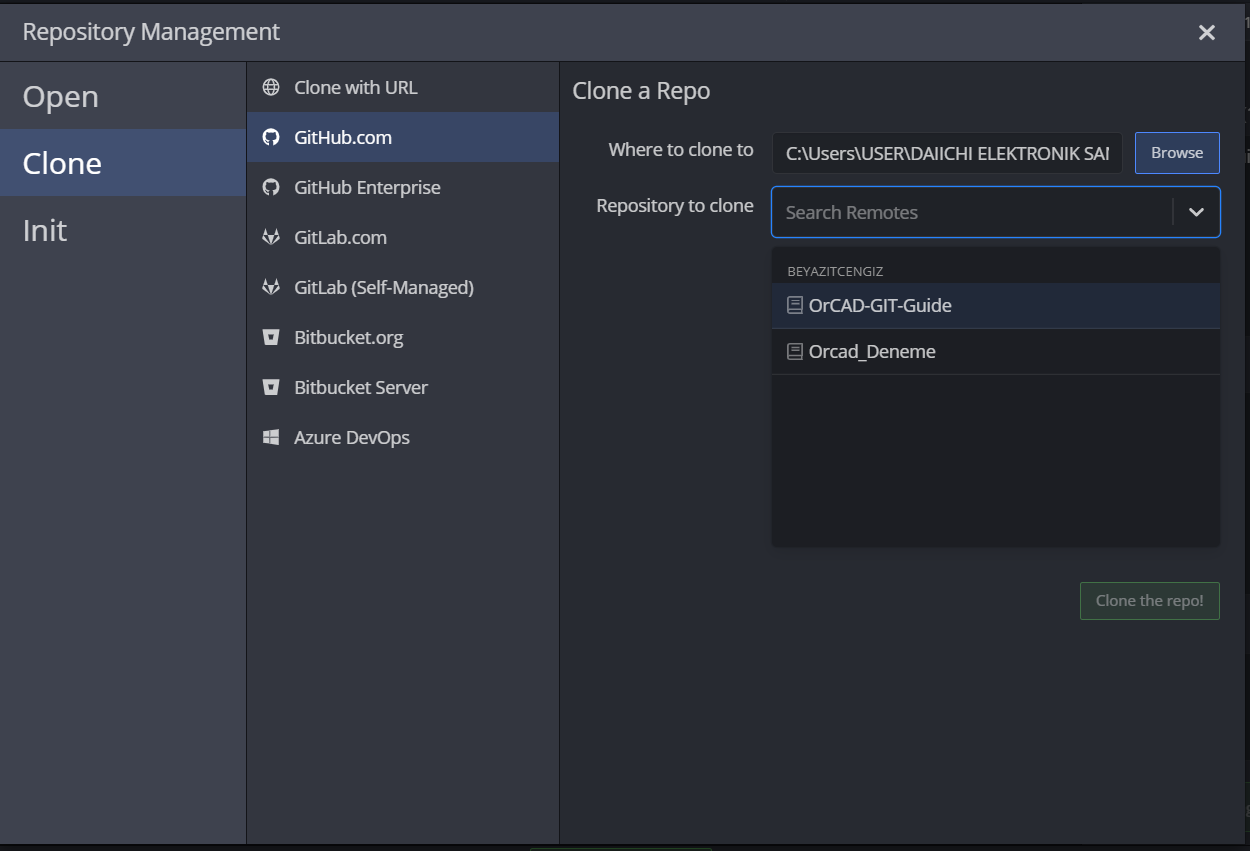
* <https://www.gitkraken.com> adresinden GitKraken programını işletim sisteminize uygun olanını kurun.
* GitKraken kurulumundan sonra GitHub hesabınız ile eşleştirin.

# GitKraken Clone Repo

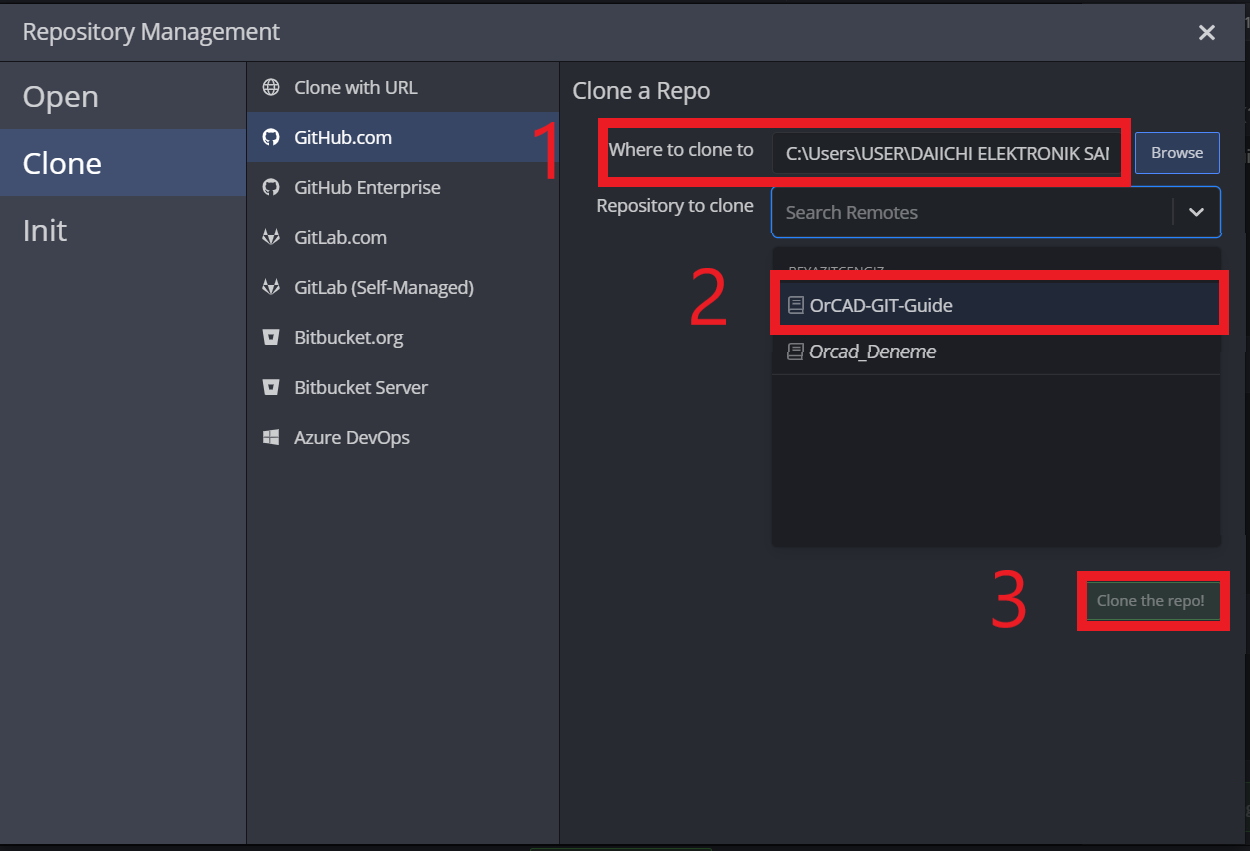
* Ctrl+N kısayolu kullanılarak veya File-Clone Repo yolu kullanılarak aşağıdaki resimde görmüş olduğunuz pencere açılır.



* 2. Sıradaki GitHub sekmesi seçilir. Aşağıda resmini görmüş olduğunuz penceredeki gibi sizin kullanabileceğiniz repolar listelenmektedir. (Repoların görüntülenebilmesi için ilgili reponun sizin ile paylaşılmış ve izin verilmiş olunması gerekmektedir.

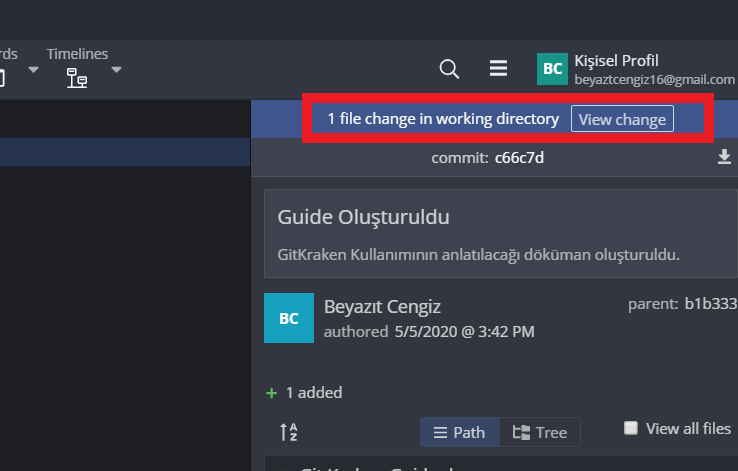


* Dosyaların nereye kopyalanacağını ve hangi repodan kopyalanacağını seçtikten sonra Clone the Repo butonuna basıyoruz.

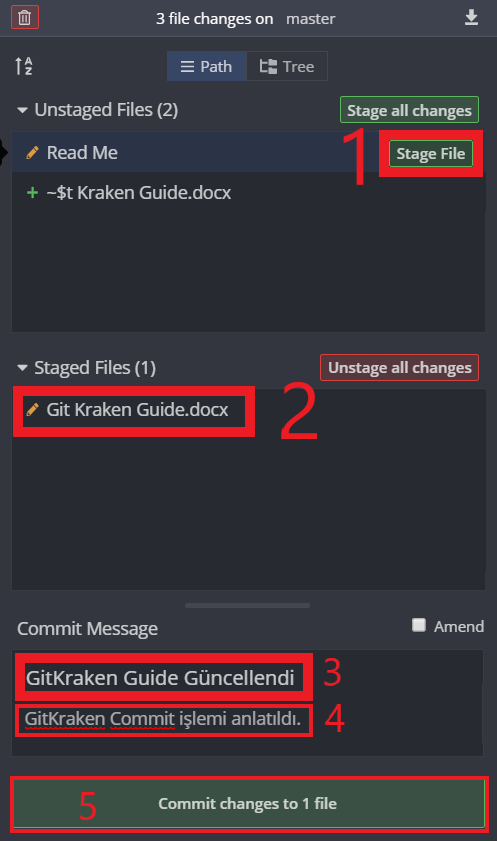


# GitKraken Commit

* Yaptığımız değişikliklerin uygun olduğunu düşünüyorsak projemizi kaydedip kapatıyoruz. Ardından Commit yapmak için GitKraken programını açıyoruz. Klasörlerinizde değişiklik olduğu için sağ üst köşede aşağıdaki resim benzeri görülecektir.

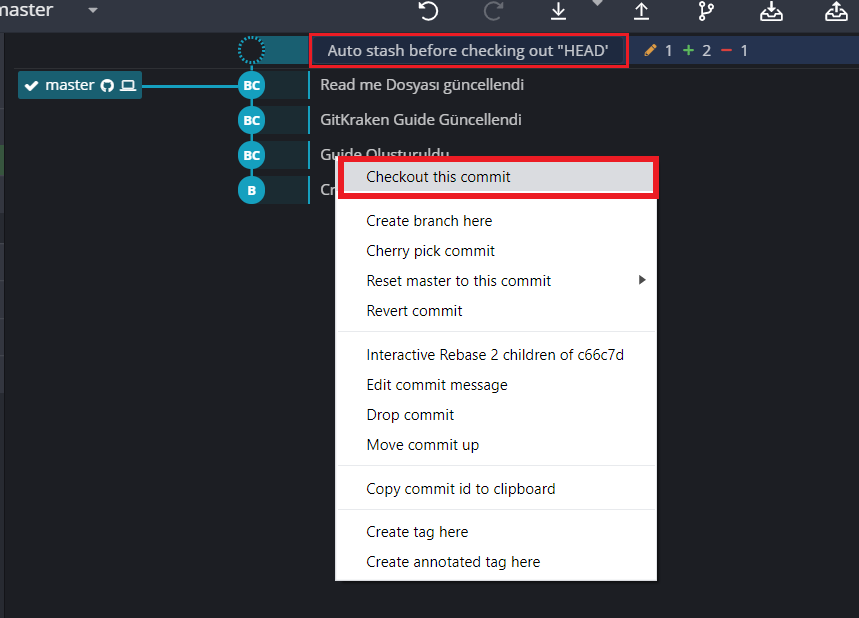


* Yaptığımız değişiklikler aşağıdaki biçimde listelenecektir.
* 1- Değişiklik yaptığımız dosyalar içerisinden Coommit yapmak istediklerimizin üzerine gelip Stage veya Stage File diyerek Commit yapılacak dosyalar arasına alıyoruz
* 2- Commit yapılacak dosyaları kontrol ediyoruz. İstenmeyen hatalı dosya var ise çıkartıyoruz.
* 3-Commit başlığını yazıyoruz.
* 4-Commit mesajını yazıyoruz.
* 5-Butona basarak commit işlemini gerçekleştiriyoruz.



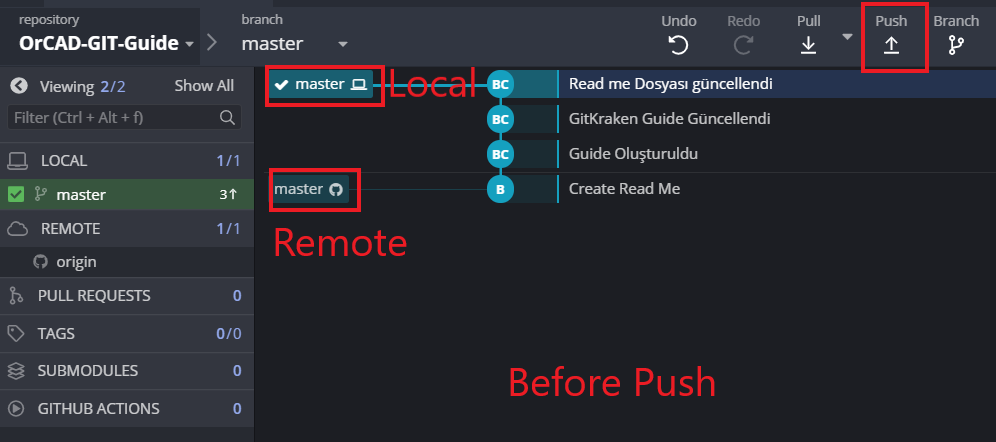
# GitKraken Checkout

* Git Checkout komutu yapılmış commitlerden herhangi birine dönmenizi sağlar. Eğer bir değişikliği tamamlamadan başka bir commite dönmek zorunda iseniz (inceleme amaçlı) aşağıda anlatılan Stash-Pop yapısı kullanılmalıdır. Checkout yaptığınızda otomatik olarak stash yapılır. Tekrardan değişikliğe dönmek için Pop komutu kullanılmalıdır.



# GitKraken Push

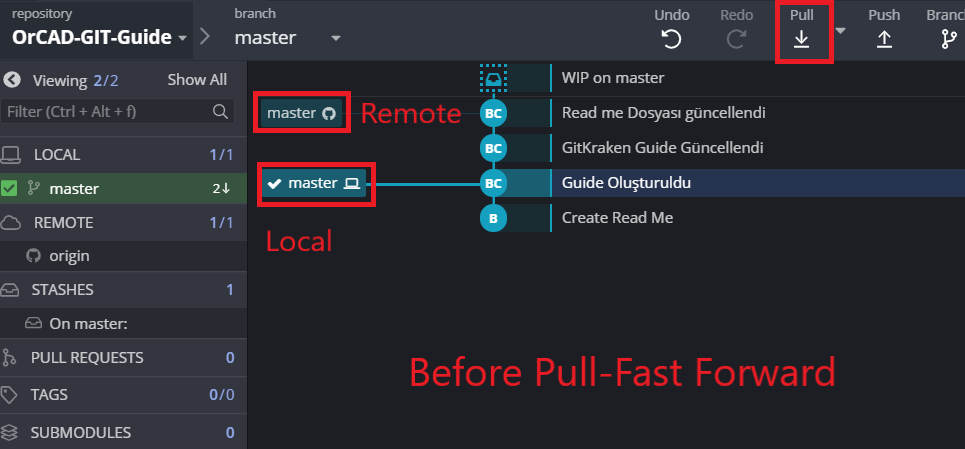
* Yaptığımız değişikler kendi kopyamızda geçerli olmasına rağmen bütün kullanıcıların aynı değişiklikleri görünteleyebilmesi ve son kopya üzerinden devam edebilmeleri için yaptığımız değişiklikleri remote repo’ya aktarmamız eşlememiz gerekmektedir.
* Yanında tik olan ve PC logolu master local master idir. Yanında github logosu olan master ise remote master idir. Biz Local ile remote dosyalarını eşlemek için “Push” komutunu kullanacağız.





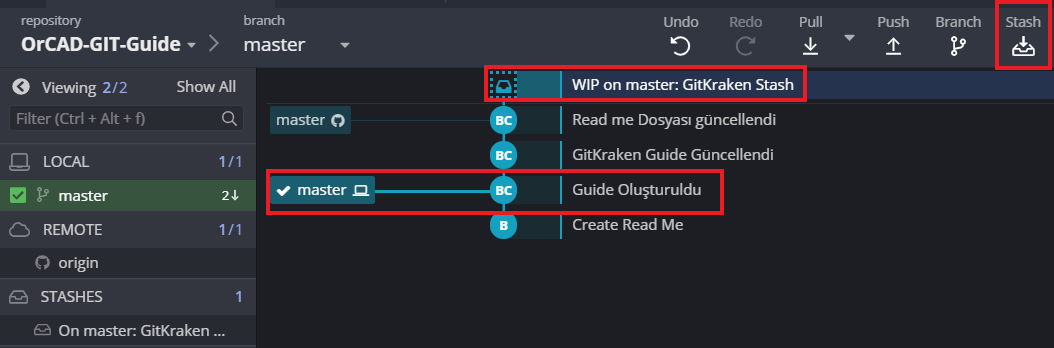
# GitKraken Pull-Fast Forward

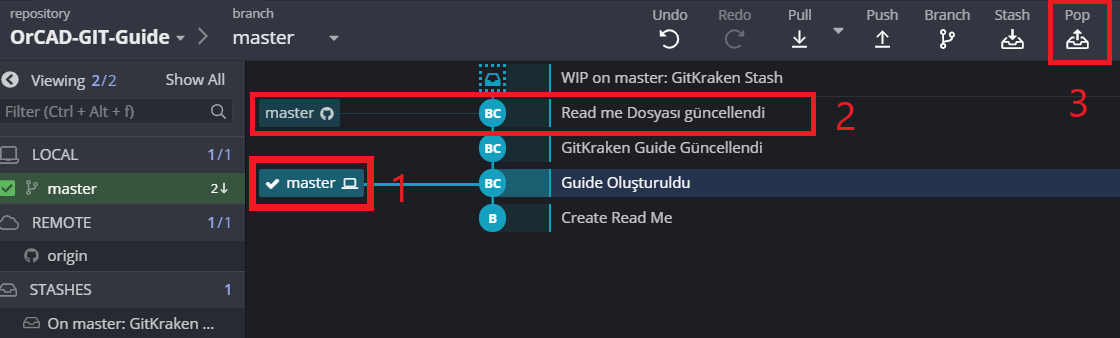
* Ekip çalışması şeklinde yürütülen işlerde diğer ekip üyeleri çalışmalarını tamamlayıp remote repo’ya push işlemini gerçekleştirebilirler. Bu durumda sizin local de tuttuğunuz kopyanız eski sürüm olarak kalmış olacaktır. Remote repodan en son sürümün indirilmesi için Pull-Fast Forward komutu kullanılır.



# GitKraken Stash-Pop

* Git Stash komutu henüz tamamlanmamış ve commit edilmemiş değişiklikler esnasında eski versiyonlara dönmemiz ve kontrol yapmamız gerektiğinde kullanmamız gereken komuttur. Stash komutu commit yapılmamış değişikliği saklamamıza yardımcı olur ve kullandığımız branch’i sonlandırır.
* Stash işlemi değişiklik yapılan dosyalar açık iken yapılmaz. Aşağıdaki resimde de görüldüğü gibi stash işleminden sonra Stash işlemi yapıldıktan sonra HEAD yada master başka bir commite taşınabilir. Açılan commit incelenebilir. Fakat değişiklik yapılması önerilmez. Daha sonra yapmış olduğunuz değişiklikler ile birleştirme ( txt tabanlı dosyalar hariç) mümkün değildir.



* Git Pop komutu için aşağıdaki sıralama takip edilmelidir.
* 1-Local master veya Head dosyası dosyası stashın yapıldığı commite checkout veya pull komutu ile götürülür. Pop işleminden önce stash işleminin yapıldığı committe olunması gerekmektedir. (İşlemciler dersindeki stack yapısındaki push-pop mantığına benzer)
* 2-3 Master stash yapılan commite döndürüldükten sonra POP komutu uygulanır.
* 

# GitKraken Tag

* GitKraken Tag önemli görülen veya kilit taşı (örn.B Sample, C Sample) commitlerin daha kolay bulunması ve izlenebilmesi için önemlidir.
* 