Contents

[Update & Remaining Issue 1](#_Toc59184463)

[Goal 2](#_Toc59184464)

[Recreate Issue 2](#_Toc59184465)

[Issues 2](#_Toc59184466)

[Steps to create the code 3](#_Toc59184467)

[Online Instruction Being Followed 3](#_Toc59184468)

[Create New Skill Template Project 3](#_Toc59184469)

[Clean unused code 4](#_Toc59184470)

[Create Skill 4](#_Toc59184471)

[Implementing changes to integrate composer dialogs into skill 5](#_Toc59184472)

[Testing Changes 6](#_Toc59184473)

[First Error (partially resolved) 6](#_Toc59184474)

[Second Error (unresolved) 8](#_Toc59184475)

[Error & Stack Trace 8](#_Toc59184476)

[3rd Issue 10](#_Toc59184477)

# Update & Remaining Issue

I was able to download the Bot.Builder SDK, compile & connect to the project to be able to debug the issue.

Determined that the issue was in Microsoft.Bot.Builder.Dialogs.Adaptive.Actions.BeginSkill.cs - Line 157

DialogOptions.ConversationIdFactory = dc.Context.TurnState.Get<SkillConversationIdFactoryBase>() ?? throw new NullReferenceException("Unable to locate SkillConversationIdFactoryBase in HostContext");

Determined that if I add dependency injection of SkillConversationIdFactoryBase & BotFrameworkClient I am able to by pass the issue.

After these changes the skill is called indicated by breakpoints in EchoBot.cs being hit.

The emulator does not display the expected text when the skill is called. I wonder if there is now an issue with the callback URL not getting back to the emulator.

# Goal

Create a workflow which allows composer created dialogs to be integrated into a bot skill project.

This will allow for customizations to the bot behavior such as, logging errors & conversation to a custom database hosted on premise

We will also be hosting the bot on premise so this structure will also allow for our CI/CD pipeline to deploy the main bot (created through the composer) & skills (created in code) to be deployed together & settings to be adjusted as the Bot moves up environments

# Recreate Issue

The final project can be found here:

<https://github.com/silverbulletgt/Integrate-Composer-Dialog-Using-Skill>

To recreate the issue:

Prerequisites

* Install Bot Emulator: <https://github.com/Microsoft/BotFramework-Emulator/blob/master/README.md>

Open the solution file

<https://github.com/silverbulletgt/Integrate-Composer-Dialog-Using-Skill/blob/master/Integrate-Composer-Dialog-Using-Skill.sln>

Put a breakpoint on line 72 of DefaultAdapter.cs - this is where the error can be seen

Start debugging the project

Open Bot Emulator

Connect to the bot: <http://localhost:3978/api/messages>

Type "Greeting"

Bot should respond with "Hello, I have recognized that you said greeting"

Type "Skill"

The breakpoint on line 72 of DefaultAdapter.cs should trigger giving details of the error

Error & stack trace can be found here: <https://github.com/silverbulletgt/Integrate-Composer-Dialog-Using-Skill/blob/master/Readme/error%20stack%20trace.txt>

# Issues

1st Issue (partially resolved)

Error is: Value cannot be null. (Parameter 'uriString')

Bypassed by changing various values in Composer-With-Skill.dialog to fixed values rather than retrieving from settings

"botId": "=settings.MicrosoftAppId",

"skillHostEndpoint": "=settings.skillHostEndpoint",

"connectionName": "=settings.connectionName",

"allowInterruptions": true,

"skillEndpoint": "=settings.skill['integrateComposerDialogUsingSkill'].endpointUrl",

"skillAppId": "=settings.skill['integrateComposerDialogUsingSkill'].msAppId",

Changed to:

"botId": "=settings.MicrosoftAppId",

"skillHostEndpoint": "<http://localhost:3978/api/skills>",

"connectionName": "=settings.connectionName",

"allowInterruptions": true,

"skillEndpoint": "<http://localhost:3978/api/echo/messages>",

"skillAppId": "00000000-0000-0000-0000-000000000000",

2nd Issue (resolved)

System.NullReferenceException: 'Unable to locate SkillConversationIdFactoryBase in HostContext'

3rd Issue (not resolved)

Emulator does not display response from Echo Skill after the Echo Skill code is executed.

# Steps to create the code

## Online Instruction Being Followed

Tutorial Being Followed - This tutorial outlines how to integrate composer created dialogs into a skill project

<https://microsoft.github.io/botframework-solutions/skills/handbook/experimental-add-composer/>

Skill Template can be found here:

<https://microsoft.github.io/botframework-solutions/skills/tutorials/create-skill/csharp/2-download-and-install/>

Bot Emulator can be found here: <https://github.com/Microsoft/BotFramework-Emulator/blob/master/README.md>

## Create New Skill Template Project

In Visual Studio create a new Skill Template project

Machine generated alternative text:
Create a new project 
Recent project templates 
Skill Template 
Unit Test Project (.NET Framework) 
ASP .NET Web Application (.NET Framework) 
Search for templates (Alt+S) 
All languages 
All platforms 
Clear all 
Web 
Skill Template 
Project template for a Bot Framework Skill. 
ma COS Windows 
Cloud 
Virtual Assistant Template 
Project template for a Bot Framework Virtual Assistant. 
ma COS Windows 
Cloud Web 
Not finding what you're looking for? 
Install more tools and features 

## Clean unused code

After I created the Skill project I removed unused dialogs, LUIS setup & cognitive models. These are not being used & were causing runtime errors later on in the setup.

Using Windows version of the Bot Composer:

<https://github.com/microsoft/BotFramework-Composer/releases/tag/v1.3.0>

Installed:

[BotFramework-Composer-1.3.0-windows-setup.exe](https://github.com/microsoft/BotFramework-Composer/releases/download/v1.3.0/BotFramework-Composer-1.3.0-windows-setup.exe)

**Create new Bot within the Composer**

This bot will use regular expressions to recognize intents.

It will have a "Greetings" trigger which triggers on "Greeting"

It will have a "Skill" trigger which will trigger on "Skill" & will call the skill.

I have placed the bot composer project directly within the skills project in the "ComposerDialogs" folder. This bypasses the need for the project to be "Exported as Zip" in the composer.

This will allow me to later Ignore this step

Machine generated alternative text:
Retrieve the Generated Files 
1. Within Composer, and your active project. Click the Export assets to . zip option under the Export Menu. This 
self-contained ZIP file contains all of your declarative assets making up your Composer project. 
Bot (VI D O) 
Bot Framework ToDoBotWithLuisSample-O (en-us) 
Cut 
Copy 
LG Move 
Delete 
Cf Export 
Export "sets to zip 
Export 's Skill 
80 
Filter Dialog 
+ Add 
WelcomelJser 
v,' Cl ndc Redo 
Unpack this ZIP file into a new sub-folder of your Skill project called composerDia10gs 
2. Copy the Generated Folder from your Composer Project into the same composerDia10gs folder. (Temporary) 

## Create Skill

Within the Skill project create a basic "Echo" bot skill.

The skill which will be implemented will be "Echo" from this sample: <https://github.com/microsoft/BotBuilder-Samples/tree/main/samples/csharp_dotnetcore/80.skills-simple-bot-to-bot>

This will involve the new Bot class, a controller (so the skill can be called) & a manifest file for the skill

I start the skill by debugging the Visual Studio project.

The skill is added to the composer project using (<http://localhost:3978/manifest/echobotskill-manifest-1.0.json> & choosing the production endpoint)& a new intent which triggers on "Skill" is created.

Machine generated alternative text:
Filter Dialog 
YOUR PROJECT 
ComposerDiaIogs 
ComposerDiaIogs 
Greetincs 
Skill 
Edit v 
Composer-With- 
Disable 
Skill 
Ski Il 
Connect to a skill 
Skill 
Intent recognized 
Connect to a skill 
Show code 
Skill Dialog 
Begin a remote skill. 
Learn more 
Skill Dialog Name 
Integrate_Composer Dialog Using Skill 
Show skill manifest 
Skill Endpoint @ 
production 
OAuth connection name (SSO) @ 
—settings.connectionName 
Activity O 
Sfturn.activity.text) 
Send response 
Will call the skill now 
to a skill 
Host zsettings.skiII('integrateComposerDiSoqlJsing... 

Starting the bot from the composer & testing in the emulator succeeds

Machine generated alternative text:
Will call the skill now 
Echo: Skill 
Say "end" or "stop" and I'll end the conversation and back to the parent. 
teuhtneoahuntoea 
2 minutes ago 
Echo: teuhtneoahuntoea 
Say "end" or "stop" and I'll end the conversation and back to the parent. 
Skill 
2 minutä ago 

## Implementing changes to integrate composer dialogs into skill

Following the tutorial here: <https://microsoft.github.io/botframework-solutions/skills/handbook/experimental-add-composer/>

To attempt to allow the Bot Emulator to connect through the Visual Studio project

Things I did differently than the tutorial outlines:

* Did not export as .zip
  + The Bot Composer folder was placed directly within the Skill project "ComposerDialogs" folder
* Used latest nuget packages (4.11.1)
* In MainDialog.cs changing line where you initialize the composer dialog to not have .dialog after the name. It worked this way & did not work with .dialog

return await stepContext.BeginDialogAsync("todobotwithluissample-0.dialog", adaptiveOptions, cancellationToken); To

return await stepContext.BeginDialogAsync("Composer-With-Skill", adaptiveOptions, cancellationToken);

* Ignoring LUIS key setup because not using LUIS

## Testing Changes

Start the debugger in Visual Studio

Open Bot Emulator & connect to the bot at <http://localhost:3978/api/messages>

Test the Greeting intent: working

Test the Skill intent:

Machine generated alternative text:
O Restart Comersatior - New user ID 
Szve transcript 
Greeting 
Hello, have recognized that you said greeting 
Skill 
Will call the skill now 
Apologies, looks like something went wrong. Try your request again later. 

## First Error (partially resolved)

Place a breakpoint on the error handling code in DefaultAdapter.cs

Machine generated alternative text:
File 
69 
72 
Edit 
- 
View Git 
Project 
Build 
Debug Test Analyze 
Any CPU 
Tools Extensions Window Help 
Search (Ctrl+Q 
Continue 
I ntegrate - Composer - Dialog 
DefaultAdapter.cs -E X 
Hand 
Ech080t. 
80tControIIer.cs 
Exception exception) 
(exception . MessageY ) ; 
NuGet: 
Ech080tControIIer.cs 
DefaultAdapter 
O 
this .1Jse (new 
this . LlseStorage (storage) ; 
this . LlseaotState (userState) ; 
this . Llse aotState (convers ationState) ; 
der, 1 hcur 1 i uthcr, 1 
private async Task HandleTurnErrorAsync(ITurnContext turncontext, 
// Log any leaked exception from the application. 
logger. LogError(exception, S" COnTurnError) unhandled error : 
await SendErrorMessageAsync (turncontext, exception); 
await SendEndOfConversationToParentAsync (turncontext, exception); 
await ClearConversationStateAsync (turncontext); 
der, 1 hcur 1 i uthcr, 1 
Exception exception) 
private async Task SendErrorMessageAsync(ITurnContext turncontext, 

Error is: Value cannot be null. (Parameter 'uriString')

This error is occurring because the setting values within Composer-With-Skill.dialog are not being resolved

Machine generated alternative text:
Composer-With-SkiII.diaIog -E X 
"intent": "Greetings' 
'pattern": "Greeting" 
NuGet: 
BctCcntrcIIer.cs 
Ech08ctCcntrcIIer.cs 
Schema: https://raw.githubusercontent.com/microsoft/BotFramework-Composer/stabWComposer/packages/server/schemas/sdk.schema 
ac 
45 
46 
47 
67 
68 
72 
- 
- 
- 
- 
- 
- 
"$kind": 
"Microsoft. aeginSkiII" , 
"$designer" . 
"taIrjLl" 
"activityprocessed": true, 
bctld ' " —settings . MicrosoftAppId" , 
"skillHostEndpoint": "—settings . skillHostEndpoint" , 
connection Name": "—settings . connectionName" 
"allowlnterruptions": true, 
"skillEndpoint": "—settings integrateComposerDiaIogLlsingSkiII') . endpointLlrI", 
"skillAppId": "—settings . ' integrateComposerDiaIogLlsingSkiII ' ) .msAppId" , 
"activity": "$faeginSkiII Activity 
"$schema" . 
"https://raw.eithubuserccntent.com/microscft/actFramewcrk-Ccmpcser/stable/Ccmpcser/packages/se'wer/schemas/sdk.schema 
generator" 
"Composer-with-Skill. Ig" , 
"Composer-With -Skill" , 
'recognizer ' 
"$kind": 
"Microsoft. RegexRecognizer" , 
"intents" . 
"intent" • 
. "skill", 
'pattern" : 
"skill" 

I expect they are meant to be picked up from the ComposerDialogs/settings/appsettings.json file

Machine generated alternative text:
Composer-With-SkiII.diaIcg 
Dialcg_using_ 
Integrate_Ccmpcser... 
Schema: https://json.schemastore.org/appsettings 
Using_SkiII.csprcJ 
NuGet: 
Integrate_ 
Skill 
EchcBct.cs 
BctCcntrcIIer.cs 
EchcBctCcntrcIIer.cs 
DefaultAdapter.cs 
appsettings.json -E X 
49 
58 
61 
63 
67 
68 
74 
79 
- 
- 
- 
- 
- 
- 
- 
- 
qna eglan : 
wes us 
' 'telemetry": 
" logpersonallnformation " : 
"10gActivities " . 
• true 
"runtime' : 
'customRuntime": false, 
"path": ' 
'command" . 
"downsampling' . 
"maxImbaIanceRatio" . 
" skillConfiguration " : 
"isSkiII": 
false, 
"allowedCaIIers" . 
"skill": 
false, 
'lintegrateccmpcs erDia gSkiII" • 
" endpointLlrI" : 
'http://IccaIhcst:3978/api/echc/messages' , 
"msApp1d": "øøøøøøøø-øøøø-øøøø-øøøø-øøøøøøøøøøøø" 
"defaultLanguage": "en-us 
"languages' . 
'en-us 
customFunctions' . 
"skillHostEndpoint" . 
"http://localhcst: 398B/ api/ skills' 

& that the line added to the Startup.cs file is meant to cause this to work

Machine generated alternative text:
Composer-With-SkiII.diaIog 
NuGet: 
Skill 
EchcBct.cs 
BctCcntrcIIer.cs 
Ech08ctCcntrcIIer.cs 
Default4cIapter.cE 
appsettings.jscn 
Startup.cs -E X 
G) Startup(IWebHostEnVlronment env) 
36 
38 
using Micros t. Extensions . Hosting; 
using System.CoIIections.Generic; 
using System. 10; 
Enamespace Integrate Composer 
- 
- 
public class 
Startu p 
public Startup(IWebHostEnvironment env) 
var builder 
new Configurationauilder() 
. Seta asepath (env. ContentRootPath ) 
. json", optional: true, reloadOnChange: true) 
.AddJsonFiIe . (env. EnvironmentNameY.json", optional: true) 
. json", optional: true) 
.AddJsonFiIe (env. EnvironmentNameY.json", optional: true) 
// from instructions: 
https : / rcscft . sit hub. ic/ botframewcrk - solutions / s kills / handbook/ experimental- add - composer/ 
.AddiIe($ " Compo s e rD ia 
. AddEnvironmentVariabIes(); 
class System.String 
Represents text as a sequence of UTF- 16 code units. 
Configuration builder. auild(); 
/ 'from instructions: https://micrcscft.github.ic/bctframewcrk-scluticns/skills/handbcck/experimental-add-ccmpcser/ 
this .HostingEnvironment env; 

It's not working though so I manually add the values to the Composer-With-Skill.dialog file to bypass this issue.

As well as change the skillEndpoint to be <http://localhost:3978/api/skills> instead of <http://localhost:3980/api/skills> to adjust for the debugging port on Visual Studio compared to the Bot Composer

## Second Error (unresolved)

Now running the Bot again and calling the "Skill" intent causes the below error

Machine generated alternative text:
Composer-With-SkiII.diaIog 
Ech08ct.cs 
var ac IVI 
if (activity. 
80tControIIer.cs 
c. on 
Ech080tControIIer.cs 
DefaultAdapter.cs 
appsettings.jscn 
Startup.cs 
Exception Thrown 
MainDiaIog.cs -E X 
IntroStepAsync(WaterfaIIStepContext stepContext, CancellationToken cal 
— Inne 
Type 
! string. IsNuIIOrEmpty (a ctivity. Text)) 
le2 
les 
les 
les 
lea 
leg 
Ile 
111 
112 
113 
114 
115 
116 
117 
118 
119 
12B 
121 
122 
123 
124 
125 
126 
127 
128 
129 
131 
132 
133 
135 
- 
- 
- 
- 
- 
- 
Activity Types . Message 88 
return interrupted; 
// Handles introduction/ continuation prompt logic. 
1 Euthcr, 
private async Task(DiaIcgTurnResuIO IntroStepAsync(WaterfaIIStepContext stepContext, 
if (stepContext.Context. IsSkiII()) 
// If the bot is in skill mode, skip directly to 'X•ute and do not prompt 
return await stepContext.NextAsync(canceIIationToken: cbnceIIationToken); 
cancellationT0ken 
cancellationToke 
object adaptiveoptions 
null; 
rn await stepContext. BeginDiaIogAsync( "Composer-with-Ski 
er, 1 hcur 1 i uthcr, 1 
adaptiveOpt 
canc 
canc 
private async Task LogLlserOutAsync(DiaIogContext dc, CancellationToken cancellation Token) 
var supported dc.Context.Adapter is ILIserTokenProvider; 
System.NuIIReferenceException: Unable to locate 
SkillConversatlonIdFactory8ase in HostContext' 
This exception was originally thrown at this call stack: 
'External Codel 
View Details Copy Details Start Live Share session.„ 
Exception Settings 
Break when this exception type is thrown 
Except when thrown from: 
Using Skil .dll 
Open Exception Settings Edit Conditions 
if (supported) 
var tokenprovider 
// Sign out user 
(ILIserToken Provid er)dc . Context. Adapter ; 
var tokens 
await tokenProvider.GetTokenStatusAsync(dc.Context, dc.Context.Activity.From. Id, 
foreach (var token in tokens) 
await tokenProvider.SignOutLlserAsync(dc .Context, token .ConnectionName, cancellationToken: 
// Cancel all active dialogs 
await dc.CanceIAIIDiaIogsAsync(canceIIationToken); 

## Error & Stack Trace

System.NullReferenceException: 'Unable to locate SkillConversationIdFactoryBase in HostContext'

at Microsoft.Bot.Builder.Dialogs.Adaptive.Actions.BeginSkill.<BeginDialogAsync>d\_\_43.MoveNext()

at System.Runtime.ExceptionServices.ExceptionDispatchInfo.Throw()

at System.Runtime.CompilerServices.TaskAwaiter.ThrowForNonSuccess(Task task)

at System.Runtime.CompilerServices.TaskAwaiter.HandleNonSuccessAndDebuggerNotification(Task task)

at System.Runtime.CompilerServices.ConfiguredTaskAwaitable`1.ConfiguredTaskAwaiter.GetResult()

at Microsoft.Bot.Builder.Dialogs.DialogContext.<BeginDialogAsync>d\_\_32.MoveNext()

at System.Runtime.ExceptionServices.ExceptionDispatchInfo.Throw()

at System.Runtime.CompilerServices.TaskAwaiter.ThrowForNonSuccess(Task task)

at System.Runtime.CompilerServices.TaskAwaiter.HandleNonSuccessAndDebuggerNotification(Task task)

at System.Runtime.CompilerServices.ConfiguredTaskAwaitable`1.ConfiguredTaskAwaiter.GetResult()

at Microsoft.Bot.Builder.Dialogs.Adaptive.Actions.ActionScope.<BeginActionAsync>d\_\_17.MoveNext()

at System.Runtime.ExceptionServices.ExceptionDispatchInfo.Throw()

at System.Runtime.CompilerServices.TaskAwaiter.ThrowForNonSuccess(Task task)

at System.Runtime.CompilerServices.TaskAwaiter.HandleNonSuccessAndDebuggerNotification(Task task)

at Microsoft.Bot.Builder.Dialogs.Adaptive.Actions.ActionScope.<OnNextActionAsync>d\_\_15.MoveNext()

at System.Runtime.ExceptionServices.ExceptionDispatchInfo.Throw()

at System.Runtime.CompilerServices.TaskAwaiter.ThrowForNonSuccess(Task task)

at System.Runtime.CompilerServices.TaskAwaiter.HandleNonSuccessAndDebuggerNotification(Task task)

at System.Runtime.CompilerServices.ConfiguredTaskAwaitable`1.ConfiguredTaskAwaiter.GetResult()

at Microsoft.Bot.Builder.Dialogs.Adaptive.Actions.ActionScope.<ResumeDialogAsync>d\_\_8.MoveNext()

at System.Runtime.ExceptionServices.ExceptionDispatchInfo.Throw()

at System.Runtime.CompilerServices.TaskAwaiter.ThrowForNonSuccess(Task task)

at System.Runtime.CompilerServices.TaskAwaiter.HandleNonSuccessAndDebuggerNotification(Task task)

at System.Runtime.CompilerServices.ConfiguredTaskAwaitable`1.ConfiguredTaskAwaiter.GetResult()

at Microsoft.Bot.Builder.Dialogs.DialogContext.<EndDialogAsync>d\_\_35.MoveNext()

at System.Runtime.ExceptionServices.ExceptionDispatchInfo.Throw()

at System.Runtime.CompilerServices.TaskAwaiter.ThrowForNonSuccess(Task task)

at System.Runtime.CompilerServices.TaskAwaiter.HandleNonSuccessAndDebuggerNotification(Task task)

at System.Runtime.CompilerServices.ConfiguredTaskAwaitable`1.ConfiguredTaskAwaiter.GetResult()

at Microsoft.Bot.Builder.Dialogs.Adaptive.Actions.SendActivity.<BeginDialogAsync>d\_\_11.MoveNext()

at System.Runtime.ExceptionServices.ExceptionDispatchInfo.Throw()

at System.Runtime.CompilerServices.TaskAwaiter.ThrowForNonSuccess(Task task)

at System.Runtime.CompilerServices.TaskAwaiter.HandleNonSuccessAndDebuggerNotification(Task task)

at System.Runtime.CompilerServices.ConfiguredTaskAwaitable`1.ConfiguredTaskAwaiter.GetResult()

at Microsoft.Bot.Builder.Dialogs.DialogContext.<BeginDialogAsync>d\_\_32.MoveNext()

at System.Runtime.ExceptionServices.ExceptionDispatchInfo.Throw()

at System.Runtime.CompilerServices.TaskAwaiter.ThrowForNonSuccess(Task task)

at System.Runtime.CompilerServices.TaskAwaiter.HandleNonSuccessAndDebuggerNotification(Task task)

at System.Runtime.CompilerServices.ConfiguredTaskAwaitable`1.ConfiguredTaskAwaiter.GetResult()

at Microsoft.Bot.Builder.Dialogs.Adaptive.Actions.ActionScope.<BeginActionAsync>d\_\_17.MoveNext()

at System.Runtime.ExceptionServices.ExceptionDispatchInfo.Throw()

at System.Runtime.CompilerServices.TaskAwaiter.ThrowForNonSuccess(Task task)

at System.Runtime.CompilerServices.TaskAwaiter.HandleNonSuccessAndDebuggerNotification(Task task)

at System.Runtime.CompilerServices.ConfiguredTaskAwaitable`1.ConfiguredTaskAwaiter.GetResult()

at Microsoft.Bot.Builder.Dialogs.Adaptive.Actions.ActionScope.<BeginDialogAsync>d\_\_6.MoveNext()

at System.Runtime.ExceptionServices.ExceptionDispatchInfo.Throw()

at System.Runtime.CompilerServices.TaskAwaiter.ThrowForNonSuccess(Task task)

at System.Runtime.CompilerServices.TaskAwaiter.HandleNonSuccessAndDebuggerNotification(Task task)

at System.Runtime.CompilerServices.ConfiguredTaskAwaitable`1.ConfiguredTaskAwaiter.GetResult()

at Microsoft.Bot.Builder.Dialogs.DialogContext.<BeginDialogAsync>d\_\_32.MoveNext()

at System.Runtime.ExceptionServices.ExceptionDispatchInfo.Throw()

at System.Runtime.CompilerServices.TaskAwaiter.ThrowForNonSuccess(Task task)

at Microsoft.Bot.Builder.Dialogs.Adaptive.AdaptiveDialog.<ContinueActionsAsync>d\_\_56.MoveNext()

at System.Runtime.ExceptionServices.ExceptionDispatchInfo.Throw()

at System.Runtime.CompilerServices.TaskAwaiter.ThrowForNonSuccess(Task task)

at System.Runtime.CompilerServices.TaskAwaiter.HandleNonSuccessAndDebuggerNotification(Task task)

at Microsoft.Bot.Builder.Dialogs.Adaptive.AdaptiveDialog.<BeginDialogAsync>d\_\_45.MoveNext()

at System.Runtime.ExceptionServices.ExceptionDispatchInfo.Throw()

at System.Runtime.CompilerServices.TaskAwaiter.ThrowForNonSuccess(Task task)

at System.Runtime.CompilerServices.TaskAwaiter.HandleNonSuccessAndDebuggerNotification(Task task)

at System.Runtime.CompilerServices.ConfiguredTaskAwaitable`1.ConfiguredTaskAwaiter.GetResult()

at Microsoft.Bot.Builder.Dialogs.DialogContext.<BeginDialogAsync>d\_\_32.MoveNext()

at System.Runtime.ExceptionServices.ExceptionDispatchInfo.Throw()

at System.Runtime.CompilerServices.TaskAwaiter.ThrowForNonSuccess(Task task)

at System.Runtime.CompilerServices.TaskAwaiter.HandleNonSuccessAndDebuggerNotification(Task task)

at System.Runtime.CompilerServices.TaskAwaiter`1.GetResult()

at Integrate\_Composer\_Dialog\_Using\_Skill.Dialogs.MainDialog.<IntroStepAsync>d\_\_6.MoveNext() in C:\Integrate-Composer-Dialog-Using-Skill\Integrate-Composer-Dialog-Using-Skill\Integrate\_Composer\_Dialog\_Using\_Skill\Dialogs\MainDialog.cs:line 117

## 3rd Issue

To resolve the 2nd issue added the code:

DefaultAdapter.cs

private readonly SkillConversationIdFactoryBase \_skillConversationIdFactoryBase;

parameters to the constructor:

SkillConversationIdFactoryBase skillConversationIdFactoryBase,

BotFrameworkClient botFrameworkClient

To the body of the constructor:

Use(new RegisterClassMiddleware<SkillConversationIdFactoryBase>(\_skillConversationIdFactoryBase));

Use(new RegisterClassMiddleware<BotFrameworkClient>(botFrameworkClient));

Startup.cs

services.AddSingleton<SkillConversationIdFactoryBase, SkillConversationIdFactory>();

//to support the needed parameters on the BotFrameworkClient constructor

services.AddSingleton<HttpClient>(new HttpClient());

services.AddSingleton<BotFrameworkClient, BotFrameworkHttpClient>();

The Skill is called successfully though nothing comes back on the Emulator.

The error occurring when the skill tries to return the turnContext.SendActivityAsync is:

Error occurs after this line executes in EchoBot.cs

var messageText = $"Echo: {turnContext.Activity.Text}";

await turnContext.SendActivityAsync(MessageFactory.Text(messageText, messageText, InputHints.IgnoringInput), cancellationToken);

Operation returned an invalid status code 'NotFound'

at Microsoft.Bot.Connector.Conversations.<ReplyToActivityWithHttpMessagesAsync>d\_\_10.MoveNext()

This error was occurring because there was not api/skills endpoint in the project. I suppose that the Bot Composer adds this in the background somehow.

To fix this I added the SkillController.cs from <https://github.com/microsoft/botbuilder-dotnet/blob/main/tests/Microsoft.Bot.Builder.TestBot.Json/Controllers/SkillController.cs>

Then added dependency injection to support the controller.

I tested a basic intent which does not use a skill to determine if there would be any conflict with the dependency injection impacting those. The basic intent worked as expected though.

//to support SkillController.cs

//from: https://github.com/microsoft/botbuilder-dotnet/blob/main/tests/Microsoft.Bot.Builder.TestBot.Json/Startup.cs

services.AddSingleton<ChannelServiceHandler, SkillHandler>();

services.AddSingleton<BotAdapter>(sp => (BotFrameworkHttpAdapter)sp.GetService<IBotFrameworkHttpAdapter>());