Bu yazımızda .gitignore dosyası ile bazı dosyalarımızı git versiyon kontrol sisteminden hariç tutma işlemini göreceğiz …

Projemizde bazı dosyalarımızı versiyon kontrol sisteminden hariç tutmak isteyebiliriz.Örneğin veritabanı dosyamızı hariç tutmak isteyebiliriz … örneklemek amacı ile istediğiniz dizinde klasör yaratalım ve git depomuz için ayarlamalarını yapalım ;

```
mkdir git-ignore
cd git-ignore
git init
```

Sırası ile yukarıda ki kodları girdiğimizde “git-ignore” dizinizi oluşturup içine gittik ve bu klasör için git ayarlamasını yaptık.Şimdi ise uzak depomuzun ayarlamasını yapalım ;

```
git remote add origin https://github.com/serkankaya/create-repository.git
```

Yukarıda ki kod ile uzak depomuzu da ayarlamış olduk.Şimdi git ignore dosyamızı oluşturalım ;

```
touch .gitignore
```

gitignore dosyamızı oluşturduk ancak dizinde görememeniz çok normal… çünkü “.” ile başlayan gitignore dosyamız gizli dosya … görmek istersek aşağıdaki kodu yazın ;

```
ls -a
```

```
.  ..  .git  .gitignore
```

Görmüş olduğunuz gibi dosyamız oluşturuldu.Şimdi db bilgilerimizin olduğunuz varsaydığımız "db.txt" dosyamızı oluşturup git status ile durumuna bakalım ;

```
nano db.txt
git status
```

Sonuç :

```
On branch master
No commits yet
Untracked files:
  (use "git add <file>..." to include in what will be committed)
        .gitignore
        db.txt

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

Şu anda .gitignore ve db.txt dosyamızın index bölümüne atılmaya hazır olduğunu görüyoruz ancak bizim istediğimiz "db.txt" dosyasını git versiyon kontol sisteminin görmemesini istiyoruz bunun için ".gitignore" dosyamıza db.txt dosyamızı tanımlamamız gerekmekte … bunun için nano ile dosyamızı açıp içerisine "db.txt" yazıp kaydedin. daha sonra "git status" komutunu çalıştırdığınızda aşağıdaki çıktıyı alacaksınız …

```
On branch master
No commits yet
Untracked files:
  (use "git add <file>..." to include in what will be committed)
        .gitignore
nothing added to commit but untracked files present (use "git add" to
track)
```

Yukarıda ki çıktı da görmüş olacağınız üzere “db.txt” dosyamızı git’in görmezden gelmesini sağladık.

Önemli Not : git in görmezden gelmesini istediğimiz dosyaları add ve commit işleminden önce “.gitignore” dosyasına yazmanız şiddet ile tavsiye olunur. Daha sonra dosyalarınız görmezden gelinmeden yok sayılmadan versiyonlanabilir …

Buraya kadar tek bir dosyayı yoksaymayı gördük şimdi farklı bir örnek için dosyalarımızı versiyonlayalım ;

```
git add .
git commit -m "first commit with .gitignore"
```

commit işlemimizi de yaptığımızda göre , git in bir klasörü nasıl yoksayabililidiğini örnekleyelim.Bunun için “ghost_folder” adında bir klasör oluşturup içerisinde 3 adet dosya oluşturalım. daha sonra .gitignore dosyamızı düzenleme yapalım ;

```
mkdir ghost_folder
touch dosya{1..3}
cd ..
nano .gitignore
```

.gitignore dosyamızı düzenlerken önceki düzenlememize bir boşluk bırakarak “ghost_folder/*” yazıp kaydeder isek ;

Bu klasör tamamen git de yoksayılacaktır.git status ile durumu kontrol edebilirsiniz …

Klasörü de yoksaydırdık güzel peki bu klasör içerisinde bir dosyayı hariç tutmak istersek ne yapacağız ? bunun da çözümü basit … hatırladığınız üzere 3 adet dosya oluşturmuştuk varsayalım ki “dosya1” in versiyonlanmasını klasörde ki diğer dosyaların versiyonlanmamasını istiyoruz.Bunun için yine .gitignore dosyamızı açıp görmesini istediğimiz dosya1 için aşağıdaki satırı ekleyin ;

```
!ghost_folder/dosya1
```

Burada ki ünlem işareti bu dosyayı yoksaymamazlık yapma ! demek …dosyamızı kaydedip kapatın ve git status komutu ile inceleyin …

```
git status ghost_folder/
On branch master
Untracked files:
  (use "git add <file>..." to include in what will be committed)
        ghost_folder/dosya1
nothing added to commit but untracked files present (use "git add" to
track)
```

Gördüğünüz üzere dosya1 in index kısmına atılmadığını söylemekte …

Şimdi dosyalarımızı index e atıp commitleyip uzak sunucumuza atalım

```
git add .
git commit -m "gitignore ile örnek yapıldı"
git push -u origin master
```

yukarıda ki komutlardan sonra uzak sunucu kullanıcı adı ve şifrenizi girdikten sonra github repository nizi kontol ettiğinizde sadece belirttiğimiz dosyaların uzak sunucuya gittiğini görmüş olacaksınız …

gitignore ile yapılan işlemlerimizde bu kadar

Umarım faydalı olmuştur

Serkan Kaya
Full Stack Java Developer