

## Contents

What is Form Recognizer? .....	2
Form Recognizer Studio .....	4
ID Document Model.....	5
Invoice Model .....	9
General Document Model .....	14
Custom Model.....	18

# Form Recognizer


## What is Form Recognizer?

Azure Form Recognizer is an [Azure Applied AI Service](#) that enables you to build automated document processing software using machine learning technology. Form Recognizer analyzes your forms and documents, extracts text and data, maps field relationships as key-value pairs, and returns a structured JSON output. You quickly get accurate results that are tailored to your specific content without excessive manual intervention or extensive data science expertise. Use Form Recognizer to automate your data processing in applications and workflows, enhance data-driven strategies, and enrich document search capabilities.

Form Recognizer easily identifies, extracts, and analyzes the following document data:

- Table structure and content.
- Form elements and field values.
- Typed and handwritten alphanumeric text.
- Relationships between elements.
- Key/value pairs
- Element location with bounding box coordinates.

The following features and development options are supported by Form Recognizer v3.0. Use the links in the table to learn more about each feature and browse the API references.


Feature	Description	Development options
 <a href="#">General document model</a>	Extract text, tables, structure, key-value pairs and, named entities.	<ul style="list-style-type: none"><li>• <a href="#">Form Recognizer Studio</a></li><li>• <a href="#">REST API</a></li><li>• <a href="#">C# SDK</a></li><li>• <a href="#">Python SDK</a></li></ul>
<a href="#">Layout model</a>	Extract text, selection marks, and tables structures, along with their bounding box coordinates, from forms and documents.  Layout API has been updated to a prebuilt model.	<ul style="list-style-type: none"><li>• <a href="#">Form Recognizer Studio</a></li><li>• <a href="#">REST API</a></li><li>• <a href="#">C# SDK</a></li><li>• <a href="#">Python SDK</a></li></ul>
<a href="#">Custom model (updated)</a>	Extraction and analysis of data from forms and documents specific to distinct business data and use cases.  Custom model API v3.0 supports <b>signature detection for custom forms</b> .	<ul style="list-style-type: none"><li>• <a href="#">Form Recognizer Studio</a></li><li>• <a href="#">REST API</a></li><li>• <a href="#">C# SDK</a></li><li>• <a href="#">Python SDK</a></li></ul>
<a href="#">Invoice model</a>	Automated data processing and extraction of key information from sales invoices.	<ul style="list-style-type: none"><li>• <a href="#">Form Recognizer Studio</a></li><li>• <a href="#">REST API</a></li><li>• <a href="#">C# SDK</a></li></ul>

		<ul style="list-style-type: none"> <li>• <a href="#">Python SDK</a></li> </ul>
<a href="#">Receipt model (updated)</a>	<p>Automated data processing and extraction of key information from sales receipts.</p> <p>Receipt model v3.0 supports processing of <b>single-page hotel receipts</b>.</p>	<ul style="list-style-type: none"> <li>• <a href="#">Form Recognizer Studio</a></li> <li>• <a href="#">REST API</a></li> <li>• <a href="#">C# SDK</a></li> <li>• <a href="#">Python SDK</a></li> </ul>
<a href="#">ID document model (updated)</a>	<p>Automated data processing and extraction of key information from US driver's licenses and international passports.</p> <p>Prebuilt ID document API supports the <b>extraction of endorsements, restrictions, and vehicle classifications from US driver's licenses</b>.</p>	<ul style="list-style-type: none"> <li>• <a href="#">Form Recognizer Studio</a></li> <li>• <a href="#">REST API</a></li> <li>• <a href="#">C# SDK</a></li> <li>• <a href="#">Python SDK</a></li> </ul>
<a href="#">Business card model</a>	Automated data processing and extraction of key information from business cards.	<ul style="list-style-type: none"> <li>• <a href="#">Form Recognizer Studio</a></li> <li>• <a href="#">REST API</a></li> <li>• <a href="#">C# SDK</a></li> <li>• <a href="#">Python SDK</a></li> </ul>

### Form Recognizer Workflow

Document type	Considerations	Solution
Invoice Receipt Business card	Is your invoice, receipt, or business card document composed of English-text?	Yes → Invoice, Receipt, or Business Card model No → Layout or General document (preview) model
ID document	Is your ID document a US driver's license or an international passport?	Yes → ID document model No → Layout or General document (preview) model
Form or Document	Is your form or document an industry-standard format commonly used in your business or industry?	Yes → Layout or General document (preview) model No → Train and build a custom model

### Model Overview

Model	Description
 <b><u>General document (preview)</u></b>	Extract text, tables, structure, key-value pairs, and named entities.
<b><u>Layout</u></b>	Extracts text and layout information from documents.
<b><u>Invoice</u></b>	Extract key information from English invoices.
<b><u>Receipt</u></b>	Extract key information from English receipts.
<b><u>ID document</u></b>	Extract key information from US driver licenses and international passports.
<b><u>Business card</u></b>	Extract key information from English business cards.
<b><u>Custom</u></b>	Extract data from forms and documents specific to your business. Custom models are trained for your distinct data and use cases.

## Form Recognizer Studio

[Form Recognizer Studio preview](#) is an online tool for visually exploring, understanding, and integrating features from the Form Recognizer service into your applications.

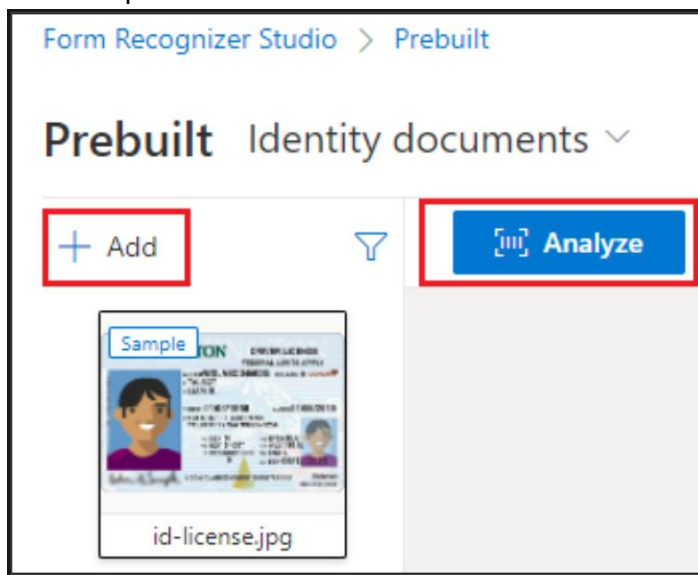
The following Form Recognizer service features are available in the Studio.

1. [Layout](#): Try out Form Recognizer's Layout feature to extract text, tables, selection marks, and structure information from documents—PDF, TIFF—and images—JPG, PNG, BMP.
2. [Prebuilt models](#): Form Recognizer's pre-built models enable you to add intelligent form processing to your apps and flows without having to train and build your own models.
3. [Custom models](#): Form Recognizer's custom models enable you to extract fields and values from models trained with your data, tailored to your forms and documents. Create standalone custom models or combine two or more custom models to create a composed model to extract data from multiple form types.
4. Custom models: [Labeling](#) features: Form Recognizer Custom model creation requires identifying the fields to be extracted and labeling those fields before training the custom models. Labeling text, selection marks, [signature detection](#), tabular data, and other content types are typically assisted with a user interface to ease the training workflow.

## ID Document Model

The ID document model combines powerful Optical Character Recognition (OCR) capabilities with deep learning models to analyze and extracts key information from U.S. Driver's Licenses (all 50 states and District of Columbia) and international passport biographical pages (excluding visa and other travel documents). The API analyzes identity documents; extracts key information such as first name, last name, address, and date of birth; and returns a structured JSON data representation.

1. On the Form Recognizer Studio home page, select Identify documents or click on link below - [FormRecognizerStudio \(azure.com\)](https://formrecognizer.azure.com)
2. You can analyze the sample ID (or passport) or select the + Add button to upload your own sample.



3. Click on +Add to select one of the data from hackathon dataset.
4. Select the Analyze button:

Prebuilt Identity documents ▾

+ Add



Analyze



Arizona\_DL.jpg



id-license.jpg



id-passport.jpg



5. After analyzing you will see the output of the document

Values	Result	Code
<div>Address #1</div> <div>2149 LAKEWOOD DRIVE PRESCOTT, AZ 86301-5471</div>	91.70%	
<div>CountryRegion</div> <div>USA</div>	99.50%	
<div>DateOfBirth #1</div> <div>1995-02-01</div>	98.80%	
<div>DateOfExpiration #1</div> <div>2031-02-01</div>	98.80%	
<div>DocumentNumber #1</div> <div>D09257418</div>	98.50%	
<div>Endorsements #1</div> <div>NONE</div>	96.10%	
<div>FirstName #1</div> <div>IDINSTATE MIDDLE</div>	88.20%	
<div>LastName #1</div>	89.10%	

6. You can also view the Result in JSON and python code for custom execution

Following are the list of the fields that are extracted out of the box

Name	Type	Description	Standardized output
<div>NEW</div> <div>Endorsements</div>	String	Additional driving privileges granted to a driver such as Motorcycle or School bus.	

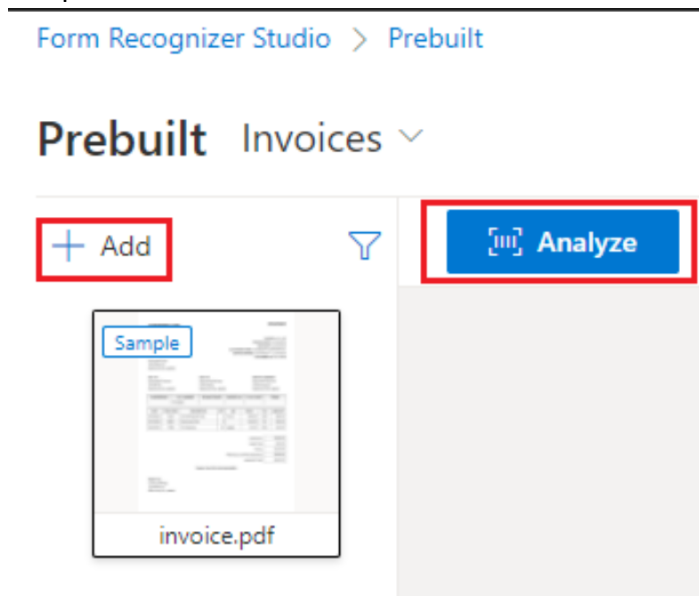
Restrictions	String	Restricted driving privileges applicable to suspended or revoked licenses.	
VehicleClassification	String	Types of vehicles that can be driven by a driver.	
CountryRegion	countryRegion	Country or region code compliant with ISO 3166 standard	
DateOfBirth	Date	DOB	yyyy-mm-dd
DateOfExpiration	Date	Expiration date DOB	yyyy-mm-dd
DocumentNumber	String	Relevant passport number, driver's license number, etc.	
FirstName	String	Extracted given name and middle initial if applicable	
LastName	String	Extracted surname	
Nationality	countryRegion	Country or region code compliant with ISO 3166 standard (Passport only)	
Sex	String	Possible extracted values include "M", "F" and "X"	
MachineReadableZone	Object	Extracted Passport MRZ including two lines of 44 characters each	"P<USABRO OKS<<JENNI FER<<<<<<<< <<<<<<<<<<<< <<<<<<< 3400200135 USA800101 4F19050547 10000307<7 15816"
DocumentType	String	Document type, for example, Passport, Driver's License	"passport"
Address	String	Extracted address (Driver's License only)	
Region	String	Extracted region, state, province, etc. (Driver's License only)	



## Invoice Model

The invoice model combines powerful Optical Character Recognition (OCR) capabilities with deep learning models to analyze and extract key fields and line items from sales invoices. Invoices can be of various formats and quality including phone-captured images, scanned documents, and digital PDFs. The API analyzes invoice text; extracts key information such as customer name, billing address, due date, and amount due; and returns a structured JSON data representation.

1. On the Form Recognizer Studio home page, select Invoices or click on link below - [FormRecognizerStudio \(azure.com\)](https://formrecognizer.azure.com)
2. You can analyze the sample invoice or select the + Add button to upload your own sample.



3. Click on +Add to select one of the data from hackathon dataset.
4. Select the Analyze button:

## Prebuilt Invoices ▾


+ Add

ServiceE...ate\_1.jpg

Sample

invoice.pdf

Analyze



ServiceE...ate\_1.jpg

Sample

invoice.pdf

Invoice

INVOICE ID	01-85-2753
Warranty CLAIM ID	24-7422105
Date of Service	05/20/2024

Customer

Name	John A. Smith
Phone	913-567-1234
Model	Rosalia CR-V
License Plate	MA 9D0-108

QTY	CODE	DESCRIPTION	PRICE	AMOUNT
1	01852753	01852753	250.00	250.00
1	01852753	01852753	250.00	250.00
1	01852753	01852753	250.00	250.00
Parts & Labor Subtotal			750.00	750.00
Manufacturer Warranty Eligible (\$113.32)				
Tax			55.00	55.00
Total			805.00	805.00

Customer Signature

Date

< 1 of 1 >

🔍

🔍

🔍

🔍

- After analyzing you will see the output of the document

Values	Result	Code
● CustomerName #1	95.60%	
Evan Kelly		
● InvoiceId #1	98.00%	
01-8527157		
● InvoiceTotal #1	93.20%	
65.55		
● Items (3) #1		
1 Amount	90.00%	
13.5		
Description	89.20%	
WHEEL SPIN BALANCE		
ProductCode	90.00%	
27-2620232		
Quantity	90.10%	
1		
UnitPrice	86.00%	
13.5		

6. You can also view the Result in JSON and python code for custom execution

Following are the header level data extracted from the invoice :

Name	Type	Description	Standardized output
CustomerName	String	Invoiced customer	
CustomerId	String	Customer reference ID	
PurchaseOrder	String	Purchase order reference number	

<b>InvoiceId</b>	String	ID for this specific invoice (often "Invoice Number")	
<b>InvoiceDate</b>	Date	Date the invoice was issued	yyyy-mm-dd
<b>DueDate</b>	Date	Date payment for this invoice is due	yyyy-mm-dd
<b>VendorName</b>	String	Vendor name	
<b>VendorAddress</b>	String	Vendor mailing address	
<b>VendorAddressRecipient</b>	String	Name associated with the VendorAddress	
<b>CustomerAddress</b>	String	Mailing address for the Customer	
<b>CustomerAddressRecipient</b>	String	Name associated with the CustomerAddress	
<b>BillingAddress</b>	String	Explicit billing address for the customer	
<b>BillingAddressRecipient</b>	String	Name associated with the BillingAddress	
<b>ShippingAddress</b>	String	Explicit shipping address for the customer	
<b>ShippingAddressRecipient</b>	String	Name associated with the ShippingAddress	
<b>SubTotal</b>	Number	Subtotal field identified on this invoice	Integer
<b>TotalTax</b>	Number	Total tax field identified on this invoice	Integer
<b>InvoiceTotal</b>	Number (USD)	Total new charges associated with this invoice	Integer
<b>AmountDue</b>	Number (USD)	Total Amount Due to the vendor	Integer
<b>ServiceAddress</b>	String	Explicit service address or property address for the customer	
<b>ServiceAddressRecipient</b>	String	Name associated with the ServiceAddress	
<b>RemittanceAddress</b>	String	Explicit remittance or payment address for the customer	
<b>RemittanceAddressRecipient</b>	String	Name associated with the RemittanceAddress	
<b>ServiceStartDate</b>	Date	First date for the service period (for example, a utility bill service period)	yyyy-mm-dd
<b>ServiceEndDate</b>	Date	End date for the service period (for example, a utility bill service period)	yyyy-mm-dd
<b>PreviousUnpaidBalance</b>	Number	Explicit previously unpaid balance	Integer

At Line level following information is extracted

Name	Type	Description	Text (line item #1)	Value (standardized output)
Items	String	Full string text line of the line item	3/4/2021 A123 Consulting Services 2 hours \$30.00 10% \$60.00	
Amount	Number	The amount of the line item	\$60.00	100
Description	String	The text description for the invoice line item	Consulting service	Consulting service
Quantity	Number	The quantity for this invoice line item	2	2
UnitPrice	Number	The net or gross price (depending on the gross invoice setting of the invoice) of one unit of this item	\$30.00	30
ProductCode	String	Product code, product number, or SKU associated with the specific line item	A123	
Unit	String	The unit of the line item, e.g, kg, lb etc.	Hours	
Date	Date	Date corresponding to each line item. Often it is a date the line item was shipped	3/4/2021	2021-03-04
Tax	Number	Tax associated with each line item. Possible values include tax amount, tax %, and tax Y/N	10%	

## General Document Model

The General document preview model combines powerful Optical Character Recognition (OCR) capabilities with deep learning models to extract key-value pairs and entities from documents.

The general document API supports most form types and will analyze your documents and associate values to keys and entries to tables that it discovers. It is ideal for extracting common key-value pairs from documents.

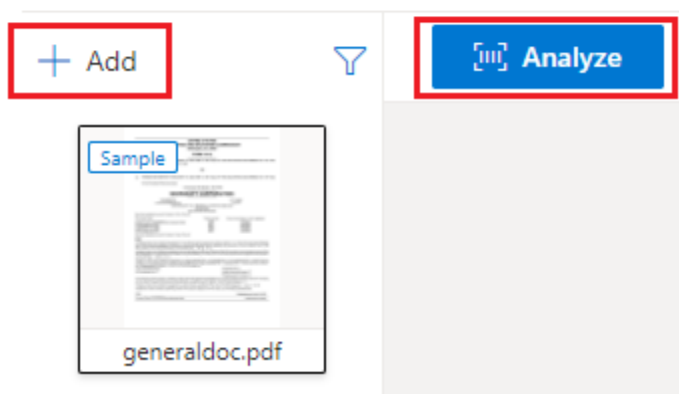
Features:

- There is no need to train a custom model to extract key-value pairs.
- A single API is used to extract key value pairs, entities, text, tables, and structure from documents.
- It is a pre-trained model that will be periodically trained on new data to improve coverage and accuracy.
- The general document model supports structured, semi-structured, and unstructured data.

1. On the Form Recognizer Studio home page, select General Documents or click on link below - [FormRecognizerStudio \(azure.com\)](https://formrecognizer.azure.com)
2. You can analyze the sample document or select the + Add button to upload your own sample.

[Form Recognizer Studio](#) > [Prebuilt](#)

**Prebuilt** General documents ▾



3. Click on +Add to select one of the data from hackathon dataset (insurance card).
4. Select the Analyze button:

+

 Add

🔍

 Analyze

📁

 ▾

Alaska Insurance.png

generaldoc.pdf

INSURANCE IDENTIFICATION CARD

ALASKA

STATE

COMPANY NUMBER

12345

COMPANY

Any Insurance Company

POLICY NUMBER

ABC987654321

EFFECTIVE DATE

1/1/2005

EXPIRATION DATE

1/1/2006

YEAR

2005

MAKE/MODEL

HONDA / ODYSSEY

VEHICLE IDENTIFICATION NUMBER

5FNRL38855B0059T1

AGENCY/COMPANY ISSUING CARD

Your Insurance Agency/Company

1234 Main Street

AnyCity, US 12345

INSURER

Empire Parts Inc

210 Washington Ave

Albany, NY 12210-1312

SEE IMPORTANT NOTICE ON REVERSE SIDE

- After analyzing you will see the output of the document

Values	Result	Code
Key-Value pairs		Entities
● (STATE)		11.80%
ALASKA		
● COMPANY NUMBER		94.60%
12345		
● COMPANY		92.90%
Any Insurance Company		
● POLICY NUMBER		94.60%
ABC987654321		
● EFFECTIVE DATE		94.60%
1/1/2005		
● EXPIRATION DATE		94.60%
1/1/2006		
● YEAR		95.40%
2005		
● MAKE/MODEL		50.00%

6. You can also view the Result in JSON and python code for custom execution

### Key-value Pairs

Key value pairs are specific spans within the document that identify a label or key and its associated response or value. In a structured form, this could be the label and the value the user entered for that field or in an unstructured document it could be the date a contract was executed on based on the text in a paragraph. The AI model is trained to extract identifiable keys and values based on a wide variety of document types, formats, and structures.



Keys can also exist in isolation when the model detects that a key exists, with no associated value or when processing optional fields. For example, a middle name field may be left blank on a form in some instances. Key value pairs are always spans of text contained in the document and if you have documents where same value is described in different ways, for example a customer or a user, the associated key will be either customer or user based on what the document contained.

## Entities

Natural language processing models can identify parts of speech and classify each token or word. The named entity recognition model is able to identify entities like people, locations, and dates to provide for a richer experience. Identifying entities enables you to distinguish between customer types, for example, an individual or an organization. The key value pair extraction model and entity identification model are run in parallel on the entire document and not just on the values of the extracted key value pairs. This ensures that complex structures where a key cannot be identified is still enriched by identifying the entities referenced. You can still match keys or values to entities based on the offsets of the identified spans.

Following NER categories are supported out of the box:

Category	Type	Description
<b>Person</b>	String	A person's partial or full name.
<b>PersonType</b>	String	A person's job type or role.
<b>Location</b>	String	Natural and human-made landmarks, structures, geographical features, and geopolitical entities.
<b>Organization</b>	String	Companies, political groups, musical bands, sport clubs, government bodies, and public organizations.
<b>Event</b>	String	Historical, social, and naturally occurring events.
<b>Product</b>	String	Physical objects of various categories.
<b>Skill</b>	String	A capability, skill, or expertise.
<b>Address</b>	String	Full mailing addresses.
<b>Phone number</b>	String	Phone numbers.
<b>Email</b>	String	Email address.
<b>URL</b>	String	Website URLs and links.
<b>IP Address</b>	String	Network IP addresses.
<b>DateTime</b>	String	Dates and times of day.
<b>Quantity</b>	String	Numerical measurements and units.

## Custom Model

Form Recognizer custom models enable you to analyze and extract data from forms and documents specific to your business. Custom models are trained for your distinct data and use cases

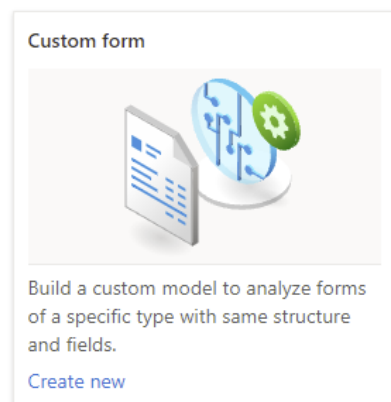
A custom model is a machine learning program trained to recognize form fields within your distinct content and extract key-value pairs and table data. You only need five examples of the same form type to get started and your custom model can be trained with or without labeled datasets.

While we could have used the “General Document” Model to extract the data from Insurance card, for hackathon instead we will build the custom model for that.

1. Navigate to the Form Recognizer Studio and select **Custom form** under Custom models

### Custom models

Extract text, structure, and fields from models trained with your own data, so they're tailored to your forms and documents.



2. Follow the workflow to create a new project:

- Enter project details
- Configure service resource
- Connect training data source
- Review and create

Enter project details

Name your project and optionally, provide a brief description.

**Project name \***

Enter project name

**Description**

Enter project description.

- To create a project, you need the following:
- Azure Form Recognizer or Azure Cognitive Services resource
  - Azure Blob Storage container with the training data
  - At least five different forms of the same type to help train your custom model. [Read requirements](#)

- ✓ Enter project details
- Configure service resource
- Connect training data source
- Review and create

Configure service resource

To create a project in Form Recognizer Studio, you will need an Azure subscription containing a service resource for usage and billing. Resources are organized in resource groups. [Learn more](#)

**Subscription \***

Shared Sub - 156

**Resource group \***

FSIUC3-478734

[Create new](#)

**Form Recognizer or Cognitive Service Resource \***

od478734frcs

od478734frcs resource

Custom Form

Enter project details

Configure service resource

Connect training data source

Review and create

Connect training data source

Link the Azure Blob Storage account and the folder that contains your training data. [Learn more](#)

Subscription \*

Shared Sub - 156

Resource group \*

FSIUC3-478734

Storage account \*

od478734sa

Blob container \*

train

Folder path

Insurance/

Back

Continue

Cancel

Custom Form

Enter project details

Configure service resource

Connect training data source

Review and create

Review and create

Project name

Insurance

Description

-

Subscription

Shared Sub - 156

Resource group

FSIUC3-478734

Form Recognizer or Cognitive Service Resource

od478734frcs

Subscription

Shared Sub - 156

Resource group

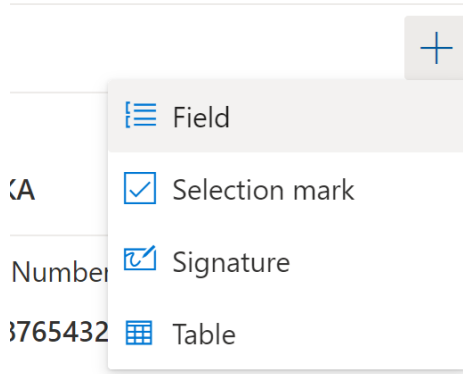
FSIUC3-478734

Back

Create project

Cancel

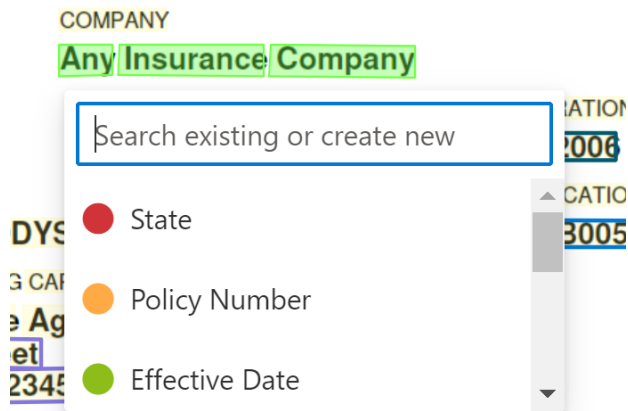
3. Select + Field to create custom tags/labels



Following fields (and any additional you want) needs to be created:

State	⋮
ALASKA	×
Policy Number	⋮
ABC987654321	×
Effective Date	⋮
1/1/2005	×
Expiration Date	⋮
1/1/2006	×
VIN	⋮
5FNRL38855B005911	×
Agency Address	⋮
1234 Main Street AnyCity, US 12345	×
Insured Address	⋮
210 Washington Ave NY 12210-1312 Albany,	×

4. Select the OCR (yellow color) text to label the sample data. Once it's labelled it will show that as labelled tag. Repeat that for all the fields you created and all 5 sample data



5. Once all the labelling is completed. Click on Train to build the model. Enter model name and click train

Train

Train a new model

Model ID \*

Insurance

Model Description

Enter model description

Train

Cancel

6. Upon successful completion of the model, it will show the accuracy for each field

## Insurance1



Field Name	Accuracy
State	40 %
Policy Number	80 %
Effective Date	80 %
Expiration Date	60 %
VIN	40 %
Agency Address	40 %
Insured Address	60 %
Company	40 %

Test

Close

7. Click on Test to test the newly built model
8. Click on +Add and select the insurance test data and click on Analyze

Test model Insurance1 ▾

+

Add

Analyze

ALASKA

INSURANCE IDENTIFICATION CARD

COMPANY NUMBER

12345

COMPANY

Any Insurance Company

POLICY NUMBER

ABC987654321

EFFECTIVE DATE

1/1/2005

EXPIRATION DATE

1/1/2006

YEAR

2005

MAKE/MODEL

HONDA / ODYSSEY

VEHICLE IDENTIFICATION NUMBER

5FNRL38855B005911

AGENCY/COMPANY ISSUING CARD

Your Insurance Agency/Company

1234 Main Street

AnyCity, US 12345

INSURED

Empire Parts Inc

210 Washington Ave

Albany, NY 12210-1312

SEE IMPORTANT NOTICE ON REVERSE SIDE

Alaska\_Insurance.png

- After completion of analysis, it will show you the custom entities/keys that are extracted.



Values	Result	Code
<div>● Agency Address #1</div> <div>1234 Main Street AnyCity, US 12345</div>	88.70%	
<div>● Company #1</div> <div>Any Insurance Company</div>	88.80%	
<div>● Insured Address #1</div> <div>210 Washington Ave Albany, NY 12210-1312</div>	94.40%	
<div>● VIN #1</div> <div>5FNRL38855B005911</div>	92.70%	
<div>● Policy Number #1</div> <div>ABC987654321</div>	95.00%	
<div>● Effective Date #1</div> <div>1/1/2005</div>	95.00%	
<div>● Expiration Date #1</div> <div>1/1/2006</div>	95.00%	
<div>● State #1</div>	94.80%	

**Note: Currently Logicapp workflow doesn't have a new connector for Form recognizer v3.0. Since Form recognizer Studio is creating the model as v3, for logic app to function, we will re-create the same model in v2.1**

Let's create the same model in FOTT labeling tool.

1. Go to Azure Portal (portal.azure.com). Login with your credentials. Go to the storage account within your resource group. Click on Shared Access signature

od478734sa | Shared access signature ...

Storage account

Search (Ctrl+/)

Diagnose and solve problems

Access Control (IAM)

Data migration

Events

Storage browser (preview)

Data storage

Containers

File shares

Queues

Tables

Security + networking

Networking

Azure CDN

Access keys

Shared access signature

Encryption

Security

Data management

Allowed services ⓘ

☒ Blob ☒ File ☒ Queue ☒ Table

Allowed resource types ⓘ

☒ Service ☒ Container ☒ Object

Allowed permissions ⓘ

☒ Read ☒ Write ☒ Delete ☒ List ☒ Add ☒ Create ☒ Update ☒ Process ☒ Immutable storage

Blob versioning permissions ⓘ

☒ Enables deletion of versions

Allowed blob index permissions ⓘ

☒ Read/Write ☒ Filter

Start and expiry date/time ⓘ

Start 10/30/2021 6:35:16 PM

End 10/31/2022 2:35:16 AM

(UTC-06:00) Central Time (US & Canada)

Allowed IP addresses ⓘ

For example, 168.1.5.65 or 168.1.5.65-168.1.5.70

Allowed protocols ⓘ

☒ HTTPS only ☐ HTTPS and HTTP

Preferred routing tier ⓘ

☒ Basic (default) ☐ Microsoft network routing ☐ Internet routing

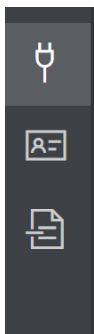
Some routing options are disabled because the endpoints are not published.

Ensure “Service, Container and Object” are selected in allowed resource type and the end date is a year from today. Click on Generate SAS String

Generate SAS and connection string

Copy the Blob Service SAS URL. You will need that in next steps.

2. Go to FOTT labelling tool - [Analyze - Form OCR Testing Tool \(fott-2-1.azurewebsites.net\)](https://fott-2-1.azurewebsites.net/)
3. Click on Connection to create a new connection where our train dataset is hosted.



4. Enter the connection name (insurance), Select provider as Azure Blob container and SAS URI that was copied from Step 1. **Note: Make sure to append /train in the copied SAS URL**

### Connection Settings

⚠ Warning: CORS (Cross Domain Resource Sharing) must be enabled on the Azure Blob Storage account, in order to use it as a source or target connection. More information on enabling CORS can be found in the [Azure Documentation](#).

Display name\*  
Insuranceodl ✓

Description  
✓

Provider  
Azure blob container ✓

Azure blob container

SAS URI\*  
https://od478734sa.blob.core.windows.net/train?sp=racwdl&st=2021-10-29T02:26:37Z&se=2022-10-29T10:26:37Z&spr=ht ✓

Shared access signature URI to the blob container

Save Connection Cancel

5. Save the connection
6. Go back to home page and click on New Project. Enter display name, select the security token, "Insurance" as Folder Path. You can get the Form recognizer service URI & API key from the Azure Portal -> Resource Group -> Form Recognizer service -> Keys & Endpoint

### Project Settings

Display name\*  
InsuranceOdl ✓

Security token  
InsuranceOdl Token ✓  
Used to encrypt sensitive data within project files

Folder path  
Insurance ✓  
Subfolder path to your files

Form recognizer service URI\*  
https://eastus.api.cognitive.microsoft.com/ ✓  
Form recognizer service URI

API key  
..... ✓

API version  
v2.1 (default) ✓

Description  
✓

Save Project Cancel

7. Save the Project

8. Since we already labelled the data in Form Recognizer Studio, you should see the labelled data and everything configured opening up the project.

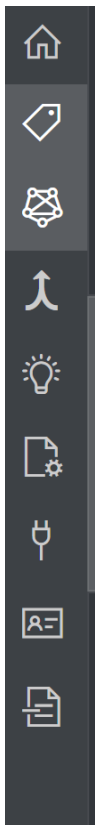
The screenshot shows the Microsoft Form Recognizer Studio interface. The main canvas displays a sample insurance identification card with various fields labeled. The labels are as follows:

- ARIZONA** (STATE)
- INSURANCE IDENTIFICATION CARD**
- COMPANY NUMBER**: 12345
- COMPANY**: Any Insurance Company
- POLICY NUMBER**: ABC987654321
- EFFECTIVE DATE**: 1/1/2005
- EXPIRATION DATE**: 1/1/2006
- YEAR**: 2005
- MAKE/MODEL**: HONDA / ODYSSEY
- VEHICLE IDENTIFICATION NUMBER**: 5FNRL38855B005911
- AGENCY/COMPANY ISSUING CARD AND PHONE NUMBER**: 123-456-7890
- Your Insurance Agency/Company**: 1234 Main Street, AnyCity, US 12345
- INSURED**: Empire Parts Inc, 210 Washington Ave, Albany, NY 12210-1312
- SEE IMPORTANT NOTICE ON REVERSE SIDE**

The right sidebar shows a list of tags with their corresponding values:

- State**: ARIZONA
- Policy Number**: ABC987654321
- Effective Date**: 1/1/2005
- Expiration Date**: 1/1/2006
- VIN**: 5FNRL38855B005911
- Agency Address**: 1234 Main Street AnyCity, US 12345
- Insured Address**: 210 Washington Ave Albany, NY 12210-1312
- Company**: Any Insurance Company

9. Click on “Train” from the icon menu



10. It will train the model using Form Recognizer v2.1 and generate Model Id that we will use. Save the Model Id for Logic app workflow

Train Result

Model ID: c65e216d-4a7e-40d8-aace-60e54f31e104

Tag	Estimated Accuracy
Agency Address	40.00%
Effective Date	80.00%
Expiration Date	60.00%
Insured Address	60.00%
Policy Number	80.00%
State	40.00%
VIN	40.00%

Train a new model

Model name

Add a model name...

Train

Training record

Model information

Model ID:  
c65e216d-4a7e-40d8-aace-60e54f31e104

Created date and time:  
10/28/2021, 9:29:43 PM

Average accuracy:  
57.10%

[Learn more about improving model accuracy.](#)

[Download JSON file](#)