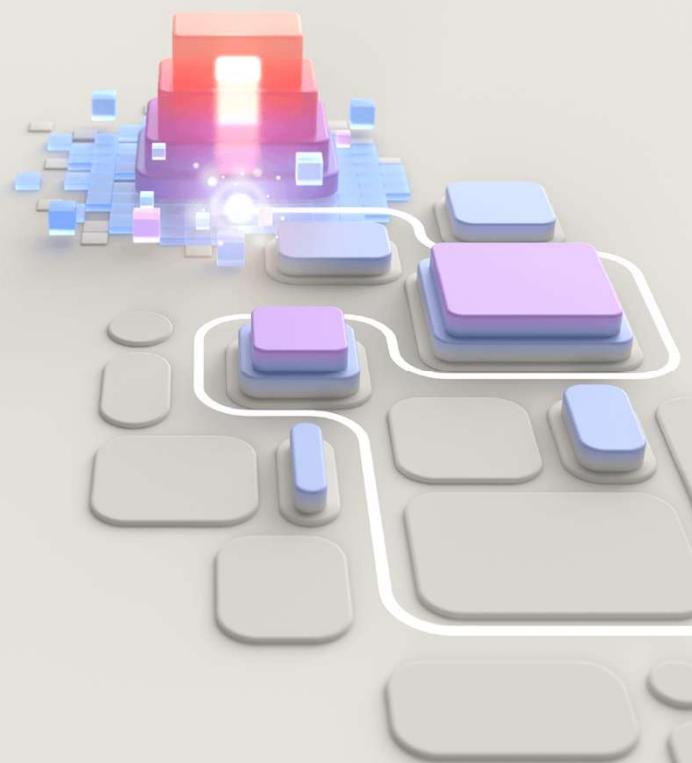




Develop solutions with Azure AI Document Intelligence



© Copyright Microsoft Corporation. All rights reserved.

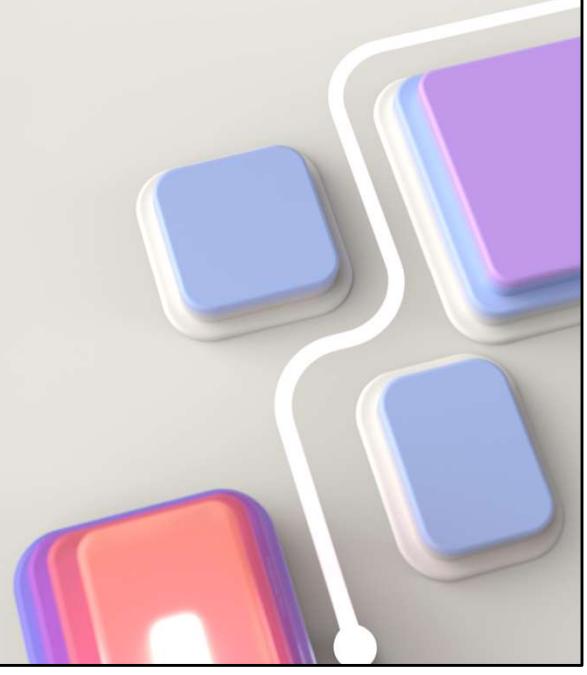
Agenda

- Use prebuilt Document Intelligence models
- Train a custom Document Intelligence model

© Copyright Microsoft Corporation. All rights reserved.

Develop a Document Intelligence solution

© Copyright Microsoft Corporation. All rights reserved.



Learning Objectives

After completing this module, you will be able to:

- 1** Understand models in Azure AI Document Intelligence
- 2** Train a custom Document Intelligence model
- 3** Connect an app to Document Intelligence APIs

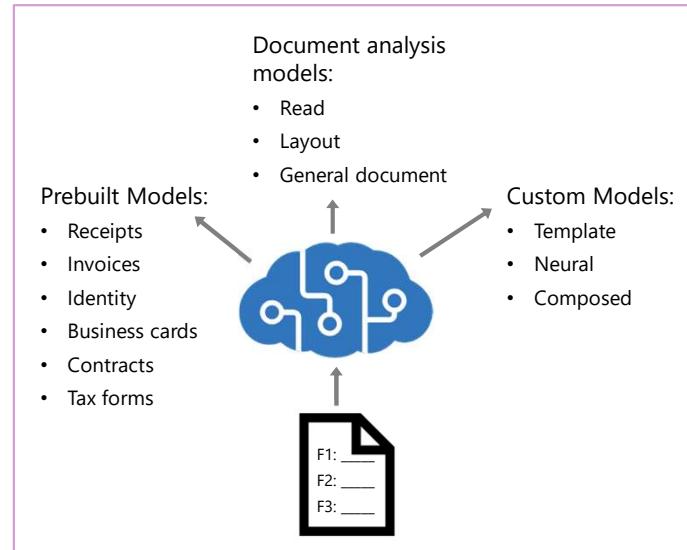
© Copyright Microsoft Corporation. All rights reserved.

The Document Intelligence Service

Data extraction from forms and documents:

- Document analysis from general documents
 - Read: OCR for printed and written text
 - Layout: Extract text and structure
 - General document: Extract text, structure, and key-value pairs
- Prebuilt models for common form types
- Train custom models for your own forms
 - Custom template: Extract data from static layouts
 - Custom neural: Extract data from mixed-type documents
 - Custom composed: Collection of multiple models assigned to a single model

Provision as single-service **Document Intelligence** resource or multi-service **Azure AI Services** resource



© Copyright Microsoft Corporation. All rights reserved.

Azure Document Intelligence Service – Översikt

Azure Document Intelligence Service (tidigare Form Recognizer) är en AI-baserad tjänst för att extrahera data från dokument och formulär. Den kan läsa, tolka och strukturera information från olika dokumenttyper med hjälp av förbyggda och anpassade modeller.

Funktioner och Modelltyper

1. Dokumentanalys för allmänna dokument

1. **Read** – OCR-baserad textigenkänning för tryckt och handskriven text.
2. **Layout** – Extraherar både text och **dokumentstruktur** (t.ex. tabeller och kolumner).
3. **General document** – Identifierar text, struktur och **nyckel-värde-par** från ostrukturerade dokument.

2. Förbyggda modeller (Prebuilt Models)

1. Redo att användas för vanliga dokumenttyper som:
 1. Kvittot, fakturor, ID-handlingar, visitkort, kontrakt och skatteblanketter.
 2. Perfekt för att snabbt extrahera information utan att behöva träna en egen modell.

3. Träna egna modeller (Custom Models)

1. **Template Model** – Fungerar bäst för dokument med **statiska layouter** (t.ex. fasta fakturaformat).
2. **Neural Model** – Anpassad AI-modell som kan hantera **blandade dokumenttyper**.
3. **Composed Model** – En kombination av flera modeller, där tjänsten väljer rätt modell baserat på dokumentet.

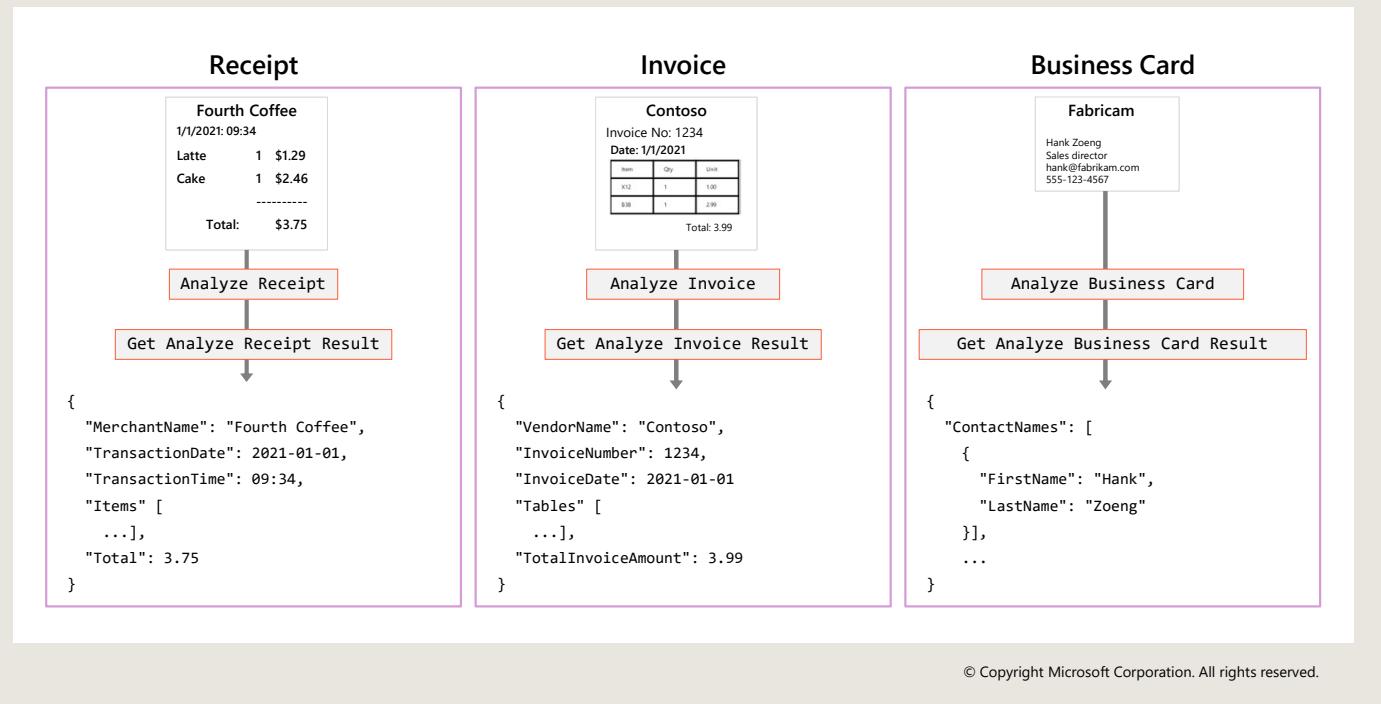
Användning och Provisionering

Tjänsten kan provisioneras som en **fristående Document Intelligence-resurs** eller som en del av en **Azure AI Services-resurs** för bredare AI-funktioner.

Sammanfattning

Azure Document Intelligence är en **kraftfull AI-lösning för dokumentbearbetning**. Med stöd för **OCR, strukturell dataextraktion och anpassningsbara modeller** kan företag automatisera hantering av fakturor, kontrakt och andra dokument, vilket sparar både tid och resurser.

Prebuilt models



These are just examples, further prebuilt models are available on the documentation pages.

Calling the API

- Each request is configured with your resource endpoint and needs your resource key
- Send the request, which when successful returns a poller to get the results
 - REST returns it in Operation-Location header
 - SDKs return an object from the request
- Query the poller received for the extracted data

REST

```
Request POST:  
{endpoint}/documentintelligence/documentModels/prebuilt-  
layout:analyze?api-version={version}
```

Operation-Location:

```
{endpoint}/documentintelligence/documentModels/prebuilt-  
layout/analyzeResults/ab12345c-12ab-23cd-b19c-  
2322a7f11034?api-version={version}
```

C#

```
AnalyzeDocumentOperation operation = await  
client.AnalyzeDocumentFromUriAsync(WaitUntil.Completed,  
"prebuilt-layout", fileUri);  
  
AnalyzeResult result = operation.Value;
```

Python

```
poller=document_analysis_client.begin_analyze_document_  
from_url("prebuilt-document", docUrl)  
  
result = poller.result()
```

© Copyright Microsoft Corporation. All rights reserved.

Calling the API – Översikt

Den här bilden beskriver hur man anropar Azure Document Intelligence API för att analysera dokument med REST, C# och Python.

Hur fungerar API-anropet?

1.Konfigurera anropet

1. Ange din **resource endpoint** och **resource key** i API-anropet.

2.Skicka begäran (Request)

1. Använd REST eller SDK för att analysera ett dokument.
2. Vid lyckat anrop returneras en **poller**, som används för att hämta resultatet.

3.Hämta resultatet

1. REST API → Returnerar **Operation-Location**-header med en URL där analysresultatet hämtas.
2. SDKs (C#, Python, etc.) → Returnerar direkt ett objekt som kan hämtas med .result().

Kodexempel för API-anrop

REST API (POST Request)

- Skicka en POST-request till: {endpoint}/documentintelligence/documentModels/prebuilt-layout:analyze?api-version={version}

• Resultatet kan hämtas via **Operation-Location**, t.ex.:

{endpoint}/documentintelligence/documentModels/prebuilt-layout/analyzeResults/{operationId}?api-version={version}

C# API-anrop

- Använd AnalyzeDocumentFromUriAsync för att analysera dokumentet: AnalyzeDocumentOperation operation = await client.AnalyzeDocumentFromUriAsync(WaitUntil.Completed, "prebuilt-layout", fileUri); AnalyzeResult result = operation.Value;

Python API-anrop

- Anropa begin_analyze_document_from_url och hämta resultatet: poller = document_analysis_client.begin_analyze_document_from_url("prebuilt-document", docUrl) result = poller.result()

Sammanfattning

- API:et använder **asynkrona anrop** där ett **poller-objekt** returneras för att hämta resultatet.
- **REST API** kräver att du manuellt hämtar resultatet via en URL.
- **SDKs (C#, Python, etc.)** förenklar processen genom att hantera anropet och resultatet mer automatiskt. Detta är ett effektivt sätt att **automatisera dokumentanalys** och extrahera data från formulär, fakturor, ID-handlingar och andra dokumenttyper. 

API response

- Response is broken down by page, lines, and words
- Subset of REST response included here
- SDK response objects have similar structure, broken down similarly
- Additional data about detected text or selection marks, such as bounding box and handwritten style

```
{
  "analyzeResult": {
    "apiVersion": "{version}",
    "modelId": "prebuilt-invoice",
    ...
    "pages": [
      {
        "pageNumber": 1,
        "angle": 0,
        "width": 8.5,
        "height": 11,
        "unit": "inch",
        "words": [
          {
            "content": "Margie's",
            "boundingBox": [
              0.5911,
              0.6857,
              1.7451,
              0.6857,
              1.7451,
              ...
            ],
            "confidence": 1,
            "span": {...}
          }
        ]
      }
    ]
}
```

© Copyright Microsoft Corporation. All rights reserved.

API Response – Översikt

Den här bilden visar ett exempel på hur ett API-svar från Azure Document Intelligence är strukturerat efter att ett dokument analyserats.

Hur är API-svaret uppbyggt?

1. Svaret är uppdelat i sidor, rader och ord

1. API:et analyserar dokument per sida och identifierar text, position och metadata.

2. REST och SDK-svar ser likadana ut

1. JSON-strukturen är densamma oavsett om du använder REST API eller SDKs (C#, Python etc.).

3. Ytterligare information om text

1. Varje ord har metadata som:
 - ✓ **Bounding box** (koordinater för var texten finns på sidan).
 - ✓ **Handskriven stil** (om texten är skriven för hand).
 - ✓ **Confidence score** (säkerhet i tolkningen).

Vad betyder de olika fälten?

- **modelId** → Anger vilken modell som användes (t.ex. *prebuilt-invoice* för fakturor).
- **pages** → Lista med sidor i dokumentet.
- **words** → Lista med ord som hittades på sidan.
- **boundingBox** → Koordinater för var ordet finns på sidan.
- **confidence** → Anger **säkerhetsnivå** för identifierad text (1 = 100% säker).

Sammanfattning

- API-svaret ger **strukturierad information** om dokumentets innehåll.
- Texten delas upp i **sidor, ord och positioner**.
- **Bounding box** gör att man kan använda textens placering, t.ex. i en visualisering.
- **Confidence score** hjälper till att bedöma hur tillförlitlig tolkningen är.

Det här gör det möjligt att **automatisera dokumenthantering och analys** i Azure! 🚀

Demo – Use prebuilt Document Intelligence models



Use the Read model

Use an app to use document analysis

© Copyright Microsoft Corporation. All rights reserved.

Instructor demo, can use this lab: [mslearn.ai-document-intelligence/Instructions/Exercises/01-use-prebuilt-models.md at main · MicrosoftLearning/mslearn.ai-document-intelligence \(github.com\)](https://mslearn.ai-document-intelligence/Instructions/Exercises/01-use-prebuilt-models.md)

Types of custom models

Custom classification

- Apply a label to the entire document
- Ideal for sorting large numbers of incoming documents into types
- Requires two different classes, and a minimum of five labeled documents per class
- One type of training model

Custom extraction

- Apply label to specific text
- Ideal for extracting custom labels from documents
- Requires five examples of the same document type
- Two training methods:
 - **Custom template (custom form)**
 - Training time: 1-5 minutes
 - Document structure: forms, templates, other structured documents
 - **Custom neural (custom document)**
 - Training time: 20-60 minutes
 - Document structure: structured and unstructured documents

© Copyright Microsoft Corporation. All rights reserved.

Types of Custom Models – Översikt

Den här bilden beskriver de två huvudtyperna av **anpassade modeller (custom models)** i Azure Document Intelligence:

1. Custom Classification (för dokumentklassificering)

2. Custom Extraction (för att extrahera specifik information från dokument)

1. Custom Classification (Dokumentklassificering)

- Använder etiketter för att klassificera hela dokument.
- Perfekt för att sortera stora volymer av inkommande dokument i olika typer (t.ex. fakturor vs. kvitton).
- Kräver minst **två klasser** och minst **fem märkta dokument per klass**.
- Använder **en typ** av träningsmodell.

Använtningsexempel: Automatiskt skilja mellan inkommande fakturor, kvitton och kontrakt.

2. Custom Extraction (Anpassad dataextraktion)

- Använder etiketter för att extrahera specifik **text** från dokument.
- Kräver **minst fem exempel av samma dokumenttyp** för träning.
- Används för att extrahera fält som **kundnamn, ordernummer eller fakturabelopp**.
- **Två träningsmetoder finns:**
 -  **Custom Template (Custom Form)**
 - **Träningstid:** 1-5 minuter
 - **Passar för:** Dokument med **fast layout**, t.ex. formulär, mallar och andra strukturerade dokument.
 -  **Custom Neural (Custom Document)**
 - **Träningstid:** 20-60 minuter
 - **Passar för:** Dokument med både **strukturerad** och **ostrukturerad layout**, t.ex. brev eller komplexa rapporter.
- Använtningsexempel:** Extrahera **fakturabelopp** från fakturor, **kvittonummer** från kvitton eller **kundnamn** från kontrakt.

Sammanfattning

- Custom Classification klassificerar hela dokumentet.
 - Custom Extraction extraherar **specifik** information från dokumentet.
 - Custom Extraction kan använda Template-modellen (för strukturerade dokument) eller Neural-modellen (för blandad dokumentstruktur).
- Dessa modeller gör det möjligt att **automatisera dokumenthantering** och extraktion av affärskritiska data i Azure
- 

Differences highlighted here. Similarities to include in lecture: trained in the same studio experience, both have template and neural model options, labeling is very similar

Training Custom Models

- 1 Create project and upload training files to your project, or connect to blob storage containing files
- 2 Add data type (such as field or signature) to start labeling your dataset
- 3 Select a word in the document, and assign one of the fields to label it
- 4 Repeat for all fields and files in your dataset
- 5 Layout and auto label (using a prebuilt model) can assist in this process
- 6 Train the model, providing a Model ID used in API requests

© Copyright Microsoft Corporation. All rights reserved.

Accuracy and confidence scores

- After training, a custom model has an estimated accuracy score
- Score is calculated by running combinations of training data predictions against the labeled values
- Confidence score is the same as using prebuilt models, indicated how accurate the model thinks that specific prediction is
- Confidence scores are provided in the response from the model for each predicted label

Accuracy

Email	80.00 %
CompanyAddress	80.00 %
Signature	80.00 %

Confidence

● Signature #1	44.80%
Wesley Smith	
● CompanyAddress #1	66.70%
342 W Wrinkle Road Bozeman MT 83839	
● Email #1	95.30%
accounts@herolimited.com	

© Copyright Microsoft Corporation. All rights reserved.

Analyze document using custom model

- Requires endpoint and key from deployed resource, similar to prebuilt models
- Needs to also include the ID of your deployed custom model
- Query the poller received for the extracted data

C#

```
AnalyzeDocumentOperation operation = await  
client.AnalyzeDocumentFromUriAsync(WaitUntil.Completed,  
{modelId}, {fileUri});
```

```
AnalyzeResult result = operation.Value;
```

Python

```
poller = document_analysis_client.begin_analyze_document(  
    model_id={modelId}, document={fileUri}  
  
result = poller.result()
```

© Copyright Microsoft Corporation. All rights reserved.

Lots of examples to expand on here, give a few a try or even get suggestions from the students

Lab – Extract Data from Forms



Train a custom model in Document Intelligence Studio

Test your custom Document Intelligence model

© Copyright Microsoft Corporation. All rights reserved.

Knowledge check



- 1** You have scanned a letter into PDF format and need to extract the text it contains. What should you do?

- Use the Image Analysis feature in Azure AI Vision.
- Use the Read model in the Document Intelligence service.
- Use a custom model in the Document Intelligence service.

- 2** You need to build an application that submits expense claims, extracting the merchant, date, and amount from scanned receipts. What's the best way to do this?

- Use the general document model.
- Use the prebuilt Contract model.
- Use the prebuilt Receipt model.

- 3** You need to extract only data from specific fields in cargo manifest forms using Document Intelligence. What should you do?

- Use a prebuilt model.
- Build a custom composed model from several custom models.
- Train custom template model with labeled documents.

© Copyright Microsoft Corporation. All rights reserved.

1 b

2 c

3 c

Självklart! Här är en förklaring av varför vissa svar är korrekta och varför andra är fel för varje fråga.

- 1** Du har skannat ett brev till PDF och behöver extrahera texten. Vad ska du göra?

- Rätt svar: "Use the Read model in the Document Intelligence service."

Förklaring:

- Read-modellen i Azure Document Intelligence är specialiserad för att extrahera text från både tryckta och handskrivna dokument, inklusive skannade PDF-filer.

- Den använder OCR (Optical Character Recognition) för att konvertera text till digitalt format.

- Fel svar: "Use the Image Analysis feature in Azure AI Vision."

- Image Analysis används mer för att identifiera objekt, ansikten och scenförståelse i bilder, men är inte optimerad för att extrahera text från skannade dokument.

- OCR-funktionerna i Azure AI Vision är mer begränsade än de i Document Intelligence.

- Fel svar: "Use a custom model in the Document Intelligence service."

- En custom model behövs endast om dokumenten har en mycket specifik struktur, där vissa fält måste extraheras på ett exakt sätt.

- Read-modellen fungerar direkt och kräver ingen träning, så en custom model är onödig i detta fall.

- 2** Du bygger en applikation för att hantera kvitton och behöver extrahera köpställe, datum och belopp. Vad är det bästa sättet?

- Rätt svar: "Use the prebuilt Receipt model."

Förklaring:

- Den förbyggda Receipt-modellen i Document Intelligence är specialiserad för att känna igen kvitton och extrahera viktiga fält som:

- Merchant (butik/affär)

- Date (inköpsdatum)
- Total amount (summa)

• Den är redan tränad och kräver ingen anpassning.

 Fel svar: "Use the general document model."

• General Document Model kan extrahera text men är inte specialiserad på kvitton.

• Den kan inte identifiera specifika fält som belopp, butik och datum lika bra som Receipt-modellen.

 Fel svar: "Use the prebuilt Contract model."

• Contract-modellen är avsedd för juridiska dokument och avtal, inte kvitton.

• Den kan känna igen juridiska termer men saknar stöd för att extrahera belopp och transaktionsdetaljer från kvitton.

3 Du behöver extrahera endast specifika fält från fraktmanifest med Document Intelligence. Vad ska du göra?

 Rätt svar: "Train custom template model with labeled documents."

Förklaring:

• Template-modellen låter dig **träna en modell på specifika fält** i strukturerade dokument som fraktmanifest.

• Eftersom manifest ofta har en **fast layout** fungerar det bäst att **träna en modell som exakt identifierar och extraherar rätt information**.

 Fel svar: "Use a prebuilt model."

• Det finns ingen **förbyggd modell för fraktmanifest**, så en generisk modell skulle **inte kunna extrahera exakt de fält du behöver**.

• Prebuilt-modeller fungerar bäst för vanliga dokumenttyper som **kvitton, fakturor och identitetshandlingar**.

 Fel svar: "Build a custom composed model from several custom models."

• Composed models används för att **kombinera flera custom models** för olika dokumenttyper.

• Om du bara har **en specifik typ av dokument (fraktmanifest)**, behöver du bara **träna en template model**, inte kombinera flera olika modeller.

◆ Sammanfattning av rätt svar och varför:

Fråga Rätt svar Varför? **1** Extrahera text från en skannad PDF (brev).  Read model i Document Intelligence
Read-modellen är optimerad för OCR och textutvinning från skannade dokument. **2** Extrahera data från kvitton (butik, datum, belopp).  Prebuilt Receipt model Receipt-modellen är specialiserad för kvitton och kan extrahera relevanta fält. **3** Extrahera specifika fält från fraktmanifest.  Train custom template model
Template-modellen gör det möjligt att träna en modell på exakt de fält du behöver.  Sammanfattning:

• **Read Model** används för generell OCR på skannade dokument.

• **Prebuilt Receipt Model** används för kvitton eftersom den redan är tränad.

• **Custom Template Model** används när du behöver extrahera specifika fält från strukturerade dokument som fraktmanifest.

Nu har du en klar bild av varför dessa svar är rätt! 😊🚀

Learning Path Recap

In this learning path, we:

Explored available prebuilt models, and how to use them in Document Intelligence Studio

Trained and deployed a custom model

Connected an app to use Document Intelligence APIs

© Copyright Microsoft Corporation. All rights reserved.



© Copyright Microsoft Corporation. All rights reserved.