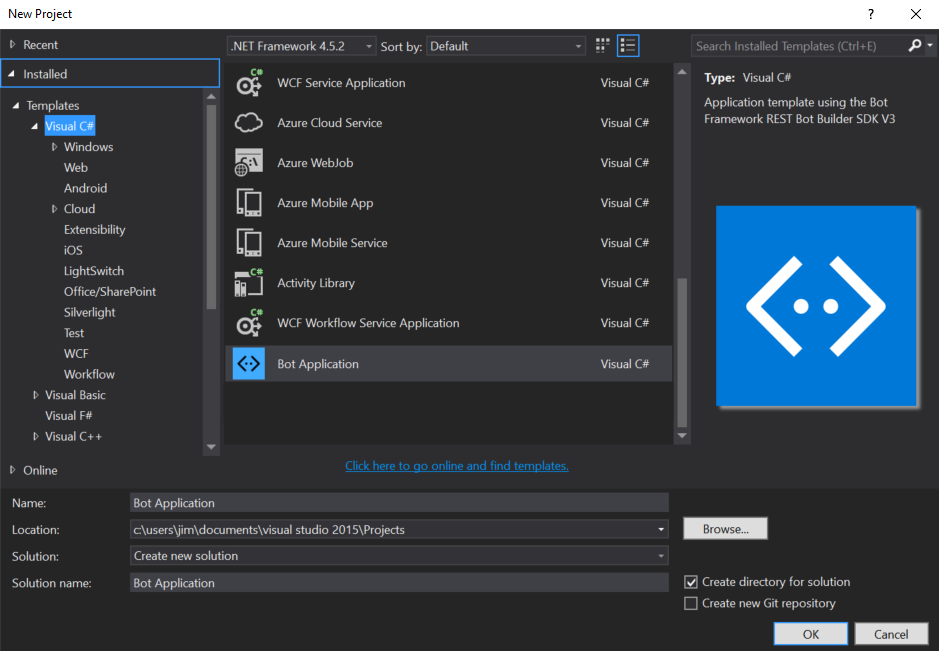
# Requirements:

* Install Visual Studio 2017
  + Update all extensions  
    <https://docs.microsoft.com/en-us/visualstudio/extensibility/how-to-update-a-visual-studio-extension>
* Install Bot Builder in NuGet Package Console (if not using Bot Framework template files), type as below:
  + Install-Package Microsoft.Bot.Builder
* Download Bot Framework template zip files:
  + Download Bot template for Visual Studio  
    <http://aka.ms/bf-bc-vstemplate>
  + Download Bot Controller  
    <http://aka.ms/bf-bc-vscontrollertemplate>
  + Download Bot Dialog  
    <http://aka.ms/bf-bc-vsdialogtemplate>
  + Copy & paste template files to the directory  
    %USERPROFILE%\Documents\Visual Studio 2017\Templates\ProjectTemplates\Visual C#\
* Download Bot emulator and install  
  <https://github.com/Microsoft/BotFramework-Emulator/releases/tag/v3.5.31>

# Create simple bot:

* Open Visual Studio and create a new C# project. Choose the Bot Application template for your new project



* Run application
* Type into the address bar of emulator, where port-number matches the port number shown in the browser where your application is running.  
  <http://localhost:port-number/api/messages>

# Development steps guide

* Design a bot:  
  <https://docs.microsoft.com/en-us/bot-framework/bot-design-principles>
  + Design principles:
    - **Factors that do not guarantee a bot’s success**
      * How “smart” the bot is
      * How much natural language the bot supports
      * Voice
    - **Factors that do influence a bot’s success**
      * Does the bot easily solve the user’s problem with the minimum number of steps? (*does not require to type too much, talk too much, repeat several times, or explain things*)
      * Does the bot solve the user’s problem better/easier/faster than any of the alternative experiences?
      * Does the bot run on the devices and platforms the user cares about?
      * Is the bot discoverable? Do the users naturally know what to do when using it?
  + Conversation flow:
    - A bot will communicate in dialog stacks. When one dialog invokes another, the Bot Builder adds new dialog to the top of the dialog stack.
    - Although it would be great if users always traveled such a linear, logical path, it seldom occurs. Humans do not communicate in "stacks".
      * Insist that the user answer the question first.
      * Disregard everything that the user had done previously, reset the whole dialog stack, and start from the beginning by attempting to answer the user's question.
      * Attempt to answer the user's question and then return to that yes/no question and try to resume from there.
* Develop bot with .NET:
* Test & Debug:  
  <https://docs.microsoft.com/en-us/bot-framework/debug-bots-emulator>
* Deploy bot:  
  <https://docs.microsoft.com/en-us/bot-framework/deploy-bot-overview>
* Manage bot:
  + Register bot:  
    <https://docs.microsoft.com/en-us/bot-framework/portal-register-bot>
  + Enable analytics:  
    <https://docs.microsoft.com/en-us/bot-framework/portal-analytics-overview>