

1. Creating Resources for Language Service

For creating a Custom Question Answering bot using Azure Cognitive Service Language Studio, we have created a new resource under the “**Language Services**” section. We choose the Subscription, Resource Group, Name, region, and Pricing Tier for the new resource and click on Create to set up a language resource for the project.

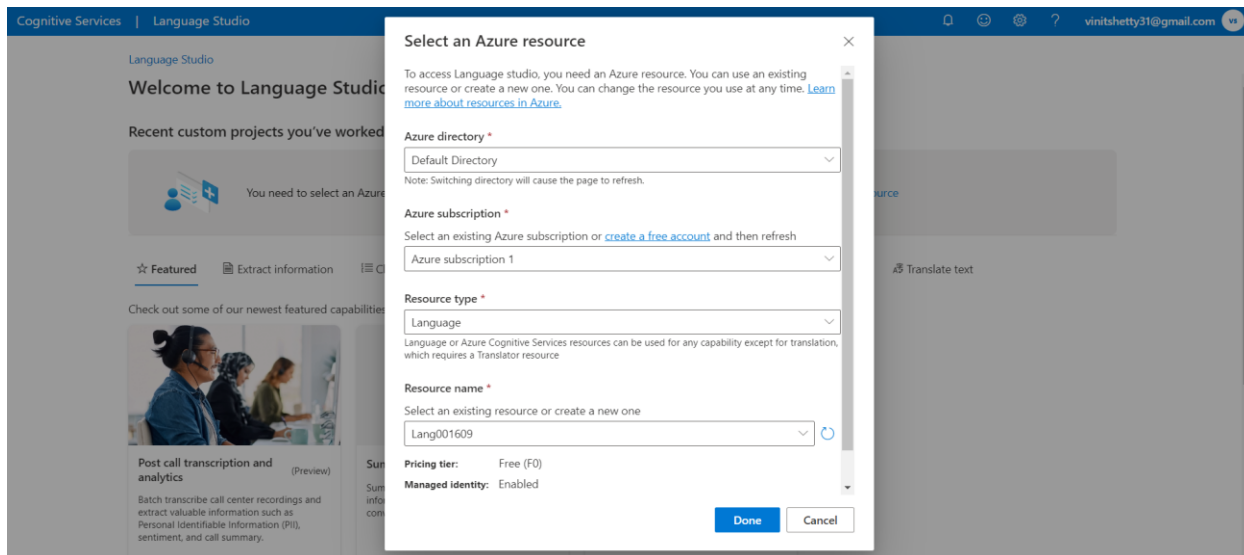
We can get the **API Keys**, **Endpoint URL** which can be used later to create the project in Language Studio and create the Bot.

The screenshot shows the 'Create Language' form in the Microsoft Azure portal. The form is divided into several sections: 'Project Details', 'Instance Details', and 'Custom question answering'. In the 'Project Details' section, the 'Subscription' is set to 'Azure subscription 1' and the 'Resource group' is '(New) Vinit_resource'. In the 'Instance Details' section, the 'Region' is 'Australia East', the 'Name' is 'Lang001609', and the 'Pricing tier' is 'Free F0 (5K Transactions per 30 days)'. The 'Custom question answering' section has the 'Azure search region' set to 'Australia East'. At the bottom, there are buttons for 'Review + create', '< Previous', and 'Next: Network >'. A 'Give feedback' link is also present.

The screenshot shows the 'Overview' page for the 'Lang001609' Language resource in the Azure portal. The page includes a left-hand navigation menu with options like 'Overview', 'Activity log', 'Access control (IAM)', 'Tags', 'Diagnose and solve problems', 'Resource Management', 'Features', 'Keys and Endpoint', 'Encryption', 'Pricing tier', 'Networking', 'Identity', 'Cost analysis', 'Properties', 'Locks', 'Monitoring', 'Alerts', 'Metrics', 'Diagnostic settings', 'Logs', and 'Automation'. The main content area displays the resource's 'Essentials' and 'Get Started' information. The 'Essentials' section lists the resource group, status, location, subscription, subscription ID, and tags. The 'Get Started' section includes a 'Discover' button and a 'Deploy' button. A 'JSON View' link is also available.

2. Selecting an Azure Resource in Language Studio

Once a Language Resource is generated in the Azure Portal, we proceed to access the language studio and choose the specific resource that was created earlier, namely "**Lang0011609**". Then, we navigate to the "**Understanding Questions and Conversational Language**" section and opt for "**Open Custom question answering**" to initiate the project creation process.



3. Creating a project

Once we have chosen the desired resource in the language studio, we proceed to click on the "**Create New Project**" option. Then, we select the preference of selecting the same language for all the projects within the resource and choose "**English**". Subsequently, we provide a suitable name for the project, such as "**Chatbot007**", furnish a brief description, and set the default answer as "**No Answer Found**". Finally, we click on the "**Create Project**" button to create the project.

Create a project

Choose language setting

Enter basic information

Review and finish

Enter basic information

Enter the basic information for your custom question answering knowledge base such as name and description.

Lang0011609

To change your resource go to [Settings](#)

Azure search resource

lang0011609-asvis3vfnea3y2g

To change your resource go to [Azure Search](#)

Name *

Chatbot007

Description

Chatbot for QNA for Delivery app

Source language * ⓘ

English

Default answer when no answer is returned * ⓘ

No answer found

Back

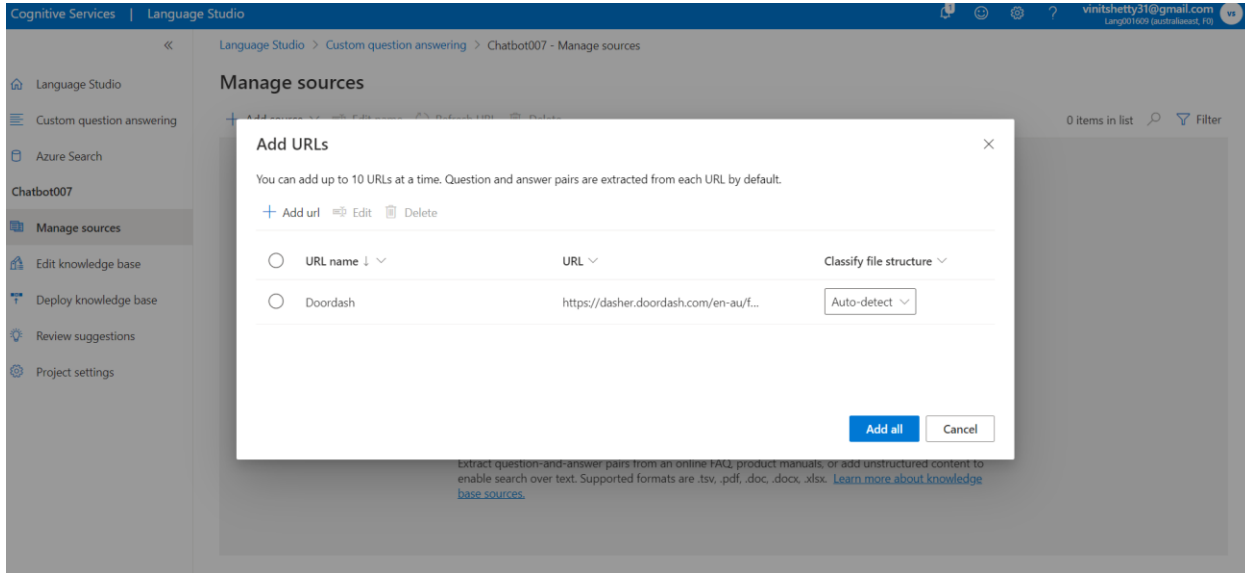
Next

Create project

Cancel

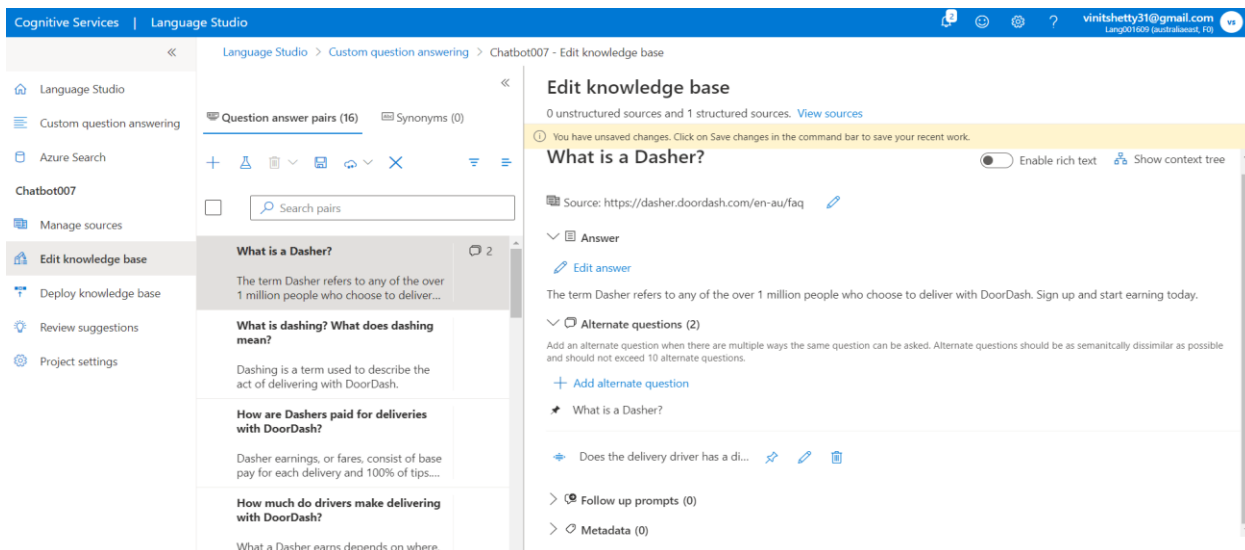
4. Add a source from the chosen URL.

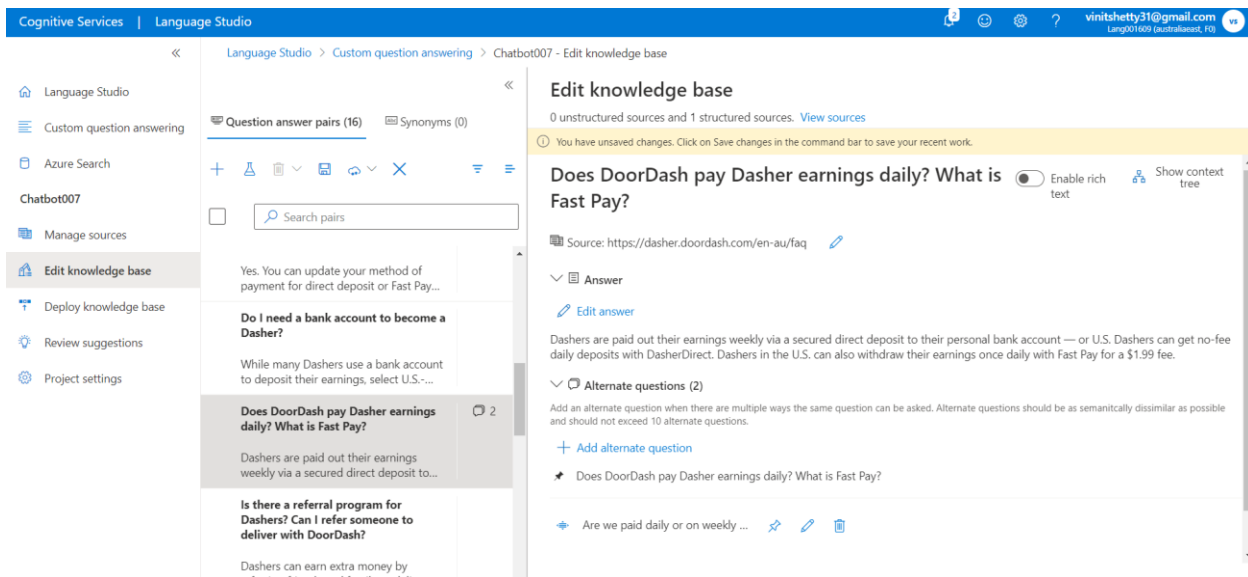
To add a fresh source, we need to access the **"Manage Sources"** tab and then click on the **"Add Source"** option, following which we select **"Add URLs"**. For the current project that centres around **"Food Delivery Services"**, we have chosen to utilize information from the **Doordash FAQs**. This website provides answers to some of the frequently asked questions by delivery partners, and its URL is <https://dasher.doordash.com/en-au/faq>



5. Editing the knowledge base with alternative questions.

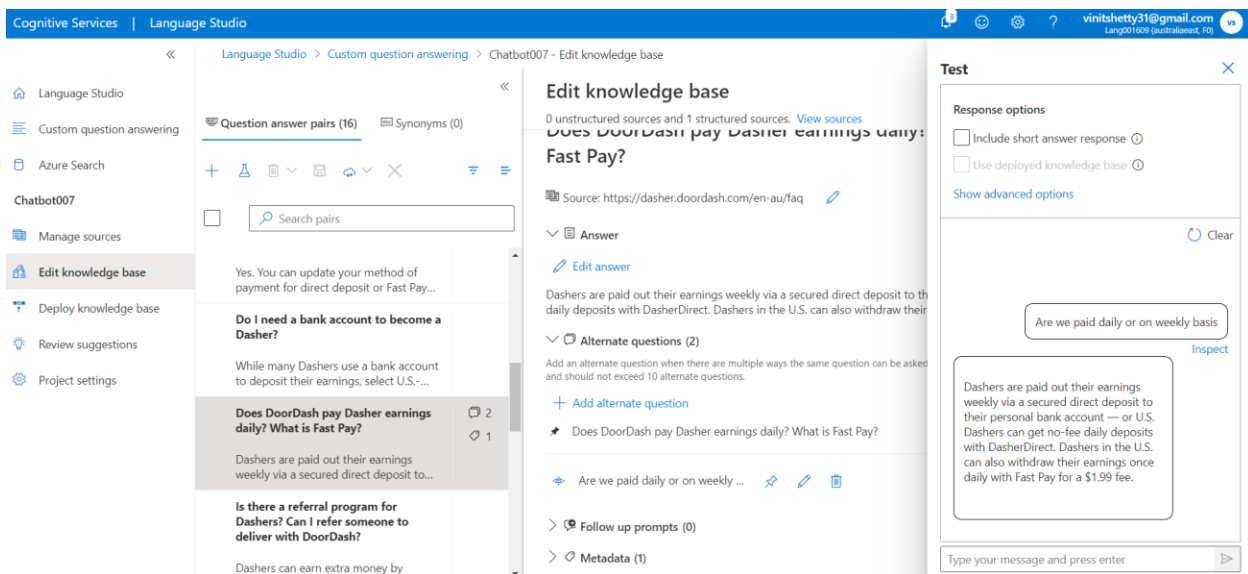
After we provide the source URL, it takes some time for the knowledge base to load. Once loaded, we can view suggested question-answer pairs that can be used as the bot's knowledge base. We also have the option to add more questions or alternate questions to improve the bot's understanding of user queries. For instance, an alternate question **"Are we paid daily or on weekly basis"** was included to enhance the original query **"Does Doordash Pay Dasher earnings daily? What is Fast Pay?"** The inclusion of alternate questions helps to enhance the knowledge base, enabling the bot to provide more accurate answers.





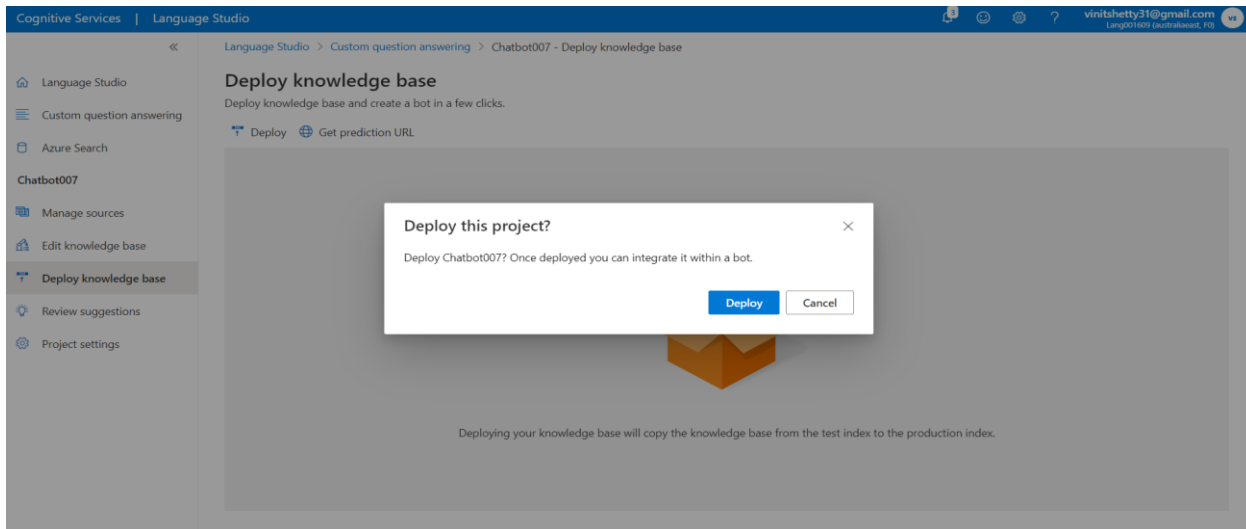
6. Testing the Project

Once we have updated the knowledge base with alternate questions, we can proceed to test the accuracy of the knowledge base in providing answers. To do this, we select the **"Test"** option from the menu in the edit knowledge base tab, which opens a pop-up window. In this window, we can ask questions, such as **"Are we paid daily or on weekly basis,"** which was an **alternate question** added in the previous tab. Based on the established knowledge base, an appropriate answer is generated to the question asked.



7. Deploying the Project

After testing the project's accuracy, the subsequent step is to deploy it by selecting the **"Deploy knowledge base"** tab and clicking the **"Deploy"** option. We must then confirm the deployment by clicking the **"Deploy"** option again. Once the project has been deployed, we can proceed with the creation of the **QnA bot**.



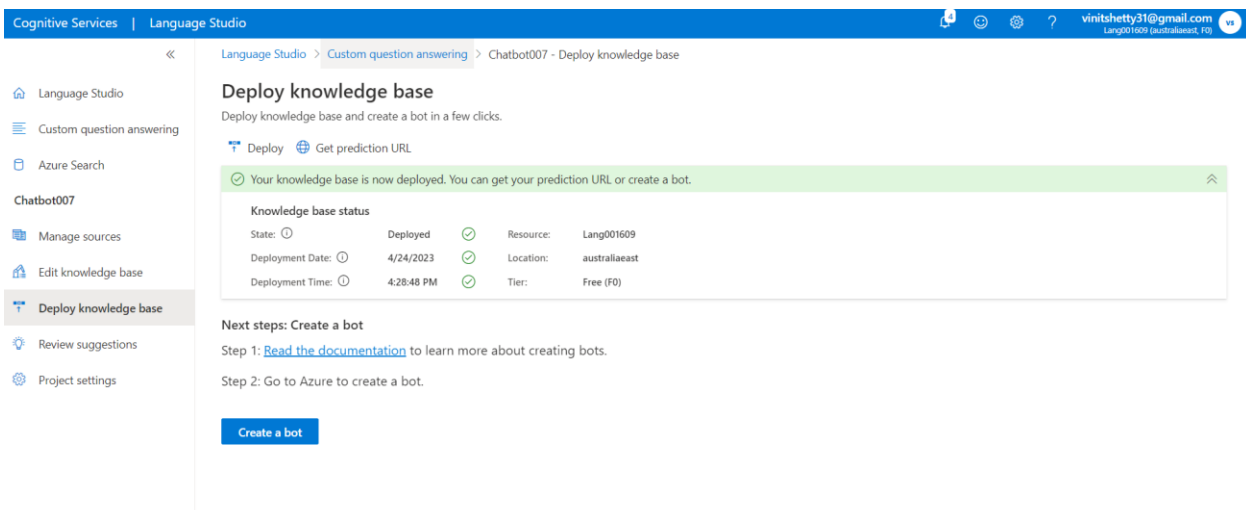
8. Creating a bot

After deploying the project, we click on **"Create a bot"** which opens another browser tab in the Azure portal to create and deploy the bot.

For creating the bot, we need to **configure** some values which are as below:

Setting	Value
Bot handle	Lang0011609-bot
Subscription	Default Value
Resource group	Default Value
Location	Australia East
Pricing tier	Free
App name	Default Value
SDK language	Node.js. (JSON file)
Language Resource Key	Key obtained from the Language resource created
App service plan/Location	Default Value

Upon configuring the necessary values, we click on **"Review+Create"** If the validation process is successful, the bot **"Lang0011609-bot"** is then created.



Microsoft Azure

Home > Custom deployment

Deploy from a custom template

your cloud environment being compromised.

Customized template of 4 resources

Edit template Edit parameters Visualize

Project details

Select the subscription to manage deployed resources and costs. Use resource groups like folders to organize and manage all your resources.

Subscription * Azure subscription 1

Resource group * Vinit_resource

Instance details

Resource group location * (Asia Pacific) Australia East

Azure Bot

Bot handle * Lang001609-bot

Choose your pricing tier

Select a pricing tier for your Azure Bot resource. You can change your selection later in the Azure portal's resource management. Learn more about available options, or request a pricing quote, by visiting the Azure Bot Services pricing

Pricing tier Free

Change plan

Microsoft App ID

Add user-assigned identities to grant the resource access to Azure Bot resource.

Review + create Previous Next: Web App >

Microsoft Azure

Home > Custom deployment

Deploy from a custom template

Basics Web App Review + create

App Service

App name * Lang001609-bot-7ab0

SDK language selection * Node.js

App Service Plan

App Service plan pricing tier determines the location, features, cost and compute resources associated with your app. Learn more

Creation type

Create new app service plan

Use existing app service plan

The pricing tier currently defaults to 'Standard'. It can be modified by visiting the app service plan resource page once the resource has been created, or you can choose an existing plan in your subscription.

App Settings

In App Service, these app settings are variables passed as environment variables to the bot code.

Language Resource Key *

Language project name Chatbot007

Language service endpoint hostname https://Lang001609.cognitiveservices.azure.com

Review + create Previous Next: Review + create >

9. Connecting the bot to the Microsoft Teams channel

Once the bot is successfully created, we can connect it to various channels by choosing the "Channels" tab within the create bot resource. In our project, we have integrated the bot with the "Microsoft Teams" channel, allowing us to directly ask the bot **Doordash Delivery agent-related queries** via Microsoft Teams.

Microsoft Azure

Home > Microsoft.Template-20230424163132 | Overview > Vinit_resource > Lang001609-bot

Lang001609-bot | Channels

Overview

Activity log

Access control (IAM)

Tags

Settings

Bot profile

Configuration

Channels

Pricing

Test in Web Chat

Encryption

Networking

Properties

Locks

Monitoring

Conversational analytics

Alerts

Metrics

Diagnostic settings

Logs

Web Chat

Healthy

Embeddable Web Chat control

Available Channels

Connect the bot with channels. Learn more

Channel	Details
Alexa	Alexa Channel
Communication Services - Chat	Communication Services - Chat Channel
Direct Line Speech	Direct Line Speech Channel
Email	Office 365 Email Channel
Facebook	Support for Text Messaging via Facebook
GroupMe	GroupMe Channel
LINE	Support for LINE Channel
Microsoft 365 Extensions	Enable message extensions in Outlook, and Microsoft 365 apps
Microsoft Teams	Microsoft Teams Channel
OmniChannel	OmniChannel Channel

10. Asking questions to bot on Microsoft Teams

Once the bot is successfully integrated into Microsoft Teams, we can begin asking it domain-specific questions. For instance, we asked the bot the question "**Do I get paid on a daily or weekly basis?**" to which the bot responded with "**Weekly**" and provided a brief explanation. Additionally, we posed a question to the bot that was **not present** in the knowledge base, and the bot returned the default response of "**No Answers found.**"

