

TributeTest4

Add

Add to a team

Permissions

Modern Mentorship App Test4

Tribute is a modern mentorship app for the workplace that empowers employee growth, collaboration and connection. With Tribute for Microsoft Teams, you can now sign up, discover and connect with mentors and mentees across your organization, and manage your mentorship engagements.

To sign up or login, visit <https://app.tributementorship.com>. Your organization must have an active subscription to use this app. For questions, please contact info@tributementorship.com.

Here is what this app can do:

- Receive data and messages that I provide to it.
- Access my profile information such as my name, email address, company name, and preferred language.

Bots
Chat with the app to ask questions and find info

Personal app
Keep track of important content and info

Created by: Tribute Co.
Version 1.0.1

Permissions
This app will have permission to:

By using TributeTest4, you agree to the privacy policy and terms of use.

Set up TributeTest4 for a team

TributeTest4 will be available for the entire team, but you can start using it in the channel you choose.

Type a team or channel name

Search

- General
Tribute Co.
- General
Leapfrog Team

< Back

Set up

Microsoft Teams

App Studio Manifest editor

Create, debug, and deploy Teams apps

Home > Edit an app

TributeTest4
By Tribute Co.

Capabilities

- Bot

App details

2 Capabilities

- Bot

Finish

- Languages
- Domains and permissions
- App Manifest (preview)
- Test and distribute
- Delete TributeTest4

Set up TributeTest4 for a team

TributeTest4 will be available for the entire team, but you can start using it in the channel you choose.

Type a team or channel name

Leapfrog Team > General

< Back

Set up a bot

```

201 OnReferences
202 protected override async Task OnTeamsMembersAddedAsync(IIList<Microsoft.Bot.Schema.Teams.TeamsChannelAccount> membersAdded, TeamInfo teamInfo, ITurnContext<IConversationUpdateActivity> turnContext, CancellationToken cancellationToken)
203 {
204     // sds:1800:83000
205
206     var team = await TeamsInfo.GetTeamDetailsAsync(turnContext);
207     //Cache.Set("channelId", context.Activity.ChannelId, new { });
208     //ConversationUpdate activity type invoked when user initiates a conversation
209     //If (context.Activity.Type == Microsoft.Bot.Schema.ActivityTypes.ConversationUpdate)
210     //{
211         // --> Get Teams Extensions.
212         // var dsdf = Microsoft.Bot.Builder.Teams.TeamsInfo.GetTeamInformation(turnContext);
213         var teamInformation = turnContext.Activity.TeamsGetTeamInformation();
214         TeamDetails teamDetails = await TeamsInfo.GetTeamDetailsAsync(turnContext);
215         //IEnumerable<TeamsChannelAccount> members = (IEnumerable<TeamsChannelAccount>)turnContext.Activity.TeamsMembers;
216         foreach (var member in members)
217         {
218             //Logger.LogInformation(member.Name);
219         }
220     }
221     //protected async Task OnTeamsMembersAddedAsync(IIList<TeamsChannelAccount> membersAdded, TeamInfo teamInfo, ITurnContext<IConversationUpdateActivity> turnContext, CancellationToken cancellationToken)
222     //if (turnContext.Activity.ChannelId != "directline" && turnContext.Activity.ChannelId != "webchat")
223     //{
224         //foreach (var member in membersAdded)
225         //{
226             //if (member.Id != turnContext.Activity.Recipient.Id)
227             //{
228                 //await ShowSignInCard(turnContext, cancellationToken);
229             //}
230         //}
231     //}
232     //}
233     //}
234     //}
235     //}
236     //}
237     //}
238     //}
239     //}

```

```

193 OnReferences
194 protected override async Task OnEventActivityAsync(ITurnContext<Microsoft.Bot.Schema.IEventActivity> turnContext, CancellationToken cancellationToken)
195 {
196     if (turnContext.Activity.Name == "webchat/join")
197     {
198         await ShowSignInCard(turnContext, cancellationToken);
199     }
200 }
201
202 OnReferences
203 protected override async Task OnTeamsMembersAddedAsync(IIList<Microsoft.Bot.Schema.Teams.TeamsChannelAccount> membersAdded, TeamInfo teamInfo, ITurnContext<IConversationUpdateActivity> turnContext, CancellationToken cancellationToken)
204 {
205     // sds:1800:83000
206     var teamInformation = turnContext.Activity.TeamsGetTeamInformation();
207     TeamDetails teamDetails = await TeamsInfo.GetTeamDetailsAsync(turnContext, turnContext.Activity.TeamsGetTeamInformation().Id, cancellationToken);
208     //IEnumerable<TeamsChannelAccount> members = (IEnumerable<TeamsChannelAccount>)turnContext.Activity.TeamsMembers;
209     foreach (var member in members)
210     {
211         //Logger.LogInformation(member.Name);
212     }
213 }
214
215

```

```
EchoBot.cs BotController.cs AdapterWithErrorHandler.cs Startup.cs Microsoft.Bot.Builder.Samples.Bots.EchoBots OnTeamsMembersAddedAsync(List<TeamChannelAccount> membersAdded, TeamInfo teamInfo, ITurnContext)
193
194 @references
195 protected override async Task OnEventActivityAsync(ITurnContext<Microsoft.Bot.Schema.IEventActivity> turnContext, CancellationToken cancellationToken)
196 {
197     if (turnContext.Activity.Name == "webchat/join")
198     {
199         await ShowSignInCard(turnContext, cancellationToken);
200     }
201 }
202
203 @references
204 protected override async Task OnTeamsMembersAddedAsync(IList<Microsoft.Bot.Schema.Teams.TeamChannelAccount> membersAdded, TeamInfo teamInfo, ITurnContext<IConversationUpdateActivity> turnContext, CancellationToken cancellationToken)
205 {
206     var teamInformation = turnContext.Activity.TeamsGetTeamInfo();
207     TeamDetails teamDetails = await TeamsInfo.GetTeamDetailsAsync(turnContext, turnContext.Activity.TeamsGetTeamInfo().Id, cancellationToken);
208
209     IEnumerable<TeamChannelAccount> members = (IEnumerable<TeamChannelAccount>)await TeamsInfo.GetMembersAsync(turnContext, cancellationToken);
210     foreach (var member in members)
211     {
212         Logger.LogInformation(member.Name);
213     }
214 }
215
216 --
```

```
EchoBot.cs BotController.cs AdapterWithErrorHandler.cs Startup.cs Microsoft.Bot.Builder.Samples.Bots.EchoBots OnTeamsMembersAddedAsync(List<TeamChannelAccount> membersAdded, TeamInfo teamInfo, ITurnContext)
193
194 @references
195 protected override async Task OnEventActivityAsync(ITurnContext<Microsoft.Bot.Schema.IEventActivity> turnContext, CancellationToken cancellationToken)
196 {
197     if (turnContext.Activity.Name == "webchat/join")
198     {
199         await ShowSignInCard(turnContext, cancellationToken);
200     }
201 }
202
203 @references
204 protected override async Task OnTeamsMembersAddedAsync(IList<Microsoft.Bot.Schema.Teams.TeamChannelAccount> membersAdded, TeamInfo teamInfo, ITurnContext<IConversationUpdateActivity> turnContext, CancellationToken cancellationToken)
205 {
206     var teamInformation = turnContext.Activity.TeamsGetTeamInfo();
207     TeamDetails teamDetails = await TeamsInfo.GetTeamDetailsAsync(turnContext, turnContext.Activity.TeamsGetTeamInfo().Id, cancellationToken);
208
209     IEnumerable<TeamChannelAccount> members = (IEnumerable<TeamChannelAccount>)await TeamsInfo.GetMembersAsync(turnContext, cancellationToken);
210     foreach (var member in members)
211     {
212         Logger.LogInformation(member.Name);
213     }
214 }
215
216 //protected async Task OnTeamsMembersAddedAsync(IList<ChannelAccount> membersAdded, ITurnContext<IConversationUpdateActivity> turnContext, CancellationToken cancellationToken)
217 // {
218 //     if (turnContext.Activity.ChannelId != "directline" && turnContext.Activity.ChannelId != "webchat")
219 //     {
220 //         foreach (var member in membersAdded)
221 //         {
222 //             if (member.Id != turnContext.Activity.Recipient.Id)
223 //             {
224 //                 await ShowSignInCard(turnContext, cancellationToken);
225 //             }
226 //         }
227 //     }
228 // }
229
230 --
```

Exception Thrown

System.NullReferenceException: Object reference not set to an instance of an object.

Microsoft.Bot.Builder.Teams.ActivityExtensions.TeamsGetTeamInfo() returned null.

View Details | Copy Details | Start Live Share session...

Exception Settings

- Break when this exception type is thrown
- EchoBot.dll

Open Exception Settings | Edit Conditions