HopeLine Application Development

This document is prepared by <u>Edmel Ricahuerta</u>

The purpose of this document is to help the development team on starting and developing the application for the semester.

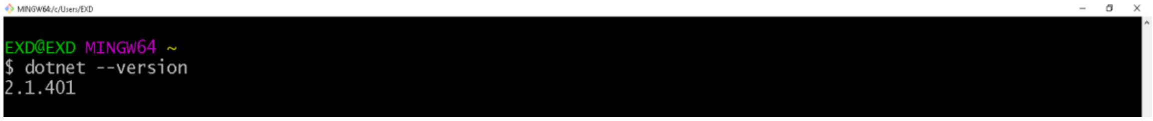
Audience: Development Team (Group7)

Follow these steps to build the application with ease:

1. PREPARE YOUR WORKINGF ENVIRONMENT

✓ DOTNET

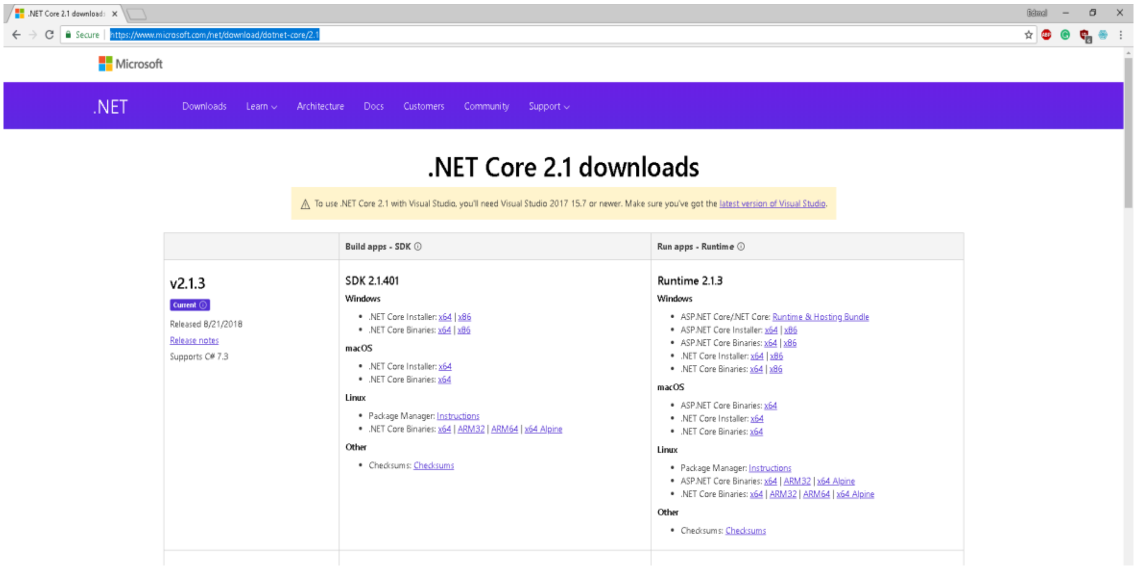
Make sure to install NET Core 2.1 SDK on your machine before doing anything else. To check, run the following code:



```
EXD@EXD MINGW64 ~
$ dotnet --version
2.1.401
```

If it does not exist or it is not recognized, download and install from here:

https://www.microsoft.com/net/download/dotnet-core/2.1



.NET Core 2.1 downloads

⚠ To use .NET Core 2.1 with Visual Studio, you'll need Visual Studio 2017 15.7 or newer. Make sure you've got the latest version of Visual Studio.

	Build apps - SDK ⓘ	Run apps - Runtime ⓘ
v2.1.3 Current ⓘ Released 8/21/2018 Release notes Supports C# 7.3	SDK 2.1.401 **Windows** • .NET Core Installer: x64 | x86 • .NET Core Binaries: x64 | x86 **macOS** • .NET Core Installer: x64 • .NET Core Binaries: x64 **Linux** • Package Manager: Instructions • .NET Core Binaries: x64 | ARM32 | ARM64 | x64 Alpine **Other** • Checksums: Checksums	Runtime 2.1.3 **Windows** • ASP.NET Core/.NET Core: Runtime & Hosting Bundle • ASP.NET Core Installer: x64 | x86 • ASP.NET Core Binaries: x64 | x86 • .NET Core Installer: x64 | x86 • .NET Core Binaries: x64 | x86 **macOS** • ASP.NET Core Binaries: x64 • .NET Core Installer: x64 • .NET Core Binaries: x64 **Linux** • Package Manager: Instructions • ASP.NET Core Binaries: x64 | ARM32 | x64 Alpine • .NET Core Binaries: x64 | ARM32 | ARM64 | x64 Alpine **Other** • Checksums: Checksums

Download the latest SDK as of September 2018 which is 2.1.401

- ✓ GIT

Make sure that you also have git install. To download the latest see https://git-scm.com/downloads.

- ✓ IDE

Highly recommend getting Visual Studio 2017 or Visual Studio Code as your editor. It has powerful extension to help you code faster and be more productive. These extensions are the following you should install to your IDE:

Productivity Power Tools 2017 – VS 2017

Web Essentials 2017 – VS Code

Beautify – VS Code

Bootstrap 4 Snippet – VS Code

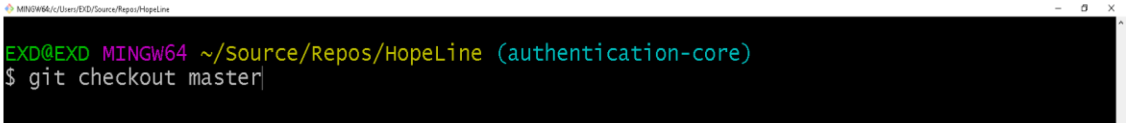
2. GITHUB/ GIT

Master – this branch is the production branch. Do not make direct changes here unless reviewed or accepted by the team.
Branch – all branches are made for separations of task related .

When working on the assigned task, consider the following:

- ✓ The first thing you do is to have a clone of the repository https://github.com/ejricahuerta/HopeLine.git
- ✓ Open IDE and switch from **master** to your assigned **branch –** in our initial branches

  master -> webui
  master -> authentication-core



```
EXD@EXD MINGW64 ~/Source/Repos/HopeLine (authentication-core)
$ git checkout master
```

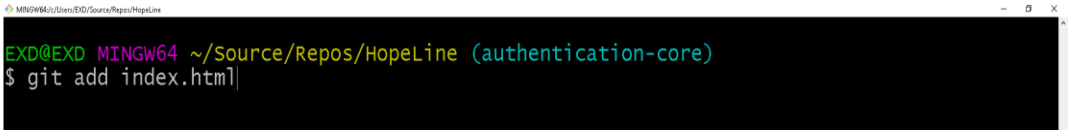
  this example shows how to switch branches from **authentication-core** to **master**
- ✓ Before changing any codes, best practice to perform **git pull** or **sync**
- ✓ When commit changes, **stage** the file you want to commit and perform **git commit**

PS: I highly recommend performing gits with cli / bash.

VS 2017 - https://docs.microsoft.com/en-us/vsts/repos/git/gitquickstart?view=vsts&tabs=visual-studio

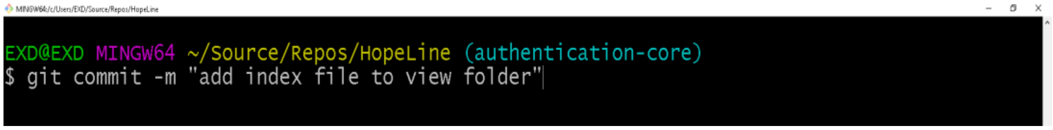
VS Code - https://code.visualstudio.com/docs/editor/versioncontrol

BASH



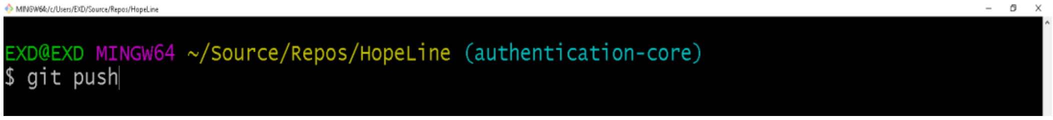
```
EXD@EXD MINGW64 ~/Source/Repos/HopeLine (authentication-core)
$ git add index.html
```

adding the file with changes



```
EXD@EXD MINGW64 ~/Source/Repos/HopeLine (authentication-core)
$ git commit -m "add index file to view folder"
```

committing the changes for the branch



```
EXD@EXD MINGW64 ~/Source/Repos/HopeLine (authentication-core)
$ git push
```

after performing all commits, perform **git push.**

MAKE SURE to pull again before start coding.

✓ When the branch is ready for review, go ahead and create a pull request.

3. INLINE DOCUMENTATION

It is important to include documentation for the code you wrote.

Summary – ensure to have a summary of the class or function on what is it all about and how to use it.

```
/// <summary>
/// Generic Repository for all derived BaseEntity classes
/// </summary>
/// <typeparam name="T"></typeparam>
public class Repository<T>: IRepository<T> where T: BaseEntity
```

TODO – include to-dos for if unfinished or needs to be reviewed

```
public void Delete(T obj)
{
    //TODO: add delete logic with error handling - Eduardo
    throw new System.NotImplementedException();
}
```

TODO: some_coments – person_needs_to_check_or_update

For **html pages**, click here.