**演習 5: クラウドへのボットの展開 (Node.js)**

**概要**

この演習では、自分のボットを登録し、Azure に展開して、他の人々がそのボットを使用できるようにする方法を学習します。

**前提条件**

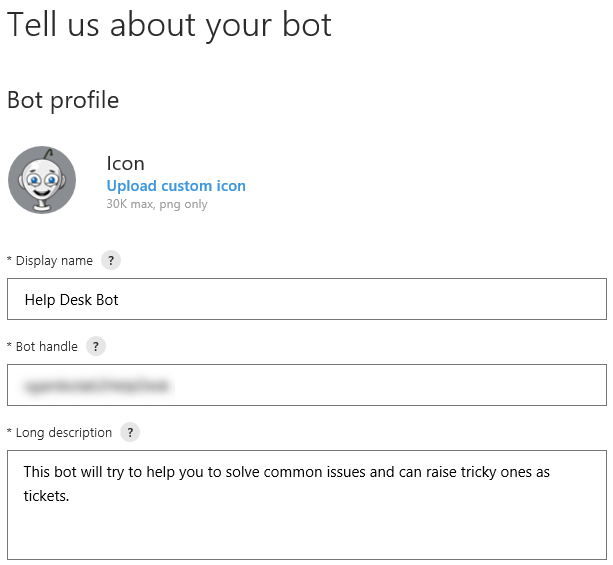
この演習を完了するには、以下のソフトウェアが必要です。

* [Visual Studio Code](https://code.visualstudio.com/download) (推奨) や Visual Studio 2017 Community 以上などのコード エディター
* [Azure](https://azureinfo.microsoft.com/us-freetrial.html?cr_cc=200744395&wt.mc_id=usdx_evan_events_reg_dev_0_iottour_0_0) サブスクリプション
* [LUIS ポータル](https://www.luis.ai/)のアカウント
* [Git コマンド ライン インターフェイス](https://git-scm.com/downloads)
* [Skype](https://www.skype.com/) アカウント (オプション)

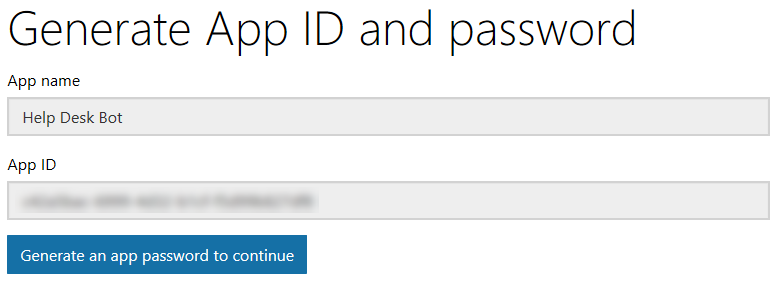
**タスク 1: Bot Framework へのボットの登録**

このタスクでは、アプリ ID とアプリ パスワードを生成し、ボットを登録します。

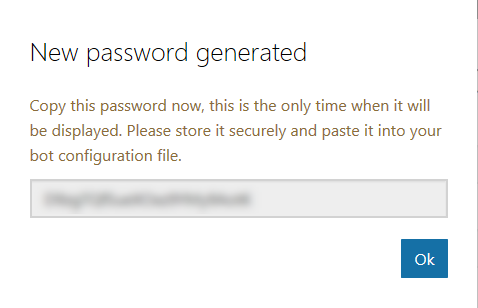
1. [Bot Framework Portal](https://dev.botframework.com/) にサインインします。
2. [My bots] ボタンをクリックし、次に [Create a bot] をクリックするか (他のボットがある場合)、または [Register] ボタンをクリックします。
3. [logo.png](https://github.com/GeekTrainer/help-desk-bot-lab/blob/develop/assets/logo.png) を**カスタム アイコン**としてアップロードします。
4. [Display Name] に「Help Desk Bot」と入力します。[Bot Handle] にグローバル一意アプリ名を入力します。
5. [Long Description] には、「This bot will try to help you to solve common issues and can raise tricky ones as tickets.」と入力してください。この説明は検索結果で表示されるため、ボットの機能を正確に記述してください。

[](https://github.com/GeekTrainer/help-desk-bot-lab/blob/develop/Node/images/exercise5-botname.png)

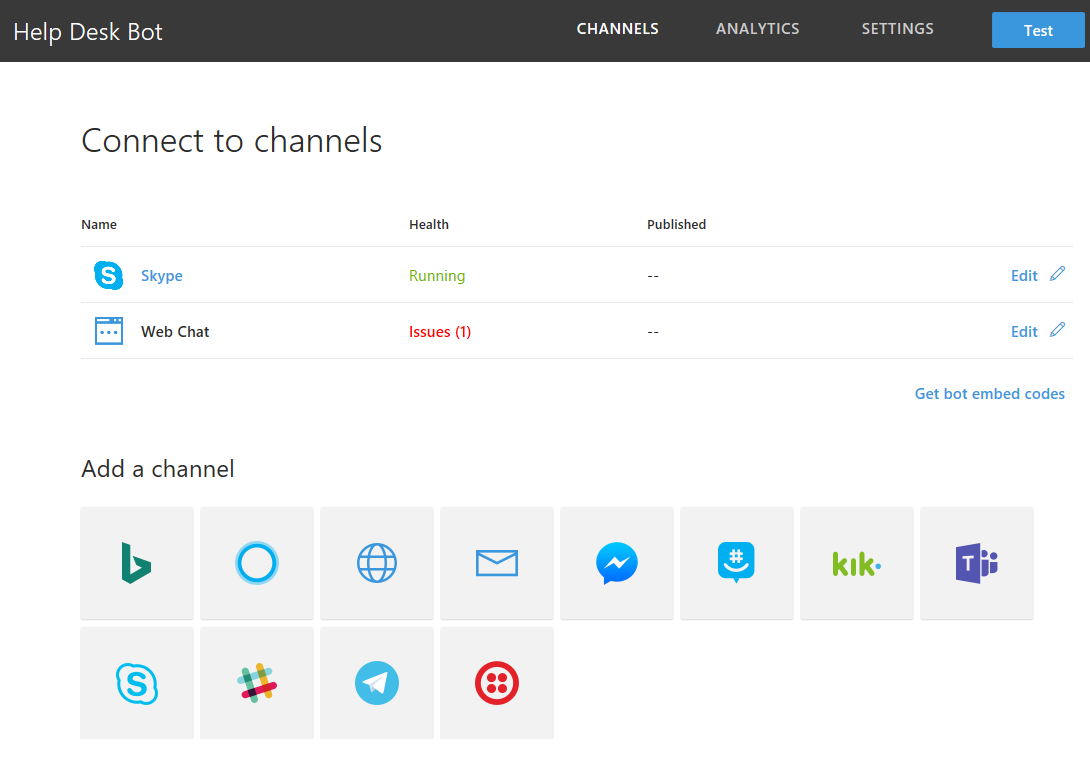
1. [Configuration] セクションで、[Create Microsoft AppID and Password] ボタンをクリックします。それにより、新しいページが開きます。必要に応じて、Bot Framework Portal で使用した資格情報を入力して、もう一度サインインしてください。このページには、先ほど入力したアプリ名と、自動生成された**アプリ ID** が表示されます。**アプリ ID** は後から確認できるように保存しておき、[アプリパスワードを生成して続行] をクリックします。

[](https://github.com/GeekTrainer/help-desk-bot-lab/blob/develop/Node/images/exercise5-generateappid.png)

1. ポップアップが開き、自動生成されたボット用パスワードが表示されます。このパスワードが表示されるのはこの 1 回限りです。後から確認できるように**安全な方法で保管**しておいてください。[OK] をクリックしてポップアップを閉じます。

[](https://github.com/GeekTrainer/help-desk-bot-lab/blob/develop/Node/images/exercise5-generatepassword.png)

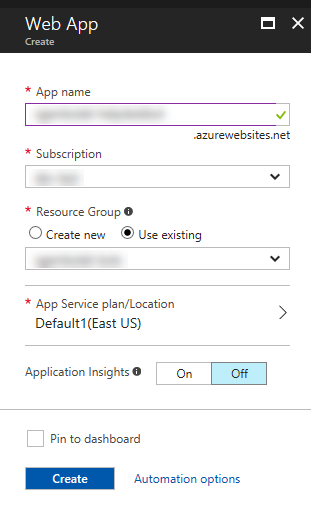
1. [Finish and go back to Bot Framework] ボタンをクリックします。ページが閉じ、ボット登録画面に戻ります。オートコンプリートによって、ページにアプリ ID が入力されています。
2. 画面を下にスクロールして、使用条件、プライバシーに関する声明、および倫理規定への同意について確認します。[Register] ボタンをクリックします。確認メッセージが表示されます。[OK] をクリックします。次に、ボットのダッシュボードが表示されます。いくつかのチャネルが利用可能になっています。

[](https://github.com/GeekTrainer/help-desk-bot-lab/blob/develop/Node/images/exercise5-botchannels.png)

**タスク 2: Azure Web アプリの作成**

このタスクでは、アプリの展開方法と、Bot Framework のチャネルと通信できるようにするためのアプリの構成方法を学習します。

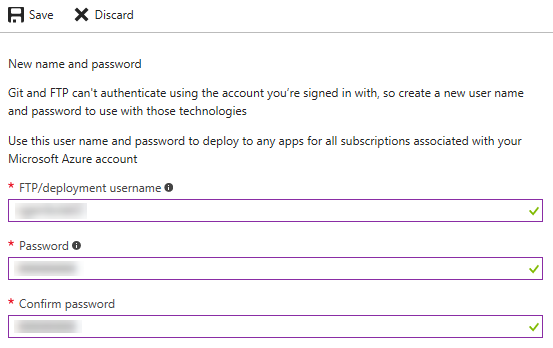
1. [Azure ポータル](https://portal.azure.com/)にサインインします。左側のバーにある [新規] ボタンをクリックし、[Web + モバイル] をクリックして [Web アプリ] をクリックします。
2. [アプリ名] にアプリ名を入力し、[サブスクリプション] でサブスクリプションを、[リソース グループ] でリソース グループをそれぞれ選択します。Cosmos DB および Search サービスで使用したものと同じリソース グループを使用できます。[作成] をクリックします。

[](https://github.com/GeekTrainer/help-desk-bot-lab/blob/develop/Node/images/exercise5-createwebapp.png)

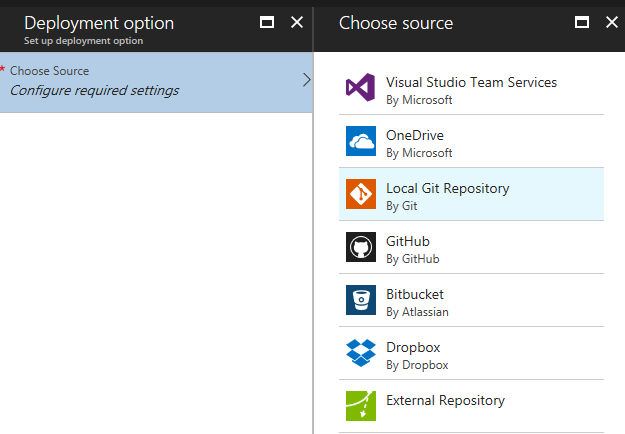
1. 前に作成したアプリ サービスを開きます (まだ開いていない場合)。左側のメニューにある [Application Settings] をクリックします。[App settings] に移動して、以下のキーを追加し、それぞれの値を説明に従って置き換えます (これらの値はボットのソース コードに含まれているはずです)。

| **キー** | **説明** |
| --- | --- |
| AZURE\_SEARCH\_ACCOUNT | **Azure Search アカウント名**を使用します。 |
| AZURE\_SEARCH\_INDEX | **Azure Search インデックス名**を使用します。 |
| AZURE\_SEARCH\_KEY | **Azure Search キー**を使用します。 |
| MICROSOFT\_APP\_ID | **ボット アプリ ID** を使用します。 |
| MICROSOFT\_APP\_PASSWORD | **ボット パスワード**を使用します。 |
| LUIS\_MODEL\_URL | **LUIS モデル URL** を使用します。 |
| TICKET\_SUBMISSION\_URL | アプリ サービスの URL (例: <https://help-desk-bot.azurewebsites.net/>) |

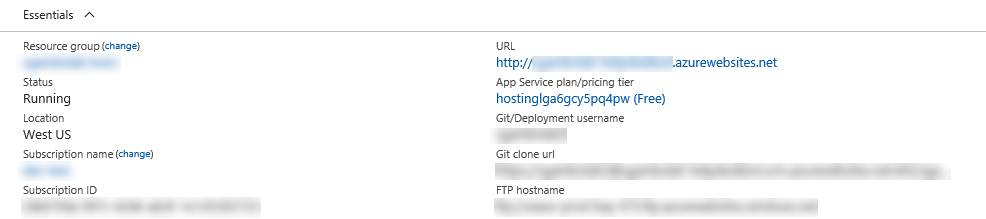
1. キーと値の入力が完了すると、次の図のような結果になるはずです。
2. [](https://github.com/GeekTrainer/help-desk-bot-lab/blob/develop/Node/images/exercise5-addappsettings.png)
3. [Save] ([](https://github.com/GeekTrainer/help-desk-bot-lab/blob/develop/Node/images/exercise5-savebutton.png)) をクリックします。
4. [Deployment] セクションに移動して、左側のバーにある [Deployment credentials] をクリックします。**ユーザー名**と**パスワード**を入力します。パスワードは、[Password] フィールドに入力した後、確認のために [Confirm Password] に再入力します。次に [Save] ボタンをクリックします。

[](https://github.com/GeekTrainer/help-desk-bot-lab/blob/develop/Node/images/exercise5-deploymentcredentials.png)

1. [Deployment] セクションに移動して、左側のメニューにある [Deployment options] をクリックします。次に、[Configure required settings] をクリックし、[Local Git Repository] をクリックします。[OK] をクリックします。

[](https://github.com/GeekTrainer/help-desk-bot-lab/blob/develop/Node/images/exercise5-deploymentoption-localgit.png)

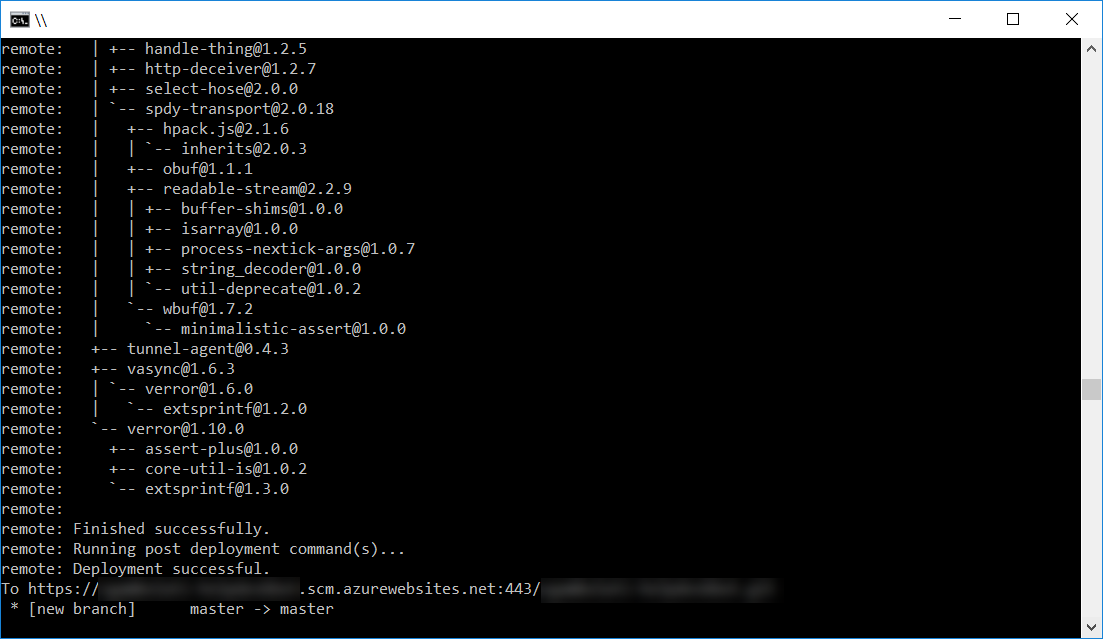
1. 左側のバーにある [Overview] オプションをクリックします。後で使用できるように、[Essentials] ウィンドウの右側の列にある [Git clone url] の URL を保存しておきます。

[](https://github.com/GeekTrainer/help-desk-bot-lab/blob/develop/Node/images/exercise5-essentials-gitclone.png)

**タスク 3: Azure へのボットの発行**

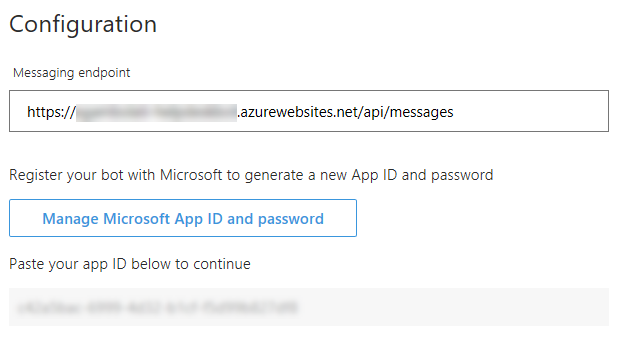
このタスクでは、コードを Git リポジトリに発行します。それにより、最近の変更内容が Azure App Service に展開されます。Git の詳細については、[こちら](https://git-scm.com/docs)をクリックして Git のリファレンスを参照してください。

1. 前の演習で得られたアプリ フォルダーでコンソールを開きます。または、[exercise4-KnowledgeBase](https://github.com/GeekTrainer/help-desk-bot-lab/blob/develop/Node/exercise4-KnowledgeBase) フォルダーにあるアプリを使用することもできます。このコード用の Git リポジトリを作成していない場合は、コンソールで次のコマンドを入力します (このコードを Git リポジトリで既に初期化している場合は、このステップを行う必要はありません)。
2. git init
3. 次に、次のコマンドを入力します。{gitcloneurl} の部分は、前のタスクで得られた **Git clone url** の URL に置き換えてください。
4. git remote add azure {gitcloneurl}
5. 以下のコマンドを入力して、変更内容をコミットし、Azure Web アプリにプッシュします。
6. git add .
7. git rm --cached node\_modules -r
8. git commit -m "initial commit"
9. git push azure master
10. 前のタスクで作成した展開の資格情報を入力します。
11. 次に表示される画面で、ファイルがアップロードされ、リモート サーバー上に npm パッケージがインストールされている様子を確認できます。最後に展開状態が表示されます。

[](https://github.com/GeekTrainer/help-desk-bot-lab/blob/develop/Node/images/exercise5-deploymentsuccessful.png)

**タスク 4: ボット構成の更新**

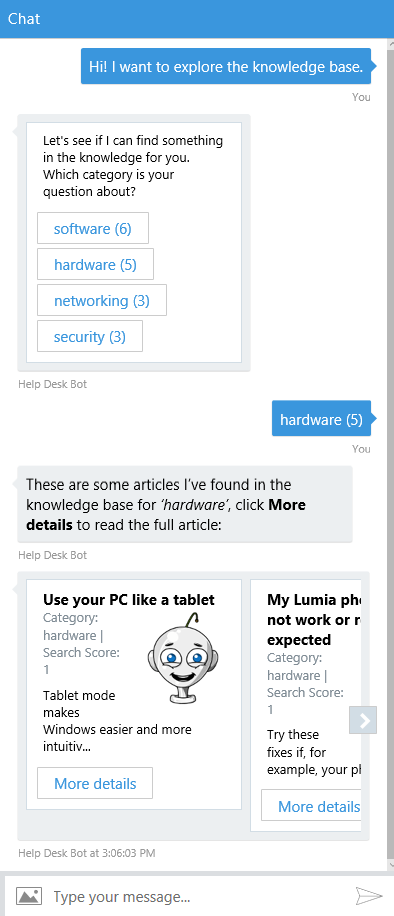
1. [Bot Framework Portal](https://dev.botframework.com/) に移動します。ボットを編集するために、ボット名をクリックします。
2. ページの右上隅にある [Settings] ボタンをクリックします。
3. [Configuration] セクションで、タスク 2 で作成したアプリ サービス URL (例: <https://help-desk-bot.azurewebsites.net/api/messages>) を入力します。URL の末尾には必ず /api/messages を付け、またプロトコルは必ず **https** にしてください。ページ下部の [Save changes] ボタンをクリックします。

[](https://github.com/GeekTrainer/help-desk-bot-lab/blob/develop/Node/images/exercise5-botconfiguration.png)

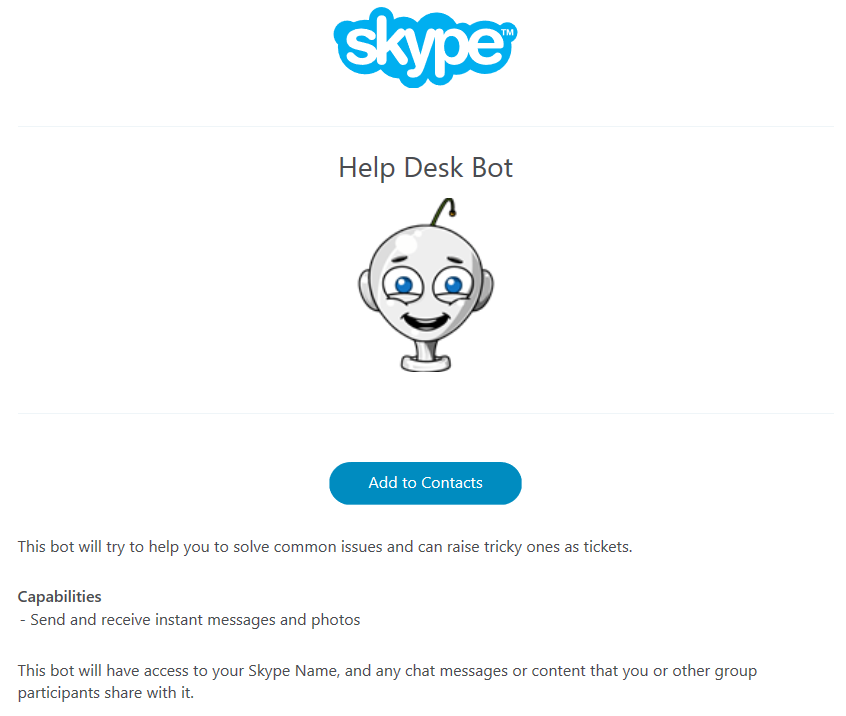
**タスク 5: 発行したボットのテスト**

このタスクでは、他のチャネルからボットをテストします。

1. [Bot Framework Portal](https://dev.botframework.com/) に移動し、ボットを編集します (まだ、ポータル上の編集画面に移動していない場合)。
2. ページの右上隅にある [Test] ボタンをクリックします。ページの右側に新しいウィンドウが開きます。これは、ボットを容易にテストできるようにする埋め込みの Web チャット チャネルです。
3. 「Hi! I want to explore the knowledge base.」と入力し、この入力への応答としてボットがカテゴリ一覧を返すことを確認します。任意のカテゴリをクリックして、そのカテゴリの記事一覧が表示されることを確認し、いずれかの記事をクリックして内容を確認します。

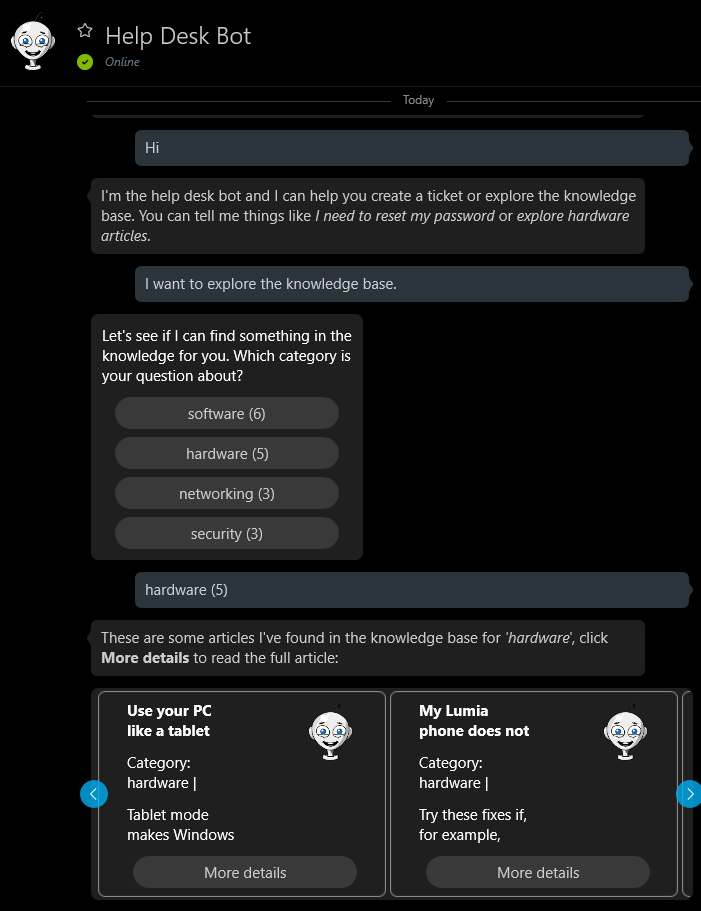
[](https://github.com/GeekTrainer/help-desk-bot-lab/blob/develop/Node/images/exercise5-testwebchannel.png)

1. [Channels] メニュー項目をクリックします。**Skype** チャネルと **Web チャット** チャネルが既定で有効になっているはずです。[Skype] リンクをクリックします。新しいページが開きます。このページで、**Skype** アカウントにボットを追加できます。[Add to Contacts] ボタンをクリックします。Skype アカウントにサインインするように求められ、Skype アプリが開かれるはずです。

[](https://github.com/GeekTrainer/help-desk-bot-lab/blob/develop/Node/images/exercise5-addbottoskype.png)

**注:** [Get bot embed codes] リンクについても確認してみてください。ここでは、ユーザーが自分の Skype アカウントにボットを追加できるようにするためのリンクの構築方法がわかります。

1. 連絡先リストでボットを検索して、新しい会話をテストします。

[](https://github.com/GeekTrainer/help-desk-bot-lab/blob/develop/Node/images/exercise5-testskype.png)

**注:** このハンズオン ラボの作成時点で、Skype はアダプティブ カードを完全にサポートしてはいないため、チケット確認メッセージが正しく表示されない可能性があります。

**その他の課題**

* Bot Emulator でボットをテストします。[ngrok](https://docs.microsoft.com/en-us/bot-framework/debug-bots-emulator#a-idngroka-install-and-configure-ngrok) を使用して、ローカル コンピューターへの応答方法をボットに知らせる必要があります。
* Skype または Web チャットを介したボットのテスト中、[コードをローカルで実行します](https://blogs.msdn.microsoft.com/jamiedalton/2016/07/29/ms-bot-framework-ngrok/)。
* [Application Insights を使用した Bot Analytics](https://docs.microsoft.com/en-us/bot-framework/portal-analytics-overview) の追加も試してみてください。
* [Slack](https://slack.com/) などの別のチャネルにボットを登録します。