Microsoft

Develop solutions with Azure AI Document Intelligence

Agenda

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

Develop a Document Intelligence solution

Learning Objectives

After completing this module, you will be able to:

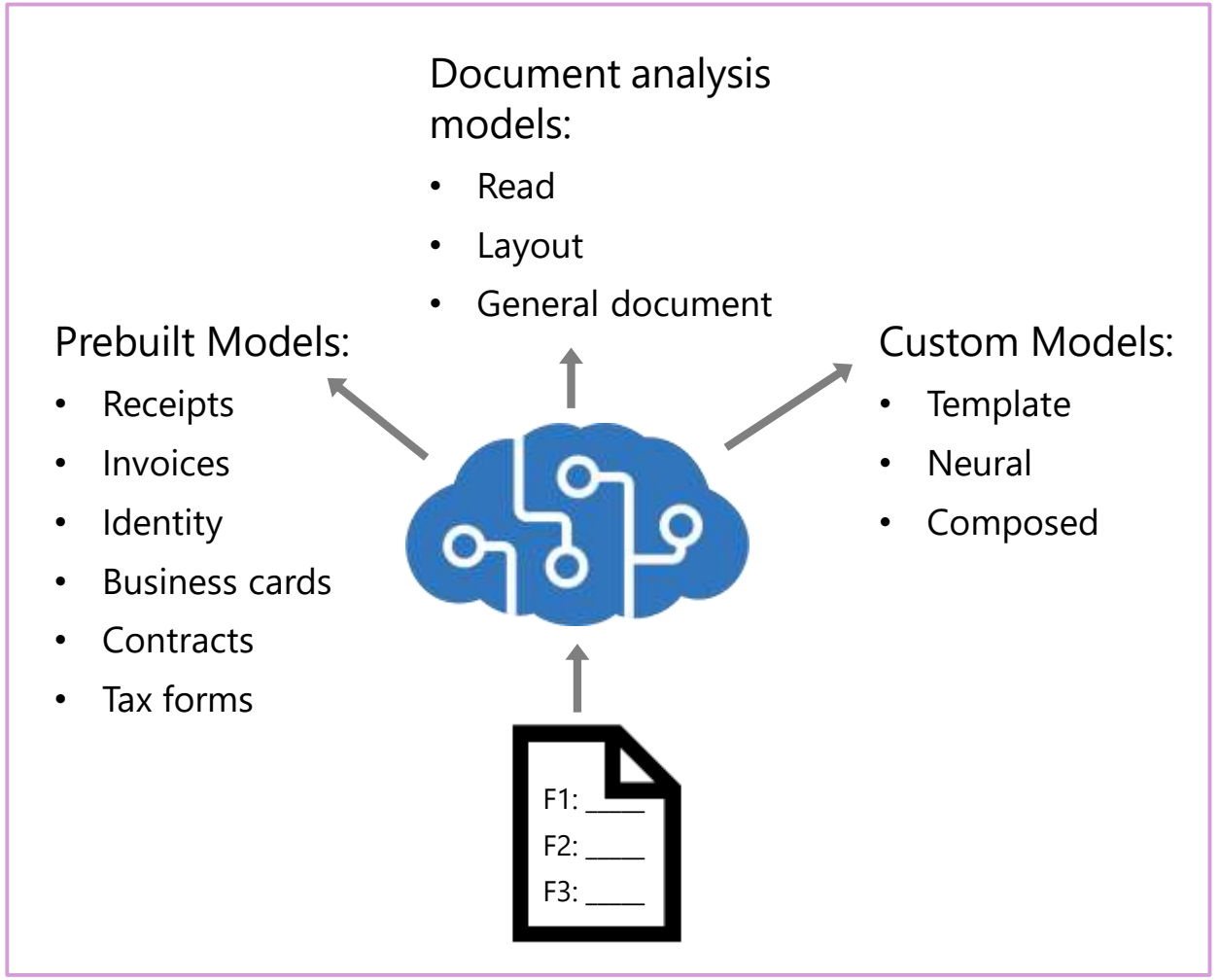
1. Understand models in Azure AI Document Intelligence
2. Train a custom Document Intelligence model
3. Connect an app to Document Intelligence APIs

The Document Intelligence Service

Data extraction from forms and documents:

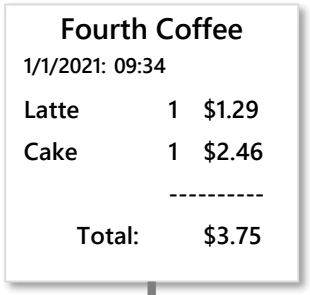
- Document analysis from general documents
  - Read: OCR for printed and written text
  - Layout: Extract text and structure
  - General document: Extract text, structure, and key-value pairs
- Prebuilt models for common form types
- Train custom models for your own forms
  - Custom template: Extract 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



Prebuilt models

Receipt

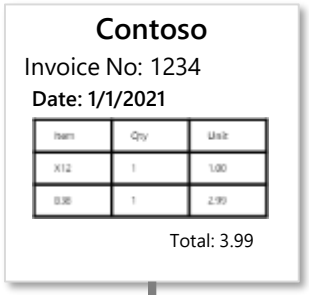


Analyze Receipt

Get Analyze Receipt Result

```
{
  "MerchantName": "Fourth Coffee",
  "TransactionDate": 2021-01-01,
  "TransactionTime": 09:34,
  "Items" [
    ...],
  "Total": 3.75
}
```

Invoice



Analyze Invoice

Get Analyze Invoice Result

```
{
  "VendorName": "Contoso",
  "InvoiceNumber": 1234,
  "InvoiceDate": 2021-01-01
  "Tables" [
    ...],
  "TotalInvoiceAmount": 3.99
}
```

Business Card

Analyze Business Card

Get Analyze Business Card Result

```
{
  "ContactNames": [
    {
      "FirstName": "Hank",
      "LastName": "Zoeng"
    }],
    ...
}
```

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}/formrecognizer/documentModels/prebuilt-layout
:analyze?api-version={version}

Operation-Location:
{endpoint}/formrecognizer/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;
```

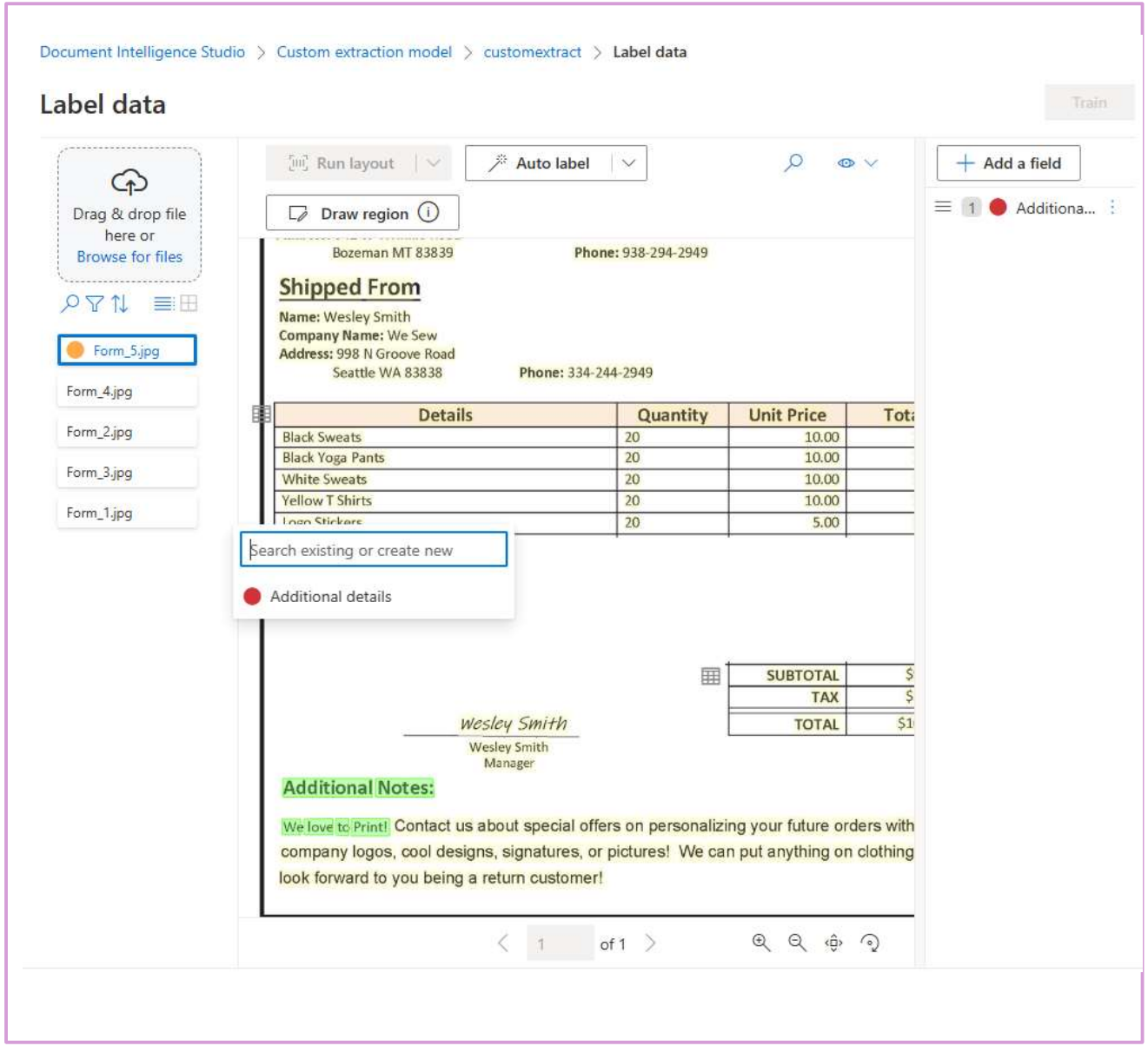
Python

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

result = poller.result()
```

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



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()
```

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