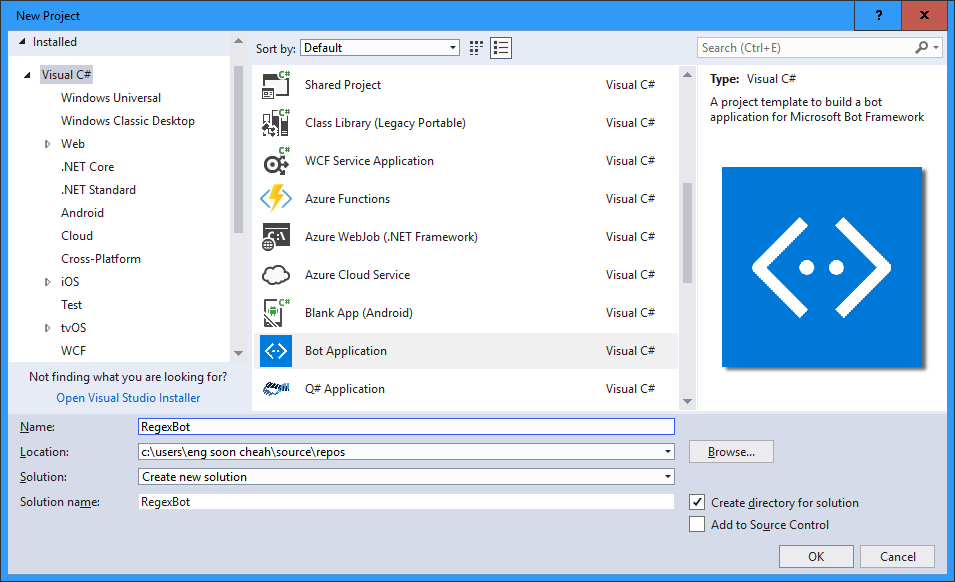
**Cloud AI Bootcamp Day 2 : Part 2 – Regex and Scorable Group**

We will be developing a bot using the C# SDK. To get started, you need two things:

1. The Bot Framework project template, which you can download [**here**](http://aka.ms/bf-bc-vstemplate). The file is called “Bot Application.zip” and you should save it into the \Documents\Visual Studio 2017\Templates\ProjectTemplates\Visual C#\ directory. Just drop the whole zipped file in there; no need to unzip.
2. Download the Bot Framework Emulator for testing your bot locally [here](https://github.com/Microsoft/BotFramework-Emulator/releases/download/v3.5.33/botframework-emulator-Setup-3.5.33.exe). The emulator installs to c:\Users\*your-username*\AppData\Local\botframework\app-3.5.33\botframework-emulator.exe or your Downloads folder, depending on browser.
3. Now Go to your Visual Studio create the bot application and name the solution as “RegexBot”.



1. Go to “RootDialog.cs” , add

**using** Microsoft.Bot.Builder.Scorables;

**using** System.Collections.Generic;

and change your class derivation to:

**public** **class** **RootDialog** : DispatchDialog<**object**>

1. Copy and paste the code inside the RootDialog.cs

[Serializable]

public class RootDialog : DispatchDialog<object>

{

[RegexPattern("^hello")]

[RegexPattern("^hi")]

[ScorableGroup(0)]

public async Task Hello(IDialogContext context, IActivity activity)

{

await context.PostAsync("Hello from RegEx! I am a Photo Organization Bot. I can search your photos, share your photos on Twitter, and order prints of your photos. You can ask me things like 'find pictures of food'.");

}

[RegexPattern("^help")]

[ScorableGroup(0)]

public async Task Help(IDialogContext context, IActivity activity)

{

// Launch help dialog with button menu

List<string> choices = new List<string>(new string[] { "Search Pictures", "Share Picture", "Order Prints" });

PromptDialog.Choice<string>(context, ResumeAfterChoice,

new PromptOptions<string>("How can I help you?", options: choices));

}

private async Task ResumeAfterChoice(IDialogContext context, IAwaitable<string> result)

{

string choice = await result;

switch (choice)

{

case "Search Pictures":

// PromptDialog.Text(context, ResumeAfterSearchTopicClarification,

// "What kind of picture do you want to search for?");

break;

case "Share Picture":

//await SharePic(context, null);

break;

case "Order Prints":

//await OrderPic(context, null);

break;

default:

await context.PostAsync("I'm sorry. I didn't understand you.");

break;

}

}

private async Task ResumeAfterSearchTopicClarification(IDialogContext context, IAwaitable<string> result)

{

string searchTerm = await result;

context.Call(new SearchDialog(searchTerm), ResumeAfterSearchDialog);

}

private async Task ResumeAfterSearchDialog(IDialogContext context, IAwaitable<object> result)

{

await context.PostAsync("Done searching pictures");

}

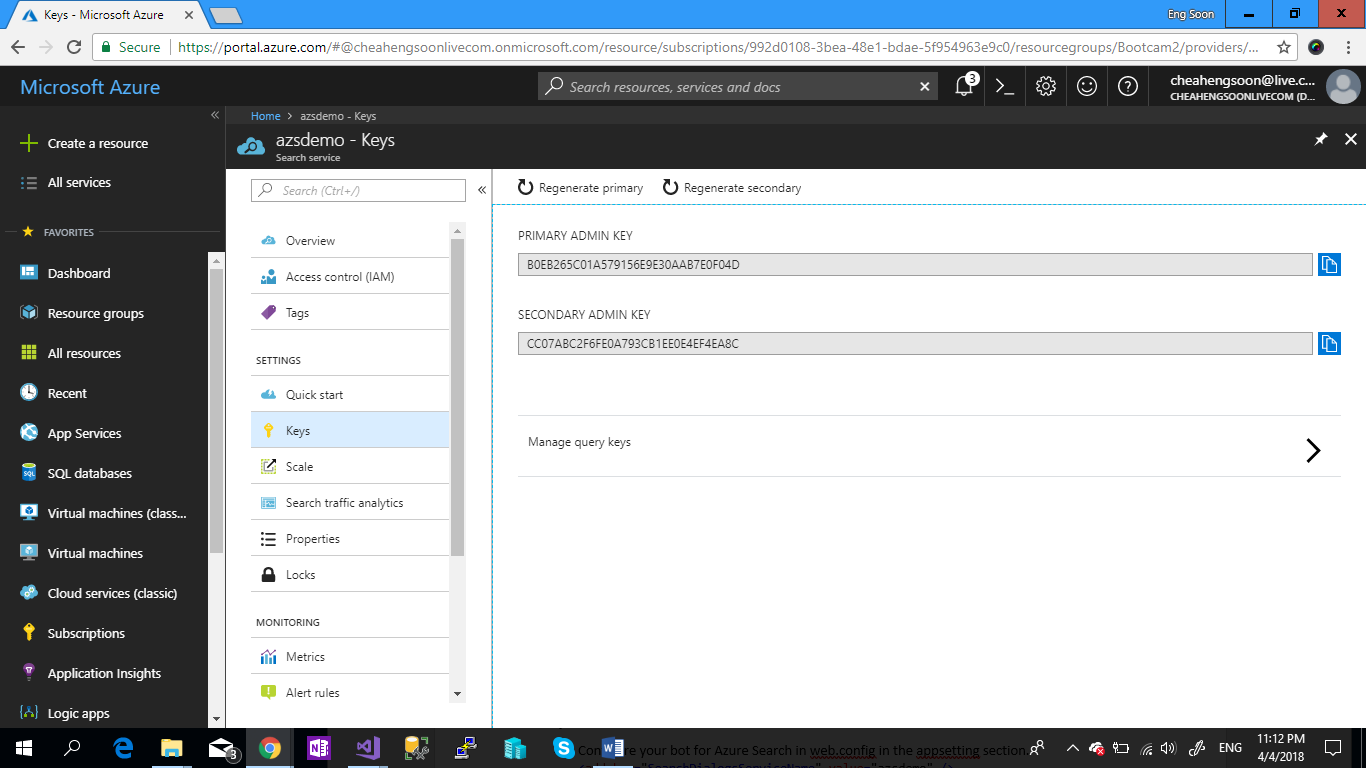
}

1. Configure your bot for Azure Search in web.config in the appsetting section.

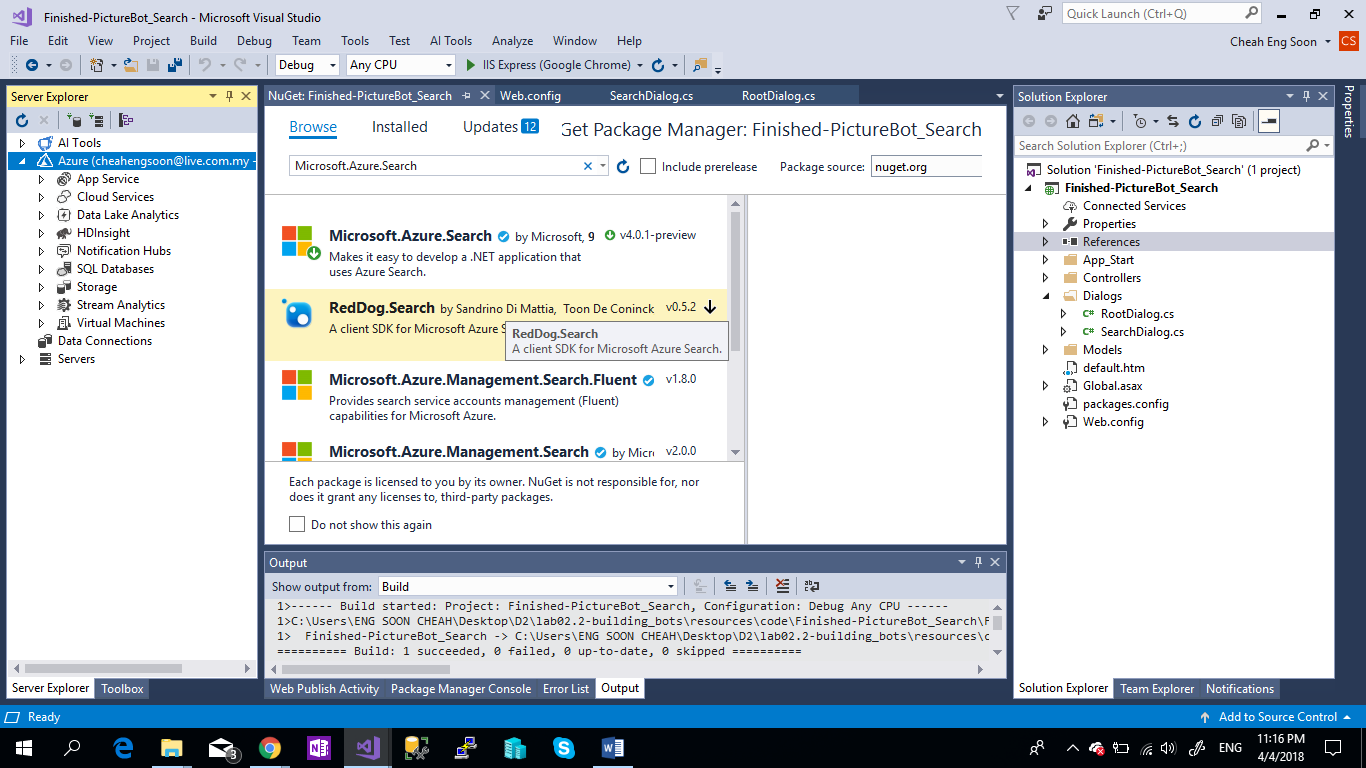
<add key="SearchDialogsServiceName" value="azsdemo" />

<add key="SearchDialogsServiceKey" value="B0EB265C01A579156E9E30AAB7E0F04D" />

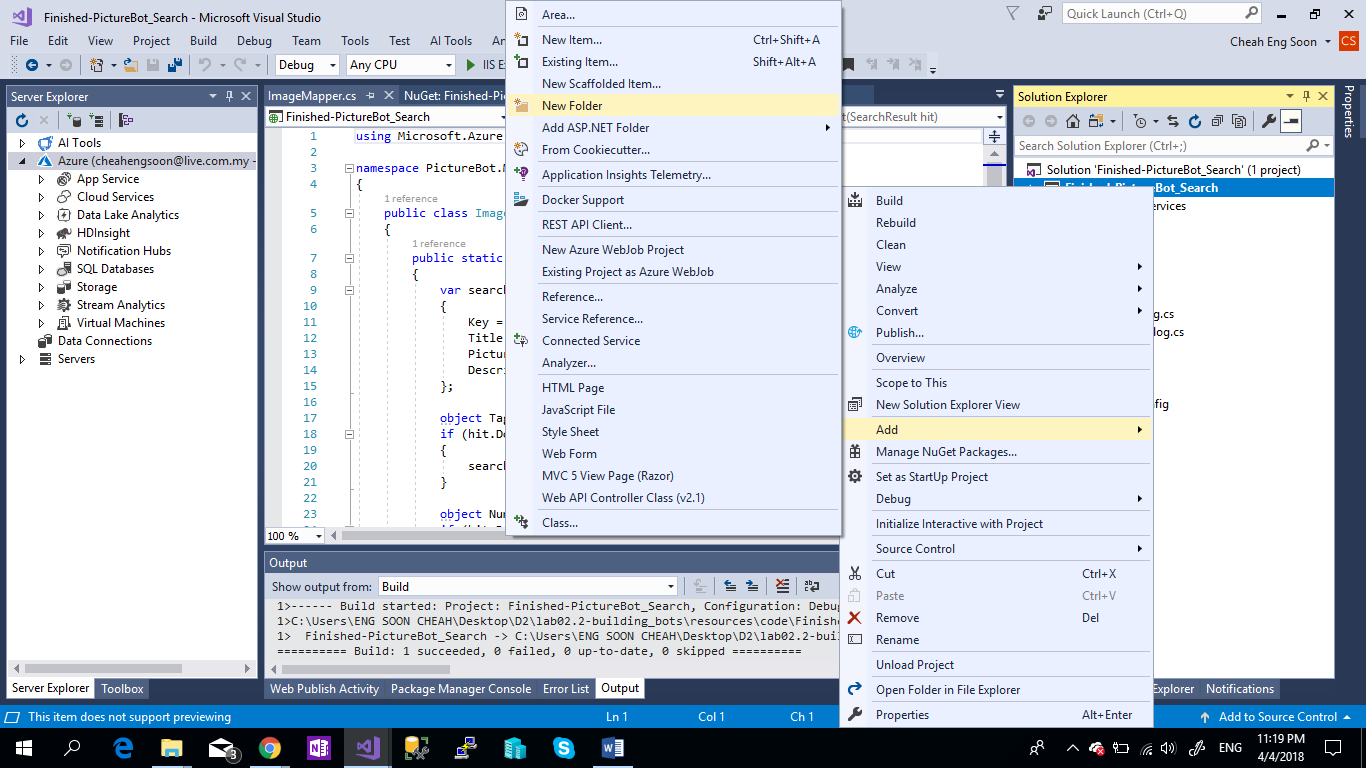
<add key="SearchDialogsIndexName" value="images" />



1. Right Click “Reference” in you Bot Application > Select “Manage Nuget Packages” > Search “Microsoft.Azure.Search” and Install.



1. Go to your Bot Solution , create a Folder name and “Models”.



1. Go to “Models” Folder, Right Click and add “Existing item”.

You can download this class.

[ImageMapper.cs](https://azure.github.io/LearnAI-Bootcamp/lab02.2-building_bots/resources/code/Models/ImageMapper.cs)

[SearchHit.cs](https://azure.github.io/LearnAI-Bootcamp/lab02.2-building_bots/resources/code/Models/SearchHit.cs)

1. Now do the same things , add “Existing item” for “Dialogs”.

SeachDialog.cs [here](https://azure.github.io/LearnAI-Bootcamp/lab02.2-building_bots/resources/code/SearchDialog.cs)

1. Go to “RootDialog.cs” , add the references

**using** Microsoft.Bot.Builder.Luis;

**using** Microsoft.Bot.Builder.Luis.Models;

1. Copy and paste the Following code to the class:

[LuisModel("f542d017-743f-418c-8a56-894aaebbae38", "d159515b5f9c4c70b805b842bb2c021b")]

[Serializable]

public class RootDialog : DispatchDialog<object>

{

[RegexPattern("^hello")]

[RegexPattern("^hi")]

[ScorableGroup(0)]

public async Task Hello(IDialogContext context, IActivity activity)

{

await context.PostAsync("Hello from RegEx! I am a Photo Organization Bot. I can search your photos, share your photos on Twitter, and order prints of your photos. You can ask me things like 'find pictures of food'.");

}

[RegexPattern("^help")]

[ScorableGroup(0)]

public async Task Help(IDialogContext context, IActivity activity)

{

// Launch help dialog with button menu

List<string> choices = new List<string>(new string[] { "Search Pictures", "Share Picture", "Order Prints" });

PromptDialog.Choice<string>(context, ResumeAfterChoice,

new PromptOptions<string>("How can I help you?", options: choices));

}

private async Task ResumeAfterChoice(IDialogContext context, IAwaitable<string> result)

{

string choice = await result;

switch (choice)

{

case "Search Pictures":

PromptDialog.Text(context, ResumeAfterSearchTopicClarification,

"What kind of picture do you want to search for?");

break;

case "Share Picture":

//await SharePic(context, null);

break;

case "Order Prints":

//await OrderPic(context, null);

break;

default:

await context.PostAsync("I'm sorry. I didn't understand you.");

break;

}

}

private async Task ResumeAfterSearchTopicClarification(IDialogContext context, IAwaitable<string> result)

{

string searchTerm = await result;

context.Call(new SearchDialog(searchTerm), ResumeAfterSearchDialog);

}

private async Task ResumeAfterSearchDialog(IDialogContext context, IAwaitable<object> result)

{

await context.PostAsync("Done searching pictures");

}

[LuisIntent("Greeting")]

[ScorableGroup(1)]

public async Task Greeting(IDialogContext context, LuisResult result)

{

// Duplicate logic, for a teachable moment on Scorables.

await context.PostAsync("Hello from LUIS! I am a Photo Organization Bot. I can search your photos, share your photos on Twitter, and order prints of your photos. You can ask me things like 'find pictures of food'.");

}

[LuisIntent("")]

[LuisIntent("None")]

[ScorableGroup(1)]

public async Task None(IDialogContext context, LuisResult result)

{

// Luis returned with "None" as the winning intent,

// so drop down to next level of ScorableGroups.

ContinueWithNextGroup();

}

[LuisIntent("SearchPics")]

[ScorableGroup(1)]

public async Task SearchPics(IDialogContext context, LuisResult result)

{

// Check if LUIS has identified the search term that we should look for.

string facet = null;

EntityRecommendation rec;

if (result.TryFindEntity("facet", out rec)) facet = rec.Entity;

// If we don't know what to search for (for example, the user said

// "find pictures" or "search" instead of "find pictures of x"),

// then prompt for a search term.

if (string.IsNullOrEmpty(facet))

{

PromptDialog.Text(context, ResumeAfterSearchTopicClarification,

"What kind of picture do you want to search for?");

}

else

{

await context.PostAsync("Searching pictures...");

context.Call(new SearchDialog(facet), ResumeAfterSearchDialog);

}

}

[LuisIntent("OrderPic")]

[ScorableGroup(1)]

public async Task OrderPic(IDialogContext context, LuisResult result)

{

await context.PostAsync("Ordering your pictures...");

}

[LuisIntent("SharePic")]

[ScorableGroup(1)]

public async Task SharePic(IDialogContext context, LuisResult result)

{

PromptDialog.Confirm(context, AfterShareAsync,

"Are you sure you want to tweet this picture?");

}

private async Task AfterShareAsync(IDialogContext context, IAwaitable<bool> result)

{

if (result.GetAwaiter().GetResult() == true)

{

// Yes, share the picture.

await context.PostAsync("Posting tweet.");

}

else

{

// No, don't share the picture.

await context.PostAsync("OK, I won't share it.");

}

}

// Since none of the scorables in previous group won, the dialog sends a help message.

[MethodBind]

[ScorableGroup(2)]

public async Task Default(IDialogContext context, IActivity activity)

{

await context.PostAsync("I'm sorry. I didn't understand you.");

await context.PostAsync("You can tell me to find photos, tweet them, and order prints. Here is an example: \"find pictures of food\".");

}

}

1. Debug your Bot application.

Congratz , now you had complete Part 1 and Part 2.