Contents

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
NEW General document model	Extract text, tables, structure, key-value pairs and, named entities.	• **Form Recognizer Studio** • **REST API** • **C# SDK** • **Python SDK**
Layout model	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.	• **Form Recognizer Studio** • **REST API** • **C# SDK** • **Python SDK**
Custom model (updated)	Extraction and analysis of data from forms and documents specific to distinct business data and use cases. Custom model API v3.0 supports **signature detection for custom forms**.	• **Form Recognizer Studio** • **REST API** • **C# SDK** • **Python SDK**
Invoice model	Automated data processing and extraction of key information from sales invoices.	• **Form Recognizer Studio** • **REST API** • **C# SDK**

		• **Python SDK**
Receipt model (updated)	Automated data processing and extraction of key information from sales receipts. Receipt model v3.0 supports processing of **single-page hotel receipts**.	• **Form Recognizer Studio** • **REST API** • **C# SDK** • **Python SDK**
ID document model (updated)	Automated data processing and extraction of key information from US driver's licenses and international passports. Prebuilt ID document API supports the **extraction of endorsements, restrictions, and vehicle classifications from US driver's licenses**.	• **Form Recognizer Studio** • **REST API** • **C# SDK** • **Python SDK**
Business card model	Automated data processing and extraction of key information from business cards.	• **Form Recognizer Studio** • **REST API** • **C# SDK** • **Python SDK**

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
NEW **General document (preview)**	Extract text, tables, structure, key-value pairs, and named entities.
Layout	Extracts text and layout information from documents.
Invoice	Extract key information from English invoices.
Receipt	Extract key information from English receipts.
ID document	Extract key information from US driver licenses and international passports.
Business card	Extract key information from English business cards.
Custom	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.

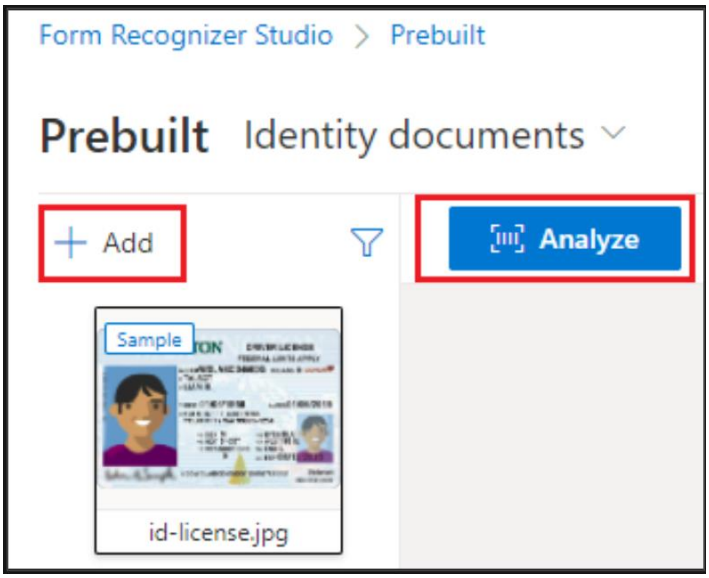
The following Form Recognizer service features are available in the Studio.

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

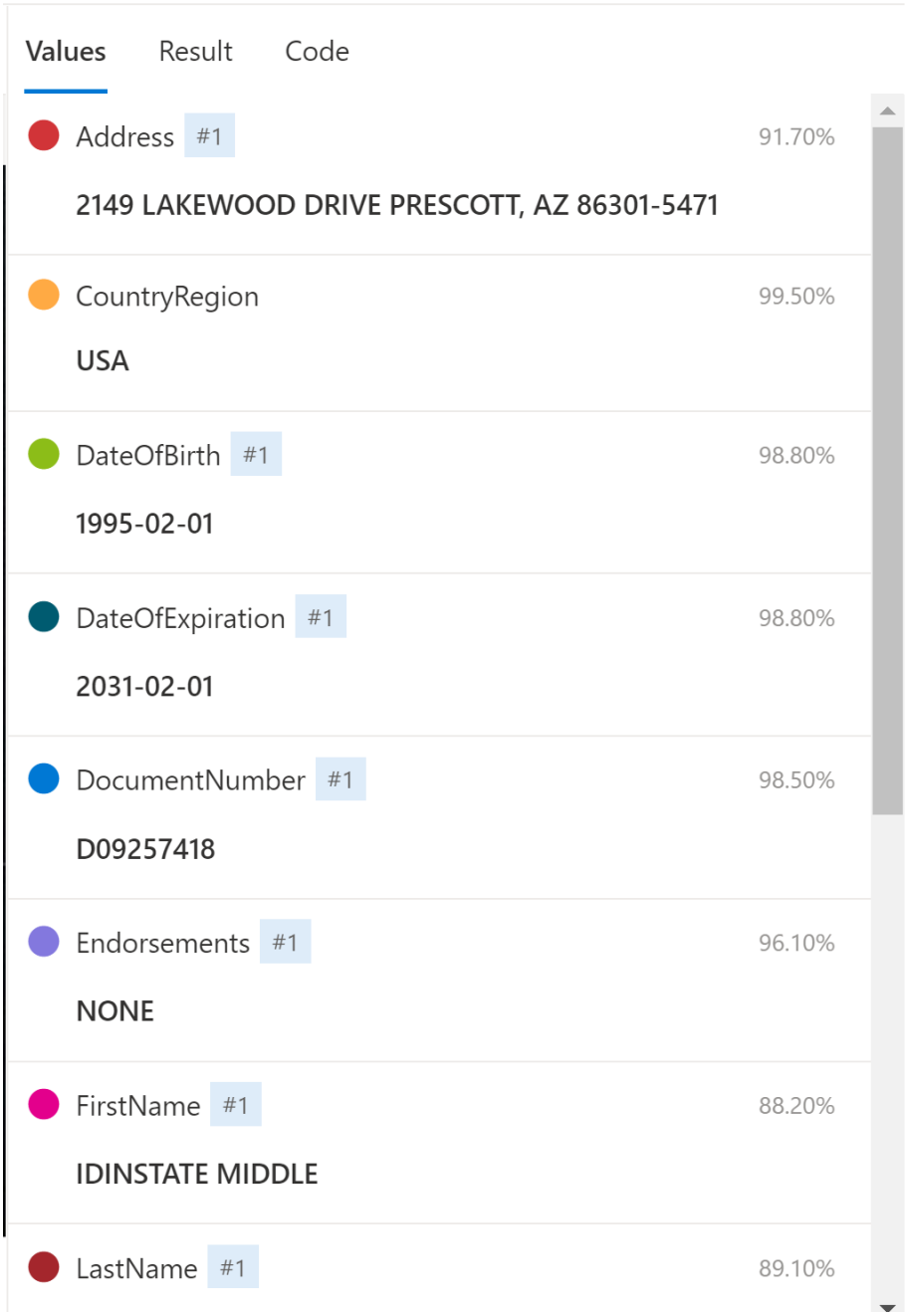
The ID document model combines powerful Optical Character Recognition (OCR) capabilities with deep learning models to analyze and extracts key information from U.S. Driver's Licenses (all 50 states and District of Columbia) and international passport biographical pages (excluding visa and other travel 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)
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:

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



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

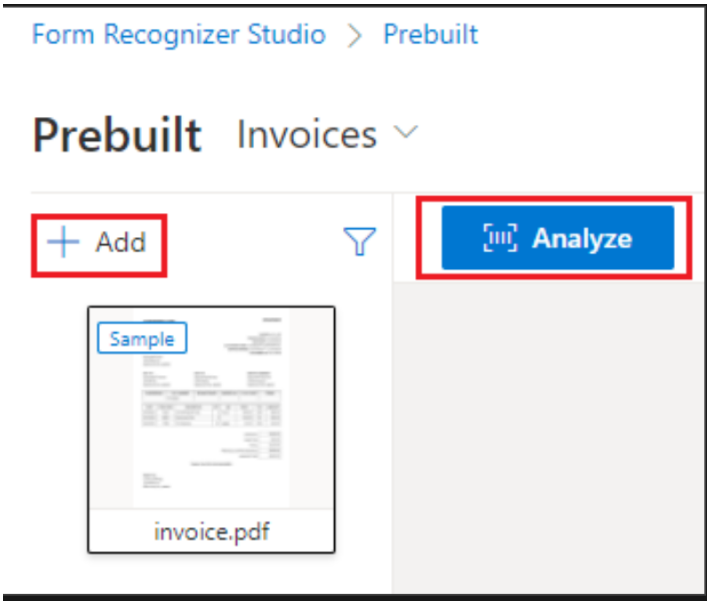
Name	Type	Description	Standardized output
NEW **Endorsements**	String	Additional driving privileges granted to a driver such as Motorcycle or School bus.	

NEW **Restrictions**	String	Restricted driving privileges applicable to suspended or revoked licenses.	
NEW **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<USABROOKS<<JENNIFER<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<3400200135USA8001014F1905054710000307<715816"
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)	

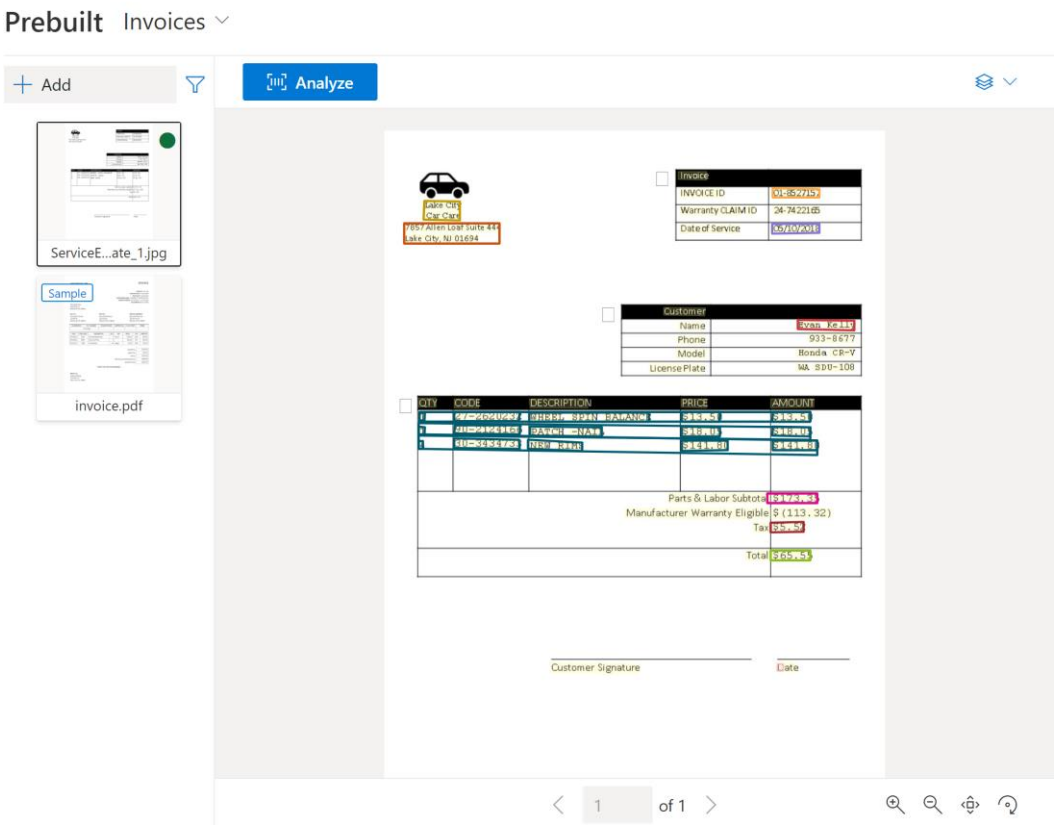
Invoice Model

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.

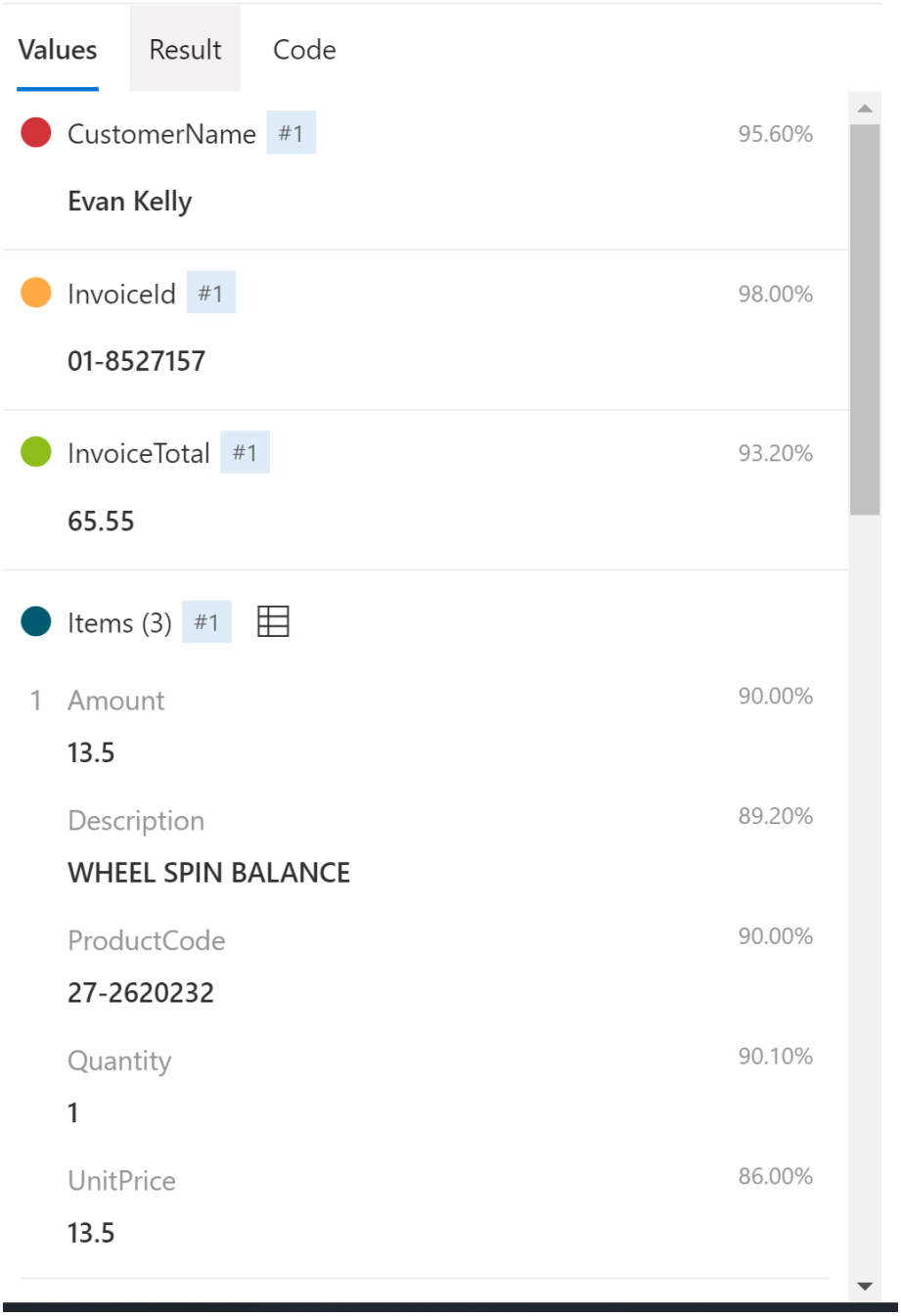
1. On the Form Recognizer Studio home page, select Invoices or click on link below - [FormRecognizerStudio (azure.com)]
2. You can analyze the sample invoice or select the + Add button to upload your own sample.



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



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



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	

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

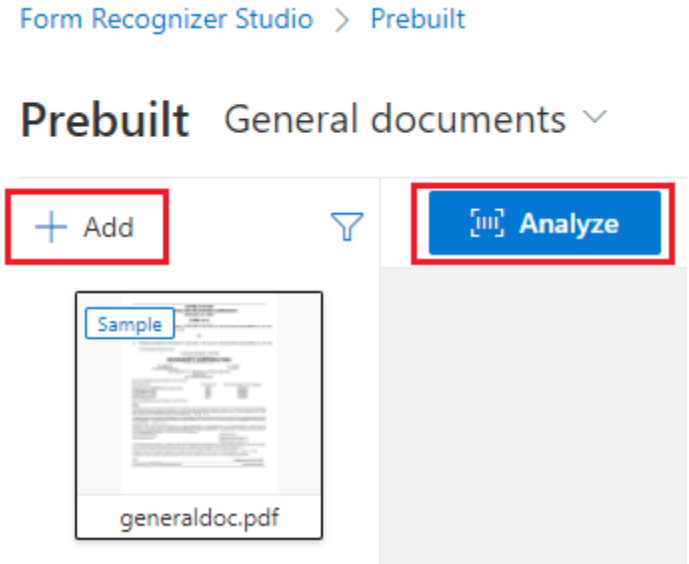
The General document preview model combines powerful Optical Character Recognition (OCR) capabilities with deep learning models to extract key-value pairs and entities from documents.

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.

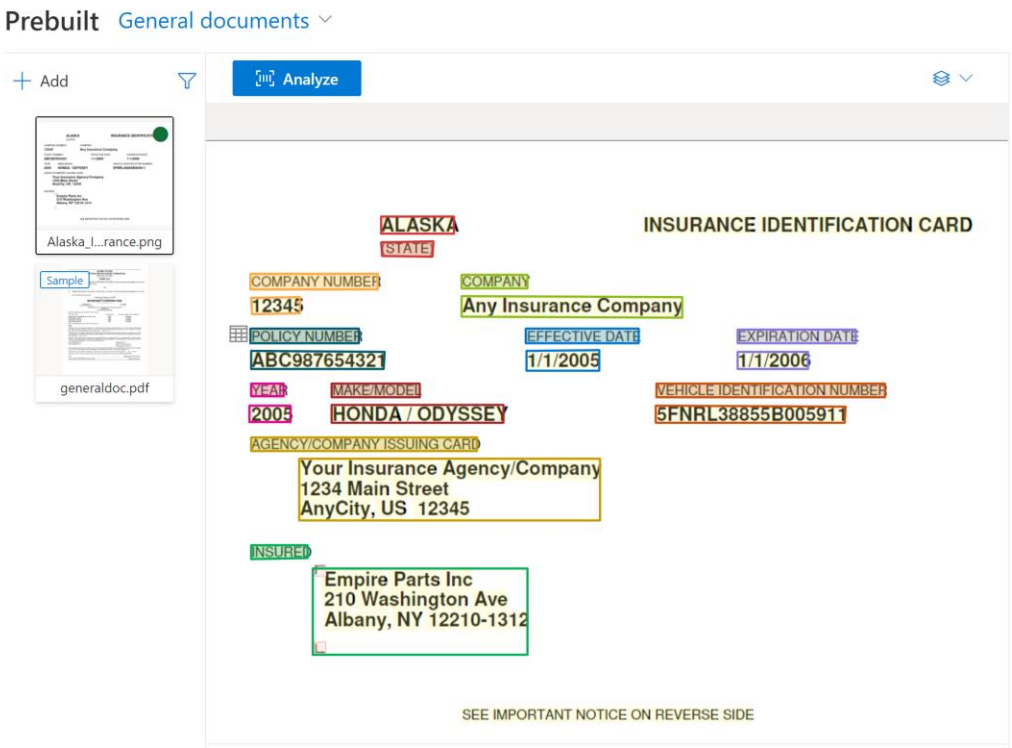
Features:

- 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.

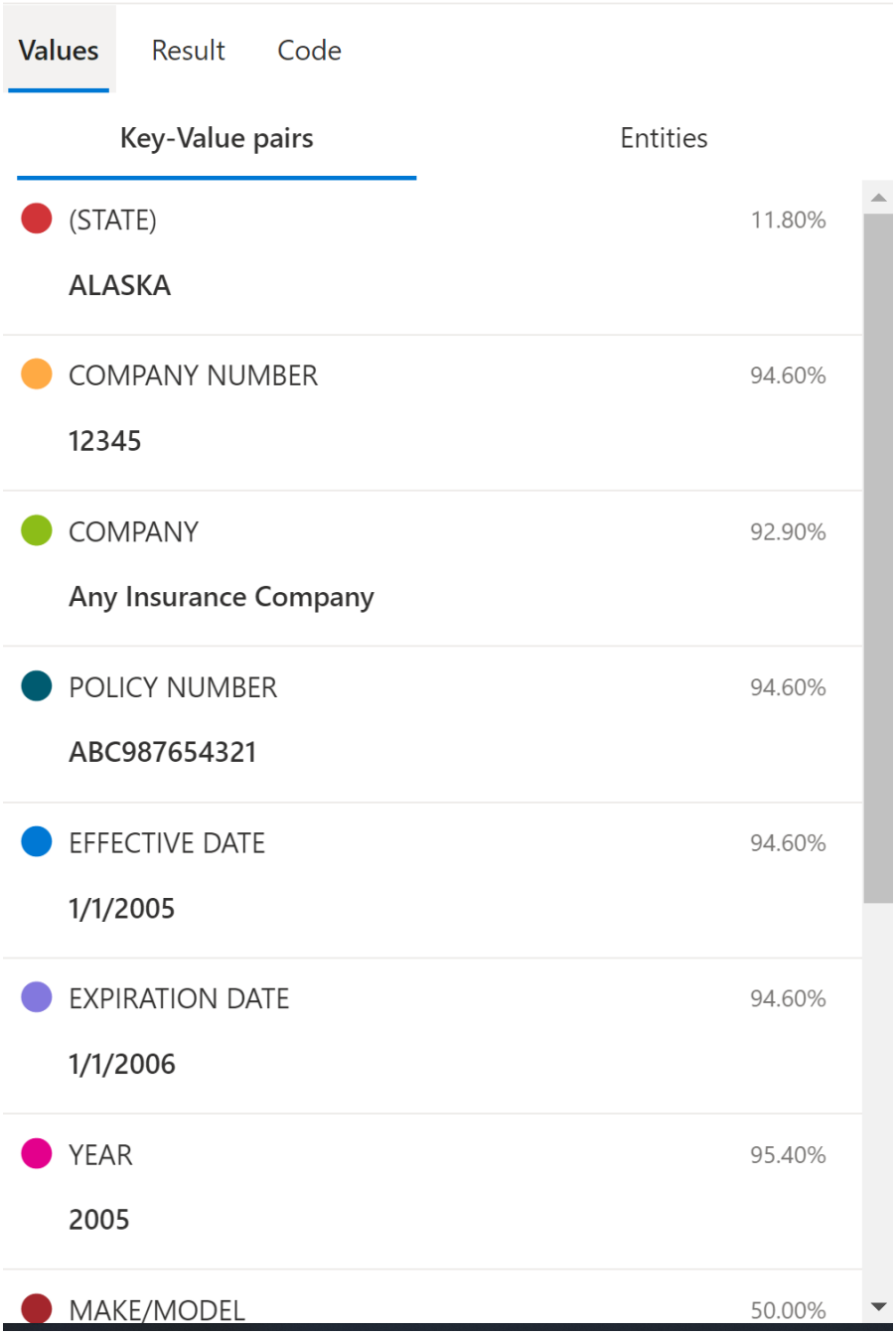
1. On the Form Recognizer Studio home page, select General Documents or click on link below - [FormRecognizerStudio (azure.com)]
2. You can analyze the sample document or select the + Add button to upload your own sample.



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



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



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

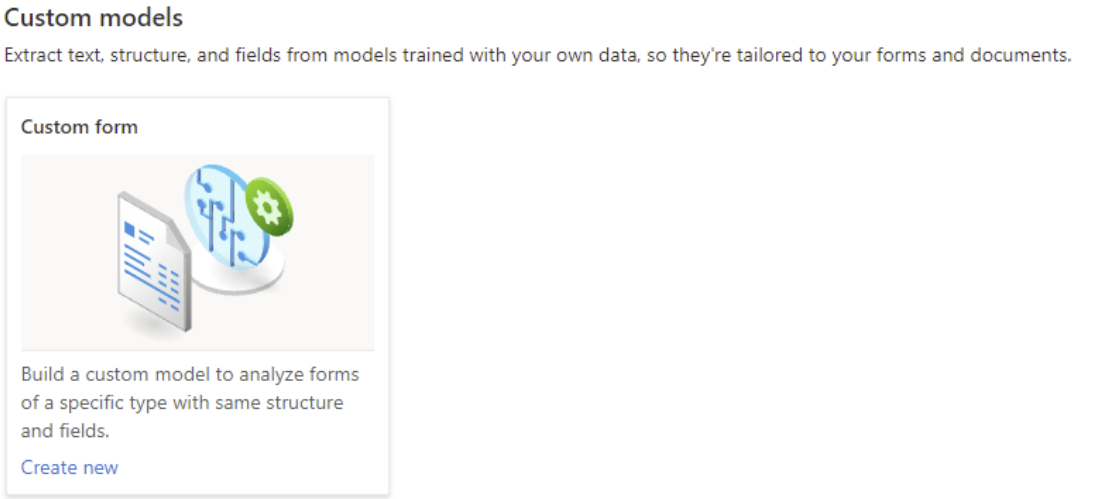
Custom Model

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

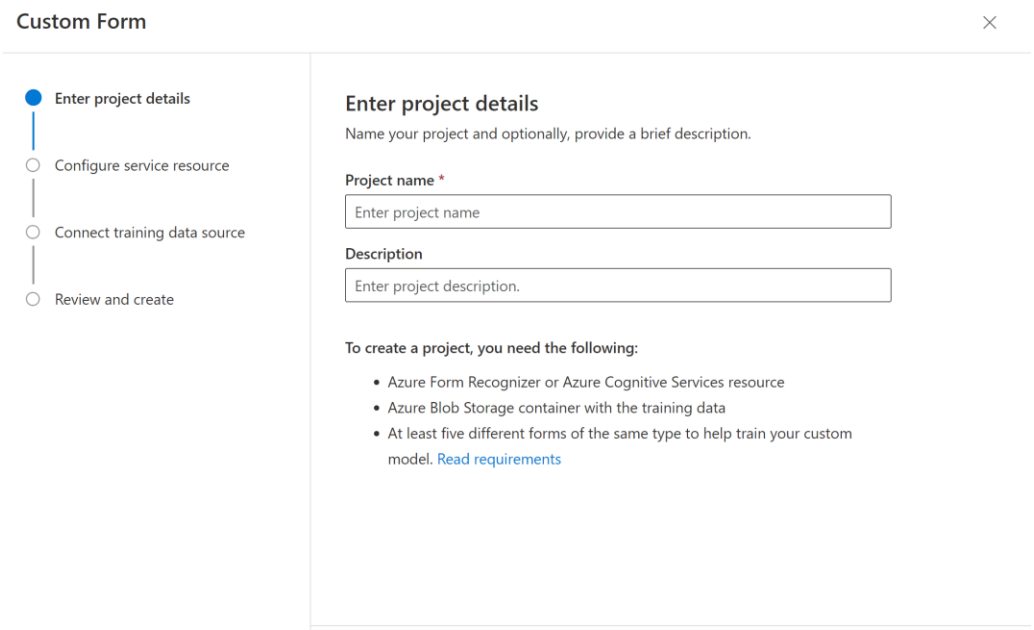
A custom model is a machine learning program trained to recognize form fields within your distinct content and extract key-value pairs and table data. You only need five examples of the same form type to get started and your custom model can be trained with or without labeled datasets.

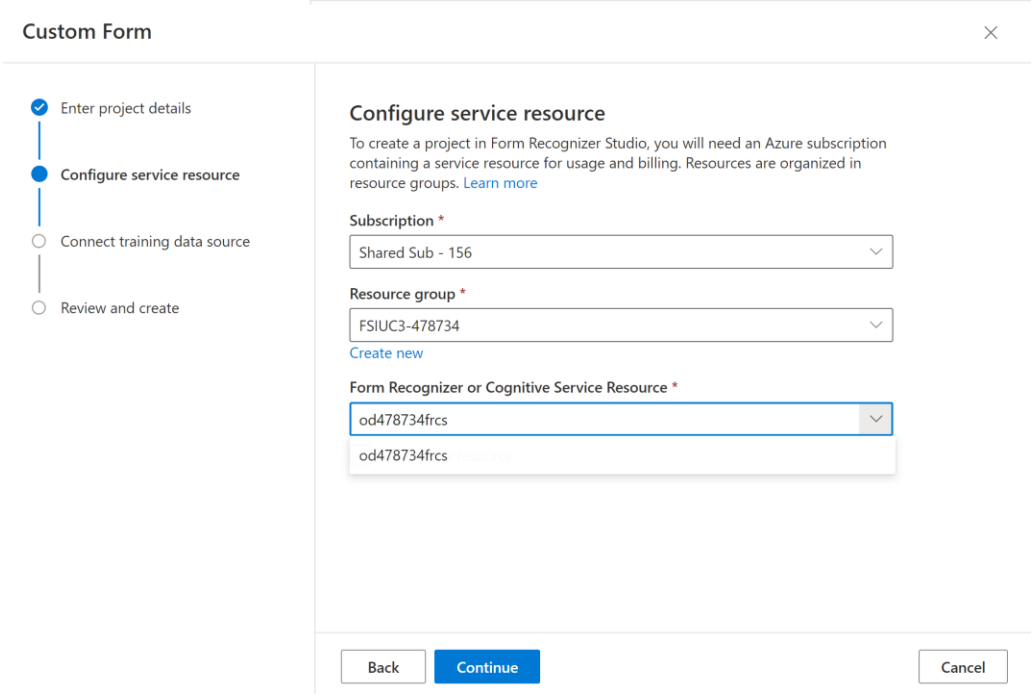
While we could have used the “General Document” Model to extract the data from Insurance card, for hackathon instead we will build the custom model for that.

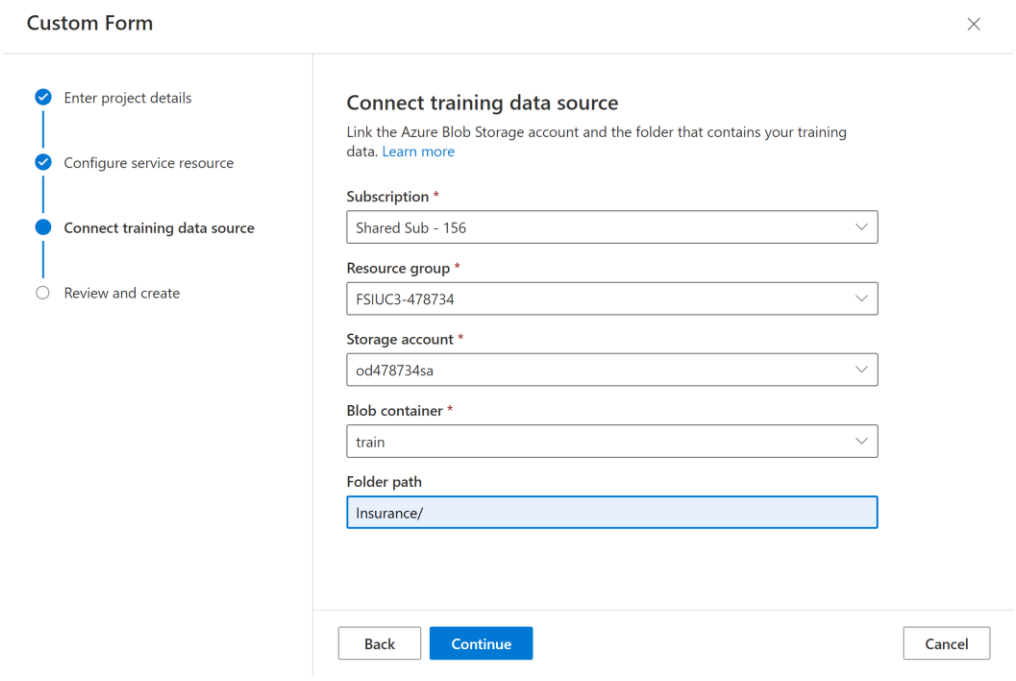
1. Navigate to the Form Recognizer Studio and select **Custom form** under Custom models

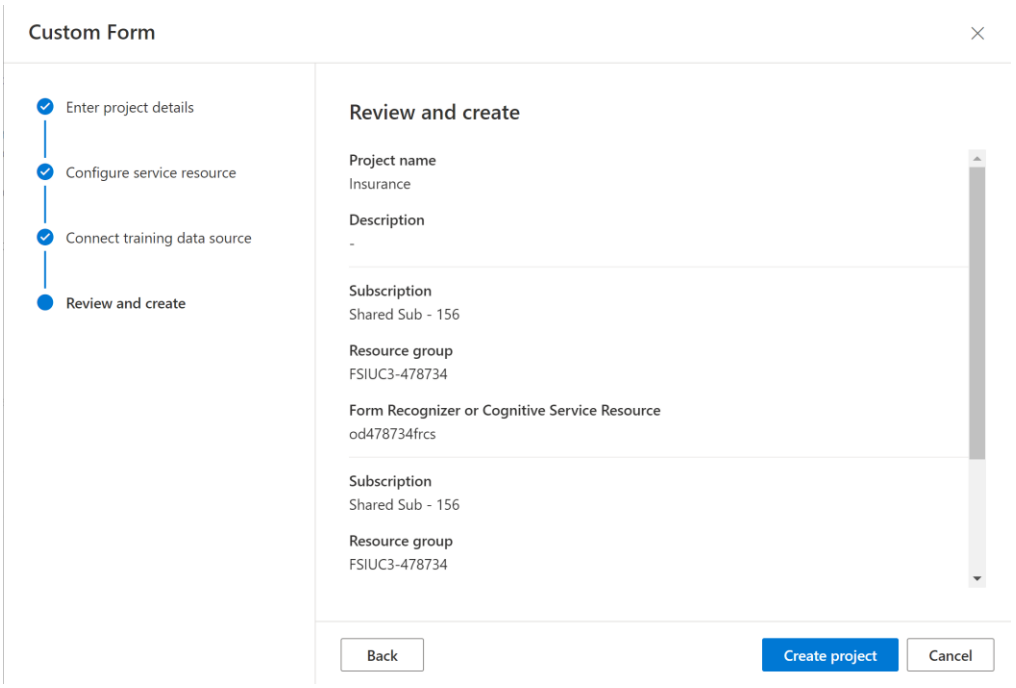


2. Follow the workflow to create a new project:

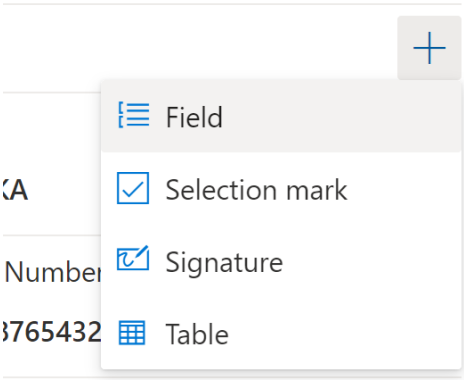

Custom Form
Enter project details
Configure service resource
Connect training data source
Review and create
Enter project details
Name your project and optionally, provide a brief description.
Project name *
Enter project name
Description
Enter project description.
To create a project, you need the following:
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


Custom Form
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
Back
Continue
Cancel

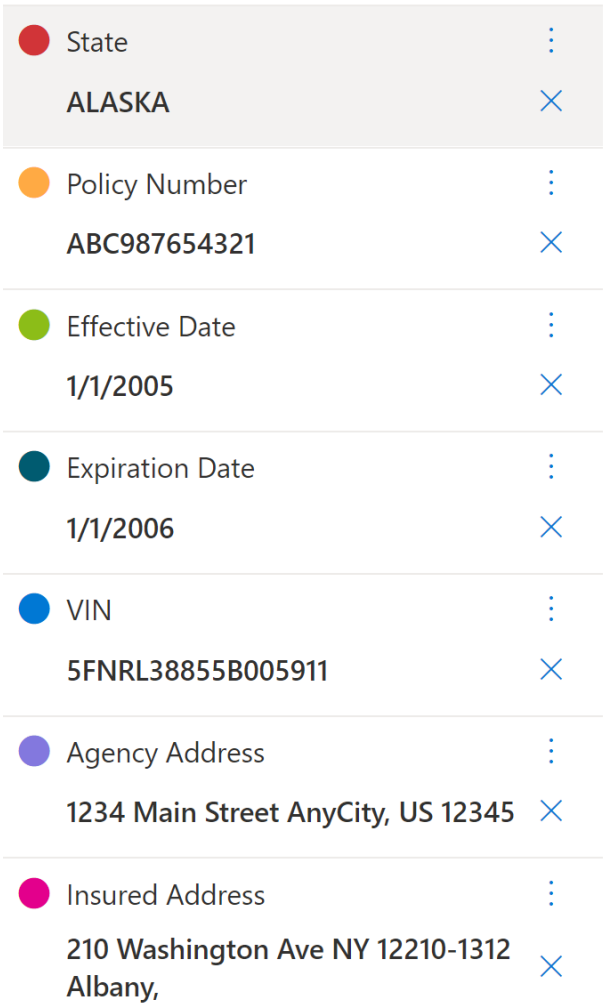




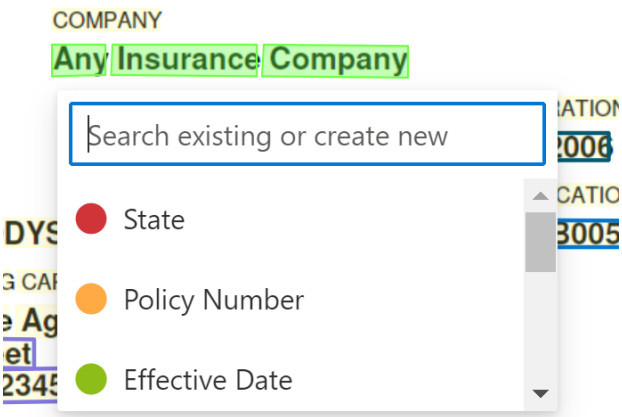
3. Select + Field to create custom tags/labels



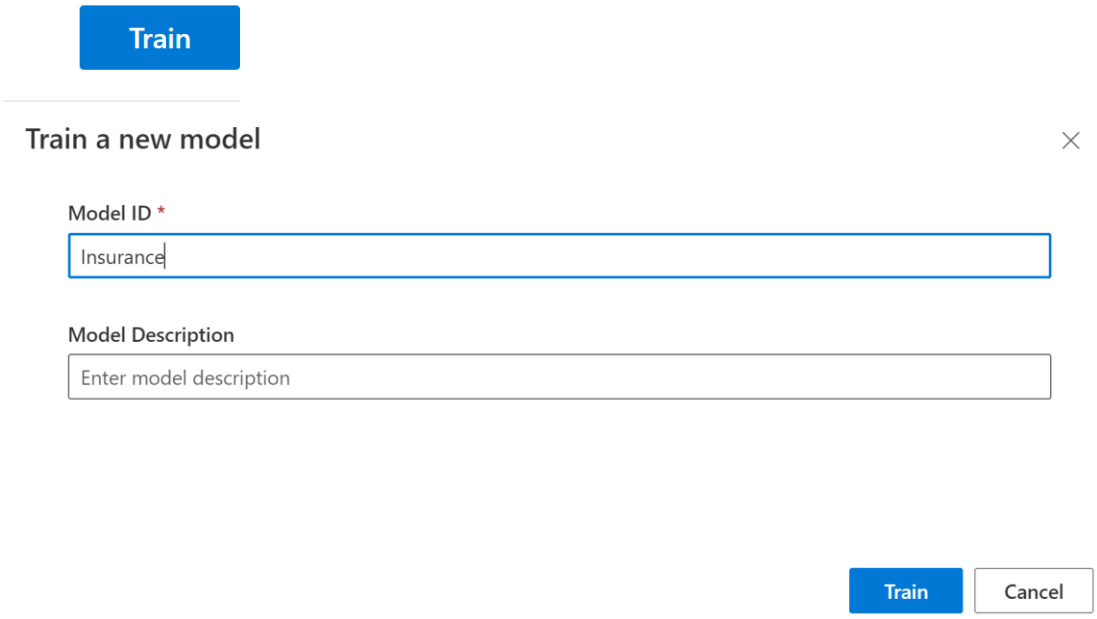
Following fields (and any additional you want) needs to be created:



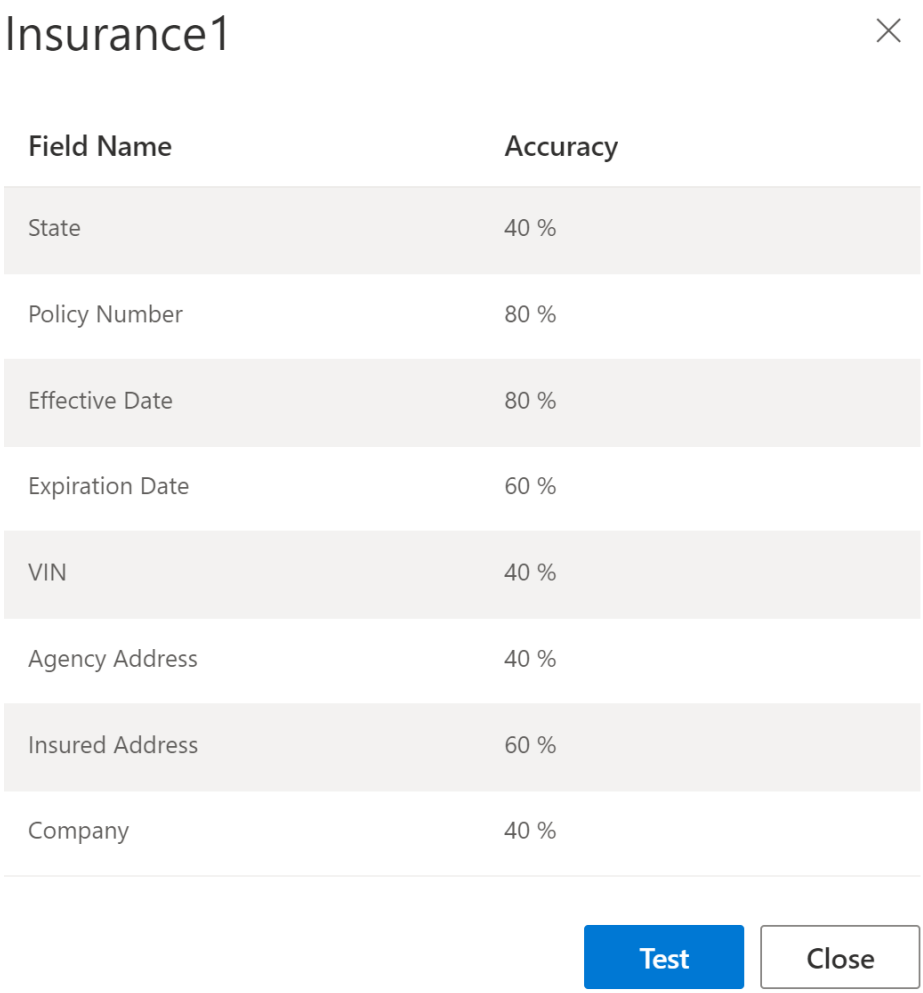
4. Select the OCR (yellow color) text to label the sample data. Once it's labelled it will show that as labelled tag. Repeat that for all the fields you created and all 5 sample data



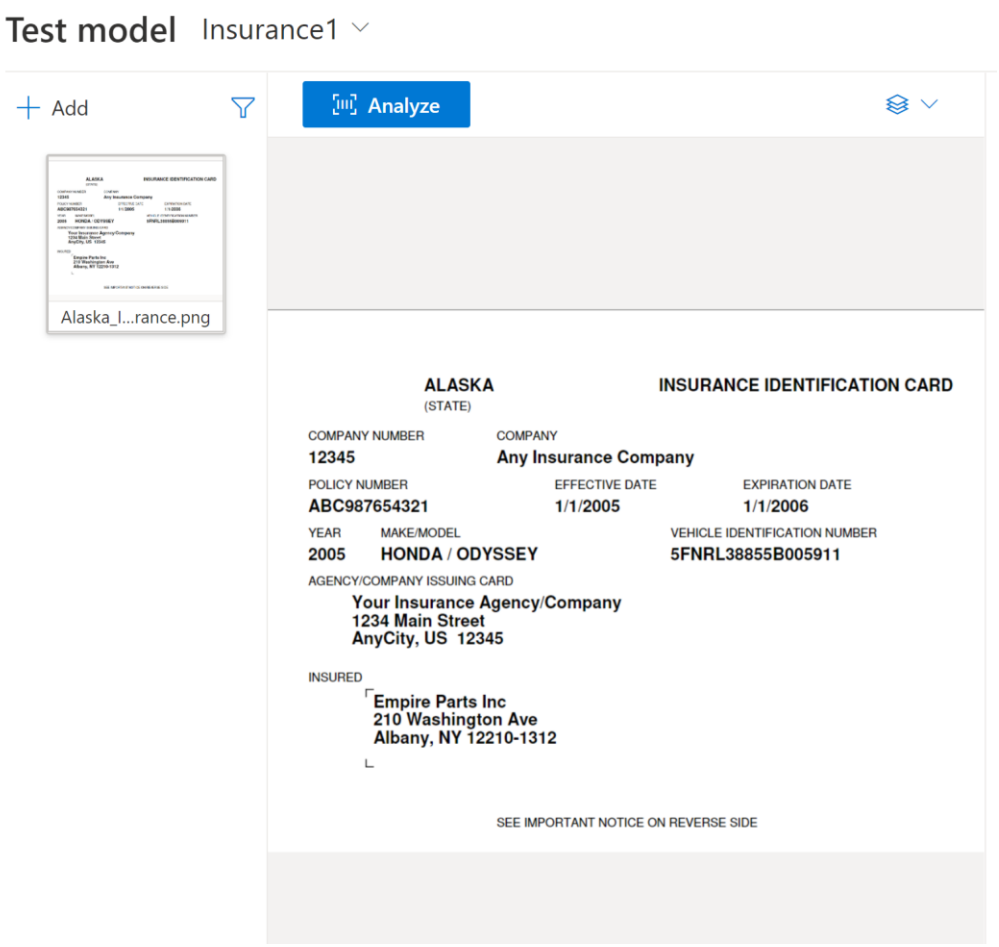
5. Once all the labelling is completed. Click on Train to build the model. Enter model name and click train



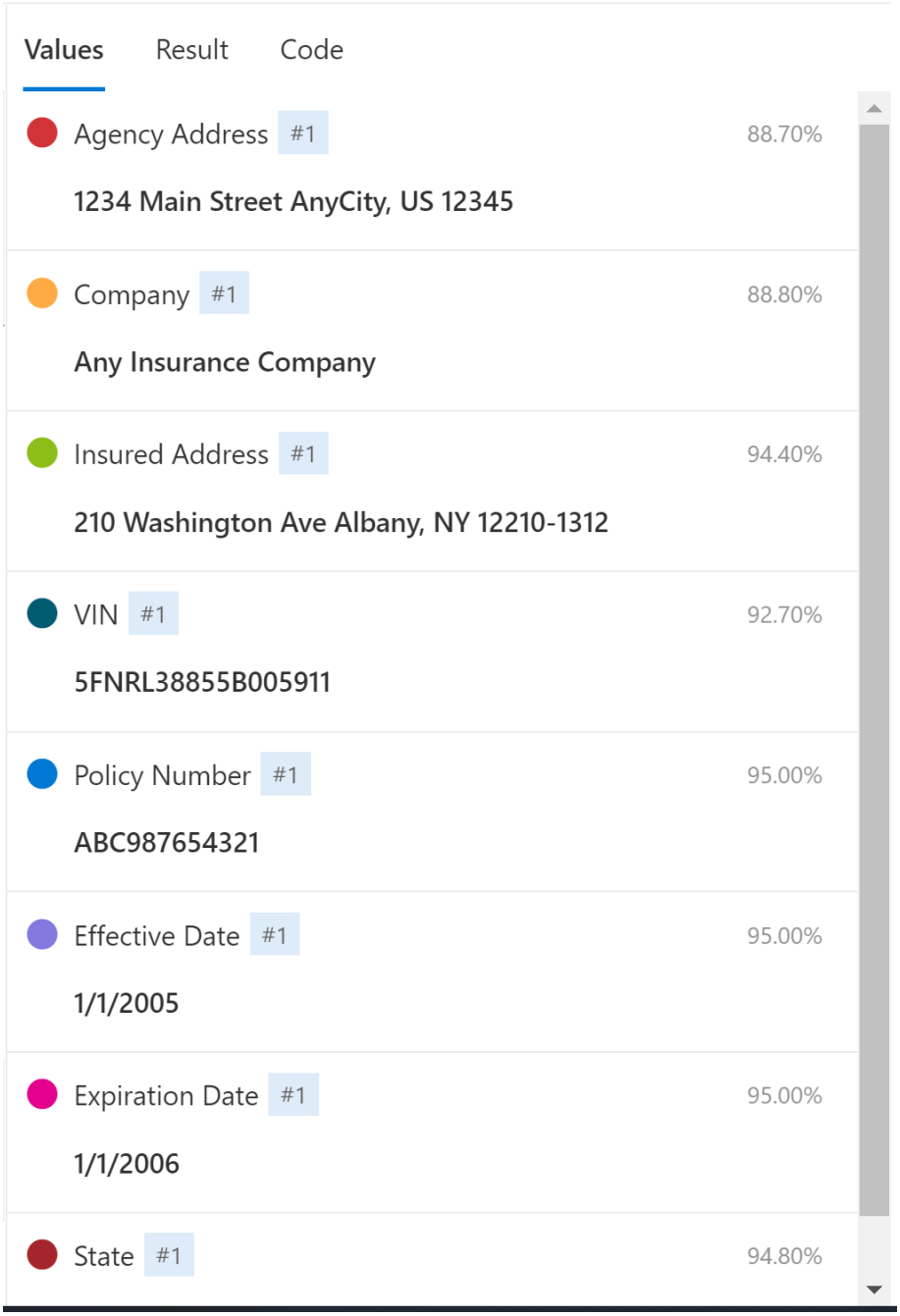
6. Upon successful completion of the model, it will show the accuracy for each field



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



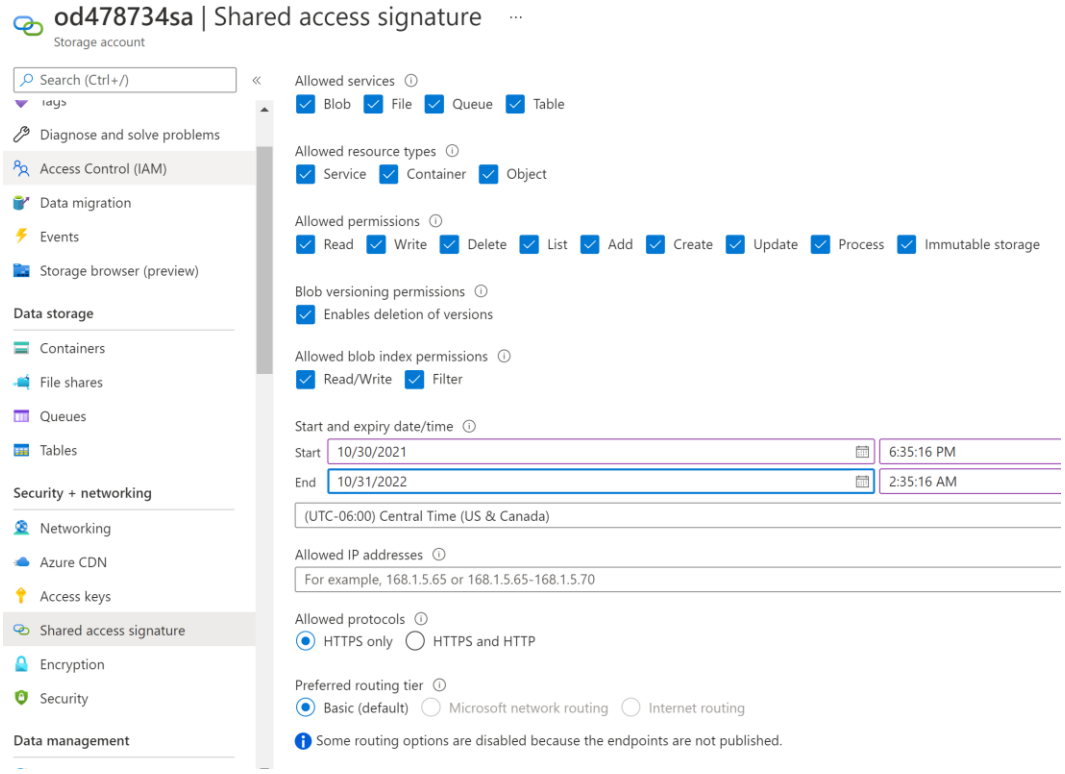
9. After completion of analysis, it will show you the custom entities/keys that are extracted.



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.

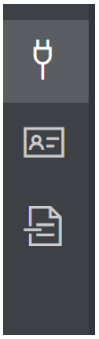
1. Go to Azure Portal (portal.azure.com). Login with your credentials. Go to the storage account within your resource group. Click on Shared Access signature



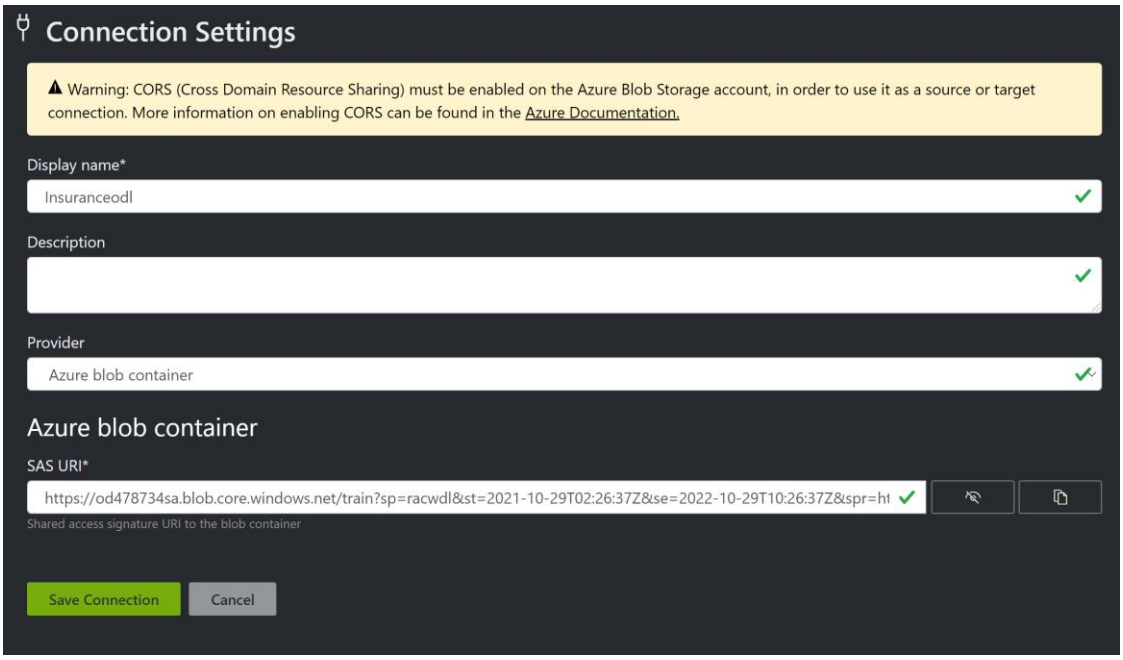
Ensure "Service, Container and Object" are selected in allowed resource type and the end date is a year from today. Click on Generate SAS String

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

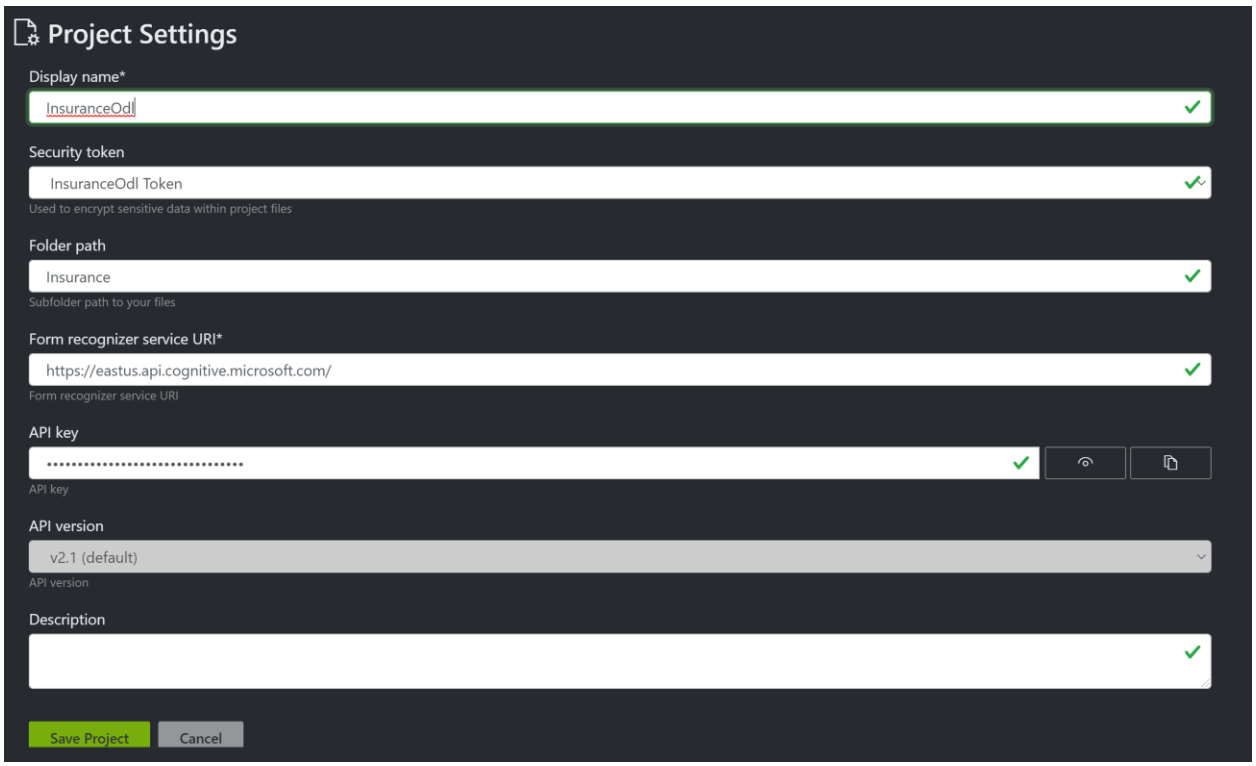
2. Go to FOTT labelling tool - [Analyze - Form OCR Testing Tool (fott-2-1.azurewebsites.net)]
3. Click on Connection to create a new connection where our train dataset is hosted.



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**

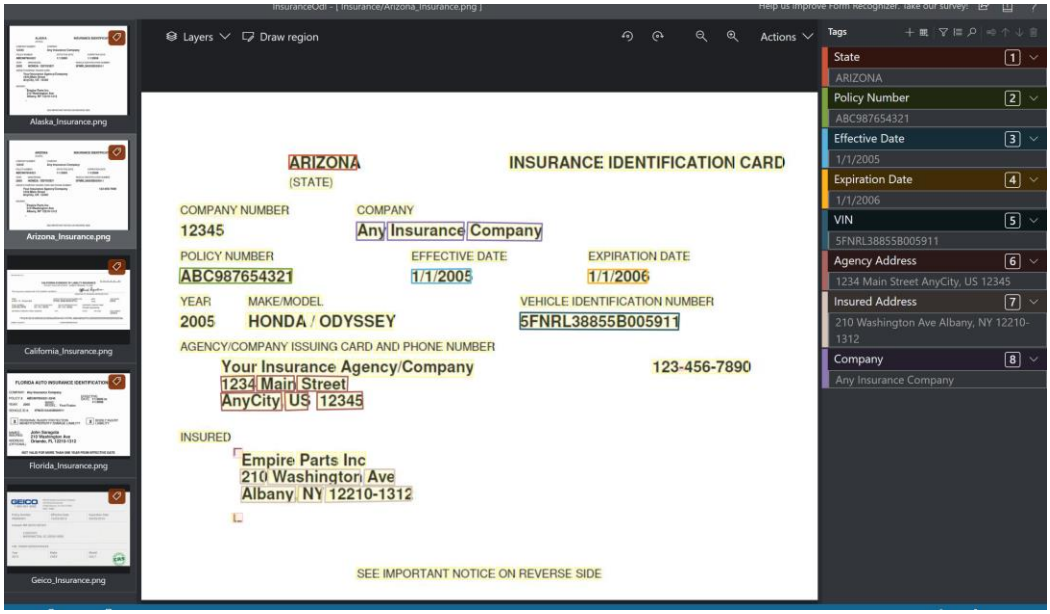


5. Save the connection
6. Go back to home page and click on New Project. Enter display name, select the security token, “Insurance” as Folder Path. You can get the Form recognizer service URI & API key from the Azure Portal -> Resource Group -> Form Recognizer service -> Keys & Endpoint

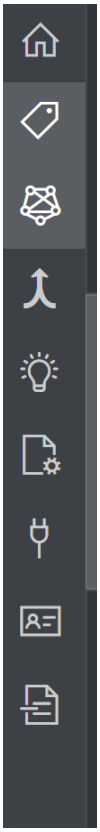


7. Save the Project

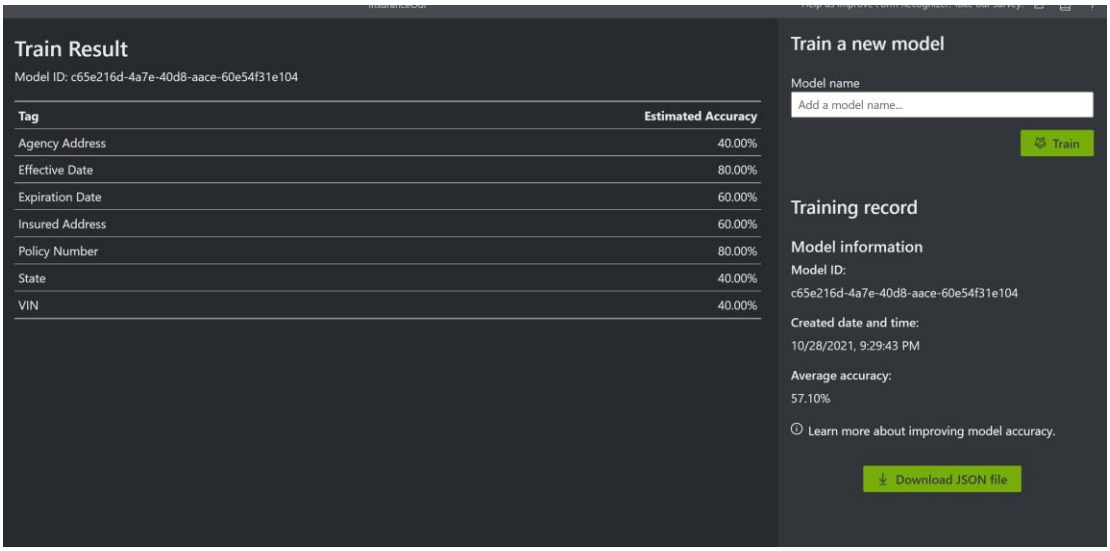
8. Since we already labelled the data in Form Recognizer Studio, you should see the labelled data and everything configured opening up the project.



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