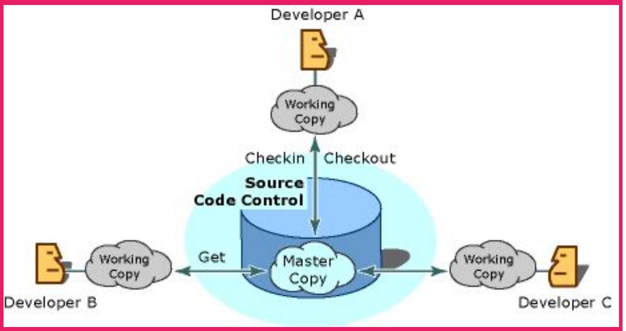
Introduction To Git

Before jump into ASP.NET MVC, I want to take a step back and look at source control.

If you are new to the term source control, source control is essentially using an available product for tracking changes of your code over time.

Even what your local, private, personel projects allows you to track changes of your self code over the time,   allows you to experiment with things,either merge this experiments into your main code, or just to discard them entirely.

It’s a best practice to track your source code.



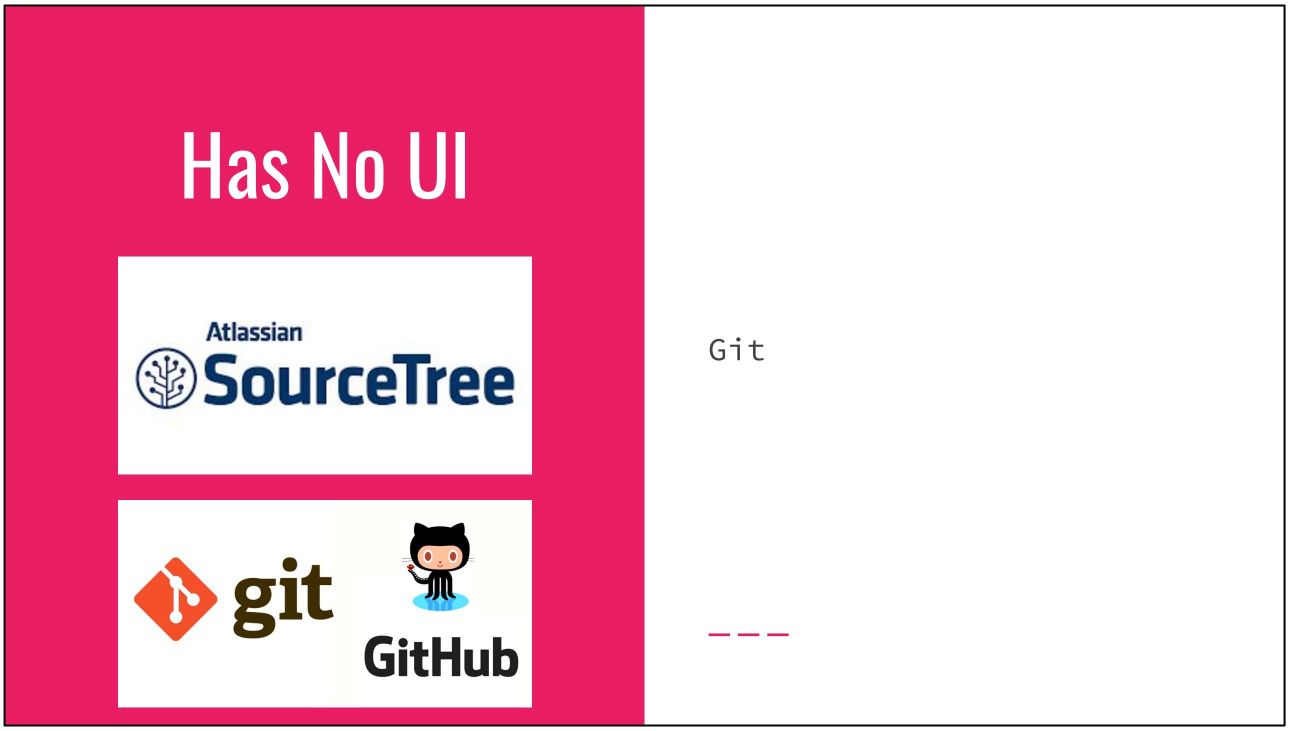
github.com
svn

Source Control

Source control does a lot of things depending on what productive use. And wider products are available. You must commly probably see svn, and more recentyl github.com.

Github is the one we will be using. We not go on every single feature of git.
Git is an amazing product, that was created specificly to handle the largest open source product in the world. Linux

So what git allows to do is, git allows to create repositories and track changes over time and see these changes whenever we want.



Has No UI

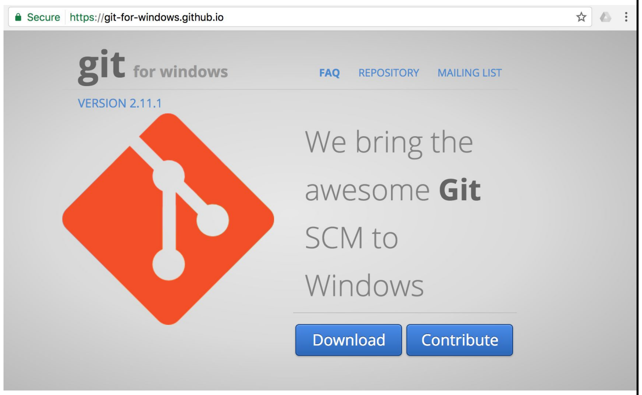
Git

Git does not come with an UI. Git is a command line tool. In fact, it is a tool of Linux but recently become stable on windows.

We will be using a combination of command line tools as well as free git user interface called source tree.

Source tree is made by Atlasian. It allows you to well use git and mercurial. Mercurial is a smilar distributed source control system as git

Download Git For Windows

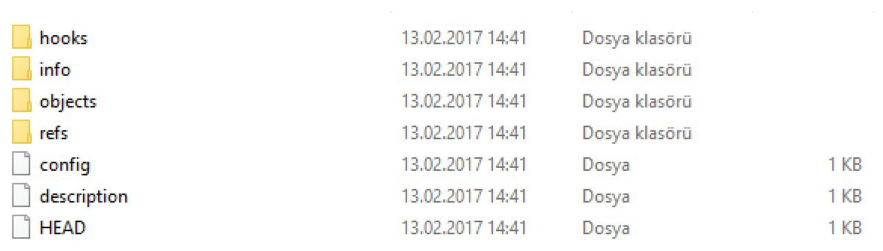

Secure https://git-for-windows.github.io
git for windows
FAQ
REPOSITORY
MAILING LIST
VERSION 2.11.1
We bring the awesome Git SCM to Windows
Download
Contribute

Download git for windows. Then you will have git installed as a comand line on system.

After that downaload source tree.

After that, open an folder and i am gonna create a git reporsitory. and i am gonna open the source tree and look at the basics of commiting files and reviewing our changes. And rewarding back to old changes.

- Create a folder
- Open command line tool
- Change path to new created folder
- Type in git and see that git is avaliable and see a list of parameters



hooks	13.02.2017 14:41	Dosya klasörü	
info	13.02.2017 14:41	Dosya klasörü	
objects	13.02.2017 14:41	Dosya klasörü	
refs	13.02.2017 14:41	Dosya klasörü	
config	13.02.2017 14:41	Dosya	1 KB
description	13.02.2017 14:41	Dosya	1 KB
HEAD	13.02.2017 14:41	Dosya	1 KB

Creating a github reporsitory

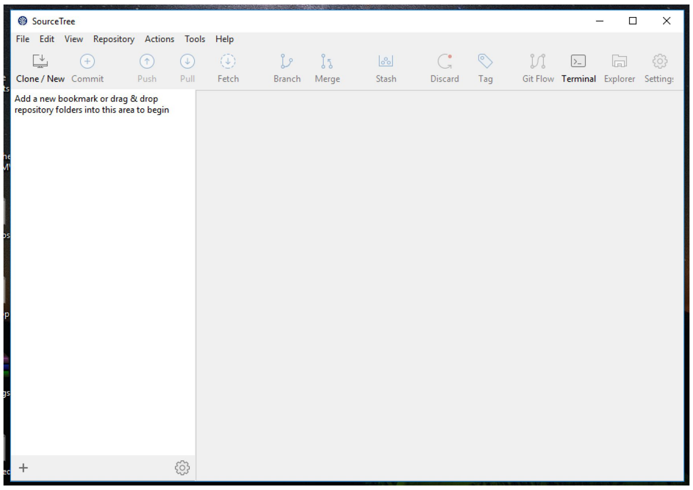
First create a folder called SimpleBlog.

And open a command line window.

Change path to new created folder. Let's issue the init command. Said initialized empty Git reporsitory. TInteresting thing is that now we have a hidden ".git" folder. It is important to understand that all the files in the SimpleBlog folder is a part of reporsitory. Reporsitory is the individual project we just created with git so you have reporsitory for project. And we look at this filesystem in this folder the reporsitory files will be here and also o folder .git. What will be stored in this folder. This is the internal git stuff. hat stores our objects and refs and hooks but we do not get into.

You can thing like this. Most of the files in reporsitory is stored in SimpleBlog folder. This .git folder will contain the history of single file since git reporsitory was created. So if you delete the .git folder well you just lost of your history. So do not delete the .git folder.

SourceTree

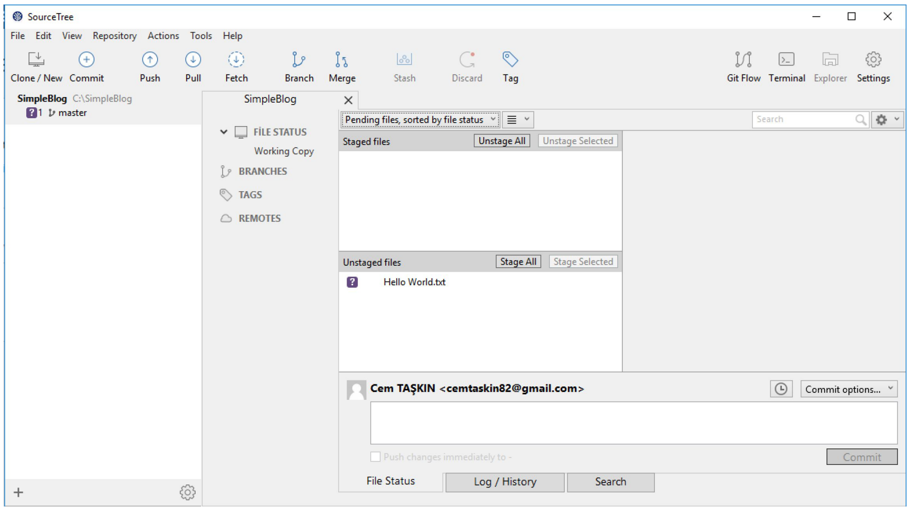

SourceTree
File Edit View Repository Actions Tools Help
Clone / New Commit Push Pull Fetch Branch Merge Stash Discard Tag Git Flow Terminal Explorer Settings

Add a new bookmark or drag & drop repository folders into this area to begin

Open source tree. Open the git reporsity folder and simple drag to folder into sourceTree.

We just create a reporsitory with command line tool but we can just use the graphical interface. Everything inside the gui can be done with command line tool gui.exe by any particular setup argument.

Let’s Look At The Details Of Our Reporsitory

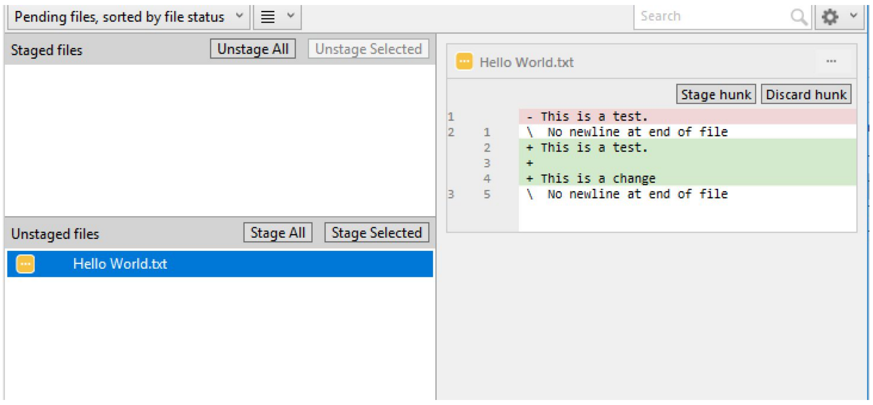

SourceTree
File Edit View Repository Actions Tools Help
Clone / New Commit Push Pull Fetch Branch Merge Stash Discard Tag Git Flow Terminal Explorer Settings
SimpleBlog C:\SimpleBlog
master
FILE STATUS
Working Copy
BRANCHES
TAGS
REMOTES
Pending files, sorted by file status
Staged files
Unstage All
Unstage Selected
Unstaged files
Stage All
Stage Selected
Hello World.txt
Cem TAŞKIN <cemtaskin82@gmail.com>
Commit options...
Push changes immediately to -
Commit
File Status
Log / History
Search

We have our SimpleBlog reporsity on this tab. We have pending changes and working copy changes. Changes are right here.
Let’s make an example. I go to my SimpleBlog folder. We create a new text file named Hello World. And write some text in it, save and close file. Something was changed. We see on Unstaged files Hello World.txt. When click on this file,we can see the changes in this file. The important thing to understand is that, this Hello World.txt file is not stored on git yet. It is considered to be a change. We changed the our empty repository by adding something in it.

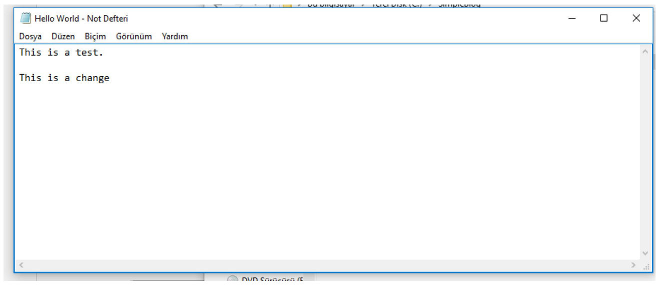
We have to go SourceTree and tell it to track this change. The reason that you have to do is not automatic. Let’s say I add a bunch of files. (Hey.txt, Who.txt). I want to logically group these changes. The change of adding tree files. We want to logically group them into a single unit. The single unit is called as “commit”. A commit is essantialy a list of changes. You added to file, modity the file or move the file.

Now, back to SourceTree and it is telling us we have made changes. You may be do not want to track every changes uniquely and may be group these changes.

Let’s commit changes. We must stage files. SourceTree told me what is changed on filesystem. And I told you I want these 3 changes which is all of them will represent my next commit. Let’s commit changes. There is a big button on toolbar. Commit. When I click on this button, we mus write a comment. It is very important to add this commit messages. It will describe what was done in this commit. You do not say I added this file, this file, ….. You say about what you have done. You may describe why this changes are made.



```
- This is a test.
\  No newline at end of file
+ This is a test.
+
+ This is a change
\  No newline at end of file
```



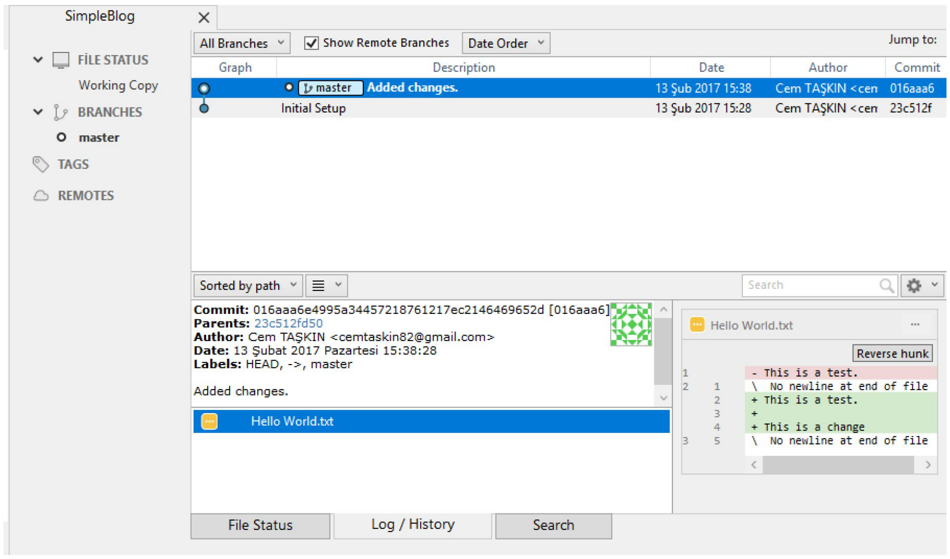
```
This is a test.

This is a change
```

Let’s make a change in Hello World File !

When we made a change in tracked files, SourceTree will show changes in details. It tells us how much modification was done. So we will stage this file. And commit. And in the comment area, i will write “Added Changes”.

In SourceTree, we have a bunch of fetures that we can use.



Graph	Description	Date	Author	Commit
	master Added changes.	13 Şub 2017 15:38	Cem TAŞKIN <cem	016aaa6
	Initial Setup	13 Şub 2017 15:28	Cem TAŞKIN <cem	23c512f

Commit: 016aaa6e4995a34457218761217ec2146469652d [016aaa6]
Parents: 23c512fd50
Author: Cem TAŞKIN <cemtaskin82@gmail.com>
Date: 13 Şubat 2017 Pazartesi 15:38:28
Labels: HEAD, ->, master

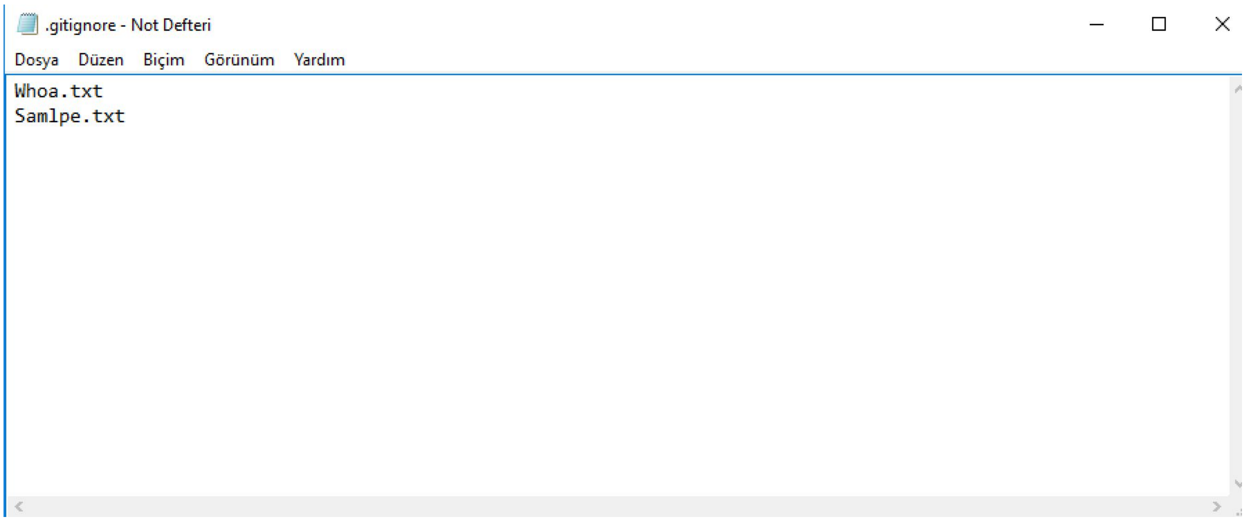
Added changes.

```
1       - This is a test.
2   1   \  No newline at end of file
    2   + This is a test.
    3   +
    4   + This is a change
3   5   \  No newline at end of file
```

Looking at Log

Instead of other features , lets have a look at Log. And I can see we have two commits. And I can also track the change details.

It is possible to go back and CheckOut Changes. Right click on the Added Changes commit and CheckOut. And gone to a back on time.



```
Whoa.txt
Samlpe.txt
```

gitignore.file

There is one more thing to learn. git.ignore file.

gitignore file tells the git how to ignore the things. Sometimes we do not want some fikes not to commit.

Create a file named gitignore. The file names in this file are ignored.

We don’t need binaries, compiled files and debug simples and other fikes in our reporsitory. So we add this files in our git ignore file.