Getting up and running with the Enterprise Bot Template

Prerequisites

- Install the following (**NOTE: these are already installed on your lab VM**):
  - Enterprise Bot Template
  - .NET Core SDK (latest version)
  - Node Package manager
  - Bot Framework Emulator (latest version)
  - Azure CLI
  - Azure Bot Service command line (CLI) tools (latest versions)

    ```
    npm install -g ludown luis-apis qnamaker botdispatch msbot chatdown
    ```

  - Install the LuisGen tool

    ```
    dotnet tool install -g luisgen
    ```

Create your bot project

1. In Visual Studio, click **File > New Project**.
2. Under **Bot Framework**, select **Enterprise Bot Template**.
3. Name your project and click, **Create**.
4. Right click your project and click **Build** to restore your NuGet packages.

Deploy your Azure services

Enterprise Template Bots require the following Azure services for end to end operation:

- Azure Web App
- Azure Storage Account (Transcripts)
- Azure Application Insights (Telemetry)
- Azure CosmosDb (Conversation State storage)
- Azure Cognitive Services - Language Understanding
- Azure Cognitive Services - QnA Maker (includes Azure Search, Azure Web App)
- Azure Cognitive Services - Content Moderator (optional manual step)

The following steps will help you to deploy these services using the provided deployment scripts:

1. Retrieve your LUIS Authoring Key.

- Review this documentation page for the correct LUIS portal for the region you plan to deploy to. Note that www.luis.ai refers to the US region and an authoring key retrieved from this portal will not work with a europe deployment.
- Once signed in click on your name in the top right hand corner.
- Choose **Settings** and make a note of the Authoring Key for later use.

2. Open up a Command Prompt (**NOTE: if using a lab VM, you can open Command Prompt from the pin on your taskbar**).
3. Login into your Azure Account using the Azure CLI. You can find a list of subscriptions you have access to in the Azure Portal Subscriptions page.

```
az login
az account set --subscription "YOUR_SUBSCRIPTION_NAME"
```

4. Run the msbot clone services command to deploy your services and configure a .bot file in your project. **NOTE: After deployment is complete, you must make note of the bot file secret that is shown in the Command Prompt window for later use.**

```
msbot clone services --name "YOUR-BOT-NAME" --luisAuthoringKey
"YOUR_AUTHORING_KEY" --folder "DeploymentScripts\LOCALE_FOLDER" --location
"REGION"
```

Notes:

- The "YOUR-BOT-NAME" parameter must be **globally unique**, and may only contain lowercase letters, numbers, and dashes ("-").
- Ensure the deployment region you provide in this step matches the region of your LUIS authoring key portal (e.g. westus for luis.ai or westeurope for eu.luis.ai).
- There is a known issue some users may experience with provisioning an MSA AppId and Password. If you recieve this error **ERROR: Unable to provision MSA id automatically. Please pass them in as parameters and try again**, please go to https://apps.dev.microsoft.com and manually create a new application, making note of the AppId and Password/Secret. Run the above msbot clone services command again, providing two new arguments **--appId** and **--appSecret** with the values you've just retrieved. You may need to escape any special characters in the password that might be interpreted by the shell to be a command:

- For *Windows command prompt*, enclose the appSecret in double quotes.
  e.g. `msbot clone services --name xxxx --luisAuthoringKey xxxx --location xxxx--folder bot.recipt ***--appSecret "!|%gr%"***`
- For *Windows PowerShell, try passing in appSecret after the --% argument.
  e.g. `msbot clone services --name xxxx --luisAuthoringKey xxxx --location xxxx --folder bot.recipt ***--% --appSecret "!|%gr%"***`
- For *MacOS or Linux*, enclose the appSecret in single quotes. e.g. `msbot clone services --name xxxx --luisAuthoringKey xxxx --location xxxx --folder bot.recipt ***--appSecret '!|%gr%'***`

5. After deployment is complete, update **appsettings.json** with your bot file secret.

```
"botFilePath": ".\\YOUR_BOT_FILE.bot",
"botFileSecret": "YOUR_BOT_SECRET",
```

6. Run the following command and retrieve the InstrumentationKey for your Application Insights instance.

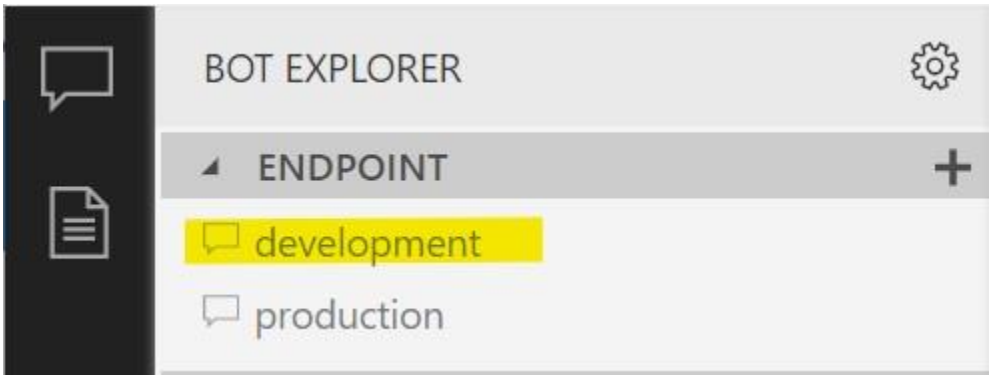
```
msbot list --bot YOUR_BOT_FILE.bot --secret "YOUR_BOT_SECRET"
```

7. Update **appsettings.json** with your instrumentation key:

```
"ApplicationInsights": {
    "InstrumentationKey": "YOUR_INSTRUMENTATION_KEY"
}
```

Test your bot

Once complete, run your bot project within your development environment and open the Bot Framework Emulator. Within the Emulator, click **File > Open Bot** and navigate to the .bot file in your directory.



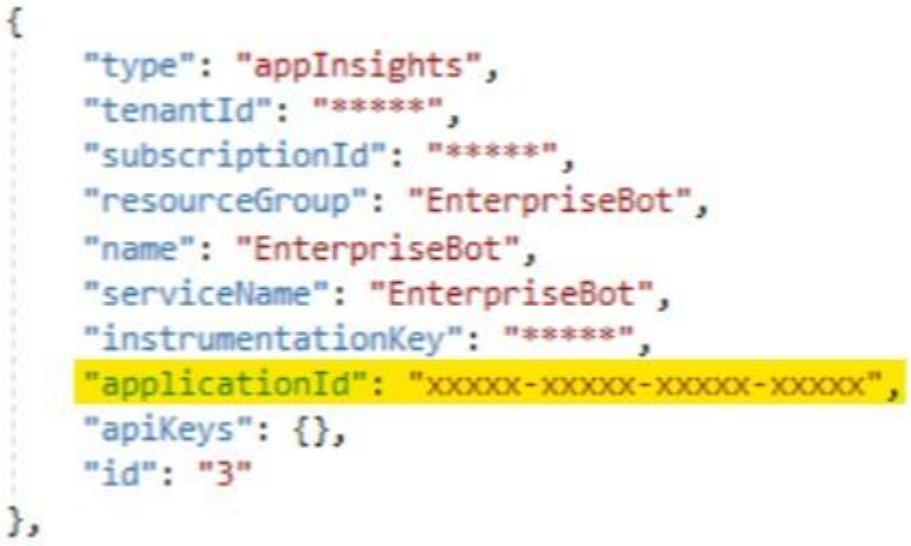
You should recieve an Introduction Card when the conversation begins.

Type **hi** to verify everything is working.

View your bot data

The Enterprise Bot Template comes with a preconfigured Power BI dashboard that connects to your Application Insights service to provide conversation analytics. Once you have tested your bot locally, you can open this dashboard to view the data.

1. Download the Power BI dashboard (.pbit file) here.
2. Open the dashboard in Power BI Desktop (installed on the lab VM).
3. Enter your Application Insights Application Id (found in your .bot file).



4. Learn more about the Power BI dashboard features here.