

A blue cartoon robot with a smiling face, holding a sign that says 'Canva'. The robot is standing on a dark blue background with a pattern of white dots. There are also white speech bubbles around the robot.

Azure AI Document Intelligence automates receipt analysis by extracting data quickly and accurately using advanced AI technology. This presentation covers setting up Azure AI Document Intelligence, creating and deploying an Azure Function, and using this function to enhance the efficiency of receipt analysis.

A diagram illustrating the sequence of steps for setting up an Azure environment. The steps are arranged in a circular flow, each represented by a light blue rounded rectangle with a corresponding icon to its right:

- Azure Subscription**: Icon shows a blue cube with a magnifying glass and a network diagram.
- API Platform (Postman)**: Icon shows a computer monitor displaying 'API' and a red gear.
- Azure CLI Installation**: Icon shows a cardboard box with a blue checkmark and the word 'Canva'.
- Visual Studio Code Setup**: Icon shows a laptop displaying a code editor with a purple background and '</>' symbols.
- Access to Azure Portal**: Icon shows a blue hexagon with a yellow padlock and the word 'Canva'.
- Microsoft Account**: Icon shows a white person silhouette with a yellow checkmark.

Arrows indicate the flow from one step to the next in a clockwise direction.

Subscription: Your Azure subscription.

Resource group: Select or create a resource group with a unique name.

Region: Select your region.

Name: Enter a globally unique name.

Pricing tier: Select "Free F0" (if you don't have a Free tier available, select "Standard S0").

Navigate to the Azure Portal by visiting portal.azure.com and log in with your credentials to access the dashboard.

In the Azure Portal, use the search bar to find 'Azure AI Document Intelligence' and select it from the results.

Click on the 'Create' button to start the resource creation process.

Click on the 'Create' button to start the resource creation process.

Choose the appropriate subscription and resource group for your new resource to organize and manage resources efficiently.

Enter a unique name for your resource and select a preferred region that suits your operational needs.

A cute, white, rounded robot with a large head, glowing blue eyes, and a blue light band around its chest. It is standing on a blue surface against a dark blue background.

Install!

```
dotnet add package  
Azure.AI.FormRecognizer
```

1

- Azure Functions: Create Function...
- Azure Functions: Deploy to Function App...
- .NET: New Project...
- .NET: Install New .NET SDK

2

← Create new

Select a template for your project's first function

HTTP trigger (recently used)

⌚ Skip for now

Timer trigger

3

AccessRights

Function (recently used)

Anonymous

Admin

4

← Create new project (3/4)

Select a .NET runtime

- .NET 8.0 Isolated LTS (recently used)
- .NET 6.0 LTS
- .NET 6.0 Isolated LTS

Important !

```
1 localsettings.json > ...  
2 {  
3   "IsEncrypted": false,  
4   "Values": {  
5     "AzureWebJobsStorage": "",  
6     "FUNCTIONS_WORKER_RUNTIME": "dotnet-isolated",  
7     "FORM_RECOGNIZER_ENDPOINT": "  
8     "FORM_RECOGNIZER_API_KEY": "  
9   },  
10   "Host": {  
11     "CORS": "http://localhost:3000"  
12   }  
}
```

- ✓ API Key
- ✓ Endpoint

Need to follow your own API Key and Endpoint

KEY 1

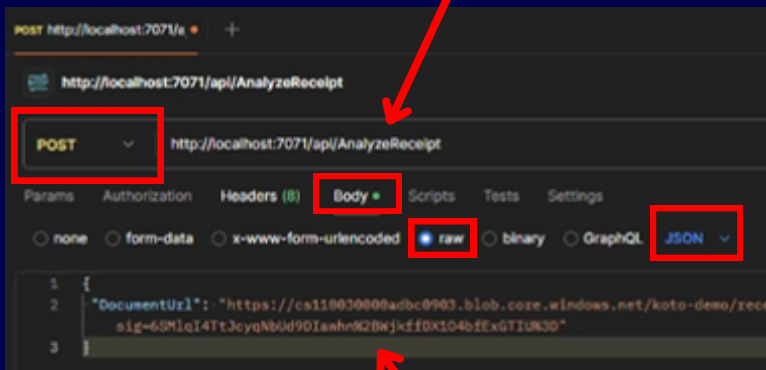
KEY 2

Endpoint

<https://kotodocintrss.cognitiveservices.azure.com/>

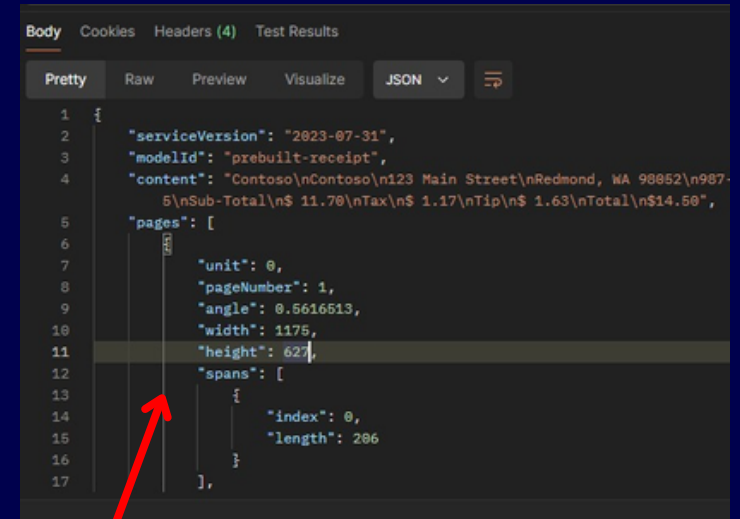
3 Steps to Test the Function (Postman)

Copy the API from the Visual Studio Code and paste here



Send a sample receipt using Postman. Make sure the receipt data follows the correct format.

Make sure you follow each steps shown and click send button



The result will be displayed like this

1 SetUp Next.js (Front-End)

Initialize a Next.js project with TypeScript

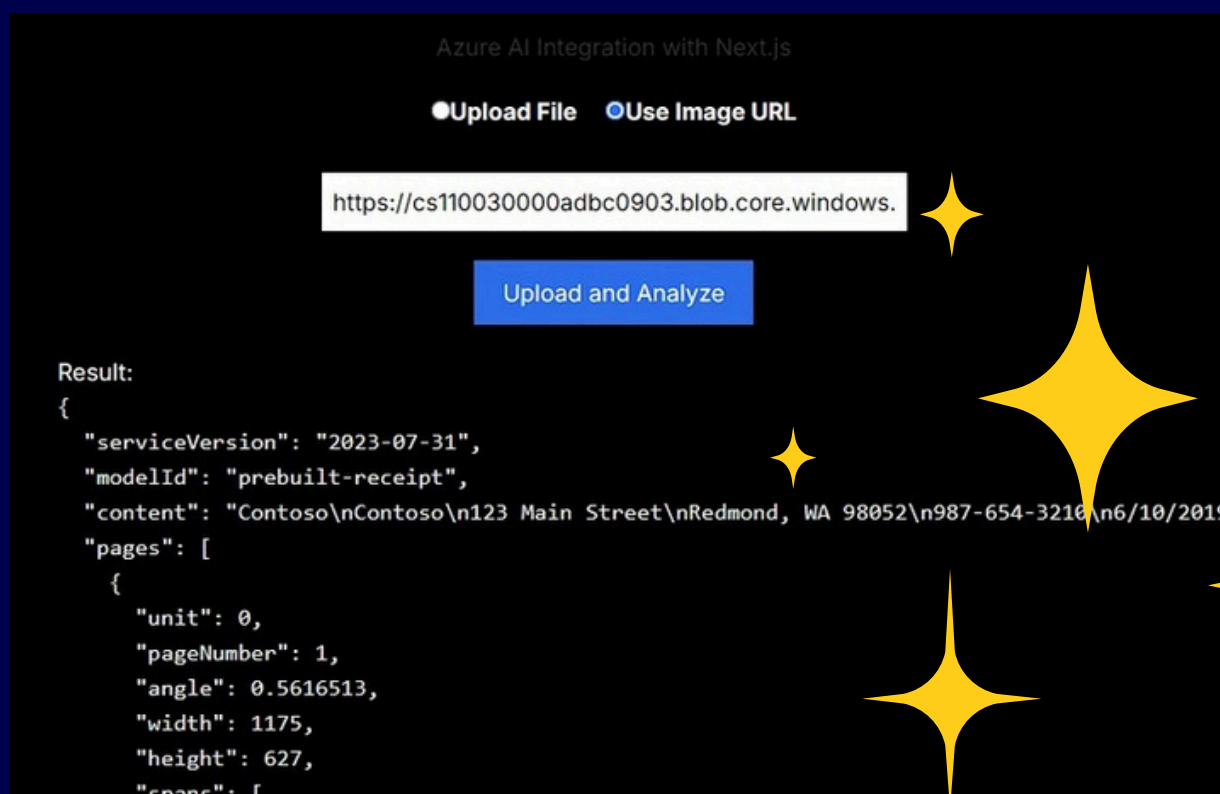
```
npx create-next-app@latest my-next-app <name_of_your_project>
```

Start the development server: npm run dev.

2 Create A Page to Fetch the API

```
// Set the URL for the Azure Function endpoint
setFetchUrl('http://localhost:7071/api/AnalyzeReceipt');
```

Test if the Azure Function is integrated on Next.js Page



Ensure both the Azure Function and the Next.js page are running concurrently. The result should be displayed on the page.

Congratulations! You have successfully integrated Azure AI with your Next.js application.

You can now analyze receipts using image URLs directly from your web interface.