

.Net Bot Framework

What is a Bot ?

A bot is a Robot. In case of this framework it is a new method to interact with the user by using speech in written or spoken form.

Businesscases

(Enter Business Cases here)

The core concepts

Channel

A Channel is a connection between the bot and communication apps. (All Microsoft communication Apps)

Bot Connector

Interface for the Channels, which designs messages for each app. (The developer doesn't need to design a message for each channel). It is possible to connect to multiple channels at the same time with the bot.

Activity

The bot Connector uses an activity to communicate with the Channel. The activity contains messages.

Message

A message contains the information the user or the bot wants to transmit. This could be text(String) or a Richcard.

Dialog

Dialogs are similar to MVC Controller. They can post information.

Richcards

Richcards are media values like gifs, which the bot can send to the user and to the other direction.

How to use .Net Bot Framework?

There are two ways to use the Botframework:

1. Azure
2. Visual Studio 2017

It is possible to write the Bot in Csharp or Node.js.

Requirements

1. Install Visual Studio 2017 for Windows.
2. In Visual Studio, update all extensions to their latest versions.
3. Download the Bot Application, Bot Controller, and Bot Dialog .zip files. Install the templates by copying the .zip files to your Visual Studio 2017 project templates directory.

Create Project

1. Right-click on the project and select Manage NuGet Packages.
2. In the Browse tab, type "Microsoft.Bot.Builder".
3. Locate the Microsoft.Bot.Builder package in the list of search results, and click the Update button for that package.
4. Follow the prompts to accept the changes and update the package.

Debug the Bot with Bot Framework emulator

1. Run the Project.
2. It will start your browser. Copy the URL
3. Start the Emulator
4. Paste the URL to the url prompt add "<the url>/api/messages"

Microsoft has many API's which could be used (most of them are not for free)

1. Visit <https://dev.botframework.com/>
2. Sign in with your Microsoft ID or Register
3. Click "Create a Bot"
4. Select "Register an existing bot built using Bot Builder SDK"
5. Fill the form
6. Click ok
7. Click Create Microsoft App ID and password

MicrosoftAppID and MicrosoftAppPassword are Key Values in the web.config

Coding

Send a Message:

It is possible to send messages to the user Everywhere, but you should only do this in a Dialog.

```
string message = $"Sorry I did not understand: ";
await context.PostAsync(message);
```

A Simple Dialog:

```
[Serializable]
public class EchoDialog : IDialog<object>
{
    public async Task StartAsync(IDialogContext context)
    {
        context.Wait(MessageReceivedAsync);
    }

    public async Task MessageReceivedAsync(IDialogContext context, IAwaitable<IMessageActivity> argument)
    {
        var message = await argument;
        await context.PostAsync("You said: " + message.Text);
        context.Wait(MessageReceivedAsync);
    }
}
```

A Dialog implements the Interface IDialog. The Implemented Methods starts the Dialog and receive Messages. To continue receiving the messages. The context have to know which method comes next. It is possible to set the Method in:

```
context.Wait(MessageReceivedAsync);
```

Every method, which handles a User Interaction should run async and return a Task. To deliver the conversation to another Dialog it is possible to forward or call a Dialog. If you call a Dialog the calling Dialog terminates after the called Dialog.

```
await context.Forward(new MyDialog(), ResumeAfter, msg, CancellationToken.None);
```

The „ResumeAfter“ method has to be implemented with minimum one constructor IAwaitable<object>. Overloading the method could parse the object:

```
public async Task ResumeAfter(IDialogContext context, IAwaitable<Place> result)
{
```

(The method is overloaded with the object Place)

Links

Bot Application	http://aka.ms/bf-bc-vstemplate
Bot Controller	http://aka.ms/bf-bc-vscontrollertemplate
Bot Dialog	http://aka.ms/bf-bc-vsdialogtemplate
Visual Studio Community	https://www.visualstudio.com/de/vs/community/
E-Portfolio Git	https://github.com/jupie/eportfolio.git