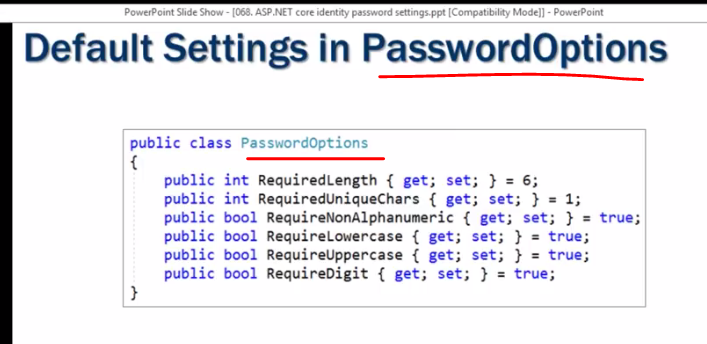
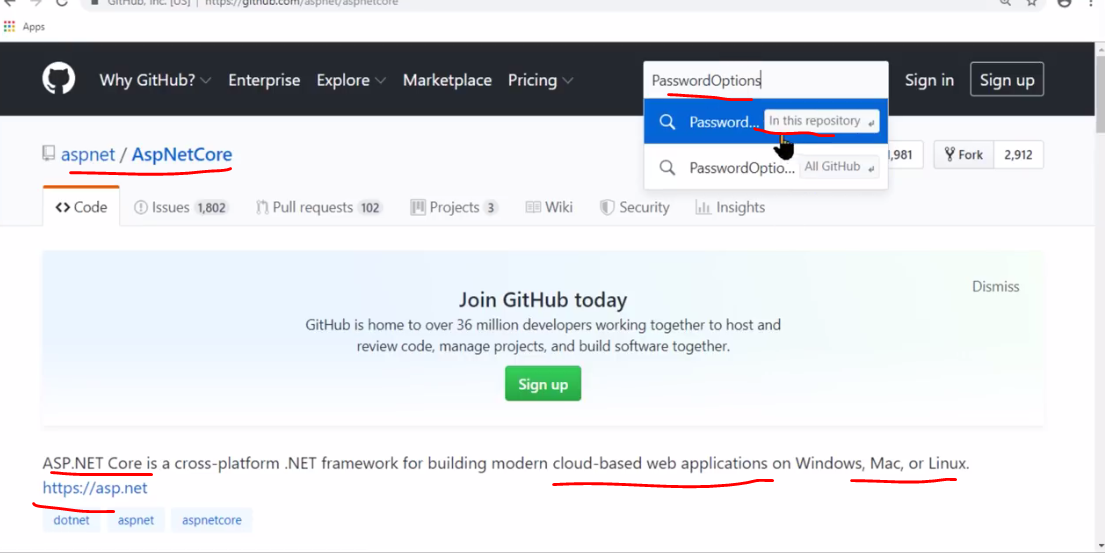
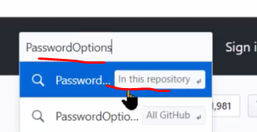


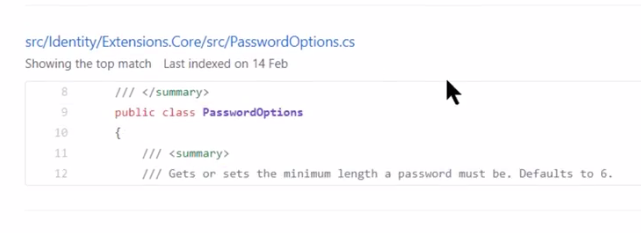
Open source class on github

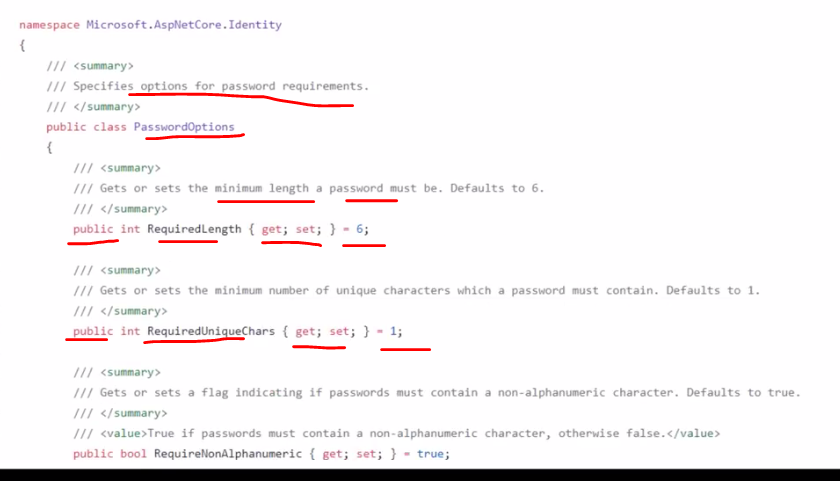


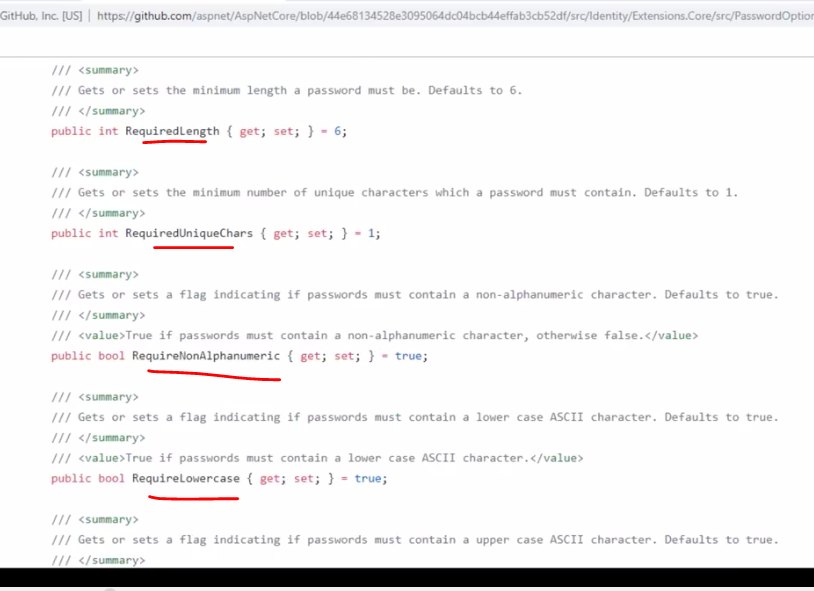




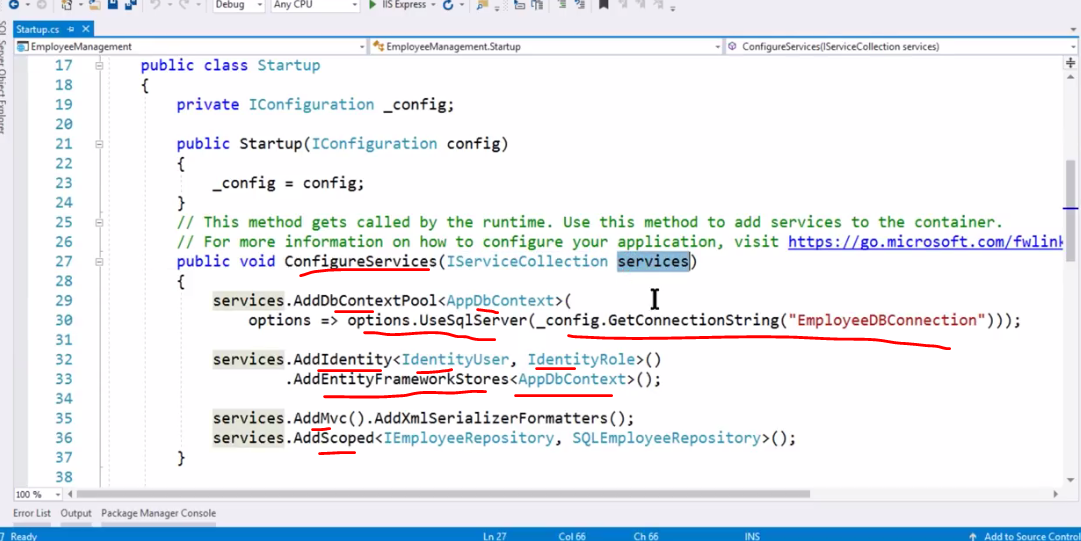








Configure those options in the Startup class



Above is adding Identity with default settings

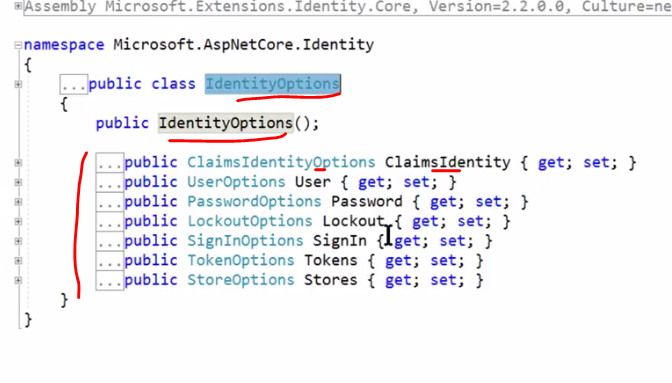
Following Identity Options can be configured, one of them is Password Option

Following is a root level class for Identity Configuration.

Previously we do this by Web.xml class by writing xml. Now in code.

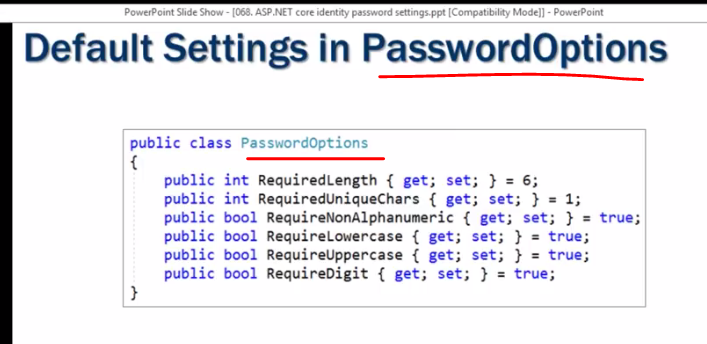
There is a complete hierarchy of Classes we follow to Configure Identity.

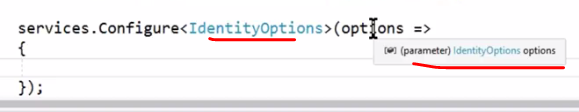
Following 7 high level aspects of Identity can be configured.



Password aspect of Identity can be configured as above

Following 6 aspects of Passwords can be configured.





Following are two ways to configure same Password Options

IdenityOptins is the root xml element to configure Identity framework

IdentityOptions is the root class that contain the all the Variability of IF

IdentityOptions class is the Configuration Class taken as input to configure IF

IdentityOptions class is the Variability Class taken as input to configure IF

IdentityOptions class is the Configuration Class that contain Configuration Markup for the IF

When we add Middleware, we provide that Configuration Options class.

Default Middleware ADD method that takes nothing and Configure IF with Default options

service.ADDXXXX() service.ADDXXXX(Options Configuration Object)

Middleware ADD method that takes Configuration Option Object as input for IF configuration. It does both ADD and CONFIGURATION work.

services.CONFIGURE<XXXXOptionsClass>()

We have separate CONFIGURE method that takes Configuration Option Object as input for IF configuration. It does only CONFIGURATION work.

