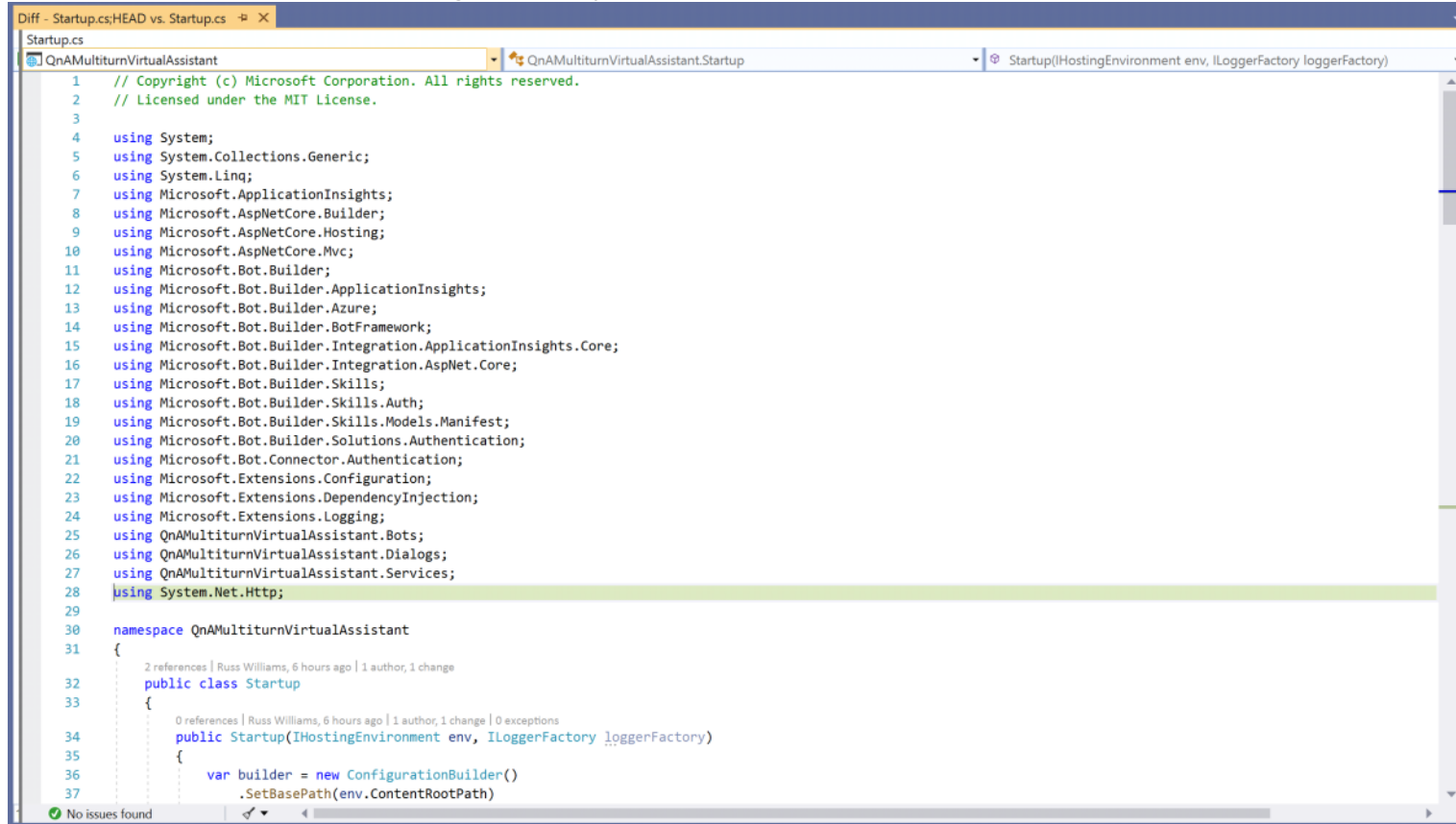


Changes to Startup.cs

Thursday, June 13, 2019 1:40 AM

There are two sets of changes you need to make to the Startup.cs file.

Diff screen shot to show first set of changes to Startup.cs



```
Diff - Startup.cs:HEAD vs. Startup.cs
Startup.cs
QnAMultiturnVirtualAssistant Startup
Startup(IHostingEnvironment env, ILoggerFactory loggerFactory)

1 // Copyright (c) Microsoft Corporation. All rights reserved.
2 // Licensed under the MIT License.
3
4 using System;
5 using System.Collections.Generic;
6 using System.Linq;
7 using Microsoft.ApplicationInsights;
8 using Microsoft.AspNetCore.Builder;
9 using Microsoft.AspNetCore.Hosting;
10 using Microsoft.AspNetCore.Mvc;
11 using Microsoft.Bot.Builder;
12 using Microsoft.Bot.Builder.ApplicationInsights;
13 using Microsoft.Bot.Builder.Azure;
14 using Microsoft.Bot.Builder.BotFramework;
15 using Microsoft.Bot.Builder.Integration.ApplicationInsights.Core;
16 using Microsoft.Bot.Builder.Integration.AspNet.Core;
17 using Microsoft.Bot.Builder.Skills;
18 using Microsoft.Bot.Builder.Skills.Auth;
19 using Microsoft.Bot.Builder.Skills.Models.Manifest;
20 using Microsoft.Bot.Builder.Solutions.Authentication;
21 using Microsoft.Bot.Connector.Authentication;
22 using Microsoft.Extensions.Configuration;
23 using Microsoft.Extensions.DependencyInjection;
24 using Microsoft.Extensions.Logging;
25 using QnAMultiturnVirtualAssistant.Bots;
26 using QnAMultiturnVirtualAssistant.Dialogs;
27 using QnAMultiturnVirtualAssistant.Services;
28 using System.Net.Http;
29
30 namespace QnAMultiturnVirtualAssistant
31 {
32     2 references | Russ Williams, 6 hours ago | 1 author, 1 change
33     public class Startup
34     {
35         0 references | Russ Williams, 6 hours ago | 1 author, 1 change | 0 exceptions
36         public Startup(IHostingEnvironment env, ILoggerFactory loggerFactory)
37         {
38             var builder = new ConfigurationBuilder()
39                 .SetBasePath(env.ContentRootPath)
40         }
41     }
42 }
```

Snippet for easy cut & paste (Click Download button and Open to copy bolded lines)

```
// Copyright (c) Microsoft Corporation. All rights reserved.
// Licensed under the MIT License.
```

```
using System;
using System.Collections.Generic;
using System.Linq;
using Microsoft.ApplicationInsights;
using Microsoft.AspNetCore.Builder;
using Microsoft.AspNetCore.Hosting;
using Microsoft.AspNetCore.Mvc;
using Microsoft.Bot.Builder;
using Microsoft.Bot.Builder.ApplicationInsights;
using Microsoft.Bot.Builder.Azure;
using Microsoft.Bot.Builder.BotFramework;
using Microsoft.Bot.Builder.Integration.ApplicationInsights.Core;
using Microsoft.Bot.Builder.Integration.AspNet.Core;
using Microsoft.Bot.Builder.Skills;
using Microsoft.Bot.Builder.Skills.Auth;
using Microsoft.Bot.Builder.Skills.Models.Manifest;
using Microsoft.Bot.Builder.Solutions.Authentication;
using Microsoft.Bot.Connector.Authentication;
using Microsoft.Extensions.Configuration;
using Microsoft.Extensions.DependencyInjection;
using Microsoft.Extensions.Logging;
using QnAMultiturnVirtualAssistant.Bots;
using QnAMultiturnVirtualAssistant.Dialogs;
using QnAMultiturnVirtualAssistant.Services;
using System.Net.Http;
```

```
namespace QnAMultiturnVirtualAssistant
```

Diff screen shot to show first set of changes to Startup.cs

```
Diff - Startup.cs:HEAD vs. Startup.cs
Startup.cs
QnAMultiturnVirtualAssistant Startup
Startup(IHostingEnvironment env, ILoggerFactory loggerFactory)

91
92 // Register skill dialogs
93 services.AddTransient(sp =>
94 {
95     var userState = sp.GetService<UserState>();
96     var skillDialogs = new List<SkillDialog>();
97
98     foreach (var skill in settings.Skills)
99     {
100         var authDialog = BuildAuthDialog(skill, settings);
101         var credentials = new MicrosoftAppCredentialsEx(settings.MicrosoftAppId, settings.MicrosoftAppPassword, skill.MSAappId);
102         skillDialogs.Add(new SkillDialog(skill, credentials, telemetryClient, userState, authDialog));
103     }
104
105     return skillDialogs;
106 });
107
108 // Configure adapters
109 services.AddSingleton<IBotFrameworkHttpAdapter, DefaultAdapter>();
110
111 // Configure bot
112 services.AddTransient<IBot, DialogBot<MainDialog>>();
113 services.AddSingleton<HttpClient>(new HttpClient());
114
115
116 // This method gets called by the runtime. Use this method to configure the HTTP request pipeline.
117 0 references | Russ Williams, 6 hours ago | 1 author, 1 change | 0 exceptions
118 public void Configure(IApplicationBuilder app, IHostingEnvironment env)
119 {
120     if (env.IsDevelopment())
121     {
122         app.UseDeveloperExceptionPage();
123     }
124
125     app.UseBotApplicationInsights()
126         .UseDefaultFiles()
127         .UseStaticFiles()
128         .UseMvc();
129 }
```

Snippet for easy cut & paste (Click Download button and Open to copy bolded lines)

```
// Register skill dialogs
services.AddTransient(sp =>
{
    var userState = sp.GetService<UserState>();
    var skillDialogs = new List<SkillDialog>();

    foreach (var skill in settings.Skills)
    {
        var authDialog = BuildAuthDialog(skill, settings);
        var credentials = new MicrosoftAppCredentialsEx(settings.MicrosoftAppId, settings.MicrosoftAppPassword, skill.MSAappId);
        skillDialogs.Add(new SkillDialog(skill, credentials, telemetryClient, userState, authDialog));
    }

    return skillDialogs;
});

// Configure adapters
services.AddSingleton<IBotFrameworkHttpAdapter, DefaultAdapter>();

// Configure bot
services.AddTransient<IBot, DialogBot<MainDialog>>();
services.AddSingleton<HttpClient>(new HttpClient());
}

// This method gets called by the runtime. Use this method to configure the HTTP request pipeline.
public void Configure(IApplicationBuilder app, IHostingEnvironment env)
{
    if (env.IsDevelopment())
    {
        app.UseDeveloperExceptionPage();
    }
}
```