Workshop on Chatbots

Microsoft Hong Kong
Hands-On Workshop by Waqas Ali (Microsoft Student Partner)
Contact: waqas.abbasi@outlook.com

1. Download and Install Node.js (LTS Version)
2. Use any IDE you prefer. I will be using Jetbrains Webstorm It's free for students (You can even use notepad but an IDE like WebStorm makes running/debugging quite easy) Visual Studio Code is an equally great alternative
3. Download and Install Bot Emulator
4. Create your project
    a. If using Webstorm:
        i. Create new empty project
        ii. Find terminal in bottom-left, type following commands:
            1. npm init (to initialize the folder as a Node.js project)
            2. npm install --save botbuilder (to install the required package made by Microsoft)
                a. the save flag also automatically adds botbuilder package to package.json which is quite important, otherwise wherever you publish, the system wouldn't know what packages your project uses and hence won't run
            3. npm install --save restify (package to setup bot server with which messenger/skype,etc can talk to)
            4. npm install --save request (package to easily send http requests, required specifically for this project, no need to add it to to projects where you are not sending http requests)
        iii. Add new javascript file named "app.js"
        iv. Click project, edit configurations, "plus" sign, select Node.js, rename to project name, select app.js in "Javascript file:" (this tells which file to run when the green play button is pressed)
    b. If using some other IDE:
        i. Create a folder for project
        ii. Open Node.js in command prompt
        iii. Navigate to folder
        iv. Run above commands
        v. Add new JavaScript file named "app.js"
5. Use the code from Github for now and paste into your app.js
6. Run the app
    a. If using WebStorm, press the green play button
    b. If using some other IDE, navigate to folder in command prompt or terminal, and type node app.js
7. Connect to bot from bot emulator
    a. Type address as http://localhost:3978/api/messages
    b. Connect
    c. Try different messages
8. Congratulations for getting this far, let's publish the bot now

9. In the terminal in WebStorm or outside run following commands:
    a. npm install azure-cli -g (to install azure package to publish to azure)
    b. azure config mode asm (to change mode)

    c. azure login (you will be given a link to go to and sign in with your azure account)
    d. azure site create --git frankgeobot (change the name to your project name, choose any region)

    e. git add . (these three commands you must always run after making changes)
    f. git commit -m "Basic Implementation" (change the message to keep a nice record of changes you are making)
    g. git push azure master (these are universal git commands, I highly recommend you to learn git if you haven't learned it before)
10. Now your bot will be published on (projectname).azurewebsites.net and services can communicate with it on https://(projectname).azurewebsites.net/api/messages
11. To publish the bot, go to https://dev.botframework.com/
12. Make an account and Register a bot
13. Fill in the basic information and write the endpoint in the following format: https://(projectname).azurewebsites.net/api/messages
14. Click "Create Microsoft App id and password", login and get the details
15. Save the password somewhere safe
16. Go to azure portal -> bot web app -> application settings and add the **app id and passwords** *(you can control and observe many other aspects of your app from Azure Portal including upgrading the web app from free to higher packages which of course would be better performing)*
17. You are almost done, go back to dev.botframework.com and you can now communicate with your bot through the built-in web chat
18. You can now publish your bot on any of the numerous supported platforms
19. If you click on messenger, you can see a detailed how-to guide on how to do that. Give it a try. If you face some difficulty, you can always contact me at waqas.abbasi@outlook.com
20. Look at the final product here
21. **The code on Github has been commented to explain what's happening**

Links:

1. Code, Review and Screenshots of the Workshop
2. Documentation
3. Core Concepts
4. Getting Started
5. Natural Language Processing (LUIS)
6. Luis.ai
7. Luis.ai Help
8. Deploy to Azure
9. Tutorial Videos
10. BotBuilder Examples
11. BotBuilder Samples
12. BotBuilder Facebook Sample
13. Ella (A project I am currently working on)
14. Curo (A project I am currently working on)