



Bu bölümde git ' in en temel komutlarını göreceğiz;

Önceki derslerde lokal depo oluşturmayı görmüştük.Hatta "lokaldepo" isminde bir klasör oluşturup "git init" komutu ile git deposu haline getirmiştik.Bu klasör üzerinden anlatımlarıma devam edeceğim.Terminal ekranımızı açalım ve "lokaldepo" dizinimize gidelim ve aşağıdaki kodu yazalım :

```
git status
```

Komutu çalıştırdığınızda aşağıdaki yazıyı göreceksiniz.(Tabi önceden kurcalamadıysanız ☐ )

```
On branch master
No commits yet
nothing to commit (create/copy files and use "git add" to track)
```

"git status" komutu bize yol gösterici komuttur son durumu ve olası yapılması gerekenleri bize gösterir.Yukarıda ki sonuca göre :

Master dalında olduğumuzu(Daha sonra açıklayacağım) klasörde herhangi bir değişiklik yapılmadığından yapılmış commit olmadığını(versiyonlama yapılmadığını) ve öneri olarak yapılması gerekenleri göstermektedir...

Şimdi klasörümüzde("lokaldepo") bir değişiklik yapalım ve bir dosya ekleyelim developers.txt dosyamızı nano ile oluşturup içine istediğiniz geliştiricinin adını yazın ben kendiminkini ☐ yazıyorum ...

```
nano developers.txt
```

Bu kod dan sonra istediğiniz ismi yazıp ctrl-s ve ctrl-x ile nano dan kaydedip çıkış yapmış olursunuz.

Dosyamızı kaydettik ee şimdi ne değişiklik oldu ? nasıl bakacağız ?

biraz önce bahsetmiştik “git status” ile son duruma bakalım ;

```
git status
```

Sonuç :

```
On branch master

No commits yet

Untracked files:
  (use "git add <file>..." to include in what will be committed)

    developers.txt

nothing added to commit but untracked files present (use "git add" to
track)
```

sonuç biraz değişti ...açıklamayı inceleyelim : bir dosya değişikliği olduğunu bu dosya isminin “developers.txt” olduğunu ve bu dosya ile ilgili yapılması gereken önerileri açıklamamızda görmekteyiz.

Şu anda “developers.txt” dosyamız daha önce bahsetmiş olduğum “Git - İş Akışı - Çalışma Mantığı” bölümünde ki adımlardan **workspace** bölümünde ... git ise bize bu dosyayı versiyonlamadan önce **index** bölümüne atmamız için gerekli komutu dahi vermiş ;

```
git add developers.txt
```

yukarıda ki kod ile **workspace** de olan “developers.txt” dosyamızı **index** bölümüne atmış olduk.Bulduğunuz klasörde ki tüm dosyaları **index** e atamak için aşağıdaki kodu kullanabilirsiniz ;

```
git add .
```

Yukarıda ki kodda “.” komutu bulunduğunuz klasör içerisinde ki tüm dosyalar manasına gelmektedir.

Şimdi tekrardan “git status” komutunu çalıştıralım , son duruma ve ne yapmamız gerektiğine bakalım ;

```
git status
On branch master
No commits yet
Changes to be committed:
  (use "git rm --cached <file>..." to unstage)
    new file:   developers.txt
```

Yukarı da yeni bir dosyanın index e eklendiğini ve yukarı da verilen komut ile silinebileceği bilgisini vermekte ...Şu an dosyamızı commit etmeye hazır duruma geldik.(Versiyonlamaya).

Şimdi commit komutunu görelim :

```
git commit -m "ilk sürüm versiyonlama"
```

burada tırnak içersine aldığımız bölümde versiyon ile ilgili tanımlayıcı notlarımızı girebiliriz ... kodu çalıştırdığımızda aşağıdaki gibi bir sonuç alacaksınız :

```
[master (root-commit) 3b87c81] ilk sürüm versiyonlama  
1 file changed, 1 insertion(+)  
create mode 100644 developers.txt
```

Burada master dalına “ilk sürüm versiyonlama” notu ile developers.txt dosyasının **Local Repository** bölümüne taşındığını yani versiyonlandığını görmekteyiz.Şimdi ise bu commit işlemini doğrulayalım ;

```
git log
```

Komutunu çalıştıralım.Sonuç ;

```
commit 3b87c818004c43f4a3674e774c457b6102ea838a (HEAD -> master)  
Author: Serkan Kaya <serkankayatr@hotmail.com>  
Date: Fri Feb 15 11:05:17 2019 +0300  
    ilk sürüm
```

Sonuç olarak ilk commit işlemimizi yapmış bulunmaktayız.Log ekranında commit id mizi daha önce ayarlarını yaptığımız e-posta ve isim bilgileri ile birlikte commit işleminin ne zaman yapıldığı ve commit mesajını görmekteyiz.

```
git status
```

Komutunu çalıştırıp tekrar sonuca baktığımızda ;

```
On branch master  
nothing to commit, working tree clean
```

Bilgisini görmekteyiz.Master dalında olduğunu ve commit işlemi yapılacak dosya bulunmadığını göstermektedir.

Şimdi sıra sizde ☐ buradaki işlemleri kendinizinde yapması önemli bu sebep ile küçük bi alıştırma yapmanız için developers.txt dosyasına yeni bir geliştirici adı ekleyip yukarı da anlattığım işlemleri tekrar edin.En son “git log” komutunu çalıştırdığınızda 2. bir commitin oluştuğunuz göreceksiniz.

Sonraki yazımda yukarıdaki alıştırmayı yapıp versiyon deęiştirme işlemini göreceğiz ...

Umarım faydalı olmuştur ☐

**Serkan Kaya**  
**Full Stack Java Developer**