

Azure AI 문서 인텔리전스 설명서

Azure AI 문서 인텔리전스(이전 Form Recognizer)는 기계 학습 모델을 사용하여 애플리케이션 및 워크플로에서 데이터 처리를 자동화하는 클라우드 기반 Azure AI 서비스입니다. 문서 인텔리전스는 데이터 기반 전략을 개선하고 문서 검색 기능을 보강하는 데 필수적입니다.

Azure AI 문서 인텔리전스 정보

개요

[Azure AI 문서 인텔리전스란?](#)

[문서 인텔리전스 FAQ](#)

새롭운 기능

[Azure AI 문서 인텔리전스의 새롭운 기능](#)

참조

 [SDK 개요\(미리 보기\) ``2023-10-31-preview``\(미리 보기\)](#)

[SDK 개요 ``2023-07-31`` 최신\(GA\)](#)

[SDK 개요 ``2022-08-31`` 레거시\(GA\)](#)

문서 인텔리전스 스튜디오

개요

[스튜디오 개요](#)

[스튜디오 개념](#)

문서 인텔리전스 모델

아키텍처

[읽음](#)

레이아웃

청구서

Receipt

ID 문서

의료 보험 카드

Contract

미국 세금 문서

문서 인텔리전스 개념

{☰} 개념

NEW 쿼리 필드 추출

NEW 추가 기능

NEW RAG(검색 증강 생성)

{☰} 개념

미리 학습된 모델 개요

정확도 및 신뢰도 점수

API 응답: 문서 분석

사용자 지정 모델 개요

빠른 시작 및 방법 가이드

💡 빠른 시작

Document Intelligence Studio

문서 인텔리전스 SDK

📘 방법 가이드

문서 인텔리전스 리소스 만들기

스토리지 컨테이너에 대한 SAS 토큰 만들기

관리 ID 생성 및 사용

사용자 지정 추출 모델 빌드

사용자 지정 분류 모델 빌드

사용자 지정 모델 작성

문서 인텔리전스 컨테이너 설치 및 실행

책임 있는 AI

💡 참조

투명성 메모

특성 및 제한 사항

통합 및 책임 있는 사용에 대한 지침

데이터 개인 정보 보호, 규정 준수 및 보안

Azure AI 문서 인텔리전스란?

아티클 • 2024. 03. 10.

① 중요

- 문서 인텔리전스 공개 미리 보기 릴리스에서는 현재 개발 중인 기능에 대한 조기 액세스를 제공합니다.
- 기능, 방식 및 프로세스는 GA(일반 공급) 전에 사용자 피드백에 따라 변경될 수 있습니다.
- Document Intelligence 클라이언트 라이브러리의 공개 미리 보기 버전은 기본적으로 REST API 버전 [2024-02-29-preview로 설정됩니다](#).
- 공개 미리 보기 버전 [2024-02-29-preview](#) 는 현재 다음 Azure 지역에서만 사용할 수 있습니다.
 - 미국 동부
 - 미국 서부2
 - 서유럽

이 콘텐츠의 적용 대상은 다음과 같습니다. v4.0(미리 보기) | 이전 버전: v3.1(GA) v3.0(GA) v2.1(GA)

① 참고

Form Recognizer는 이제 Azure AI 문서 인텔리전스입니다!

- 2023년 7월부터 Azure AI 서비스는 이전에 Cognitive Services 및 Azure Applied AI Services로 알려진 모든 것을 포함합니다.
- 가격 책정에는 변화가 없습니다.
- Cognitive Services* 및 *Azure Applied AI*라는 이름은 Azure 청구, 비용 분석, 가격 목록 및 가격 API에서 계속 사용됩니다.
- v3.1을 포함하여 그 이전 API(애플리케이션 프로그래밍 인터페이스) 또는 SDK에 대한 주요 변경 내용은 없습니다. v4.0부터 API와 SDK가 문서 인텔리전스로 업데이트됩니다.
- 일부 플랫폼은 여전히 이름 변경 업데이트를 기다리고 있습니다. 설명서에서 Form Recognizer 또는 문서 인텔리전스에 대한 모든 언급은 동일한 Azure 서비스를 나타냅니다.

Azure AI 문서 인텔리전스는 지능형 문서 처리 솔루션을 빌드할 수 있는 클라우드 기반 [Azure AI 서비스](#)입니다. 다양한 데이터 형식에 걸친 방대한 양의 데이터가 양식과 문서에 저장됩니다. 문서 인텔리전스를 사용하면 데이터가 수집 및 처리되는 속도를 효과적으로 관리할 수 있으며 향상된 운영, 정보에 입각한 데이터 기반 의사 결정 및 인식 가능한 혁신의 핵심입니다.

|  **문서 분석 모델** |  **미리 빌드된 모델** |  **사용자 지정 모델** |

문서 분석 모델

문서 분석 모델을 사용하면 양식 및 문서에서 텍스트를 추출하고 조직의 작업, 사용 또는 진행 상황에 맞는 구조화된 비즈니스 지원 콘텐츠를 반환할 수 있습니다.



읽기 | 인쇄 및
필기 텍스트를 추출합니다.



레이아웃 | 문자, 테이블,
및 문서 구조를 추출합니다.

미리 빌드된 모델

미리 빌드된 모델을 사용하면 자체 모델을 학습하고 빌드할 필요 없이 앱 및 흐름에 지능형 문서 처리를 추가할 수 있습니다.



청구서 | 고객 및
공급업체 세부 정보를 추출합니다.



영수증 | 판매 거래
세부 정보를 추출합니다.



식별 | 식별 및
확인 세부 정보를 추출합니다.



1003 EULA | 모기지 세부 정보를 추출합니다.



양식 1008 | 모기지 세부 정보를 추출합니다.



공개 닫기 | 모기지 세부 정보를 추출합니다.



건강 보험 카드 | 건강 보험
세부 정보를 추출합니다.



계약 | 계약 및
당사자 세부 정보를 추출합니다.



신용/직불 카드 | 은행 카드 정보를 추출합니다.



결혼 인증서 | 결혼 인증서에서 정보를 추출합니다.



미국 세금 W-2 양식 | 과세 보상
세부 정보를 추출합니다.



미국 세금 1098 양식 | 모기지 이자 세부 정보를 추출합니다.



미국 세금 1098-E 양식 | 학자금 대출 이자 세부 정보를 추출합니다.



미국 세금 1098-T 양식 | 정규화된 수업료 세부 정보를 추출합니다.



미국 세금 1099 양식 | 1099 양식의 변형에서 정보를 추출합니다.



미국 세금 1040 양식 | 1040 양식의 변형에서 정보를 추출합니다.

사용자 지정 모델

- 사용자 지정 모델은 레이블이 지정된 데이터 세트를 사용하여 사용 사례와 관련된 양식 및 문서에서 고유한 데이터를 추출하도록 학습됩니다.
- 독립 실행형 사용자 지정 모델을 결합하여 작성된 모델을 만들 수 있습니다.
- 추출 모델**
 - ✓ 사용자 지정 추출 모델은 문서에서 레이블이 지정된 필드를 추출하도록 학습됩니다.



사용자 지정 템플릿 | 정적 레이아웃에서 데이터를 추출합니다.



사용자 지정 인공신경망 | 혼합 형식 문서에서 데이터를 추출합니다.



사용자 지정 구성형 | 모델 컬렉션을 사용하여 데이터를 추출합니다.

- **분류 모델**

✓ 사용자 지정 분류자는 추출 모델을 호출하기 전에 문서 형식을 식별합니다.



사용자 지정 분류자 | 추출 모델을 호출하기 전에 지정된 문서 형식(클래스)을 식별합니다.

추가 기능

문서 인텔리전스는 문서 추출 시나리오에 따라 사용하고 사용하지 않을 수 있는 선택적 기능을 지원합니다. 다음 추가 기능은 `2023-07-31 (GA)` 이상 릴리스에 사용할 수 있습니다.

- [ocr.highResolution](#)
- [ocr.formula](#)
- [ocr.font](#)
- [ocr.barcode](#)

문서 인텔리전스는 문서 추출 시나리오에 따라 사용하고 사용하지 않을 수 있는 선택적 기능을 지원합니다. 다음 추가 기능 기능은 이후 릴리스에 `2024-02-29-preview` `2023-10-31-preview` 사용할 수 있습니다.

- [queryFields](#)

분석 기능

테이블 확장

Model ID	콘텐츠 추출	쿼리 필드	단락 역할	단락 표시	선택 항목	테이블	키/값 쌍	언어 코드	바코드 분석	문서 분석	수식 *	스타일 꼴	고해상도 *
prebuilt-read		✓					○	○		○	○	○	
prebuilt-layout		✓	✓	✓	✓	✓	✓		○	○	○	○	○
prebuilt-document		✓	✓	✓	✓	✓	✓	✓	○	○	○	○	○
prebuilt-businessCard		✓	✓								✓		
prebuilt-contract		✓	✓	✓	✓			○	○	✓	○	○	○
prebuilt-healthInsuranceCard.us		✓	✓						○	○	✓	○	○
prebuilt-idDocument		✓	✓						○	○	✓	○	○
prebuilt-invoice		✓	✓				✓	✓	○	○	○	✓	○
prebuilt-receipt		✓	✓						○	○	✓	○	○
prebuilt-marriageCertificate.us		✓	✓						○	○	✓	○	○
prebuilt-creditCard		✓	✓						○	○	✓	○	○
prebuilt-mortgage.us.1003		✓	✓						○	○	✓	○	○
prebuilt-mortgage.us.1008		✓	✓						○	○	✓	○	○
prebuilt-mortgage.us.closingDisclosure		✓	✓						○	○	✓	○	○
prebuilt-tax.us.w2		✓	✓			✓			○	○	✓	○	○
prebuilt-tax.us.1098		✓	✓			✓			○	○	✓	○	○
prebuilt-tax.us.1098E		✓	✓			✓			○	○	✓	○	○
prebuilt-tax.us.1098T		✓	✓			✓			○	○	✓	○	○
prebuilt-tax.us.1099(변형)		✓	✓			✓			○	○	✓	○	○
prebuilt-tax.us.1040(변형)		✓	✓						○	○	✓	○	○
{ customModelName }		✓	✓	✓	✓	✓	✓	✓	○	○	✓	○	○

- ✓ - 사용함
- - 선택적
- * - 프리미엄 기능에는 추가 비용이 발생합니다

모델 및 개발 옵션

① 참고

다음 문서 이해 모델 및 개발 옵션은 문서 인텔리전스 서비스 v3.0에서 지원됩니다.

문서 인텔리전스를 사용하여 애플리케이션 및 워크플로에서 문서 처리를 자동화하고, 데이터 기반 전략을 개선하고, 문서 검색 기능을 보강할 수 있습니다. 이 표의 링크를 사용하여 각 모델에 대해 자세히 알아보고 개발 옵션을 찾아보세요.

읽음

The screenshot shows the Microsoft AI Analyze interface with the "Content" tab selected. The main pane displays a snippet of text from an article by Tom Lawry, National Director for AI, Health and Life Sciences at Microsoft. The text discusses the early stages of AI adoption in healthcare, mentioning pharmaceutical and life sciences organizations making major investments in AI and related technologies. Below the text, there are several callout boxes with annotations:

- While healthcare is still in the early stages of its AI journey, we are seeing pharmaceutical and other life sciences organizations making major investments in AI and related technologies.** (TOM LAWRY | National Director for AI, Health and Life Sciences | Microsoft)
- At pharmaceutical and other life sciences organizations invest in and deploy advanced technologies, they are beginning to see benefits in diverse areas across their organizations. Companies are looking to incorporate automation and continuing smart factory concepts into their drug discovery, research and development, and manufacturing and supply chain management. Many life sciences organizations are also choosing to stay with more virtual approaches in the “new normal – particularly in clinical trials and sales and marketing areas.”**
- Enhancing the patient and provider experience**
- Clinical trial sponsors are continually seeking to make clinical trials faster and to improve the experience for patients and physicians. The COVID-19 pandemic has accelerated the adoption of decentralized clinical trials, with an increasing trend away from in-person visits to remote sites at home. In a McKinsey survey up to 98 percent of patients reported satisfaction with telemedicine. In the same report, 72 percent of physicians surveyed reported similar or better experiences with**

The right pane shows the JSON representation of the analyzed content, specifically for the highlighted quote. The JSON structure includes fields like 'text', 'pageNumber', 'angle', 'width', 'height', 'unit', and 'words'.

```

{
  "text": "While healthcare is still in the early stages of its AI journey, we are seeing pharmaceutical and other life sciences organizations making major investments in AI and related technologies.", "pageNumber": 1, "angle": 0, "width": 915, "height": 1190, "unit": "pixel", "words": [
    {
      "text": "ed", "startPage": 1, "endPage": 1, "startAngle": 0, "endAngle": 0, "startWidth": 10, "endWidth": 10, "startHeight": 10, "endHeight": 10, "order": 1
    },
    {
      "text": "2023-02-21T19:27:23Z", "startPage": 1, "endPage": 1, "startAngle": 0, "endAngle": 0, "startWidth": 10, "endWidth": 10, "startHeight": 10, "endHeight": 10, "order": 2
    },
    {
      "text": "re", "startPage": 1, "endPage": 1, "startAngle": 0, "endAngle": 0, "startWidth": 10, "endWidth": 10, "startHeight": 10, "endHeight": 10, "order": 3
    },
    {
      "text": "2023-02-21T19:27:25Z", "startPage": 1, "endPage": 1, "startAngle": 0, "endAngle": 0, "startWidth": 10, "endWidth": 10, "startHeight": 10, "endHeight": 10, "order": 4
    },
    {
      "text": "2022-08-31", "startPage": 1, "endPage": 1, "startAngle": 0, "endAngle": 0, "startWidth": 10, "endWidth": 10, "startHeight": 10, "endHeight": 10, "order": 5
    },
    {
      "text": "uilt-read", "startPage": 1, "endPage": 1, "startAngle": 0, "endAngle": 0, "startWidth": 10, "endWidth": 10, "startHeight": 10, "endHeight": 10, "order": 6
    },
    {
      "text": ":", "startPage": 1, "endPage": 1, "startAngle": 0, "endAngle": 0, "startWidth": 10, "endWidth": 10, "startHeight": 10, "endHeight": 10, "order": 7
    },
    {
      "text": "utf16CodeUnit", "startPage": 1, "endPage": 1, "startAngle": 0, "endAngle": 0, "startWidth": 10, "endWidth": 10, "startHeight": 10, "endHeight": 10, "order": 8
    },
    {
      "text": "e healthcare is still in the early stages of its A", "startPage": 1, "endPage": 1, "startAngle": 0, "endAngle": 0, "startWidth": 10, "endWidth": 10, "startHeight": 10, "endHeight": 10, "order": 9
    }
  ]
}

```

데이터 확장

Model ID	설명	자동화 사용 사례	개발 옵션
prebuilt-read	<ul style="list-style-type: none"> 문서에서 텍스트를 추출합니다. 데이터 추출 	<ul style="list-style-type: none"> 모든 문서를 디지털화합니다. 규정 준수 및 감사. 번역하기 전에 필기 노트를 처리합니다. 	<ul style="list-style-type: none"> 문서 인텔리전스 스튜디오 REST API C# SDK Python SDK Java SDK JavaScript

모델 형식으로 돌아가기

레이아웃

Analyze | All pages | Range

NEWS TODAY
Latest news and bulletin updates

Role: Content
Content: NEWS TODAY Latest news and bulletin updates
Polygon: 139, 9, 608, 8, 608, 89, 139, 90

title: "NEWS TODAY Latest news and bulletin updates"
stateTime: "2023-02-21T19:39:32Z",
tedDateTime: "2023-02-21T19:39:34Z",
"succeeded":

```

5   "analyzeResult": {
6     "apiVersion": "2022-08-31",
7     "modelId": "prebuilt-layout",
8     "stringIndexType": "utf16CodeUnit",
9     "content": "Tuesday, Sep 20, YYYY\nNEWS TODAY Latest new
10    "pages": [
11      {
12        "pageNumber": 1,
13        "angle": 0,
14        "width": 756,
15        "height": 1066,
16        "unit": "pixel",
17        "words": [

```

테이블 확장

Model ID	설명	자동화 사용 사례	개발 옵션
prebuilt-layout	<ul style="list-style-type: none"> 문서에서 텍스트 및 레이아웃 정보를 추출합니다. 데이터 추출 	<ul style="list-style-type: none"> 구조별 문서 인덱싱 및 검색. 재무 및 의료 보고서 분석. 	<ul style="list-style-type: none"> 문서 인텔리전스 스튜디오 REST API C# SDK Python SDK Java SDK JavaScript

모델 형식으로 돌아가기

송장

Analyze | All pages | Range

INVOICE

Fields: Prebuilt invoice

Key-Value pairs:

- INVOICE: INV-100
- INVOICE DATE: 11/15/2019
- DUE DATE: 12/15/2019
- CUSTOMER NAME: MICROSOFT CORPORATION
- SERVICE PERIOD: 10/14/2019 – 11/14/2019
- CUSTOMER ID: CID-12345

Content: Prebuilt invoice

Result: Key-Value pairs

Code: Prebuilt invoice

테이블 확장

Model ID	설명	자동화 사용 사례	개발 옵션
prebuilt-invoice	<ul style="list-style-type: none"> 청구서에서 주요 정보를 추출합니다. 데이터 및 필드 추출 	<ul style="list-style-type: none"> 미지불 처리. 자동 세금 기록 및 보고. 	<ul style="list-style-type: none"> 문서 인텔리전스 스튜디오 REST API C# SDK Python SDK Java SDK JavaScript

[모델 형식으로 돌아가기](#)

Receipt

The screenshot shows a receipt document from Contoso Hotel. The document details a stay for Alex Morgan at Room 515, with a rate of \$127.00 per night. It includes a breakdown of charges and credits, with a total amount of 104.92 USD. To the right, an analysis pane displays the following fields and their confidence scores:

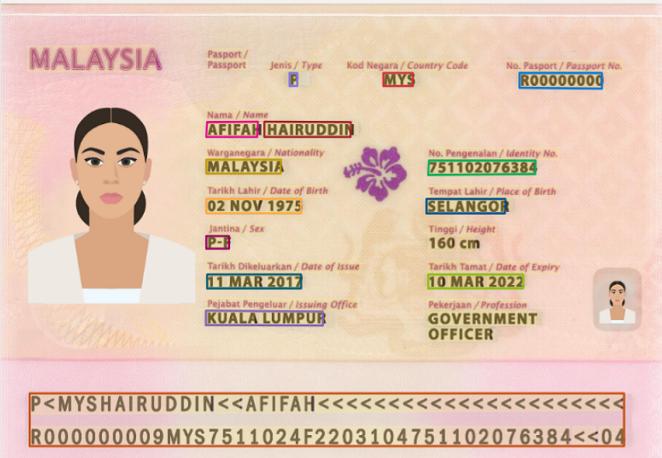
Field	Result	Code	Confidence
DocType	receipt.hotel		
ArrivalDate	#1		99.40%
	2021-03-27		
Currency	USD		99.50%
DepartureDate	#1		99.30%
	2021-03-28		
Items	(6) #1		98.70%
MerchantAddress	5600 148th Ave NE, Redmond, WA 98052		
	HouseNumber		
	5600		

[\[+\] 테이블 확장](#)

Model ID	설명	자동화 사용 사례	개발 옵션
prebuilt-receipt	<ul style="list-style-type: none"> 영수증에서 주요 정보를 추출합니다. 데이터 및 필드 추출 영수증 모델 v3.0은 단일 페이지 호텔 영수증 처리를 지원합니다. 	<ul style="list-style-type: none"> 비용 관리. 소비자 행동 데이터 분석. 고객 충성도 프로그램. 상품 반품 처리. 자동 세금 기록 및 보고. 	<ul style="list-style-type: none"> 문서 인텔리전스 스튜디오 REST API C# SDK Python SDK Java SDK JavaScript

[모델 형식으로 돌아가기](#)

ID(신원)



Fields Result Code

DocType: idDocument.passport

CountryRegion #1	99.00%
MYS	
DateOfBirth #1	99.00%
1975-11-02	
DateOfExpiration #1	99.00%
2022-03-10	
DateOfIssue #1	99.00%
2017-03-11	
DocumentNumber #1	99.00%
R00000000	
DocumentType #1	99.00%

[+] 테이블 확장

Model ID	설명	자동화 사용 사례	개발 옵션
prebuilt-idDocument	<ul style="list-style-type: none">여권 및 ID 카드에서 주요 정보를 추출합니다.문서 유형미국 운전면허증에서 보증, 제한 및 차량 분류를 추출합니다.	<ul style="list-style-type: none">KYC(Know Your Customer) 금융 서비스 지원.의료 계정 관리.ID 검사점 및 게이트웨이.호텔 등록.	<ul style="list-style-type: none">문서 인텔리전스 스튜디오REST APIC# SDKPython SDKJava SDKJavaScript

모델 형식으로 돌아가기

의료 보험 카드

Fields Result Code

- Copays (2) #1
 - Amount: \$1,500
 - Benefit: Deductible
- Amount: \$1,000
- Benefit: Coinsurance Max
- GroupNumber #1: 1000000 (99.50%)
- IdNumber #1: 123456789 (99.50%)
- Insurer #1: PREMERA BLUE CROSS (99.50%)

테이블 확장

Model ID	설명	자동화 사용 사례	개발 옵션
prebuilt-healthInsuranceCard.us	<ul style="list-style-type: none"> 미국 건강 보험 카드에서 주요 정보를 추출합니다. 데이터 및 필드 추출 	<ul style="list-style-type: none"> 적용 범위 및 자격 확인. 예측 모델링. 가치 기반 분석. 	<ul style="list-style-type: none"> 문서 인텔리전스 스튜디오 REST API C# SDK Python SDK Java SDK JavaScript

모델 형식으로 돌아가기

계약 모델

Analyze All pages Range

Fields Content Result Code

- DocType: contract
- EffectiveDate #1: 15 day of October, 2022 (99.99%)
- ExecutionDate #1: 15 day of October, 2022 (99.99%)
- Jurisdictions #1: Clause (100.00%)

This Agreement shall be governed by and construed in accordance with the internal laws of the State of Washington applicable to agreements made and to be performed entirely within such state.
- Region: Washington (100.00%)

Washington
- Parties (2) #1: Parties (1)
 - Contoso Corporation By: Angel Brown Title: CEO Angel Brown
 - Adventure Works Cycle By: Aaron Smith Title: CEO Aaron Smith
- Title #1: WEB HOSTING AGREEMENT (99.98%)

Model ID	설명	개발 옵션
prebuilt-contract	계약 및 당사자 세부 정보를 추출합니다.	<ul style="list-style-type: none"> 문서 인텔리전스 스튜디오 ↗ REST API

모델 형식으로 돌아가기

미국 세금 W-2 모델

The screenshot shows a document analysis interface with a W-2 tax form on the left and its corresponding extracted data on the right. The W-2 form fields include:

- Copy 2 -- To Be Filed with Employee's State Tax Return, City or Local Income Tax Return.
- a. Employee's Soc Sec No: 123-45-6789
- b. Employer ID number (EIN): 98-7654321
- c. Employer's name, address and ZIP code: CONTOSO LTD, 123 MICROSOFT WAY, REDMOND, WA 98765
- d. Control Number: 000086242
- e. Employee's name, address, and ZIP code: ANGEL BROWN, 4567 MAIN STREET, BUFFALO, WA 12345
- 7 Social security tips: 302.30
- 8 Allocated tips: 874.20
- 9 Federal income tax withheld: 3894.54
- 10 Dependent care benefits: 9873.20
- 11 Nonqualified plans: 653.21
- 12a Code See Inst. for box 12: DD
- 12b Code: 5432.00
- 12c Code: 876.30
- 12d Code: 123.30
- PA: 87654321
- WA: 12345678
- 15 State Employer's state ID number: 9831.20
- 16 State wages, tips, etc.: 1032.30
- 17 State income tax: 51.00
- 18 Local wages, tips, etc.: 37160.56
- 19 Local income tax: 594.54
- 20 Locality name: Cumberland Vly/Mdd
- 21 Postal Code: E.Pennsboro/E.Pnns
- 22 Zip Code: 12345

The extracted data on the right includes:

- AllocatedTips #1: 874.2 (99.90%)
- ControlNumber #1: 000086242 (99.90%)
- DependentCareBenefits #1: 9873.2 (99.90%)
- Employee #1: 4567 MAIN STREET BUFFALO, WA 12345 (99.90%)
 - Address: 4567 MAIN STREET BUFFALO, WA 12345
 - HouseNumber: 4567
 - Road: MAIN STREET
 - PostalCode: 12345

Model ID	설명	자동화 사용 사례	개발 옵션
prebuilt-tax.us.W-2	<ul style="list-style-type: none"> IRS 미국 W2 세금 양식(2018-2021년)에서 주요 정보를 추출합니다. 데이터 및 필드 추출 	<ul style="list-style-type: none"> 자동화된 세금 문서 관리. 모기지 대출 신청 처리. 	<ul style="list-style-type: none"> 문서 인텔리전스 스튜디오 ↗ REST API C# SDK Python SDK Java SDK JavaScript

모델 형식으로 돌아가기

미국 세금 1098 양식

Analyze | All pages | Range

8181 VOID CORRECTED

RECIPIENT'S/LENDER'S name, street address, city or town, state or province, country, ZIP or foreign postal code, and telephone no.		OMB No. 1545-1380 Form 1098 (Rev. January 2022) For calendar year 20
RECIPIENT'S/LENDER'S TIN	PAYER'S/BORROWER'S TIN	1 Mortgage interest received from payer(s)/borrower(s) 2 Outstanding mortgage principal 3 Mortgage origination date 4 Refund of overpaid interest 5 Mortgage insurance premiums 6 Points paid on purchase of principal residence 7 If address of property securing mortgage is the same as PAYER'S/BORROWER'S address, check the box, or enter the address or description in box 8. 8 Address or description of property securing mortgage (see instructions) 9 Number of properties securing the mortgage 10 Other 11 Mortgage acquisition date
Street address (including apt. no.)		
City or town, state or province, country, and ZIP or foreign postal code		
Account number (see instructions)		

Form 1098 (Rev. 1-2022) Cat. No. 14402K www.irs.gov/Form1098 Department of the Treasury - Internal Revenue Service
Do Not Cut or Separate Forms on This Page — Do Not Cut or Separate Forms on This Page

Mortgage Interest Statement

Copy A For Internal Revenue Service Center

File with Form 1098.

For Privacy Act and Paperwork Reduction Act Notice, see the current General Instructions for Certain Information Returns.

Fields Result Code

Address
8181 VOID
HouseNumber
8181
Road
VOID
StreetAddress
8181 VOID

● MortgageInsurancePremium #1 99.59%
\$
● MortgageInterest #1 99.99%
\$
● OutstandingMortgagePrincipal #1 99.95%
\$
● OverpaidInterestRefund #1 98.60%
\$
● PointsPaid #1 99.98%

□ 테이블 확장

Model ID	설명	개발 옵션
prebuilt-tax.us.1098	모기지 이자 정보 및 세부 정보를 추출합니다.	<ul style="list-style-type: none"> 문서 인텔리전스 스튜디오 REST API

모델 형식으로 돌아가기

미국 세금 1098-E 양식

Analyze | All pages | Range

8484 VOID CORRECTED

RECIPIENT'S/LENDER'S name, street address, city or town, state or province, country, ZIP or foreign postal code, and telephone number		OMB No. 1545-1576 2023 Form 1098-E
RECIPIENT'S TIN	BORROWER'S TIN	1 Student loan interest received by lender 2 Check if box 1 does not include loan origination fees and/or capitalized interest and the loan was made before September 1, 2004
BORROWER'S name		
Street address (including apt. no.)		
City or town, state or province, country, and ZIP or foreign postal code		
Account number (see instructions)		

Form 1098-E Cat. No. 25088U www.irs.gov/Form1098E Department of the Treasury - Internal Revenue Service
Do Not Cut or Separate Forms on This Page — Do Not Cut or Separate Forms on This Page

Student Loan Interest Statement

Copy A For Internal Revenue Service Center

File with Form 1098.

For Privacy Act and Paperwork Reduction Act Notice, see the 2023 General Instructions for Certain Information Returns.

Fields Result Code

DocType: tax.us.1098E
● ExcludesFeesOrInterest
false
● StudentLoanInterest #1 100.00%
\$
● TaxYear #1 2023

□ 테이블 확장

Model ID	설명	개발 옵션
prebuilt-tax.us.1098E	학자금 대출 정보 및 세부 정보를 추출합니다.	<ul style="list-style-type: none"> 문서 인텔리전스 스튜디오 REST API

모델 형식으로 돌아가기

미국 세금 1098-T 양식

Analyze | All pages Range

8383		<input type="checkbox"/> VOID	<input type="checkbox"/> CORRECTED
FILER'S name, street address, city or town, state or province, country, ZIP or foreign postal code, and telephone number		1 Payments received for qualified tuition and related expenses OMB No. 1545-1574 S 2 Form 1098-T	
FILER'S employer identification no.	STUDENT'S TIN	2023	
STUDENT'S name		3	Tuition Statement
Street address (including apt. no.)		4 Adjustments made for a prior year	5 Scholarships or grants
City or town, state or province, country, and ZIP or foreign postal code		6 Adjustments to scholarships or grants for a prior year	7 Checked if the amount in box 1 includes amounts for an academic period beginning January–March 2024 S 8 Checked if at least half-time student
Service Provider/Acct. No. (see instr.)	9 Checked if a graduate student	10 Ins. contract reimb./refund	Copy A For Internal Revenue Service Center File with Form 1096. For Privacy Act and Paperwork Reduction Act Notice, see the 2023 General Instructions for Certain Information Returns.
Form 1098-T Cat. No. 25087J www.irs.gov/Form1098T Department of the Treasury - Internal Revenue Service Do Not Cut or Separate Forms on This Page — Do Not Cut or Separate Forms on This Page			

< 1 of 1 >

Fields Result Code

- IncludesAmountForNextPeriod false
- InsuranceContractReimbursements #1 100.00% \$
- IsAtLeastHalfTimeStudent false
- IsGraduateStudent false
- PaymentReceived #1 99.30% \$
- Scholarships #1 99.98% \$
- ScholarshipsAdjustments #1 99.99% \$
- TaxYear #1 100.00% 2023

데이터 확장

Model ID	설명	개발 옵션
prebuilt-tax.us.1098T	수업료 정보 및 세부 정보를 추출합니다.	<ul style="list-style-type: none"> 문서 인텔리전스 스튜디오 ↗ REST API

모델 형식으로 돌아가기

미국 세금 1099 (및 변형) 양식

Run analysis Query fields Analyze options

7171		<input type="checkbox"/> VOID	<input type="checkbox"/> CORRECTED
PAYER'S name, street address, city or town, state or province, country, ZIP or foreign postal code, and telephone no.		OMB No. 1545-0116 Form 1099-NEC (Rev. January 2022) For calendar year 2021	
PAYER'S TIN	RECIPIENT'S TIN	1 Nonemployee compensation \$	Nonemployee Compensation
RECIPIENT'S name		2 Payer made direct sales totaling \$5,000 or more of consumer products to recipient for resale	Copy A For Internal Revenue Service Center File with Form 1096. For Privacy Act and Paperwork Reduction Act Notice, see the current General Instructions for Certain Information Returns.
Street address (including apt. no.)		3	
City or town, state or province, country, and ZIP or foreign postal code		4 Federal income tax withheld \$	
Account number (see Instructions)		5 State tax withheld \$	6 State/Payer's state no. \$
		7 State income \$	
Form 1099-NEC (Rev. 1-2022) Cat. No. 72590N www.irs.gov/Form1099NEC Department of the Treasury - Internal Revenue Service Do Not Cut or Separate Forms on This Page — Do Not Cut or Separate Forms on This Page			

Fields Result Code

- Payer #2 85.50% PhoneNumber
- Box2 #2 99.90% false
- TaxYear #2 99.90% 20

데이터 확장

Model ID	설명	개발 옵션
prebuilt-tax.us.1099(변형)	1099 양식 변형에서 정보를 추출합니다.	<ul style="list-style-type: none"> 문서 인텔리전스 스튜디오 문서 인텔리전스 스튜디오 REST API REST API

모델 형식으로 돌아가기

사용자 지정 모델 개요

Label data

Receipt No	2468
Sold To	Fabrikam Residences
ID #	1197531
Live Delivery?	unselected
Online Delivery?	selected
Video Delivery?	selected

데이터 확장

정보	설명	자동화 사용 사례	개발 옵션
사용자 지정 모델	일련의 대표적인 학습 문서 세트에서 만들어진 모델을 기반으로 양식 및 문서에서 정형 데이터로 정보를 추출합니다.	사용자 비즈니스 및 사용 사례와 관련된 양식 및 문서에서 고유 데이터를 추출합니다.	<ul style="list-style-type: none"> 문서 인텔리전스 스튜디오 문서 인텔리전스 스튜디오 REST API C# SDK Java SDK JavaScript SDK Python SDK

사용자 지정 모델 형식으로 돌아가기

사용자 지정 템플릿

W-9

Form
(Rev. October 2018)
Department of the Treasury
Internal Revenue Service

**Request for Taxpayer
Identification Number and Certification**

► Go to www.irs.gov/FormW9 for instructions and the latest information.

Give Form to the
requester. Do not
send to the IRS.

Print or type.
See Specific Instructions on page 3.

1. Name (as shown on your income tax return). Name is required on this line; do not leave this line blank.				
John Doe				
2. Business name/dissolved entity name, if different from above Arctex Inc.				
3. Check appropriate box for federal tax classification of the person whose name is entered on line 1. Check only one of the following seven boxes.				
<input type="checkbox"/> Individual/sole proprietor or	<input type="checkbox"/> C Corporation	<input type="checkbox"/> S Corporation	<input type="checkbox"/> Partnership	<input type="checkbox"/> Trust/estate single-member LLC
<input checked="" type="checkbox"/> Limited liability company. Enter the tax classification (C=C corporation, S=S corporation, P=Partnership) P				
Note: Check the appropriate box in the line above for the tax classification of the single-member owner. Do not check LLC if the LLC is classified as a single-member LLC that is disregarded from the owner unless the owner of the LLC is another LLC that is not disregarded from the owner for U.S. federal tax purposes. Otherwise, a single-member LLC that is disregarded from the owner should check the appropriate box for the tax classification of its owner.				
<input type="checkbox"/> Other (see instructions) ►				
5. Address (number, street, and apt. or suite no.) See instructions. 101 Test Address		Requester's name and address (optional) Jamie Doe		
6. City, state, and ZIP code Bellevue WA 98005		(Applies to accounts maintained outside the U.S.)		
7. List account number(s) here (optional) 123456789123, 123456789987				
Part I Taxpayer Identification Number (TIN)				
Enter your TIN in the appropriate box. The TIN provided must match the name given on line 1 to avoid backup withholding. For individuals, this is generally your social security number (SSN). However, for a resident alien, sole proprietor, or disregarded entity, see the instructions for Part I, later. For other entities, it is your employer identification number (EIN). If you do not have a number, see How to get a TIN, later.				
Note: If the account is in more than one name, see the instructions for line 1. Also see What Name and				
<div style="display: flex; align-items: center;"> <div style="flex: 1;"> <input style="width: 100%; height: 1.2em; border: 1px solid #ccc; padding: 2px; margin-right: 10px;" type="text"/> Social security number </div> <div style="flex: 1; text-align: right;"> <input style="width: 100%; height: 1.2em; border: 1px solid #ccc; padding: 2px; margin-right: 10px;" type="text"/> Employer identification number </div> </div>				

① 참고

사용자 지정 템플릿 모델을 학습하려면 `buildMode` 속성을 `template` 으로 설정합니다. 자세한 내용은 [see 템플릿 모델 학습](#)을 참조하세요.

▣ 테이블 확장

정보	설명	자동화 사용 사례	개발 옵션
사용자 지정 템플릿 모델	사용자 지정 템플릿 모델은 정형 및 반정형 문서에서 레이아웃이 지정된 값과 필드를 추출합니다.	정의된 시각적 템플릿 또는 일반적인 시각적 레이아웃, 양식을 사용하여 고도로 구조화된 문서에서 주요 데이터를 추출합니다.	<ul style="list-style-type: none"> 문서 인텔리전스 스튜디오 REST API C# SDK Python SDK Java SDK JavaScript SDK

[사용자 지정 모델 형식으로 돌아가기](#)

사용자 지정 인공신경망

HOUSE RENTAL AGREEMENT

This House Rental Agreement ("Agreement," "rental agreement," or "lease") is entered into between Opayi LLC (Landlord) and Sam Zamora and Garry Byrme and Sarah Cantrell (Tenants). If more than one person is named as Tenant they shall be jointly and severally liable and responsible under the terms of this Agreement. This lease Agreement involves a residential house, yard, and related facilities located at 123 Bellevue Dr. Salt Lake City Utah 84158 (the "premises"). The date of this Agreement is January 15 2021.

1. Landlord rents to Tenant, unfurnished, the premises on a month to month basis, terminable by either party at the end of any calendar month on at least 30 days notice to the other party. Tenant shall be entitled to possession of the premises and rent shall commence on April 15 2021. Tenant shall not assign, sublease, or allow anyone other than persons permitted under this lease to at any time be in possession of any portion of the premises. Landlord will provide five (5) keys to Tenant; each key fits all outside door locks. Landlord will also provide two keys to the garage, and one remote garage door opener. All keys and the remote door opener will be returned to Landlord at the end of the tenancy.

2. Tenant agrees to pay the stipulated monthly rent in advance on the first day of each calendar month without offset or reduction. All rent is payable to Landlord at the address of Landlord, 123 Old Fernwood Road Salt Lake City Utah 84998 or at such other place as Landlord

① 참고

사용자 지정 인공신경망 모델을 학습하려면 buildMode 속성을 neural로 설정합니다. 자세한 내용은 [see 인공신경망 모델 학습](#)을 참조하세요.

[] 테이블 확장

정보	설명	자동화 사용 사례	개발 옵션
사용자 지정 인공신경망 모델	사용자 지정 인공신경망 모델은 정형(설문 조사, 설문지), 반정형(청구서, 구매 주문) 및 비정형 문서(계약, 편지)에서 레이블이 지정된 데이터를 추출하는 데 사용됩니다.	정형 문서와 비정형 문서에서 텍스트 데이터, 확인란 및 테이블 형식 필드를 추출합니다.	문서 인텔리전스 스튜디오 • REST API • C# SDK • Java SDK • JavaScript SDK • Python SDK

[사용자 지정 모델 형식으로 돌아가기](#)

사용자 지정 구성형

Custom model

Models

Model ID	Model Description
test	test
composed-fr-model	five custom models composed
composed-fr	composed from three models

Compose **Test** **Delete** **Copy**

Label data

Models

Test

Settings

테이블 확장

정보	설명	자동화 사용 사례	개발 옵션
구성형 사용자 지정 모 델	구성형 모델은 사용자 지정 모델 컬렉션을 가져와서 사용자의 양식 유형에서 빌드된 단일 모델에 할당하여 만듭니다.	여러 모델을 학습시키고 그룹화하여 구매 주문처럼 비슷한 양식 유형을 분석하려는 경우에 유용합니다.	<ul style="list-style-type: none"> 문서 인텔리전스 스튜디오 REST API C# SDK Java SDK JavaScript SDK Python SDK

사용자 지정 모델 형식으로 돌아가기

사용자 지정 분류 모델

Label data

Sort Filter 17/17 labeled

Drag and drop similar files here to create a new document type and assign labels

car-maint

sample1/car-maint/Comme... X
sample1/car-maint/Comme... X
sample1/car-maint/Comme... X
sample1/car-maint/Comme... X
sample1/car-maint/Comme... X

cc-auth

sample1/cc-auth/CCAuth-1... X
sample1/cc-auth/CCAuth-2... X
sample1/cc-auth/CCAuth-3... X
sample1/cc-auth/CCAuth-4... X
sample1/cc-auth/CCAuth-5... X
sample1/cc-auth/CCAuth-6... X
sample1/cc-auth/CCAuth-7... X

정보	설명	자동화 사용 사례	개발 옵션
구성된 분류 모 델	사용자 지정 분류 모델은 레이아웃 및 언어 기능을 결합하여 입력 파일 내에서 문서를 검색, 식별 및 분류합니다.	<ul style="list-style-type: none">신청서, 급여 명세서 및 은행 명세서를 포함하는 패키지 대출 신청.스캔한 청구서 컬렉션.	<ul style="list-style-type: none">문서 인텔리전스 스튜디오REST API

사용자 지정 모델 형식으로 돌아가기

데이터 개인 정보 보호 및 보안

모든 AI와 마찬가지로 문서 인텔리전스 서비스를 사용하는 개발자는 고객 데이터에 대한 Microsoft 정책을 알고 있어야 합니다. [문서 인텔리전스의 데이터, 개인 정보 보호 및 보안 페이지](#)를 참조하세요.

다음 단계

- 문서 인텔리전스 모델을 선택합니다.
- Document Intelligence Studio를 사용하여 사용자 고유의 양식 및 문서를 처리해 보세요.
- Document Intelligence 빠른 시작을 완료하고 원하는 개발 언어로 문서 처리 앱 만들기를 시작해 보세요.

어떤 모델을 선택해야 하나요?

아티클 • 2024. 04. 13.

① 중요

- 문서 인텔리전스 공개 미리 보기 릴리스에서는 현재 개발 중인 기능에 대한 조기 액세스를 제공합니다.
- 기능, 방식 및 프로세스는 GA(일반 공급) 전에 사용자 피드백에 따라 변경될 수 있습니다.
- 문서 인텔리전스 클라이언트 라이브러리의 공개 미리 보기 버전은 기본적으로 REST API 버전 [2024-02-29-미리 보기](#)로 설정됩니다.
- 공개 미리 보기 버전 [2024-02-29-미리 보기](#)는 현재 다음 Azure 지역에서만 사용할 수 있습니다.
 - 미국 동부
 - 미국 서부2
 - 서유럽

이 콘텐츠의 적용 대상은 다음과 같습니다. v4.0(미리 보기) | 이전 버전: v3.1(GA) v3.0(GA)

Azure AI 문서 인텔리전스는 애플리케이션에 지능형 문서 처리를 추가하고 워크플로를 최적화할 수 있는 다양한 모델을 지원합니다. 올바른 모델을 선택하는 것은 기업의 성공을 보장하는 데 필수적입니다. 이 문서에서는 사용 가능한 문서 인텔리전스 모델을 살펴보고 프로젝트에 가장 적합한 솔루션을 선택하는 방법에 대한 지침을 제공합니다.

<https://www.microsoft.com/ko-kr/videoplayer/embed/RE5fX1b?postJsIMsg=true&autoCaptions=ko-kr>

다음 의사 결정 차트는 각 [문서 인텔리전스 v3.0](#) 지원 모델의 기능을 강조하고 애플리케이션의 요구 사항에 가장 적합한 모델을 선택하는 데 도움이 됩니다.

① 중요

지원되는 언어 텍스트와 기능별 필드 추출은 [언어 지원](#) 페이지를 확인합니다.

미리 학습된 문서 분석 모델

Document type	예시	추출할 데이터	최적의 솔루션
일반 문서.	계약서 또는 서신.	주로 서면 또는 인쇄된 텍스트 줄, 단어, 위치 및 검색된 언어를 추출하려고 합니다.	OCR 모델 읽기
구조적 정보가 포함된 문서.	보고서 또는 연구.	서면 또는 인쇄된 텍스트 외에도 테이블, 선택 표시, 단락, 제목, 머리글 및 부제목과 같은 구조적 정보를 추출해야 합니다.	레이아웃 분석 모델
필드(키) 및 값 형식의 콘텐츠를 포함하는 구조화된 또는 반구조화된 문서입니다.	신용 신청서나 설문 조사와 같이 비즈니스나 업계에서 일반적으로 사용되는 표준화된 형식인 양식 또는 문서입니다.	사용자 지정 모델을 학습시킬 필요 없이 미리 빌드된 시나리오별 모델에서 다루지 않는 필드와 값을 추출할 수 있습니다.	**선택적 쿼리 문자열 매개 변수 features=keyValuePairs가 사용하도록 설정된 모델 **

미리 학습된 시나리오별 모델

[+] 테이블 확장

Document type	추출할 데이터	최적의 솔루션
미국 W-2 세금 양식	급여, 임금, 원천징수된 세금과 같은 주요 정보를 추출하려고 합니다.	미국 세금 W-2 모델
미국 세금 1098 양식	원금, 포인트 및 세금과 같은 모기지 이자 세부 정보를 추출하려고 합니다.	미국 세금 1098 양식
미국 세금 1098-E 양식	대출 및 이자 금액과 같은 학자금 대출 이자 세부 정보를 추출하려고 합니다.	미국 세금 1098-E 양식
미국 세금 1098T 양식	장학금 조정, 학생 상태 및 대출 정보와 같은 정규 장학금 세부 정보를 추출하려고 합니다.	미국 세금 1098-T 모드
미국 세금 1099(변형) 양식	1099 양식 및 해당 변형(A, B, C, CAP, DIV, G, H, INT, K, LS, LTC, MISC, NEC, OID, PATR, Q, QA, R, S, SA, SB)에서 정보를 추출하려고 합니다.	미국 세금 1099 양식

Document type	추출할 데이터	최적의 솔루션
미국 세금 1040(변형) 양식	1040 양식 및 해당 변형(일정 1, 일정 2, 일정 3, 일정 8812, 일정 A, 일정 B, 일정 C, 일정 D, 일정 E, 일정 E, 일정 EIC, 일정 F, 일정 H, 일정 J, 일정 R, 일정 SE, 일정 선임)에서 정보를 추출하려고 합니다.	미국 세금 1040 모델
계약(당사자 간의 법적 계약)	당사자, 날짜 및 간격과 같은 계약 세부 정보를 추출하려고 합니다.	계약 모델
건강 보험 카드 또는 건강 보험 ID.	보험사, 멤버 ID, 처방 보장 범위, 그룹 번호 등 주요 정보를 추출하려고 합니다.	의료 보험 카드 모델
신용 또는 직불 카드	카드 번호 및 은행 이름과 같은 주요 정보 은행 카드를 추출하려고 합니다.	신용/직불 카드 모델
결혼 인증서 .	결혼 증명서에서 주요 정보를 추출하려고 합니다.	결혼 인증서 모델
청구서 또는 청구서.	고객 이름, 청구 주소, 미청구 금액과 같은 주요 정보를 추출하려고 합니다.	청구서 모델
영수증, 바우처 또는 단일 페이지 호텔 영수증.	가맹점명, 거래 일자, 거래 총액 등 주요 정보를 추출하고자 합니다.	영수증 모델
미국 운전면허증이나 국가별 여권과 같은 신분증(ID)입니다.	이름, 성, 생년월일, 주소 및 서명과 같은 주요 정보를 추출하려고 합니다.	ID 문서 (ID) 모델
미국 모기지 1003 .	Uniform Residential Loan 애플리케이션에서 주요 정보를 추출하려고 합니다.	1003 양식 모델
미국 모기지 1008 .	Uniform Underwriting and Transmittal 요약에서 키 정보를 추출하려고 합니다.	1008 양식 모델
미국 모기지 마감 공시 .	모기지 마감 공시 양식에서 주요 정보를 추출하려고 합니다.	모기지 마감 공시 양식 모델
구조화된 요소, 반구조화된 요소 및/또는 구조화되지 않은 요소가 포함된 혼합 형식 문서.	키-값 쌍, 선택 표시, 테이블, 서명 필드 및 사전 구축 또는 일반 문서 모델에서 추출하지 않은 선택된 영역을 추출할 수 있습니다.	사용자 지정 모델

💡 팁

- 어떤 미리 학습된 모델을 사용할지 아직 확실하지 않은 경우 선택적 쿼리 문자열 매개 변수 `features=keyValuePairs` 가 사용하도록 설정된 레이아웃 모델을

사용해 보세요.

- 레이아웃 모델은 읽기 OCR 엔진을 통해 페이지, 표, 스타일, 텍스트, 줄, 단어, 위치 및 언어를 검색합니다.

사용자 지정 추출 모델

[+] 테이블 확장

학습 집합	문서 예	최적의 솔루션
정적 레이아웃을 갖춘 구조화되고 일관된 문서.	설문지나 애플리케이션과 같은 구조화된 형태.	사용자 지정 템플릿 모델
정형, 반정형 및 비정형 문서.	<ul style="list-style-type: none">구조화 → 설문 조사반구조화 → 청구서구조화되지 않음 → 서신	사용자 지정 신경망 모델
유사한 형식의 문서에 대해 각각 학습된 여러 모델의 컬렉션입니다.	<ul style="list-style-type: none">공급 구매 주문서장비 구매 주문서가구 구매 주문서 <p>모두 단일 모델로 구성됩니다.</p>	구성형 사용자 지정 모델

사용자 지정 분류 모델

[+] 테이블 확장

학습 집합	문서 예	최적의 솔루션
최소 두 가지 유형의 문서	양식, 문자 또는 문서	사용자 지정 분류 모델

다음 단계

- 문서 인텔리전스 스튜디오 [☞](#)를 사용하여 고유의 양식과 문서를 처리하는 방법을 알아봅니다.

문서 인텔리전스 스튜디오란?

아티클 • 2024. 03. 13.

이 콘텐츠의 적용 대상은 다음과 같습니다. ✓ v4.0(미리 보기) | 이전 버전: ✓ v3.1(GA) ✓ v3.0(GA)

Document Intelligence Studio[↗] 는 Document Intelligence 서비스의 기능을 애플리케이션에 시각적으로 탐색, 이해, 학습 및 통합하는 온라인 도구입니다. 이 스튜디오는 다양한 문서 인텔리전스 모델을 실험하고 코드를 작성할 필요 없이 대화형 방식으로 반환된 데이터를 샘플링할 수 있는 플랫폼을 제공합니다. 문서 인텔리전스 스튜디오를 사용하여 다음을 수행합니다.

- 문서 인텔리전스의 다양한 기능에 대해 자세히 알아봅니다.
- 문서 인텔리전스 리소스를 사용하여 샘플 문서에서 모델을 테스트하거나 자체 문서를 업로드합니다.
- 다양한 추가 기능 및 미리 보기 기능을 실험하여 필요에 맞게 출력을 조정합니다.
- 사용자 지정 분류 모델을 학습시켜 문서를 분류합니다.
- 사용자 지정 추출 모델을 학습시켜 문서에서 필드를 추출합니다.
- 애플리케이션에 통합할 언어별 SDK의 샘플 코드를 가져옵니다.

스튜디오는 모델 분석 및 사용자 지정 모델 학습을 위해 문서 인텔리전스 v3.0 이상 API 버전을 지원합니다. 레이블이 지정된 데이터가 있는 이전에 학습된 v2.1 모델은 지원되지만 v2.1 모델 학습은 지원되지 않습니다. v2.1에서 v3.0으로 마이그레이션하는 방법에 대한 자세한 내용은 [REST API 마이그레이션 가이드](#)를 참조하세요.

문서 인텔리전스 스튜디오 사용 시작

1. 문서 인텔리전스 스튜디오를 사용하려면 다음 자산이 필요합니다.

- Azure 구독 - [무료 구독 만들기](#).
- Azure AI 서비스 또는 문서 인텔리전스 리소스. Azure 구독이 있으면 Azure Portal에서 [단일 서비스](#) 또는 [다중 서비스](#) 리소스를 만들어 키와 앤드포인트를 가져옵니다. 평가판 가격 책정 계층(F0)을 사용하여 서비스를 사용해보고, 나중에 프로덕션용 유료 계층으로 업그레이드합니다.

2. 문서 인텔리전스 스튜디오[↗]로 이동합니다. 처음 로그인하는 경우 서비스 리소스를 구성하라는 팝업 창이 나타납니다. 이때 다음 두 가지 방법을 사용할 수 있습니다.

a. 리소스별 액세스(권장).

- 기존 구독을 선택합니다.

- 구독 내에서 기존 리소스 그룹을 선택하거나 새로 만듭니다.
- 기존 문서 인텔리전스 또는 Azure AI 서비스 리소스를 선택합니다.

b. API 엔드포인트 및 키별 액세스.

- Azure Portal에서 엔드포인트 및 키를 검색합니다.
- 리소스에 대한 개요 페이지로 이동하고 왼쪽 탐색 모음에서 **키 및 엔드포인트**를 선택합니다.
- 해당 필드에 값을 입력합니다.

Home > Contoso-DI

Contoso-DI | Keys and Endpoint

Search Regenerate Key1 Regenerate Key2

Overview Activity log Access control (IAM) Tags Diagnose and solve problems

Resource Management

Keys and Endpoint

Encryption Pricing tier Networking Identity Cost analysis Properties Locks

Monitoring Automation Help

These keys are used to access your Azure AI service API. Do not share your keys. Store them securely—for example, using Azure Key Vault. We also recommend regenerating these keys regularly. Only one key is necessary to make an API call. When regenerating the first key, you can use the second key for continued access to the service.

Show Keys

KEY 1

KEY 2

Location/Region westus2

Endpoint https://contoso-di.cognitiveservices.azure.com/

3. 리소스가 구성되면 문서 인텔리전스 스튜디오에서 제공하는 다양한 모델을 사용해 볼 수 있습니다. 첫 페이지에서 코드 없는 방법으로 사용할 문서 인텔리전스 모델을 선택합니다.

Extract text, tables, structure, key-value pairs, and named entities from documents.



Read

Extract printed and handwritten text from images and documents.

[Try it out](#)



Layout

Extract tables, check boxes, and text from forms and documents.

[Try it out](#)



General documents

Extract labels, values and entities such as names, dates, and amounts from documents.

[Try it out](#)



General documents with query fields

Extract labels, values and entities such as names, dates, and amounts from documents.

[Private preview – sign up.](#) 

Prebuilt models

Extract data from unique document types using the following prebuilt models.



Invoices

Extract invoice ID, customer details, vendor details, ship to, bill to, total tax, subtotal, line items and more.

[Try it out](#)



Receipts

Extract time and date of the transaction, merchant information, amounts of taxes, totals and more.

[Try it out](#)



Business cards

Extract person name, job title, address, email, company, and phone numbers from business cards.

[Try it out](#)



Identity documents

Extract name, expiration date, machine readable zone, and more from passports and ID cards.

[Try it out](#)



Health insurance cards

Extract insurer, member, prescription, group number and more information from US health insurance cards.

[Try it out](#)



W-2

Extract employee, employer, wage information, etc. from US W-2 Tax Form.

[Try it out](#)



1098

Extract mortgage interest information from US 1098 Tax Form.

[Private preview – sign up.](#) 



1098-E

Extract student loan information from US 1098-E Tax Form.

[Private preview – sign up.](#) 



1098-T

Extract tuition information from US 1098-T Tax Form.

[Private preview – sign up.](#) 



Contracts

Extract the title and signatory parties' information (including names, reference names, and addresses) from contracts.

[Private preview – sign up.](#) 



Vaccination cards

Extract card holder, health agency and vaccination records from US COVID-19 vaccination cards.

[Private preview – sign up.](#) 

Custom models

Train custom models to classify documents and extract text, structure and fields from your forms or documents.



Custom extraction model

Label and build a custom model to extract a specific schema from your forms and documents.

[Create new](#)



Custom classification model

Build a custom classification model to split and classify documents.

[Create new](#)

4. 문서 분석 또는 미리 빌드된 모델을 테스트하려면 모델을 선택하고 샘플 문서를 하나 사용하거나 사용자 고유의 문서를 업로드하여 분석합니다. 분석 결과는 콘텐츠

결과 코드 창의 오른쪽에 표시됩니다.

5. 사용자 지정 모델은 문서에 대해 학습해야 합니다. 사용자 지정 모델에 대한 개요는 [사용자 지정 모델 개요](#)를 참조하세요.

6. 문서 인텔리전스 스튜디오에서 시나리오의 유효성을 검사한 후 [C#, Java, JavaScript](#) 또는 [Python](#) 클라이언트 라이브러리 또는 [REST API](#)를 사용하여 문서 인텔리전스 모델을 고유의 애플리케이션에 통합하기 시작합니다.

각 모델에 대해 자세히 알아보려면 개념 페이지를 참조합니다.

[] 테이블 확장

모델 유형	모델 이름
문서 분석 모델	<ul style="list-style-type: none">• OCR 읽기• 레이아웃 분석• 일반 문서(사용되지 않음 2023-10-31-preview)
미리 빌드된 모델	<ul style="list-style-type: none">• 건강 보험 카드• W-2 양식• 미국 1098 세금 양식• 미국 1098-E 세금 양식• 미국 1098-T 세금 양식• 청구서• 영수증• ID 문서• 명함(사용되지 않음 2023-10-31-preview)• 계약
사용자 지정 모델	<ul style="list-style-type: none">• 사용자 지정 모델 개요• 추출 모델<ul style="list-style-type: none">1 사용자 지정 템플릿1 사용자 지정 신경망• 분류자 모델<ul style="list-style-type: none">1 사용자 지정 분류자• 작성 모델

리소스 관리

이름 및 가격 책정 계층과 같은 리소스 세부 정보를 보려면 문서 인텔리전스 스튜디오 홈 페이지의 오른쪽 상단에 있는 **설정** 아이콘을 선택하고 **리소스** 탭을 선택합니다. 다른 리소스에 액세스할 수 있는 경우 리소스를 전환할 수도 있습니다.

Settings

Directory

Resource

Resources are your unique aliases for the service and allow usage and billing. Choose your default resource. [Learn more about Azure resources.](#)

문서 인텔리전스를 사용하면 애플리케이션 및 워크플로에서 데이터 처리를 신속하게 자동화하고 데이터 기반 전략을 쉽게 보강하며 문서 검색 기능을 능숙하게 보강할 수 있습니다.

다음 단계

- 서비스에서 제공하는 모델을 사용하려면 [문서 인텔리전스 스튜디오](#)를 방문합니다.
- 문서 인텔리전스 기능에 대한 자세한 내용은 [Azure AI 문서 인텔리전스 개요](#)를 참조하세요.

SDK 변경 로그 및 릴리스 내역

아티클 • 2024. 03. 22.

이 참조 문서는 문서 인텔리전스 기능 및 기능 릴리스, 변경 내용, 업데이트 및 개선 사항에 대한 버전 기반 설명을 제공합니다.

① 중요

- 문서 인텔리전스 공개 미리 보기 릴리스에서는 현재 개발 중인 기능에 대한 초기 액세스를 제공합니다.
- 기능, 방식 및 프로세스는 GA(일반 공급) 전에 사용자 피드백에 따라 변경될 수 있습니다.
- Document Intelligence 클라이언트 라이브러리의 공개 미리 보기 버전은 기본적으로 REST API 버전 [2024-02-29-preview로 설정됩니다](#).
- 공개 미리 보기 버전 [2024-02-29-preview](#)는 현재 다음 Azure 지역에서만 사용할 수 있습니다.
 - 미국 동부
 - 미국 서부2
 - 서유럽

2024년 3월(미리 보기) 릴리스

.NET (C#)

- 문서 인텔리전스 1.0.0-beta.2
- 기본적으로 REST API 2024-02-29-preview를 대상으로 합니다.

변경 로그/릴리스 기록 ↗

패키지(NuGet) ↗

추가 정보 ↗

샘플 ↗

マイグ레이션 가이드 ↗

2023년 11월(미리 보기) 릴리스

.NET (C#)

- 문서 인텔리전스 1.0.0-beta.1
- 기본적으로 REST API 2023-10-31-preview를 대상으로 합니다.

[패키지\(NuGet\)](#)

[추가 정보](#)

[샘플](#)

[マイグ레이션 가이드](#)

2023년 8월(GA) 릴리스

C#

- Form Recognizer 4.1.0(2023-08-10)
- 기본적으로 REST API 2023-07-31을 대상으로 합니다.
- REST API 대상 2023-02-28-preview는 더 이상 지원되지 않습니다.
- [주요 변경 내용](#)

[변경 로그/릴리스 기록](#)

[패키지\(NuGet\)](#)

[추가 정보](#)

[샘플](#)

2023년 4월(미리 보기) 릴리스

이 릴리스에는 다음과 같은 업데이트 사항이 포함되어 있습니다.

C#

- Form Recognizer 4.1.0-beta.1(2023-04-13)
- 기본적으로 2023-02-28-미리 보기를 대상 지정합니다.
- 주요 변경 내용 없음

[변경 로그/릴리스 기록](#)

[패키지\(NuGet\)](#)

[추가 정보](#)

[샘플](#)

2022년 9월(GA) 릴리스

이 릴리스에는 다음과 같은 업데이트 사항이 포함되어 있습니다.

① 중요

DocumentAnalysisClient DocumentModelAdministrationClient 2022-08-31년 릴리스
된 API v3.0 GA를 대상으로 합니다. 이러한 클라이언트는 API 2020-06-30-preview
또는 이전 버전에서 더 이상 지원되지 않습니다.

C#

- Form Recognizer 4.0.0 GA(2022-09-08)
- REST API v3.0 및 v2.0 클라이언트 지원

[변경 로그/릴리스 기록](#)

[패키지\(NuGet\)](#)

[マイグ레이션 가이드](#)

[추가 정보](#)

[샘플](#)

2022년 8월(미리 보기) 릴리스

이 릴리스에는 다음과 같은 업데이트 사항이 포함되어 있습니다.

C#

- Form Recognizer 4.0.0-beta.5(2022-08-09)
- REST API 2022-06-30-preview 클라이언트 지원

[변경 로그/릴리스 기록](#)

[패키지\(NuGet\)](#)

[SDK 참조 설명서](#)

2022년 6월(미리 보기) 릴리스

이 릴리스에는 다음과 같은 업데이트 사항이 포함되어 있습니다.

C#

- Form Recognizer 4.0.0-beta.4(2022-06-08)

[변경 로그/릴리스 기록](#)

[패키지\(NuGet\)](#)

[SDK 참조 설명서](#)

SDK 대상: REST API 2024-02-29-preview

아티클 • 2024. 03. 22.

① 중요

- 문서 인텔리전스 공개 미리 보기 릴리스에서는 현재 개발 중인 기능에 대한 조기 액세스를 제공합니다.
- 기능, 방식 및 프로세스는 GA(일반 공급) 전에 사용자 피드백에 따라 변경될 수 있습니다.
- Document Intelligence 클라이언트 라이브러리의 공개 미리 보기 버전은 기본적으로 REST API 버전 [2024-02-29-preview로 설정됩니다](#).
- 공개 미리 보기 버전 [2024-02-29-preview](#) 는 현재 다음 Azure 지역에서만 사용 할 수 있습니다.
- 미국 동부**
- 미국 서부2**
- 서유럽**

✓ REST API 버전 2024-02-29-preview

Azure AI 문서 인텔리전스는 기계 학습을 사용하여 문서의 텍스트와 구조화된 데이터를 분석하는 클라우드 서비스입니다. 문서 인텔리전스 소프트웨어 개발 키트(SDK)는 문서 인텔리전스 모델 및 기능을 애플리케이션에 쉽게 통합할 수 있는 라이브러리 및 도구 세트입니다. 문서 인텔리전스 SDK는 C#/.NET, Java, JavaScript 및 Python 프로그래밍 언어의 플랫폼 전반에 걸쳐 사용할 수 있습니다.

지원되는 프로그래밍 언어

문서 인텔리전스 SDK는 다음 언어와 플랫폼을 지원합니다.

[+] 테이블 확장

언어 → 문서 인텔리전스 SDK 버전	Package(패키지)	지원되는 API 버전	플랫폼 지원
.NET/C# → 1.0.0-beta.2(미리 보기)	NuGet ↗	<ul style="list-style-type: none">2024-02-29(미리 보기)2023-10-31 및(미리 보기)2023-07-31(GA)2022-08-31(GA)	Windows, macOS, Linux, Docker ↗

언어 → 문서 인텔리전스 SDK 버전	Package(패키지)	지원되는 API 버전	플랫폼 지원
		<ul style="list-style-type: none"> v2.1 ↗ v2.0 ↗ 	
Java → 1.0.0-beta.2(미리 보기)	Maven 리포지토리 ↗	<ul style="list-style-type: none"> 2024-02-29(미리 보기) 2023-10-31 및(미리 보기) 2023-07-31(GA) 2022-08-31(GA) v2.1 ↗ v2.0 ↗ 	Windows, macOS, Linux
JavaScript → 1.0.0-beta.2(미리 보기)	npm ↗	<ul style="list-style-type: none"> 2024-02-29(미리 보기) 2023-10-31 및(미리 보기) 2023-07-31(GA) 2022-08-31(GA) v2.1 ↗ v2.0 ↗ 	Browser, Windows, macOS, Linux ↗
Python → 1.0.0b2(미리 보기)	PyPI ↗	<ul style="list-style-type: none"> 2024-02-29(미리 보기) 2023-10-31 및(미리 보기) 2023-07-31(GA) 2022-08-31(GA) v2.1 ↗ v2.0 ↗ 	Windows, macOS, Linux

지원되는 클라이언트

다음 표에는 문서 인텔리전스 서비스의 각 SDK 버전과 지원되는 API 버전 간의 상관 관계가 나와 있습니다.

C#/.NET

[+] 테이블 확장

언어	SDK 별칭	API 버전(기본값)	지원되는 클라이언트
.NET/C# 1.0.0-beta.2(미리 보기)	v4.0(미리 보기)	2024-02-29-preview	DocumentIntelligenceClient DocumentIntelligenceAdministrationClient
.NET/C# 4.1.0	v3.1 최신(GA)	2023-07-31	DocumentAnalyticClient DocumentModelAdministrationClient
.NET/C# 4.0.0	v3.0(GA)	2022-08-31	DocumentAnalyticClient DocumentModelAdministrationClient
.NET/C# 3.1.x	v2.1	v2.1	FormRecognizerClient FormTrainingClient
.NET/C# 3.0.x	v2.0	v2.0	FormRecognizerClient FormTrainingClient

애플리케이션에서 문서 인텔리전스 SDK 사용

문서 인텔리전스 SDK를 사용하면 애플리케이션에서 문서 인텔리전스 서비스를 사용하고 관리할 수 있습니다. SDK는 기본 문서 인텔리전스 REST API를 기반으로 구축되어 프로그래밍 언어 패러다임 내에서 해당 API를 쉽게 사용할 수 있습니다. 원하는 언어로 문서 인텔리전스 SDK를 사용하는 방법은 다음과 같습니다.

1. SDK 클라이언트 라이브러리 설치

C#/.NET

.NET CLI

```
dotnet add package Azure.AI.DocumentIntelligence --version 1.0.0-beta.2
```

PowerShell

```
Install-Package Azure.AI.FormRecognizer -Version 1.0.0-beta.2
```

2. SDK 클라이언트 라이브러리를 애플리케이션으로 가져오기

C#

```
using Azure;
using Azure.AI.DocumentIntelligence;
```

3. 인증 설정

인증에 지원되는 두 가지 방법이 있습니다.

- `azure.core.credentials`의 `AzureKeyCredential`과 함께 [문서 인텔리전스 API 키](#)를 사용하세요.
- `azure-identity`의 [토큰 자격 증명](#)을 사용하여 [Microsoft Entra ID](#)로 인증합니다.

API key 사용

Azure portal에서 문서 인텔리전스 API 키를 찾을 수 있는 위치는 다음과 같습니다.

The screenshot shows the Azure portal interface for a resource named 'Contoso-DI'. The left sidebar lists various service management options like Overview, Activity log, Access control (IAM), Tags, and Resource Management. Under Resource Management, the 'Keys and Endpoint' option is selected and highlighted with a red box. The main content area displays the 'Keys and Endpoint' settings. A note at the top right says: 'These keys are used to access your Azure AI service API. Do not share your keys. Store them securely—for example, using Azure Key Vault. We also recommend regenerating these keys regularly. Only one key is necessary to make an API call. When regenerating the first key, you can use the second key for continued access to the service.' Below this note, there are two input fields: 'KEY 1' and 'KEY 2', both of which are also highlighted with red boxes. There is also a 'Location/Region' dropdown set to 'westus2' and an 'Endpoint' URL field containing 'https://contoso-di.cognitiveservices.azure.com/'.

C#

```
//set `<your-endpoint>` and `<your-key>` variables with the values from  
the Azure portal to create your `AzureKeyCredential` and  
`DocumentIntelligenceClient` instance  
string key = "<your-key>";  
string endpoint = "<your-endpoint>";  
AzureKeyCredential credential = new AzureKeyCredential(key);  
DocumentIntelligenceClient client = new DocumentIntelligenceClient(new  
Uri(endpoint), new AzureKeyCredential(key));
```

Microsoft Entra 토큰 자격 증명 사용

① 참고

지역별 앤드포인트는 Microsoft Entra 인증을 지원하지 않습니다. 이러한 유형의 인증을 사용하기 위해 리소스에 대한 [사용자 지정 하위 도메인](#)을 만듭니다.

권한 부여는 `DefaultAzureCredential`을 사용하면 가장 쉽습니다. 실행 중인 환경에 따라 기본 토큰 자격 증명을 제공하므로 대부분의 Azure 인증 시나리오를 처리할 수 있습니다.

C#/.NET

.NET 애플리케이션용 `DefaultAzureCredential`을 획득하고 사용하는 방법은 다음과 같습니다.

1. .NET용 Azure ID 라이브러리를 설치합니다.

콘솔

```
dotnet add package Azure.Identity
```

PowerShell

```
Install-Package Azure.Identity
```

2. Microsoft Entra 애플리케이션을 등록하고 새 서비스 주체를 만듭니다.
3. 서비스 주체에 `Cognitive Services User` 역할을 할당하여 문서 인텔리전스에 대한 액세스 권한을 부여하세요.
4. Microsoft Entra 애플리케이션의 클라이언트 ID, 테넌트 ID 및 클라이언트 암호 값을 각각 환경 변수 `AZURE_CLIENT_ID`, `AZURE_TENANT_ID` 및

`AZURE_CLIENT_SECRET`으로 설정합니다.

5. `DefaultAzureCredential`을 포함하여 `DocumentIntelligenceClient` 인스턴스를 만듭니다.

C#

```
string endpoint = "<your-endpoint>";  
var client = new DocumentIntelligenceClient(new Uri(endpoint), new  
DefaultAzureCredential());
```

자세한 내용은 클라이언트 인증을 참조하세요 [☞](#).

4. 애플리케이션 빌드

문서 인텔리전스 SDK와 상호 작용할 클라이언트 개체를 만든 다음 해당 클라이언트 개체에 대한 메서드를 호출하여 서비스와 상호 작용합니다. SDK는 동기 메서드와 비동기 메서드를 모두 제공합니다. 더 많은 인사이트를 얻으려면 원하는 언어의 [빠른 시작](#)을 시도해 보세요.

도움말 옵션

개발자 커뮤니티에서 [☞ Microsoft Q&A](#) Azure AI Document Intelligence 및 기타 서비스에 대한 질문을 하고 답변할 수 있는 Stack Overflow 포럼을 사용할 수 있습니다. Microsoft는 포럼을 모니터링하고 커뮤니티에서 아직 답변하지 않은 질문에 답변합니다. 질문을 볼 수 있도록 다음 태그를 사용하세요.

- Microsoft Q&A: `Azure AI Document Intelligence`.
- 스택 오버플로: `azure-ai-document-intelligence`.

다음 단계

문서 인텔리전스 REST API 2023-10-31-rest

작업을 살펴봅니다.

언어 지원: 문서 분석

아티클 • 2024. 02. 28.

이 콘텐츠의 적용 대상은 다음과 같습니다. ✓ v4.0(미리 보기) | 이전 버전: ✓ v3.1(GA) ✓ v3.0(GA) ✓ v2.1(GA)

Azure AI 문서 인텔리전스 모델은 다국어 문서 처리 지원을 제공합니다. 언어 지원 기능을 사용하면 사용자가 자연스럽게 애플리케이션과 통신하고 글로벌 지원에 힘을 실어줄 수 있습니다. 문서 분석 모델을 사용하면 양식 및 문서에서 텍스트를 추출하고 조직의 작업, 사용 또는 진행 상황에 맞는 구조화된 비즈니스 지원 콘텐츠를 반환할 수 있습니다. 다음 표에는 모델 및 기능별 사용 가능한 언어 및 로캘 지원이 나와 있습니다.

- 읽기**: 읽기 모델을 사용하면 인쇄 및 필기 텍스트를 추출하고 분석할 수 있습니다. 이 모델은 사용자 지정 모델 외에도 레이아웃, 일반 문서, 청구서, 영수증, ID 문서, 건강보험증, 세금 문서 및 사용자 지정 모델과 같은 다른 문서 인텔리전스 미리 빌드된 모델에 대한 기본 OCR 엔진입니다. 자세한 내용은 [읽기 모델 개요](#)를 참조하세요.
- 레이아웃**: 레이아웃 모델을 사용하면 양식 및 문서에서 텍스트, 표, 문서 구조 및 선택 표시(예: 라디오 단추 및 확인란)를 추출하고 분석할 수 있습니다.

① 참고

언어 코드(선택 사항)

- 문서 인텔리전스의 딥 러닝 기반 유니버설 모델은 혼합된 언어가 있는 텍스트 줄을 포함하여 문서의 다국어 텍스트를 모두 추출하며 언어 코드를 지정할 필요가 없습니다.
- 언어에 대해 잘 모르는 경우에는 언어 코드를 매개 변수로 제공하지 말고 서비스에서 관련 모델만 강제로 적용하도록 합니다. 그렇지 않으면 서비스에서 불완전하고 잘못된 텍스트를 반환할 수 있습니다.
- 로캘을 지정할 필요도 없습니다. 이것은 선택적 매개 변수입니다. 문서 인텔리전스 딥 러닝 기술은 이미지에 있는 텍스트의 언어를 자동으로 감지합니다.

읽기 모델

모델 ID: prebuilt-read

읽기: 인쇄된 텍스트

다음 표에서는 **인쇄된** 텍스트를 추출하고 분석하기 위한 읽기 모델 언어 지원을 나열합니다.

[] 테이블 확장

언어	코드(선택 사항)
Abaza	abq
Abkhazian	ab
Achinese	ace
Acoli	ach
Adangme	ada
Adyghe	ady
Afar	aa
아프리칸스어	af
Akan	ak
알바니아어	sq
Algonquin	alq
양기카(데바나가리어)	anp
아랍어	ar
아스투리아스어	ast
Asu(탄자니아)	asa
Avaric	av
아와디-힌디(데바나가리어)	awa
Aymara	ay
아제르바이잔어(라틴어)	az
Bafia	ksf
Bagheli	bfy

언어	코드(선택 사항)
Bambara	bm
바슈키르어	ba
바스크어	eu
벨로루시어(키릴 자모)	be , be-cyrl
벨로루시어(라틴어)	be , be-latn
Bemba(잠비아)	bem
Bena(탄자니아)	bez
보즈푸리-힌디(데바나가리어)	bho
Bikol	bik
Bini	bin
비슬라마어	bi
보도(데바나가리어)	brx
보스니아어(라틴 문자)	bs
Brajbha	bra
브르타뉴어	br
불가리아어	bg
Bundeli	bns
부랴트어(키릴 자모)	bua
카탈로니아어	ca
세부아노어	ceb
Chamling	rab
차모로어	ch
Chechen	ce
차티스가르히(데바나가리어)	hne
Chiga	cgg
중국어 간체	zh-Hans

언어	코드(선택 사항)
중국어 번체	zh-Hant
Choctaw	cho
Chukot	ckt
Chuvash	cv
콘월어	kw
코르시카어	co
Cree	cr
Creek	mus
크림 타타르어(라틴어)	crh
크로아티아어	hr
Crow	cro
체코어	cs
덴마크어	da
Dargwa	dar
다리어	prs
디말(데바나가리어)	dhi
도그리(데바나가리어)	doi
Duala	dua
Dungan	dng
네덜란드어	nl
Efik	efi
영어	en
에르자어(키릴 자모)	myv
에스토니아어	et
페로어	fo
피지어	fj

언어	코드(선택 사항)
필리핀어	fil
핀란드어	fi

□ 테이블 확장

언어	코드(선택 사항)
Fon	fon
프랑스어	fr
프리울리안어	fur
Ga	gaa
가가우즈어(라틴어)	gag
갈리시아어	gl
Ganda	lg
Gayo	gay
독일어	de
길버트어	gil
곤디(데바나가리어)	gon
그리스어	el
그린란드어	k1
과라니어	gn
구릉(데바나가리어)	gvr
Gusii	guz
아이티어	ht
할비(데바나가리어)	h1b
하니어	hni
Haryanvi	bgc
하와이어	haw

언어	코드(선택 사항)
하브리어	he
Herero	hz
Hiligaynon	hil
힌디어	hi
몽 다오어(라틴어)	mww
Ho(데바나기리어)	hoc
헝가리어	hu
Iban	iba
아이슬란드어	is
이그보어	ig
Iloko	ilo
이나리 사미어	snn
인도네시아어	id
Ingush	inh
국제어	ia
이누크티투트어 (라틴어)	iu
아일랜드어	ga
이탈리아어	it
일본어	ja
자운사리(데바나가리어)	Jns
자바어	jv
Jola-Fonyi	dyo
Kabardian	kbd
Kabuverdianu	kea
카친어(라틴어)	kac
Kalenjin	kln

언어	코드(선택 사항)
Kalmyk	xal
캉그리(데바나가리어)	xnr
Kanuri	kr
카라차이발카르어	krc
카라칼파크어(키릴 자모)	kaa-cyr1
카라칼파크어(라틴어)	kaa
카슈비아어	csb
카자흐어(키릴 자모)	kk-cyr1
카자흐어(라틴어)	kk-latn
Khakas	kjh
칼링	klr
카시어	kha
키체어	quc
Kikuyu	ki
Kildin Sami	sjd
키냐르완다어	rw
Komi	kv
Kongo	kg
한국어	ko
Korku	kfq
코랴크어	kpy
코스라에어	kos
Kpelle	kpe
Kuanyama	kj
쿠믹어(키릴 자모)	kum
쿠르드어(아랍어)	ku-arab

언어	코드(선택 사항)
쿠르드어(라틴 문자)	ku-latn
쿠루크(데바나가리어)	kru
키르기스어(키릴 자모)	ky
Lak	lbe
라코타어	lkt

[+] 테이블 확장

언어	코드(선택 사항)
라틴어	la
라트비아어	lv
Lezghian	lex
Lingala	ln
리투아니아어	lt
저지 소르브어	dsb
Lozi	loz
룰레 사미어	smj
Luo(케냐 및 탄자니아)	luo
룩셈부르크어	lb
Luyia	luy
마케도니아어	mk
Machame	jmc
Madurese	mad
마하수 파하리(데바나가리어)	bfz
Makhuwa-Meetto	mgh
Makonde	kde
말라가시어	mg

언어	코드(선택 사항)
말레이어(라틴 문자)	ms
몰타어	mt
말토어(데바나가리어)	kmj
Mandinka	mnk
맹크스어	gv
마오리어	mi
Mapudungun	arn
마라티어	mr
Mari(러시아)	chm
Masai	mas
Mende(시에라리온)	men
Meru	mer
Meta'	mgo
Minangkabau	min
모호크어	moh
몽골어(키릴 자모)	mn
Mongondow	mog
몬테네그린어(키릴 자모)	cnr-cyr1
몬테네그린어(라틴어)	cnr-latn
Morisyen	mfe
Mundang	mua
Nahuatl	nah
Navajo	nv
Ndonga	ng
나폴리어	nap
네팔어	ne

언어	코드(선택 사항)
Ngomba	jgo
니우에안어	niu
노가이어	nog
North Ndebele	nd
북부 사미어(라틴어)	sme
노르웨이어	no
Nyanja	ny
Nyankole	yn
Nzima	nzi
오크어	oc
Ojibwa	oj
오로모어	om
오세트어	os
Pampanga	pam
Pangasinan	pag
Papiamento	pap
파슈토어	ps
Pedi	nso
페르시아어	fa
폴란드어	pl
포르투갈어	pt
편잡어(아랍어)	pa
케추아어	qu
리프아리안어	ksh
루마니아어	ro
로망슈어	rm

언어	코드(선택 사항)
Rundi	rn
러시아어	ru
Rwa	rwk
사드리(데바나가리어)	sck
사하어	sah
Samburu	saq
사모아어(라틴어)	sm
Sango	sg

[+] 테이블 확장

언어	코드(선택 사항)
Sangu(가봉)	snq
Sanskrit(데바나가리)	sa
Santali(데바나기리어)	sat
스코틀랜드어	sco
스코틀랜드 게일어	gd
Sena	seh
세르비아어(키릴 자모)	sr-cyrl
세르비아어(라틴 문자)	sr, sr-latn
Shambala	ksb
쇼나어	sn
Siksika	bla
시르마우리(데바나가리어)	srx
스콜트 사미어	sms
슬로바키아어	sk
슬로베니아어	sl

언어	코드(선택 사항)
Soga	xog
소말리어(아랍어)	so
Somali(라틴어)	so-latn
Songhai	son
South Ndebele	nr
Southern Altai	alt
남부 사미어	sma
Southern Sotho	st
스페인어	es
순다어	su
스와힐리어(라틴어)	sw
Swati	ss
스웨덴어	sv
Tabassaran	tab
Tachelhit	shi
타히티어	ty
Taita	dav
타지크어(키릴 자모)	tg
타밀어	ta
타타르어(키릴 자모)	tt-cyr1
타타르어 (라틴어)	tt
Teso	teo
테툼어	tet
태국어	th
탕미	thf
Tok Pisin	tpi

언어	코드(선택 사항)
통가어	to
Tsonga	ts
Tswana	tn
터키어	tr
투르크멘어(라틴어)	tk
투반어	tyv
Udmurt	udm
Uighur(키릴 자모)	ug-cyr1
우크라이나어	uk
북부 슬라브어	hsb
우르두어	ur
위구르어(아랍어)	ug
우즈베키스탄어(아랍어)	uz-arab
우즈베크어(키릴 자모)	uz-cyr1
우즈베크어(라틴 문자)	uz
베트남어	vi
볼라퓌크어	vo
Vunjo	vun
Walser	wae
웨일스어	cy
서 프리지아어	fy
월라프어	wo
코사어	xh
유카텍 마야어	yua
Zapotec	zap
Zarma	dje

언어	코드(선택 사항)
장족어	za
줄루어	zu

레이아웃

모델 ID: prebuilt-layout

레이아웃: 인쇄된 텍스트		테이블 확장
<p>다음 표에는 인쇄된 텍스트에 지원되는 언어가 나와 있습니다.</p>		
언어	코드(선택 사항)	
Abaza	abq	
Abkhazian	ab	
Achinese	ace	
Acoli	ach	
Adangme	ada	
Adyghe	ady	
Afar	aa	
아프리칸스어	af	
Akan	ak	
알바니아어	sq	
Algonquin	alq	
앙기카(데바나가리어)	anp	
아랍어	ar	
아스투리아스어	ast	

언어	코드(선택 사항)
Asu(탄자니아)	asa
Avaric	av
아와디-힌디(데바나가리어)	awa
Aymara	ay
아제르바이잔어(라틴어)	az
Bafia	ksf
Bagheli	bfy
Bambara	bm
바슈키르어	ba
바스크어	eu
벨로루시어(키릴 자모)	be, be-cyril
벨로루시어(라틴어)	be, be-latn
Bemba(잠비아)	bem
Bena(탄자니아)	bez
보즈푸리-힌디(데바나가리어)	bho
Bikol	bik
Bini	bin
비슬라마어	bi
보도(데바나가리어)	brx
보스니아어(라틴 문자)	bs
Brajbha	bra
브르타뉴어	br
불가리아어	bg
Bundeli	bns
부랴트어(키릴 자모)	bua
카탈로니아어	ca

언어	코드(선택 사항)
세부아노어	ceb
Chamling	rab
차모로어	ch
Chechen	ce
차티스가르히(데바나가리어)	hne
Chiga	cgg
중국어 간체	zh-Hans
중국어 번체	zh-Hant
Choctaw	cho
Chukot	ckt
Chuvash	cv
콘월어	kw
코르시카어	co
Cree	cr
Creek	mus
크림 타타르어(라틴어)	crh
크로아티아어	hr
Crow	cro
체코어	cs
덴마크어	da
Dargwa	dar
다리어	prs
디말(데바나가리어)	dhi
도그리(데바나가리어)	doi
Duala	dua
Dungan	dng

언어	코드(선택 사항)
네덜란드어	nl
Efik	efi
영어	en
에르자어(키릴 자모)	myv
에스토니아어	et
페로어	fo
피지어	fj
필리핀어	fil
핀란드어	fi

[+] 테이블 확장

언어	코드(선택 사항)
Fon	fon
프랑스어	fr
프리울리안어	fur
Ga	gaa
가가우즈어(라틴어)	gag
갈리시아어	gl
Ganda	lg
Gayo	gay
독일어	de
길버트어	gil
곤디(데바나가리어)	gon
그리스어	el
그린란드어	kl
과라니어	gn

언어	코드(선택 사항)
구릉(데바나가리어)	gvr
Gusii	guz
아이티어	ht
할비(데바나가리어)	hlb
하니어	hni
Haryanvi	bgc
하와이어	haw
히브리어	he
Herero	hz
Hiligaynon	hil
힌디어	hi
몽 다오어(라틴어)	mww
Ho(데바나기리어)	hoc
헝가리어	hu
Iban	iba
아이슬란드어	is
이그보어	ig
Iloko	ilo
이나리 사미어	smn
인도네시아어	id
Ingush	inh
국제어	ia
이누크티투트어 (라틴어)	iu
아일랜드어	ga
이탈리아어	it
일본어	ja

언어	코드(선택 사항)
자운사리(데바나가리어)	Jns
자바어	jv
Jola-Fonyi	dyo
Kabardian	kbd
Kabuverdianu	kea
카친어(라틴어)	kac
Kalenjin	kln
Kalmyk	xal
캉그리(데바나가리어)	xnr
Kanuri	kr
카라차이발카르어	krc
카라칼파크어(키릴 자모)	kaa-cyr1
카라칼파크어(라틴어)	kaa
카슈비아어	csb
카자흐어(키릴 자모)	kk-cyr1
카자흐어(라틴어)	kk-latn
Khakas	kjh
칼링	klr
카시어	kha
키체어	quc
Kikuyu	ki
Kildin Sami	sjd
키냐르완다어	rw
Komi	kv
Kongo	kg
한국어	ko

언어	코드(선택 사항)
Korku	kfq
코랴크어	kpy
코스라에어	kos
Kpelle	kpe
Kuanyama	kj
쿠믹어(키릴 자모)	kum
쿠르드어(아랍어)	ku-arab
쿠르드어(라틴 문자)	ku-latn

[+] 테이블 확장

언어	코드(선택 사항)
쿠루크(데바나가리어)	kru
키르기스어(키릴 자모)	ky
Lak	lbe
라코타어	lkt
라틴어	la
라트비아어	lv
Lezghian	lex
Lingala	ln
리투아니아어	lt
저지 소르브어	dsb
Lozi	loz
룰레 사미어	smj
Luo(케냐 및 탄자니아)	luo
룩셈부르크어	lb
Luyia	luy

언어	코드(선택 사항)
마케도니아어	mk
Machame	jmc
Madurese	mad
마하수 파하리(데바나가리어)	b fz
Makhuwa-Meetto	mgh
Makonde	kde
말라가시어	mg
말레이어(라틴 문자)	ms
몰타어	mt
말토어(데바나가리어)	kmj
Mandinka	mnk
맹크스어	gv
마오리어	mi
Mapudungun	arn
마라티어	mr
Mari(러시아)	chm
Masai	mas
Mende(시에라리온)	men
Meru	mer
Meta'	mgo
Minangkabau	min
모호크어	moh
몽골어(키릴 자모)	mn
Mongondow	mog
몬테네그린어(키릴 자모)	cnr-cyr1
몬테네그린어(라틴어)	cnr-latn

언어	코드(선택 사항)
Morisyen	mfe
Mundang	mua
Nahuatl	nah
Navajo	nv
Ndonga	ng
나폴리어	nap
네팔어	ne
Ngomba	jgo
니우에안어	niu
노가이어	nog
North Ndebele	nd
북부 사미어(라틴어)	sme
노르웨이어	no
Nyanja	ny
Nyankole	yn
Nzima	nzi
오크어	oc
Ojibwa	oj
오로모어	om
오세트어	os
Pampanga	pam
Pangasinan	pag
Papiamento	pap
파슈토어	ps
Pedi	nso
페르시아어	fa

언어	코드(선택 사항)
폴란드어	p1
포르투갈어	pt
편잡어(아랍어)	pa
케추아어	qu
리프아리안어	ksh
루마니아어	ro
로망슈어	rm
Rundi	rn
러시아어	ru

[+] 테이블 확장

언어	코드(선택 사항)
Rwa	rwk
사드리(데바나가리어)	sck
사하어	sah
Samburu	saq
사모아어(라틴어)	sm
Sango	sg
Sangu(가봉)	snq
Sanskrit(데바나가리)	sa
Santali(데바나기리어)	sat
스코틀랜드어	sco
스코틀랜드 게일어	gd
Sena	seh
세르비아어(키릴 자모)	sr-cyrl
세르비아어(라틴 문자)	sr, sr-latn

언어	코드(선택 사항)
Shambala	ksb
쇼나어	sn
Siksika	bla
시르마우리(데바나가리어)	srx
스콜트 사미어	sms
슬로바키아어	sk
슬로베니아어	sl
Soga	xog
소말리어(아랍어)	so
Somali(라틴어)	so-latn
Songhai	son
South Ndebele	nr
Southern Altai	alt
남부 사미어	sma
Southern Sotho	st
스페인어	es
순다어	su
스와힐리어(라틴어)	sw
Swati	ss
스웨덴어	sv
Tabassaran	tab
Tachelhit	shi
타히티어	ty
Taita	dav
타지크어(키릴 자모)	tg
타밀어	ta

언어	코드(선택 사항)
타타르어(키릴 자모)	tt-cyrl
타타르어 (라틴어)	tt
Teso	teo
테툼어	tet
태국어	th
탕미	thf
Tok Pisin	tpi
통가어	to
Tsonga	ts
Tswana	tn
터키어	tr
투르크멘어(라틴어)	tk
투반어	tyv
Udmurt	udm
Uighur(키릴 자모)	ug-cyrl
우크라이나어	uk
북부 슬라브어	hsb
우르두어	ur
위구르어(아랍어)	ug
우즈베키스탄어(아랍어)	uz-arab
우즈베크어(키릴 자모)	uz-cyrl
우즈베크어(라틴 문자)	uz
베트남어	vi
볼라퓌크어	vo
Vunjo	vun
Walser	wae

언어	코드(선택 사항)
웨일스어	cy
서 프리지아어	fy
월라프어	wo
코사어	xh
유카텍 마야어	yua
Zapotec	zap
Zarma	dje
장족어	za
줄루어	zu

일반 문서

① 중요

문서 인텔리전스 v4.0:2023-10-31-preview부터 일반 문서 모델(prebuilt-document)은 레이아웃(prebuilt-layout)에 추가됩니다. 키-값 쌍, 선택 표시, 텍스트, 테이블 및 구조를 추출하려면 다음 모델을 사용하세요.

▣ 테이블 확장

키 값 쌍	version	Model ID
쿼리 문자열 <code>features=keyValuePairs</code> 이 지정된 레이아웃 모델입니다.	<ul style="list-style-type: none"> v4:2023-10-31-preview v3.1:2023-07-31(GA) 	prebuilt-layout
일반 문서 모델	<ul style="list-style-type: none"> v3.1:2023-07-31(GA) v3.0:2022-08-31(GA) 	prebuilt-document

언어 지원: 미리 빌드된 모델

아티클 • 2024. 03. 10.

이 콘텐츠의 적용 대상은 다음과 같습니다. ✓ v4.0(미리 보기) | 이전 버전: ✓ v3.1(GA) ✓ v3.0(GA) ✓ v2.1(GA)

Azure AI 문서 인텔리전스 모델은 다국어 문서 처리 지원을 제공합니다. 언어 지원 기능을 사용하면 사용자가 자연스럽게 애플리케이션과 통신하고 글로벌 지원에 힘을 실어줄 수 있습니다. 미리 빌드된 모델을 사용하면 자체 모델을 학습하고 빌드할 필요 없이 앱 및 흐름에 지능형 도메인 특정 문서 처리를 추가할 수 있습니다. 다음 표에는 모델 및 기능별 사용 가능한 언어 및 로캘 지원이 나와 있습니다.

명함

① 중요

문서 인텔리전스 v4.0(미리 보기)부터 명함 모델(preebuilt-businessCard)은 더 이상 사용되지 않습니다. 명함에서 데이터를 추출하려면 다음을 사용하세요.

☰ 테이블 확장

기능	version	Model ID
명함 모델	<ul style="list-style-type: none">v3.1:2023-07-31(GA)v3.0:2022-08-31(GA)v2.1(GA)	prebuilt-businessCard

Contract

모델 ID: prebuilt-contract

☰ 테이블 확장

언어—로캘 코드	기본값
영어(미국) en-US	영어(미국) en-US

의료 보험 카드

모델 ID: prebuilt-healthInsuranceCard.us

[+] 테이블 확장

언어—로캘 코드	기본값
영어(미국)	영어(미국) en-US

ID 문서

모델 ID: prebuilt-idDocument

지원되는 문서 유형

[+] 테이블 확장

지역	문서 형식
전 세계	여권 수첩, 여권 카드
미국	운전 면허증, 신분증, 거주 허가증(녹색 카드), 사회 보장 카드, 군인 ID
유럽	운전 면허증, 신분증, 거주 허가증
인도	운전 면허증, PAN 카드, 아드하르 카드
캐나다	운전 면허증, 신분증, 거주 허가증(메이플 카드)
오스트레일리아	운전 면허증, 사진 카드, 키 패스 ID(디지털 버전 포함)

송장

모델 ID: prebuilt-invoice

지원되는 언어

[+] 테이블 확장

언어	세부 정보
• 알바니아어(sq)	알바니아(a1)
• 아랍어(ar)	아랍어(ar)

언어	세부 정보
• 불가리아어(bg)	불가리아 (bg)
• 중국어(간체(zh-hans))	중국 (zh-hans-cn)
• 중국어(번체(zh-hant))	홍콩 특별 행정구 (zh-hant-hk), 대만 (zh-hant-tw)
• 크로아티아어(hr)	보스니아 헤르체고비나(ba), 크로아티아(hr), 세르비아 (rs)
• 체코어(cz)	체코 (cz)
• 덴마크어(da)	덴마크 (dk)
• 네덜란드어(nl)	네덜란드 (n1)
• 영어(en)	미국 (us), 오스트레일리아 (au), 캐나다 (ca), 영국 (-uk), 인도 (-in)
• 에스토니아어(et)	에스토니아 (ee)
• 핀란드어(fi)	핀란드 (f1)
• 프랑스어(fr)	프랑스 (fr)
• 독일어(de)	독일 (de)
• 그리스어(el)	그리스 (e1)
• 히브리어(he)	히브리어 (he)
• 헝가리어(hu)	헝가리 (hu)
• 아이슬란드어(is)	아이슬란드 (is)
• 이탈리아어(it)	이탈리아 (it)
• 일본어(ja)	일본 (ja)
• 한국어(ko)	한국 (kr)
• 라트비아어(lv)	라트비아 (lv)
• 리투아니아어(lt)	리투아니아 (lt)
• 마케도니아어(mk)	마케도니아어 (mk)
• 말레이어(ms)	말레이시아 (ms)
• 노르웨이어(nb)	노르웨이 (no)

언어	세부 정보
• 폴란드어(pt)	폴란드(pt)
• 포르투갈어(pt)	포르투갈(pt), 브라질(br)
• 루마니아어(ro)	루마니아(ro)
• 러시아어(ru)	러시아(ru)
• 세르비아어(키릴 자모)(sr-cyr1)	세르비아(sr)
• 세르비아어(sr-Latn)	세르비아(latn-rs)
• 슬로바키아어(sk)	슬로바키아(sv)
• 슬로베니아어(sl)	슬로베니아(sl)
• 스페인어(es)	스페인(es)
• 스웨덴어(sv)	스웨덴(se)
• 태국어(th)	태국(th)
• 터키어(tr)	터키(tr)
• 우크라이나어(uk)	우크라이나(uk)
• 베트남어(vi)	베트남(vi)

Receipt

모델 ID: prebuilt-receipt

열 영수증

[+] 테이블 확장

language_name	언어 코드	language_name	언어 코드
영어	en	리투아니아어	lt
아프리칸스어	af	룩셈부르크어	lb
Akan	ak	마케도니아어	mk

language_name	언어 코드	language_name	언어 코드
알바니아어	sq	말라가시어	mg
아랍어	ar	말레이어	ms
아제르바이잔어	az	몰타어	mt
바마낭칸어	bm	마오리어	mi
바스크어	eu	마라티어	mr
벨로루시어	be	마야, 유카탄	yua
보지푸리	bho	몽골어	mn
보스니아 헤르체고비나어	bs	네팔어	ne
불가리아어	bg	노르웨이어	no
카탈로니아어	ca	Nyanja	ny
세부아노어	ceb	오로모어	om
코르시카어	co	파슈토어	ps
크로아티아어	hr	페르시아어	fa
체코어	cs	페르시아어(다리어)	prs
덴마크어	da	폴란드어	pl
네덜란드어	nl	포르투갈어	pt
에스토니아어	et	편잡어	pa
페로어	fo	케추아어	qu
피지어	fj	루마니아어	ro
필리핀어	fil	러시아어	ru
핀란드어	fi	사모아어	sm
프랑스어	fr	산스크리트어	sa
갈리시아어	gl	스코틀랜드 게일어	gd
Ganda	lg	세르비아어(키릴 자모)	sr-cyr1
독일어	de	세르비아어(라틴 문자)	sr-latn
그리스어	el	세소토어	st

language_name	언어 코드	language_name	언어 코드
과라니어	gn	Sesotho sa Leboa	nso
아이티어	ht	쇼나어	sn
하와이어	haw	슬로바키아어	sk
히브리어	he	슬로베니아어	sl
힌디어	hi	Somali(라틴어)	so-latn
몽다우어	mww	스페인어	es
헝가리어	hu	순다어	su
아이슬란드어	is	스웨덴어	sv
이그보어	ig	타히티어	ty
Iloko	ilo	타지키스탄어	tg
인도네시아어	id	타밀어	ta
아일랜드어	ga	타타르어	tt
코사어	xh	타타르어 (라틴어)	tt-latn
줄루어	zu	태국어	th
이탈리아어	it	통가어	to
일본어	ja	터키어	tr
자바어	jv	투르크멘어	tk
카자흐어	kk	우크라이나어	uk
카자흐어(라틴어)	kk-latn	북부 슬라브어	hsb
키냐르완다어	rw	위구르어	ug
스와힐리어	sw	위구르어(아랍어)	ug-arab
한국어	ko	우즈베크어	uz
쿠르드어	ku	우즈베크어(라틴 문자)	uz-latn
쿠르드어(라틴 문자)	ku-latn	베트남어	vi
키르기스어	ky	웨일스어	cy
라틴어	la	서프리지아어	fy

language_name	언어 코드	language_name	언어 코드
라트비아어	lv	종가어	ts
Lingala	ln		

세금 문서

 테이블 확장

Model ID	언어—로캘 코드	기본값
prebuilt-tax.us.w2	영어(미국)	영어(미국) en-US
prebuilt-tax.us.1098	영어(미국)	영어(미국) en-US
prebuilt-tax.us.1098E	영어(미국)	영어(미국) en-US
prebuilt-tax.us.1098T	영어(미국)	영어(미국) en-US
prebuilt-tax.us.1099	영어(미국)	영어(미국) en-US

언어 지원: 사용자 지정 모델

아티클 • 2024. 04. 08.

이 콘텐츠의 적용 대상은 다음과 같습니다. ✓ v4.0(미리 보기) | 이전 버전: ✓ v3.1(GA) ✓ v3.0(GA) ✓ v2.1(GA)

Azure AI 문서 인텔리전스 모델은 다국어 문서 처리 지원을 제공합니다. 언어 지원 기능을 사용하면 사용자가 자연스럽게 애플리케이션과 통신하고 글로벌 지원에 힘을 실어줄 수 있습니다. 사용자 지정 모델은 레이블이 지정된 데이터 세트를 사용하여 사용 사례와 관련된 고유한 데이터를 추출하도록 학습됩니다. 독립 실행형 사용자 지정 모델을 결합하여 작성된 모델을 만들 수 있습니다. 다음 표에는 모델 및 기능별 사용 가능한 언어 및 로캘 지원이 나와 있습니다.

사용자 지정 분류자

[+] 테이블 확장

언어	코드(선택 사항)
아프리칸스어	af
알바니아어	sq
아랍어	ar
불가리아어	bg
중국어(한자(간체 변형))	zh-Hans
중국어(한자(번체 변형))	zh-Hant
크로아티아어	hr
체코어	cs
덴마크어	da
네덜란드어	nl
에스토니아어	et
핀란드어	fi
프랑스어	fr
독일어	de

언어	코드(선택 사항)
히브리어	he
힌디어	hi
헝가리어	hu
인도네시아어	id
이탈리아어	it
일본어	ja
한국어	ko
라트비아어	lv
리투아니아어	lt
마케도니아어	mk
마라티어	mr
현대 그리스어(1453-)	el
네팔어(매크로 언어)	ne
노르웨이어	no
펀자브어	pa
페르시아어	fa
폴란드어	pl
포르투갈어	pt
루마니아어	rm
러시아어	ru
슬로바키아어	sk
슬로베니아어	sl
소말리어(아랍어)	so
Somali(라틴어)	so-latn
스페인어	es
스와힐리어(매크로 언어)	sw

언어	코드(선택 사항)
스웨덴어	sv
타밀어	ta
태국어	th
터키어	tr
우크라이나어	uk
우르두어	ur
베트남어	vi

사용자 지정 인공신경망

인쇄된 텍스트

다음 표에는 인쇄된 텍스트에 지원되는 언어가 나와 있습니다.

테이블 확장

언어	코드(선택 사항)
아프리칸스어	af
알바니아어	sq
아랍어	ar
불가리아어	bg
중국어 간체	zh-Hans
중국어 번체	zh-Hant
크로아티아어	hr
체코어	cs
덴마크어	da
네덜란드어	nl
에스토니아어	et
핀란드어	fi

언어	코드(선택 사항)
프랑스어	fr
독일어	de
히브리어	he
힌디어	hi
헝가리어	hu
인도네시아어	id
이탈리아어	it
일본어	ja
한국어	ko
라트비아어	lv
리투아니아어	lt
마케도니아어	mk
마라티어	mr
현대 그리스어(1453-)	el
네팔어(매크로 언어)	ne
노르웨이어	no
펀자브어	pa
페르시아어	fa
폴란드어	pl
포르투갈어	pt
루마니아어	rm
러시아어	ru
슬로바키아어	sk
슬로베니아어	sl
소말리어(아랍어)	so
Somali(라틴어)	so-latn

언어	코드(선택 사항)
스페인어	es
스와힐리어(매크로 언어)	sw
스웨덴어	sv
타밀어	ta
태국어	th
터키어	tr
우크라이나어	uk
우르두어	ur
베트남어	vi

인공신경망 모델은 v3.1 이상 API에서 추가된 언어를 지원합니다.

[] 테이블 확장

언어	API 버전
영어	v4.0:2024-02-29-preview, 2023-10-31-preview, v3.1:2023-07-31 (GA), v3.0:2022-08-31 (GA)
독일어	v4.0:2024-02-29-preview, 2023-10-31-preview, v3.1:2023-07-31 (GA)
이탈리아어	v4.0:2024-02-29-preview, 2023-10-31-preview, v3.1:2023-07-31 (GA)
프랑스어	v4.0:2024-02-29-preview, 2023-10-31-preview, v3.1:2023-07-31 (GA)
스페인어	v4.0:2024-02-29-preview, 2023-10-31-preview, v3.1:2023-07-31 (GA)
네덜란드어	v4.0:2024-02-29-preview, 2023-10-31-preview, v3.1:2023-07-31 (GA)

사용자 지정 템플릿

인쇄

다음 표에는 인쇄된 텍스트에 지원되는 언어가 나와 있습니다.

언어	코드(선택 사항)
Abaza	abq
Abkhazian	ab
Achinese	ace
Acoli	ach
Adangme	ada
Adyghe	ady
Afar	aa
아프리칸스어	af
Akan	ak
알바니아어	sq
Algonquin	alq
앙기카(데바나가리어)	anp
아랍어	ar
아스투리아스어	ast
Asu(탄자니아)	asa
Avaric	av
아와디-힌디(데바나가리어)	awa
Aymara	ay
아제르바이잔어(라틴어)	az
Bafia	ksf
Bagheli	bfy
Bambara	bm
바슈키르어	ba
바스크어	eu

언어	코드(선택 사항)
벨로루시어(키릴 자모)	be, be-cyrl
벨로루시어(라틴어)	be, be-latn
Bemba(잠비아)	bem
Bena(탄자니아)	bez
보즈푸리-힌디(데바나가리어)	bho
Bikol	bik
Bini	bin
비슬라마어	bi
보도(데바나가리어)	brx
보스니아어(라틴 문자)	bs
Brajbha	bra
브르타뉴어	br
불가리아어	bg
Bundeli	bns
부랴트어(키릴 자모)	bua
카탈로니아어	ca
세부아노어	ceb
Chamling	rab
차모로어	ch
Chechen	ce
차티스가르히(데바나가리어)	hne
Chiga	cgg
중국어 간체	zh-Hans
중국어 번체	zh-Hant
Choctaw	cho
Chukot	ckt

언어	코드(선택 사항)
Chuvash	cv
콘월어	kw
코르시카어	co
Cree	cr
Creek	mus
크림 타타르어(라틴어)	crh
크로아티아어	시간
Crow	cro
체코어	cs
덴마크어	da
Dargwa	dar
다리어	prs
디말(데바나가리어)	dhi
도그리(데바나가리어)	doi
Duala	dua
Dungan	dng
네덜란드어	nl
Efik	efi
영어	en
에르자어(키릴 자모)	myv
에스토니아어	et
페로어	fo
피지어	fj
필리핀어	fil
핀란드어	fi

언어	코드(선택 사항)
Fon	fon
프랑스어	fr
프리울리안어	fur
Ga	gaa
가가우즈어(라틴어)	gag
갈리시아어	gl
Ganda	lg
Gayo	gay
독일어	de
길버트어	gil
곤디(데바나가리어)	gon
그리스어	el
그린란드어	kl
과라니어	gn
구룽(데바나가리어)	gvr
Gusii	guz
아이티어	ht
할비(데바나가리어)	hlb
하니어	hni
Haryanvi	bgc
하와이어	haw
히브리어	he
Herero	hz
Hiligaynon	hil

언어	코드(선택 사항)
힌디어	hi
몽 다오어(라틴어)	mww
Ho(데바나기리어)	hoc
헝가리어	hu
Iban	iba
아이슬란드어	is
이그보어	ig
Iloko	ilo
이나리 사미어	smn
인도네시아어	id
Ingush	inh
국제어	ia
이누크티투트어 (라틴어)	iu
아일랜드어	ga
이탈리아어	it
일본어	ja
자운사리(데바나가리어)	Jns
자바어	jv
Jola-Fonyi	dyo
Kabardian	kbd
Kabuverdianu	kea
카친어(라틴어)	kac
Kalenjin	kln
Kalmyk	xal
캉그리(데바나가리어)	xnr
Kanuri	kr

언어	코드(선택 사항)
카라차이발카르어	krc
카라칼파크어(키릴 자모)	kaa-cyrl
카라칼파크어(라틴어)	kaa
카슈비아어	csb
카자흐어(키릴 자모)	kk-cyrl
카자흐어(라틴어)	kk-latn
Khakas	kjh
칼링	klr
카시어	kha
키체어	quc
Kikuyu	ki
Kildin Sami	sjd
키냐르완다어	rw
Komi	kv
Kongo	kg
한국어	ko
Korku	kfq
코랴크어	kpy
코스라에어	kos
Kpelle	kpe
Kuanyama	kj
쿠믹어(키릴 자모)	kum
쿠르드어(아랍어)	ku-arab
쿠르드어(라틴 문자)	ku-latn
쿠루크(데바나가리어)	kru
키르기스어(키릴 자모)	ky

언어	코드(선택 사항)
Lak	lbe
라코타어	lkt

[+] 테이블 확장

언어	코드(선택 사항)
라틴어	la
라트비아어	lv
Lezghian	lex
Lingala	ln
리투아니아어	lt
저지 소르브어	dsb
Lozi	loz
룰레 사미어	smj
Luo(케냐 및 탄자니아)	luo
룩셈부르크어	lb
Luyia	luy
마케도니아어	mk
Machame	jmc
Madurese	mad
마하수 파하리(데바나가리어)	bfz
Makhuwa-Meetto	mgh
Makonde	kde
말라가시어	mg
말레이어(라틴 문자)	밀리초
몰타어	mt
말토어(데바나가리어)	kmj

언어	코드(선택 사항)
Mandinka	mnk
맹크스어	gv
마오리어	mi
Mapudungun	arn
마라티어	mr
Mari(러시아)	chm
Masai	mas
Mende(시에라리온)	남자
Meru	mer
Meta'	mgo
Minangkabau	최소
모호크어	moh
몽골어(키릴 자모)	mn
Mongondow	mog
몬테네그린어(키릴 자모)	cnr-cyrl
몬테네그린어(라틴어)	cnr-latn
Morisyen	mfe
Mundang	mua
Nahuatl	nah
Navajo	nv
Ndonga	ng
나폴리어	nap
네팔어	ne
Ngomba	jgo
니우에안어	niu
노가이어	nog

언어	코드(선택 사항)
North Ndebele	nd
북부 사미어(라틴어)	sme
노르웨이어	아니요
Nyanja	ny
Nyankole	yn
Nzima	nzi
오크어	oc
Ojibwa	oj
오로모어	om
오세트어	os
Pampanga	pam
Pangasinan	pag
Papiamento	pap
파슈토어	ps
Pedi	nso
페르시아어	fa
폴란드어	pl
포르투갈어	pt
편집어(아랍어)	pa
케추아어	qu
리프아리안어	ksh
루마니아어	ro
로망슈어	rm
Rundi	rn
러시아어	ru
Rwa	rwk

언어	코드(선택 사항)
사드리(데바나가리어)	sck
사하어	sah
Samburu	saq
사모아어(라틴어)	sm
Sango	sg

 테이블 확장

언어	코드(선택 사항)
Sangu(가봉)	snq
Sanskrit(데바나가리)	sa
Santali(데바나기리어)	sat
스코틀랜드어	sco
스코틀랜드 게일어	gd
Sena	seh
세르비아어(키릴 자모)	sr-cyrl
세르비아어(라틴 문자)	sr, sr-latn
Shambala	ksb
쇼나어	sn
Siksika	bla
시르마우리(데바나가리어)	srx
스콜트 사미어	SMS
슬로바키아어	sk
슬로베니아어	sl
Soga	xog
소말리어(아랍어)	so
Somali(라틴어)	so-latn

언어	코드(선택 사항)
Songhai	son
South Ndebele	nr
Southern Altai	alt
남부 사미어	sma
Southern Sotho	st
스페인어	es
순다어	su
스와힐리어(라틴어)	sw
Swati	ss
스웨덴어	sv
Tabassaran	탭
Tachelhit	shi
타히티어	ty
Taita	dav
타지크어(키릴 자모)	tg
타밀어	ta
타타르어(키릴 자모)	tt-cyril
타타르어 (라틴어)	tt
Teso	teo
테툼어	tet
태국어	th
텅미	thf
Tok Pisin	tpi
통가어	을
Tsonga	ts
Tswana	tn

언어	코드(선택 사항)
터키어	tr
투르크멘어(라틴어)	tk
투반어	tyv
Udmurt	udm
Uighur(키릴 자모)	ug-cyrl
우크라이나어	uk
북부 슬라브어	hsb
우르두어	ur
위구르어(아랍어)	ug
우즈베키스탄어(아랍어)	uz-arab
우즈베크어(키릴 자모)	uz-cyrl
우즈베크어(라틴 문자)	uz
베트남어	vi
볼라퓌크어	vo
Vunjo	vun
Walser	wae
웨일스어	cy
서 프리지아어	fy
월라프어	wo
코사어	xh
유카텍 마야어	yua
Zapotec	zap
Zarma	dje
장족어	za
줄루어	zu

서비스 할당량 및 제한

아티클 • 2024. 03. 12.

이 콘텐츠의 적용 대상은 다음과 같습니다. ✓ v4.0(미리 보기) | 이전 버전: ✓ v3.1(GA) ✓ v3.0(GA)

이 문서에는 모든 [가격 책정 계층](#)에서 Azure AI 문서 인텔리전스 서비스 할당량 및 제한에 대한 빠른 참조 및 자세한 설명이 포함되어 있습니다. 또한 요청 제한을 방지하기 위한 몇 가지 모범 사례가 포함되어 있습니다.

모델 사용

[+] 테이블 확장

지원되는 문서 형식	읽음	레이아웃	미리 빌드된 모델	사용자 지정 모델
PDF	✓	✓	✓	✓
이미지(JPEG/JPG), PNG, BMP, TIFF, HEIF	✓	✓	✓	✓
Office 파일 형식 DOCX, PPTX, XLS	✓	✓	✗	✗

- ✓ 문서 인텔리전스 SDK
- ✓ 문서 인텔리전스 REST API
- ✓ 문서 인텔리전스 Studio v3.0

[+] 테이블 확장

할당량	무료(F0) ¹	Standard(S0)
초당 트랜잭션 수 제한	1	15(기본값)
조정 가능	아니요	예 ²
최대 문서 크기	4MB	500 MB
조정 가능	아니요	아니요
최대 페이지 수(분석)	2	2000
조정 가능	아니요	아니요
레이블 파일의 최대 크기	10MB	10MB
조정 가능	아니요	아니요

할당량	무료(F0) ¹	Standard(S0)
OCR json 응답의 최대 크기	500 MB	500 MB
조정 가능	아니요	아니요
템플릿 모델의 최대 수	500	5,000
조정 가능	아니요	아니요
인공신경망 모델의 최대 수	100	500
조정 가능	아니요	아니요

사용자 지정 모델 사용

- ✓ 사용자 지정 템플릿 모델
- ✓ 사용자 지정 신경망 모델
- ✓ 구성된 분류 모델
- ✓ 구성형 사용자 지정 모델

[+] 테이블 확장

할당량	무료(F0) ¹	Standard(S0)
모델 한도 구성	5	200(기본값)
조정 가능	아니요	아니요
학습 데이터 세트 크기 * 인공신경망	1GB ³	1GB(기본값)
조정 가능	아니요	아니요
학습 데이터 세트 크기 * 템플릿	50MB ⁴	50MB(기본값)
조정 가능	아니요	아니요
최대 페이지 수(학습) * 템플릿	500	500(기본값)
조정 가능	아니요	아니요
최대 페이지 수(학습) * 인공신경망	50,000	50,000(기본값)
조정 가능	아니요	아니요
사용자 지정 신경망 모델 학습	매월 10	매월 20
조정 가능	아니요	예 ³

할당량	무료(F0) ¹	Standard(S0)
최대 페이지 수(학습) * 분류자	10,000	10,000(기본값)
조정 가능	아니요	아니요
최대 문서 형식(클래스) 수 * 분류자	500	500(기본값)
조정 가능	아니요	아니요
학습 데이터 세트 크기 * 분류자	1GB	1GB(기본값)
조정 가능	아니요	아니요
클래스당 최소 샘플 수 * 분류자	5	5(기본값)
조정 가능	아니요	아니요

¹무료(F0) 가격 책정 계층에 대해서는 [가격 책정 페이지](#)에서 월별 허용량도 참조하세요.

²모범 사례 및 [조정 지침](#create-and-submit-support-request)을 참조하세요.

³ 신경망 모델 학습 수는 매월 다시 설정됩니다. 지원 요청을 열어 월별 학습 한도를 늘립니다.

⁴ 이 제한은 레이블 지정 관련 업데이트 전에 학습 데이터 폴더에 있는 모든 문서에 적용됩니다.

자세한 설명, 할당량 조정 및 모범 사례

할당량 증가를 요청하려면(해당하는 경우) 먼저 해당 할당량이 필요한지 확인합니다. Document Intelligence 서비스는 자동 스케일링을 통해 필요한 계산 리소스를 주문형으로 가져오고, 과도한 양의 하드웨어 용량을 유지 관리하지 않음으로써 고객의 비용을 낮게 유지하는 동시에 사용되지 않는 리소스의 프로비전을 해제합니다.

애플리케이션이 응답 코드 429(요청이 너무 많음)를 반환하고 워크로드가 정의된 제한 내에 있는 경우 서비스가 수요에 맞게 스케일 업되었지만 아직 필요한 규모에 도달하지 않았을 가능성이 높습니다. 따라서 요청을 즉시 처리할 수 있을 만큼 충분한 리소스가 서비스에 없습니다. 이 상태는 일시적이며 오래 지속되지 않아야 합니다.

자동 크기 조정 중 제한을 완화하기 위한 일반적인 모범 사례

제한과 관련된 문제(응답 코드 429)를 최소화하려면 다음 방법을 사용하는 것이 좋습니다.

- 애플리케이션에서 다시 시도 논리를 구현합니다.
- 워크로드가 급격히 변경되지 않도록 합니다. 워크로드를 점진적으로 늘립니다.
예제. 애플리케이션이 Document Intelligence를 사용하고 있으며 현재 워크로드는 10TPS(초당 트랜잭션 수)입니다. 그 다음 1초에 로드를 40TPS(즉, 4배)로 늘립니다. 서비스는 새 로드를 충족하기 위해 즉시 확장을 시작하지만 1초 이내에 수행할 수 없으므로 일부 요청에 응답 코드 429가 수신됩니다.

다음 섹션에서는 할당량 조정의 특정 사례에 대해 설명합니다. [Document Intelligence: 동시 요청 제한 증가](#)로 이동

초당 트랜잭션 수 증가 요청 제한

기본적으로 초당 트랜잭션 수는 Document Intelligence 리소스에 대해 초당 15개의 트랜잭션으로 제한됩니다. 표준 가격 책정 계층의 경우 이 크기를 늘릴 수 있습니다. 요청을 제출하기 전에 [이 섹션](#)의 자료를 숙지하고 [모범 사례](#)에 대해 알고 있는지 확인하세요.

동시 요청 제한을 늘려도 비용에 직접적인 영향을 주지 않습니다. Document Intelligence 서비스는 "사용한 만큼만 요금 지불" 모델을 사용합니다. 이 제한은 서비스에서 요청 제한을 시작하기 전에 크기를 조정할 수 있는 수준을 정의합니다.

동시 요청 제한 매개 변수의 기존 값은 Azure Portal, 명령줄 도구 또는 API 요청을 통해 표시되지 않습니다. 기존 값을 확인하려면 Azure 지원 요청을 만듭니다.

초당 트랜잭션을 늘리려면 리소스에서 자동 크기 조정을 사용하도록 설정하면 됩니다. 리소스에서 자동 스케일링을 사용하도록 설정하려면 이 문서 * [자동 스케일링 사용](#)을 참조하세요. TPS 증가 지원 요청을 제출할 수도 있습니다.

필요한 정보 준비

- Document Intelligence 리소스 ID
- 지역
- 정보를 가져오는 방법(기본 모델):
 - [Azure 포털](#)
 - 트랜잭션 제한을 늘리려는 Document Intelligence 리소스를 선택
 - 속성 선택(리소스 관리 그룹)
 - 다음 필드의 값을 복사하고 저장합니다.
 - 리소스 ID
 - 위치(엔드포인트 영역)

지원 요청 만들기 및 제출

지원 요청을 제출하여 리소스에 대한 TPS(초당 트랜잭션 수) 한도 증가를 시작합니다.

- 필요한 정보가 있는지 확인
- Azure 포털 ↗
- TPS 제한을 늘리려는 Document Intelligence 리소스를 선택
- 새 지원 요청(지원 + 문제 해결 그룹) 선택
- Azure 구독 및 Azure 리소스에 대한 정보가 자동으로 채워진 새 창이 표시됨
- 요약 입력(예: "Document Intelligence TPS 제한 증가")
- 문제 유형에서 *"할당량 또는 사용 유효성 검사"를 선택합니다.
- 다음: 솔루션을 선택합니다.
- 요청 만들기 계속 진행
- 세부 정보 탭의 설명 필드에 다음 정보를 입력합니다.
 - 참고로 요청은 문서 인텔리전스 할당량에 관한 것입니다.
 - 충족하도록 스케일링하려는 TPS 기대치를 제공합니다.
 - 수집한 Azure 리소스 정보.
 - 필요한 정보 입력을 완료하고 검토 + 만들기 탭에서 만들기 단추를 클릭합니다.
 - Azure Portal 알림에서 지원 요청 번호를 확인합니다. 추가 처리를 위해 곧 연락을 드릴 것입니다.

워크로드 패턴 모범 사례의 예

이 예에서는 [진행 중인 자동 크기 조정](#)으로 인해 가능한 요청 제한을 완화하기 위해 다음과 같은 권장되는 접근 방식을 제시합니다. 이는 정확한 해결 방법이 아니라 우리가 필요에 맞게 따르고 조정하도록 초대하는 템플릿일 뿐입니다.

Document Intelligence 리소스에 기본 제한이 설정되어 있다고 가정합니다. 워크로드를 시작하여 분석 요청을 제출합니다. 응답 코드 429로 대역폭 제한이 자주 발생하는 경우 GET 분석 응답 요청에 지수 백오프를 구현하여 시작합니다. 연속적인 오류 응답에 대한 다시 시도 사이에 점점 더 긴 대기 시간을 사용합니다(예: 요청 간의 지연 패턴 2-5-13-34). 일반적으로 해당 POST 요청에 대해 2초마다 두 번 이상 분석 가져오기 응답을 호출하지 않는 것이 좋습니다.

제출 중인 문서에 대한 POST 요청 수에 제한이 있는 경우 요청 사이에 지연을 추가하는 것이 좋습니다. 워크로드에 더 높은 수준의 동시 처리가 필요한 경우 초당 트랜잭션에 대한 서비스 제한을 늘리기 위해 지원 요청을 만들어야 합니다.

일반적으로 프로덕션으로 이동하기 전에 워크로드 및 워크로드 패턴을 테스트하는 것이 좋습니다.

다음 단계

오류 코드 및 문제 해결에 대해 알아보기

Azure AI 문서 인텔리전스의 새로운 기능

아티클 • 2024. 02. 21.

이 콘텐츠는 : v4.0(미리 보기) v3.1(GA) v3.0(GA) v2.1(GA)에 적용됩니다.

문서 인텔리전스 서비스는 지속적으로 업데이트됩니다. 릴리스 정보, 향상된 기능, 최신 설명서를 최신 상태로 유지하려면 이 페이지에 책갈피를 지정하세요.

① 중요

미리 보기 API 버전은 GA API가 릴리스되면 재구성됩니다. 미리 보기 API 또는 관련 SDK 버전을 계속 사용하는 경우 2023-02-28-preview API 버전이 사용 중지됩니다. 2023-07-31 API 버전을 대상으로 코드를 업데이트하세요.

2023년 12월

이제 REST API [2023-10-31-preview](#)를 대상으로 하는 문서 인텔리전스 클라이언트 라이브러리를 사용할 수 있습니다.

2023년 11월

이제 Document Intelligence [2023-10-31-preview](#) REST API를 사용할 수 있습니다. 이 미리 보기 API는 다음과 같은 새로운 기능과 업데이트된 기능을 소개합니다.

- 공개 미리 보기 버전 [2023-10-31-preview](#)는 현재 다음 Azure 지역에서만 사용할 수 있습니다.
 - 미국 동부
 - 미국 서부 2
 - 서유럽
- 읽기 모델
 - 필기 언어 확장: 러시아어(`ru`), 아랍어(`ar`), 태국어(`th`).
 - EO(사이버 행정 명령) 규정 준수.
- 레이아웃 모델
 - Office 및 HTML 파일을 지원합니다.
 - Markdown 출력 지원.
 - 테이블 추출, 읽기 순서 및 섹션 제목 검색이 향상되었습니다.
 - Document Intelligence 2023-10-31-preview를 사용하면 일반 문서 모델(미리 빌드된 문서)이 더 이상 사용되지 않습니다. 앞으로 문서에서 키-값 쌍을 추출하려

면 선택적 쿼리 문자열 매개 변수 `features=keyValuePairs` 를 `prebuilt-layout` 사용하도록 설정된 모델을 사용합니다.

- [영수증 모델](#)
 - 이제 모든 가격 관련 필드에 대한 통화를 추출합니다.
- [건강 보험 카드 모델](#)
 - 메디케어 및 메디케이드 정보에 대한 새로운 현장 지원.
- [미국 세금 문서 모델](#)
 - 새로운 1099 세금 모델. 기본 1099 양식 및 다음과 같은 변형을 지원합니다. A, B, C, CAP, DIV, G, H, INT, K, LS, LTC, MISC, NEC, OID, PATR, Q, QA, R, S, SA, SB.
- [청구서 모델](#)
 - 필드에 대한 `KVK` 지원.
 - 필드에 대한 `BPAY` 지원.
 - 다양한 필드 구체화.
- [사용자 지정 분류](#)
 - 다 언어 문서에 대한 지원.
 - 새 페이지 분할 옵션: 자동 맞춤, 항상 페이지로 분할, 분할 없음
- [추가 가능](#)
 - [쿼리 필드는](#) 릴리스에서 `2023-10-31-preview` 사용할 수 있습니다.
 - 추가 기능 기능은 읽기 [모델을 제외한](#) 모든 모델 내에서 사용할 수 있습니다.

① 참고

2022년 8월 31일 API GA(일반 공급) 릴리스부터 관련 미리 보기 API가 더 이상 사용되지 않습니다. 2021-09-30-미리 보기, 2022-01-30-미리 보기 또는 2022-06-30-미리 보기 API 버전을 사용하는 경우 2022-08-31 API 버전을 대상으로 하도록 애플리케이션을 업데이트하세요. 몇 가지 사소한 변경이 관련되어 있습니다. 자세한 내용은 [マイグレーションガイド](#)를 참조하세요.

2023년 7월

① 참고

Form Recognizer는 이제 Azure AI 문서 인텔리전스입니다!

- Document, Azure AI 서비스는 이전에 Cognitive Services 및 Azure Applied AI Services로 알려진 모든 서비스를 포함합니다.
- 가격 책정에는 변화가 없습니다.
- *Cognitive Services* 및 *Azure Applied AI*라는 이름은 Azure 청구, 비용 분석, 가격 목록 및 가격 API에서 계속 사용됩니다.
- API(애플리케이션 프로그래밍 인터페이스) 또는 클라이언트 라이브러리에 대한 주요 변경 내용은 없습니다.
- 일부 플랫폼은 여전히 이름 변경 업데이트를 기다리고 있습니다. 설명서에서 Form Recognizer 또는 문서 인텔리전스에 대한 모든 언급은 동일한 Azure 서비스를 나타냅니다.

문서 인텔리전스 v3.1(GA)

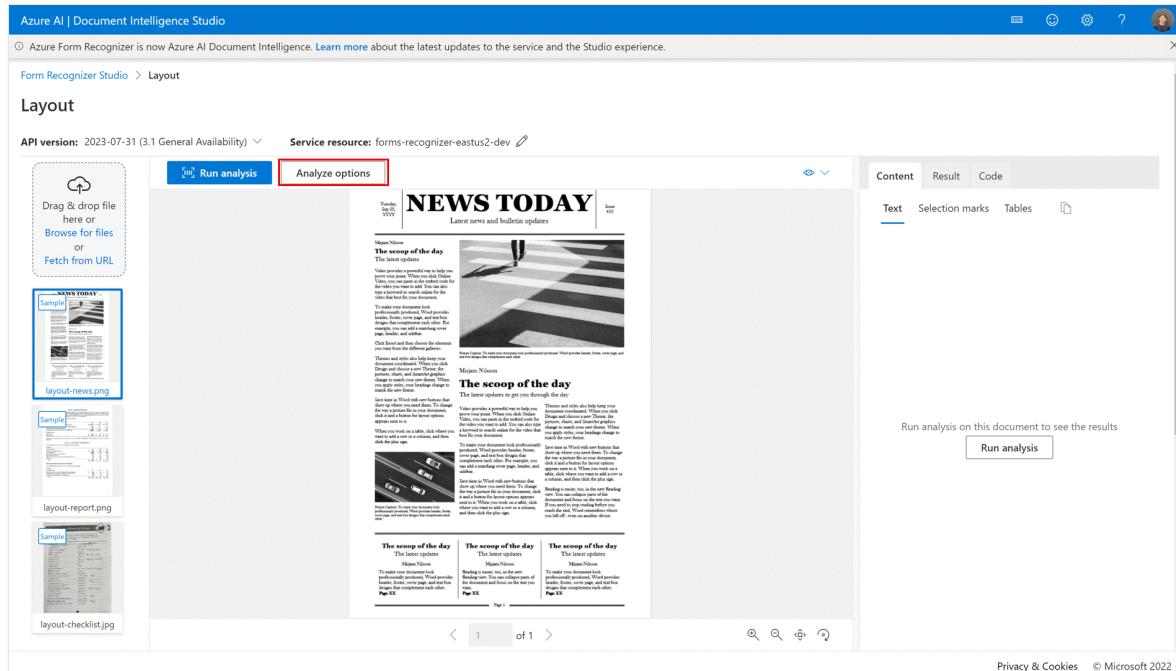
문서 인텔리전스 버전 3.1 API가 이제 일반 공급(GA)되었습니다! API 버전은 2023-07-31에 해당합니다. v3.1 API에는 새로운 기능과 업데이트된 기능이 도입되었습니다.

- 이제 문서 인텔리전스 API가 모듈화되었으며 선택적 기능을 지원합니다. 이제 필요한 기능을 구체적으로 포함하도록 출력을 사용자 지정할 수 있습니다. [선택 매개변수](#)에 대해 자세히 알아보세요.
- 단일 파일을 개별 문서로 분할하기 위한 문서 분류 API입니다. 문서 분류에 대해 [자세히 알아보세요](#).
- 미리 빌드된 계약 모델입니다.
- 미리 빌드된 미국 세금 양식 1098 모델.
- 읽기 API를 사용하는 Office 파일 형식 지원
- [문서의 바코드 인식](#)입니다.
- 수식 인식 [추가 기능입니다](#).
- 글꼴 인식 [추가 기능입니다](#).
- [고해상도 문서를 지원합니다](#).
- 이제 사용자 지정 신경망 모델을 학습하려면 레이블이 지정된 단일 샘플이 필요합니다.
- 사용자 지정 인공신경망 모델 언어 확장. 30개 언어로 된 문서에 대한 인공신경망 모델을 학습합니다. 지원되는 언어의 전체 목록은 언어 지원을 [참조하세요](#).
- NEW [사전 빌드된 건강 보험 카드 모델](#).
- [사전 빌드된 청구서 모델 언어 확장](#).
- 100개 이상의 언어가 지원되는 [사전 빌드된 영수증 모델 언어 및 로캘 확장](#)
- [사전 빌드된 ID 모델](#)은 이제 유럽 ID를 지원합니다.

문서 인텔리전스 스튜디오 UX 업데이트

✓ 옵션 분석

- 문서 인텔리전스는 이제 보다 정교한 분석 기능을 지원하며 스튜디오에서는 추가 기능을 쉽게 구성할 수 있는 하나의 진입점(분석 옵션 버튼)을 허용합니다.
- 문서 추출 시나리오에 따라 분석 범위, 문서 페이지 범위, 선택적 탐지, 프리미엄 탐지 기능을 구성합니다.



① 참고

문서 인텔리전스 스튜디오에서는 글꼴 추출이 시각화되지 않습니다. 그러나 글꼴 검색 결과에 대한 JSON 출력의 스타일 섹션을 검사 수 있습니다.

✓ 사전 제작된 모델 또는 자체 모델 중 하나를 사용하여 문서에 자동 레이블 지정

- 이제 사용자 지정 추출 모델 레이블 지정 페이지에서 이전에 학습한 Document Intelligent Service 미리 빌드된 모델 또는 모델 중 하나를 사용하여 문서에 자동으로 레이블을 지정할 수 있습니다.

- 일부 문서의 경우 자동 레이블을 실행한 후 중복 레이블이 있을 수 있습니다. 이후 레이블 지정 페이지에 중복 레이블이 없도록 레이블을 수정하시기 바랍니다.

Labels	Related fields
New Belgium Brewery	<ul style="list-style-type: none"> ● CustomerAddressRecipient ● CustomerName

✓ 자동 레이블 지정 표

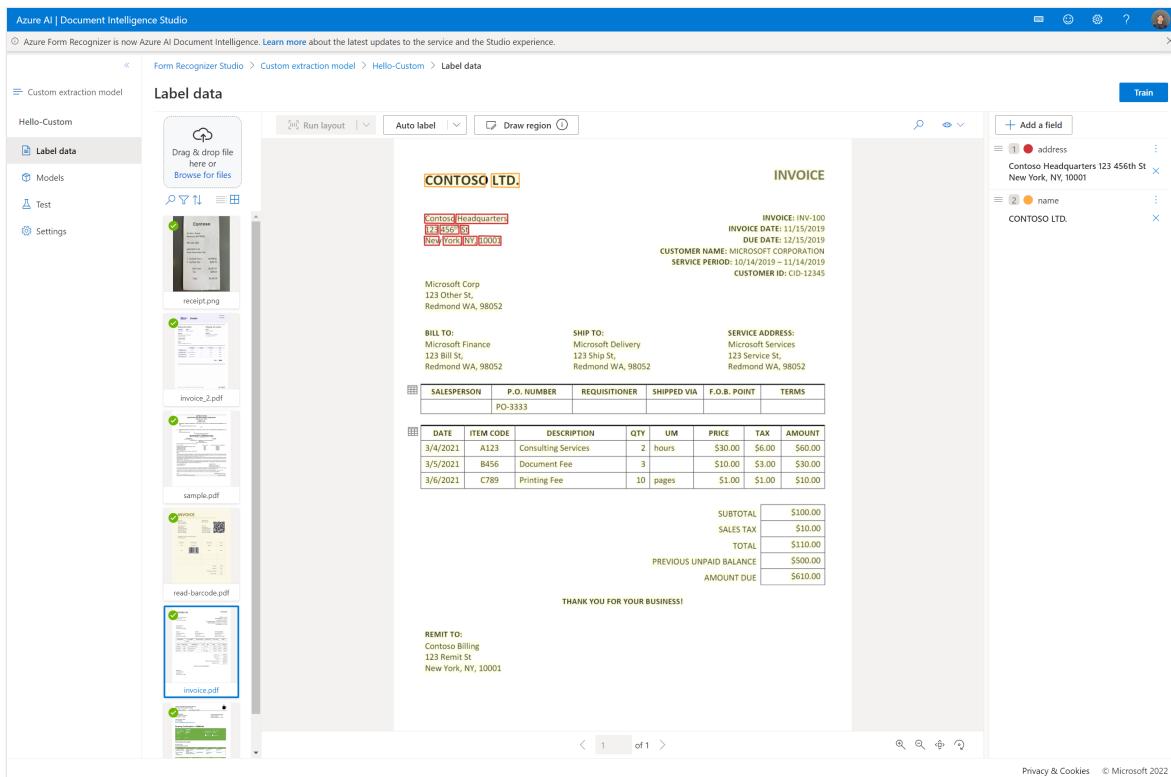
- 사용자 지정 추출 모델 레이블 지정 페이지에서 이제 테이블에 수동으로 레이블을 지정할 필요 없이 문서의 테이블에 자동으로 레이블을 지정할 수 있습니다.

✓ 학습 데이터 세트에 직접 테스트 파일 추가

- 사용자 지정 추출 모델을 학습한 후에는 테스트 페이지를 사용하여 필요한 경우 테스트 문서를 학습 데이터 세트에 업로드하여 모델 품질을 향상시킵니다.
- 일부 레이블에 대해 낮은 신뢰도 점수가 반환되면 레이블이 올바르게 지정되었는지 확인하세요. 그렇지 않은 경우 학습 데이터 세트에 추가하고 레이블을 다시 지정하여 모델 품질을 향상시키세요.

✓ 맞춤 프로젝트에서 문서 목록 옵션 및 필터를 활용하세요.

- 사용자 지정 추출 모델 레이블 지정 페이지의 경우 이제 기능별로 검색, 필터링 및 정렬을 사용하여 학습 문서를 쉽게 탐색할 수 있습니다.
- 그리드 보기를 활용하여 문서를 미리 보거나 목록 보기 사용하여 문서를 더 쉽게 스크롤할 수 있습니다.



✓ 프로젝트 공유

- 사용자 지정 추출 프로젝트를 쉽게 공유하세요. 자세한 내용은 [맞춤 모델과 프로젝트 공유](#)를 참조하세요.

2023년 5월

빌드 2023에 대한 새로워진 문서 소개

- NEW 문서 인텔리전스 개요** 향상된 탐색, 구조적 액세스 지점 및 보강된 이미지
- NEW 문서 인텔리전스 모델 선택**은 프로젝트 및 워크플로에 가장 적합한 문서 인텔리전스 솔루션을 선택하기 위한 지침을 제공합니다.

2023년 4월

최신 문서 인텔리전스 클라이언트 라이브러리 공개 미리 보기 릴리스 발표

- Document Intelligence REST API 버전 2023-02-28-preview [☞](#) 는 공개 미리 보기 릴리스 클라이언트 라이브러리를 지원합니다. 이 릴리스에는 .NET/C#(4.1.0-beta-1), Java(4.1.0-beta-1), JavaScript(4.1.0-beta-1) 및 Python(3.3.0b1) 클라이언트 라이브러리에 사용할 수 있는 다음과 같은 새로운 기능과 기능이 포함되어 있습니다.
 - 사용자 지정 분류 모델
 - 쿼리 필드 추출
 - 추가 기능
- 자세한 내용은 *Document Intelligence SDK(공개 미리 보기)* 및 [2023년 3월 릴리스 정보](#)를 참조하세요.

2023년 3월

① 중요

[2023-02-28-preview](#) [☞](#) 기능은 현재 다음 지역에서만 사용할 수 있습니다.

- 서유럽
- 미국 서부2
- 미국 동부

- **사용자 지정 분류 모델**은 2023-02-28-preview API로 시작하는 문서 인텔리전스의 새로운 기능입니다. [문서 인텔리전스 스튜디오](#) [☞](#) 또는 [REST API](#) [☞](#) 를 사용하여 문서 분류 기능을 사용해 보세요.
- 일반 문서 모델에 추가된 **쿼리 필드** 기능은 Azure OpenAI 모델을 사용하여 문서에서 특정 필드를 추출합니다. [문서 인텔리전스 스튜디오](#) [☞](#) 를 사용하여 **쿼리 필드가 있는 일반 문서** 기능을 사용해 보세요. 쿼리 필드는 현재 해당 지역의 리소스에 **East US** 대해서만 활성화되어 있습니다.
- **추가 기능:**
 - [이제 API를 사용하여 글꼴 추출](#) 이 2023-02-28-preview 인식됩니다.
 - [이제 API를 사용하여 수식 추출](#) 이 2023-02-28-preview 인식됩니다.
 - [이제 API를 사용하여 고해상도 추출](#) 이 2023-02-28-preview 인식됩니다.
- **사용자 지정 추출 모델 업데이트:**
 - [이제 사용자 지정 신경망 모델은](#) 학습 및 분석을 위해 추가된 언어를 지원합니다. 네덜란드어, 프랑스어, 독일어, 이탈리아어 및 스페인어용 신경망 모델을 학습합니다.
 - [이제 사용자 지정 템플릿 모델에](#) 향상된 서명 검색 기능이 있습니다.
- **Document Intelligence Studio** [☞](#) 업데이트:

- 분류 및 쿼리 필드와 같은 모든 새로운 기능을 지원하는 것 외에도 Studio는 이제 사용자 지정 모델 프로젝트에 대한 프로젝트 공유를 사용하도록 설정합니다.
 - 제한적 미리 보기에 새로운 모델 추가: **예방 접종 카드**, **계약서**, **미국 세금 1098**, **미국 세금 1098-E** 및 **미국 세금 1098-T**. 제한적 미리 보기 모델에 대한 액세스를 요청하려면 **문서 인텔리전스 비공개 미리 보기 요청 양식** 을 작성하여 제출하세요.
- **영수증 모델 업데이트:**
 - 영수증 모델은 열 영수증에 대한 지원을 추가합니다.
 - 영수증 모델은 이제 18개 언어와 세 개의 지역 언어(영어, 프랑스어, 포르투갈어)에 대한 언어 지원을 추가합니다.
 - 이제 영수증 모델이 추출을 지원합니다 **TaxDetails**.
 - **이제 레이아웃 모델이** 테이블 인식을 향상시킵니다.
 - **이제 읽기 모델**에서 한 자리 문자 인식에 대한 개선이 추가되었습니다.
-

2023년 2월

- 이제 v3.0용 Select 문서 인텔리전스 컨테이너를 사용할 수 있습니다!
- 현재 읽기 v3.0 및 레이아웃 v3.0 컨테이너를 사용할 수 있습니다.

자세한 내용은 **문서 인텔리전스 컨테이너 설치 및 실행을 참조하세요**.

2023년 1월

- 미리 빌드된 영수증 모델 - 지원되는 언어가 추가되었습니다. 이제 영수증 모델은 이러한 추가된 언어 및 로캘을 지원합니다.
 - 일본어 - 일본(ja-JP)
 - 프랑스어 - 캐나다(fr-CA)
 - 네덜란드어 - 네덜란드어(nl-NL)
 - 영어 - 아랍에미리트(en-AE)
 - 포르투갈어 - 브라질(pt-BR)
- 미리 빌드된 청구서 모델 - 지원되는 언어가 추가되었습니다. 이제 청구서 모델은 이러한 추가된 언어 및 로캘을 지원합니다.
 - 영어 - 미국(en-US), 오스트레일리아(en-AU), 캐나다(en-CA), 영국(en-UK), 인도(en-IN)
 - 스페인어 - 스페인(es-ES)
 - 프랑스어 - 프랑스(fr-FR)
 - 이탈리아어 - 이탈리아(it-IT)
 - 포르투갈어 - 포르투갈(pt-PT)

- 네덜란드어 - 네덜란드어(nl-NL)
- 미리 빌드된 청구서 모델 - 인식된 필드가 추가되었습니다. 이제 청구서 모델에서 이러한 추가 필드를 인식합니다.
 - 통화 코드
 - 지불 옵션
 - 총 할인
 - 세금 항목(en-IN에만 해당)
- 미리 빌드된 ID 모델 - 지원되는 문서 형식이 추가되었습니다. 이제 ID 모델은 이러한 추가된 문서 형식을 지원합니다.
 - 미국 군사 ID

💡 팁

모든 2023년 1월 업데이트는 REST API 버전 2022-08-31(GA) 에서 사용할 수 있습니다.

• 미리 빌드된 영수증 모델 - 추가 언어 지원:

- 미리 빌드된 영수증 모델은 다음 언어에 대한 지원을 추가합니다.
- 영어 - 아랍에미리트(en-AE)
 - 네덜란드어 - 네덜란드어(nl-NL)
 - 프랑스어 - 캐나다(fr-CA)
 - 독일어 - (de-DE)
 - 이탈리아어 - (it-IT)
 - 일본어 - 일본(ja-JP)
 - 포르투갈어 - 브라질(pt-BR)

• 미리 빌드된 청구서 모델 - 추가 언어 지원 및 필드 추출

미리 빌드된 청구서 모델은 다음 언어에 대한 지원을 추가합니다.

- 영어 - 오스트레일리아(en-AU), 캐나다(en-CA), 영국(en-UK), 인도(en-IN)
- 포르투갈어 - 브라질(pt-BR)

이제 미리 빌드된 청구서 모델이 다음 필드 추출에 대한 지원을 추가합니다.

- 통화 코드
- 지불 옵션
- 총 할인
- 세금 항목(en-IN에만 해당)

• 미리 빌드된 ID 문서 모델 - 추가 문서 형식 지원

이제 미리 빌드된 ID 문서 모델이 다음 문서 형식에 대한 지원을 추가합니다.

- 인도, 캐나다, 영국 및 호주를 지원하는 운전 면허증 확장
 - 미군 ID 카드 및 문서
 - 인도 ID 카드 및 문서(PAN 및 Aadhaar)
 - 오스트레일리아 ID 카드 및 문서(사진 카드, 키 패스 ID)
 - 캐나다 ID 카드 및 문서(id 카드, Maple 카드)
 - 영국 신분증 및 서류(국가/지역 신분증)
-

2022년 12월

- **문서 인텔리전스 스튜디오 업데이트 ↗**

12월 문서 인텔리전스 스튜디오 릴리스에는 문서 인텔리전스 스튜디오에 대한 최신 업데이트가 포함되어 있습니다. 사용자 환경은 사용자 지정 모델 레이블 지정 지원을 통해 크게 개선되었습니다.

- **페이지 범위입니다.** 이제 Studio에서 문서에서 지정된 페이지 분석을 지원합니다.
- **사용자 지정 모델 레이블 지정:**
 - **레이아웃 API를 자동으로 실행합니다.** 사용자 지정 모델에 대한 설정 프로세스 중에 Blob Storage의 모든 문서에 대해 레이아웃 API를 자동으로 실행하도록 선택할 수 있습니다.
 - **검색.** 이제 Studio에는 문서 내에서 단어를 찾는 검색 기능이 포함되어 있습니다. 이렇게 개선하면 레이블을 지정하는 동안 탐색이 더 쉬워질 수 있습니다.
 - **탐색.** 레이블을 선택하여 문서 내에서 레이블이 지정된 단어를 대상으로 지정 할 수 있습니다.
 - **자동 테이블 레이블 지정** 문서 내에서 테이블 아이콘을 선택한 후 레이블 지정 보기에서 추출된 테이블의 레이블을 자동으로 지정하도록 선택할 수 있습니다.
 - **레이블 하위 형식 및 두 번째 수준 하위 형식** The Studio는 이제 날짜 및 숫자와 같은 형식에 대해 테이블 열, 테이블 행 및 두 번째 수준 하위 형식에 대한 하위 형식을 지원합니다.
- 이제 US Gov 버지니아 지역에서 사용자 지정 신경망 모델 빌드가 지원됩니다.
- 미리 보기 API 버전 `2022-01-30-preview` 이며 `2021-09-30-preview` 2023년 1월 31일에 사용 중지됩니다. 서비스 중단을 `2022-08-31` 방지하도록 API 버전으로 업데이트

합니다.

2022년 11월

- Azure AI 문서 인텔리전스 라이브러리의 최신 안정 릴리스 발표
 - 이 릴리스에는 .NET, Java, JavaScript 및 Python 클라이언트 라이브러리에 대한 중요한 변경 내용과 업데이트가 포함되어 있습니다. 자세한 내용은 [Azure SDK DevBlog](#)을 참조하세요.
 - 가장 중요한 향상된 기능은 두 개의 새 클라이언트 `DocumentAnalysisClient` `DocumentModelAdministrationClient`와 두 클라이언트의 도입입니다.

2022년 10월

- 문서 인텔리전스 버전이 지정된 콘텐츠
 - 버전이 지정된 환경을 표시하도록 문서 인텔리전스 설명서가 업데이트됩니다. 이제 `v3.0 GA` 환경 또는 `v2.1 GA` 환경을 대상으로 지정하는 콘텐츠를 보도록 선택할 수 있습니다. `v3.0` 환경이 기본값입니다.

The screenshot shows the Microsoft Ignite website for October 12-14, 2022. The navigation bar includes links for Microsoft Learn, Documentation, Training, Certifications, Q&A, Code Samples, Shows, and Events. The main content area is for the Azure Form Recognizer documentation. A sidebar on the left has a dropdown menu for 'Version' with options 'Form Recognizer v3.0', 'v3.0', and 'v2.1'. The 'v3.0' option is highlighted with a red box. The main content area includes sections for 'About Azure Form Recognizer', 'Quickstarts and how-to-guides', and 'Form Recognizer Studio (v3.0)'. The 'About Azure Form Recognizer' section contains links for 'OVERVIEW' and 'What is Azure Form Recognizer?'. The 'Quickstarts and how-to-guides' section contains links for 'QUICKSTART', 'Form Recognizer Studio', and 'Form Recognizer Sample Labeling tool'. The 'Form Recognizer Studio (v3.0)' section contains links for 'OVERVIEW' and 'Studio overview'.

- 문서 인텔리전스 스튜디오 샘플 코드
 - 이제 [문서 인텔리전스 스튜디오 레이블 지정 환경](#)의 샘플 코드를 GitHub에서 사용할 수 있습니다. 고객은 문서 인텔리전스를 개발하여 자신의 UX에 통합하거나 문서 인텔리전스 스튜디오 샘플 코드를 사용하여 자신만의 새로운 UX를 빌드 할 수 있습니다.
- 언어 확장

- 최신 미리 보기 릴리스를 통해 문서 인텔리전스의 OCR(읽기), 레이아웃 및 사용자 지정 템플릿 모델은 134개의 새로운 언어를 지원합니다. 이러한 언어 추가에는 그리스어, 라트비아어, 세르비아어, 태국어, 우크라이나어 및 베트남어와 여러 라틴어 및 키릴 자모 언어가 포함됩니다. 문서 인텔리전스는 이제 최신 GA 및 새 미리 보기 버전에서 총 299개 언어를 지원합니다. 지원되는 모든 언어를 보려면 [지원되는 언어](#) 페이지를 참조하세요.
 - API 또는 해당 SDK를 사용하여 애플리케이션의 새 언어를 지원하는 경우 REST API 매개 변수 `api-version=2022-06-30-preview`를 사용합니다.
- 새 미리 빌드된 계약 모델
 - 당사자, 타이틀, 계약 ID, 실행 날짜 등의 정보를 계약에서 추출하는 새로운 미리 빌드된 모델입니다. 계약 모델은 현재 미리 보기로 제공되며 [여기](#)에서 액세스를 요청합니다.
 - 사용자 지정 신경망 모델 학습을 위한 지역 확장
 - 이제 추가 지역에서 사용자 지정 신경망 모델 학습이 지원됩니다.
 - ✓ 미국 동부
 - ✓ 미국 동부2
 - ✓ US Gov 애리조나

2022년 9월

① 참고

버전 4.0.0부터 문서 인텔리전스 서비스의 최신 기능을 활용하기 위해 새로운 클라이언트 세트가 도입되었습니다.

SDK 버전 4.0.0 GA 릴리스에는 다음 업데이트가 포함되어 있습니다.

C#

- 버전 4.0.0 GA(2022-09-08)
- REST API v3.0 및 v2.0 클라이언트 지원

[패키지\(NuGet\)](#)

[변경 로그/릴리스 기록](#)

[마이그레이션 가이드](#)

[추가 정보](#)

샘플 ↗

- 사용자 지정 신경망 모델 학습을 위한 지역 확장은 이제 6개의 새 지역에서 지원됩니다.
 - ✓ 오스트레일리아 동부
 - ✓ 미국 중부
 - ✓ 동아시아
 - ✓ 프랑스 중부
 - ✓ 영국 남부
 - ✓ 미국 서부2
- 학습이 지원되는 지역의 전체 목록은 [사용자 지정 인공신경망 모델](#)을 참조하세요.
- Document Intelligence SDK 버전 4.0.0 GA 릴리스:
 - Document Intelligence 클라이언트 라이브러리 버전 4.0.0(.NET/C#, Java, JavaScript) 및 버전 3.2.0(Python)은 일반적으로 사용할 수 있으며 프로덕션 애플리케이션에서 사용할 준비가 되어 있습니다!.
 - 문서 인텔리전스 클라이언트 라이브러리에 대한 자세한 내용은 [SDK 개요](#)를 참조하세요.
 - 프로그래밍 언어의 [マイグレーションガイド](#)를 사용하여 애플리케이션을 업데이트합니다.

2022년 8월

문서 인텔리전스 SDK 베타 2022년 8월 미리 보기 릴리스에는 다음 업데이트가 포함되어 있습니다.

C#

버전 4.0.0-beta.5(2022-08-09)

[변경 로그/릴리스 기록 ↗](#)

[패키지\(NuGet\) ↗](#)

[SDK 참조 설명서](#)

- 문서 인텔리전스 v3.0 정식 출시
 - 문서 인텔리전스 REST API v3.0이 이제 정식 출시되어 프로덕션 애플리케이션에서 사용할 수 있습니다! [REST API 버전 2022-08-31](#)로 애플리케이션을 업데이트

합니다.

- 문서 인텔리전스 스튜디오 업데이트
 - ✓ **다음 단계.** 이제 스튜디오의 각 모델 페이지에는 다음 단계 섹션이 있습니다. 사용자는 샘플 코드, 문제 해결 지침 및 가격 정보를 빠르게 참조할 수 있습니다.
 - ✓ **사용자 지정 모델.** 이제 스튜디오에는 레이블 지정 효율성을 개선하기 위해 사용자 지정 모델 프로젝트에서 레이블을 다시 정렬하는 기능이 포함되어 있습니다.
 - ✓ **모델 복사** 사용자 지정 모델은 스튜디오 내에서 문서 인텔리전스 서비스 간에 복사할 수 있습니다. 이 작업을 통해 학습된 모델을 다른 환경 및 지역으로 승격 할 수 있습니다.
 - ✓ **문서 삭제.** 이제 스튜디오가 사용자 지정 프로젝트 내에서 레이블이 지정된 데이터 세트에서의 문서 삭제를 지원합니다.
- 문서 인텔리전스 서비스 업데이트
 - **prebuilt-read.** 읽기 OCR 모델은 이제 두 가지 새로운 기능인 단락 및 언어 탐지 와 함께 문서 인텔리전스에서도 사용할 수 있습니다. 문서 인텔리전스 Read는 문서 인텔리전스의 광범위한 문서 인텔리전스 기능과 연계된 고급 문서 시나리오 를 대상으로 합니다.
 - **prebuilt-layout.** 레이아웃 모델은 단락을 추출하며, 추출된 텍스트는 단락, 제목, 섹션 머리글, 각주, 페이지 머리글, 페이지 바닥글 또는 페이지 번호입니다.
 - **prebuilt-invoice.** 이제 TotalVAT 및 Line/VAT 필드가 각각 기존 필드 TotalTax 및 Line/Tax로 확인됩니다.
 - **prebuilt-idDocument.** 미국 주 ID, 사회 보장 및 그린 카드에 대한 데이터 추출 지원. 여권 비자 정보 지원.
 - **prebuilt-receipt.** 프랑스어(fr-FR), 스페인어(es-ES), 포르투갈어(pt-PT), 이탈리아 어(it-IT) 및 독일어(de-DE)에 대한 로캘 지원이 확장되었습니다.
 - **prebuilt-businessCard.** 주소, 도시, 주, 국가/지역 및 우편번호와 같은 주소 구성 요소에 대한 하위 필드를 추출하는 주소 구문 분석 지원.
- AI 품질 향상
 - **prebuilt-read.** 단일 문자, 필기 날짜, 금액, 이름, 영수증 및 송장에 일반적으로 있는 기타 주요 데이터에 대한 지원이 향상되고 디지털 PDF 문서의 처리가 향상되었습니다.
 - **prebuilt-layout.** 잘린 표, 테두리 없는 테이블 및 긴 스패닝 셀의 향상된 인식에 대한 향상된 검색을 지원합니다.
 - **prebuilt-document.** 값 및 확인란 검색이 향상되었습니다.
 - **custom-neural.** 테이블 검색 및 추출에 대한 정확도가 향상되었습니다.

- 문서 인텔리전스 SDK 베타 2022년 6월 미리 보기 릴리스에는 다음 업데이트가 포함되어 있습니다.

C#

버전 4.0.0-beta.4(2022-06-08)

[변경 로그/릴리스 기록](#)

[패키지\(NuGet\)](#)

[SDK 참조 설명서](#)

- 문서 인텔리전스 스튜디오 [6월 릴리스](#)는 문서 인텔리전스 스튜디오의 최신 업데이트입니다. 이 업데이트에서는 사용자 환경과 손쉬운 사용 부분이 크게 향상되었습니다.
 - JavaScript 및 C#에 대한 코드 샘플입니다. 이제 스튜디오 코드 탭은 기존 Python 샘플 외에도 JavaScript 및 C# 코드 샘플을 추가합니다.
 - 새 문서 업로드 UI. 이제 Studio는 새 업로드 사용자 인터페이스로 끌어서 놓기를 사용하여 문서 업로드를 지원합니다.
 - 사용자 지정 프로젝트를 위한 새로운 기능. 이제 사용자 지정 프로젝트는 프로젝트를 구성할 때 스토리지 계정 및 Blob 만들기를 지원합니다. 또한 사용자 지정 프로젝트는 이제 Studio 내에서 직접 학습 파일 업로드 및 기존 사용자 지정 모델 복사를 지원합니다.
- 문서 인텔리전스 v3.0 2022-06-30-preview 릴리스에서는 기능 API 전반에 걸쳐 광범위한 업데이트를 제공합니다.
 - 레이아웃은 구조 추출을 확장합니다. 이제 레이아웃에는 섹션, 섹션 머리글 및 단락을 포함한 추가된 구조 요소가 포함됩니다. 이 업데이트를 사용하면 문서 세분화 시나리오가 더 세분화됩니다. 식별된 구조 요소의 전체 목록은 [향상된 구조를 참조하세요](#).
 - 사용자 지정 신경망 모델 테이블 형식 필드가 지원됩니다. 이제 사용자 지정 문서 모델이 테이블 형식 필드를 지원합니다. 기본적으로 테이블 형식 필드는 다중 페이지에서도 합니다. 사용자 지정 신경망 모델의 테이블 형식 필드에 대한 자세한 내용은 [테이블 형식 필드를 참조하세요](#).
 - 페이지 간 테이블에 대한 사용자 지정 템플릿 모델 테이블 형식 필드를 지원합니다. 이제 사용자 지정 양식 모델이 여러 페이지에서 테이블 형식 필드를 지원합니다. 사용자 지정 템플릿 모델의 테이블 형식 필드에 대한 자세한 내용은 [테이블 형식 필드를 참조하세요](#).
 - 이제 청구서 모델 출력에는 일반 문서 키-값 쌍이 포함됩니다. 청구서에 미리 빌드된 모델에 포함된 필드 이외의 필수 필드가 포함된 경우 일반 문서 모델은 출력을 키-값 쌍으로 보완합니다. [키/값 쌍을 참조하세요](#).

- **청구서 언어 확장**. 청구서 모델에는 확장된 언어 지원이 포함됩니다. 지원되는 언어를 [참조하세요](#).
- **미리 빌드된 명함**에는 이제 일본어 지원이 포함됩니다. 지원되는 언어를 [참조하세요](#).
- **미리 빌드된 ID 문서 모델**. 이제 ID 문서 모델은 미국 운전 면허증에서 DateOfIssue, Height, Weight, EyeColor, HairColor 및 DocumentDiscriminator를 추출합니다. 필드 추출을 [참조하세요](#).
- **이제 읽기 모델은 일반적인 Microsoft Office 문서 형식을 지원합니다.** Word(docx), Excel(xlsx), PowerPoint(pptx)와 같은 문서 형식은 이제 읽기 API에서 지원됩니다. 데이터 추출 읽기를 [참조하세요](#).

2022년 2월

C#

버전 4.0.0-beta.3(2022-02-10)

[변경 로그/릴리스 기록](#)

[패키지\(NuGet\)](#)

[SDK 참조 설명서](#)

- Document Intelligence v3.0 미리 보기 릴리스에는 다음과 같은 몇 가지 새로운 기능, 기능 및 향상된 기능이 도입되었습니다.
 - **사용자 지정 신경망 모델** 또는 사용자 지정 문서 모델은 구조화된 양식, 반정형 및 구조화되지 않은 문서에서 텍스트 및 선택 표시를 추출하는 새로운 사용자 지정 모델입니다.
 - **W-2 미리 빌드된 모델**은 세금 보고 및 소득 확인 시나리오를 위해 W-2 양식에서 필드를 추출하는 새로운 미리 빌드된 모델입니다.
 - **읽기 API**는 검색된 경우 출력된 텍스트 줄, 단어, 텍스트 위치, 검색된 언어 및 필기 텍스트를 추출합니다.
 - **이제 양식 및 문서의 API 텍스트, 테이블, 구조 및 키-값 쌍 외에도 선택 표시를 지원하도록 일반 문서** 미리 학습된 모델이 업데이트되었습니다.
 - **청구서 API** 미리 빌드된 청구서 모델은 스페인어 청구서를 지원하도록 확장합니다.
 - **문서 인텔리전스 스튜디오**는 읽기, W2, 호텔 영수증 샘플에 대한 새로운 데모를 추가하고 새로운 사용자 지정 인공신경망 모델 학습을 지원합니다.
 - **언어 확장** 문서 인텔리전스 읽기, 레이아웃 및 사용자 지정 양식은 아랍어 및 데바나가리 스크립트를 사용하여 아랍어, 힌디어 및 기타 언어를 포함한 42개의 새

로운 언어에 대한 지원을 추가하여 적용 범위를 164개 언어로 확장합니다. 필기 언어 지원은 일본어와 한국어로 확장됩니다.

- v3.0 미리 보기 API에 대한 새로운 REST API [↗](#), Python 또는 .NET SDK를 시작합니다.
- 문서 인텔리전스 모델 데이터 추출:

테이블 확장

모델	텍스트 추출	키/값 쌍	선택 표시	테이블	시그니처
읽음	✓				
일반 문서	✓	✓	✓	✓	
Layout	✓		✓	✓	
송장	✓	✓	✓	✓	
Receipt	✓	✓			✓
ID 문서	✓	✓			
명함	✓	✓			
사용자 지정 템플릿	✓	✓	✓	✓	✓
사용자 지정 인공신경망	✓	✓	✓	✓	

- 문서 인텔리전스 SDK 베타 미리 보기 릴리스에는 다음 업데이트가 포함되어 있습니다.
 - 사용자 지정 문서 모델 및 모드:
 - 사용자 지정 템플릿 (이전의 사용자 지정 양식).
 - 사용자 지정 신경망
 - 사용자 지정 모델 - 빌드 모드
 - W-2 미리 빌드된 모델(preebuilt-tax.us.w2)
 - 미리 빌드된 모델 읽기(preebuilt-read)
 - 청구서 미리 빌드된 모델(스페인어)(preebuilt-invoice)

2021년 11월

- 문서 인텔리전스 v3.0 미리 보기 SDK 출시 업데이트(beta.2)에는 버그 수정 및 사소한 기능 업데이트가 포함되어 있습니다.

2021년 10월

- 문서 인텔리전스 v3.0 미리 보기 출시 버전 4.0.0-beta.1(2021-10-07)에는 몇 가지 새로운 기능이 도입되었습니다.
 - 일반 문서** 모델은 미리 학습된 모델을 사용하여 양식 및 문서에서 텍스트, 테이블, 구조 및 키-값 쌍을 추출하는 새로운 API입니다.
 - 호텔 영수증** 모델이 미리 빌드된 영수증 처리에 추가되었습니다.
 - ID 문서에 대한 확장된 필드** ID 모델은 미국 운전 면허증에서 보증, 제한 및 차량 분류 추출을 지원합니다.
 - 서명 필드**는 양식 필드에 서명이 있는지 감지하기 위한 사용자 지정 양식의 새로운 필드 형식입니다.
 - 언어 확장** 122개 언어(인쇄) 및 7개 언어(필기)를 지원합니다. 문서 인텔리전스 레이아웃 및 사용자 지정 양식은 최신 미리 보기 를 통해 [지원 언어](#)를 122개로 확장합니다. 미리 보기에는 러시아어, 불가리아어 및 기타 키릴 자모 및 기타 라틴 언어를 포함하여 새로운 49개 언어의 인쇄 텍스트에 대한 텍스트 추출이 포함됩니다. 또한 필기 텍스트 추출 기능은 이제 영어를 포함한 7개 언어를 지원하고 중국어 간체, 프랑스어, 독일어, 이탈리아어, 포르투갈어 및 스페인어의 새로운 미리 보기 를 지원합니다.
 - 향상된 테이블 및 텍스트 추출 기능** 이제 레이아웃에서 키-값 테이블이라고도 하는 단일 행 테이블 추출을 지원합니다. 향상된 텍스트 추출 기능에는 일반 성능과 함께 ID 문서에서 향상된 디지털 PDF 및 MRZ(Machine Readable Zone) 텍스트 기능이 포함됩니다.
 - Document Intelligence Studio** 서비스 사용을 간소화하기 위해 이제 Document Intelligence Studio에 액세스하여 미리 빌드된 다양한 모델을 테스트하거나 사용자 지정 모델에 레이블을 지정하고 학습할 수 있습니다.
 - v3.0 미리 보기 API에 대한 새로운 [REST API](#), [Python](#) 또는 [.NET](#) SDK를 시작합니다.

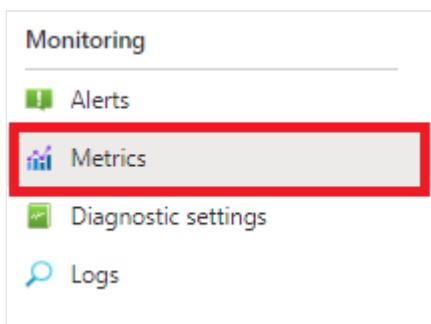
- 문서 인텔리전스 모델 데이터 추출

 테이블 확장

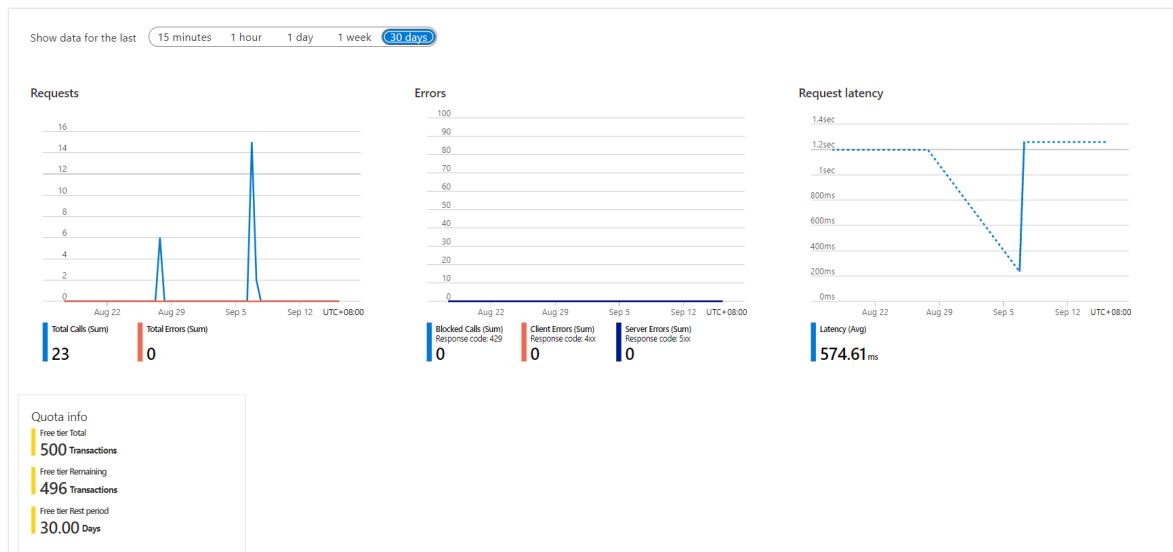
모델	텍스트 추출	키/값 쌍	선택 표시	테이블
일반 문서	✓	✓	✓	✓
Layout	✓		✓	✓
송장	✓	✓	✓	✓
Receipt	✓	✓		
ID 문서	✓	✓		
명함	✓	✓		
사용자 지정	✓	✓	✓	✓

2021년 9월

- Azure 메트릭 탐색기 고급 기능은 Azure portal의 문서 인텔리전스 리소스 개요 페이지에서 사용할 수 있습니다.
- 모니터링 메뉴:



- 차트:



- **ID 문서 모델 업데이트:** 마침표의 유무에 관계없이 접미사가 포함된 이름을 지정하면 성공적으로 처리됩니다.

▣ 테이블 확장

입력 텍스트	업데이트 결과
William Isaac Kirby Jr.	FirstName: William Isaac LastName: Kirby Jr.
Henry Caleb Ross Sr	FirstName: Henry Caleb LastName: Ross Sr

2021년 7월

- 시스템 할당 관리 ID 지원: 이제 시스템 할당 관리 ID를 사용하도록 설정하여 Virtual Network, 방화벽 또는 BYOS(Bring-your-own-Storage)로 보호되는 계정을 포함하여 프라이빗 스토리지 계정에 대한 문서 인텔리전스 제한된 액세스 권한을 부여할 수 있습니다. 자세한 내용은 [문서 인텔리전스 리소스에 대한 관리 ID 생성 및 사용](#)을 참조하세요.

2021년 6월

C#	참조 설명서 NuGet 패키지 버전 3.1.1 ↴
----	-------------------------------

- 제한 미리 보기로 출시된 문서 인텔리전스 컨테이너 v2.1은 이제 6가지 기능 컨테이너(레이아웃, 명함, ID 문서, 영수증, 청구서 및 사용자 지정)에서 지원됩니다. 이를 사용하려면 온라인 요청을 [제출](#)하고 승인을 받아야 합니다.
 - **문서 인텔리전스용 Docker 컨테이너 설치 및 실행** 및 **문서 인텔리전스 컨테이너 구성**을 참조하세요.
- 미리 보기로 출시된 문서 인텔리전스 커넥터: [문서 인텔리전스 커넥터는 Azure Logic Apps, Microsoft Power Automate](#) 및 [Microsoft Power Apps](#)와 통합됩니다. 커넥터는 사용자 지정 및 미리 빌드된 양식, 청구서, 영수증, 비즈니스 카드 및 ID 문서에서 문서 데이터 및 구조를 추출하고 분석하는 워크플로 작업 및 트리거를 지원합니다.
- 문서 인텔리전스 SDK v3.1.0은 C#, Java 및 Python용 v3.1.1로 패치되었습니다. 이 패치는 `Text` 가 있지만 `BoundingBox` 또는 `Page` 정보가 없는 `FormField`와 같이 검색된 하위 품목 필드가 없는 청구서를 처리합니다.

2021년 5월

C#

- 버전 3.1.0(2021-05-26)

[Changelog/릴리스 기록](#) | [참조 설명서](#) | [NuGet 패키지 버전 3.0.1](#) |

- 문서 인텔리전스 2.1이 정식 출시되었습니다. GA 릴리스는 이전 2.1 미리 보기 패키지 버전에서 도입된 변경 내용의 안정성을 표시합니다. 이 릴리스를 사용하면 문서 형식에서 정보 및 데이터를 검색하고 추출할 수 있습니다.
 - ✓ 문서
 - ✓ Receipts
 - ✓ 명함
 - ✓ 송장
 - ✓ ID 문서
 - ✓ 사용자 지정 양식
- 시작하려면 [문서 인텔리전스 샘플 도구](#)를 사용해 보고 [빠른 시작](#)을 따르세요.
- 업데이트된 레이아웃 API 테이블 기능은 여러 행에 걸쳐 있을 수 있는 열 머리글을 사용하여 머리글 인식을 추가합니다. 각 테이블 셀에는 머리글의 일부인지 여부를 나타내는 특성이 있습니다. 이 업데이트는 테이블 머리글을 구성하는 행을 식별하는 데 사용할 수 있습니다.

2021년 4월

C#

- *NuGet 패키지 버전 3.1.0-beta.4
- [변경 로그/릴리스 기록](#)
- ID 문서의 데이터를 분석하는 새로운 방법:

[StartRecognizeIdDocumentsFromUriAsync](#)

[StartRecognizeIdDocumentsAsync](#)

필드 값 목록은 문서 인텔리전스 문서의 참조[추출된 필드](#)를 참조하세요.

- [StartRecognizeContent](#) 메서드에 제공할 수 있는 문서 언어 세트가 확장되었습니다.
- 다음 클래스에서 지원하는 새 `Pages` 속성:

[RecognizeBusinessCardsOptions](#)

[RecognizeCustomFormsOptions](#)

[RecognizeInvoicesOptions](#)

[RecognizeReceiptsOptions](#)

이 `Pages` 속성을 사용하면 다중 페이지 PDF 및 TIFF 문서에 대한 개별 또는 페이지 범위를 선택할 수 있습니다. 개별 페이지의 경우 페이지 번호(예: 3)를 입력합니다. 페이지 범위(예: 2페이지 및 5~7페이지)의 경우 페이지 번호와 범위를 쉼표로 구분하여 입력합니다(2, 5-7).

- 다음 클래스에 대해 지원되는 새 속성 `ReadingOrder`:

[RecognizeContentOptions](#)

`ReadingOrder` 속성은 `basic` 또는 `natural`의 읽기 순서 알고리즘 중에서 텍스트 요소의 추출 순서에 적용할 알고리즘을 지정할 수 있는 선택적 매개 변수입니다. 지정하지 않으면 기본값은 `basic`입니다.

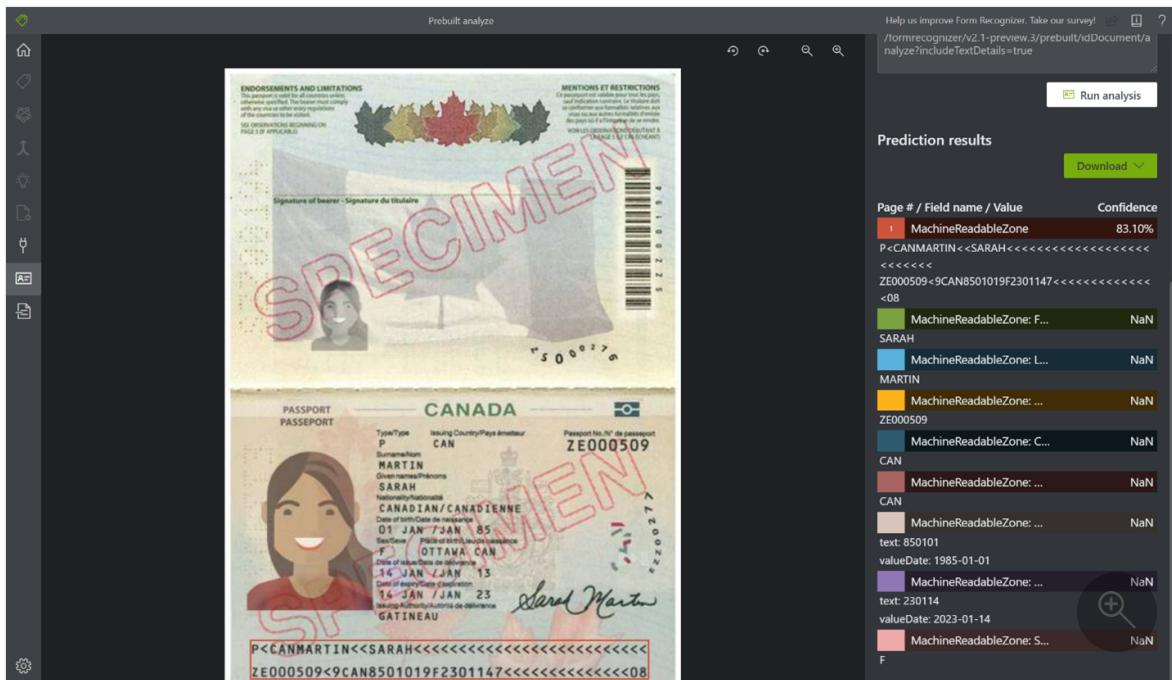
- API 버전 `2.1-preview.3` 에 대한 SDK 미리 보기 업데이트는 기능 업데이트 및 향상된 기능을 도입합니다.

2021년 3월

문서 인텔리전스 v2.1 공개 미리 보기 v2.1-preview.3이 출시되었으며 다음 기능이 포함되어 있습니다.

- **새 미리 빌드된 ID 모델** 미리 빌드된 새 ID 모델을 사용하면 고객이 ID를 가져와서 구조화된 데이터를 반환하여 처리를 자동화할 수 있습니다. 강력한 OCR(광학 인식) 기능과 ID 이해 모델을 결합하여 여권 및 미국 운전 면허증에서 주요 정보를 추출합니다.

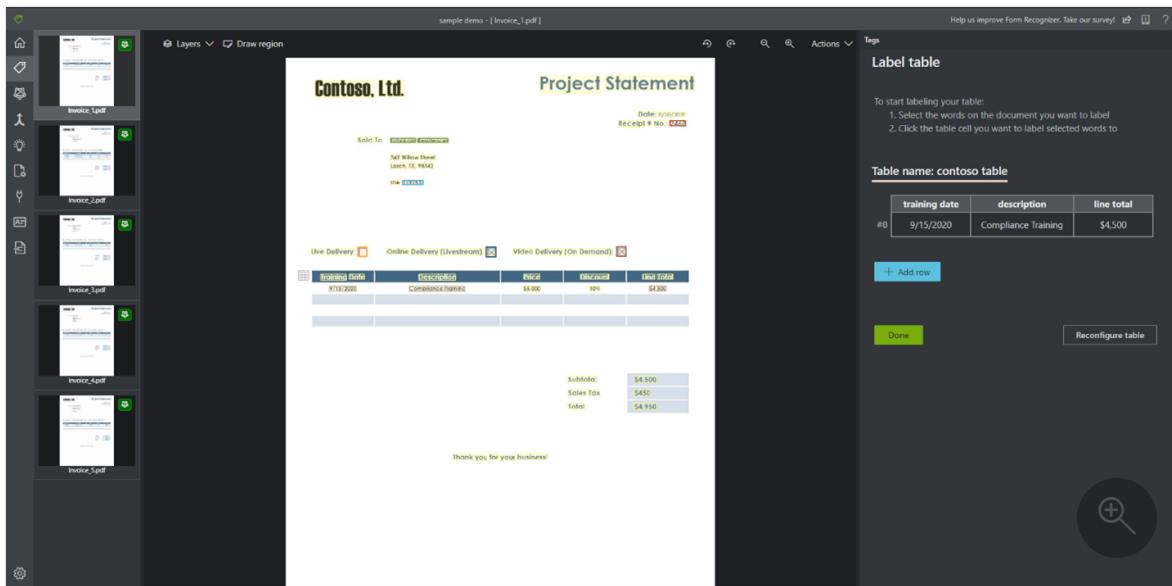
미리 빌드된 ID 모델에 대해 자세히 알아보기



- **청구서 모델에 대한 품목 추출** - 미리 빌드된 청구서 모델은 이제 품목 추출을 지원합니다. 이제 설명, 금액, 수량, 제품 ID, 날짜 등 전체 항목 및 해당 부분을 추출합니다. 간단한 API/SDK 호출로 청구서에서 유용한 데이터(텍스트, 표, 키-값 쌍, 품목)를 추출할 수 있습니다.

청구서 모델에 대한 자세한 정보

- **감독된 테이블 레이블 지정 및 학습, 빈 값 레이블 지정** - 문서 인텔리전스의 [최첨단 딥러닝 자동 테이블 추출 기능](#) 외에도 이제 고객이 테이블에서 레이블을 지정하고 학습할 수 있습니다. 이 새 릴리스에는 품목/테이블(동적 및 고정)에 레이블을 지정하고 학습하고 사용자 지정 모델을 학습하여 키-값 쌍 및 품목을 추출하는 기능이 포함되어 있습니다. 모델을 학습하면 모델은 documentResults 섹션에서 JSON 출력의 일부로 품목을 추출합니다.



이제 테이블에 레이블을 지정하는 것 외에도 빈 값 및 지역에 레이블을 지정할 수 있습니다. 학습 집합의 일부 문서에 특정 필드에 대한 값이 없는 경우 모델에서 분석된 문서에서 값을 제대로 추출할 수 있도록 레이블을 지정할 수 있습니다.

- **66개의 새로운 언어 지원** - 문서 인텔리전스용 레이아웃 API 및 사용자 지정 모델은 이제 73개 언어를 지원합니다.

[문서 인텔리전스의 언어 지원에 대해 자세히 알아보기](#)

- **자연 읽기 순서, 필기 분류 및 페이지 선택** - 이 업데이트를 사용하면 기본 왼쪽에서 오른쪽 및 위쪽에서 아래쪽 순서 대신 텍스트 줄 출력을 자연 읽기 순서로 가져올 수 있습니다. 새 readingOrder 쿼리 매개 변수를 사용하고 보다 인간 친화적인 읽기 순서 출력을 위해 "자연" 값으로 설정합니다. 또한 라틴어 언어의 경우 문서 인텔리전스는 텍스트 줄을 손으로 쓴 스타일인지 여부를 분류하고 신뢰도 점수를 제공합니다.
- **미리 빌드된 영수증 모델 품질 향상** 이 업데이트에는 미리 빌드된 영수증 모델, 특히 품목 추출에 대한 많은 품질 개선 사항이 포함되어 있습니다.

2020년 11월

- **문서 인텔리전스 v2.1-preview.2가 출시되었으며 다음 기능이 포함되어 있습니다.**
 - **미리 빌드된 새 청구서 모델** - 미리 빌드된 새 청구서 모델을 사용하면 고객이 다양한 형식으로 청구서를 받고 구조화된 데이터를 반환하여 청구서 처리를 자동화할 수 있습니다. 강력한 OCR(광학 인식) 기능을 청구서 해석 딥 러닝 모델과 결합하여 영어로 된 청구서에서 주요 정보를 추출합니다. 고객, 공급업체, 청구서 ID, 송장 기한, 총액, 기한, 세금 금액, 배송지 및 청구와 같은 주요 텍스트, 테이블 및 정보를 추출합니다.

미리 빌드된 청구서 모델에 대해 자세히 알아보기

```

{
    "InvoiceId": {
        "type": "string",
        "valueString": "INV-100",
        "text": "INV-100",
        "boundingBox": [
            7.4926,
            1.4203,
            7.9938,
            1.4203,
            7.9938,
            1.5198,
            7.4926,
            1.5198
        ],
        "page": 1,
        "confidence": 0.999,
        "elements": [
            "#/readResults/0/lines/3/words/1"
        ]
    },
    "VendorAddress": {
        "type": "string",
        "valueString": "123 456th St New York, NY, 10001",
        "text": "123 456th St New York, NY, 10001",
        "boundingBox": [
            0.594,
            1.6077,
            1.9918
        ],
        "page": 1,
        "confidence": 0.999,
        "elements": [
            "#/readResults/0/lines/3/words/1"
        ]
    }
}

```

- 향상된 테이블 추출 - 문서 인텔리전스는 이제 강력한 광학 문자 인식(OCR) 기능과 딥 러닝 테이블 추출 모델을 결합한 향상된 테이블 추출을 제공합니다. 문서 인텔리전스는 병합된 열, 행, 테두리 없음 등이 포함된 복잡한 테이블을 포함하여 테이블에서 데이터를 추출할 수 있습니다.

```

{
    "Customer": {
        "type": "string",
        "valueString": "Adventure Works Cycles",
        "text": "Adventure Works Cycles",
        "boundingBox": [
            394.5,
            395.5,
            400.5,
            401.5
        ],
        "page": 1,
        "confidence": 0.999,
        "elements": [
            "#/readResults/0/lines/3/words/1"
        ]
    },
    "Customer": {
        "type": "string",
        "valueString": "123 Remit St New York, NY, 10001",
        "text": "123 Remit St New York, NY, 10001",
        "boundingBox": [
            394.5,
            395.5,
            400.5,
            401.5
        ],
        "page": 1,
        "confidence": 0.999,
        "elements": [
            "#/readResults/0/lines/3/words/1"
        ]
    }
}

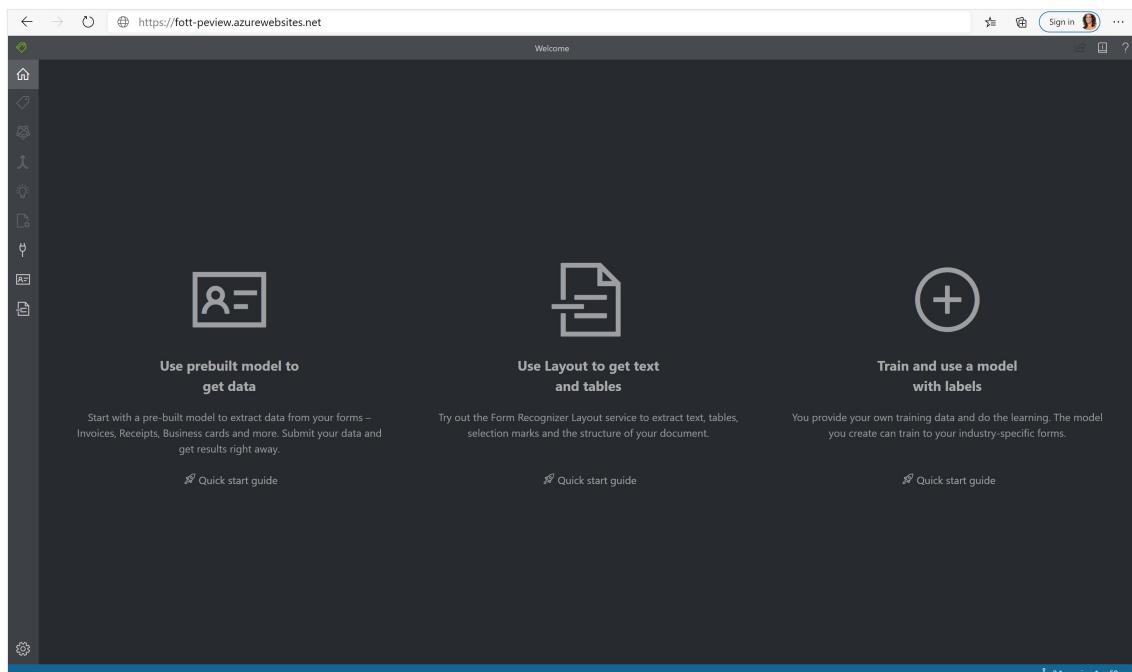
```

레이아웃 추출에 대해 자세히 알아보기

- 클라이언트 라이브러리 업데이트 - .NET, Python, Java 및 JavaScript용 [클라이언트 라이브러리](#)의 최신 버전은 문서 인텔리전스 2.1 API를 지원합니다.
- 지원되는 새 언어: 일본어 - 이제 다음과 같은 새 언어가 지원됩니다. `AnalyzeLayout` 및 `AnalyzeCustomForm`: 일본어(ja). 언어 지원
- 텍스트 줄 스타일 표시(필기/기타)(라틴어만 해당) - 문서 인텔리전스는 이제 각 텍스트 줄이 필기 스타일인지 여부를 분류하는 `appearance` 개체를 신뢰도 점수

와 함께 출력합니다. 이 기능은 라틴어로만 지원됩니다.

- **품질 개선** - 한 자릿수 추출 향상을 포함한 추출 개선 사항.
- **문서 인텔리전스 샘플 및 레이블 지정 도구의 새로운 시험 기능** - 문서 인텔리전스 샘플 레이블 지정 도구를 사용하여 미리 작성된 청구서, 영수증, 명함 모델과 레이아웃 API를 시험해 볼 수 있는 기능입니다. 코드를 작성하지 않고 데이터를 추출하는 방법을 알아보세요.
- **문서 인텔리전스 샘플 레이블 지정 도구를 사용해 보세요.** ↗



- **피드백 반복** - 샘플 레이블 지정 도구를 통해 파일을 분석할 때 이제 이를 학습 세트에 추가하고, 필요한 경우 레이블을 조정하고, 모델을 향상시키기 위해 학습시킬 수도 있습니다.
- **문서 자동 레이블 지정** - 프로젝트에서 이전에 레이블이 지정된 문서를 기준으로 추가된 문서에 자동으로 레이블을 지정합니다.

2020년 8월

- **문서 인텔리전스 v2.1-preview.1** 에는 다음 기능이 포함되어 있습니다.
 - REST API 참조를 사용할 수 있습니다. [v2.1-preview.1 reference](#) ↗
 - 영어 외에 지원되는 새 언어, 이제 다음 언어가 지원됩니다. `Layout` 및 `Train` `Custom Model`에 대해 영어(en), 중국어(간체)(zh-Hans), 네덜란드어(nl), 프랑스어(fr), 독일어(de), 이탈리아어(it), 포르투갈어(pt), 스페인어(es)가 지원됩니다.
 - 확인란/선택 표시 탐지 – 문서 인텔리전스는 체크박스, 라디오 버튼과 같은 선택 표시의 탐지 및 추출을 지원합니다. 선택 표시는 `Layout`에서 추출되며, 이제

Train Custom Model - 레이블을 사용하여 문서에서 레이블을 지정하고 학습시켜 선택 표시에 대한 키 값 쌍을 추출할 수도 있습니다.

- **모델 작성** - 여러 모델을 단일 모델 ID로 구성하고 호출할 수 있습니다. 작성된 모델 ID를 사용하여 분석할 문서를 제출하면 먼저 분류 단계를 수행하여 올바른 사용자 지정 모델로 라우팅합니다. 모델 작성은 레이블이 있는 학습에 **Train Custom Model** - 사용할 수 있습니다.
 - **모델 이름** - 더 쉽게 관리하고 추적할 수 있도록 사용자 지정 모델에 식별 이름을 추가합니다.
 - **명함용 새로운 미리 빌드된 모델** - 영어 명함에서 일반 필드를 추출합니다.
 - **미리 빌드된 영수증용 새로운 로캘** - EN-US 외에 이제 EN-AU, EN-CA, EN-GB, EN-IN에 대한 지원이 제공됩니다.
 - **품질 향상 Layout**, - **Train Custom Model** 레이블 없이 학습 및 레이블로 학습
- v2.0에는 다음 업데이트가 포함됩니다.
- .NET, Python, Java 및 JavaScript용 클라이언트 라이브러리는 일반 공급됩니다.

새 샘플 은 GitHub에서 사용할 수 있습니다.

- [Knowledge Extraction Recipes - Forms Playbook](#) 은 실제 문서 인텔리전스 고객 참여에서 모범 사례를 수집하고 이러한 프로젝트 개발에 사용되는 사용 가능한 코드 샘플, 체크리스트 및 샘플 파이프라인을 제공합니다.
- [샘플 레이블 지정 도구](#) 는 새 v2.1 기능을 지원하도록 업데이트됩니다. 도구를 시작하려면 이 [빠른 시작](#)을 참조하세요.
- [Intelligent Kiosk](#) 문서 인텔리전스 샘플은 **Analyze Receipt** 및 **Train Custom Model** - 레이블 없이 학습을 통합하는 방법을 보여줍니다.

2020년 7월

- **사용 가능한** 문서 인텔리전스 v2.0 참조 - .NET, Python, Java 및 JavaScript용 v2.0 API 참조 및 업데이트된 클라이언트 라이브러리를 봅니다.
 - **테이블 개선 사항 및 추출 향상** 기능 - 정확도 향상 및 테이블 추출 향상, 특히 레이블이 없는 사용자 지정 학습에서 테이블 헤더 및 구조를 학습하는 기능이 포함됩니다.
 - **통화 지원** - 글로벌 통화 기호 검색 및 추출
 - **Azure Gov** - 이제 Azure Gov에서도 문서 인텔리전스를 사용할 수 있습니다.
 - **향상된 보안 기능:**
 - **자체 키 가져오기** - 문서 인텔리전스는 데이터를 클라우드에 보관할 때 자동으로 암호화하여 데이터를 보호하고 조직의 보안 및 규정 준수 약속을 이행하

도록 돕습니다. 기본적으로 구독은 Microsoft에서 관리하는 암호화 키를 사용합니다. 이제 사용자 고유의 암호화 키를 사용하여 구독을 관리할 수도 있습니다. BYOK(Bring Your Own Key)라고도 하는 고객 관리형 키는 액세스 제어를 만들고, 회전하고, 사용하지 않도록 설정하고, 해지할 수 있는 더 큰 유연성을 제공합니다. 데이터를 보호하는 데 사용되는 암호화 키를 감사할 수도 있습니다.

- **프라이빗 엔드포인트** – 가상 네트워크에서 Private Link를 통해 데이터에 안전하게 액세스할 수 있습니다.
-

2020년 6월

- **클라이언트 라이브러리**에 추가된 CopyModel API - 이제 클라이언트 라이브러리를 사용하여 한 구독에서 다른 구독으로 모델을 복사할 수 있습니다. 이 기능에 대한 일반적인 정보는 백업 및 복구 모델을 참조하세요.
- **Azure Active Directory 통합** - 이제 Azure AD 자격 증명을 사용하여 클라이언트 라이브러리에서 Document Intelligence 클라이언트 개체를 인증할 수 있습니다.
- **SDK 관련 변경** - 이 변경에는 사소한 기능 추가와 호환성이 손상되는 변경 내용이 모두 포함됩니다. 자세한 내용은 SDK 변경 로그를 참조하세요.
 - [C# SDK 미리 보기 3 변경 로그 ↗](#)
 - [Python SDK 미리 보기 3 변경 로그 ↗](#)
 - [Java SDK 미리 보기 3 변경 로그 ↗](#)
 - [JavaScript SDK 미리 보기 3 변경 로그 ↗](#)

2020년 4월

- **문서 인텔리전스 API v2.0 공개 미리 보기**에 대한 SDK 지원 - 이번 달에는 문서 인텔리전스 v2.0 릴리스용 미리 보기 SDK를 포함하도록 서비스 지원을 확장했습니다. 다음 링크를 사용하여 원하는 언어를 시작합니다.
 - [.NET SDK](#)
 - [Java SDK](#)
 - [Python SDK](#)
 - [JavaScript SDK](#)

새 SDK는 문서 인텔리전스용 v2.0 REST API의 모든 기능을 지원합니다. SDK 피드백 양식을 통해 클라이언트 라이브러리에 ↗ 대한 피드백을 공유할 수 있습니다.

- **사용자 지정 모델** 복사 이제 새 사용자 지정 모델 복사 기능을 사용하여 지역과 구독 간에 모델을 복사할 수 있습니다. 사용자 지정 모델 복사 API를 호출하기 전에 먼저 대상 리소스로 복사할 권한 부여를 가져와야 합니다. 이 권한 부여는 대상 리소스 엔드포인트에 대해 권한 부여 복사 작업을 호출하여 보호됩니다.

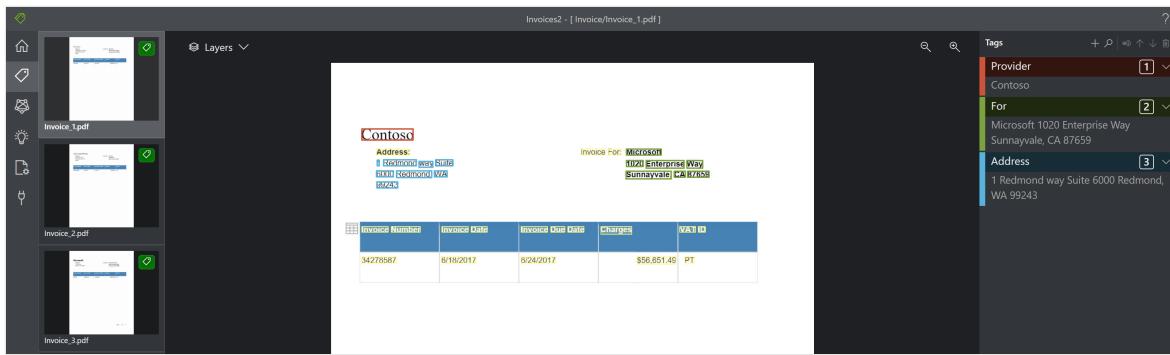
- 복사 권한 부여 ↗ REST API 생성
 - 사용자 지정 모델 ↗ REST API 복사
 - 향상된 보안 기능
 - 이제 FormRecognizer에서 고객 관리형 키를 사용할 수 있습니다. 자세한 내용은 [문서 인텔리전스를 위한 저장 데이터 암호화](#)를 참조하세요.
 - Azure Active Directory로 Azure 리소스에 액세스하기 위해 관리 ID를 사용합니다. 자세한 내용은 관리 ID에 대한 액세스 권한 부여를 참조 [하세요](#).
-

2020년 3월

- **레이블 지정을 위한 값 유형** 이제 문서 인텔리전스 샘플 레이블 지정 도구를 사용하여 레이블을 지정하는 값 유형을 지정할 수 있습니다. 현재 지원되는 값 형식 및 변형은 다음과 같습니다.
 - `string`
 - `default`, `no-whitespaces`, `alphanumeric`
 - `number`
 - 기본 `currency`
 - `date`
 - `default`, `dmy`, `mdy`, `ymd`
 - `time`
 - `integer`

이 기능을 사용하는 방법에 대한 자세한 내용은 [샘플 레이블 지정 도구](#) 가이드를 참조하세요.

- **테이블 시각화** 이제 샘플 레이블 지정 도구에는 문서에서 인식된 테이블이 표시됩니다. 이 기능을 사용하면 레이블을 지정하고 분석하기 전에 문서에서 인식되고 추출된 테이블을 볼 수 있습니다. 이 기능은 레이어 옵션을 사용하여 켜거나 끌 수 있습니다.
- 다음 이미지는 표를 인식하고 추출하는 방법의 예입니다.



- 추출된 테이블은 아래의 JSON 출력 "pageResults"에서 사용할 수 있습니다.

ⓘ 중요

테이블 레이블 지정은 지원되지 않습니다. 테이블이 자동으로 인식되고 추출되지 않는 경우 키/값 쌍으로만 레이블을 지정할 수 있습니다. 테이블에 키/값 쌍으로 레이블을 지정할 때 각 셀에 고유한 값으로 레이블을 지정합니다.

- 추출 기능 향상
- 이 릴리스에는 추출 향상 및 정확도 향상, 특히 동일한 텍스트 줄에서 여러 키/값 쌍의 레이블을 지정하고 추출하는 기능이 포함됩니다.
- 이제 샘플 레이블 지정 도구는 오픈 소스임
- 이제 문서 인텔리전스 샘플 레이블 지정 도구를 오픈 소스 프로젝트로 사용할 수 있습니다. 솔루션 내에 통합하고 요구 사항에 맞게 고객별 변경을 수행할 수 있습니다.
- 문서 인텔리전스 샘플 레이블 지정 도구에 대한 자세한 내용은 [GitHub](#)에서 제공되는 문서를 검토하세요.
- TLS 1.2 적용**
- TLS** 이제 이 서비스에 대한 모든 HTTP 요청에 1.2가 적용됩니다. 자세한 내용은 [Azure AI 서비스 보안](#)을 참조하세요.

2020년 1월

이번 릴리스에는 문서 인텔리전스 2.0이 도입되었습니다. 다음 섹션에서는 새로운 기능, 향상된 기능 및 변경 내용에 대한 자세한 정보를 확인할 수 있습니다.

- 새로운 기능
 - 사용자 지정 모델

- 레이블을 사용하여 학습 이제 수동으로 레이블이 지정된 데이터로 사용자 지정 모델을 학습시킬 수 있습니다. 이 메서드는 성능이 우수한 모델을 생성하고 키가 없는 값을 포함하는 복잡한 양식 또는 양식에서 작동하는 모델을 생성할 수 있습니다.
 - 비동기 API 비동기 API 호출을 사용하여 큰 데이터 집합 및 파일을 학습시키고 분석할 수 있습니다.
 - TIFF 파일 지원 이제 TIFF 문서에서 데이터를 학습시키고 추출할 수 있습니다.
 - 추출 정확도 개선
-
- 미리 빌드된 영수증 모델
 - 팁 금액 이제 팁 금액 및 기타 필기 값을 추출할 수 있습니다.
 - 품목 추출 영수증에서 품목 값을 추출할 수 있습니다.
 - 신뢰도 값 추출된 각 값에 대한 모델의 신뢰도를 볼 수 있습니다.
 - 추출 정확도 개선
-
- 레이아웃 추출 이제 레이아웃 API를 사용하여 양식에서 텍스트 데이터 및 테이블 데이터를 추출할 수 있습니다.

- 사용자 지정 모델 API 변경 내용

사용자 지정 모델을 학습하고 사용하기 위한 모든 API의 이름이 바뀌고 일부 동기 메서드가 비동기화됩니다. 다음은 주요 변경 내용입니다.

- 모델을 학습시키는 프로세스는 이제 비동기식입니다. /custom/models API 호출을 통해 학습을 시작합니다. 이 호출은 사용자 지정/모델/{modelID}에 전달하여 학습 결과를 반환할 수 있는 작업 ID를 반환합니다.
- 키/값 추출은 이제 /custom/models/{modelID}/analyze API 호출에 의해 시작됩니다. 이 호출은 사용자 지정/모델/{modelID}/analyzeResults/{resultID}에 전달하여 추출 결과를 반환할 수 있는 작업 ID를 반환합니다.
- 이제 학습 작업을 위한 작업 ID가 Operation-Location 헤더가 아니라 HTTP 응답의 Location 헤더에 있습니다.

- 영수증 API 변경 내용

- 판매 영수증을 읽기 위한 API의 이름이 바뀝니다.
- 이제 영수증 데이터 추출이 /prebuilt/receipt/analyze API 호출에 의해 시작됩니다. 이 호출은 /prebuilt/receipt/analyzeResults/{resultID}에 전달하여 추출 결과를 반환할 수 있는 작업 ID를 반환합니다.

- 출력 형식 변경

- 모든 API 호출에 대한 JSON 응답에는 새 형식이 있습니다. 일부 키와 값은 추가, 제거 또는 이름이 바뀝니다. 현재의 JSON 형식 예에 대해서는 빠른 시작을 참조하세요.

다음 단계

- 문서 [인텔리전스 스튜디오](#)를 사용하여 고유의 양식과 문서를 처리해 보세요.
- 문서 [인텔리전스 빠른 시작](#)을 완료하고 원하는 개발 언어로 문서 처리 앱 만들기를 시작해 보세요.

Azure AI 문서 인텔리전스 FAQ

FAQ

이 콘텐츠의 적용 대상: ✓ v4.0(미리 보기) ✓ v3.1(GA) ✓ v3.0(GA) ✓ v2.1(GA)

일반 개념

Azure AI 문서 인텔리전스란 무엇이며 Azure AI Form Recognizer는 어떻게 되었나요?

Azure AI 문서 인텔리전스는 기계 학습 모델을 사용하여 문서에서 키-값 쌍, 텍스트 및 테이블을 추출하는 클라우드 기반 서비스입니다. 반환된 결과는 정형 JSON 출력입니다. 문서 인텔리전스 사용 사례에는 자동화된 데이터 처리, 향상된 데이터 기반 전략, 보강된 문서 검색 기능이 포함됩니다.

문서 인텔리전스는 Azure AI 서비스의 일부입니다. Azure AI 서비스는 이전에 Azure Cognitive Services 및 Azure Applied AI Services로 알려진 모든 서비스를 포함합니다.

문서 인텔리전스의 이전 이름은 Azure AI Form Recognizer였습니다. Form Recognizer는 2023년 7월에 공식적으로 문서 인텔리전스가 되었습니다.

가격 책정에는 변화가 없습니다. Cognitive Services 및 Applied AI Services라는 이름은 Azure 청구, 비용 분석, 가격표, 가격 API에서 계속 사용됩니다.

API 또는 클라이언트 라이브러리(SDK)에 대한 호환성이 손상되는 변경은 없습니다. REST API 및 SDK 버전 2024-02-29-preview, 2023-10-31-preview 이상은 `document intelligence`로 이름이 바뀝니다.

일부 플랫폼은 여전히 이름 변경 업데이트를 기다리고 있습니다. Microsoft 설명서에서 Form Recognizer 및 문서 인텔리전스에 대한 모든 언급은 동일한 Azure 서비스를 나타냅니다.

문서 생성 AI와 문서 인텔리전스는 어떤 관련이 있나요?

문서 생성 AI 솔루션을 사용하여 문서와 채팅하고, 해당 문서에서 매력적인 콘텐츠를 생성하고, 데이터에서 Azure OpenAI Service 모델에 액세스할 수 있습니다. Azure AI 문서 인텔리전스와 Azure OpenAI를 결합하면 자연어를 사용하여 문서와 원활하게 상호 작용하고, 쉽게 답변을 찾고, 귀중한 인사이트를 얻고, 기존 문서에서 새롭고 매력적인 콘텐츠

를 생성하는 엔터프라이즈 애플리케이션을 빌드할 수 있습니다. [기술 커뮤니티 블로그](#)에서 자세한 내용을 확인하세요.

문서 인텔리전스는 검색 증강 생성과 어떤 관계가 있나요?

의미 체계 청크는 효율적인 스토리지 및 검색을 보장하기 위한 RAG(검색 증강 생성)의 핵심 단계입니다. 문서 인텔리전스 [레이아웃 모델](#)은 고급 콘텐츠 추출 및 문서 구조 분석 기능을 위한 포괄적인 솔루션을 제공합니다.

레이아웃 모델을 사용하면 텍스트 및 구조 요소를 쉽게 추출하여 텍스트의 큰 본문을 임의의 분할이 아닌 의미 체계 콘텐츠에 따라 더 작고 의미 있는 청크로 나눌 수 있습니다. 그런 다음, 제공된 구성 요소를 기반으로 의미 체계 청크 전략을 정의할 수 있도록 추출된 정보를 Markdown 형식으로 편리하게 출력할 수 있습니다. [문서 인텔리전스 RAG의 개요](#)에서 자세한 내용을 확인하세요.

특별한 고려가 필요한 문서 인텔리전스 사용 사례는 무엇인가요?

금융 데이터, 보호된 건강 데이터, 개인 데이터 또는 매우 중요한 데이터를 포함하는 문서 처리 프로젝트를 신중하게 고려하세요.

모든 [국가/지역 및 산업별 요구 사항](#)을 준수해야 합니다.

문서 인텔리전스는 어떤 언어를 지원하나요?

문서 인텔리전스의 딥 러닝 기반 유니버설 모델은 혼합 언어가 포함된 텍스트 줄을 포함하여 이미지와 문서에서 다국어 텍스트를 추출할 수 있는 다양한 언어를 지원합니다.

언어 지원은 문서 인텔리전스 서비스 기능에 따라 다릅니다. 문서 인텔리전스에서 지원하는 필기 및 인쇄 텍스트의 전체 목록은 [언어 지원](#)을 참조하세요.

내 Azure 지역에서 문서 인텔리전스를 사용할 수 있나요?

문서 인텔리전스는 일반적으로 [60개 이상의 Azure 글로벌 인프라 지역](#)에서 사용할 수 있습니다.

사용자와 사용자의 고객에게 가장 적합한 [지역을 선택](#)하세요.

문서 인텔리전스는 다른 Microsoft 서비스와 통합되나요?

예, 문서 인텔리전스는 다음 서비스와 통합됩니다.

- [AI Builder 워크플로](#)
- [Azure AI 검색](#)
- [Azure Functions](#)
- [Azure Logic Apps](#)

문서 인텔리전스는 광학 인식과 어떤 관련이 있나요?

문서 인텔리전스는 Azure AI 서비스의 OCR(광학 인식), 텍스트 분석 및 사용자 지정 텍스트 분류를 통합하는 클라우드 기반 서비스입니다.

문서 인텔리전스는 OCR을 사용하여 AI에서 지원하는 서체 및 필기 텍스트 문서에서 정보를 검색하고 추출하여 텍스트 추출에 더 많은 구조와 정보를 제공합니다.

사용자 지정 모델을 사용할 수 있는 기간은 얼마인가요?

모델에는 모델을 학습시키는 데 사용하는 API 버전과 동일한 수명 주기를 갖습니다. API의 GA(일반 공급) 버전으로 학습한 사용자 지정 모델은 API 버전과 동일한 수명 주기를 갖습니다. API 버전이 더 이상 사용되지 않으면 모델은 더 이상 유추에 사용할 수 없습니다. API의 미리 보기 버전으로 학습한 모델도 미리 보기 API와 동일한 수명 주기를 갖습니다.

업데이트된 미리 보기 API 버전 또는 최신 GA API 버전이 출시된 후 3개월 이내에 미리 보기 API가 더 이상 사용되지 않을 것으로 예상됩니다.

정확도 점수란 무엇이며 어떻게 계산하나요?

`build`(v3.0 이상 버전) 또는 `train`(v2.1) 사용자 지정 모델 작업의 출력에는 예상 정확도 점수가 포함됩니다. 이 점수는 시각적으로 유사한 문서에서 레이블이 지정된 값을 정확하게 예측하는 모델의 기능을 나타냅니다.

정확도는 0%(낮음)에서 100%(높음)까지 백분율 값 범위 내에서 측정됩니다.

자세한 내용은 [정확도 및 신뢰도 점수](#)를 참조하세요.

정확도 점수를 향상하려면 어떻게 해야 하나요?

문서의 시각적 구조의 차이는 모델의 정확도에 영향을 줄 수 있습니다. 다음 팁을 참조하세요.

- 학습 데이터 세트에 문서의 모든 변형을 포함합니다. 변형에는 여러 형식이 포함됩니다(예: 디지털 PDF와 스캔된 PDF).
- 시각적으로 고유한 문서 형식을 구분하고 다른 모델을 학습시킵니다.
- 관련 없는 레이블이 없는지 확인합니다.
- 서명 및 지역 레이블을 지정할 때는 주변 텍스트를 포함하지 마세요.

자세한 내용은 [정확도 및 신뢰도 점수](#)를 참조하세요.

신뢰도 점수란 무엇이며 어떻게 계산되나요?

신뢰도 점수는 추출된 결과가 올바르게 검색되는 통계적 확실성의 정도를 측정하여 확률을 나타냅니다.

신뢰도 값 범위는 0%(낮음)에서 100%(높음)까지의 백분율입니다. 80% 이상의 점수를 목표로 하는 것이 가장 좋습니다. 금융 또는 의료 기록과 같은 매우 민감한 사례에는 100%에 가까운 점수를 권장합니다. 사람의 검토를 요구할 수도 있습니다.

자세한 내용은 [정확도 및 신뢰도 점수](#)를 참조하세요.

신뢰도 점수를 높이려면 어떻게 해야 하나요?

분석 작업을 마친 후 JSON 출력을 검토합니다. `pageResults` 노드 아래에서 각 키/값 결과에 대한 `confidence` 값을 검사합니다. `readResults` 노드에서 텍스트 읽기 작업에 해당하는 신뢰도 점수도 확인해야 합니다. 읽기 결과의 신뢰도는 키/값 추출 결과의 신뢰도에 영향을 주지 않으므로 둘 다 확인해야 합니다. 다음 팁을 참조하세요.

- `readResults` 개체의 신뢰도 점수가 낮으면 입력 문서의 품질을 높입니다.
- `pageResults` 개체의 신뢰도 점수가 낮으면 분석 중인 문서가 동일한 형식인지 확인합니다.
- 워크플로에 사람의 검토를 통합하는 것이 좋습니다.
- 각 필드에 서로 다른 값이 있는 양식을 사용합니다.
- 사용자 지정 모델의 경우 더 큰 학습 문서 세트를 사용합니다. 더 많은 문서에 태그를 지정하면 모델이 필드를 보다 정확하게 인식할 수 있습니다.

자세한 내용은 [정확도 및 신뢰도 점수](#)를 참조하세요.

경계 상자란?

경계 상자(v3.0 이상 버전의 `polygon`)는 문서 또는 양식의 텍스트 요소를 둘러싸는 추상 사각형입니다. 개체 감지를 위한 참조 지점으로 사용됩니다.

경계 상자는 4개의 숫자 쌍 배열로 표시되는 x 및 y 좌표 평면을 사용하여 위치를 지정합니다. 각 쌍은 상자의 모서리를 왼쪽 위, 오른쪽 위, 오른쪽 아래, 왼쪽 아래 순서로 나타냅니다.

이미지의 경우 좌표는 픽셀 단위입니다. PDF의 경우 좌표는 인치 단위입니다.

문서 인텔리전스를 통해 문서를 분류할 수 있나요?

문서 인텔리전스는 단일 파일 또는 다중 파일 문서를 분석하여 입력 파일에 학습된 문서 형식이 포함되어 있는지 여부를 식별할 수 있는 사용자 지정 분류 모델을 제공합니다. 이 서비스는 다음 시나리오를 지원합니다.

- 대출 신청 양식과 같은 하나의 문서 형식을 포함하는 단일 파일.
- 여러 문서를 포함하는 단일 파일. 예를 들어 대출 신청 양식, 급여 명세서 및 은행 명세서를 포함하는 대출 신청 패키지가 있습니다.
- 동일한 문서의 여러 인스턴스를 포함하는 단일 파일. 예를 들어 스캔한 청구서의 컬렉션이 있습니다.

자세한 내용은 [사용자 지정 분류 모델의 개요](#)를 참조하세요.

앱 개발

문서 인텔리전스의 개발 옵션은 무엇인가요?

문서 인텔리전스는 다음 플랫폼 내에서 최신 개발 옵션을 제공합니다.

- [REST API](#)
- [문서 인텔리전스 스튜디오](#)
- [C#/.NET](#)
- [Java](#)
- [JavaScript/TypeScript](#)

- Python ↗

최신 프로그래밍 언어 SDK를 지원하는 API 버전은 어디서 찾을 수 있나요?

다음 표는 최신 SDK 버전의 링크를 제공하고 지원되는 문서 인텔리전스 SDK 버전과 API 버전 간의 관계를 보여 줍니다.

[+] 테이블 확장

지원되는 언어 Azure SDK 참조	지원되는 API 버전
• C#/.NET: 4.0.0 ↗	2023-10-31-preview
• Java: 4.0.0 ↗	v3.0
• JavaScript: 4.0.0 ↗	v2.1 ↗
• Python 3.2.0 ↗	v2.0 ↗

자세한 내용은 [v4.0용 지원되는 클라이언트](#) 및 [v3.1용 지원되는 클라이언트](#)를 참조하세요.

문서 인텔리전스 v3.0과 v2.1의 차이점은 무엇이며 최신 버전으로 마이그레이션하려면 어떻게 해야 하나요?

개선된 유용성을 위해 문서 인텔리전스 v3.0은 완전히 재설계된 클라이언트 라이브러리를 도입합니다. 최신 문서 인텔리전스 API 기능을 성공적으로 사용하려면 최신 SDK가 필요하며 새 클라이언트를 사용하도록 애플리케이션 코드를 업데이트해야 합니다.

이 표에서는 최신 버전의 문서 인텔리전스로 마이그레이션하기 위한 자세한 지침에 대한 링크를 제공합니다.

[+] 테이블 확장

언어/API	マイグ레이션 ガイド
REST API	v3
C#/.NET	4.0.0 ↗
Java	4.0.0 ↗
JavaScript	4.0.0 ↗
Python	3.2.0 ↗

문서 인텔리전스는 어떤 파일 형식을 지원하나요? 입력 문서에 대한 크기 제한이 있나요?

최상의 결과를 얻으려면 [입력 요구 사항](#)을 참조하세요.

문서에서 분석할 페이지 범위를 지정하려면 어떻게 해야 하나요?

`pages` 매개 변수(v2.1, v3.0 이상 버전의 REST API에서 지원됨)를 사용하여 여러 페이지 PDF 및 TIFF 문서의 페이지를 지정합니다. 허용되는 입력에는 다음 범위가 포함됩니다.

- 단일 페이지. 예를 들어 `1, 2`를 지정하면 1페이지와 2페이지가 처리됩니다.
- 유한한 범위. 예를 들어 `2-5`를 지정하면 2~5페이지가 처리됩니다.
- 열린 범위. 예를 들어 `5-`를 지정하면 5페이지부터 모든 페이지가 처리됩니다. 예를 들어 `-10`을 지정하면 1~10페이지가 처리됩니다.

이러한 매개 변수를 함께 혼합할 수 있으며 범위가 겹칠 수 있습니다. 예를 들어 `-5, 1, 3, 5-10`을 지정하면 1~10페이지가 처리됩니다.

서비스는 문서의 한 페이지 이상을 처리할 수 있는 경우 요청을 수락합니다. 예를 들어 5페이지 문서에서 `5-100`을 사용하는 것은 5페이지가 처리됨을 의미하는 유효한 입력입니다.

페이지 범위를 제공하지 않으면 전체 문서가 처리됩니다.

문서 인텔리전스 스튜디오와 FOTT 샘플 레이블 지정 도구를 모두 사용할 수 있습니다. 어느 것을 사용해야 합니까?

대부분의 경우 문서 인텔리전스 리소스 및 스토리지 서비스를 구성하는 시간을 줄일 수 있으므로 [문서 인텔리전스 스튜디오](#)를 권장합니다.

다음 시나리오에는 FOTT(양식 OCR 테스트 도구)를 사용하는 것이 좋습니다.

- 데이터가 단일 머신 내에 유지되어야 합니다. [FOTT 샘플 레이블 지정 도구](#)와 [문서 인텔리전스 컨테이너](#)를 사용합니다.
- 귀하의 프로젝트는 [문서 인텔리전스 v2.1](#)에 크게 의존하고 있으며 v2.1 API를 계속 사용하려고 합니다.

서비스 제한 및 가격 책정

Azure에서는 문서 인텔리전스 사용 가격을 어떻게 계산하나요?

문서 인텔리전스 청구는 모델 형식 및 분석된 페이지 수를 기준으로 매월 계산됩니다. 다음은 몇 가지 세부 정보입니다.

- 분석을 위해 문서를 제출할 때 요청에서 `pages` 매개 변수를 사용하여 페이지 범위를 지정하지 않으면 서비스에서 모든 페이지를 분석합니다. 서비스는 읽기, OCR 또는 레이아웃 모델을 통해 Microsoft Excel 및 PowerPoint 문서를 분석할 때 각 Excel 워크시트와 PowerPoint 슬라이드를 하나의 페이지로 계산합니다.
- 서비스는 PDF 및 TIFF 파일을 분석할 때 PDF 파일의 각 페이지 또는 TIFF 파일의 각 이미지를 최대 문자 제한 없이 한 페이지로 계산합니다.
- 서비스는 읽기 및 레이아웃 모델이 지원하는 Microsoft Word 및 HTML 파일을 분석할 때 각각 3,000자 블록으로 페이지 수를 계산합니다. 예를 들어 문서에 7,000자가 포함된 경우 각각 3,000자의 두 페이지와 1,000자의 한 페이지를 합하여 총 3페이지가 됩니다.
- 읽기 또는 레이아웃 모델을 사용하여 Microsoft Word, Excel, PowerPoint, HTML 파일을 분석하는 경우 포함되거나 연결된 이미지는 지원되지 않습니다. 따라서 서비스는 이를 추가된 이미지로 계산하지 않습니다.

- 문서 인텔리전스를 사용하면 사용자 지정 모델을 항상 무료로 학습할 수 있습니다. 서비스에서 모델을 사용하여 문서를 분석하는 경우에만 요금이 청구됩니다.
- 컨테이너 가격 책정은 클라우드 서비스 가격 책정과 동일합니다.
- 문서 인텔리전스는 모든 문서 인텔리전스 기능을 테스트할 수 있는 무료 계층(F0)을 제공합니다.
- 문서 인텔리전스에는 대규모 워크로드에 대한 약정 기반 가격 책정 모델이 있습니다.

[Azure AI 문서 인텔리전스 가격 책정 옵션](#)에 대해 자세히 알아봅니다.

문서 인텔리전스 사용량을 확인하고 가격을 예상 하려면 어떻게 해야 하나요?

Azure Portal의 메트릭 대시보드에서 사용 현황 메트릭을 찾을 수 있습니다. 대시보드에는 Azure AI 문서 인텔리전스에서 처리한 페이지 수가 표시됩니다. [Azure 가격 계산기](#)를 사용하여 리소스에 사용된 예상 비용을 확인할 수 있습니다. 자세한 지침은 [사용량 확인 및 예상 비용](#)을 참조하세요.

제한을 완화하는 모범 사례는 무엇인가요?

문서 인텔리전스는 자동 크기 조정을 사용하여 필요한 계산 리소스를 주문형으로 제공하는 동시에 고객 비용을 낮게 유지합니다. 자동 스케일링 중에 제한을 완화하려면 다음 방법을 사용하는 것이 좋습니다.

- 애플리케이션에서 다시 시도 논리를 구현합니다.
- `POST` 요청 수에 제한이 있는 경우 요청 사이에 지연을 추가하는 것이 좋습니다.
- 워크로드를 점진적으로 늘립니다. 급격한 변화를 피합니다.
- TPS(초당 트랜잭션 수) 제한을 늘리기 위한 [지원 요청을 만듭니다](#).

문서 인텔리전스 [서비스 할당량 및 한도](#)에 대해 자세히 알아봅니다.

문서를 분석하는 데 얼마나 걸리나요?

문서를 분석하는 데 걸리는 시간은 크기(예: 페이지 수) 및 각 페이지의 관련 콘텐츠에 따라 다릅니다.

문서 인텔리전스는 유사한 문서의 대기 시간이 비슷하지만 항상 동일하지는 않은 다중 테넌트 서비스입니다. 대기 시간은 API 서버가 들어오는 요청을 처리하고 클라이언트에

나가는 응답을 전달하는 데 걸리는 시간의 양입니다. 대기 시간 및 성능의 가변성은 이미지 및 대량의 문서를 대규모로 처리하는 마이크로 서비스 기반의 상태 비저장 비동기 서비스의 고유한 특성입니다.

하드웨어 및 용량과 스케일링 기능을 지속적으로 스케일 업하고 있지만, 아직은 런타임에 대기 시간 문제가 발생할 수 있습니다.

사용자 지정 모델

최상의 학습 데이터를 모으려면 어떻게 해야 하나요?

문서 인텔리전스 사용자 지정 모델을 사용하는 경우 자체 학습 데이터를 제공합니다. 다음은 모델을 효과적으로 학습시키는데 도움이 되는 몇 가지 팁입니다.

- 되도록이면 이미지 기반 PDF 대신 텍스트 기반을 사용합니다. 이미지 기반 PDF를 식별하는 한 가지 방법은 문서에서 특정 텍스트를 선택하는 것입니다. 텍스트의 전체 이미지만 선택할 수 있는 경우 문서는 텍스트 기반이 아닌 이미지 기반입니다.
- 각 형식(JPEG/JPG, PNG, BMP, PDF 또는 TIFF)에 대한 하위 폴더를 사용하여 학습 문서를 구성합니다.
- 사용 가능한 필드가 모두 완료된 양식을 사용합니다.
- 각 필드의 값이 서로 다른 양식을 사용합니다.
- 이미지 품질이 낮은 경우 더 큰 데이터 세트(학습 문서가 5개 이상)를 사용합니다.

[학습 데이터 세트 만들기](#)에 대해 자세히 알아보세요.

매우 정확한 사용자 지정 모델을 학습시키는 모범 사례는 무엇인가요?

모델의 정확도는 학습 자료의 질에 따라 달라집니다. 다음 팁을 참조하세요.

- 단일 모델을 사용해야 하는지 아니면 단일 모델로 구성된 여러 모델을 사용해야 하는지 확인합니다.
- 단일 모델로 여러 형식을 분석하면 모델 정확도가 떨어질 수 있습니다. 각 폴더가 고유한 템플릿인 폴더로 데이터 세트를 분할하는 계획을 세웁니다. 폴더당 하나의 모델을 학습시키고 결과 모델을 단일 엔드포인트로 구성합니다.

- 사용자 지정 양식은 일관된 시각적 템플릿을 사용합니다. 양식에 서식 및 페이지 나 누기 변형이 있는 경우 데이터 세트를 분할하여 여러 모델을 학습시키는 것이 좋습니다.
- 형식, 문서 형식 및 구조를 고려하여 균형 잡힌 데이터 세트가 있는지 확인합니다.

구성된 모델에 대해 자세히 알아보세요.

사용자 지정 모델을 다시 학습시킬 수 있나요?

문서 인텔리전스에는 명시적인 재학습 작업이 없습니다. 각 학습 작업은 새 모델을 생성합니다.

모델을 다시 학습시켜야 경우 학습 데이터 세트에 더 많은 샘플을 추가하고 새 모델을 학습시킵니다.

단일 사용자 지정 모델로 작성할 수 있는 사용자 지정 모델은 몇 개인가요?

[모델 작성](#) 작업을 통해 최대 200개의 모델을 단일 모델 ID에 할당할 수 있습니다. 구성된 모델 ID를 사용하여 [Analyze Document](#) 요청을 하면 문서 인텔리전스는 제출된 양식을 분류하고 최상의 모델을 선택한 후 결과를 반환합니다. 현재 모델 만들기는 레이블을 사용하여 학습된 사용자 지정 모델에만 사용할 수 있습니다.

구성된 모델을 사용하여 문서를 분석하는 것은 단일 모델을 사용하여 문서를 분석하는 것과 동일합니다. [Analyze Document](#) 결과는 문서를 분석하기 위해 선택한 구성 요소 모델을 나타내는 `docType` 속성을 반환합니다. 개별 사용자 정의 모델 또는 구성된 사용자 정의 모델을 사용하여 문서를 분석하는 경우 가격은 변경되지 않습니다.

구성된 모델에 대해 자세히 알아보세요.

내가 구성하려는 모델의 수가 구성된 모델의 상한을 초과하는 경우 대안은 무엇인가요?

다음 대안 중 하나를 사용할 수 있습니다.

- 사용자 지정 모델을 호출하기 전에 문서를 분류합니다. [읽기 모델](#)을 사용하고 코드, 정규식 또는 검색과 같은 소스를 사용하여 문서에서 추출된 텍스트 및 특정 구를 기반으로 분류를 만들 수 있습니다.
- 다양한 구조적, 반구조적 및 비구조적 문서에서 동일한 필드를 추출하려는 경우 딥러닝 [사용자 지정 신경망 모델](#)을 사용하는 것이 좋습니다. [사용자 지정 템플릿 모델](#)

과 사용자 지정 신경망 모델의 차이점에 대해 자세히 알아봅니다.

초기 학습을 넘어 모델을 구체화하려면 어떻게 해야 하나요?

각 학습 작업은 새 모델을 생성합니다.

1. 새 템플릿에 대한 데이터 세트를 만듭니다.
2. 새 모델의 레이블을 지정하고 학습시킵니다.
3. 새 모델이 특정 문서 형식에 대해 잘 작동하는지 확인합니다.
4. 기존 모델을 사용하여 새 모델을 단일 엔드포인트로 작성합니다. 그런 다음 문서 인텔리전스는 분석할 각 문서에 가장 적합한 모델을 결정할 수 있습니다.

구성된 모델에 대해 자세히 알아보세요.

사용자 지정 모델을 만들고 있습니다. 서명 감지 레이블은 무엇을 반환하나요?

[서명 감지](#)는 문서에 서명하는 사람의 ID가 아니라 서명의 존재를 찾습니다.

모델이 서명 감지에 대해 `unsigned`를 반환하면 모델이 정의된 필드에서 서명을 찾지 못한 것입니다.

문서에서 테이블을 추출하는 모범 사례는 무엇이며 무엇을 고려해야 하나요?

문서 인텔리전스 [레이아웃 모델](#)로 시작하여 문서와 이미지에서 텍스트, 테이블, 선택 표시, 구조 정보를 추출할 수 있습니다. 또한 다음 요소를 고려할 수 있습니다.

- 추출하려는 데이터가 테이블로 존재하고 테이블 구조가 의미 있나요?
- 데이터가 테이블 형식이 아닌 경우 데이터가 2차원 그리드에 잘 맞나요?
- 테이블이 여러 페이지에 걸쳐 있나요? 그렇다면 모든 페이지에 레이블을 지정할 필요가 없도록 PDF를 문서 인텔리전스로 보내기 전에 여러 페이지로 분할합니다. 분석 후 페이지를 단일 테이블로 후처리합니다.
- 사용자 지정 모델을 만드는 경우 [테이블로 레이블 지정](#)을 참조하세요. 각 열에 대한 동적 테이블의 행 수는 가변적입니다. 각 열에 대한 고정 테이블의 행 수는 일관적입니다.

학습된 모델을 한 환경(예: 베타)에서 다른 환경(예: 프로덕션)으로 이동하려면 어떻게 해야 하나요?

복사 API를 사용하여 한 문서 인텔리전스 계정에서 지원되는 지리적 지역에 있는 다른 계정으로 사용자 지정 모델을 복사할 수 있습니다. 자세한 지침은 [재해 복구](#)를 참조하세요.

복사 작업은 모델을 학습시킨 특정 클라우드 환경 내에서 모델을 복사하는 것으로 제한됩니다. 예를 들어 퍼블릭 클라우드에서 Azure Government 클라우드로 모델을 복사하는 것은 지원되지 않습니다.

사용자 지정 학습을 실행할 때 레이아웃에 대한 요금이 청구된 이유는 무엇인가요?

데이터 세트의 레이블을 생성하려면 레이아웃이 필요합니다. 사용자 지정 학습에 사용하는 데이터 세트에 사용할 수 있는 레이블 파일이 없으면 서비스에서 이를 생성합니다.

스토리지 계정

며칠 전에는 스토리지 계정에 액세스할 수 있었습니다. 현재는 다시 연결하는 데 문제가 있는 이유는 무엇인가요?

공유 액세스 서명을 만들 때 기본 기간은 48시간입니다. 48시간이 지나면 새 토큰을 만들어야 합니다.

문서 인텔리전스에서 스토리지 계정을 사용하는 시간에 대해 더 긴 시간을 설정하는 것이 좋습니다.

내 스토리지 계정이 가상 네트워크 또는 방화벽 뒤에 있는 경우 문서 인텔리전스에 데이터에 대한 액세스 권한을 부여하려면 어떻게 해야 하나요?

가상 네트워크 또는 방화벽으로 보호되는 Azure Storage 계정이 있는 경우 문서 인텔리전스는 스토리지 계정에 직접 액세스할 수 없습니다. 하지만 프라이빗 Azure 스토리지 계정 액세스 및 인증은 [Azure 리소스에 대한 관리 ID](#)를 지원합니다. 관리 ID를 사용하는 경우 문서 인텔리전스 서비스는 할당된 자격 증명을 사용하여 스토리지 계정에 액세스할 수 있습니다.

FOTT를 사용하여 프라이빗 스토리지 계정 데이터를 분석하려는 경우 가상 네트워크 또는 방화벽 뒤에 도구를 배포해야 합니다.

[문서 인텔리전스 리소스에 대한 관리 ID를 만들기 및 사용하는 방법](#)을 알아봅니다.

문서 인텔리전스 스튜디오

문서 인텔리전스 스튜디오에 액세스하려면 어떤 권한이 필요하나요?

문서 인텔리전스 스튜디오에 액세스하려면 활성 [Azure 계정](#)과 최소한 읽기 권한자 역할이 있는 구독이 필요합니다.

문서 분석 및 미리 빌드된 모델의 경우 사용자 시나리오의 역할 요구 사항은 다음과 같습니다.

- 기본
 - **Cognitive Services 사용자:** 문서 인텔리전스 스튜디오를 사용하려면 [문서 인텔리전스](#) 또는 [Azure Cognitive Services 다중 서비스](#) 리소스에 이 역할이 필요합니다.
- 고급
 - **기여자:** 리소스 그룹 또는 문서 인텔리전스 리소스를 만들려면 이 역할이 필요합니다. 기여자 역할은 Cognitive Services에 대한 키를 나열할 수 없습니다. 문서 인텔리전스 스튜디오를 사용하려면 Cognitive Services 사용자 역할이 필요합니다.

사용자 지정 모델 프로젝트의 경우 사용자 시나리오의 역할 요구 사항은 다음과 같습니다.

- 기본
 - **Cognitive Services 사용자:** 사용자 지정 모델을 학습시키거나 학습된 모델을 사용하여 분석하려면 [문서 인텔리전스](#) 또는 [Cognitive Services 다중 서비스](#) 리소스에 이 역할이 필요합니다.
 - **Storage Blob 데이터 기여자:** 프로젝트를 만들고 데이터에 레이블을 지정하려면 스토리지 계정에 이 역할이 필요합니다.
- 고급
 - **Storage 계정 기여자:** CORS(원본 간 리소스 공유) 설정을 설정하려면 스토리지 계정에 이 역할이 필요합니다. 동일한 스토리지 계정을 다시 사용하는 경우 일회성 활동입니다.

기여자 역할은 Blob의 데이터에 액세스할 수 없습니다. 문서 인텔리전스 스튜디오를 사용하려면 여전히 Storage Blob 데이터 기여자 역할이 필요합니다.

- **기여자:** 리소스 그룹과 리소스를 만들려면 이 역할이 필요합니다. 기여자 역할은 생성된 리소스 또는 스토리지를 사용할 수 있는 액세스 권한을 제공하지 않습니다. 문서 인텔리전스 스튜디오를 사용하려면 여전히 기본 역할이 필요합니다.

자세한 내용은 [Microsoft Entra 기본 제공 역할과 문서 인텔리전스 스튜디오 빠른 시작](#)의 Azure 역할 할당에 대한 섹션을 참조하세요.

문서에 여러 페이지가 있습니다. 문서 인텔리전스 스튜디오에서 두 페이지지만 분석되는 이유는 무엇인가요?

무료(F0) 계층 리소스의 경우 문서 인텔리전스 스튜디오, REST API 또는 SDK를 사용하는지에 관계없이 처음 두 페이지만 분석됩니다.

문서 인텔리전스 스튜디오에서 설정(기어) 단추를 선택하고, 리소스 탭을 선택하고, 문서 분석에 사용 중인 가격 계층을 확인합니다. 문서의 모든 페이지를 분석하려면 유료(S0) 리소스로 변경합니다.

문서 인텔리전스 스튜디오에서 딜렉터리나 구독을 어떻게 변경하나요?

문서 인텔리전스 스튜디오에서 딜렉터리를 변경하려면 설정(기어) 단추를 선택합니다. 딜렉터리 아래의 목록에서 딜렉터리를 선택한 다음 딜렉터리 전환을 선택합니다. 딜렉터리를 전환한 후 다시 로그인하라는 메시지가 표시됩니다.

구독 또는 리소스를 변경하려면 설정 아래의 리소스 탭으로 이동합니다.

내 스토리지 계정 리소스가 방화벽 또는 가상 네트워크로 구성된 경우 프로젝트 공유, 자동 레이블 지정 또는 OCR 업그레이드 작업에서 스토리지 오류가 발생하는 이유는 무엇인가요?

Azure 리소스를 설정하려면 문서 인텔리전스의 관리 ID를 참조하세요.

내 문서 인텔리전스 리소스가 방화벽 또는 가상 네트워크로 구성된 경우 자동 레이블 지정 또는 OCR

업그레이드 작업에서 “가상 네트워크/방화벽 규칙으로 인해 액세스가 거부됨”이 표시되는 이유는 무엇인가요?

전용 IP 주소 20.3.165.95를 문서 인텔리전스 리소스의 방화벽 허용 목록에 추가해야 합니다.

문서 인텔리전스 스튜디오의 레이블 지정 환경을 재사용하거나 사용자 지정하여 내 애플리케이션에 빌드할 수 있나요?

예. 문서 인텔리전스 스튜디오의 레이블 지정 환경은 [도구 키트 리포지토리](#)에 오픈 소스로 제공됩니다.

사용자 지정 프로젝트를 열 때 “Form Recognizer를 찾을 수 없음” 오류가 표시되는 이유는 무엇인가요?

이 사용자 지정 프로젝트에 바인딩된 문서 인텔리전스 리소스가 삭제되었거나 다른 리소스 그룹으로 이동되었습니다. 이 문제를 해결하는 방법에는 두 가지가 있습니다.

- 동일한 이름의 동일한 구독 및 동일한 리소스 그룹에서 문서 인텔리전스 리소스를 다시 만듭니다.
- 마이그레이션된 문서 인텔리전스 리소스를 사용하여 사용자 지정 프로젝트를 다시 만들고 동일한 스토리지 계정을 지정합니다.

컨테이너

문서 인텔리전스 컨테이너를 사용하려면 인터넷 연결이 필요하나요?

예. 문서 인텔리전스 컨테이너가 Azure에 [청구 정보](#)를 보내려면 인터넷 연결이 필요합니다. [Azure 컨테이너 보안](#)에 대해 자세히 알아보세요.

연결이 끊긴 컨테이너와 연결된 컨테이너의 차이점은 무엇인가요?

연결된 컨테이너는 Azure 계정의 문서 인텔리전스 리소스를 사용하여 Azure에 청구 정보를 보냅니다. 연결된 컨테이너를 사용하는 경우 Azure로 [청구 정보](#)를 보내려면 인터넷 연결이 필요합니다.

연결이 끊긴 컨테이너를 사용하면 인터넷에서 연결이 끊긴 API를 사용할 수 있습니다. [청구 정보](#)는 인터넷을 통해 전송되지 않습니다. 대신 구매한 약정 계층에 따라 요금이 청구됩니다. 현재 문서 인텔리전스 사용자 지정 및 청구서 모델에 연결되지 않은 컨테이너 사용이 가능합니다.

연결된 컨테이너와 연결되지 않은 컨테이너에서 제공되는 모델 기능은 동일하며 문서 인텔리전스 v2.1에서 지원됩니다.

연결된 컨테이너는 클라우드에 어떤 데이터를 보내나요?

문서 인텔리전스 연결된 컨테이너는 Azure 계정의 문서 인텔리전스 리소스를 사용하여 Azure에 청구 정보를 보냅니다. 연결된 컨테이너는 고객 데이터(예: 분석 중인 이미지 또는 텍스트)를 Microsoft에 보내지 않습니다.

연결된 컨테이너가 청구를 위해 Microsoft에 보내는 정보의 예는 [Azure AI 컨테이너 FAQ](#)를 참조하세요.

**"컨테이너가 유효한 상태가 아닙니다.
'OutOfQuota' API 키가 할당량을 초과했기 때문에
구독 유효성 검사에 실패했습니다."라는 오류가 표
시되는 이유는 무엇인가요?**

문서 인텔리전스 연결된 컨테이너는 Azure 계정의 문서 인텔리전스 리소스를 사용하여 Azure에 청구 정보를 보냅니다. 컨테이너가 청구 엔드포인트와 통신할 수 없는 경우 이 메시지가 표시될 수 있습니다.

문서 인텔리전스 샘플 레이블 지정 도구(FOTT) 컨 테이너에 로컬 스토리지를 사용할 수 있나요?

FOTT에는 로컬 스토리지를 사용하는 버전이 있습니다. 이 버전을 Windows 머신에 설치해야 합니다. [이 위치](#)에서 설치할 수 있습니다.

레이블 지정 파일이 하위 디렉터리에 있는 경우 프로젝트 페이지에서 레이블 폴더 URI를 `/shared` 또는 `/shared/sub-dir`로 지정합니다. 다른 모든 문서 인텔리전스 샘플 레이블 지정 도구 동작은 호스트된 서비스와 동일합니다.

스케일업의 모범 사례는 무엇인가요?

비동기 호출의 경우 공유 스토리지를 사용하여 여러 컨테이너를 실행할 수 있습니다.

`POST` 분석 호출을 처리하는 컨테이너는 출력을 스토리지에 저장합니다. 그러면 다른 컨테이너가 스토리지에서 결과를 가져오고 `GET` 호출을 처리할 수 있습니다. 요청 ID는 컨테이너에 연결되지 않습니다.

동기 호출의 경우 여러 컨테이너를 실행할 수 있지만 하나의 컨테이너만 요청을 처리합니다. 이는 차단 호출이므로 풀의 모든 컨테이너가 요청을 처리하고 응답을 보낼 수 있습니다. 여기서는 한 번에 하나의 컨테이너만 요청에 연결되며 폴링이 필요하지 않습니다.

공유 스토리지를 사용하여 컨테이너를 설정하려면 어떻게 해야 하나요?

컨테이너는 시작하는 동안 처리 파일을 저장할 공유 스토리지를 지정하기 위해

`Mounts:Shared` 속성을 사용합니다. 이 속성의 사용을 보려면 [컨테이너 설명서](#)를 참조하세요.

보안 및 개인 정보

Azure AI 서비스에 대한 요청을 인증하기 위한 방법과 요구 사항은 무엇인가요?

Azure 서비스에 대한 각 요청에 인증 헤더가 포함되어야 합니다. 다음과 같은 여러 방법을 사용하여 요청을 인증할 수 있습니다.

- 단일 서비스 [또는](#) 다중 서비스 [키로 인증합니다.](#)
- Microsoft Entra ID로 [인증합니다.](#)
- 고객 관리형 키를 [사용합니다.](#)
- 관리 ID에 [권한을 부여합니다.](#)

문서 인텔리전스는 내 데이터를 저장하나요?

모든 기능에 대해 문서 인텔리전스는 요청과 동일한 지역에 있는 Azure Storage에 데이터와 결과를 일시적으로 저장합니다. 그러면 분석 요청을 제출한 시간으로부터 24시간 이내에 데이터가 삭제됩니다.

[문서 인텔리전스의 데이터, 개인 정보 보호 및 보안](#)에 대해 자세히 알아봅니다.

학습된 사용자 지정 모델은 문서 인텔리전스에 어떻게 저장되고 사용되나요?

분석 및 레이블 지정 후의 중간 출력은 학습 데이터를 저장하는 동일한 Azure Storage 위치에 저장됩니다. 학습된 사용자 지정 모델은 동일한 지역의 Azure Storage에 저장되고 Azure 구독 및 API 자격 증명과 함께 논리적으로 격리됩니다.

추가 도움말 및 지원

Azure AI 문서 인텔리전스 질문에 대한 추가 솔루션은 어디에서 찾을 수 있나요?

[Microsoft Q&A](#)는 기술 관련 질문과 답변을 위한 기본 공간입니다. 문서 인텔리전스와 관련된 쿼리를 필터링할 수 있습니다.

문서에 레이블을 지정할 때 서비스에서 특정 텍스트를 인식하지 못하거나 잘못 인식하는 경우 어떻게 해야 하나요?

Microsoft는 문서 인텔리전스 OCR 모델을 지속적으로 업데이트하고 개선합니다. [문서 인텔리전스 팀에 이메일을 보내실 수 있습니다.](#) 가능하다면 이슈가 강조 표시된 샘플 문서를 공유해 주세요.

시작: 문서 인텔리전스 스튜디오

아티클 • 2024. 04. 11.

이 콘텐츠의 적용 대상은 다음과 같습니다. v4.0(미리 보기) | 이전 버전: v3.1(GA) v3.0(GA)

문서 인텔리전스 스튜디오는 문서 인텔리전스 서비스의 기능을 애플리케이션에서 시각적으로 탐색, 이해 및 통합하기 위한 온라인 도구입니다. 샘플 또는 자체 문서를 사용하여 미리 학습된 모델을 탐색하여 시작할 수 있습니다. Python SDK 및 기타 빠른 시작을 사용하여 사용자 지정 템플릿 모델을 빌드하고 애플리케이션에서 모델을 참조하는 프로젝트를 만들 수도 있습니다.

새 사용자를 위한 필수 구성 요소

- 활성 [Azure 계정](#). 계정이 없는 경우 [무료 계정](#)에 만들 수 있습니다.
- [문서 인텔리전스](#) 또는 [다중 서비스](#) 리소스입니다.

💡 팁

단일 엔드포인트/키에서 여러 Azure AI 서비스에 액세스하려는 경우 Azure AI 서비스 리소스를 만듭니다. 문서 인텔리전스 액세스에 대해서만 문서 인텔리전스 리소스를 만듭니다. 현재 [Microsoft Entra 인증](#)은 문서 인텔리전스 스튜디오에서 문서 인텔리전스 서비스 API에 액세스할 수 없습니다. 문서 인텔리전스 스튜디오를 사용하려면 액세스 키 기반 인증/로컬 인증을 사용하도록 설정해야 합니다.

Azure 역할 할당

문서 분석 및 미리 빌드된 모델의 경우 다양한 시나리오에 다음 역할 할당이 필요합니다.

- Basic
 - Cognitive Services 사용자:** 분석 페이지에 들어가려면 문서 인텔리전스 또는 Azure AI 서비스 리소스에 이 역할이 필요합니다.
- 고급
 - 기여자:** 리소스 그룹, 문서 인텔리전스 서비스 또는 Azure AI 서비스 리소스를 만들려면 이 역할이 필요합니다.

모델

미리 빌드된 모델을 사용하면 자체 모델을 빌드, 학습 및 게시할 필요 없이 앱에 문서 인텔리전스 기능을 추가할 수 있습니다. 미리 빌드된 몇 가지 모델 중에서 선택할 수 있으며, 각 모델에는 지원되는 자체 필드 세트가 있습니다. 분석 작업에 사용할 모델 선택은 분석 할 문서 유형에 따라 달라집니다. 문서 인텔리전스는 현재 다음과 같은 미리 빌드된 모델을 지원합니다.

문서 분석

- [레이아웃](#): 문서(PDF, TIFF) 및 이미지(JPG, PNG, BMP)에서 텍스트, 테이블, 선택 표시 및 구조 정보를 추출합니다.
- [읽기](#): 문서(PDF, TIFF) 및 이미지(JPG, PNG, BMP)에서 검색된 경우 텍스트 줄, 단어, 위치, 검색된 언어 및 필기 스타일을 추출합니다.

사전 제작

- [청구서](#): 청구서에서 텍스트, 선택 표시, 테이블, 키-값 쌍 및 키 정보를 추출합니다.
- [영수증](#): 영수증에서 텍스트 및 키 정보를 추출합니다.
- [의료 보험 카드](#): 미국 의료 보험 카드에서 보험사, 회원, 처방전, 그룹 번호 및 기타 주요 정보를 추출합니다.
- [W-2](#): W-2 세금 양식에서 텍스트 및 키 정보를 추출합니다.
- [ID 문서](#): 운전 면허증과 국제 여권에서 텍스트 및 키 정보를 추출합니다.

사용자 지정

- [사용자 지정 추출 모델](#): 사용자 지정 추출 모델을 사용하여 양식 및 문서에서 정보를 추출합니다. 샘플 문서에 5개까지 레이블을 지정하여 모델을 신속하게 학습시킵니다.
- [사용자 지정 분류 모델](#): 사용자 지정 분류자를 학습시켜 애플리케이션 내의 다양한 문서 유형을 구분합니다. 2개 이내의 클래스, 클래스당 5개 샘플로 모델을 빠르게 학습시킵니다.

필수 조건을 완료한 후 [문서 인텔리전스 스튜디오](#)로 이동합니다.

1. 스튜디오 홈페이지에서 문서 인텔리전스 서비스 기능을 선택합니다.
2. 이 단계는 이전 사용에서 이미 서비스 리소스를 선택하지 않은 경우 일회용 프로세스입니다. Azure 구독, 리소스 그룹 및 리소스를 선택합니다. (상단 메뉴의 "설정"에서 언제든지 리소스를 변경할 수 있습니다.) 선택 사항을 검토 및 확인합니다.
3. [분석] 단추를 선택하여 샘플 문서에 대한 분석을 실행하거나 [추가] 명령을 사용하여 문서를 사용해 봅니다.

- 화면 아래쪽의 컨트롤을 사용하여 문서 보기 확대/축소하고 회전시킵니다.
- 문서 보기에서 강조 표시된 추출된 콘텐츠를 관찰합니다. 키와 값 위로 마우스를 가져가 세부 정보를 확인합니다.
- 출력 섹션의 [결과] 탭에서 JSON 출력을 찾아 서비스 응답 형식을 이해합니다.
- 코드 탭에서 통합을 위한 샘플 코드를 찾습니다. 시작하려면 복사하고 다운로드합니다.

사용자 지정 프로젝트에 대한 추가 필수 구성 요소

Azure 계정과 문서 인텔리전스 또는 Azure AI 서비스 리소스 외에 다음이 필요합니다.

Azure Blob Storage 컨테이너

표준 성능[Azure Blob Storage 계정](#). 스토리지 계정 내에서 학습 문서를 저장하고 구성하는 컨테이너를 만듭니다. 컨테이너를 사용하여 Azure 스토리지 계정을 만드는 방법을 모르는 경우 다음 빠른 시작을 따릅니다.

- 스토리지 계정 만들기 스토리지 계정을 만드는 경우 인스턴스 세부 정보 → 성능 필드에서 표준 성능을 선택해야 합니다.
- 컨테이너를 만듭니다. 컨테이너를 만드는 경우 새 컨테이너 창에서 퍼블릭 액세스 수준 필드를 컨테이너(컨테이너 및 Blob에 대한 익명 읽기 액세스)로 설정합니다.

Azure 역할 할당

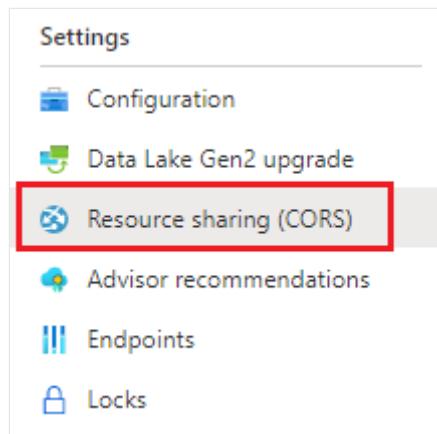
사용자 지정 프로젝트의 경우 다양한 시나리오에 다음 역할 할당이 필요합니다.

- Basic
 - Cognitive Services 사용자: 사용자 지정 모델을 학습시키거나 학습된 모델을 사용하여 분석하려면 문서 인텔리전스 또는 Azure AI 서비스 리소스에 이 역할이 필요합니다.
 - Storage Blob 데이터 기여자: 스토리지 계정이 프로젝트를 만들고 데이터에 레이블을 지정하려면 이 역할이 필요합니다.
- 고급
 - 스토리지 계정 기여자: 스토리지 계정이 CORS 설정을 설정하려면 이 역할이 필요합니다(동일한 스토리지 계정을 다시 사용하는 경우 일회성 작업).
 - 기여자: 리소스 그룹과 리소스를 만들려면 이 역할이 필요합니다.

CORS 구성

문서 인텔리전스 스튜디오에서 액세스할 수 있으려면 Azure Storage 계정에 CORS(원본 간 리소스 공유)를 구성해야 합니다. Azure Portal에서 CORS를 구성하려면 스토리지 계정의 CORS 탭에 액세스해야 합니다.

1. 스토리지 계정에 대한 CORS 탭을 선택합니다.



2. Blob 서비스에서 새 CORS 항목을 만들어 시작합니다.

3. 허용된 원본을 <https://documentintelligence.ai.azure.com>로 설정합니다.

A screenshot of the Blob service CORS configuration page. The 'Allowed origins' field contains the value 'https://documentintelligence.ai.azure.com'. Other columns include 'Allowed methods', 'Allowed headers', 'Exposed headers', and 'Max age'.

💡 팁

지정된 도메인 대신 와일드카드 문자 '*'를 사용하여 모든 원본 도메인이 CORS를 통해 요청을 수행하도록 허용할 수 있습니다.

4. 허용된 메서드에 사용할 수 있는 8가지 옵션을 모두 선택합니다.

5. 각 필드에 *를 입력하여 허용된 헤더 및 노출된 헤더를 모두 승인합니다.

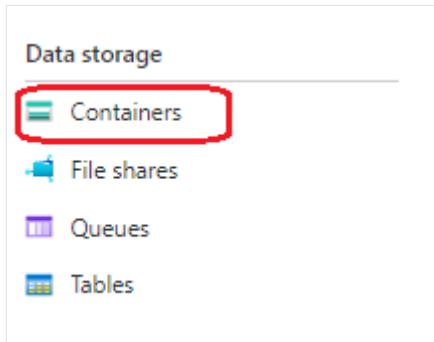
6. Max Age를 120초 또는 허용되는 값으로 설정합니다.

7. 페이지 위쪽에 있는 저장

이제 문서 인텔리전스 스튜디오의 스토리지 계정을 사용하도록 CORS를 구성해야 합니다.

샘플 문서 세트

1. Azure Portal [에](#) 로그인하고 **스토리지 계정**>**데이터 스토리지**>**컨테이너**로 이동합니다.



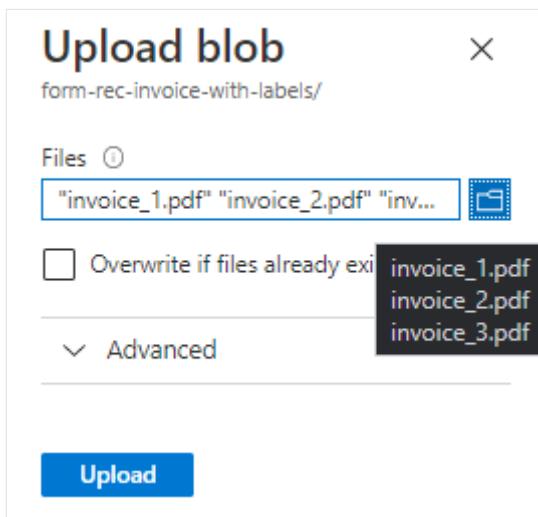
2. 목록에서 **컨테이너**를 선택합니다.

3. 페이지 위쪽의 메뉴에서 **업로드**를 선택합니다.



4. **Blob 업로드** 창이 표시됩니다.

5. 업로드할 파일을 선택합니다.



① 참고

기본적으로 Studio는 컨테이너의 루트에 있는 문서를 사용합니다. 그러나 사용자 지정 양식 프로젝트 만들기 단계에서 폴더 경로를 지정하여 폴더에 구성된 데이터를 사용할 수 있습니다. 하위 폴더에 데이터 구성 [을 참조하세요](#).

사용자 지정 모델

사용자 지정 모델을 만들려면 프로젝트 구성부터 시작합니다.

1. Studio 홈에서 사용자 지정 모델 카드를 선택하여 사용자 지정 모델 페이지를 엽니다.
2. "프로젝트 만들기" 명령을 사용하여 새 프로젝트 구성 마법사를 시작합니다.
3. 프로젝트 세부 정보를 입력하고, Azure 구독 및 리소스, 데이터가 포함된 Azure Blob 스토리지 컨테이너를 선택합니다.
4. 설정을 검토하고 제출하여 프로젝트를 만듭니다.
5. 레이블 지정 프로세스를 빠르게 시작하려면 자동 레이블 기능을 사용하여 이미 학습된 모델이나 미리 빌드된 모델 중 하나를 사용하여 레이블을 지정합니다.
6. 처음부터 수동으로 레이블을 지정하려면 추출하려는 레이블과 해당 형식을 정의합니다.
7. 문서의 텍스트를 선택하고 드롭다운 목록 또는 레이블 창에서 레이블을 선택합니다.
8. 레이블을 4개 이상의 문서에 지정하여 레이블을 5개 이상의 문서에 지정합니다.
9. 학습 명령을 선택하고 모델 이름을 입력한 후 사용자 지정 모델 학습을 시작할 신경(권장) 모델 또는 템플릿 모델을 선택합니다.
10. 모델이 준비되면 테스트 명령을 사용하여 테스트 문서를 통해 모델의 유효성을 검사하고 결과를 관찰합니다.

Applied AI | Form Recognizer Studio - Preview

Form Recognizer Studio

Get started with Form Recognizer Studio

Extract text, key-value pairs, tables, and structures from forms and documents using common layouts and prebuilt models, or create your own custom models. [Learn more](#)

Extract employee, employer, wage information, etc. from US W-2 forms.

[Try it out](#)

Custom models

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

Custom model

Build a custom model to extract key-value pairs, selection marks, tables and more from forms and documents.

[Create new](#)

Learning resources

- Read the documentation**
Learn more about extracting text, structure, and fields and values from pre-trained and custom models.
[Form Recognizer documentation](#)
- Explore our quick start guides**
Explore our samples and discover the things you can build with Azure Cognitive Services.
[Quick start guides](#)
- Microsoft Q&A**
For quick and reliable answers, engage with us on Microsoft Q&A, Azure's preferred destination for community support.
[Post a question](#)
- Microsoft Learn**
Discover new skills, find certifications, and advance your career in minutes with interactive, hands-on learning paths.
[Explore Learn](#)

테이블로 레이블 지정

① 참고

- API 버전 2022-06-30-preview 이상의 릴리스에서 사용자 지정 템플릿 모델은 페이지 간 테이블 형식 필드(테이블)에 대한 지원을 추가합니다.
- API 버전 2022-06-30-preview 이상 릴리스와 함께 사용자 지정 신경망 모델은 테이블 형식 필드(테이블)를 지원하고 API 버전 2022-08-31 이상으로 학습된 모델은 테이블 형식 필드 레이블을 허용합니다.

1. [삭제] 명령을 사용하여 필요하지 않은 모델을 삭제합니다.
2. 오프라인 보기是为了 위한 모델 세부 정보를 다운로드합니다.
3. 여러 모델을 선택하고, 애플리케이션에서 사용할 새 모델로 구성합니다.

테이블을 시각적 패턴으로 사용:

사용자 지정 양식 모델의 경우 사용자 지정 모델을 만드는 동안 문서에서 데이터 컬렉션을 추출해야 할 수 있습니다. 데이터 컬렉션은 몇 가지 형식으로 나타날 수 있습니다. 테이블을 시각적 패턴으로 사용:

- 지정된 필드(열) 세트에 대한 동적 또는 변수 값(행)의 개수
- 지정된 필드(열 및/또는 행) 세트에 대한 특정 값(행)의 컬렉션

동적 테이블로 레이블 지정

동적 테이블을 사용하여 지정된 필드(열) 세트에 대한 값(행)의 변수 수를 추출합니다.

1. 새 "테이블" 형식 레이블을 추가하고, "동적 테이블" 형식을 선택한 다음, 레이블 이름을 지정합니다.
2. 필요한 열(필드) 및 행(데이터용)의 수를 추가합니다.
3. 페이지에서 텍스트를 선택한 다음 텍스트에 할당할 셀을 선택합니다. 모든 문서의 모든 페이지에 있는 모든 행과 열에 대해 반복합니다.

Cognitive Services | Form Recognizer Studio - Preview

Form Recognizer Studio > Custom Form > Fabricam > Label data

Label data

Region

DELIVERED ITEMS

Product	Unit price	Quantity	Total
Item 1	10.00	2	20.00
Item 2	10.00	1	10.00
Item 3	20.00	1	10.00

Reviewed and approved by

1 of 1

고정 테이블로 레이블 지정

고정 테이블을 사용하여 지정된 필드(열 및/또는 행) 세트에 대한 특정 값(행)의 컬렉션을 추출합니다.

1. 새 "테이블" 형식 레이블을 만들고, "고정 테이블" 형식을 선택한 다음, 이름을 지정 합니다.
2. 두 필드 세트에 해당하는 필요한 열과 행의 수를 추가합니다.
3. 페이지에서 텍스트를 선택한 다음 텍스트에 할당할 셀을 선택합니다. 다른 문서에 대해 반복합니다.

Cognitive Services | Form Recognizer Studio - Preview

Form Recognizer Studio > Custom Form > Fabricam > Label data

Label data

Region

CONTOSO

CONTOSO PACKAGING QUALITY ASSURANCE

Delivery report for package delivered to Arlington WA from Seattle via the Contoso Messenger service. Please contact the delivery manager for questions related to this delivery. Quality inspection by package quality assurance team at the last mile facility.

Courier: Preston M.

Delivery date: 07/23/2019 Number of items: 3

PACKAGING QUALITY

	%	Factor	Result
Structure	0.00	2.0	0.00
Integrity	0.00	1.5	0.00
Durability	1.98	0.3	0.59
Deformation	1.02	0.3	0.31

DELIVERED ITEMS

Product	Unit price	Quantity	Total
Item 1	10.00	2	20.00
Item 2	10.00	1	10.00
Item 3	20.00	1	10.00

Reviewed and approved by

1 of 1

서명 검색

① 참고

서명 필드는 현재 사용자 지정 템플릿 모델에 대해서만 지원됩니다. 사용자 지정 신경망 모델을 학습할 때 레이블이 지정된 서명 필드는 무시됩니다.

서명 검색 레이블 지정(사용자 지정 양식에만 해당)하려면 다음을 수행합니다.

- 레이블 보기에서 새 "서명" 형식 레이블을 만들고, 이름을 지정합니다.
- [영역] 명령을 사용하여 사각형 영역을 서명의 예상 위치에 만듭니다.
- 그린 영역을 선택하고 서명 형식 레이블을 선택하여 그린 영역에 할당합니다. 다른 문서에 대해 반복합니다.

Product	Unit price	Quantity	Total
Item 1	10.00	2	20.00
Item 2	10.00	1	10.00
Item 3	20.00	1	10.00

다음 단계

- マイグレーション 버전의 REST API와의 차이점을 알아보려면 [문서 인텔리전스 v3.1 마이그레이션 가이드](#)를 따릅니다.
- 새 SDK를 통해 애플리케이션에서 v3.0 기능을 사용해 보려면 [v3.0 SDK 빠른 시작](#)을 살펴보세요.
- 새 REST API를 통해 v3.0 기능을 사용해 보려면 [v3.0 REST API 빠른 시작](#)을 참조하세요.

[문서 인텔리전스 스튜디오를 시작합니다](#).

문서 인텔리전스 시작

아티클 • 2024. 03. 25.

① 중요

- Azure Cognitive Services Form Recognizer는 이제 Azure AI 문서 인텔리전스입니다.
- 일부 플랫폼은 여전히 이름 변경 업데이트를 기다리고 있습니다.
- 설명서에서 Form Recognizer 또는 문서 인텔리전스에 대한 모든 언급은 동일한 Azure 서비스를 나타냅니다.

이 콘텐츠의 적용 대상: v4.0(미리 보기) 이전 버전: v3.1(GA) v3.0(GA)

- Azure AI 문서 인텔리전스 최신 미리 보기 버전(2024-02-29-preview)을 시작합니다.
- Azure AI 문서 인텔리전스/Form Recognizer는 기계 학습을 사용하여 문서에서 키-값 쌍, 텍스트, 테이블 및 주요 데이터를 추출하는 클라우드 기반 Azure AI 서비스입니다.
- 프로그래밍 언어 SDK를 사용하거나 REST API를 호출하여 문서 처리 모델을 워크플로 및 애플리케이션에 쉽게 통합할 수 있습니다.
- 이 빠른 시작의 경우 기술을 학습할 때 무료 서비스를 사용하는 것이 좋습니다. 체험판 페이지는 한 달에 500페이지로 제한됩니다.

API 기능 및 개발 옵션에 대한 자세한 내용은 [개요](#) 페이지를 참조하세요.

[클라이언트 라이브러리](#) | [SDK 참조](#) | [REST API 참조](#) | [패키지](#) | [샘플](#) | [지원되는 REST API 버전](#)

이 빠른 시작에서는 다음 기능을 사용하여 양식 및 문서에서 데이터와 값을 분석하고 추출합니다.

- 레이아웃 모델** - 모델을 학습시킬 필요 없이 문서의 라디오 단추 및 확인란과 같은 테이블, 선, 단어, 선택 표시를 분석하고 추출합니다.
- 미리 빌드된 모델** - 미리 빌드된 모델을 사용하여 특정 문서 형식에서 공통 필드를 분석하고 추출합니다.

필수 조건

- Azure 구독 - [체험 구독 만들기](#)
- 현재 버전의 [Visual Studio IDE](#)
- Azure AI 서비스 또는 문서 인텔리전스 리소스입니다. Azure 구독이 있으면 Azure Portal에서 [단일 서비스](#) 또는 [Azure AI 다중 서비스](#) 리소스를 만들어 키와 앤드포인트를 가져옵니다.
- 평가판 가격 책정 계층([F0](#))을 통해 서비스를 사용해보고, 나중에 프로덕션용 유료 계층으로 업그레이드할 수 있습니다.

💡 팁

단일 엔드포인트/키에서 여러 Azure AI 서비스에 액세스하려는 경우 Azure AI 서비스 리소스를 만듭니다. 문서 인텔리전스 액세스에 대해서만 문서 인텔리전스 리소스를 만듭니다. [Microsoft Entra 인증](#)을 사용하려는 경우 단일 서비스 리소스가 필요합니다.

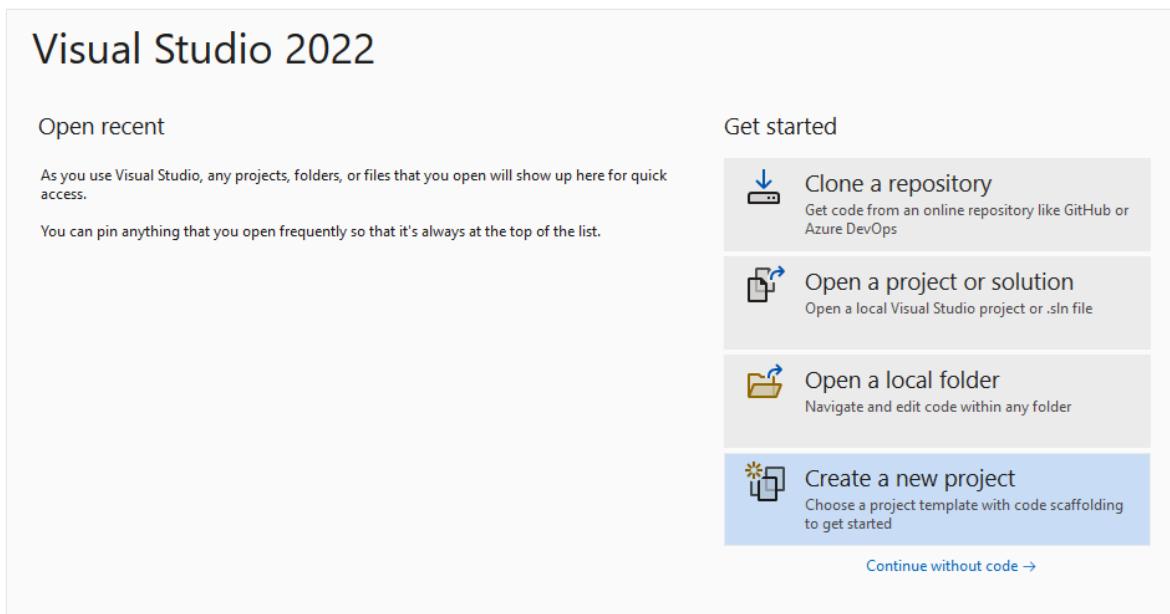
- 리소스를 배포한 후 [리소스로 이동](#)을 선택합니다. 애플리케이션을 문서 인텔리전스 API에 연결하려면 만든 리소스의 키와 앤드포인트가 필요합니다. 키와 앤드포인트는 이 빠른 시작의 뒷부분에서 코드에 붙여넣습니다.

The screenshot shows the Azure portal interface for managing a 'Contoso-DI' resource. The left sidebar lists various management options like Overview, Activity log, Access control (IAM), Tags, and Resource Management. Under Resource Management, the 'Keys and Endpoint' option is selected and highlighted with a red box. The main content area shows a summary of keys and endpoints. A callout box provides instructions about securely storing the keys. Below this, a 'Show Keys' button leads to a detailed view where 'KEY 1' and 'KEY 2' are listed as masked text inputs, each with a copy icon. Further down, the 'Endpoint' field contains the URL 'https://contoso-di.cognitiveservices.azure.com/' with its own copy icon. The entire 'KEY 1' input field and the 'Endpoint' field are also highlighted with red boxes.

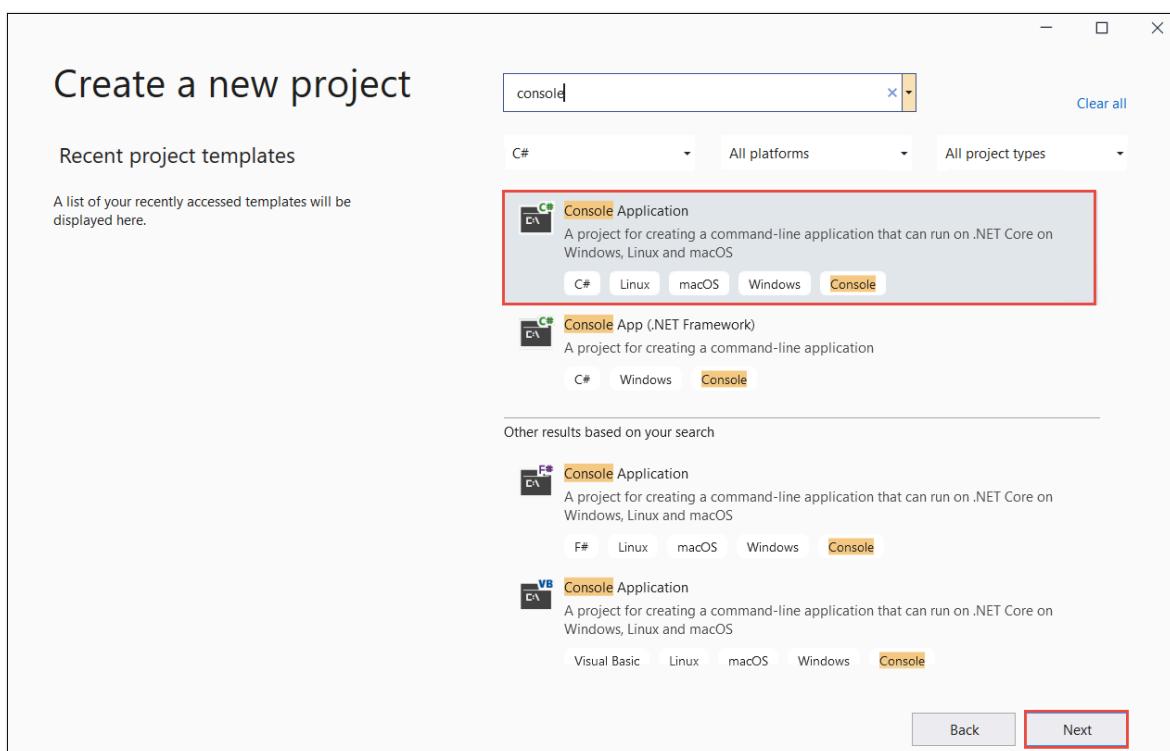
설정

1. Visual Studio를 시작합니다.

2. 시작 페이지에서 새 프로젝트 만들기를 선택합니다.



3. 새 프로젝트 만들기 페이지의 검색 상자에서 콘솔을 입력합니다. 콘솔 애플리케이션 템플릿을 선택하고, 다음을 선택합니다.



4. 새 프로젝트 구성 대화 상자 창의 [프로젝트 이름] 상자에서 `doc_intel_quickstart`를 입력합니다. 다음을 선택합니다.

5. 추가 정보 대화 창에서 .NET 6.0(장기 지원)을 선택한 다음, 만들기를 선택합니다.

Additional information

Console App

C#

Linux

macOS

Windows

Console

Framework

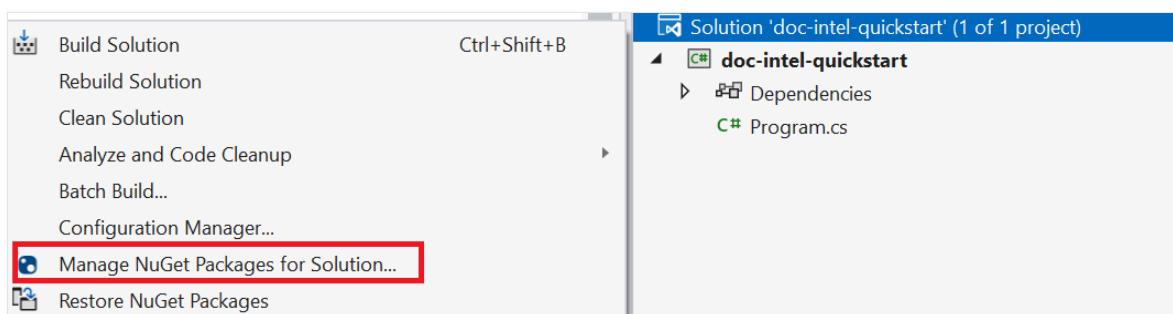
.NET 8.0 (Long Term Support)

Do not use top-level statements

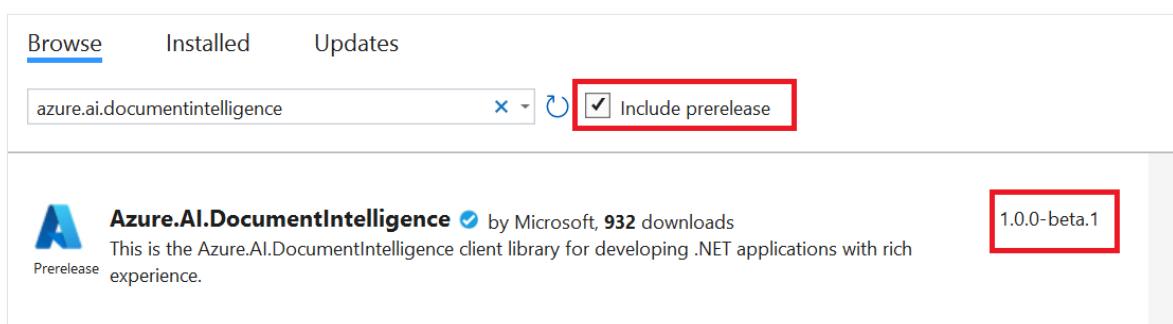
Enable native AOT publish

NuGet을 사용하여 클라이언트 라이브러리 설치

1. doc_intel_quickstart 프로젝트를 마우스 오른쪽 단추로 클릭하고 NuGet 패키지 관리...를 선택합니다.



2. 찾아보기 탭을 선택하고 Azure.AI.DocumentIntelligence를 입력합니다. **Include prerelease** 확인란을 선택하고 드롭다운 메뉴에서 버전 1.0.0-beta.1을 선택합니다.



애플리케이션 빌드

문서 인텔리전스 서비스와 상호 작용하려면 `DocumentIntelligenceClient` 클래스의 인스턴스를 만들어야 합니다. 이렇게 하려면 Azure Portal에서 `key`를 사용하여 `AzureKeyCredential`을 만들고, `AzureKeyCredential` 및 문서 인텔리전스 `endpoint`를 사용하여 `DocumentIntelligenceClient` 인스턴스를 만듭니다.

① 참고

- .NET 6부터 `console` 템플릿을 사용하는 새 프로젝트는 이전 버전과 다른 새 프로그램 스타일을 생성합니다.
- 새 출력에서는 작성해야 하는 코드를 간소화하는 최신 C# 기능을 사용합니다.
- 최신 버전을 사용하는 경우 `Main` 메서드 본문을 작성하기만 하면 됩니다. 최상위 문, 전역 `using` 지시문 또는 암시적 `using` 지시문을 포함할 필요가 없습니다.
- 자세한 내용은 [최상위 문을 생성하는 새 C# 템플릿](#)을 참조하세요.

1. `Program.cs` 파일을 엽니다.

2. `Console.WriteLine("Hello World!")` 줄을 포함하여 기존 코드를 삭제하고 다음 코드 샘플 중 하나를 선택하여 복사하고 애플리케이션의 `Program.cs` 파일에 붙여넣습니다.

- [레이아웃 모델](#)
- [미리 빌드된 모델](#)

② 중요

완료되면 코드에서 키를 제거하고 공개적으로 게시하지 마세요. 프로덕션의 경우 [Azure Key Vault](#)와 같은 자격 증명을 안전하게 저장하고 액세스하는 방법을 사용합니다. 자세한 내용은 Azure AI 서비스 [보안](#)을 참조하세요.

레이아웃 모델

문서에서 텍스트, 선택 표시, 텍스트 스타일, 테이블 구조, 경계 영역 좌표를 추출합니다.

- ✓ 이 예에서는 **URI의 문서 파일**이 필요합니다. 이 빠른 시작을 위해 [샘플 문서](#)를 사용할 수 있습니다.
- ✓ 파일 URI 값을 스크립트 상단에 있는 `Uri fileUri` 변수에 추가합니다.
- ✓ URI의 지정된 파일에서 레이아웃을 추출하려면 `StartAnalyzeDocumentFromUri` 메서드를 사용하고 `prebuilt-layout`을 모델 ID로 전달합니다. 반환된 값은 제출된 문서의 데이터가 포함된 `AnalyzeResult` 개체입니다.

`Program.cs` 파일에 다음 코드 샘플을 추가합니다. Azure Portal 문서 인텔리전스 인스턴스의 값으로 키 및 엔드포인트 변수를 업데이트해야 합니다.

```
using Azure;
using Azure.AI.DocumentIntelligence;

//set `<your-endpoint>` and `<your-key>` variables with the values from the
Azure portal to create your `AzureKeyCredential` and
`DocumentIntelligenceClient` instance
string endpoint = "<your-endpoint>";
string key = "<your-key>";
AzureKeyCredential credential = new AzureKeyCredential(key);
DocumentIntelligenceClient client = new DocumentIntelligenceClient(new
Uri(endpoint), credential);

//sample document
Uri fileUri = new Uri ("https://raw.githubusercontent.com/Azure-
Samples/cognitive-services-REST-api-samples/master/curl/form-
recognizer/sample-layout.pdf");

Operation<AnalyzeResult> operation = await
client.AnalyzeDocumentAsync(WaitUntil.Completed, "prebuilt-layout",
fileUri);

AnalyzeResult result = operation.Value;

foreach (DocumentPage page in result.Pages)
{
    Console.WriteLine($"Document Page {page.PageNumber} has
{page.Lines.Count} line(s), {page.Words.Count} word(s)," +
    $" and {page.SelectionMarks.Count} selection mark(s.)");

    for (int i = 0; i < page.Lines.Count; i++)
    {
        DocumentLine line = page.Lines[i];

        Console.WriteLine($"  Line {i}:");
        Console.WriteLine($"    Content: '{line.Content}'");

        Console.Write("    Bounding polygon, with points ordered
clockwise:");
        for (int j = 0; j < line.Polygon.Count; j += 2)
        {
            Console.Write($" ({line.Polygon[j]}, {line.Polygon[j + 1]})");
        }

        Console.WriteLine();
    }

    for (int i = 0; i < page.SelectionMarks.Count; i++)
    {
        DocumentSelectionMark selectionMark = page.SelectionMarks[i];

        Console.WriteLine($"  Selection Mark {i} is
{selectionMark.State}");
        Console.WriteLine($"    State: {selectionMark.State}");
    }
}
```

```

        Console.WriteLine("    Bounding polygon, with points ordered
clockwise:");
        for (int j = 0; j < selectionMark.Polygon.Count; j++)
        {
            Console.WriteLine($" ({selectionMark.Polygon[j]}, 
{selectionMark.Polygon[j + 1]})");
        }

        Console.WriteLine();
    }
}

for (int i = 0; i < result.Paragraphs.Count; i++)
{
    DocumentParagraph paragraph = result.Paragraphs[i];

    Console.WriteLine($"Paragraph {i}:");
    Console.WriteLine($"  Content: {paragraph.Content}");

    if (paragraph.Role != null)
    {
        Console.WriteLine($"    Role: {paragraph.Role}");
    }
}

foreach (DocumentStyle style in result.Styles)
{
    // Check the style and style confidence to see if text is handwritten.
    // Note that value '0.8' is used as an example.

    bool isHandwritten = style.IsHandwritten.HasValue && style.IsHandwritten
== true;

    if (isHandwritten && style.Confidence > 0.8)
    {
        Console.WriteLine($"Handwritten content found:");

        foreach (DocumentSpan span in style.Spans)
        {
            var handwrittenContent = result.Content.Substring(span.Offset,
span.Length);
            Console.WriteLine($"    {handwrittenContent}");
        }
    }
}

for (int i = 0; i < result.Tables.Count; i++)
{
    DocumentTable table = result.Tables[i];

    Console.WriteLine($"Table {i} has {table.RowCount} rows and
{table.ColumnCount} columns.");

    foreach (DocumentTableCell cell in table.Cells)

```

```
{  
    Console.WriteLine($" Cell ({cell.RowIndex}, {cell.ColumnIndex}) is  
a '{cell.Kind}' with content: {cell.Content}");  
}  
}
```

애플리케이션 실행

애플리케이션에 코드 샘플을 추가한 후 `formRecognizer_quickstart` 옆에 있는 녹색 시작 단추를 선택하여 프로그램을 빌드하고 실행하거나 F5를 누릅니다.



미리 빌드된 모델

미리 빌드된 모델을 사용하여 특정 문서 형식에서 공통 필드를 분석하고 추출합니다. 이 예제에서는 **미리 빌드된 청구서 모델**을 사용하여 청구서를 분석합니다.

💡 팁

청구서에 국한되지 않습니다. 몇 가지 미리 빌드된 모델 중에서 선택할 수 있으며, 각 모델에는 자체 지원 필드 집합이 있습니다. `analyze` 작업에 사용할 모델은 분석할 문서 형식에 따라 달라집니다. [모델 데이터 추출](#)을 참조하세요.

- ✓ 미리 빌드된 청구서 모델을 사용하여 청구서를 분석합니다. 이 빠른 시작에는 [샘플 청구서 문서](#)를 사용할 수 있습니다.
- ✓ 파일 URI 값을 `Program.cs` 파일의 상단에 있는 `Uri invoiceUri` 변수에 추가했습니다.
- ✓ URI에서 지정된 파일을 분석하려면 `StartAnalyzeDocumentFromUri` 메서드를 사용하고 `prebuilt-invoice`를 모델 ID로 전달합니다. 반환된 값은 제출된 문서의 데이터가 포함된 `AnalyzeResult` 개체입니다.
- ✓ 간단히 하기 위해 서비스에서 반환하는 모든 키-값 쌍이 여기에 표시되지는 않습니다. 지원되는 모든 필드 및 해당 형식의 목록을 보려면 [청구서 개념 페이지](#)를 참조하세요.

`Program.cs` 파일에 다음 코드 샘플을 추가합니다. Azure Portal 문서 인텔리전스 인스턴스의 값으로 키 및 앤드포인트 변수를 업데이트해야 합니다.

C#

```
using Azure;
using Azure.AI.DocumentIntelligence;

//set `<your-endpoint>` and `<your-key>` variables with the values from the
Azure portal to create your `AzureKeyCredential` and
`DocumentIntelligenceClient` instance
string endpoint = "<your-endpoint>";
string key = "<your-key>";
AzureKeyCredential credential = new AzureKeyCredential(key);
DocumentIntelligenceClient client = new DocumentIntelligenceClient(new
Uri(endpoint), credential);

//sample invoice document

Uri invoiceUri = new Uri ("https://raw.githubusercontent.com/Azure-
Samples/cognitive-services-REST-api-samples/master/curl/form-
recognizer/sample-invoice.pdf");

Operation<AnalyzeResult> operation = await
client.AnalyzeDocumentAsync(WaitUntil.Completed, "prebuilt-invoice",
invoiceUri);

AnalyzeResult result = operation.Value;

for (int i = 0; i < result.Documents.Count; i++)
{
    Console.WriteLine($"Document {i}:");

    AnalyzedDocument document = result.Documents[i];

    if (document.Fields.TryGetValue("VendorName", out DocumentField
vendorNameField)
        && vendorNameField.Type == DocumentFieldType.String)
    {
        string vendorName = vendorNameField.ValueString;
        Console.WriteLine($"Vendor Name: '{vendorName}', with confidence
{vendorNameField.Confidence}");
    }

    if (document.Fields.TryGetValue("CustomerName", out DocumentField
customerNameField)
        && customerNameField.Type == DocumentFieldType.String)
    {
        string customerName = customerNameField.ValueString;
        Console.WriteLine($"Customer Name: '{customerName}', with confidence
{customerNameField.Confidence}");
    }

    if (document.Fields.TryGetValue("Items", out DocumentField itemsField)
        && itemsField.Type == DocumentFieldType.Array)
    {
        foreach (DocumentField itemField in itemsField.ValueArray)
        {

```

```

        Console.WriteLine("Item:");

        if (itemField.Type == DocumentFieldType.Object)
        {
            IReadOnlyDictionary<string, DocumentField> itemFields =
            itemField.ValueObject;

                if (itemFields.TryGetValue("Description", out DocumentField
itemDescriptionField)
                    && itemDescriptionField.Type ==
DocumentFieldType.String)
                {
                    string itemDescription =
itemDescriptionField.ValueString;
                    Console.WriteLine($" Description: '{itemDescription}',
with confidence {itemDescriptionField.Confidence}");
                }

                if (itemFields.TryGetValue("Amount", out DocumentField
itemAmountField)
                    && itemAmountField.Type == DocumentFieldType.Currency)
                {
                    CurrencyValue itemAmount =
itemAmountField.ValueCurrency;
                    Console.WriteLine($" Amount:
'{itemAmount.CurrencySymbol}{itemAmount.Amount}', with confidence
{itemAmountField.Confidence}");
                }
            }
        }

        if (document.Fields.TryGetValue("SubTotal", out DocumentField
subTotalField)
            && subTotalField.Type == DocumentFieldType.Currency)
        {
            CurrencyValue subTotal = subTotalField.ValueCurrency;
            Console.WriteLine($"Sub Total: '{subTotal.CurrencySymbol}
{subTotal.Amount}', with confidence {subTotalField.Confidence}");
        }

        if (document.Fields.TryGetValue("TotalTax", out DocumentField
totalTaxField)
            && totalTaxField.Type == DocumentFieldType.Currency)
        {
            CurrencyValue totalTax = totalTaxField.ValueCurrency;
            Console.WriteLine($"Total Tax: '{totalTax.CurrencySymbol}
{totalTax.Amount}', with confidence {totalTaxField.Confidence}");
        }

        if (document.Fields.TryGetValue("InvoiceTotal", out DocumentField
invoiceTotalField)
            && invoiceTotalField.Type == DocumentFieldType.Currency)
        {
            CurrencyValue invoiceTotal = invoiceTotalField.ValueCurrency;

```

```
        Console.WriteLine($"Invoice Total: '{invoiceTotal.CurrencySymbol}  
{invoiceTotal.Amount}', with confidence {invoiceTotalField.Confidence}");  
    }  
}
```

애플리케이션 실행

애플리케이션에 코드 샘플을 추가한 후 formRecognizer_quickstart 옆에 있는 녹색 시작 단추를 선택하여 프로그램을 빌드하고 실행하거나 F5를 누릅니다.



완료되었습니다. 축하합니다!

이 빠른 시작에서는 문서 인텔리전스 모델을 사용하여 다양한 양식과 문서를 분석했습니다. 다음으로 문서 인텔리전스 스튜디오 및 참조 설명서를 탐색하여 문서 인텔리전스 API에 대해 자세히 알아봅니다.

다음 단계

- 향상된 환경과 고급 모델 품질을 위해 [문서 인텔리전스 스튜디오](#) 사용해 보기
- v3.1에서 v4.0으로의 마이그레이션은 [변경 로그 마이그레이션 가이드](#)를 참조하세요.

문서 인텔리전스 모델 사용

아티클 • 2024. 04. 12.

이 콘텐츠의 적용 대상은 다음과 같습니다. ✓ v4.0(미리 보기) | 이전 버전: ✓ v3.1(GA) ✓ v3.0(GA) ✓ v2.1(GA)

이 가이드에서는 애플리케이션 및 워크플로에 문서 인텔리전스 모델을 추가하는 방법을 알아봅니다. 선택한 프로그래밍 언어 SDK 또는 REST API를 사용합니다.

Azure AI 문서 인텔리전스는 기계 학습을 사용하여 문서에서 핵심 텍스트와 구조 요소를 추출하는 클라우드 기반 Azure AI 서비스입니다. 기술을 학습할 때 체험판 서비스를 이용하는 것이 좋습니다. 체험판 페이지는 한 달에 500페이지로 제한됩니다.

다음 문서 인텔리전스 모델 중에서 선택하고 양식 및 문서에서 데이터 및 값을 분석 및 추출합니다.

- ✓ [사전 빌드된 읽기](#) 모델은 모든 문서 인텔리전스 모델의 핵심이며 행, 단어, 위치 및 언어를 감지할 수 있습니다. 레이아웃, 일반 문서, 미리 빌드 및 사용자 지정 모델은 모두 `read` 모델을 문서에서 텍스트를 추출하기 위한 기반으로 사용합니다.
- ✓ [prebuilt-layout](#) 모델은 문서와 이미지에서 텍스트와 텍스트 위치, 표, 선택 표시, 구조 정보를 추출합니다. 선택적 쿼리 문자열 매개 변수 `features=keyValuePairs` 가 사용하도록 설정된 레이아웃 모델을 사용하여 키/값 쌍을 추출할 수 있습니다.
- ✓ [prebuilt-contract](#) 모델은 계약에서 주요 정보를 추출합니다.
- ✓ [prebuilt-healthInsuranceCard.us](#) 모델은 미국 의료 보험 카드에서 주요 정보를 추출합니다.
- ✓ [미리 만들어진 세금 문서 모델](#)은 미국 세금 양식에 보고된 정보를 추출합니다.
- ✓ [prebuilt-invoice](#) 모델은 다양한 형식과 품질의 판매 청구서에서 주요 필드와 항목을 추출합니다. 필드에는 전화로 캡처한 이미지, 스캔한 문서 및 디지털 PDF가 포함됩니다.
- ✓ [prebuilt-receipt](#) 모델은 인쇄 및 필기 판매 영수증에서 주요 정보를 추출합니다.
- ✓ [prebuilt-idDocument](#) 모델은 미국 운전면허증, 국가별 여권 약력 페이지, 미국 주 ID, 사회 보장 카드 및 영주권 카드에서 주요 정보를 추출합니다.

필수 조건

- Azure 구독 – [체험 구독을 만듭니다](#).

- Visual Studio IDE ↗
- Azure AI 서비스 또는 문서 인텔리전스 리소스입니다. 단일 서비스 ↗ 또는 다중 서비스 ↗를 만듭니다. 평가판 가격 책정 계층(F0)을 통해 서비스를 사용해보고, 나중에 프로덕션용 유료 계층으로 업그레이드할 수 있습니다.
- 애플리케이션을 Azure 문서 인텔리전스 서비스에 연결하려면 만든 리소스의 키와 엔드포인트
 1. 리소스를 배포한 후 **리소스로 이동**을 선택합니다.
 2. 왼쪽 탐색 메뉴에서 **키 및 엔드포인트**를 선택합니다.
 3. 이 문서에서 나중에 사용할 키와 **엔드포인트** 중 하나를 복사합니다.

- URL 위치의 문서 파일이 프로젝트에서는 각 기능에 대해 다음 표에 제공된 샘플 양식을 사용할 수 있습니다.

[] 테이블 확장

기능	modelID	document-url
읽기 모델	prebuilt-read	샘플 브로셔 ↗
레이아웃 모델	prebuilt-layout	샘플 예약 확인서 ↗
W-2 양식 모델	prebuilt-tax.us.w2	샘플 W-2 양식 ↗
청구서 모델	prebuilt-invoice	샘플 청구서 ↗
영수증 모델	prebuilt-receipt	샘플 영수증 ↗

기능	modelID	document-url
ID 문서 모델	prebuilt-idDocument	샘플 ID 문서
명함 모델	prebuilt-businessCard	샘플 명함

환경 변수 설정

문서 인텔리전스 서비스와 상호 작용하려면 `DocumentAnalysisClient` 클래스의 인스턴스를 만들어야 합니다. 이렇게 하려면 Azure portal에서 `key` 및 `endpoint`를 사용하여 클라이언트를 인스턴스화합니다. 이 프로젝트에서는 환경 변수를 사용하여 자격 증명을 저장하고 액세스합니다.

① 중요

키를 코드에 직접 포함하지 말고 공개적으로 게시하지 마세요. 프로덕션의 경우 [Azure Key Vault](#)와 같은 안전한 방법을 사용하여 자격 증명을 저장하고 액세스합니다. 자세한 내용은 [Azure AI 서비스 보안](#)을 참조하세요.

문서 인텔리전스 리소스 키에 대한 환경 변수를 설정하려면 콘솔 창을 열고 운영 체제 및 개발 환경에 대한 지침을 따르세요. `<yourKey>` 및 `<yourEndpoint>`를 Azure Portal 리소스의 값으로 바꿉니다.

Windows

Windows의 환경 변수는 대/소문자를 구분하지 않습니다. 일반적으로 대문자로 선언되며 단어는 밑줄로 조인됩니다. 명령 프롬프트에서 다음 명령을 실행합니다.

1. 주요 변수를 설정합니다.

콘솔

```
setx DI_KEY <yourKey>
```

2. 엔드포인트 변수 설정

콘솔

```
setx DI_ENDPOINT <yourEndpoint>
```

3. 환경 변수를 설정한 후 명령 프롬프트 창을 닫습니다. 값은 다시 변경할 때까지 유지됩니다.

4. 환경 변수를 읽는 실행 중인 프로그램을 다시 시작합니다. 예를 들어, Visual Studio 또는 Visual Studio Code를 편집기로 사용하는 경우 샘플 코드를 실행하기 전에 다시 시작합니다.

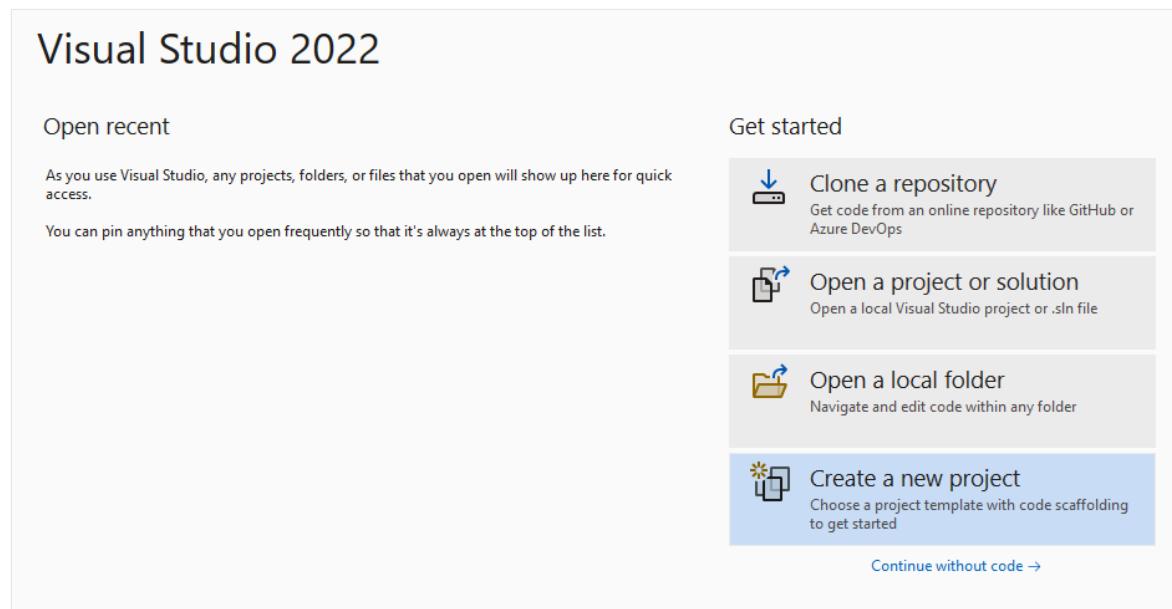
다음은 환경 변수와 함께 사용할 수 있는 몇 가지 유용한 명령입니다.

테이블 확장

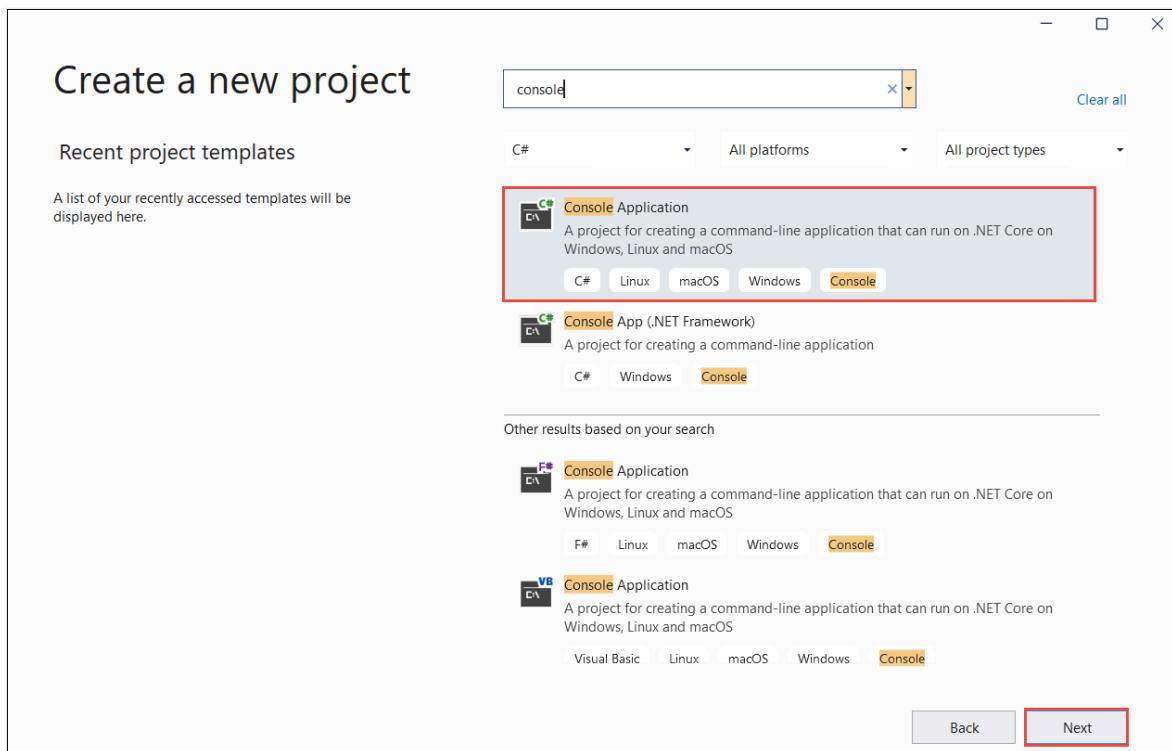
명령	작업	예시
<code>setx VARIABLE_NAME=</code>	값을 빈 문자열로 설정하여 환경 변수를 삭제합니다.	<code>setx DI_KEY=</code>
<code>setx VARIABLE_NAME=value</code>	환경 변수 값을 설정하거나 변경합니다.	<code>setx DI_KEY=<yourKey></code>
<code>set VARIABLE_NAME</code>	특정 환경 변수의 값을 표시합니다.	<code>set DI_KEY</code>
<code>set</code>	모든 환경 변수를 표시합니다.	<code>set</code>

프로그래밍 환경 설정

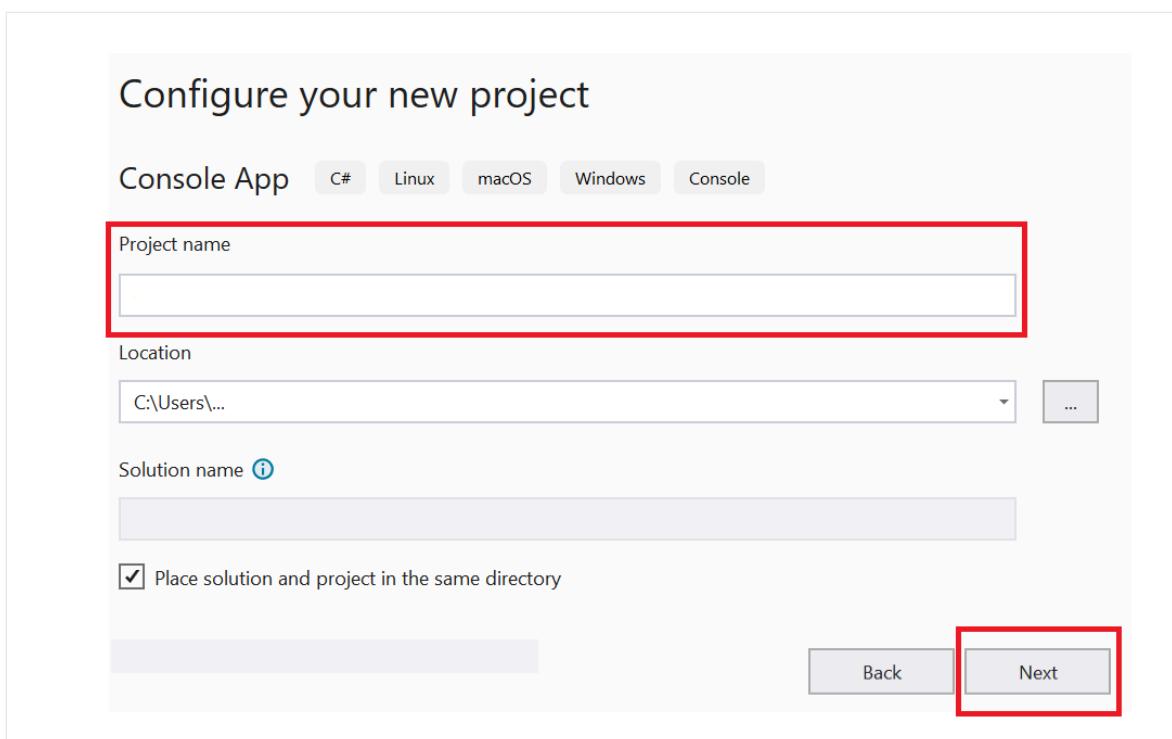
1. Visual Studio를 시작합니다.
2. 시작 페이지에서 새 프로젝트 만들기를 선택합니다.



3. 새 프로젝트 만들기 페이지의 검색 상자에서 콘솔을 입력합니다. 콘솔 애플리케이션 템플릿을 선택하고, 다음을 선택합니다.



4. 새 프로젝트 구성 페이지의 **프로젝트 이름** 아래에 `docIntelligence_app`을 입력합니다. 그런 후 **다음**을 선택합니다.



5. 추가 정보 페이지에서 .NET 8.0(장기 지원)을 선택한 다음, 만들기를 선택합니다.

Additional information

Console App

C#

Linux

macOS

Windows

Console

Framework ⓘ

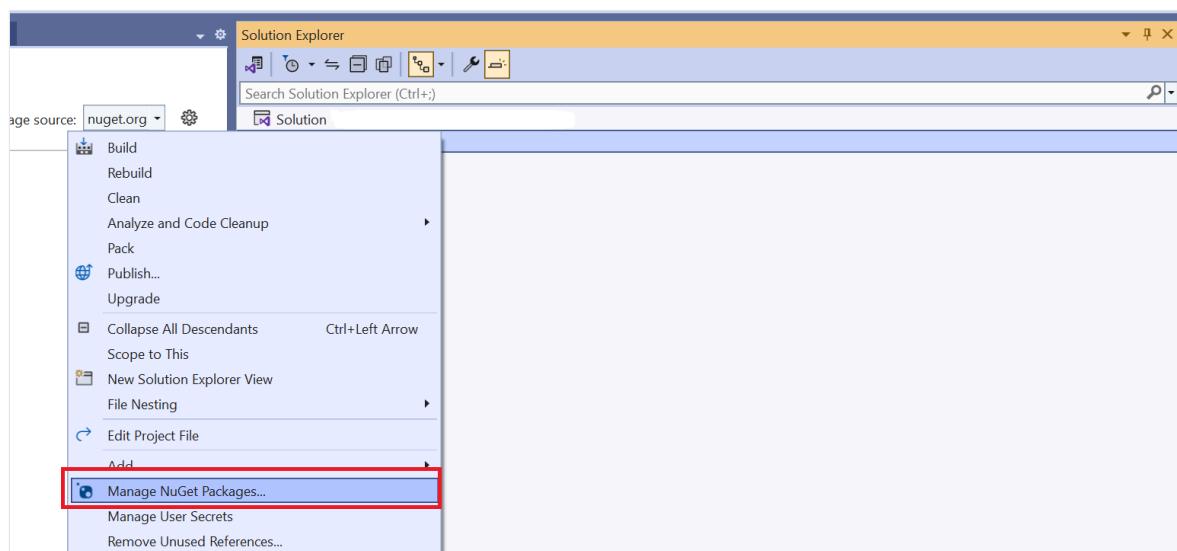
.NET 8.0 (Long Term Support)

Do not use top-level statements ⓘ

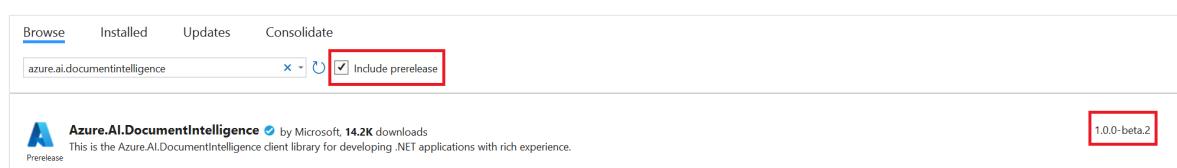
Enable native AOT publish ⓘ

NuGet을 사용하여 클라이언트 라이브러리 설치

1. docIntelligence_app 프로젝트를 마우스 오른쪽 단추로 클릭하고 NuGet 패키지 관리...를 선택합니다.



2. 찾아보기 탭을 선택하고, Azure.AI.FormRecognizer를 입력합니다.



3. 드롭다운 메뉴에서 버전을 선택하고 프로젝트에 패키지를 설치합니다.

애플리케이션 빌드

① 참고

.NET 6부터 `console` 템플릿을 사용하는 새 프로젝트는 이전 버전과 다른 새 프로그램 스타일을 생성합니다. 새 출력에서는 작성해야 하는 코드를 간소화하는 최신 C# 기능을 사용합니다.

최신 버전을 사용하는 경우 `Main` 메서드 본문을 작성하기만 하면 됩니다. 최상위 문, 전역 `using` 지시문 또는 암시적 `using` 지시문을 포함할 필요가 없습니다. 자세한 내용은 [C# 콘솔 앱 템플릿에서 최상위 명령문 생성](#)을 참조하세요.

1. `Program.cs` 파일을 엽니다.
2. `Console.WriteLine("Hello World!")` 줄을 포함하여 기존 코드를 삭제합니다.
3. 다음 코드 샘플 중 하나를 선택하고 애플리케이션의 `Program.cs` 파일에 복사/붙여 넣습니다.
 - `prebuilt-read`
 - `prebuilt-layout`
 - `prebuilt-tax.us.w2`
 - `prebuilt-invoice`
 - `prebuilt-receipt`
 - `prebuilt-idDocument`
4. 애플리케이션에 코드 샘플을 추가한 후에는 프로젝트 이름 옆에 있는 녹색 **시작** 단추를 선택하여 프로그램을 빌드하고 실행하거나 F5 키를 누릅니다.



읽기 모델 사용

```
C#  
  
using Azure;  
using Azure.AI.FormRecognizer.DocumentAnalysis;  
  
//use your `key` and `endpoint` environment variables to create your  
`AzureKeyCredential` and `DocumentAnalysisClient` instances  
string key = Environment.GetEnvironmentVariable("DI_KEY");  
string endpoint = Environment.GetEnvironmentVariable("DI_ENDPOINT");  
AzureKeyCredential credential = new AzureKeyCredential(key);  
DocumentAnalysisClient client = new DocumentAnalysisClient(new  
Uri(endpoint), credential);  
  
//sample document  
Uri fileUri = new Uri("https://raw.githubusercontent.com/Azure-")
```

```
Samples/cognitive-services-REST-api-samples/master/curl/form-
recognizer/rest-api/read.png");

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

foreach (DocumentPage page in result.Pages)
{
    Console.WriteLine($"Document Page {page.PageNumber} has
{page.Lines.Count} line(s), {page.Words.Count} word(s),");
    Console.WriteLine($"and {page.SelectionMarks.Count} selection
mark(s).");

    for (int i = 0; i < page.Lines.Count; i++)
    {
        DocumentLine line = page.Lines[i];
        Console.WriteLine($" Line {i} has content: '{line.Content}'.");

        Console.WriteLine($" Its bounding polygon (points ordered
clockwise):");

        for (int j = 0; j < line.BoundingPolygon.Count; j++)
        {
            Console.WriteLine($" Point {j} => X:
{line.BoundingPolygon[j].X}, Y: {line.BoundingPolygon[j].Y}");
        }
    }
}

foreach (DocumentStyle style in result.Styles)
{
    // Check the style and style confidence to see if text is handwritten.
    // Note that value '0.8' is used as an example.

    bool isHandwritten = style.IsHandwritten.HasValue && style.IsHandwritten
== true;

    if (isHandwritten && style.Confidence > 0.8)
    {
        Console.WriteLine($"Handwritten content found:");

        foreach (DocumentSpan span in style.Spans)
        {
            Console.WriteLine($" Content:
{result.Content.Substring(span.Index, span.Length)}");
        }
    }
}

Console.WriteLine("Detected languages:");

foreach (DocumentLanguage language in result.Languages)
{
```

```
        Console.WriteLine($"  Found language with locale'{language.Locale}' with
confidence {language.Confidence}.");
}
```

GitHub의 Azure 샘플 리포지토리를 방문하여 [read 모델 출력](#)을 확인합니다.

레이아웃 모델 사용

C#

```
using Azure;
using Azure.AI.FormRecognizer.DocumentAnalysis;

//use your `key` and `endpoint` environment variables to create your
`AzureKeyCredential` and `DocumentAnalysisClient` instances
string key = Environment.GetEnvironmentVariable("DI_KEY");
string endpoint = Environment.GetEnvironmentVariable("DI_ENDPOINT");
AzureKeyCredential credential = new AzureKeyCredential(key);
DocumentAnalysisClient client = new DocumentAnalysisClient(new
Uri(endpoint), credential);

// sample document document
Uri fileUri = new Uri ("https://raw.githubusercontent.com/Azure-
Samples/cognitive-services-REST-api-samples/master/curl/form-
recognizer/rest-api/layout.png");

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

AnalyzeResult result = operation.Value;

foreach (DocumentPage page in result.Pages)
{
    Console.WriteLine($"Document Page {page.PageNumber} has
{page.Lines.Count} line(s), {page.Words.Count} word(s),");
    Console.WriteLine($"and {page.SelectionMarks.Count} selection
mark(s).");

    for (int i = 0; i < page.Lines.Count; i++)
    {
        DocumentLine line = page.Lines[i];
        Console.WriteLine($"  Line {i} has content: '{line.Content}'.");

        Console.WriteLine($"      Its bounding polygon (points ordered
clockwise):");

        for (int j = 0; j < line.BoundingPolygon.Count; j++)
        {
            Console.WriteLine($"          Point {j} => X:
{line.BoundingPolygon[j].X}, Y: {line.BoundingPolygon[j].Y}");
        }
    }
}
```

```

        }

    for (int i = 0; i < page.SelectionMarks.Count; i++)
    {
        DocumentSelectionMark selectionMark = page.SelectionMarks[i];

        Console.WriteLine($"  Selection Mark {i} is
{selectionMark.State}.");
        Console.WriteLine($"      Its bounding polygon (points ordered
clockwise):");

        for (int j = 0; j < selectionMark.BoundingPolygon.Count; j++)
        {
            Console.WriteLine($"          Point {j} => X:
{selectionMark.BoundingPolygon[j].X}, Y:
{selectionMark.BoundingPolygon[j].Y}");
        }
    }

Console.WriteLine("Paragraphs:");

foreach (DocumentParagraph paragraph in result.Paragraphs)
{
    Console.WriteLine($"  Paragraph content: {paragraph.Content}");

    if (paragraph.Role != null)
    {
        Console.WriteLine($"      Role: {paragraph.Role}");
    }
}

foreach (DocumentStyle style in result.Styles)
{
    // Check the style and style confidence to see if text is handwritten.
    // Note that value '0.8' is used as an example.

    bool isHandwritten = style.IsHandwritten.HasValue && style.IsHandwritten
== true;

    if (isHandwritten && style.Confidence > 0.8)
    {
        Console.WriteLine($"Handwritten content found:");

        foreach (DocumentSpan span in style.Spans)
        {
            Console.WriteLine($"          Content:
{result.Content.Substring(span.Index, span.Length)}");
        }
    }
}

Console.WriteLine("The following tables were extracted:");

```

```

for (int i = 0; i < result.Tables.Count; i++)
{
    DocumentTable table = result.Tables[i];
    Console.WriteLine($" Table {i} has {table.RowCount} rows and
{table.ColumnCount} columns.");

    foreach (DocumentTableCell cell in table.Cells)
    {
        Console.WriteLine($" Cell ({cell.RowIndex}, {cell.ColumnIndex})
has kind '{cell.Kind}' and content: '{cell.Content}'.");
    }
}

```

GitHub의 Azure 샘플 리포지토리를 방문하여 [레이아웃 모델 출력](#)을 확인합니다.

일반 문서 모델 사용

C#

```

using Azure;
using Azure.AI.FormRecognizer.DocumentAnalysis;

//use your `key` and `endpoint` environment variables to create your
`AzureKeyCredential` and `DocumentAnalysisClient` instances
string key = Environment.GetEnvironmentVariable("DI_KEY");
string endpoint = Environment.GetEnvironmentVariable("DI_ENDPOINT");
AzureKeyCredential credential = new AzureKeyCredential(key);
DocumentAnalysisClient client = new DocumentAnalysisClient(new
Uri(endpoint), credential);

// sample document document
Uri fileUri = new Uri("https://raw.githubusercontent.com/Azure-
Samples/cognitive-services-REST-api-samples/master/curl/form-
recognizer/sample-layout.pdf");

AnalyzeDocumentOperation operation = await
client.AnalyzeDocumentFromUriAsync(WaitUntil.Completed, "prebuilt-document",
fileUri);

AnalyzeResult result = operation.Value;

Console.WriteLine("Detected key-value pairs:");

foreach (DocumentKeyValuePair kvp in result.KeyValuePairs)
{
    if (kvp.Value == null)
    {
        Console.WriteLine($" Found key with no value:
'{kvp.Key.Content}'");
    }
    else

```

```

    {
        Console.WriteLine($"  Found key-value pair: '{kvp.Key.Content}' and
'{kvp.Value.Content}'");
    }
}

foreach (DocumentPage page in result.Pages)
{
    Console.WriteLine($"Document Page {page.PageNumber} has
{page.Lines.Count} line(s), {page.Words.Count} word(s),");
    Console.WriteLine($"and {page.SelectionMarks.Count} selection
mark(s).");

    for (int i = 0; i < page.Lines.Count; i++)
    {
        DocumentLine line = page.Lines[i];
        Console.WriteLine($"  Line {i} has content: '{line.Content}'.");

        Console.WriteLine($"    Its bounding polygon (points ordered
clockwise):");

        for (int j = 0; j < line.BoundingPolygon.Count; j++)
        {
            Console.WriteLine($"      Point {j} => X:
{line.BoundingPolygon[j].X}, Y: {line.BoundingPolygon[j].Y}");
        }
    }

    for (int i = 0; i < page.SelectionMarks.Count; i++)
    {
        DocumentSelectionMark selectionMark = page.SelectionMarks[i];

        Console.WriteLine($"  Selection Mark {i} is
{selectionMark.State}.");
        Console.WriteLine($"    Its bounding polygon (points ordered
clockwise):");

        for (int j = 0; j < selectionMark.BoundingPolygon.Count; j++)
        {
            Console.WriteLine($"      Point {j} => X:
{selectionMark.BoundingPolygon[j].X}, Y:
{selectionMark.BoundingPolygon[j].Y}");
        }
    }
}

foreach (DocumentStyle style in result.Styles)
{
    // Check the style and style confidence to see if text is handwritten.
    // Note that value '0.8' is used as an example.

    bool isHandwritten = style.IsHandwritten.HasValue && style.IsHandwritten
== true;

    if (isHandwritten && style.Confidence > 0.8)
}

```

```

{
    Console.WriteLine($"Handwritten content found:");

        foreach (DocumentSpan span in style.Spans)
    {
        Console.WriteLine($"  Content:
{result.Content.Substring(span.Index, span.Length)}");
    }
}

Console.WriteLine("The following tables were extracted:");

for (int i = 0; i < result.Tables.Count; i++)
{
    DocumentTable table = result.Tables[i];
    Console.WriteLine($"  Table {i} has {table.RowCount} rows and
{table.ColumnCount} columns.");

        foreach (DocumentTableCell cell in table.Cells)
    {
        Console.WriteLine($"    Cell ({cell.RowIndex}, {cell.ColumnIndex})
has kind '{cell.Kind}' and content: '{cell.Content}'.");
    }
}

```

GitHub의 Azure 샘플 리포지토리를 방문하여 [일반 문서 모델 출력](#) 확인합니다.

W-2 세금 모델 사용

C#

```

using Azure;
using Azure.AI.FormRecognizer.DocumentAnalysis;

//use your `key` and `endpoint` environment variables to create your
`AzureKeyCredential` and `DocumentAnalysisClient` instances
string key = Environment.GetEnvironmentVariable("DI_KEY");
string endpoint = Environment.GetEnvironmentVariable("DI_ENDPOINT");
AzureKeyCredential credential = new AzureKeyCredential(key);
DocumentAnalysisClient client = new DocumentAnalysisClient(new
Uri(endpoint), credential);

// sample document document
Uri w2Uri = new Uri("https://raw.githubusercontent.com/Azure-
Samples/cognitive-services-REST-api-samples/master/curl/form-
recognizer/rest-api/w2.png");

AnalyzeDocumentOperation operation = await
client.AnalyzeDocumentFromUriAsync(WaitUntil.Completed, "prebuilt-

```

```
tax.us.w2", w2Uri);

AnalyzeResult result = operation.Value;

for (int i = 0; i < result.Documents.Count; i++)
{
    Console.WriteLine($"Document {i}:");

    AnalyzedDocument document = result.Documents[i];

    if (document.Fields.TryGetValue("AdditionalInfo", out DocumentField? additionalInfoField))
    {
        if (additionalInfoField.FieldType == DocumentFieldType.List)
        {
            foreach (DocumentField infoField in additionalInfoField.Value.AsList())
            {
                Console.WriteLine("AdditionalInfo:");

                if (infoField.FieldType == DocumentFieldType.Dictionary)
                {
                    IReadOnlyDictionary<string, DocumentField> infoFields =
infoField.Value.AsDictionary();

                    if (infoFields.TryGetValue("Amount", out DocumentField? amountField))
                    {
                        if (amountField.FieldType ==
DocumentFieldType.Double)
                        {
                            double amount = amountField.Value.AsDouble();

                            Console.WriteLine($" Amount: '{amount}', with
confidence {amountField.Confidence}");
                        }
                    }

                    if (infoFields.TryGetValue("LetterCode", out
DocumentField? letterCodeField))
                    {
                        if (letterCodeField.FieldType ==
DocumentFieldType.String)
                        {
                            string letterCode =
letterCodeField.ValueAsString();

                            Console.WriteLine($" LetterCode:
'{letterCode}', with confidence {letterCodeField.Confidence}");
                        }
                    }
                }
            }
        }
    }
}
```

```

    if (document.Fields.TryGetValue("AllocatedTips", out DocumentField? allocatedTipsField))
    {
        if (allocatedTipsField.FieldType == DocumentFieldType.Double)
        {
            double allocatedTips = allocatedTipsField.Value.AsDouble();
            Console.WriteLine($"Allocated Tips: '{allocatedTips}', with confidence {allocatedTipsField.Confidence}");
        }
    }

    if (document.Fields.TryGetValue("Employer", out DocumentField? employerField))
    {
        if (employerField.FieldType == DocumentFieldType.Dictionary)
        {
            IReadOnlyDictionary<string, DocumentField> employerFields = employerField.Value.AsDictionary();

            if (employerFields.TryGetValue("Name", out DocumentField? employerNameField))
            {
                if (employerNameField.FieldType == DocumentFieldType.String)
                {
                    string name = employerNameField.ValueAsString();

                    Console.WriteLine($"Employer Name: '{name}', with confidence {employerNameField.Confidence}");
                }
            }

            if (employerFields.TryGetValue("IdNumber", out DocumentField? idNumberField))
            {
                if (idNumberField.FieldType == DocumentFieldType.String)
                {
                    string id = idNumberField.ValueAsString();

                    Console.WriteLine($"Employer ID Number: '{id}', with confidence {idNumberField.Confidence}");
                }
            }

            if (employerFields.TryGetValue("Address", out DocumentField? addressField))
            {
                if (addressField.FieldType == DocumentFieldType.Address)
                {
                    Console.WriteLine($"Employer Address: '{addressField.Content}', with confidence {addressField.Confidence}");
                }
            }
        }
    }
}

```

```
    }  
}
```

GitHub의 Azure 샘플 리포지토리를 방문하여 [W-2 세금 모델 출력](#)을 확인합니다.

청구서 모델 사용

C#

```
using Azure;  
using Azure.AI.FormRecognizer.DocumentAnalysis;  
  
//use your `key` and `endpoint` environment variables to create your  
`AzureKeyCredential` and `DocumentAnalysisClient` instances  
string key = Environment.GetEnvironmentVariable("DI_KEY");  
string endpoint = Environment.GetEnvironmentVariable("DI_ENDPOINT");  
AzureKeyCredential credential = new AzureKeyCredential(key);  
DocumentAnalysisClient client = new DocumentAnalysisClient(new  
Uri(endpoint), credential);  
  
// sample document document  
Uri invoiceUri = new Uri("https://github.com/Azure-Samples/cognitive-  
services-REST-api-samples/raw/master/curl/form-recognizer/rest-  
api/invoice.pdf");  
  
AnalyzeDocumentOperation operation = await  
client.AnalyzeDocumentFromUriAsync(WaitUntil.Completed, "prebuilt-invoice",  
invoiceUri);  
  
AnalyzeResult result = operation.Value;  
  
for (int i = 0; i < result.Documents.Count; i++)  
{  
    Console.WriteLine($"Document {i}:");  
  
    AnalyzedDocument document = result.Documents[i];  
  
    if (document.Fields.TryGetValue("VendorName", out DocumentField  
vendorNameField))  
    {  
        if (vendorNameField.FieldType == DocumentFieldType.String)  
        {  
            string vendorName = vendorNameField.ValueAsString();  
            Console.WriteLine($"Vendor Name: '{vendorName}', with confidence  
{vendorNameField.Confidence}");  
        }  
    }  
  
    if (document.Fields.TryGetValue("CustomerName", out DocumentField  
customerNameField))  
    {  
        if (customerNameField.FieldType == DocumentFieldType.String)
```

```

        {
            string customerName = customerNameField.ValueAsString();
            Console.WriteLine($"Customer Name: '{customerName}', with
confidence {customerNameField.Confidence}");
        }
    }

    if (document.Fields.TryGetValue("Items", out DocumentField itemsField))
    {
        if (itemsField.FieldType == DocumentFieldType.List)
        {
            foreach (DocumentField itemField in itemsField.Value.AsList())
            {
                Console.WriteLine("Item:");

                if (itemField.FieldType == DocumentFieldType.Dictionary)
                {
                    IReadOnlyDictionary<string, DocumentField> itemFields =
itemField.Value.AsDictionary();

                    if (itemFields.TryGetValue("Description", out
DocumentField itemDescriptionField))
                    {
                        if (itemDescriptionField.FieldType ==
DocumentFieldType.String)
                        {
                            string itemDescription =
itemDescriptionField.ValueAsString();

                            Console.WriteLine($" Description:
'{itemDescription}', with confidence {itemDescriptionField.Confidence}");
                        }
                    }

                    if (itemFields.TryGetValue("Amount", out DocumentField
itemAmountField))
                    {
                        if (itemAmountField.FieldType ==
DocumentFieldType.Currency)
                        {
                            CurrencyValue itemAmount =
itemAmountField.Value.AsCurrency();

                            Console.WriteLine($" Amount:
'{itemAmount.Symbol}{itemAmount.Amount}', with confidence
{itemAmountField.Confidence}");
                        }
                    }
                }
            }
        }
    }

    if (document.Fields.TryGetValue("SubTotal", out DocumentField
subTotalField))

```

```

    {
        if (subTotalField.FieldType == DocumentFieldType.Currency)
        {
            CurrencyValue subTotal = subTotalField.Value.AsCurrency();
            Console.WriteLine($"Sub Total: '{subTotal.Symbol}
{subTotal.Amount}', with confidence {subTotalField.Confidence}");
        }
    }

    if (document.Fields.TryGetValue("TotalTax", out DocumentField
totalTaxField))
    {
        if (totalTaxField.FieldType == DocumentFieldType.Currency)
        {
            CurrencyValue totalTax = totalTaxField.Value.AsCurrency();
            Console.WriteLine($"Total Tax: '{totalTax.Symbol}
{totalTax.Amount}', with confidence {totalTaxField.Confidence}");
        }
    }

    if (document.Fields.TryGetValue("InvoiceTotal", out DocumentField
invoiceTotalField))
    {
        if (invoiceTotalField.FieldType == DocumentFieldType.Currency)
        {
            CurrencyValue invoiceTotal =
invoiceTotalField.Value.AsCurrency();
            Console.WriteLine($"Invoice Total: '{invoiceTotal.Symbol}
{invoiceTotal.Amount}', with confidence {invoiceTotalField.Confidence}");
        }
    }
}

```

GitHub의 Azure 샘플 리포지토리를 방문하여 [청구서 모델 출력](#)을 확인합니다.

영수증 모델 사용

C#

```

using Azure;
using Azure.AI.FormRecognizer.DocumentAnalysis;

//use your `key` and `endpoint` environment variables to create your
`AzureKeyCredential` and `DocumentAnalysisClient` instances
string key = Environment.GetEnvironmentVariable("DI_KEY");
string endpoint = Environment.GetEnvironmentVariable("DI_ENDPOINT");
AzureKeyCredential credential = new AzureKeyCredential(key);
DocumentAnalysisClient client = new DocumentAnalysisClient(new
Uri(endpoint), credential);

// sample document document

```

```
Uri receiptUri = new Uri("https://raw.githubusercontent.com/Azure-
Samples/cognitive-services-REST-api-samples/master/curl/form-
recognizer/rest-api/receipt.png");

AnalyzeDocumentOperation operation = await
client.AnalyzeDocumentFromUriAsync(WaitUntil.Completed, "prebuilt-receipt",
receiptUri);

AnalyzeResult receipts = operation.Value;

foreach (AnalyzedDocument receipt in receipts.Documents)
{
    if (receipt.Fields.TryGetValue("MerchantName", out DocumentField
merchantNameField))
    {
        if (merchantNameField.FieldType == DocumentFieldType.String)
        {
            string merchantName = merchantNameField.ValueAsString();

            Console.WriteLine($"Merchant Name: '{merchantName}', with
confidence {merchantNameField.Confidence}");
        }
    }

    if (receipt.Fields.TryGetValue("TransactionDate", out DocumentField
transactionDateField))
    {
        if (transactionDateField.FieldType == DocumentFieldType.Date)
        {
            DateTimeOffset transactionDate =
transactionDateField.Value.AsDate();

            Console.WriteLine($"Transaction Date: '{transactionDate}', with
confidence {transactionDateField.Confidence}");
        }
    }

    if (receipt.Fields.TryGetValue("Items", out DocumentField itemsField))
    {
        if (itemsField.FieldType == DocumentFieldType.List)
        {
            foreach (DocumentField itemField in itemsField.Value.AsList())
            {
                Console.WriteLine("Item:");

                if (itemField.FieldType == DocumentFieldType.Dictionary)
                {
                    IReadOnlyDictionary<string, DocumentField> itemFields =
itemField.Value.AsDictionary();

                    if (itemFields.TryGetValue("Description", out
DocumentField itemDescriptionField))
                    {
                        if (itemDescriptionField.FieldType ==
DocumentFieldType.String)
```

```

        {
            string itemDescription =
itemDescriptionField.Value.AsString();

                Console.WriteLine($" Description:
'{itemDescription}', with confidence {itemDescriptionField.Confidence}");
        }
    }

        if (itemFields.TryGetValue("TotalPrice", out
DocumentField itemTotalPriceField))
{
    if (itemTotalPriceField.FieldType ==
DocumentFieldType.Double)
    {
        double itemTotalPrice =
itemTotalPriceField.Value.AsDouble();

                Console.WriteLine($" Total Price:
'{itemTotalPrice}', with confidence {itemTotalPriceField.Confidence}");
    }
}
}

}

if (receipt.Fields.TryGetValue("Total", out DocumentField totalField))
{
    if (totalField.FieldType == DocumentFieldType.Double)
    {
        double total = totalField.Value.AsDouble();

                Console.WriteLine($"Total: '{total}', with confidence
'{totalField.Confidence}'");
    }
}
}

```

GitHub의 Azure 샘플 리포지토리를 방문하여 [영수증 모델 출력](#)을 확인합니다.

ID 문서 모델

C#

```

using Azure;
using Azure.AI.FormRecognizer.DocumentAnalysis;

//use your `key` and `endpoint` environment variables to create your
`AzureKeyCredential` and `DocumentAnalysisClient` instances

```

```
string key = Environment.GetEnvironmentVariable("DI_KEY");
string endpoint = Environment.GetEnvironmentVariable("DI_ENDPOINT");
AzureKeyCredential credential = new AzureKeyCredential(key);
DocumentAnalysisClient client = new DocumentAnalysisClient(new
Uri(endpoint), credential);

// sample document document

Uri idDocumentUri = new Uri("https://raw.githubusercontent.com/Azure-
Samples/cognitive-services-REST-api-samples/master/curl/form-
recognizer/rest-api/identity_documents.png");

AnalyzeDocumentOperation operation = await
client.AnalyzeDocumentFromUriAsync(WaitUntil.Completed, "prebuilt-
idDocument", idDocumentUri);

AnalyzeResult identityDocuments = operation.Value;

AnalyzedDocument identityDocument = identityDocuments.Documents.Single();

if (identityDocument.Fields.TryGetValue("Address", out DocumentField
addressField))
{
    if (addressField.FieldType == DocumentFieldType.String)
    {
        string address = addressField.Value.AsString();
        Console.WriteLine($"Address: '{address}', with confidence
{addressField.Confidence}");
    }
}

if (identityDocument.Fields.TryGetValue("CountryRegion", out DocumentField
countryRegionField))
{
    if (countryRegionField.FieldType == DocumentFieldType.CountryRegion)
    {
        string countryRegion = countryRegionField.Value.AsCountryRegion();
        Console.WriteLine($"CountryRegion: '{countryRegion}', with
confidence {countryRegionField.Confidence}");
    }
}

if (identityDocument.Fields.TryGetValue("DateOfBirth", out DocumentField
dateOfBirthField))
{
    if (dateOfBirthField.FieldType == DocumentFieldType.Date)
    {
        DateTimeOffset dateOfBirth = dateOfBirthField.Value.AsDate();
        Console.WriteLine($"Date Of Birth: '{dateOfBirth}', with confidence
{dateOfBirthField.Confidence}");
    }
}

if (identityDocument.Fields.TryGetValue("DateOfExpiration", out
DocumentField dateOfExpirationField))
```

```
{  
    if (dateOfExpirationField.FieldType == DocumentFieldType.Date)  
    {  
        DateTimeOffset dateOfExpiration =  
dateOfExpirationField.Value.AsDate();  
        Console.WriteLine($"Date Of Expiration: '{dateOfExpiration}', with  
confidence {dateOfExpirationField.Confidence}");  
    }  
}  
  
if (identityDocument.Fields.TryGetValue("DocumentNumber", out DocumentField  
documentNumberField))  
{  
    if (documentNumberField.FieldType == DocumentFieldType.String)  
    {  
        string documentNumber = documentNumberField.ValueAsString();  
        Console.WriteLine($"Document Number: '{documentNumber}', with  
confidence {documentNumberField.Confidence}");  
    }  
}  
  
if (identityDocument.Fields.TryGetValue("FirstName", out DocumentField  
firstNameField))  
{  
    if (firstNameField.FieldType == DocumentFieldType.String)  
    {  
        string firstName = firstNameField.ValueAsString();  
        Console.WriteLine($"First Name: '{firstName}', with confidence  
{firstNameField.Confidence}");  
    }  
}  
  
if (identityDocument.Fields.TryGetValue("LastName", out DocumentField  
lastNameField))  
{  
    if (lastNameField.FieldType == DocumentFieldType.String)  
    {  
        string lastName = lastNameField.ValueAsString();  
        Console.WriteLine($"Last Name: '{lastName}', with confidence  
{lastNameField.Confidence}");  
    }  
}  
  
if (identityDocument.Fields.TryGetValue("Region", out DocumentField  
regionfield))  
{  
    if (regionfield.FieldType == DocumentFieldType.String)  
    {  
        string region = regionfield.ValueAsString();  
        Console.WriteLine($"Region: '{region}', with confidence  
{regionfield.Confidence}");  
    }  
}  
  
if (identityDocument.Fields.TryGetValue("Sex", out DocumentField sexfield))
```

```

{
    if (sexfield.FieldType == DocumentFieldType.String)
    {
        string sex = sexfield.ValueAsString();
        Console.WriteLine($"Sex: '{sex}', with confidence
{sexfield.Confidence}");
    }
}

```

GitHub의 Azure 샘플 리포지토리를 방문하여 [id-문서 모델 출력](#)을 확인합니다.

명함 모델 사용

C#

```

using Azure;
using Azure.AI.FormRecognizer.DocumentAnalysis;

//use your `key` and `endpoint` environment variables to create your
//`AzureKeyCredential` and `DocumentAnalysisClient` instances
string key = Environment.GetEnvironmentVariable("DI_KEY");
string endpoint = Environment.GetEnvironmentVariable("DI_ENDPOINT");
AzureKeyCredential credential = new AzureKeyCredential(key);
DocumentAnalysisClient client = new DocumentAnalysisClient(new
Uri(endpoint), credential);

// sample document document
Uri businessCardUri = new Uri("https://raw.githubusercontent.com/Azure-
Samples/cognitive-services-REST-api-samples/master/curl/form-
recognizer/business-card-english.jpg");

AnalyzeDocumentOperation operation = await
client.AnalyzeDocumentFromUriAsync(WaitUntil.Completed, "prebuilt-
businessCard", businessCardUri);

AnalyzeResult businessCards = operation.Value;

foreach (AnalyzedDocument businessCard in businessCards.Documents)
{
    if (businessCard.Fields.TryGetValue("ContactNames", out DocumentField
ContactNamesField))
    {
        if (ContactNamesField.FieldType == DocumentFieldType.List)
        {
            foreach (DocumentField contactNameField in
ContactNamesField.Value.AsList())
            {
                Console.WriteLine("Contact Name: ");

                if (contactNameField.FieldType ==
DocumentFieldType.Dictionary)

```

```

    {
        IReadOnlyDictionary<string, DocumentField>
contactNameFields = contactNameField.Value.AsDictionary();

        if (contactNameFields.TryGetValue("FirstName", out
DocumentField firstNameField))
        {
            if (firstNameField.FieldType ==
DocumentFieldType.String)
            {
                string firstName =
firstNameField.ValueAsString();

                Console.WriteLine($" First Name: '{firstName}',
with confidence {firstNameField.Confidence}");
            }
        }

        if (contactNameFields.TryGetValue("LastName", out
DocumentField lastNameField))
        {
            if (lastNameField.FieldType ==
DocumentFieldType.String)
            {
                string lastName =
lastNameField.ValueAsString();

                Console.WriteLine($" Last Name: '{lastName}',
with confidence {lastNameField.Confidence}");
            }
        }
    }

    if (businessCard.Fields.TryGetValue("JobTitles", out DocumentField
jobTitlesFields))
    {
        if (jobTitlesFields.FieldType == DocumentFieldType.List)
        {
            foreach (DocumentField jobTitleField in
jobTitlesFields.Value.AsList())
            {
                if (jobTitleField.FieldType == DocumentFieldType.String)
                {
                    string jobTitle = jobTitleField.ValueAsString();

                    Console.WriteLine($"Job Title: '{jobTitle}', with
confidence {jobTitleField.Confidence}");
                }
            }
        }
    }
}

```

```

        if (businessCard.Fields.TryGetValue("Departments", out DocumentField
departmentFields))
    {
        if (departmentFields.FieldType == DocumentFieldType.List)
        {
            foreach (DocumentField departmentField in
departmentFields.Value.AsList())
            {
                if (departmentField.FieldType == DocumentFieldType.String)
                {
                    string department = departmentField.ValueAsString();

                    Console.WriteLine($"Department: '{department}', with
confidence {departmentField.Confidence}");
                }
            }
        }
    }

    if (businessCard.Fields.TryGetValue("Emails", out DocumentField
emailFields))
    {
        if (emailFields.FieldType == DocumentFieldType.List)
        {
            foreach (DocumentField emailField in emailFields.Value.AsList())
            {
                if (emailField.FieldType == DocumentFieldType.String)
                {
                    string email = emailField.ValueAsString();

                    Console.WriteLine($"Email: '{email}', with confidence
{emailField.Confidence}");
                }
            }
        }
    }

    if (businessCard.Fields.TryGetValue("Websites", out DocumentField
websiteFields))
    {
        if (websiteFields.FieldType == DocumentFieldType.List)
        {
            foreach (DocumentField websiteField in
websiteFields.Value.AsList())
            {
                if (websiteField.FieldType == DocumentFieldType.String)
                {
                    string website = websiteField.ValueAsString();

                    Console.WriteLine($"Website: '{website}', with
confidence {websiteField.Confidence}");
                }
            }
        }
    }
}

```

```

    if (businessCard.Fields.TryGetValue("MobilePhones", out DocumentField
mobilePhonesFields))
    {
        if (mobilePhonesFields.FieldType == DocumentFieldType.List)
        {
            foreach (DocumentField mobilePhoneField in
mobilePhonesFields.Value.AsList())
            {
                if (mobilePhoneField.FieldType ==
DocumentFieldType.PhoneNumber)
                {
                    string mobilePhone =
mobilePhoneField.Value.AsPhoneNumber();

                    Console.WriteLine($"Mobile phone number:
'{mobilePhone}', with confidence {mobilePhoneField.Confidence}");
                }
            }
        }
    }

    if (businessCard.Fields.TryGetValue("WorkPhones", out DocumentField
workPhonesFields))
    {
        if (workPhonesFields.FieldType == DocumentFieldType.List)
        {
            foreach (DocumentField workPhoneField in
workPhonesFields.Value.AsList())
            {
                if (workPhoneField.FieldType ==
DocumentFieldType.PhoneNumber)
                {
                    string workPhone = workPhoneField.Value.AsPhoneNumber();

                    Console.WriteLine($"Work phone number: '{workPhone}', with
confidence {workPhoneField.Confidence}");
                }
            }
        }
    }

    if (businessCard.Fields.TryGetValue("Faxes", out DocumentField
faxesFields))
    {
        if (faxesFields.FieldType == DocumentFieldType.List)
        {
            foreach (DocumentField faxField in faxesFields.Value.AsList())
            {
                if (faxField.FieldType == DocumentFieldType.PhoneNumber)
                {
                    string fax = faxField.Value.AsPhoneNumber();

                    Console.WriteLine($"Fax phone number: '{fax}', with
confidence {faxField.Confidence}");
                }
            }
        }
    }
}

```

```

        }
    }
}

if (businessCard.Fields.TryGetValue("Addresses", out DocumentField
addressesFields))
{
    if (addressesFields.FieldType == DocumentFieldType.List)
    {
        foreach (DocumentField addressField in
addressesFields.Value.AsList())
        {
            if (addressField.FieldType == DocumentFieldType.String)
            {
                string address = addressField.ValueAsString();

                Console.WriteLine($"Address: '{address}', with
confidence {addressField.Confidence}");
            }
        }
    }
}

if (businessCard.Fields.TryGetValue("CompanyNames", out DocumentField
companyNamesFields))
{
    if (companyNamesFields.FieldType == DocumentFieldType.List)
    {
        foreach (DocumentField companyNameField in
companyNamesFields.Value.AsList())
        {
            if (companyNameField.FieldType == DocumentFieldType.String)
            {
                string companyName = companyNameField.ValueAsString();

                Console.WriteLine($"Company name: '{companyName}', with
confidence {companyNameField.Confidence}");
            }
        }
    }
}
}

```

GitHub의 Azure 샘플 리포지토리를 방문하여 [명함 모델 출력](#)을 확인합니다.

다음 단계

축하합니다! 문서 인텔리전스 모델을 사용하여 다양한 문서를 다양한 방식으로 분석하는 방법을 배웠습니다. 다음으로 문서 인텔리전스 Studio 및 참조 문서를 살펴보세요.

[문서 인텔리전스 스튜디오 사용해 보기](#)

[문서 인텔리전스 REST API 살펴보기](#)

문서 인텔리전스 리소스 만들기

아티클 • 2023. 11. 15.

이 콘텐츠는 : v4.0(미리 보기) v3.1(GA) v3.0(GA) v2.1(GA)에 적용됩니다.

Azure AI 문서 인텔리전스는 기계 학습 모델을 사용하여 문서에서 키-값 쌍, 텍스트 및 테이블을 추출하는 클라우드 기반 [Azure AI 서비스](#)입니다. 이 문서에서는 Azure portal에서 문서 인텔리전스 리소스를 만드는 방법을 알아봅니다.

Azure Portal을 방문합니다.

Azure Portal은 Azure 서비스를 만들고 관리하는 데 사용할 수 있는 단일 플랫폼입니다.

시작하기:

1. [Azure Portal](#)에 로그인합니다.
2. Azure 홈페이지에서 **리소스 만들기**를 선택합니다.
3. 검색창에서 **문서 인텔리전스**를 검색하고 선택합니다.
4. **만들기** 단추를 선택합니다.

리소스 만들기

1. 다음으로, 다음 값으로 **문서 인텔리전스 만들기** 필드를 입력합니다.
 - **구독**. 현재 구독을 선택합니다.
 - **리소스 그룹**. 리소스가 포함된 [Azure 리소스 그룹](#)입니다. 새 그룹을 만들거나 기존 그룹에 추가할 수 있습니다.
 - **지역**. 지역을 선택합니다.
 - **이름**. 리소스 이름을 입력합니다. 설명이 포함된 이름(예: *YourNameFormRecognizer*)을 사용하는 것이 좋습니다.
 - **가격 책정 계층**. 리소스 비용은 선택한 가격 책정 계층과 사용량에 따라 달라집니다. 자세한 내용은 [가격 책정 정보](#)를 참조하세요. 체험 가격 책정 계층(F0)을 통해 서비스를 사용해 보고, 나중에 프로덕션을 위한 유료 계층으로 업그레이드할 수 있습니다.
2. **검토 + 생성**을 선택합니다.

Create Form Recognizer ...

[Basics](#) [Network](#) [Identity](#) [Tags](#) [Review + create](#)

Accelerate your business processes by automating information extraction. Form Recognizer applies advanced machine learning to accurately extract text, key/value pairs, and tables from documents. With just a few samples, Form Recognizer tailors its understanding to your documents, both on-premises and in the cloud. Turn forms into usable data at a fraction of the time and cost, so you can focus more time acting on the information rather than compiling it. [Learn more.](#)

Project Details

Subscription * ⓘ

your-subscription

Resource group * ⓘ

your-resource-group

[Create new](#)

Instance Details

Region ⓘ

West US

Name * ⓘ

your-form-recognizer-resource

Pricing tier * ⓘ

Free F0 (500 Pages per month, 20 Calls per minute for recognizer AP...)

[View full pricing details](#)[Review + create](#)

< Previous

Next : Network >

3. Azure는 빠른 유효성 검사를 실행합니다. 몇 초 후에 **유효성 검사 통과**라는 녹색 배너가 표시됩니다.
4. 유효성 검사 배너가 나타나면 왼쪽 아래 모서리에서 **만들기** 단추를 선택합니다.
5. 만들기를 선택하면 **배포 진행 중**이라는 새 페이지로 리디렉션됩니다. 몇 초 후에 **배포가 완료됨**이라는 메시지가 표시됩니다.

엔드포인트 URL 및 키 가져오기

1. **배포가 완료됨** 메시지가 표시되면 **리소스로 이동** 단추를 선택합니다.
2. 문서 인텔리전스 리소스에서 키와 엔드포인트 값을 복사하여 Microsoft 메모장과 같은 편리한 위치에 붙여넣습니다. 애플리케이션을 문서 인텔리전스 API에 연결하려면 키와 엔드포인트 값이 필요합니다.
3. 개요 페이지에 키와 엔드포인트가 표시되지 않는 경우 왼쪽 탐색 모음에서 **키 및 엔드포인트** 단추를 선택하고 검색할 수 있습니다.

Home > Contoso-DI

Contoso-DI | Keys and Endpoint

Search Regenerate Key1 Regenerate Key2

Overview Activity log Access control (IAM) Tags Diagnose and solve problems Resource Management Keys and Endpoint

Encryption Pricing tier Networking Identity Cost analysis Properties Locks Monitoring Automation Help

Show Keys

KEY 1

KEY 2

Location/Region westus2

Endpoint https://contoso-di.cognitiveservices.azure.com/

정말 간단하죠. 이제 Azure AI 문서 인텔리전스를 사용하여 데이터 추출 자동화를 시작할 준비가 되었습니다.

다음 단계

- 문서 인텔리전스 서비스의 기능을 시각적으로 탐색, 이해 및 애플리케이션에 통합하기 위한 온라인 도구인 [문서 인텔리전스 Studio](#)를 사용해 보세요.
- 문서 인텔리전스 빠른 시작을 완료하고 선택한 개발 언어로 문서 처리 앱 만들기를 시작하세요.
 - [C#](#)
 - [Python](#)
 - [Java](#)
 - [JavaScript](#)

사용량 확인 및 비용 예상

아티클 • 2024. 02. 29.

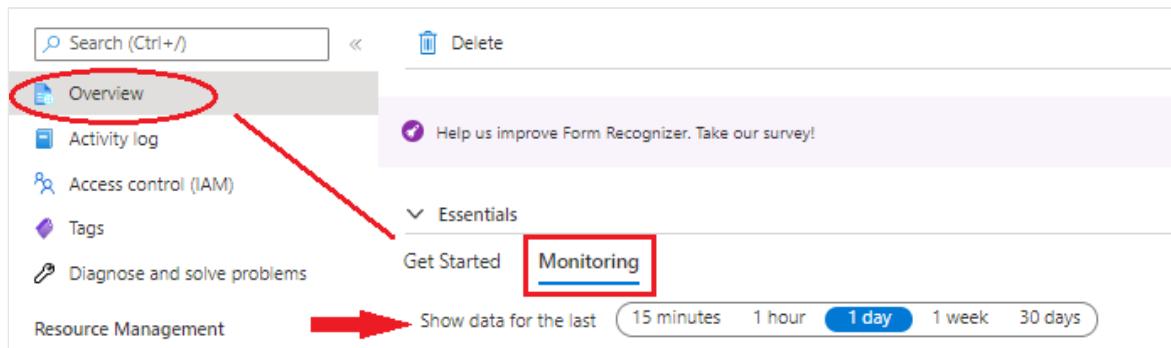
이 콘텐츠의 적용 대상은 다음과 같습니다. ✓ v4.0(미리 보기) ✓ v3.1(GA) ✓ v3.0(GA)
✓ v2.1(GA)

이 가이드에서는 Azure Portal에서 메트릭 대시보드를 사용하여 Azure AI 문서 인텔리전스에서 처리한 페이지 수를 확인하는 방법을 알아봅니다. Azure 가격 책정 계산기를 사용하여 해당 페이지를 처리하는 비용을 예상하는 방법도 알아봅니다.

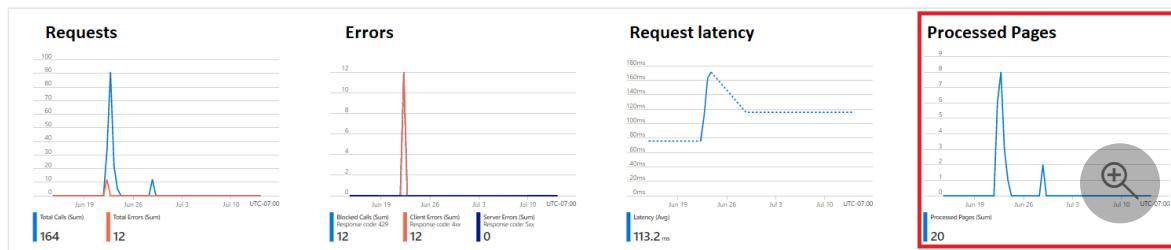
처리된 페이지 수 확인

먼저 지정된 기간 동안 페이지 처리 데이터를 살펴보겠습니다.

1. [Azure Portal](#)에 로그인합니다.
2. 문서 인텔리전스 리소스로 이동합니다.
3. **개요** 페이지에서 페이지 중앙 근처에 있는 **모니터링** 탭을 선택합니다.



4. 시간 범위를 선택하면 처리된 페이지 차트가 표시됩니다.



분석된 페이지 검사

이제 각 모델의 분석된 페이지를 자세히 살펴보겠습니다.

1. 모니터링 섹션 아래의 왼쪽 탐색 메뉴에서 **메트릭**을 선택합니다.

Monitoring

Alerts

Metrics

Diagnostic settings

Logs

2. 메트릭 페이지에서 **메트릭 추가**를 선택합니다.

3. 메트릭 드롭다운 메뉴를 선택하고 사용에서 처리된 페이지를 선택합니다.

The screenshot shows the 'Metrics' blade in the Azure portal. At the top, there are buttons for 'Add metric', 'Add filter', and 'Apply splitting'. Below these are sections for 'Scope' (set to 'form-recognizer') and 'Metric Namespace' (set to 'Cognitive Service stand...'). A large list of metrics is displayed under the 'Metric' heading, with 'Processed Pages' selected. The 'Aggregation' dropdown is set to 'Sum'. On the right, there's a section titled 'USAGE' with 'Processed Pages' listed. The chart area shows a line graph for 'Processed Pages' over time.

4. 오른쪽 상단에서 시간 범위를 구성하고 적용 단추를 선택합니다.

The screenshot shows the 'Metrics' blade with a line chart for 'Processed Pages'. On the right, a modal dialog is open for setting the time range and granularity. The 'Time range' section shows 'Local Time: Last 4 hours (Automatic - 1 minute)' selected. The 'Time granularity' section has 'Automatic' selected. The 'Apply' button at the bottom of the dialog is highlighted with a red box.

5. 분할 적용을 선택합니다.

The screenshot shows the 'Metrics' blade with a line chart for 'Processed Pages'. At the top, there are buttons for 'Add metric', 'Add filter', and 'Apply splitting'. The 'Apply splitting' button is highlighted with a red box. Below the buttons, the metric configuration is shown: 'Scope' is 'form-recognizer', 'Metric Namespace' is 'Cognitive Service stand...', and the metric name is 'Processed Pages, Sum'. The chart area shows a line graph for 'Processed Pages' over time.

6. 값 드롭다운 메뉴에서 FeatureName을 선택합니다.

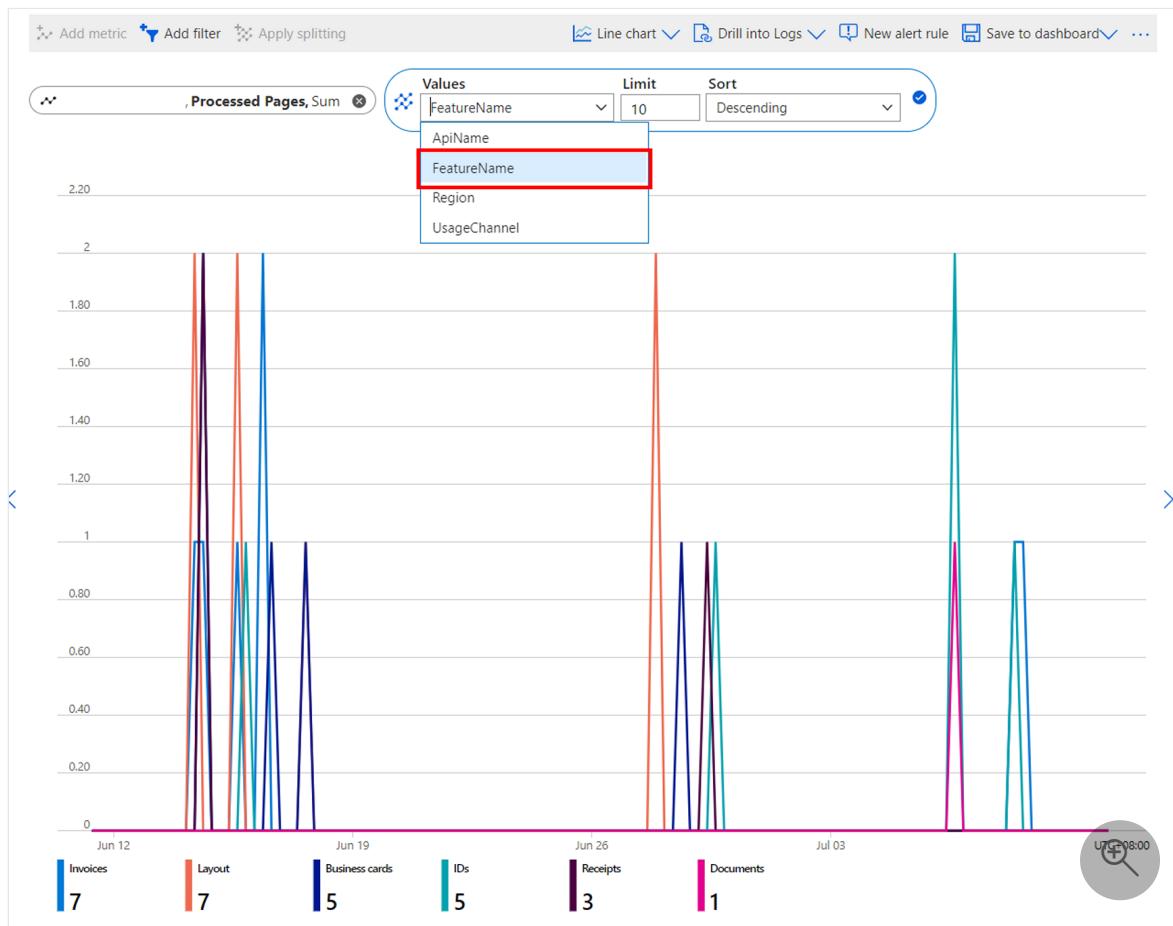
Add metric Add filter Apply splitting

form-recognizer, Processed Pages, Sum

Values Limit
Select value(s) 10

ApiName
FeatureName (highlighted)
Region
UsageChannel

7. 각 모델별로 분석된 페이지 분석이 표시됩니다.



가격 견적

이제 페이지에서 포털의 데이터를 처리했으므로 Azure 가격 책정 계산기를 사용하여 비용을 예상할 수 있습니다.

1. Azure Portal에 사용하는 것과 동일한 자격 증명으로 [Azure 가격 책정 계산기](#)에 로그인합니다.

Ctrl + 마우스 오른쪽 단추를 눌러 새 탭에서 엽니다!

2. 제품 검색 검색 상자에서 Azure AI 문서 인텔리전스를 검색합니다.

3. Azure AI 문서 인텔리전스를 선택하면 페이지에 추가된 것을 볼 수 있습니다.
4. 예상 금액에서 문서 인텔리전스 리소스에 대한 관련 지역, 결제 옵션 및 인스턴스를 선택합니다. 자세한 내용은 [Azure AI 문서 인텔리전스 가격 책정 옵션](#)을 확인하세요.
5. Azure Portal 메트릭 대시보드에서 처리된 페이지 수를 입력합니다. 해당 데이터는 위의 [처리된 페이지 수 확인](#) 또는 [분석된 페이지 조사](#) 섹션의 단계에서 찾을 수 있습니다.
6. 예상 가격은 등호(=) 뒤 오른쪽에 있습니다.

The screenshot shows the Azure Form Recognizer pricing calculator interface. At the top, there are dropdown menus for REGION (West US), PAYMENT OPTION (Pay as you go), and INSTANCE (S0). Below this, there are two sections: 'Custom' and 'Pre-built (S1)'. Both sections show a formula: <number of pages> Pages * \$ <price> Per 1,000 pages = \$ <subtotal>. The 'Pre-built (S1)' section includes a note: 'The prebuilt SKU includes the following Form Recognizer APIs: Layout, Receipt, Business Card, ID, invoice, Document, and W-2.' At the bottom, it shows Upfront cost (\$0.00), Monthly cost (\$0.00), and a total of \$ <total>.

모두 끝났습니다. 이제 문서 인텔리전스를 사용하여 처리한 페이지 수와 비용을 예상하는 방법을 찾을 수 있는 위치를 알게 되었습니다.

다음 단계

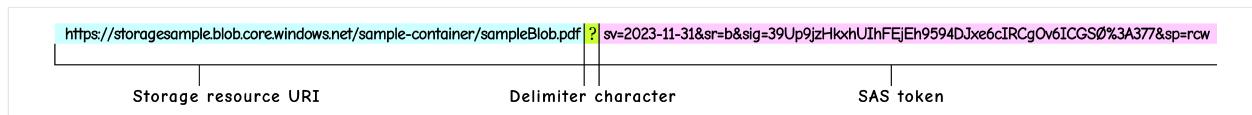
[문서 인텔리전스 서비스 할당량 및 한도에 대한 자세한 정보](#)

스토리지 컨테이너에 대한 SAS 토큰 만들기

아티클 • 2023. 11. 15.

이 콘텐츠는 : ✓ v4.0(미리 보기) ✓ v3.1(GA) ✓ v3.0(GA) ✓ v2.1(GA)에 적용됩니다.

이 문서에서는 Azure Portal 또는 Azure Storage Explorer를 사용하여 사용자 위임, SAS(공유 액세스 서명) 토큰을 만드는 방법을 알아봅니다. 사용자 위임 SAS 토큰은 Microsoft Entra 자격 증명으로 보호됩니다. SAS 토큰은 Azure 스토리지 계정의 리소스에 대한 안전한 위임된 액세스를 제공합니다.



높은 수준에서 SAS 토큰의 작동 방식은 다음과 같습니다.

- 애플리케이션은 REST API 요청의 일부로 SAS 토큰을 Azure Storage에 제출합니다.
- 스토리지 서비스에서 SAS가 유효하다고 확인하면 요청이 승인됩니다.
- SAS 토큰이 유효하지 않은 것으로 간주되면 요청이 거부되고 오류 코드 403(사용할 수 없음)이 반환됩니다.

Azure Blob Storage는 세 가지 종류의 리소스를 제공합니다.

- 스토리지 계정**은 데이터에 대해 Azure에서 고유의 네임스페이스를 제공합니다.
- 데이터 스토리지 컨테이너**는 스토리지 계정에 있으며 Blob 집합을 구성합니다.
- Blob**은 컨테이너에 있으며 텍스트 및 파일, 텍스트 및 이미지와 같은 이진 데이터를 저장합니다.

SAS 토큰을 사용하는 경우

- 사용자 지정 모델 학습.** 어셈블된 학습 문서 집합을 *Azure Blob Storage 컨테이너에 업로드해야 합니다.* SAS 토큰을 사용하여 학습 문서에 대한 액세스 권한을 부여하도록 선택할 수 있습니다.
- 공용 액세스 권한이 있는 스토리지 컨테이너 사용** SAS 토큰을 사용하여 공용 읽기 액세스 권한이 있는 스토리지 리소스에 대한 제한된 액세스 권한을 부여하도록 선택할 수 있습니다.

ⓘ 중요

- Azure Storage 계정이 가상 네트워크 또는 방화벽으로 보호되는 경우 SAS 토큰으로 액세스 권한을 부여할 수 없습니다. [관리 ID](#)를 사용하여 스토리지 리소스에 대한 액세스 권한을 부여해야 합니다.
- [관리 ID](#)는 프라이빗 및 퍼블릭 액세스가 가능한 Azure Blob Storage 계정을 둘 다 지원합니다.
- SAS 토큰은 스토리지 리소스에 대한 권한을 부여하며 계정 키와 동일한 방식으로 보호되어야 합니다.
- SAS 토큰을 사용하는 작업은 HTTPS 연결을 통해서만 수행되어야 하며 SAS URI는 HTTPS와 같은 보안 연결에서만 배포되어야 합니다.

필수 조건

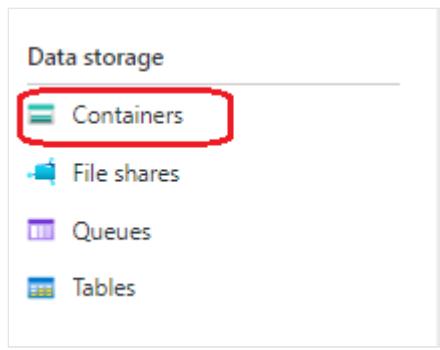
시작하려면 다음이 필요합니다.

- 활성 [Azure 계정](#). 계정이 없는 경우 [무료 계정](#)에 만들 수 있습니다.
- 문서 [인텔리전스](#) 또는 [다중 서비스](#) 리소스입니다.
- **표준 성능**[Azure Blob Storage 계정](#). 스토리지 계정 내에서 Blob 데이터를 저장하고 구성하는 컨테이너를 만들어야 합니다. 스토리지 컨테이너를 사용하여 Azure 스토리지 계정을 만드는 방법을 모르는 경우 다음 빠른 시작을 따릅니다.
 - [스토리지 계정 만들기](#) 스토리지 계정을 만들 때 **인스턴스 세부 사항**>**성능** 필드에서 **표준** 성능을 선택합니다.
 - [컨테이너를 만듭니다](#). 컨테이너를 만드는 경우 새 컨테이너 창에서 **퍼블릭 액세스 수준을 컨테이너**(컨테이너 및 Blob에 대한 익명 읽기 액세스)로 설정합니다.

문서 업로드

1. [Azure Portal](#)에 로그인합니다.

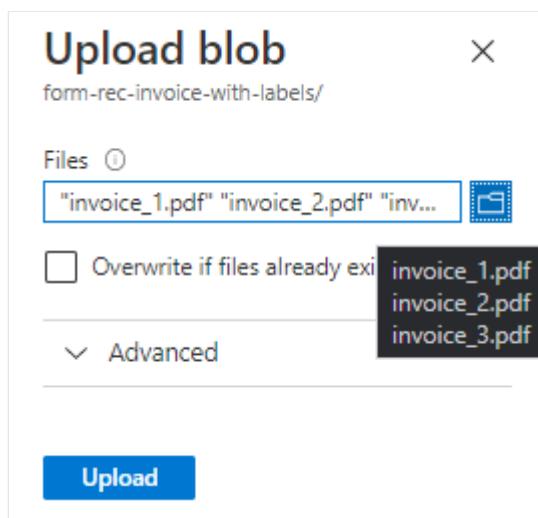
- 내 스토리지 계정 → 데이터 스토리지 → 컨테이너를 선택합니다.



2. 목록에서 컨테이너를 선택합니다.
3. 페이지 맨 위에 있는 메뉴에서 업로드를 선택합니다.



4. Blob 업로드 창이 표시됩니다. 업로드할 파일을 선택합니다.



① 참고

기본적으로 REST API는 컨테이너 루트에 있는 문서를 사용합니다. API 호출에 지정된 경우 하위 폴더에 구성된 데이터를 사용할 수도 있습니다. 자세한 내용은 [하위 폴더에서 데이터 구성](#)을 참조하세요.

Azure Portal 사용

Azure Portal은 GUI(그래픽 사용자 인터페이스)를 사용하여 Azure 구독 및 리소스를 관리 할 수 있는 웹 기반 콘솔입니다.

1. [Azure Portal](#)에 로그인합니다.
2. 스토리지 계정 > 컨테이너 > 컨테이너로 이동합니다.

3. 페이지 위쪽에 있는 메뉴에서 SAS 생성을 선택합니다.

4. 사용자 위임 키를 → 서명 방법을 선택합니다.

5. 적절한 확인란을 선택하거나 선택 취소하여 사용 권한을 정의합니다.

- 읽기, 쓰기, 삭제, 목록 보기 권한이 선택되어 있는지 확인합니다.

Generate SAS

A shared access signature (SAS) is a URI that grants restricted access to an Azure Storage container. Use it when you want to grant access to storage account resources for a specific time range without sharing your storage account key. [Learn more](#)

Signing method

Account key User delegation key

Permissions * ⓘ

4 selected

Read (UTC-08:00) Pacific Time (US & Canada) 2:19:14 PM

Add (UTC-08:00) Pacific Time (US & Canada) 10:19:14 PM

Create (UTC-08:00) Pacific Time (US & Canada) 10:19:14 PM

Write (UTC-08:00) Pacific Time (US & Canada) 10:19:14 PM

Delete (UTC-08:00) Pacific Time (US & Canada) 10:19:14 PM

List (UTC-08:00) Pacific Time (US & Canada) 10:19:14 PM

Allowed IP addresses ⓘ
for example, 168.1.5.65 or 168.1.5.65-168.1...

Allowed protocols ⓘ

HTTPS only HTTPS and HTTP

Generate SAS token and URL

ⓘ 중요

- 다음과 유사한 메시지가 표시되는 경우 스토리지 계정의 Blob 데이터에 대한 액세스도 할당해야 합니다.

⚠️ You don't have permissions to grant read, write, delete access. You can still create a shared access signature, but you'll need an RBAC role with additional permissions before you can grant that level of access to your signature recipient.
[Learn more about Azure roles for access to blob data](#)

- Azure RBAC([Azure 역할 기반 액세스 제어](#))는 Azure 리소스에 대한 액세스를 관리하는 데 사용하는 권한 부여 시스템입니다. Azure RBAC를 사용하면 Azure 리소스에 대한 액세스 및 권한을 관리할 수 있습니다.
- [Blob 데이터에 액세스하기 위한 Azure 역할 할당](#)에서는 Azure Storage 컨테이너에 대한 읽기, 쓰기, 삭제 권한을 허용하는 역할을 할당합니다. [Storage Blob 데이터 기여자를 참조하세요.](#)

6. 서명된 키 시작 및 만료 시간을 지정합니다.

- SAS 토큰을 생성할 때 기본 기간은 48시간입니다. 48시간이 지나면 새 토큰을 만들어야 합니다.
- 문서 인텔리전스 Service 작업에 스토리지 계정을 사용하는 기간을 더 길게 설정하는 것이 좋습니다.
- 만료 시간 값은 계정 키 또는 [사용자 위임 키 서명 방법을 사용하는지 여부에 따라 결정됩니다.](#)
 - **계정 키:** 최대 시간 제한은 없습니다. 그러나 모범 사례에서는 간격을 제한하고 손상을 최소화하도록 만료 정책을 구성하는 것이 좋습니다. [공유 액세스 서명에 대한 만료 정책을 구성합니다.](#)
 - **사용자 위임 키:** 만료 시간 값은 SAS 토큰 생성 스크립트 후 최대 7일입니다. 사용자 위임 키가 만료된 후에는 SAS가 유효하지 않으므로 만료 시간이 7일 보다 긴 SAS는 여전히 7일 동안만 유효합니다. 자세한 내용은 [Microsoft Entra 자격 증명을 사용하여 SAS를 보호하세요.](#)

7. 허용된 IP 주소 필드는 선택 사항이며 요청을 수락할 IP 주소 또는 IP 주소 범위를 지정합니다. 요청 IP 주소가 SAS 토큰에 지정된 IP 주소 또는 주소 범위와 일치하지 않으면 권한 부여가 실패합니다. IP 주소 또는 IP 주소 범위는 프라이빗이 아닌 공용 IP여야 합니다. 자세한 내용은, [IP 주소 또는 IP 범위 지정을 참조하세요.](#)

8. 허용되는 프로토콜 필드는 선택 사항으로 SAS 토큰으로 만든 요청에 대해 허용되는 프로토콜을 지정합니다. 기본값은 HTTPS입니다.

9. SAS 토큰 및 URL 생성을 선택합니다.

10. **Blob SAS 토큰** 쿼리 문자열 및 **Blob SAS URL**이 창의 아래쪽 영역에 표시됩니다. Blob SAS 토큰을 사용하려면 스토리지 서비스 URI에 추가합니다.

11. **Blob SAS 토큰** 및 **Blob SAS URL** 값을 복사하여 안전한 위치에 붙여넣습니다. 이는 한 번만 표시되며 창을 닫은 후에는 검색할 수 없습니다.

12. [SAS URL을 구성](#)하려면 스토리지 서비스의 URL에 SAS 토큰(URI)을 추가합니다.

Azure Storage Explorer 사용

Azure Storage Explorer는 데스크톱에서 Azure 클라우드 스토리지 리소스를 쉽게 관리할 수 있는 무료 독립 실행형 앱입니다.

시작하기

- Windows, macOS 또는 Linux 개발 환경에 Azure Storage Explorer 앱이 설치되어 있어야 합니다.
- Azure Storage Explorer 앱을 설치한 후 문서 인텔리전스에 사용 중인 [스토리지 계정](#)에 연결합니다.

SAS 토큰 만들기

1. 로컬 컴퓨터에서 Azure Storage Explorer 앱을 열고 연결된 [스토리지 계정](#)으로 이동합니다.
2. 스토리지 계정 노드를 확장하고 Blob 컨테이너를 선택합니다.
3. Blob 컨테이너 노드를 확장하고 스토리지 컨테이너 노드를 마우스 오른쪽 단추로 클릭하여 옵션 메뉴를 표시합니다.
4. 옵션 메뉴에서 **공유 액세스 서명 가져오기**를 선택합니다.
5. **공유 액세스 서명** 창에서 다음을 선택합니다.
 - **Access 정책**을 선택합니다(기본값은 없음).
 - **서명된 키 시작 및 만료** 날짜 및 시간을 지정합니다. 생성된 후에는 SAS를 철회할 수 없으므로 짧은 수명이 권장됩니다.
 - **시작 및 만료** 날짜와 시간에 대한 **표준 시간대**(기본값은 로컬)를 선택합니다.
 - **읽기, 쓰기, 목록 및 삭제** 확인란을 선택하여 컨테이너 **권한**을 정의합니다.
 - **key1** 또는 **key2**를 선택합니다.
 - 검토하고 **만들기**를 선택합니다.
6. 컨테이너의 컨테이너 이름, **SAS URL** 및 **쿼리 문자열**이 포함된 새 창이 나타납니다.
7. **SAS URL** 및 **쿼리 문자열** 값을 복사하여 안전한 위치에 붙여넣습니다. 이는 한 번만 표시되면 창을 닫은 후에는 검색할 수 없습니다.
8. [SAS URL을 구성](#)하려면 스토리지 서비스의 URL에 SAS 토큰(URI)을 추가합니다.

SAS URL을 사용하여 액세스 권한 부여

SAS URL에는 특수한 [쿼리 매개 변수](#) 세트가 포함되어 있습니다. 이러한 매개 변수는 클라이언트가 리소스에 액세스하는 방법을 나타냅니다.

REST API

[REST API](#)와 함께 SAS URL을 사용하려면 요청 본문에 SAS URL을 추가합니다.

JSON

```
{  
    "source": "<BLOB SAS URL>"  
}
```

정말 간단하죠. 클라이언트가 데이터에 액세스하는 방법을 승인하기 위해 SAS 토큰을 생성하는 방법을 알아보았습니다.

다음 단계

[학습 데이터 집합 빌드](#)

사용자 지정 추출 모델 빌드 및 학습

아티클 • 2024. 04. 09.

이 콘텐츠의 적용 대상은 다음과 같습니다. ✓ v4.0(미리 보기) | 이전 버전: ✓ v3.1(GA) ✓ v3.0(GA) ✓ v2.1

문서 인텔리전스 모델을 시작하려면 5개의 학습 문서가 필요합니다. 문서가 5개 이상 있는 경우 사용자 지정 모델 학습을 시작할 수 있습니다. [사용자 지정 템플릿 모델](#)(사용자 지정 양식) 또는 [사용자 지정 신경망 모델](#)(사용자 지정 문서)을 학습시킬 수 있습니다. 학습 프로세스는 두 모델 모두에 대해 동일하며 이 문서에서는 두 모델 중 하나를 학습하는 과정을 안내합니다.

사용자 지정 모델 입력 요구 사항

먼저 학습 데이터 세트에서 문서 인텔리전스의 입력 요구 사항을 따르는지 확인합니다.

- 최상의 결과를 위해 문서당 하나의 명확한 사진 또는 고품질 스캔을 제공합니다.
- 지원 파일 형식:

[+] 테이블 확장

모델	PDF	이미지: JPEG/JPG, PNG, BMP, TIFF, HEIF	Microsoft Office: Word(DOCX), Excel(XLSX), PowerPoint(PPTX) 및 HTML
읽음	✓	✓	✓
레이아웃	✓	✓	✓ (2024-02-29-preview, 2023-10-31-preview)
일반 문서	✓	✓	
사전 제작	✓	✓	
사용자 지정 추출	✓	✓	
사용자 지정 분류	✓	✓	✓ (2024-02-29-preview)

- PDF 및 TIFF의 경우 최대 2000페이지를 처리할 수 있습니다(무료 계층 구독의 경우 처음 2페이지만 처리됨).

- 문서 분석을 위한 파일 크기는 유료(S0) 계층의 경우 500MB, 무료(F0) 계층의 경우 4MB입니다.
- 이미지 크기는 50 x 50픽셀에서 10,000 x 10,000픽셀 사이여야 합니다.
- PDF가 암호로 잠긴 경우에는 제출하기 전에 잠금을 해제해야 합니다.
- 추출할 텍스트의 최소 높이는 1024 x 768 픽셀 이미지의 경우 12픽셀입니다. 이 차원은 150DPI(인치당 도트 수)에서 약 8포인트 텍스트에 해당합니다.
- 사용자 지정 모델 학습의 경우 학습 데이터의 최대 페이지 수는 사용자 지정 템플릿 모델의 경우 500개, 사용자 지정 인공신경망 모델의 경우 50,000개입니다.
 - 사용자 지정 추출 모델 학습의 경우 학습 데이터의 총 크기는 템플릿 모델의 경우 50MB, 신경망 모델의 경우 1G-MB입니다.
 - 사용자 지정 분류 모델 학습의 경우 학습 데이터의 총 크기는 최대 10,000페이지 인 1GB 입니다.

학습 데이터 팁

학습을 위해 데이터 집합을 추가로 최적화하려면 다음 팁을 따르세요.

- 이미지 기반 문서 대신 텍스트 기반 PDF 문서를 사용합니다. 스캔한 PDF는 이미지로 처리됩니다.
- 입력 필드가 있는 양식에 대해 모든 필드가 완료된 예제를 사용합니다.
- 각 필드에 서로 다른 값이 있는 양식을 사용합니다.
- 양식 이미지의 품질이 낮은 경우 더 큰 데이터 세트(예: 이미지 10~15개)를 사용합니다.

학습 데이터 업로드

학습용으로 양식 또는 문서 집합을 모은 후에는 Azure Blob Storage 컨테이너에 업로드해야 합니다. 컨테이너를 사용하여 Azure 스토리지 계정을 만드는 방법을 모르는 경우 [Azure Portal의 Azure Storage 빠른 시작](#)을 따릅니다. 체험 가격 책정 계층(F0)을 통해 서비스를 사용해 보고, 나중에 프로덕션을 위한 유료 계층으로 업그레이드할 수 있습니다.

동영상: 사용자 지정 모델 학습

- 학습 데이터 세트를 수집하고 업로드하면 사용자 지정 모델을 학습시킬 준비가 된 것입니다. 다음 동영상에서는 프로젝트를 만들고 성공적인 모델 레이블 지정 및 학습을 위한 몇 가지 기본 사항을 살펴보겠습니다.

문서 인텔리전스 스튜디오에서 프로젝트 만들기

문서 인텔리전스 스튜디오는 데이터 세트를 완성하고 모델을 학습하는 데 필요한 모든 API 호출을 제공하고 조정합니다.

1. 문서 인텔리전스 스튜디오로 이동하여 시작합니다. Studio를 처음 사용하는 경우 구독, 리소스 그룹 및 리소스를 초기화해야 합니다. 그런 다음, 사용자 지정 프로젝트의 필수 구성 요소에 따라 학습 데이터 세트에 액세스하도록 Studio를 구성합니다.
2. Studio에서 사용자 지정 모델 타일을 선택하고 사용자 지정 모델 페이지에서 프로젝트 만들기 단추를 선택합니다.

Custom models

Extract information from forms and documents with custom models. Train a model by labeling as few as 5 example documents. (The same labeled dataset can train all types of custom models.) [Learn more about custom models](#).

Custom form models

Custom form models work well when the target documents share a common visual layout. Training only takes a few minutes, and more than 100 languages are supported.

Request for Taxpayer Identification Number and Certification

Form W-9
(Rev. October 2018)
Department of the Treasury
Internal Revenue Service

Go to www.irs.gov/FormW9 for instructions and the latest information.

Name as shown on your income tax return. Name is required on this line. Do not leave this line blank.
 John Doe

Check appropriate box for federal tax classification of the person whose name is entered on line 1. Check only one of the following boxes. If you are a corporation, partnership, or LLC, check the box for the classification of the entity under which you are doing business. If you are an individual, check the box for your classification as an individual.
 Artefact Inc.

Individual proprietor or single-member LLC
 Corporation
 Partnership
 Institution
 Exempt payee code (if any)

Limit liability company. Enter the tax classification for the LLC if it is a corporation, partnership, or LLC. Otherwise, check LLC if the LLC is classified as a single-member LLC that is disregarded from the owner unless the owner of the LLC is a corporation, partnership, or LLC itself. If the LLC is a corporation, partnership, or LLC, enter the classification of the LLC if it is disregarded from the owner; check the appropriate box for the tax classification of the owner.
 Other (see instructions) [See instructions](#)

[See instructions](#) [See instructions](#) [See instructions](#)

Custom document models

Custom document models can flexibly handle both structured and unstructured documents. Training takes up to half an hour, and currently only English language documents are supported. The current version can extract inline data and checkboxes.

HOUSE RENTAL AGREEMENT

This House Rental Agreement ("Agreement," "rental agreement," or "lease") is entered into between **John Doe** (Landlord) and **Jane Smith** **123 Main Street Apt. 101** (Tenant). If more than one person is named as Tenant they shall be jointly and severally liable and responsible under the terms of this Agreement. This lease Agreement involves a residential house, yard, and related facilities located at **123 Main Street Apt. 101** **Anytown, USA** ("premises"). The date of this Agreement is **January 1, 2024**.

1. Landlord rents to Tenant, unfurnished, the premises on a month to month basis, terminable by either party at the end of any calendar month on at least 30 days notice to the other party. Tenant shall be entitled to possession of the premises and rent shall commence on **January 1, 2024**. Tenant shall not assign, sublease, or allow anyone other than persons permitted under this lease to at any time be in possession of any portion of the premises. Landlord will provide five (5) keys to Tenant; each key fits all outside door locks. Landlord will also provide two keys to the **garage**, and one remote **garage door opener**. All keys and the remote **garage door opener** will be retained.

My Projects

+ Create a project Delete Search

- a. 프로젝트 만들기 대화 상자에서 프로젝트 이름, 선택적으로 설명을 입력하고 계속을 선택합니다.
- b. 워크플로의 다음 단계에서 계속을 선택하기 전에 문서 인텔리전스 리소스를 선택하거나 만듭니다.

ⓘ 중요

사용자 지정 신경망 모델은 일부 지역에서만 사용할 수 있습니다. 신경망 모델을 학습하려는 경우 지원되는 지역 중 하나에서 리소스를 선택하거나 만듭니다.

Custom models

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 *

Resource group *

[Create new](#)

Form Recognizer or Cognitive Service Resource *

Create new resource

API version *

3. 다음으로 사용자 지정 모델 학습 데이터 세트를 업로드할 때 사용한 스토리지 계정을 선택합니다. 학습 문서가 컨테이너의 루트에 있는 경우 **폴더 경로**는 비어 있어야 합니다. 문서가 하위 폴더에 있는 경우 **폴더 경로** 필드의 컨테이너 루트에서 상대 경로를 입력합니다. 스토리지 계정이 구성되면 계속을 선택합니다.

Custom models

Connect training data source

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

Subscription *

Resource group *

Storage account *

Blob container *

Folder path

Back **Continue** **Cancel**

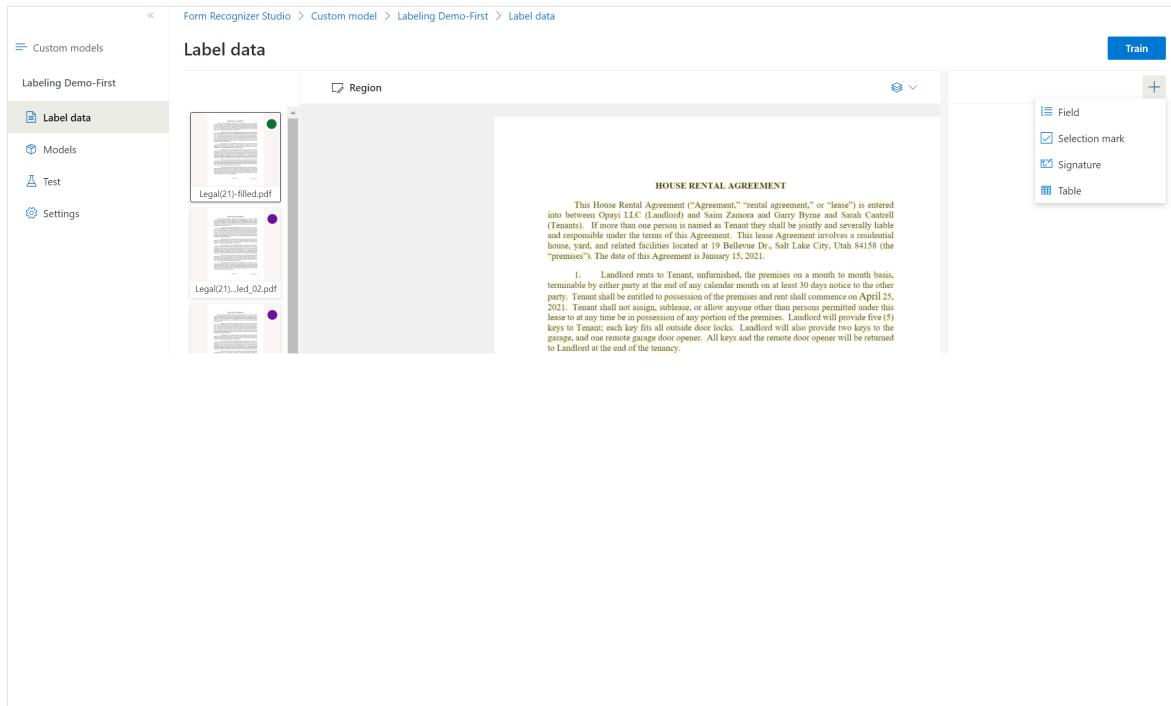
4. 마지막으로, 프로젝트 설정을 검토하고 **프로젝트 만들기**를 선택하여 새 프로젝트를 만듭니다. 이제 레이블 지정 창에 있고 나열된 데이터 세트의 파일을 확인해야 합니다.

데이터에 레이블 지정

프로젝트에서 첫 번째 작업은 추출하려는 필드로 데이터 세트에 레이블을 지정하는 것입니다.

스토리지에 업로드한 파일이 첫 번째 파일에 레이블을 지정할 준비가 된 상태로 화면 왼쪽에 표시됩니다.

1. 화면 오른쪽 위에 있는 더하기(+) 단추를 선택하여 데이터 세트에 레이블을 지정하고 첫 번째 필드를 만들기 시작합니다.



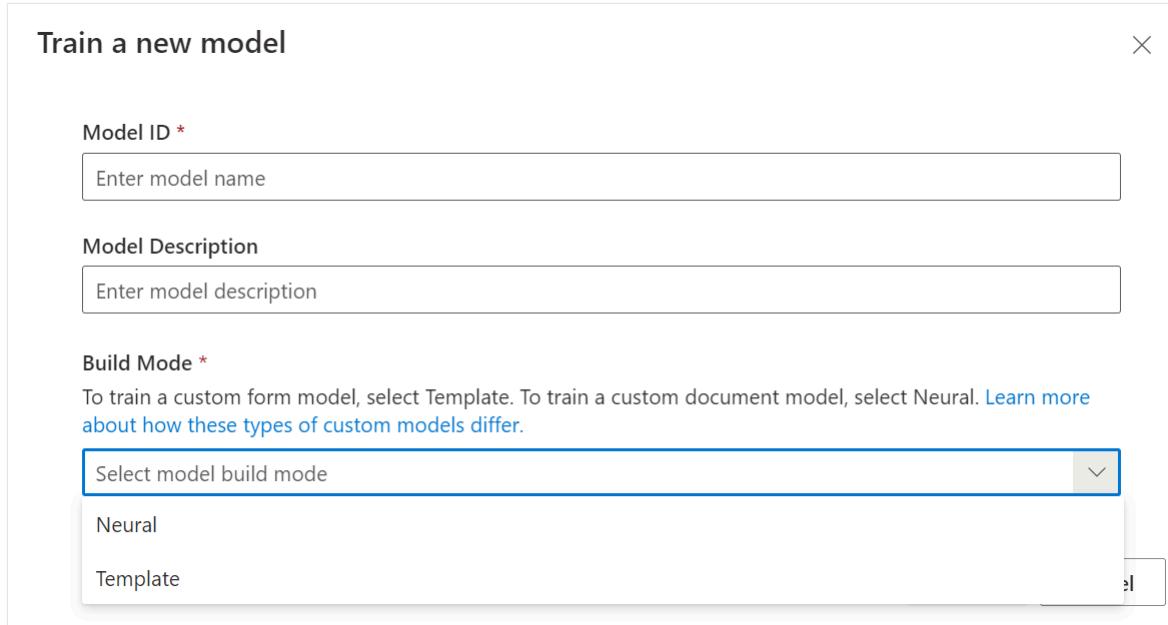
2. 필드의 이름을 입력합니다.
3. 문서에서 한 단어 또는 여러 단어를 선택하여 필드에 값을 할당합니다. 드롭다운 또는 오른쪽 탐색 모음의 필드 목록에서 필드를 선택합니다. 레이블이 지정된 값은 필드 목록의 필드 이름 아래에 표시됩니다.
4. 데이터 세트에 레이블을 지정하려는 모든 필드에 대해 프로세스를 반복합니다.
5. 각 문서를 선택하고 레이블을 지정할 텍스트를 선택하여 데이터 세트의 나머지 문서에 레이블을 지정합니다.

이제 데이터 세트에 레이블이 지정된 모든 문서가 있습니다. 학습 데이터 세트의 각 문서에 해당하는 `.labels.json` 및 `.ocr.json` 파일과 새 `fields.json` 파일입니다. 이 학습 데이터 세트는 모델 학습을 위해 제출됩니다.

모델 학습

레이블이 지정된 데이터 세트를 사용하면 이제 모델을 학습할 준비가 되었습니다. 오른쪽 위 모서리에서 학습 단추를 선택합니다.

- 모델 학습 대화 상자에서 고유한 모델 ID와 설명(선택 사항)을 제공합니다. 모델 ID는 문자열 데이터 형식을 허용합니다.
- 빌드 모드의 경우 학습할 모델 유형을 선택합니다. [모델 형식 및 기능](#)에 대해 자세히 알아봅니다.



- 학습을 선택하여 학습 프로세스를 시작합니다.
- 템플릿 모델은 몇 분 안에 학습됩니다. 신경망 모델은 학습하는 데 최대 30분이 걸릴 수 있습니다.
- 모델 메뉴로 이동하여 학습 작업의 상태를 확인합니다.

모델 테스트

모델 학습이 완료되면 모델 목록 페이지에서 모델을 선택하여 모델을 테스트할 수 있습니다.

- 모델을 선택하고 **테스트** 단추를 선택합니다.
- + Add 단추를 선택하여 모델을 테스트할 파일을 선택합니다.
- 파일을 선택한 상태에서 **분석** 단추를 선택하여 모델을 테스트합니다.
- 모델 결과가 주 창에 표시되고 추출된 필드가 오른쪽 탐색 모음에 나열됩니다.
- 각 필드에 대한 결과를 평가하여 모델의 유효성을 검사합니다.

6. 오른쪽 탐색 모음에는 모델을 호출하는 샘플 코드와 API의 JSON 결과도 있습니다.

축하합니다! 문서 인텔리전스 스튜디오에서 사용자 지정 모델을 학습시키는 방법을 알아보았습니다. 모델은 REST API 또는 SDK와 함께 문서를 분석하는 데 사용할 준비가 된 것입니다.

다음 단계

이제 학습 데이터 세트를 빌드하는 방법을 알아보았으므로 빠른 시작에 따라 사용자 지정 문서 인텔리전스 모델을 학습시키고 양식에서 사용하기 시작합니다.

[사용자 지정 모델 형식에 대한 자세한 정보](#)

[사용자 지정 모델을 사용하는 정확도 및 신뢰도에 대한 자세한 정보](#)

참고 항목

- [클라이언트 라이브러리 또는 REST API를 사용하여 모델 학습 및 문서 데이터 추출](#)
- [문서 인텔리전스란?](#)

사용자 지정 분류 모델 빌드 및 학습

아티클 • 2024. 03. 05.

이 콘텐츠의 적용 대상은 다음과 같습니다. ✓ v4.0(미리 보기) | 이전 버전: ✓ v3.1(GA) ✓ v3.0(GA)

① 중요

사용자 지정 분류 모델은 현재 공개 미리 보기로 제공됩니다. 기능, 방식 및 프로세스는 GA(일반 공급) 전에 사용자 피드백에 따라 변경될 수 있습니다.

사용자 지정 분류 모델은 입력 파일의 각 페이지를 분류하여 내 문서를 식별할 수 있습니다. 분류자 모델은 입력 파일에서 여러 문서 또는 단일 문서의 여러 인스턴스를 식별할 수도 있습니다. 문서 인텔리전스 사용자 지정 모델을 시작하려면 학습 문서가 최소 5개 필요합니다. 사용자 지정 분류 모델 학습을 시작하려면 각 클래스에 대해 최소 5개의 문서와 두 개의 문서 클래스가 필요합니다.

사용자 지정 분류 모델 입력 요구 사항

먼저 학습 데이터 집합에서 문서 인텔리전스의 입력 요구 사항을 따르는지 확인합니다.

- 최상의 결과를 위해 문서당 하나의 명확한 사진 또는 고품질 스캔을 제공합니다.
- 지원 파일 형식:

☰ 테이블 확장

모델	PDF	이미지: JPEG/JPG, PNG, BMP, TIFF, HEIF	Microsoft Office: Word(DOCX), Excel(XLSX), PowerPoint(PPTX) 및 HTML
읽음	✓	✓	✓
레이아웃	✓	✓	✓ (2023-10-31-preview)
일반 문서	✓	✓	
사전 제작	✓	✓	
사용자	✓	✓	

모델	PDF	이미지: JPEG/JPG, PNG, BMP, TIFF, HEIF	Microsoft Office: Word(DOCX), Excel(XLSX), PowerPoint(PPTX) 및 HTML
지정			

- PDF 및 TIFF의 경우 최대 2000페이지를 처리할 수 있습니다(무료 계층 구독의 경우 처음 2페이지만 처리됨).
- 문서 분석을 위한 파일 크기는 유료(S0) 계층의 경우 500MB, 무료(F0) 계층의 경우 4MB입니다.
- 이미지 크기는 50 x 50픽셀에서 10,000 x 10,000픽셀 사이여야 합니다.
- PDF가 암호로 잠긴 경우에는 제출하기 전에 잠금을 해제해야 합니다.
- 추출할 텍스트의 최소 높이는 1024 x 768 픽셀 이미지의 경우 12픽셀입니다. 이 차원은 150DPI(인치당 도트 수)에서 약 8포인트 텍스트에 해당합니다.
- 사용자 지정 모델 학습의 경우 학습 데이터의 최대 페이지 수는 사용자 지정 템플릿 모델의 경우 500개, 사용자 지정 인공신경망 모델의 경우 50,000개입니다.
 - 사용자 지정 추출 모델 학습의 경우 학습 데이터의 총 크기는 템플릿 모델의 경우 50MB, 신경망 모델의 경우 1G-MB입니다.
 - 사용자 지정 분류 모델 학습의 경우 학습 데이터의 총 크기는 최대 10,000페이지인 1GB입니다.

학습 데이터 팁

학습을 위해 데이터 집합을 추가로 최적화하려면 다음 팁을 따르세요.

- 가능하면 이미지 기반 문서 대신 텍스트 기반 PDF 문서를 사용합니다. 스캔한 PDF는 이미지로 처리됩니다.
- 양식 이미지의 품질이 낮은 경우 더 큰 데이터 집합(예: 10-15개 이미지)를 사용합니다.

학습 데이터 업로드

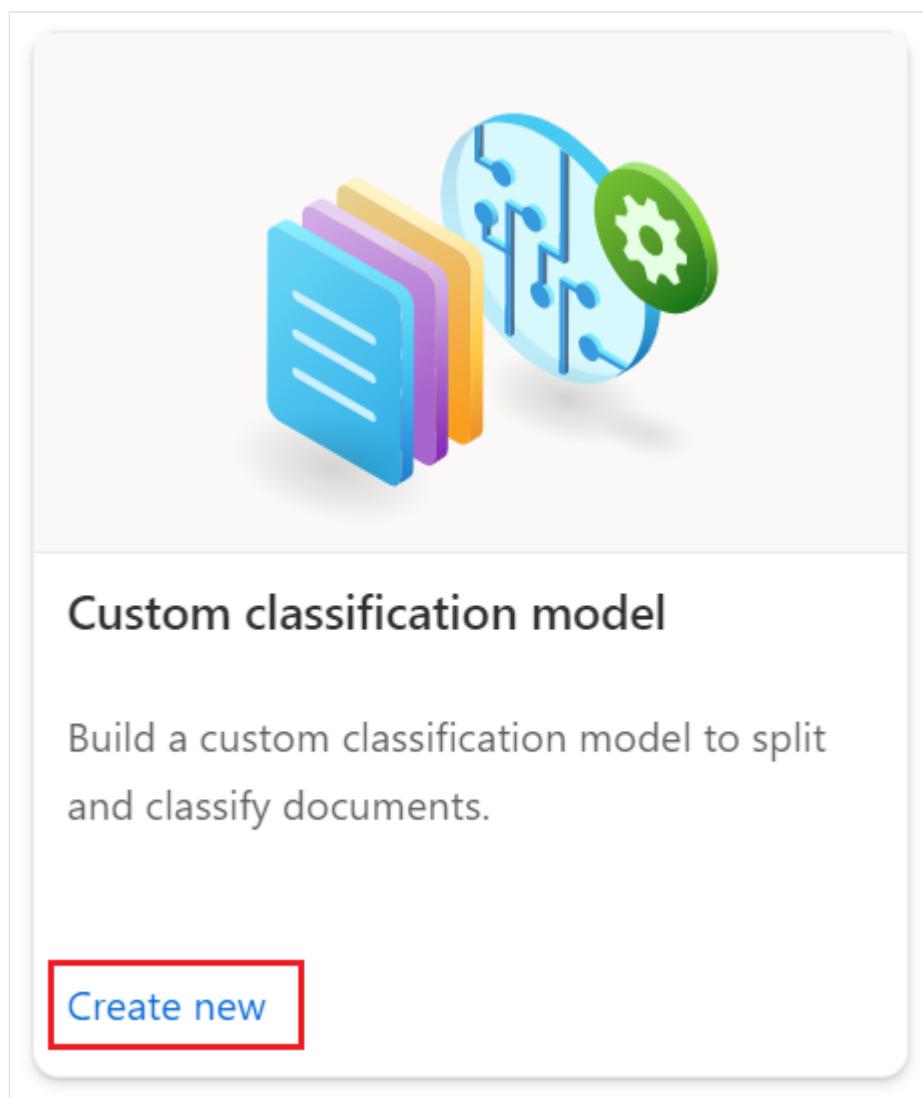
학습용으로 양식 또는 문서 집합을 구성한 후에는 Azure Blob Storage 컨테이너에 업로드해야 합니다. 컨테이너를 사용하여 Azure 스토리지 계정을 만드는 방법을 모르는 경우 [Azure Portal의 Azure Storage 빠른 시작](#)을 따릅니다. 체험 가격 책정 계층(F0)을 통해 서비스를 사용해 보고, 나중에 프로덕션을 위한 유료 계층으로 업그레이드할 수 있습니다.

데이터 세트가 폴더로 구성된 경우 Studio에서 레이블에 폴더 이름을 사용하여 레이블 지정 프로세스를 간소화할 수 있으므로 해당 구조를 유지합니다.

문서 인텔리전스 스튜디오에서 분류 프로젝트 만들기

문서 인텔리전스 스튜디오는 데이터 세트를 완성하고 모델을 학습하는 데 필요한 모든 API 호출을 제공하고 조정합니다.

1. 문서 [인텔리전스 스튜디오](#)로 이동하여 시작합니다. Studio를 처음 사용하는 경우 구독, 리소스 그룹 및 리소스를 초기화해야 합니다. 그런 다음, [사용자 지정 프로젝트의 필수 구성 요소](#)에 따라 학습 데이터 세트에 액세스하도록 Studio를 구성합니다.
2. Studio에서 **사용자 지정 분류 모델** 타일을 선택하고 페이지의 사용자 지정 모델 섹션에서 **프로젝트 만들기** 단추를 선택합니다.



- a. 프로젝트 만들기 대화 상자에서 프로젝트 이름, 선택적으로 설명을 입력하고 계속을 선택합니다.

- b. 다음 단계에서, 계속을 선택하기 전에 문서 인텔리전스 리소스를 선택하거나 만듭니다.

Custom models

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 *

Resource group *

Create new

Form Recognizer or Cognitive Service Resource *

Create new resource

API version *

3. 다음으로 사용자 지정 모델 학습 데이터 세트를 업로드할 때 사용한 스토리지 계정을 선택합니다. 학습 문서가 컨테이너의 루트에 있는 경우 **폴더 경로**는 비어 있어야 합니다. 문서가 하위 폴더에 있는 경우 **폴더 경로** 필드의 컨테이너 루트에서 상대 경로를 입력합니다. 스토리지 계정이 구성되면 계속을 선택합니다.

ⓘ 중요

폴더 이름이 문서의 레이블 또는 클래스인 폴더별로 학습 데이터 세트를 구성하거나 Studio에서 레이블을 할당할 수 있는 문서의 플랫 목록을 만들 수 있습니다.

Custom models

X

- 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 *

Resource group *

Storage account *

Blob container *

Folder path

Back

Continue

Cancel

4. 사용자 지정 분류자를 학습하려면 데이터 세트의 각 문서에 대한 레이아웃 모델의 출력이 필요합니다. 모델 학습 프로세스 전에 모든 문서에서 레이아웃을 실행합니다.
5. 마지막으로, 프로젝트 설정을 검토하고 **프로젝트 만들기**를 선택하여 새 프로젝트를 만듭니다. 이제 레이블 지정 창에 있고 나열된 데이터 세트의 파일을 확인해야 합니다.

데이터에 레이블 지정

프로젝트에서는 적절한 클래스 레이블을 사용하여 각 문서에 레이블을 지정하기만 하면 됩니다.

The screenshot shows the Microsoft Applied AI | Form Recognizer Studio interface. At the top, there's a navigation bar with 'Applied AI | Form Recognizer Studio' and a user profile icon. Below it, a breadcrumb navigation shows 'Form Recognizer Studio > Custom classification model'. A main title 'Custom classification models' is followed by a message: 'Custom classification model welcome message (To be update). Learn more about custom classification model.' On the left, there's a sidebar with a 'Drag and drop' area for uploading files and a 'My Projects' section with a 'Create a project' button and a search bar. The main content area shows a grid of project thumbnails, each labeled with a unique identifier like 'sampleDoc1', 'sampleDoc2', etc. A tooltip 'Search existing or create new' points to a search input field.

파일 목록에서 스토리지에 업로드한 파일이 레이블 지정을 위해 준비된 것을 볼 수 있습니다. 데이터 세트에 레이블을 지정하는 몇 가지 옵션이 있습니다.

1. 문서가 폴더로 구성된 경우 스튜디오에서 폴더 이름을 레이블로 사용하라는 메시지를 표시합니다. 이 단계에서는 레이블을 단일 선택으로 간소화합니다.
2. 문서에 레이블을 할당하려면 레이블 선택 표시 추가를 선택하여 레이블을 할당합니다.
3. 레이블을 할당하기 위해 다중 선택 문서에 대한 선택 제어

이제 데이터 세트에 레이블이 지정된 모든 문서가 있습니다. 스토리지 계정을 살펴보면 학습 데이터 세트의 각 문서에 해당하는 `.ocr.json` 파일과 각 레이블 지정된 클래스에 대한 새 `class-name.jsonl` 파일을 찾을 수 있습니다. 이 학습 데이터 세트는 모델 학습을 위해 제출됩니다.

모델 학습

레이블이 지정된 데이터 세트를 사용하면 이제 모델을 학습할 준비가 되었습니다. 오른쪽 위 모서리에서 학습 단추를 선택합니다.

1. 모델 학습 대화 상자에서 고유한 분류자 ID와 설명(선택 사항)을 제공합니다. 분류자 ID는 문자열 데이터 형식을 허용합니다.
2. 학습을 선택하여 학습 프로세스를 시작합니다.
3. 분류자 모델은 몇 분 안에 학습됩니다.
4. 모델 메뉴로 이동하여 학습 작업의 상태를 확인합니다.

모델 테스트

모델 학습이 완료되면 모델 목록 페이지에서 모델을 선택하여 모델을 테스트할 수 있습니다.

1. 모델을 선택하고 **테스트** 단추를 선택합니다.
2. 파일을 검색하거나 문서 선택기에 파일을 삭제하여 새 파일을 추가합니다.
3. 파일을 선택한 상태에서 **분석** 단추를 선택하여 모델을 테스트합니다.
4. 모델 결과는 식별된 문서 목록, 식별된 각 문서에 대한 신뢰도 점수 및 식별된 각 문서의 페이지 범위와 함께 표시됩니다.
5. 식별된 각 문서에 대한 결과를 평가하여 모델의 유효성을 검사합니다.

SDK 또는 API를 사용하여 사용자 지정 분류자 학습

스튜디오는 사용자 지정 분류자를 학습시키도록 API 호출을 오케스트레이션합니다. 분류자 학습 데이터 세트에는 학습 모델의 API 버전과 일치하는 레이아웃 API의 출력이 필요합니다. 이전 API 버전의 레이아웃 결과를 사용하면 모델의 정확도가 낮아질 수 있습니다.

데이터 세트에 레이아웃 결과가 포함되어 있지 않으면 스튜디오가 학습 데이터 세트의 레이아웃 결과를 생성합니다. API 또는 SDK를 사용하여 분류자를 학습시키는 경우 개별 문서가 포함된 폴더에 레이아웃 결과를 추가해야 합니다. 레이아웃을 직접 호출할 때 레이아웃 결과는 API 응답 형식이어야 합니다. SDK 개체 모델은 다르며, `layout results` 가 `SDK response` 가 아니라 API 결과인지 확인합니다.

문제 해결

[분류 모델](#)에는 각 학습 문서에 대한 [레이아웃 모델](#)의 결과가 필요합니다. 레이아웃 결과를 제공하지 않으면 스튜디오는 분류자를 학습시키기 전에 각 문서의 레이아웃 모델을 실행하려고 시도합니다. 이 프로세스는 제한되며 429 응답이 발생할 수 있습니다.

Studio에서 분류 모델을 사용하여 학습하기 전에 각 문서에서 [레이아웃 모델](#)을 실행하고 원래 문서와 동일한 위치에 업로드합니다. 레이아웃 결과가 추가되면 문서를 사용하여 분류자 모델을 학습시킬 수 있습니다.

다음 단계

사용자 지정 모델 형식에 대한 자세한 정보

사용자 지정 모델을 사용하는 정확도 및 신뢰도에 대한 자세한 정보

문서 인텔리전스 스튜디오를 사용하여 프로젝트 공유

아티클 • 2024. 02. 29.

이 콘텐츠의 적용 대상은 다음과 같습니다. ✓ v4.0(미리 보기) | 이전 버전: ✓ v3.1(GA) ✓ v3.0(GA) ✓ v2.1(GA)

문서 인텔리전스 스튜디오는 문서 인텔리전스 서비스의 기능을 시각적으로 탐색, 이해, 훈련 및 애플리케이션에 통합하기 위한 온라인 도구입니다. 문서 인텔리전스 스튜디오를 사용하면 사용자 지정 추출 모델 내에서 프로젝트 공유 기능을 사용할 수 있습니다. 프로젝트 토큰을 통해 프로젝트를 쉽게 공유할 수 있습니다. 동일한 프로젝트 토큰을 사용하여 프로젝트를 가져올 수도 있습니다.

전제 조건

사용자 지정 추출 프로젝트를 원활하게 공유하고 가져오려면 사용자(공유하는 사용자 및 가져오는 사용자) 모두 활성 [Azure 계정](#)이 필요합니다. 계정이 없는 경우 [무료 계정](#)에 만들 수 있습니다. 또한 두 사용자 모두 문서 인텔리전스 및 스토리지 리소스에 대한 액세스 권한을 부여하도록 권한을 구성해야 합니다.

일반적으로 사용자 지정 모델 프로젝트를 만드는 과정에서 프로젝트 공유를 위해 대부분의 요구 사항이 충족되어야 합니다. 그러나 프로젝트 공유 기능이 작동하지 않는 경우 [권한을 확인하세요](#).

액세스 및 사용 권한 부여

① 중요

사용자 지정 모델 프로젝트는 가져오려는 프로젝트와 연결된 스토리지 계정에 대한 액세스 권한이 있는 경우에만 가져올 수 있습니다. 프로젝트를 다른 사용자와 공유하거나 가져오기 전에 스토리지 계정 권한을 확인하세요.

가상 네트워크 및 방화벽

스토리지 계정 VNet을 사용하도록 설정하거나 방화벽 제약 조건이 있는 경우 프로젝트를 공유할 수 없습니다. 이러한 제한을 무시하려면 해당 설정이 꺼져 있는지 확인하세요.

해결 방법은 공유 중인 프로젝트와 동일한 설정을 사용하여 프로젝트를 수동으로 만드는 것입니다.

문서 인텔리전스 스튜디오와 사용자 지정 추출 모델 공유

문서 인텔리전스 스튜디오를 사용하여 프로젝트를 공유하려면 다음 단계를 수행합니다.

1. 문서 인텔리전스 스튜디오로 이동하여 시작합니다.
2. 스튜디오의 사용자 지정 모델 섹션에서 사용자 지정 추출 모델 타일을 선택합니다.

Custom models
Train custom models to classify documents and extract text, structure and fields from your forms or documents.

Custom extraction model
Label and build a custom model to extract a specific schema from your forms and documents.
[Create new](#)

Custom classification model
Build a custom classification model to split and classify documents.
[Create new](#)

3. 사용자 지정 추출 모델 페이지에서 공유할 원하는 모델을 선택한 다음 **공유** 단추를 선택합니다.

Applied AI | Form Recognizer Studio

Form Recognizer Studio > Custom extraction model

Custom extraction models

Extract information from forms and documents with custom extraction models. Train a model by labeling as few as 5 example documents. (The same labeled dataset can train all types of custom extraction models.) [Learn more about custom extraction models.](#)

Template (Custom form) models

Template models work well when the target documents share a common visual layout. Training only takes a few minutes, and more than 100 languages are supported.

Request for Taxpayer Identification Number and Certification

Neural (Custom document) models

Neural models can handle both structured and unstructured documents. Training takes up to half an hour, and currently only English language documents are supported. The current version can extract inline field data and checkboxes. Neural models are available only in select regions. [Click here for details.](#)

HOUSE RENTAL AGREEMENT

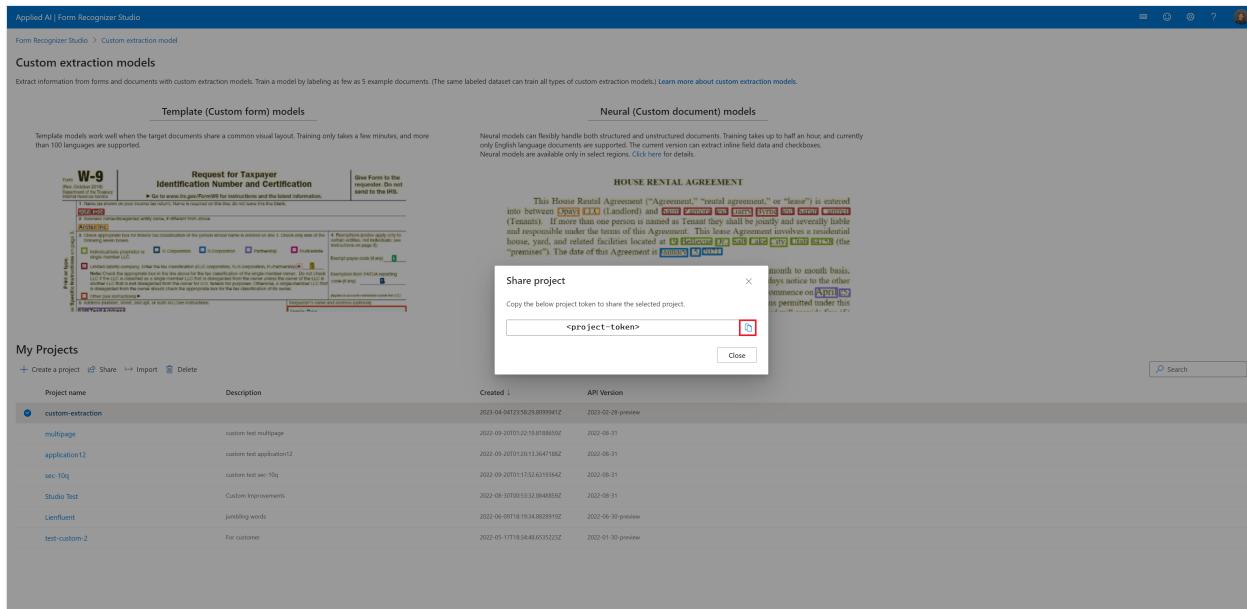
This House Rental Agreement ("Agreement," "rental agreement," or "lease") is entered into between **Landlord** (Landlord) and **Tenant** (Tenant). If more than one person is named as Tenant they shall be jointly and severally liable and responsible under the terms of this Agreement. This lease Agreement involves a residential house located at **Address** (the "premises"). Landlord and Tenant shall be referred to as "parties" and each party shall be referred to as "the party". The date of this Agreement is **01/01/2023**.

My Projects

+ Create a project **Share** Import Delete

Project name	Description	Created	API Version
custom-extraction		2023-04-04T23:58:29.809944Z	2023-02-28-preview
multipage	custom test/multipage	2022-09-20T01:22:19.188651Z	2022-08-31
application12	custom test/application12	2022-09-20T01:20:13.3647188Z	2022-08-31
sec-10q	custom test/sec-10q	2022-09-20T01:17:52.631934Z	2022-08-31
Studio Test	Custom Improvements	2022-09-30T03:33:21.8848893Z	2022-08-31
Lienfuent	Junking words	2022-06-09T19:19.4848891Z	2022-06-30-preview
test-custom-2	For customer	2022-05-17T18:34:48.631932Z	2022-05-30-preview

4. 프로젝트 공유 대화 상자에서 선택한 프로젝트에 대한 프로젝트 토큰을 복사합니다.



문서 인텔리전스 스튜디오와 사용자 지정 추출 모델 가져오기

문서 인텔리전스 스튜디오를 사용하여 프로젝트를 가져오려면 다음 단계를 수행합니다.

1. 문서 인텔리전스 스튜디오로 이동하여 시작합니다.

2. 스튜디오의 사용자 지정 모델 섹션에서 사용자 지정 추출 모델 타일을 선택합니다.

Custom models

Train custom models to classify documents and extract text, structure and fields from your forms or documents.

Custom extraction model

Label and build a custom model to extract a specific schema from your forms and documents.

Create new

Custom classification model

Build a custom classification model to split and classify documents.

Create new

3. 사용자 지정 추출 모델 페이지에서 가져오기 단추를 선택합니다.

4. 프로젝트 가져오기 대화 상자에서 공유된 프로젝트 토큰을 붙여넣고 가져오기를 선택합니다.

Form Recognizer Studio > Custom extraction model

Custom extraction models

Extract information from forms and documents with custom extraction models. Train a model by labeling as few as 5 example documents. (The same labeled dataset can train all types of custom extraction models.) [Learn more about custom extraction models](#).

Template (Custom form) models

Template models work well when the target documents share a common visual layout. Training only takes a few minutes, and more than 100 languages are supported.

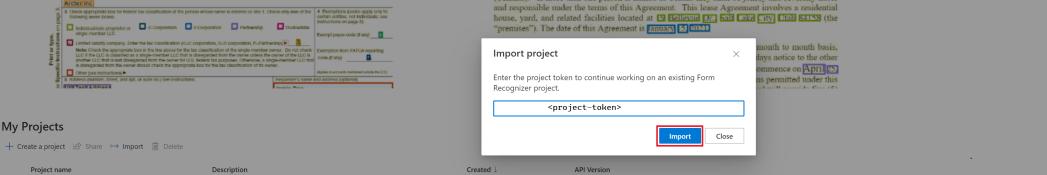
 

Neural (Custom document) models

Neural models can handle both structured and unstructured documents. Training takes up to half an hour, and currently only English language documents are supported. The current version can extract inline field data and checkboxes. Neural models are available only in select regions. [Click here to learn more](#).

My Projects

다음 단계

모델 백업 및 복구

사용자 지정 모델 작성

아티클 • 2024. 02. 29.

이 콘텐츠의 적용 대상은 다음과 같습니다. ✓ v4.0(미리 보기) | 이전 버전: ✓ v3.1(GA) ✓ v3.0(GA) ✓ v2.1(GA)

구성된 모델은 사용자 지정 모델 컬렉션을 가져와 단일 모델 ID에 할당하여 만듭니다. 구성된 단일 모델 ID에 최대 200개의 학습된 사용자 지정 모델을 할당할 수 있습니다. 구성형 모델에 문서가 제출되면 이 서비스에서는 분류 단계를 수행하여 분석용으로 제공된 양식을 정확히 나타내는 사용자 지정 모델을 결정합니다. 구성된 모델은 여러 모델을 학습시키고 비슷한 양식 유형을 분석하도록 그룹화하려는 경우에 유용합니다. 예를 들어 구성형 모델은 물자, 장비 및 가구 구매 주문을 분석하도록 학습된 사용자 지정 모델을 포함할 수 있습니다. 적절한 모델을 수동으로 선택하는 대신, 구성형 모델을 사용하여 각 분석 및 추출에 적합한 사용자 지정 모델을 결정할 수 있습니다.

자세한 내용은 [구성된 사용자 지정 모델](#)을 참조하세요.

이 문서에서는 작성된 사용자 지정 모델을 만들고 사용하여 양식과 문서를 분석하는 방법을 알아봅니다.

필수 조건

시작하기 전에 다음 리소스가 필요합니다.

- **Azure 구독.** [체험 Azure 구독을 만들](#) 수 있습니다.
- **문서 인텔리전스 인스턴스.** Azure를 구독하고 나면 Azure portal에서 [문서 인텔리전스 리소스를 생성](#)하여 키와 엔드포인트를 가져옵니다. 기존 문서 인텔리전스 리소스가 있는 경우 리소스 페이지로 직접 이동하세요. 체험 가격 책정 계층(F0)을 통해 서비스를 사용해 보고, 나중에 프로덕션을 위한 유료 계층으로 업그레이드할 수 있습니다.
 1. 리소스가 배포된 후 **리소스로 이동**을 선택합니다.
 2. Azure Portal에서 **키 및 엔드포인트** 값을 복사하여 Microsoft 메모장 같은 편리한 위치에 붙여넣습니다. 애플리케이션을 문서 인텔리전스 API에 연결하려면 키와 엔드포인트 값이 필요합니다.

The screenshot shows the Azure portal interface for a Cognitive Services resource named 'Contoso-DI'. The left sidebar lists various management options like Overview, Activity log, Access control (IAM), Tags, Diagnose and solve problems, Resource Management, and Keys and Endpoint (which is selected and highlighted with a red box). The main content area shows a summary of keys and endpoint. A callout box provides information about the keys: 'These keys are used to access your Azure AI service API. Do not share your keys. Store them securely—for example, using Azure Key Vault. We also recommend regenerating these keys regularly. Only one key is necessary to make an API call. When regenerating the first key, you can use the second key for continued access to the service.' Below this, there are fields for 'KEY 1' (redacted), 'KEY 2' (redacted), 'Location/Region' set to 'westus2', and 'Endpoint' (highlighted with a red box) set to 'https://contoso-di.cognitiveservices.azure.com/'. There are also download icons next to each key field.

💡 팁

자세한 내용은 [문서 인텔리전스 리소스 만들기](#)를 참조하세요.

- **Azure 스토리지 계정.** Azure 스토리지 계정을 만드는 방법을 모르는 경우 [Azure Portal의 Azure Storage 빠른 시작](#)을 따르세요. 체험 가격 책정 계층(F0)을 통해 서비스를 사용해 보고, 나중에 프로덕션을 위한 유료 계층으로 업그레이드할 수 있습니다.

사용자 지정 모델 만들기

먼저 구성할 사용자 지정 모델 세트가 필요합니다. 문서 인텔리전스 스튜디오, REST API 또는 클라이언트-라이브러리 SDK를 사용할 수 있습니다. 절차는 다음과 같습니다.

- [학습 데이터 세트 조합](#)
- [Azure Blob 스토리지에 학습 세트 업로드](#)
- [사용자 지정 모델 학습](#)

학습 데이터 세트 조합

사용자 지정 모델 빌드는 학습 데이터 세트를 설정하는 것으로 시작합니다. 샘플 데이터 세트에 대해 동일한 유형으로 완성된 5개 이상의 양식이 필요합니다. 파일 형식이 다를 수 있으며(jpg, png, pdf, tiff) 텍스트와 필기를 모두 포함할 수 있습니다. 양식은 문서 인텔리전스의 [입력 요구 사항](#)을 따라야 합니다.

💡 팁

학습을 위해 데이터 세트를 최적화하려면 다음 팁을 따릅니다.

- 가능하면 이미지 기반 문서 대신 텍스트 기반 PDF 문서를 사용합니다. 스캔한 PDF는 이미지로 처리됩니다.
- 채워진 양식의 경우 모든 필드가 채워진 예제를 사용합니다.
- 각 필드에 서로 다른 값이 있는 양식을 사용합니다.
- 양식 이미지의 품질이 낮은 경우 더 큰 데이터 집합(예: 10-15개 이미지)를 사용합니다.

학습 문서를 수집하는 방법에 대한 팁은 [학습 데이터 세트 빌드](#)를 참조하세요.

학습 데이터 세트 업로드

학습 문서 집합을 수집한 경우 Azure Blob Storage 컨테이너에 [학습 데이터를 업로드](#)해야 합니다.

수동으로 레이블이 지정된 데이터를 사용하려면 학습 문서에 해당하는 *.labels.json* 및 *.ocr.json* 파일도 업로드해야 합니다.

사용자 지정 모델 학습

레이블이 지정된 데이터로 [모델을 학습](#) 시키는 경우 모델은 감독 학습을 통해 사용자가 제공하는 레이블 지정 양식을 사용하여 관심 있는 값을 추출합니다. 레이블이 지정된 데이터는 모델 성능이 향상되며, 복잡한 양식 또는 키 없는 값을 포함하는 양식과 함께 작동하는 모델을 생성할 수 있습니다.

문서 인텔리전스는 [미리 빌드된 레이아웃 모델](#) API를 사용하여 서체 및 필기된 텍스트 요소의 예상 크기와 위치를 학습하고 테이블을 추출합니다. 그런 다음, 사용자 지정 레이블을 사용하여 문서의 키/값 연결 및 테이블을 학습합니다. 새 모델을 학습시킬 때 동일한 유형(동일한 구조)의 수동 레이블 지정 양식 5개를 사용하여 시작하는 것이 좋습니다. 그런 다음, 필요에 따라 레이블이 지정된 데이터를 더 추가하여 모델 정확도를 향상시킵니다. 문서 인텔리전스를 통해 감독 학습 기능을 사용하여 키 값 쌍과 테이블을 추출하도록 모델을 학습할 수 있습니다.

문서 인텔리전스 스튜디오

사용자 지정 모델을 만들려면 프로젝트 구성부터 시작합니다.

1. Studio 홈페이지의 사용자 지정 모델 카드에서 **새로 만들기** 를 선택합니다.
2. **+ 프로젝트 만들기** 명령을 사용하여 새 프로젝트 구성 마법사를 시작합니다.
3. 프로젝트 세부 정보를 입력하고, Azure 구독 및 리소스, 데이터가 포함된 Azure Blob 스토리지 컨테이너를 선택합니다.
4. 설정을 검토하고 제출하여 프로젝트를 만듭니다.



사용자 지정 모델을 만드는 동안 문서에서 데이터 컬렉션을 추출해야 할 수 있습니다. 컬렉션은 두 가지 형식 중 하나로 나타날 수 있습니다. 테이블을 시각적 패턴으로 사용:

- 지정된 필드(열) 세트에 대한 동적 또는 변수 값(행)의 개수
- 지정된 필드(열 및/또는 행) 세트에 대한 특정 값(행)의 컬렉션

[문서 인텔리전스 스튜디오: 테이블로 레이블 지정을 참조하세요.](#)

구성형 모델 만들기

① 참고

`create compose model` 작업은 레이블을 사용하여 학습된 사용자 지정 모델에만 사용할 수 있습니다. 레이블이 없는 모델을 작성하려고 하면 오류가 발생합니다.

[작성 모델 만들기](#) 작업을 사용하면 최대 100개의 학습된 모델을 단일 모델 ID에 할당할 수 있습니다. 작성된 모델 ID를 사용하여 문서를 분석하면 문서 인텔리전스는 먼저 사용

자가 제출한 양식을 분류하고, 할당된 모델 중 가장 정확하게 일치하는 모델을 선택하고, 해당 모델에 대한 결과를 반환합니다. 이 작업은 들어오는 양식이 여러 템플릿 중 하나에 속할 수 있을 때 유용합니다.

문서 인텔리전스 스튜디오

학습 프로세스가 성공적으로 완료되면 작성된 모델 빌드를 시작할 수 있습니다. 작성된 모델을 만들고 사용하는 단계는 다음과 같습니다.

- [사용자 지정 모델 ID 수집](#)
- [사용자 지정 모델 작성](#)
- [문서 분석](#)
- [작성된 모델 관리](#)

모델 ID 수집

[문서 인텔리전스 스튜디오](#) 를 사용하여 모델을 학습시키는 경우 모델 ID는 프로젝트 아래의 모델 메뉴에 있습니다.

Model ID	Model Description
8aa16866-16fe-44ca-b13a-8bfc6ad1d	
773fb140-f173-47a2-8aa9-a5fce1ceb	
4c493f98-87c3-4f6d-b0d8-3a1aab49e	

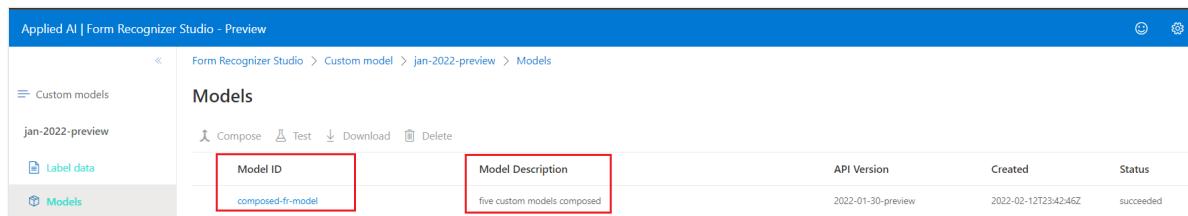
사용자 지정 모델 작성

1. 사용자 지정 모델 프로젝트를 선택합니다.
2. 프로젝트에서 `Models` 메뉴 항목을 선택합니다.
3. 결과 모델 목록에서 작성하려는 모델을 선택합니다.
4. 왼쪽 위 모서리에서 **작성 단추**를 선택합니다.
5. 팝업 창에서 새로 작성된 모델의 이름을 지정하고 **작성**을 선택합니다.
6. 작업이 완료되면 새로 작성된 모델이 목록에 표시됩니다.

7. 모델이 준비되면 **테스트** 명령을 사용하여 테스트 문서를 통해 모델의 유효성을 검사하고 결과를 관찰합니다.

문서 분석

사용자 지정 모델 **분석** 작업을 수행하려면 문서 인텔리전스에 대한 호출에서 **modelID**를 제공해야 합니다. 애플리케이션에서 **modelID** 매개 변수에 대해 작성된 모델 ID를 제공해야 합니다.



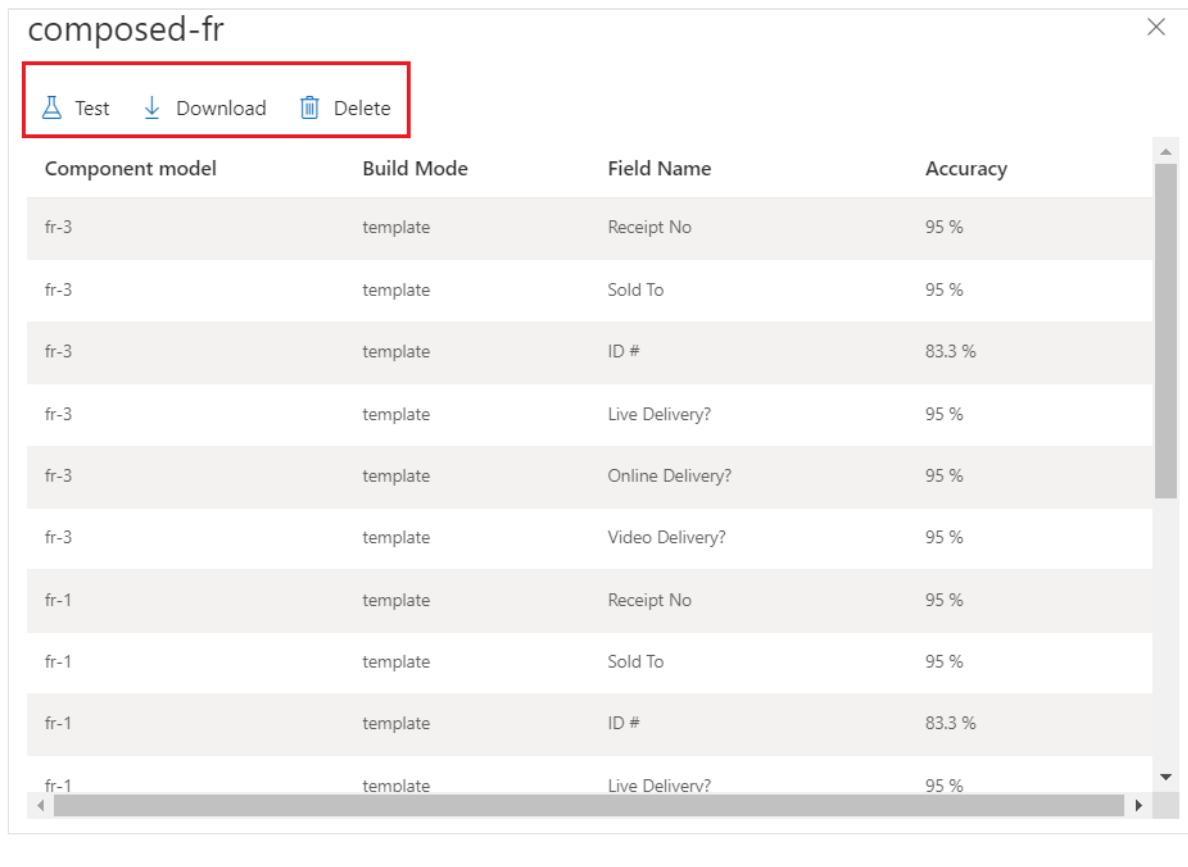
The screenshot shows the 'Models' section of the Form Recognizer Studio. The 'Model ID' column for the 'composed-fr-model' row is highlighted with a red box. The table includes columns for Model ID, Model Description, API Version, Created, and Status. The status for this model is 'succeeded'.

Model ID	Model Description	API Version	Created	Status
composed-fr-model	five custom models composed	2022-01-30-preview	2022-02-12T23:42:46Z	succeeded

작성된 모델 관리

수명 주기 동안 사용자 지정 모델을 관리할 수 있습니다.

- 새 문서를 테스트하고 유효성을 검사합니다.
- 애플리케이션에서 사용할 모델을 다운로드합니다.
- 수명 주기가 완료되면 모델을 삭제합니다.



The screenshot shows the details for the 'composed-fr' model. The 'Test', 'Download', and 'Delete' buttons are highlighted with a red box. Below the buttons is a table listing component models, build mode, field names, and accuracy. The table has columns for Component model, Build Mode, Field Name, and Accuracy.

Component model	Build Mode	Field Name	Accuracy
fr-3	template	Receipt No	95 %
fr-3	template	Sold To	95 %
fr-3	template	ID #	83.3 %
fr-3	template	Live Delivery?	95 %
fr-3	template	Online Delivery?	95 %
fr-3	template	Video Delivery?	95 %
fr-1	template	Receipt No	95 %
fr-1	template	Sold To	95 %
fr-1	template	ID #	83.3 %
fr-1	template	Live Delivery?	95 %

좋습니다! 사용자 지정 모델 및 구성형 모델을 만들어 문서 인텔리전스 프로젝트 및 애플리케이션에 사용하는 단계를 배웠습니다.

다음 단계

문서 인텔리전스 빠른 시작 중 하나를 사용해 보세요.

[문서 인텔리전스 스튜디오](#)

[REST API](#)

[C#](#)

[Java](#)

[JavaScript](#)

[Python](#)

재해 복구

아티클 • 2024. 04. 17.

이 콘텐츠의 적용 대상은 다음과 같습니다. ✓ v4.0(미리 보기) | 이전 버전: ✓ v3.1(GA) ✓ v3.0(GA) ✓ v2.1(GA)

Azure portal에서 문서 인텔리전스 리소스를 만들 때 지역을 지정합니다. 그때부터 리소스와 모든 작업이 해당 특정 Azure 서버 지역과 연결된 상태를 유지합니다. 전체 지역에 적용되는 네트워크 문제가 발생하는 것은 드물지만 불가능한 것도 아닙니다. 솔루션을 항상 사용할 수 있어야 하는 경우, 다른 지역으로 장애 조치(failover)하거나 둘 이상의 지역 간에 워크로드를 분할하도록 설계해야 합니다. 두 접근 방식 모두 서로 다른 지역에 있는 두 개 이상의 문서 인텔리전스 리소스와 지역 간에 사용자 지정 모델을 동기화하는 기능이 필요합니다.

복사 API를 사용하면 하나의 문서 인텔리전스 계정에서 사용자 지정 모델을 복사하거나 지원되는 모든 지역에 존재할 수 있는 다른 계정으로 사용자 지정 모델을 복사할 수 있으므로 이 시나리오가 가능해집니다. 이 가이드에서는 cURL과 함께 복사 REST API를 사용하는 방법을 보여 줍니다. HTTP 요청 서비스를 사용하여 요청을 실행할 수도 있습니다.

비즈니스 시나리오

앱이나 비즈니스가 문서 인텔리전스 사용자 지정 모델 사용에 의존하는 경우 모델을 다른 지역의 다른 문서 인텔리전스 계정에 복사하는 것이 좋습니다. 지역 가동 중단이 발생하는 경우 모델이 복사된 지역에 있는 모델에 액세스할 수 있습니다.

필수 조건

- 서로 다른 Azure 지역에 있는 두 개의 문서 인텔리전스 Azure 리소스입니다. 해당 리소스가 없으면 Azure portal로 이동하여 [새 문서 인텔리전스 리소스를 생성](#) 하세요.
- 문서 인텔리전스 리소스에 대한 키, 엔드포인트 URL 및 구독 ID입니다. [Azure Portal](#)의 리소스 **개요** 탭에서 이러한 값을 찾을 수 있습니다.

복사 API 개요

사용자 지정 모델을 복사하는 프로세스는 다음 단계로 구성됩니다.

- 먼저 대상 리소스, 즉 복사된 모델을 수신하는 리소스에 복사 인증 요청을 발행합니다. 복사된 모델을 수신하는 새로 생성된 대상 모델의 URL을 다시 받습니다.

- 그런 다음, 이전 호출에서 반환된 페이로드(복사 권한 부여)를 사용하여 복사할 모델을 포함하는 리소스인 원본 리소스에 복사 요청을 보냅니다. 작업 진행률을 추적하기 위해 쿼리할 수 있는 URL을 다시 받습니다.
- 작업이 성공할 때까지 원본 리소스 자격 증명을 사용하여 진행률 URL을 쿼리합니다. 또한 대상 리소스의 새 모델 ID를 쿼리하여 새 모델의 상태를 가져올 수도 있습니다.

복사 권한 부여 요청 생성

다음 HTTP 요청은 대상 리소스에서 복사 권한 부여를 가져옵니다. 대상 리소스의 엔드포인트와 키를 헤더로 입력해야 합니다.

HTTP

```
POST https://<your-resource-endpoint>/documentintelligence/documentModels/{modelId}:copyTo?api-version=2024-02-29-preview
Ocp-Apim-Subscription-Key: {<your-key>}
```

요청 본문

JSON

```
{
  "modelId": "target-model-name",
  "description": "Copied from SCUS"
}
```

복사를 시작하는 데 필요한 JSON 페이로드가 포함된 응답 본문과 함께 200 응답 코드를 수신합니다.

JSON

```
{
  "targetResourceId": "/subscriptions/{targetSub}/resourceGroups/{targetRG}/providers/Microsoft.CognitiveServices/accounts/{targetService}",
  "targetResourceRegion": "region",
  "targetModelId": "target-model-name",
  "targetModelLocation": "model path",
  "accessToken": "access token",
  "expirationDateTime": "timestamp"
}
```

복사 작업 시작

다음 HTTP 요청은 원본 리소스에 대한 복사 작업을 시작합니다. 원본 리소스의 엔드포인트와 키를 URL과 헤더로 입력해야 합니다. 요청 URL에는 복사하려는 원본 모델의 모델 ID가 포함되어 있습니다.

HTTP

```
POST https://<your-resource-
endpoint>/documentintelligence/documentModels/{modelId}:copyTo?api-
version=2024-02-29-preview
Ocp-Apim-Subscription-Key: {<your-key>}
```

요청 본문은 이전 단계의 응답입니다.

JSON

```
{
  "targetResourceId": 
    "/subscriptions/{targetSub}/resourceGroups/{targetRG}/providers/Microsoft.CognitiveServices/accounts/{targetService}",
  "targetResourceRegion": "region",
  "targetModelId": "target-model-name",
  "targetModelLocation": "model path",
  "accessToken": "access token",
  "expirationDateTime": "timestamp"
}
```

Operation-Location 헤더를 포함하는 202\Accepted 응답을 받습니다. 이 값은 작업 진행률을 추적하는 데 사용하는 URL입니다. 다음 단계를 위해 임시 위치에 복사합니다.

HTTP

```
HTTP/1.1 202 Accepted
Operation-Location: https://<your-resource-
endpoint>.cognitiveservices.azure.com/documentintelligence/operations/{opera-
tion-id}?api-version=2024-02-29-preview
```

① 참고

복사 API는 [AEK/CMK](#) 기능을 투명하게 지원합니다. 이를 위해 특별한 처리가 필요하지는 않지만 암호화되지 않은 리소스를 암호화된 리소스로 복사하는 경우 요청 헤더 `x-ms-forms-copy-degrade: true`를 포함해야 합니다. 이 헤더가 포함되지 않은 경우 복사 작업이 실패하고 `DataProtectionTransformServiceError`를 반환합니다.

복사 진행률 추적

콘솔

```
GET https://<your-resource-endpoint>.cognitiveservices.azure.com/documentintelligence/operations/{<operation-id>}?api-version=2024-02-29-preview  
Ocp-Apim-Subscription-Key: {<your-key>}
```

대상 모델 ID 추적

또한 [모델 가져오기](#) API를 통해 대상 모델을 쿼리하여 작업 상태를 추적할 수 있습니다. 복사 권한 부여 생성 요청 응답에서 복사한 대상 모델 ID를 사용하여 API를 호출합니다.

HTTP

```
GET https://<your-resource-endpoint>/documentintelligence/documentModels/{modelId}?api-version=2024-02-29-preview" -H "Ocp-Apim-Subscription-Key: <your-key>
```

응답 본문에는 모델에 대한 정보가 표시됩니다. "status" 필드에서 모델 상태를 확인합니다.

HTTP

```
HTTP/1.1 200 OK  
Content-Type: application/json; charset=utf-8  
{ "modelInfo": { "modelId": "33f4d42c-cd2f-4e74-b990-a1aeafab5a5d", "status": "ready", "createdDateTime": "2020-02-26T16:59:28Z", "lastUpdatedDateTime": "2020-02-26T16:59:34Z" }, "trainResult": { "trainingDocuments": [ { "documentName": "0.pdf", "pages": 1, "errors": [], "status": "succeeded" }, { "documentName": "1.pdf", "pages": 1, "errors": [], "status": "succeeded" }, { "documentName": "2.pdf", "pages": 1, "errors": [], "status": "succeeded" }, { "documentName": "3.pdf", "pages": 1, "errors": [], "status": "succeeded" }, { "documentName": "4.pdf", "pages": 1, "errors": [], "status": "succeeded" } ], "errors": [] }
```

cURL 샘플 코드

다음 코드 조각은 cURL을 사용하여 API 호출을 수행합니다. 또한 자신의 리소스와 관련된 모델 ID 및 구독 정보를 입력해야 합니다.

복사 권한 부여 생성

요청

Bash

```
curl -i -X POST "<your-resource-
endpoint>/documentintelligence/documentModels:authorizeCopy?api-
version=2024-02-29-preview"
-H "Content-Type: application/json"
-H "Ocp-Apim-Subscription-Key: <YOUR-KEY>"
--data-ascii "{
    'modelId': '{modelId}',
    'description': '{description}'
}"
```

성공적인 응답

JSON

```
{
    "targetResourceId": "string",
    "targetResourceRegion": "string",
    "targetModelId": "string",
    "targetModelLocation": "string",
    "accessToken": "string",
    "expirationDateTime": "string"
}
```

복사 작업 시작

요청

Bash

```
curl -i -X POST "<your-resource-
endpoint>/documentintelligence/documentModels/{modelId}:copyTo?api-
version=2024-02-29-preview"
-H "Content-Type: application/json"
-H "Ocp-Apim-Subscription-Key: <YOUR-KEY>"
--data-ascii "{
    'targetResourceId': '{targetResourceId}',
    'targetResourceRegion': '{targetResourceRegion}',
    'targetModelId': '{targetModelId}',
    'targetModelLocation': '{targetModelLocation}',
    'accessToken': '{accessToken}',
    'expirationDateTime': '{expirationDateTime}'
}"
```

성공적인 응답

HTTP

HTTP/1.1 202 Accepted

Operation-Location: <https://<your-resource-endpoint>.cognitiveservices.azure.com/documentintelligence/operations/{operation-id}?api-version=2024-02-29-preview>

복사 작업 진행률 추적

작업 가져오기 API를 사용하면 문서 인텔리전스 리소스와 관련된 모든 문서 모델 작업(성공, 진행 중 또는 실패)을 나열할 수 있습니다. 작업 정보는 24시간 동안만 유지됩니다. 다음은 반환할 수 있는 작업(operationId)의 목록입니다.

- documentModelBuild
- documentModelCompose
- documentModelCopyTo

대상 모델 ID 추적

작업이 성공한 경우 **getModel**(단일 모델 가져오기) 또는 **GetModels**(모델 목록 가져오기) API를 사용하여 문서 모델에 액세스할 수 있습니다.

일반적인 오류 코드 메시지

[+] 테이블 확장

오류	해결
400 / 잘못된 요청 ("code: " "1002")	유효성 검사 오류 또는 잘못된 형식의 복사 요청을 나타냅니다. 일반적인 문제는 다음과 같습니다. a) 잘못되었거나 수정된 <code>copyAuthorization</code> 페이로드. b) <code>expirationDateTimeTicks</code> 토큰의 만료된 값(<code>copyAuthorization</code> 페이로드는 24시간 동안 유효). c) 잘못되었거나 지원되지 않는 <code>targetResourceRegion</code> . d) 잘못되었거나 형식이 잘못된 <code>targetResourceId</code> 문자열.
승인 요청이 누락되거나 잘못되어 승인이 실패했습니다.	<code>copyAuthorization</code> 페이로드 또는 콘텐츠가 <code>copyAuthorization</code> API에서 수 정되면 발생합니다. 페이로드가 이전 <code>copyAuthorization</code> 호출에서 반환된 것과 똑같은 내용인지 확인합니다.
인증 메타데이터를 검색할 수 없습니다.	<code>copyAuthorization</code> 페이로드가 복사 요청에 재사용되고 있음을 나타냅니다. 성공한 복사 요청은 동일한 <code>copyAuthorization</code> 페이로드를 사용하는 추가 요청을 허용하지 않습니다. 별도의 오류가 발생하고 나중에 동일한 인증 페이로드로 복사를 다시 시도하면 이 오류가 발생합니다. 해결 방법은 새

오류	해결
	<p><code>copyAuthorization</code> 페이로드를 생성한 다음, 복사 요청을 다시 실행하는 것입니다.</p>
보안 수준이 낮은 데 이터 보호 체계로 다른 업그레이드되므로 데이터 전송 요청이 허용되지 않습니다.	<p><code>AEK</code> 가 설정된 리소스와 <code>AEK</code> 가 설정되지 않은 리소스 간에 복사하는 경우에 발생합니다. 암호화된 모델을 암호화되지 않은 모델로 대상에 복사하는 것을 허용하려면 복사 요청이 있는 <code>x-ms-forms-copy-degrade: true</code> 헤더를 지정합니다.</p>
"ID의 인지 리소스에 대한 정보를 가져올 수 없습니다."	<p><code>targetResourceId</code>로 표시된 Azure 리소스가 유효한 Cognitive 리소스가 아니거나 존재하지 않음을 나타냅니다. 이 문제를 해결하려면 복사 요청을 확인하고 다시 실행하세요.</p> <p>리소스가 올바르며 지정된 지역(예: <code>westus2</code>)에 있는지 확인</p>

다음 단계

이 가이드에서는 Copy API를 사용하여 사용자 지정 모델을 보조 문서 인텔리전스 리소스에 백업하는 방법을 배웠습니다. 다음으로 API 참조 문서를 탐색하여 문서 인텔리전스로 수행할 수 있는 다른 작업을 알아보세요.

- [REST API 참조 설명서](#)

Azure AI 서비스 가상 네트워크 구성

아티클 • 2024. 04. 05.

Azure AI 서비스는 계층화된 보안 모델을 제공합니다. 이 모델을 사용하여 Azure AI 서비스 계정을 특정 네트워크 하위 집합으로 보호할 수 있습니다. 네트워크 규칙이 구성되면 지정된 네트워크 세트를 통해 데이터를 요청하는 애플리케이션만 계정에 액세스할 수 있습니다. 지정된 IP 주소, IP 범위 또는 [Azure Virtual Networks](#)의 서브넷 목록에서 시작하는 요청만을 허용하는 [요청 필터링](#)을 통해 리소스에 대한 액세스를 제한할 수 있습니다.

네트워크 규칙이 적용될 때 Azure AI 서비스에 액세스하는 애플리케이션에는 권한 부여가 필요합니다. 권한 부여는 [Microsoft Entra ID](#) 자격 증명 또는 유효한 API 키를 사용하여 지원됩니다.

ⓘ 중요

Azure AI 서비스 계정에 대한 방화벽 규칙을 설정하면 기본적으로 데이터에 대해 들어오는 요청이 차단됩니다. 요청을 허용하려면 다음 조건 중 하나를 충족해야 합니다.

- 요청은 대상 Azure AI 서비스 계정의 허용된 서브넷 목록에 있는 Azure VNet 내에서 작동하는 서비스로부터 시작됩니다. 가상 네트워크에서 시작된 요청의 엔드포인트는 Azure AI 서비스 계정의 [사용자 지정 하위 도메인](#)으로 설정되어야 합니다.
- 요청은 허용되는 IP 주소 목록에서 시작됩니다.

차단되는 요청에는 다른 Azure 서비스로부터의 요청, Azure Portal의 요청, 로깅 및 메트릭 서비스로부터의 요청이 포함됩니다.

ⓘ 참고

Azure Az PowerShell 모듈을 사용하여 Azure와 상호 작용하는 것이 좋습니다. 시작 하려면 [Azure PowerShell 설치](#)를 참조하세요. Az PowerShell 모듈로 마이그레이션 하는 방법에 대한 자세한 내용은 [Azure PowerShell을 AzureRM에서 Azure로 마이그레이션](#)을 참조하세요.

시나리오

Azure AI 서비스 리소스를 보호하려면 먼저 인터넷 트래픽을 비롯한 모든 네트워크의 트래픽에 대한 액세스를 거부하도록 규칙을 구성해야 합니다. 그런 다음, 특정 가상 네트워크에 대한 접근을 허용하는 규칙을 설정합니다.

크의 트래픽에 대한 액세스를 허가하는 규칙을 구성합니다. 이 구성을 사용하면 애플리케이션에 대한 보안 네트워크 경계를 구축할 수 있습니다. 또한 특정 공용 인터넷 IP 주소 범위의 트래픽에 대한 액세스를 허가하도록 규칙을 구성하고 특정 인터넷 또는 온-프레미스 클라이언트의 연결을 사용하도록 설정할 수도 있습니다.

네트워크 규칙은 REST 및 WebSocket을 포함하여 Azure AI 서비스에 대한 모든 네트워크 프로토콜에 적용됩니다. Azure 테스트 콘솔 등의 도구를 사용하여 데이터에 액세스하려면 명시적 네트워크 규칙을 구성해야 합니다. 네트워크 규칙을 기준 Azure AI 서비스 리소스에 적용하거나 새 Azure AI 서비스 리소스를 만들 때 적용할 수 있습니다. 네트워크 규칙이 적용된 이후에는 모든 요청에 적용됩니다.

지원되는 지역 및 서비스 제공 사항

가상 네트워크는 [Azure AI 서비스를 사용할 수 있는 지역](#)에서 지원됩니다. Azure AI 서비스는 네트워크 규칙 구성에 대한 서비스 태그를 지원합니다. 여기에 나열된 서비스는 `CognitiveServicesManagement` 서비스 태그에 포함되어 있습니다.

- ✓ Anomaly Detector
- ✓ Azure OpenAI
- ✓ Content Moderator
- ✓ Custom Vision
- ✓ Face
- ✓ 언어 이해(LUIS)
- ✓ Personalizer
- ✓ Speech Service
- ✓ 언어
- ✓ QnA Maker
- ✓ Translator

① 참고

Azure OpenAI, LUIS, Speech Services 또는 언어 서비스를 사용하는 경우 `CognitiveServicesManagement` 태그를 통해서는 SDK 또는 REST API를 사용하는 서비스만 사용할 수 있습니다. 가상 네트워크에서 Azure OpenAI 스튜디오, LUIS 포털, Speech Studio 또는 Language Studio에 액세스하고 이를 사용하려면 다음 태그를 사용해야 합니다.

- `AzureActiveDirectory`
- `AzureFrontDoor.Frontend`
- `AzureResourceManager`
- `CognitiveServicesManagement`

- CognitiveServicesFrontEnd
- Storage (Speech Studio에만 해당)

Azure AI Studio 구성에 대한 자세한 내용은 [Azure AI Studio 설명서](#)를 참조하세요.

기본 네트워크 액세스 규칙 변경

기본적으로 Azure AI 서비스 리소스는 네트워크에 있는 클라이언트로부터의 연결을 허용합니다. 선택한 네트워크에 대한 액세스를 제한하려면 먼저 기본 동작을 변경해야 합니다.

⚠ 경고

네트워크 규칙을 변경하면 Azure AI 서비스에 연결하는 애플리케이션의 기능에 영향을 미칠 수 있습니다. 기본 네트워크 규칙을 거부로 설정하면 액세스를 허용하는 특정 네트워크 규칙이 적용되지 않는 한 데이터에 대한 모든 액세스가 차단됩니다.

액세스를 거부하도록 기본 규칙을 변경하기 전에 네트워크 규칙을 사용하여 허용된 모든 네트워크에 대한 액세스를 허가해야 합니다. 온-프레미스 네트워크에 대한 IP 주소를 나열하도록 허용하는 경우 온-프레미스 네트워크에서 사용 가능한 모든 나가는 공용 IP 주소를 추가해야 합니다.

기본 네트워크 액세스 규칙 관리

Azure Portal, PowerShell 또는 Azure CLI를 통해 Azure AI 서비스 리소스에 대한 기본 네트워크 액세스 규칙을 관리할 수 있습니다.

Azure Portal

1. 보안을 유지하려는 Azure AI 서비스 리소스로 이동합니다.
2. 리소스 관리를 선택하여 확장한 다음 **네트워킹**을 선택합니다.

The screenshot shows the Azure portal interface for managing networking settings. On the left, there's a sidebar with various options like Overview, Activity log, Access control (IAM), Tags, Diagnose and solve problems, Resource Management, Keys and Endpoint, Encryption, Pricing tier, Networking (which is highlighted with a red box), Identity, Cost analysis, Properties, Locks, Monitoring, and Automation. The main content area is titled 'Firewalls and virtual networks' and shows a summary of network security settings. It includes sections for 'Virtual networks' (with buttons to 'Add existing virtual network' or 'Add new virtual network'), 'Firewall' (with a checkbox for 'Add your client IP address'), and 'Address range' (with a text input field for 'IP address or CIDR'). At the top, there are 'Save', 'Discard', and 'Refresh' buttons. A note at the top states: 'Access control settings allowing access to Azure AI services account will remain in effect for up to three minutes after saving updated settings restricting access.' Below the main content, there are tabs for 'Virtual Network', 'Subnet', 'Address range', 'Endpoint Status', 'Resource group', and 'Subscription'. The 'Address range' tab is currently selected.

3. 기본으로 액세스를 거부하려면 방화벽 및 가상 네트워크에서 선택한 네트워크 및 프라이빗 엔드포인트를 선택합니다.

구성된 가상 네트워크 또는 주소 범위를 같이 사용하지 않고 이 설정만 단독으로 사용하면 모든 액세스가 사실상 거부됩니다. 모든 액세스가 거부되면 Azure AI 서비스 리소스를 사용하려는 요청이 허용되지 않습니다. 계속 Azure Portal, Azure PowerShell 또는 Azure CLI를 사용하여 Azure AI 서비스 리소스를 구성할 수 있습니다.

4. 모든 네트워크에서 트래픽을 허용하려면 모든 네트워크를 선택합니다.

This screenshot is similar to the previous one but shows a different configuration. Under 'Allow access from', the radio button for 'All networks' is selected, while 'Selected Networks and Private Endpoints' is unselected. A note below says 'All networks, including the internet, can access this resource.' The rest of the interface is identical to the first screenshot, including the sidebar with the 'Networking' item highlighted.

5. 저장을 선택하여 변경 내용을 적용합니다.

가상 네트워크의 액세스 허가

특정 서브넷에서만 액세스를 허용하도록 Azure AI 서비스 리소스를 구성할 수 있습니다. 허용된 서브넷은 동일하거나 다른 구독 내의 가상 네트워크에 속할 수 있습니다. 다른 구독은 다른 Microsoft Entra 테넌트에 속할 수 있습니다. 서브넷이 다른 구독에 속하는 경우 Microsoft.CognitiveServices 리소스 공급자도 해당 구독에 등록해야 합니다.

가상 네트워크 내에서 Azure AI 서비스에 서비스 엔드포인트를 사용하도록 설정합니다. 서비스 엔드포인트는 가상 네트워크의 트래픽을 Azure AI 서비스에 대한 최적의 경로를 통해 라우팅합니다. 자세한 내용은 [가상 네트워크 서비스 엔드포인트](#)를 참조하세요.

서브넷 및 가상 네트워크의 ID 또한 각 요청과 함께 전송됩니다. 그러면 관리자가 가상 네트워크의 특정 서브넷 요청을 허용하는 Azure AI 서비스 리소스에 대한 네트워크 규칙을 구성할 수 있습니다. 이러한 네트워크 규칙을 통해 액세스가 허가된 클라이언트는 데이터에 액세스하기 위해 Azure AI 서비스 리소스의 인증 요구 사항을 계속 충족해야 합니다.

각 Azure AI 서비스 리소스는 IP 네트워크 규칙과 결합될 수 있는 최대 100개의 가상 네트워크 규칙을 지원합니다. 자세한 내용은 이 문서의 뒷부분에 나오는 [인터넷 IP 범위에서 액세스 권한 부여](#)를 참조하세요.

필요한 권한 설정

가상 네트워크 규칙을 Azure AI 서비스 리소스에 적용하려면 추가할 서브넷에 대한 적절한 권한이 있어야 합니다. 필요한 권한은 기본 기여자 역할 또는 *Cognitive Services* 기여자 역할입니다. 필요한 사용 권한을 사용자 지정 역할 정의에 추가할 수도 있습니다.

액세스 권한이 허용된 Azure AI 서비스 리소스 및 가상 네트워크는 다른 Microsoft Entra 테넌트에 속하는 구독을 포함한 다른 구독에 있을 수 있습니다.

① 참고

다른 Microsoft Entra 테넌트의 일부인 가상 네트워크의 서브넷에 대한 액세스 권한을 부여하는 규칙 구성은 현재 PowerShell, Azure CLI 및 REST API를 통해서만 지원됩니다. Azure Portal에서 이러한 규칙을 볼 수 있지만 구성할 수는 없습니다.

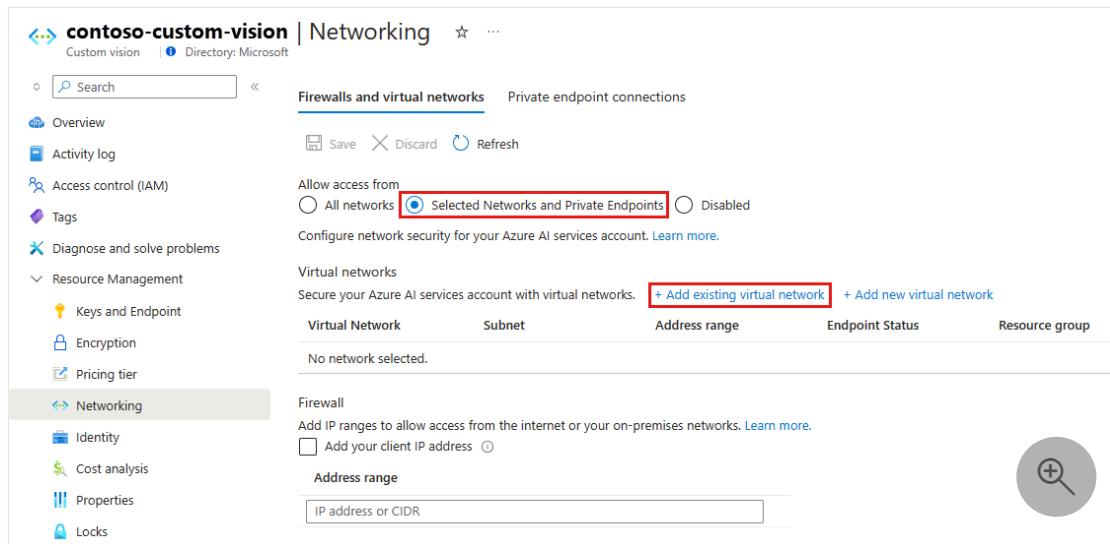
가상 네트워크 규칙 구성

Azure Portal, PowerShell 또는 Azure CLI를 통해 Azure AI 서비스 리소스에 대한 가상 네트워크 규칙을 관리할 수 있습니다.

Azure Portal

기존 네트워크 규칙을 이용해 가상 네트워크에 액세스 권한을 부여하려면 다음을 수행합니다.

1. 보안을 유지하려는 Azure AI 서비스 리소스로 이동합니다.
2. 리소스 관리를 선택하여 확장한 다음 **네트워킹**을 선택합니다.
3. 선택한 **네트워크 및 프라이빗 엔드포인트**를 선택했는지 확인합니다.
4. 다음에서 **액세스 허용**에서 기존 가상 네트워크 추가를 선택합니다.



The screenshot shows the Networking page for the 'contoso-custom-vision' resource group. On the left, there's a sidebar with various service icons. The 'Networking' icon is highlighted with a grey background. The main area has a title bar 'contoso-custom-vision | Networking'. Below it, there are tabs for 'Firewalls and virtual networks' (which is selected) and 'Private endpoint connections'. A search bar and some save/discard/refresh buttons are at the top. Under 'Allow access from', there are three radio buttons: 'All networks' (unchecked), 'Selected Networks and Private Endpoints' (checked and highlighted with a red box), and 'Disabled' (unchecked). A note below says 'Configure network security for your Azure AI services account. Learn more.' Under 'Virtual networks', there's a table header with columns 'Virtual Network', 'Subnet', 'Address range', 'Endpoint Status', and 'Resource group'. A note says 'Secure your Azure AI services account with virtual networks.' Below the table, there's a 'Firewall' section with a note 'Add IP ranges to allow access from the internet or your on-premises networks. Learn more.' and a checkbox 'Add your client IP address'. At the bottom right is a magnifying glass icon inside a circle.

5. 가상 네트워크 및 서브넷 옵션을 선택한 다음, 사용을 선택합니다.

Add networks

X

Subscription *

Contoso Subscription

Virtual networks *

contoso-rg

Subnets *

default (Service endpoint required)

i The following networks don't have service endpoints enabled for 'Microsoft.CognitiveServices'. Enabling access will take up to 15 minutes to complete. After starting this operation, it is safe to leave and return later if you do not wish to wait.

Virtual network	Service endpoint status	
contoso-rg	Not enabled	...
default	Not enabled	...

Enable

ⓘ 참고

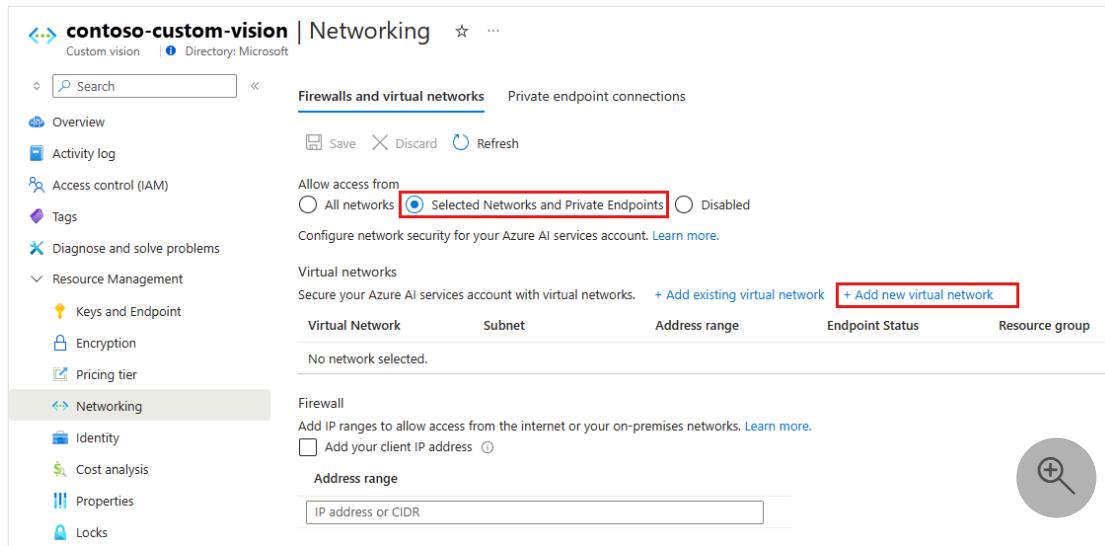
이전에 Azure AI 서비스에 대한 서비스 엔드포인트가 선택한 가상 네트워크 및 서브넷에 대해 구성되지 않은 경우 이 작업의 일환으로 구성할 수 있습니다.

현재는 동일한 Microsoft Entra 테넌트에 속한 가상 네트워크만 규칙을 만드는 동안 선택할 수 있습니다. 다른 테넌트에 속하는 가상 네트워크 내의 서브넷에 대한 액세스 권한을 부여하려면 PowerShell이나 Azure CLI 또는 REST API를 사용하세요.

6. 저장을 선택하여 변경 내용을 적용합니다.

새 가상 네트워크를 만들어 액세스 권한을 부여하려면 다음을 수행합니다.

1. 이전 절차와 동일한 페이지에서 새 가상 네트워크 추가를 선택합니다.



The screenshot shows the 'Networking' page for the 'contoso-custom-vision' resource in the Azure portal. The left sidebar has 'Networking' selected. The main area shows 'Firewalls and virtual networks' settings. Under 'Allow access from', the 'Selected Networks and Private Endpoints' radio button is selected. In the 'Virtual networks' section, there is a table with columns: Virtual Network, Subnet, Address range, Endpoint Status, and Resource group. Below the table, it says 'No network selected.' To the right, there is a 'Firewall' section with an 'Address range' input field containing 'IP address or CIDR' and a search icon. A red box highlights the '+ Add new virtual network' button.

2. 새 가상 네트워크를 만드는 데 필요한 정보를 입력하고 만들기를 선택합니다.

Create virtual network

* Name
widgets-vnet

* Address space ⓘ
10.1.0.0/16
10.1.0.0 - 10.1.255.255 (65536 addresses)

* Subscription
widgets-subscription

* Resource group
widgets-resource-group
[Create new](#)

* Location
(US) West US 2

Subnet

* Name
default

* Address range ⓘ
10.1.0.0/24
10.1.0.0 - 10.1.0.255 (256 addresses)

DDoS protection ⓘ
 Basic Standard

Service endpoint ⓘ
Microsoft.CognitiveServices

Firewall ⓘ
 Disabled Enabled

Create

3. 저장을 선택하여 변경 내용을 적용합니다.

가상 네트워크나 서브넷 규칙을 제거하려면 다음을 수행합니다.

1. 이전 절차와 동일한 페이지에서 ...(**추가 옵션**)을 선택하여 가상 네트워크와 서브넷용 바로 가기 메뉴를 열고 **제거**를 선택합니다.

Firewalls and virtual networks Private endpoint connections

Save Discard Refresh

Allow access from
All networks Selected Networks and Private Endpoints Disabled

Configure network security for your Azure AI services account. [Learn more.](#)

Virtual networks
Secure your Azure AI services account with virtual networks. + Add existing virtual network + Add new virtual network

Virtual Network	Subnet	Address range	Endpoint Status	Resource group	Subscription
contoso-01-vnet	1			contoso-rg	Remove ...

Firewall
Add IP ranges to allow access from the internet or your on-premises networks. [Learn more.](#)

Add your client IP address [?](#)

Address range
IP address or CIDR

2. 저장을 선택하여 변경 내용을 적용합니다.

① 중요

반드시 [기본 규칙](#)을 거부로 설정해야 합니다. 그렇지 않으면 네트워크 규칙이 적용되지 않습니다.

인터넷 IP 범위의 액세스 허가

특정 공용 인터넷 IP 주소 범위에서 액세스할 수 있도록 Azure AI 서비스 리소스를 구성할 수 있습니다. 이 구성은 특정 서비스와 온-프레미스 네트워크에 대한 액세스 권한을 부여하고 일반 인터넷 트래픽을 효과적으로 차단합니다.

192.168.0.0/16 양식의 [CIDR 형식\(RFC 4632\)](#)을 사용하거나 192.168.0.1 같은 개별 IP 주소로 허용된 인터넷 주소 범위를 지정할 수 있습니다.

💡 팁

접두사 크기가 /31 또는 /32인 작은 주소 범위는 지원하지 않습니다. 해당 범위는 개별 IP 주소 규칙을 사용하여 구성합니다.

IP 네트워크 규칙은 [공용 인터넷](#) IP 주소에 대해서만 허용됩니다. 프라이빗 네트워크에 예약된 IP 주소 범위는 IP 규칙에서 허용되지 않습니다. 사설망에는 10.*, 172.16.* - 172.31.* 및 192.168.*로 시작하는 주소가 포함됩니다. 자세한 내용은 [프라이빗 주소 공간\(RFC 1918\)](#)을 참조하세요.

현재는 IPv4 주소만 지원합니다. 각 Azure AI 서비스 리소스는 [가상 네트워크 규칙](#)과 결합될 수 있는 최대 100개의 IP 네트워크 규칙을 지원합니다.

온-프레미스 네트워크에서의 액세스 구성

IP 네트워크 규칙을 사용하여 온-프레미스 네트워크에서 Azure AI 서비스 리소스로의 액세스를 허가하려면 네트워크에서 사용되는 인터넷 연결 IP 주소를 식별합니다. 네트워크 관리자에게 도움을 요청합니다.

공용 피어링 또는 Microsoft 피어링을 위해 Azure ExpressRoute 온-프레미스를 사용하는 경우 NAT IP 주소를 식별해야 합니다. 자세한 내용은 [Azure ExpressRoute란?](#)을 참조하세요.

공용 피어링의 경우 기본적으로 각 ExpressRoute 회로에서 두 개의 NAT IP 주소를 사용합니다. 각각은 트래픽이 Microsoft Azure 네트워크 백본으로 들어갈 때 Azure 서비스 트래픽에 적용됩니다. Microsoft 피어링의 경우 사용되는 NAT IP 주소는 고객이 제공하거나 서비스 공급자가 제공합니다. 서비스 리소스에 대한 액세스를 허용하려면 리소스 IP 방화벽 설정에서 이러한 공용 IP 주소를 허용해야 합니다.

공용 피어링 ExpressRoute 회로 IP 주소를 찾으려면 Azure Portal을 통해 [ExpressRoute에서 지원 티켓을 엽니다](#). 자세한 내용은 [Azure 공용 피어링에 대한 NAT 요구 사항](#)을 참조하세요.

IP 네트워크 규칙 관리

Azure Portal, PowerShell 또는 Azure CLI를 통해 Azure AI 서비스 리소스에 대한 IP 네트워크 규칙을 관리할 수 있습니다.

Azure Portal

- 보안을 유지하려는 Azure AI 서비스 리소스로 이동합니다.
- 리소스 관리를 선택하여 확장한 다음 **네트워킹**을 선택합니다.
- 선택한 **네트워크 및 프라이빗 엔드포인트**를 선택했는지 확인합니다.
- 방화벽 및 가상 네트워크에서 주소 범위** 옵션을 찾습니다. 인터넷 IP 범위에 대한 액세스 권한을 부여하려면 [CIDR 형식](#)으로 된 IP 주소나 주소 범위를 입력합니다. 유효한 공용 IP(예약되지 않음) 주소만 허용됩니다.

Allow access from

All networks Selected Networks and Private Endpoints Disabled

Configure network security for your Azure AI services account. [Learn more](#).

Virtual networks

Secure your Azure AI services account with virtual networks. [+ Add existing virtual network](#) [+ Add new virtual network](#)

Virtual Network	Subnet	Address range	Endpoint Status	Resource group
No network selected.				

Firewall

Add IP ranges to allow access from the internet or your on-premises networks. [Learn more](#).

Add your client IP address ⓘ

Address range

IP address or CIDR

IP 네트워크 규칙을 제거하려면 주소 범위 옆에 있는 휴지통 ✖ 아이콘을 선택합니다.

5. 저장을 선택하여 변경 내용을 적용합니다.

ⓘ 중요

반드시 [기본 규칙을 거부로 설정해야 합니다](#). 그렇지 않으면 네트워크 규칙이 적용되지 않습니다.

프라이빗 엔드포인트 사용

Azure AI 서비스 리소스에서 [프라이빗 엔드포인트](#)를 사용하면 가상 네트워크의 클라이언트가 [Azure Private Link](#)를 통해 안전하게 데이터에 액세스하도록 할 수 있습니다. 프라이빗 엔드포인트는 Azure AI 서비스 리소스에 가상 네트워크 주소 공간의 IP 주소를 사용합니다. 가상 네트워크의 클라이언트와 리소스 간의 네트워크 트래픽이 Microsoft Azure 백본 네트워크에서 가상 네트워크와 프라이빗 링크를 통과하며 공용 인터넷에서의 노출을 제거합니다.

Azure AI 서비스 리소스의 프라이빗 엔드포인트를 사용하면 다음과 같은 작업을 할 수 있습니다.

- Azure AI 서비스에 대한 퍼블릭 엔드포인트의 모든 연결을 차단하도록 방화벽을 구성하여 Azure AI 서비스를 보호합니다.
- 가상 네트워크에서의 데이터 반출을 차단하여 가상 네트워크 보안을 강화합니다.

- Azure VPN 게이트웨이 또는 ExpressRoute의 개인 피어링을 사용하여 가상 네트워크에 연결하는 온-프레미스 네트워크에서 Azure AI 서비스 리소스로 안전하게 연결합니다.

프라이빗 엔드포인트 이해하기

프라이빗 엔드포인트는 [가상 네트워크](#) 내부의 Azure 리소스를 위한 특별한 네트워크 인터페이스입니다. Azure AI 서비스 리소스에 대한 프라이빗 엔드포인트를 만들면 가상 네트워크 내 클라이언트와 리소스 간에 보안 연결을 제공합니다. 프라이빗 엔드포인트에는 가상 네트워크의 IP 주소 범위에서 IP 주소가 할당됩니다. 프라이빗 엔드포인트와 Azure AI 서비스 간의 연결은 보안 프라이빗 링크를 사용합니다.

가상 네트워크 내의 애플리케이션은 프라이빗 엔드포인트를 통해 서비스에 원활하게 연결할 수 있습니다. 연결은 달리 사용할 수도 있는 동일한 연결 문자열 및 권한 부여 메커니즘을 사용합니다. 별도의 엔드포인트가 필요한 Speech Services는 예외입니다. 자세한 내용은 이 문서의 [Speech Services를 사용하는 프라이빗 엔드포인트](#)를 참조하세요. 프라이빗 엔드포인트는 REST를 포함하여 Azure AI 서비스 리소스에서 지원하는 모든 프로토콜과 함께 사용할 수 있습니다.

서비스 엔드포인트를 사용하는 서브넷에서 프라이빗 엔드포인트를 만들 수 있습니다. 서브넷의 클라이언트는 서비스 엔드포인트를 사용해 Azure AI 서비스 리소스에 액세스하는 동안 프라이빗 엔드포인트를 사용하여 다른 Azure AI 서비스 리소스에 연결할 수 있습니다. 자세한 내용은 [가상 네트워크 서비스 엔드포인트](#)를 참조하세요.

가상 네트워크에서 Azure AI 서비스 리소스에 대한 프라이빗 엔드포인트를 만들 때 Azure는 Azure AI 서비스 리소스 소유자에게 승인 동의 요청을 보냅니다. 프라이빗 엔드포인트 만들기를 요청한 사용자가 리소스의 소유자인 경우 이 동의 요청이 자동으로 승인됩니다.

Azure AI 서비스 리소스 소유자는 [Azure Portal](#)의 Azure AI 서비스 리소스 관련 [프라이빗 엔드포인트](#) 연결 탭을 통해 동의 요청 및 프라이빗 엔드포인트를 관리할 수 있습니다.

프라이빗 엔드포인트 지정

프라이빗 엔드포인트를 만들 때 연결할 Azure AI 서비스 리소스를 지정합니다. 프라이빗 엔드포인트를 만드는 방법에 대한 자세한 내용은 다음을 참조하세요.

- [Azure Portal](#)을 사용하여 프라이빗 엔드포인트 만들기
- [Azure PowerShell](#)을 사용하여 프라이빗 엔드포인트 만들기
- [Azure CLI](#)를 사용하여 프라이빗 엔드포인트 만들기.

프라이빗 엔드포인트에 연결

① 참고

Azure OpenAI Service는 다른 Azure AI 서비스와는 다른 프라이빗 DNS 영역 및 공용 DNS 영역 전달자를 사용합니다. 올바른 영역 및 전달자 이름은 [Azure 서비스 DNS 영역 구성](#)을 참조하세요.

프라이빗 엔드포인트를 사용하는 가상 네트워크의 클라이언트는 퍼블릭 엔드포인트에 연결하는 클라이언트와 동일한 Azure AI 서비스 리소스 연결 문자열을 사용합니다. 별도의 엔드포인트가 필요한 Speech Services는 예외입니다. 자세한 내용은 이 문서의 [Speech Services를 이용해 프라이빗 엔드포인트 사용하기](#)를 참조하세요. DNS 확인은 자동으로 프라이빗 링크를 통해 가상 네트워크에서 Azure AI 서비스 리소스로의 연결을 라우팅합니다.

기본적으로 Azure는 프라이빗 엔드포인트에 필요한 업데이트를 사용하여 가상 네트워크에 연결된 [프라이빗 DNS 영역](#)을 만듭니다. 자체 DNS 서버를 사용하는 경우 DNS 구성을 추가로 변경해야 할 수 있습니다. 프라이빗 엔드포인트에 필요할 수도 있는 업데이트는 이 문서의 [프라이빗 엔드포인트에 대한 DNS 변경 내용 적용](#)을 참조하세요.

Speech Service를 이용해 프라이빗 엔드포인트 사용하기

프라이빗 엔드포인트를 통해 [Speech service 사용하기](#)를 참조하세요.

프라이빗 엔드포인트에 대한 DNS 변경 내용 적용

프라이빗 엔드포인트를 만들 때 Azure AI 서비스 리소스에 대한 DNS `CNAME` 리소스 레코드는 `privatelink` 접두사가 있는 하위 도메인의 별칭으로 업데이트됩니다. 또한, 기본적으로 Azure는 프라이빗 엔드포인트에 대한 DNS A 리소스 레코드를 사용하여 `privatelink` 하위 도메인에 해당하는 프라이빗 DNS 영역을 만듭니다. 자세한 내용은 [Azure 프라이빗 DNS란?](#)을 참조하세요.

프라이빗 엔드포인트를 사용하여 가상 네트워크 외부에서 엔드포인트 URL을 확인하는 경우 Azure AI 서비스 리소스의 퍼블릭 엔드포인트로 확인됩니다. 프라이빗 엔드포인트를 호스트하는 가상 네트워크에서 확인하는 경우 엔드포인트 URL은 프라이빗 엔드포인트의 IP 주소로 확인됩니다.

이 접근 방식을 사용하면 프라이빗 엔드포인트를 호스트하는 가상 네트워크의 클라이언트와 가상 네트워크 외부의 클라이언트에 동일한 연결 문자열을 사용하여 Azure AI 서비스 리소스에 액세스할 수 있습니다.

네트워크에서 사용자 지정 DNS 서버를 사용하는 경우 클라이언트는 프라이빗 엔드포인트 IP 주소에 대한 Azure AI 서비스 리소스 엔드포인트의 FQDN(정규화된 도메인 이름)을

확인할 수 있어야 합니다. 프라이빗 링크 하위 도메인을 가상 네트워크의 개인 DNS 영역에 위임하도록 DNS 서버를 구성합니다.

💡 팁

사용자 지정 또는 온-프레미스 DNS 서버를 사용하는 경우 `privatelink` 하위 도메인의 Azure AI 서비스 리소스 이름을 프라이빗 엔드포인트 IP 주소로 확인하도록 DNS 서버를 구성해야 합니다. `privatelink` 하위 도메인을 가상 네트워크의 프라이빗 DNS 영역에 위임합니다. 또는 DNS 서버에서 DNS 영역을 구성하고 DNS A 레코드를 추가합니다.

프라이빗 엔드포인트를 지원하기 위해 자체 DNS 서버를 구성하는 방법에 대한 자세한 내용은 다음 리소스를 참조하세요.

- 자체 DNS 서버를 사용하는 이름 확인
- DNS 구성

신뢰할 수 있는 Azure 서비스에 Azure OpenAI에 대한 액세스 권한 부여

다른 앱의 네트워크 규칙을 유지하면서 신뢰할 수 있는 Azure 서비스의 하위 집합에 Azure OpenAI에 대한 액세스 권한을 부여할 수 있습니다. 그러면 이러한 신뢰할 수 있는 서비스는 관리 ID를 사용하여 Azure OpenAI 서비스를 인증합니다. 다음 표에는 해당 서비스의 관리 ID에 적절한 역할 할당이 있는 경우 Azure OpenAI에 액세스할 수 있는 서비스가 나와 있습니다.

[+] 테이블 확장

서비스	리소스 공급자 이름
Azure AI 서비스	<code>Microsoft.CognitiveServices</code>
Azure Machine Learning	<code>Microsoft.MachineLearningServices</code>
Azure AI 검색	<code>Microsoft.Search</code>

REST API를 사용하여 네트워크 규칙 예외를 만들면 신뢰할 수 있는 Azure 서비스에 네트워킹 액세스 권한을 부여할 수 있습니다.

Bash

```
accessToken=$(az account get-access-token --resource
```

```

https://management.azure.com --query "accessToken" --output tsv
$rid="/subscriptions/<your subscription id>/resourceGroups/<your resource
group>/providers/Microsoft.CognitiveServices/accounts/<your Azure AI
resource name>"

curl -i -X PATCH https://management.azure.com$rid?api-version=2023-10-01-
preview \
-H "Content-Type: application/json" \
-H "Authorization: Bearer $accessToken" \
-d \
'
{
  "properties": {
    {
      "networkAcls": {
        "bypass": "AzureServices"
      }
    }
  }
'

```

① 참고

신뢰할 수 있는 서비스 기능은 위에서 설명한 명령줄에서만 사용할 수 있으며 Azure Portal에서는 수행할 수 없습니다.

예외를 철회하려면 `networkAcls.bypass` 를 `None` 으로 설정합니다.

Azure Portal에서 신뢰할 수 있는 서비스를 사용하도록 설정했는지 확인하려면

1. Azure OpenAI 리소스 개요 페이지에서 JSON 보기 사용

The screenshot shows the Azure OpenAI service overview page for a resource named 'my-openai-resource'. The left sidebar includes links for Activity log, Access control (IAM), Tags, Diagnose and solve problems, Resource visualizer, Resource Management (Keys and Endpoint, Model deployments, Encryption), Overview, and Essentials. The main content area displays resource details under 'Essentials': Resource group (move) : my-resource-group, Status : Active, Location : East US 2, Subscription (move) : My_subscription, Subscription ID : abcd1234, and Tags (edit) : Add tags. To the right of these details are API Kind : OpenAI, Pricing tier : Standard, Endpoints : Click here to view endpoints, and Manage keys : Click here to manage keys. At the bottom of the main content area, there is a call-to-action: 'Build your own secure copilot and generative AI applications with Azure OpenAI Service' and 'Deploy an Azure OpenAI model and start making API calls. Connect your own data, call functions, and improve workflow with Azure OpenAI language, image and speech'. A 'Get Started' button is also present. In the top right corner of the main content area, there is a 'View Cost' link and a 'JSON View' button, which is highlighted with a red box.

2. API 버전에서 최신 API 버전을 선택합니다. 최신 API 버전인 2023-10-01-preview 만 지원됩니다.

Resource JSON

X

Resource ID

/subscriptions/

/resourceGroups/

/providers/Microsoft/

API Versions

2023-10-01-preview



```
75     "networkAcls": {  
76       "bypass": "AzureServices",  
77       "defaultAction": "Deny",  
78       "virtualNetworkRules": [],  
79       "ipRules": []  
80     },
```

가격 책정

가격 책정에 대한 자세한 내용은 [Azure Private Link 가격 책정](#)을 참조하세요.

다음 단계

- 다양한 Azure AI 서비스 살펴보기
- [가상 네트워크 서비스 엔드포인트](#)에 대한 자세한 내용

문서 인텔리전스에 대한 관리 ID

아티클 • 2023. 12. 06.

이 콘텐츠는 : ✓ v4.0(미리 보기) ✓ v3.1(GA) ✓ v3.0(GA) ✓ v2.1(GA)에 적용됩니다.

Azure 리소스에 대한 관리 ID는 Azure 관리되는 리소스에 대한 Microsoft Entra ID 및 특정 권한을 만드는 서비스 주체입니다.



- 관리 ID를 사용하여 사용자 고유의 애플리케이션을 포함하여 Microsoft Entra 인증을 지원하는 모든 리소스에 대한 액세스 권한을 부여할 수 있습니다. 보안 키 및 인증 토큰과 달리 관리 ID는 개발자가 자격 증명을 관리할 필요가 없습니다.
- Azure 리소스에 대한 액세스 권한을 부여하려면 [Azure RBAC\(Azure 역할 기반 액세스 제어\)](#)를 사용하여 관리 ID에 Azure 역할을 할당합니다.
- Azure에서 관리 ID를 사용하는 데 추가되는 비용은 없습니다.

① 중요

- 관리 ID는 SAS(공유 액세스 서명) 토큰을 포함하여 자격 증명을 관리할 필요가 없습니다.
- 관리 ID는 코드에 자격 증명 없이 데이터에 대한 액세스 권한을 부여하는 안전한 방법입니다.

프라이빗 스토리지 계정 액세스

프라이빗 Azure Storage 계정 액세스 및 인증은 Azure 리소스에 대한 관리 ID를 지원 [합니다](#). VNet(가상 네트워크) 또는 방화벽으로 보호되는 Azure Storage 계정이 있는 경우 문서 인텔리전스는 스토리지 계정 데이터에 직접 액세스할 수 없습니다. 단, 관리 ID가 사용하도록 설정되면 문서 인텔리전스는 할당된 관리 ID 자격 증명을 사용하여 스토리지 계정에 액세스할 수 있습니다.

① 참고

- FOTT(문서 인텔리전스 샘플 레이블 지정 도구) [\[링크\]](#)를 사용하여 스토리지 데이터를 분석하려면 VNet 또는 방화벽 뒤에 도구를 배포해야 합니다.
- 분석 확인 [\[링크\]](#), 명함 [\[링크\]](#), 청구서 [\[링크\]](#), ID 문서 [\[링크\]](#), 사용자 지정 양식 [\[링크\]](#) API는 요청을 원시 이진 콘텐츠로 게시하여 단일 문서에서 데이터를 추출할 수 있습니다. 이러한 시나리오에서는 관리 ID 자격 증명에 대한 요구 사항이 없습니다.

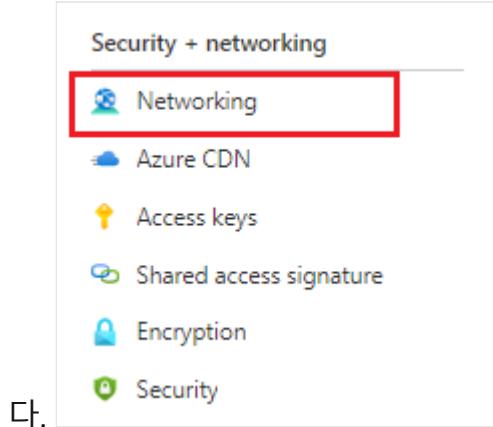
필수 조건

시작하려면 다음이 필요합니다.

- 활성 [Azure 계정](#) [\[링크\]](#) - 계정이 없는 경우 [체험 계정을 만들](#) 수 있습니다.
- Azure Portal의 [Document Intelligence](#) [\[링크\]](#) 또는 [Azure AI 서비스](#) [\[링크\]](#) 리소스 자세한 단계는 [다중 서비스 리소스 만들기](#)를 참조하세요.
- Document Intelligence 리소스와 동일한 지역의 [Azure Blob Storage 설정](#) [\[링크\]](#). 또한 스토리지 계정 내에서 Blob 데이터를 저장하고 구성하는 컨테이너를 만들어야 합니다.
 - 스토리지 계정이 방화벽 뒤에 있는 경우

구성을 사용하도록 설정해야 합니다.

- 스토리지 계정 페이지의 왼쪽 메뉴에서 **보안 + 네트워킹** → **네트워킹**을 선택합니



- 주 창에서 선택한 네트워크에서 액세스 허용을 선택합니다.

The screenshot shows the 'Firewalls and virtual networks' section of the Azure Storage account settings. It includes tabs for 'Firewalls and virtual networks', 'Private endpoint connections', and 'Custom domain'. Below the tabs are buttons for 'Save', 'Discard', and 'Refresh'. A note says: 'Firewall settings allowing access to storage services will remain in effect for up to a minute after saving updated settings restricting access.' Under 'Allow access from', there are two options: 'All networks' (unchecked) and 'Selected networks' (checked and highlighted with a red circle). A note below says: 'Configure network security for your storage accounts. Learn more'.

- 선택한 네트워크 페이지에서 예외 범주로 이동하여 신뢰할 수 있는 서비스 목록의 Azure 서비스가 이 스토리지 계정에 액세스하도록 허용 확인란이 사용하도록 설정되어 있는지 확인합니다.

The screenshot shows the 'Exceptions' section. It contains three checkboxes: 'Allow Azure services on the trusted services list to access this storage account' (checked and highlighted with a red box), 'Allow read access to storage logging from any network' (unchecked), and 'Allow read access to storage metrics from any network' (unchecked).

- Azure Portal을 사용하여 Azure RBAC(Azure 역할 기반 액세스 제어)를 간략히 이해 합니다.

관리 ID 할당

관리 ID에는 시스템 할당 및 사용자 할당의 두 가지 유형이 있습니다. 현재 단계에서 문서 인텔리전스는 시스템 할당 관리 ID만 지원합니다.

- 시스템이 할당한 관리 ID는 서비스 인스턴스에서 직접 사용하도록 설정됩니다. 기본적으로 사용하도록 설정되지 않습니다. 리소스로 이동하여 ID 설정을 업데이트해야 합니다.
- 시스템이 할당한 관리 ID는 수명 주기 내내 리소스에 연결됩니다. 리소스를 삭제하면 관리 ID도 삭제됩니다.

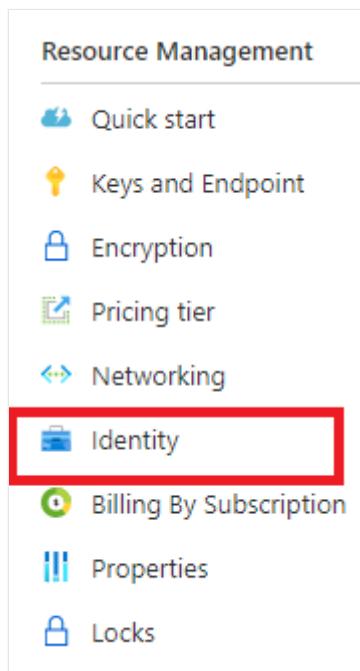
다음 단계에서는 시스템이 할당한 관리 ID를 사용하도록 설정하고 문서 인텔리전스에 Azure Blob Storage 계정에 대한 제한된 액세스 권한을 부여합니다.

시스템 할당 관리 ID 사용

① 중요

시스템 할당 관리 ID를 사용하도록 설정하려면 소유자 또는 사용자 액세스 관리 **istrator**와 같은 Microsoft.Authorization/roleAssignments/write 권한이 필요합니다. 관리 그룹, 구독, 리소스 그룹 또는 리소스의 네 가지 수준에서 범위를 지정할 수 있습니다.

1. Azure 구독과 연결된 계정을 사용하여 [Azure Portal](#)에 로그인합니다.
2. Azure Portal에서 문서 인텔리전스 리소스 페이지로 이동합니다.
3. 왼쪽 레일의 리소스 관리 목록에서 ID를 선택합니다.



4. 기본 창에서 시스템 할당 상태 탭을 켜기로 전환합니다.

스토리지 계정에 대한 액세스 권한 부여

Blob을 읽기 전에 스토리지 계정에 대한 Document Intelligence 액세스 권한을 부여해야 합니다. 이제 시스템이 할당한 관리 ID로 문서 인텔리전스를 사용하도록 설정했으므로 Azure RBAC(Azure 역할 기반 액세스 제어)를 사용하여 문서 인텔리전스에 Azure Storage에 대한 액세스 권한을 부여할 수 있습니다. **Storage Blob 데이터 읽기 권한자** 역할은 Blob 컨테이너 및 데이터에 대한 읽기 및 나열 권한을 문서 인텔리전스(시스템이 할당한 관리 ID로 표시됨)에 제공합니다.

1. 사용 권한에서 Azure 역할 할당을 선택합니다.

System assigned User assigned

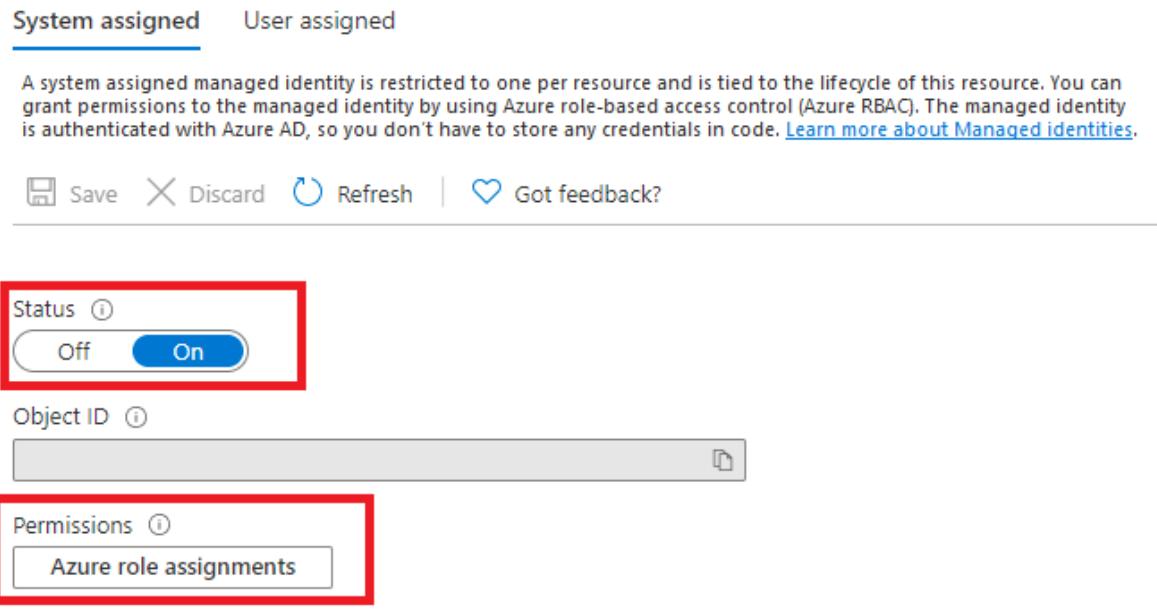
A system assigned managed identity is restricted to one per resource and is tied to the lifecycle of this resource. You can grant permissions to the managed identity by using Azure role-based access control (Azure RBAC). The managed identity is authenticated with Azure AD, so you don't have to store any credentials in code. [Learn more about Managed identities.](#)

Save Discard Refresh Got feedback?

Status ⓘ Off On

Object ID ⓘ

Permissions ⓘ Azure role assignments



2. 열리는 Azure 역할 할당 페이지의 드롭다운 메뉴에서 구독을 선택한 다음 + 역할 할당 추가를 선택합니다.

Azure role assignments ...

+ Add role assignment (Preview) Refresh

If this identity has role assignments that you don't have permission to read, they won't be shown in the list. [Learn more](#)

Subscription *



① 참고

추가 > 역할 할당 추가 옵션이 사용되지 않기 때문에 Azure Portal에서 역할을 할당할 수 없거나 “이 범위에서 역할 할당을 추가할 권한이 없습니다” 권한 오류가 표시되는 경우 현재 스토리지 리소스의 스토리지 범위에서 소유자 또는 사용자 액세스 관리자와 같은 Microsoft.Authorization/roleAssignments/write 권한이 있는 할당된 역할을 가진 사용자로 로그인되어 있는지 확인합니다.

3. 다음으로, **Storage Blob 데이터 읽기 권한자** 역할을 문서 인텔리전스 서비스 리소스에 할당합니다. 역할 할당 추가 팝업 창에서 다음과 같이 필드를 완료하고 저장을 선택합니다.

Expand table

필드	값
범위	스토리지

필드	값
구독	스토리지 리소스와 연결된 구독입니다.
리소스	스토리지 리소스의 이름
역할	Storage Blob 데이터 읽기 권한자 - Azure Storage Blob 컨테이너 및 데이터에 대한 읽기 액세스를 허용합니다.

Add role assignment (Preview) ×

Scope ⓘ ▼
Storage

Subscription ▼

Resource ⓘ ▼

Role ⓘ ▼
Storage Blob Data Reader ⓘ

[Learn more about RBAC](#)

Save Discard

4. 추가된 역할 할당 확인 메시지를 받은 후 페이지를 새로 고쳐 추가된 역할 할당을 확인합니다.



5. 변경 내용이 즉시 표시되지 않으면 기다렸다가 페이지를 다시 새로 고쳐 보세요. 역할 할당을 할당하거나 제거하면 변경 내용이 적용되는 데 최대 30분이 걸릴 수 있습니다.

Azure role assignments ...

+ Add role assignment (Preview) ↻ Refresh

If this identity has role assignments that you don't have permission to read, they won't be shown in the list. [Learn more](#)

Subscription * ▼

Role	Resource Name	Resource Type	Assigned To
Storage Blob Data Reader	YourStorage 	Storage account	Your-Form-Recognizer-Service

정말 간단하죠. 시스템이 할당한 관리 ID를 사용하도록 설정하는 단계를 완료했습니다. 관리 ID 및 Azure RBAC를 사용하여 SAS 토큰과 같은 자격 증명을 관리하지 않고도 스토

리지 리소스에 대한 문서 인텔리전스 특정 액세스 권한을 부여했습니다.

Document Intelligence Studio에 대한 추가 역할 할당

Document Intelligence Studio를 사용하려는 경우 스토리지 계정이 방화벽 또는 가상 네트워크와 같은 네트워크 제한으로 구성된 경우 추가 역할인 **Storage Blob 데이터 기여자**는 Document Intelligence 서비스에 할당되어야 합니다. Document Intelligence Studio에서는 자동 레이블, OCR 업그레이드, 루프의 휴면 또는 프로젝트 공유 작업을 수행할 때 스토리지 계정에 Blob을 쓰려면 이 역할이 필요합니다.

다음 단계

관리 ID 및 프라이빗 엔드포인트를 사용하여 보안 액세스 구성

관리 ID 및 가상 네트워크를 사용하여 보안 액세스 구성

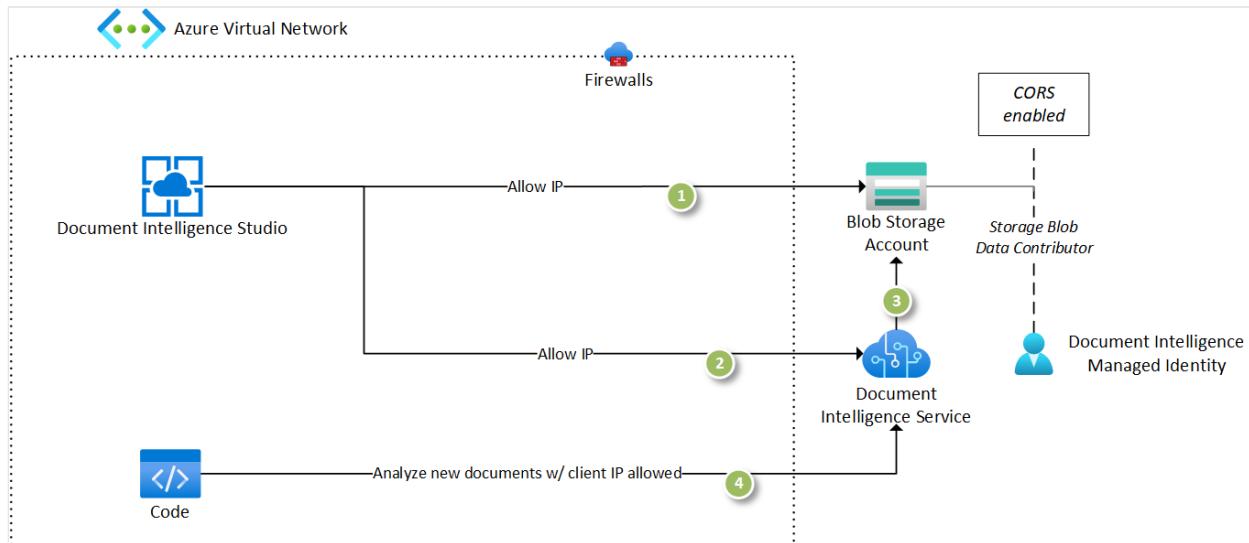
아티클 • 2024. 03. 05.

이 콘텐츠의 적용 대상은 다음과 같습니다. ✓ v4.0(미리 보기) ✓ v3.1(GA) ✓ v3.0(GA)
✓ v2.1(GA)

이 방법 가이드에서는 Document Intelligence 리소스에 보안 연결을 사용하도록 설정하는 프로세스를 안내합니다. 다음 연결의 보안을 유지할 수 있습니다.

- Virtual Network(VNET) 내의 클라이언트 응용 프로그램과 Document Intelligence 리소스 간의 통신
- Document Intelligence Studio와 Document Intelligence 리소스 간 통신
- Document Intelligence 리소스와 스토리지 계정 간의 통신(사용자 지정 모델을 학습 시킬 때 필요)

리소스를 보호하기 위해 환경을 설정합니다.



필수 조건

시작하려면 다음이 필요합니다.

- 활성 [Azure 계정](#) - 계정이 없는 경우 [체험 계정을 만들기](#) 수 있습니다.
- Azure Portal의 [Document Intelligence](#) 또는 [Azure AI 서비스](#) 리소스 자세한 단계는 [다중 서비스 리소스 만들기](#)를 참조하세요.

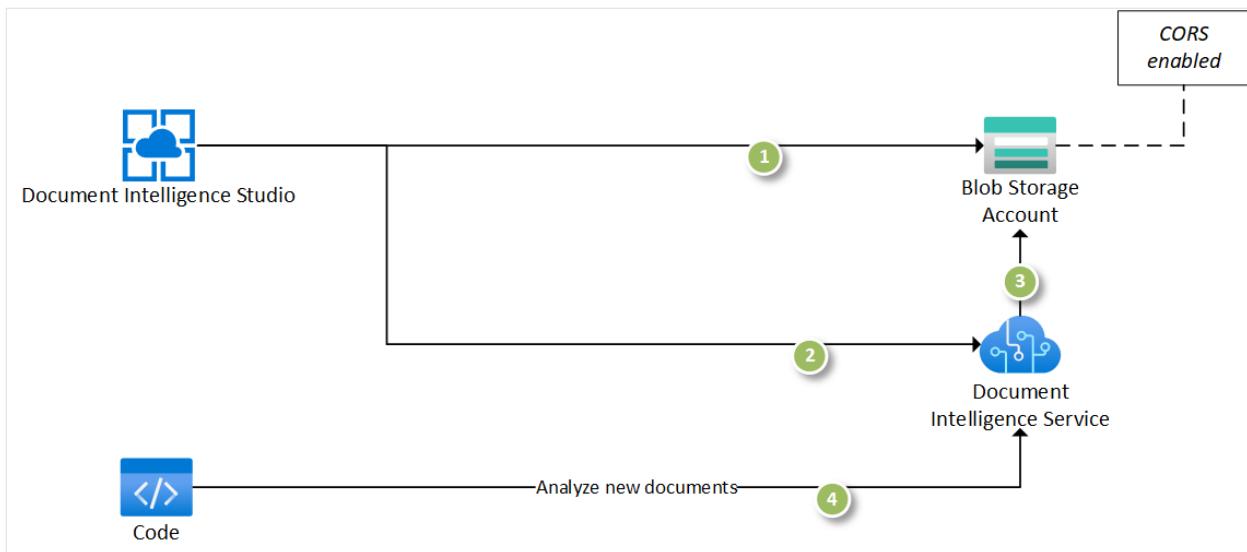
- Document Intelligence 리소스와 동일한 지역의 [Azure Blob Storage 계정](#). 스토리지 계정 내에서 Blob 데이터를 저장하고 구성하는 컨테이너를 만듭니다.
- Document Intelligence 리소스와 동일한 지역의 [Azure 가상 네트워크](#). 모델을 학습시키고 문서를 분석하는 애플리케이션 리소스를 배포하기 위한 가상 네트워크를 만듭니다.
- [Windows](#) 또는 [Linux/Ubuntu](#)용 Azure 데이터 과학 VM은 선택적으로 가상 네트워크에 데이터 과학 VM을 배포하여 설정되는 보안 연결을 테스트합니다.

리소스 구성

리소스가 서로 통신할 수 있도록 각 리소스를 구성합니다.

- 설정 페이지에 액세스하고 리소스를 선택하여 새로 만든 Document Intelligence 리소스를 사용하도록 Document Intelligence Studio를 구성합니다.
- 읽기 API를 선택하고 샘플 문서를 분석하여 구성이 작동하는지 확인합니다. 리소스가 올바르게 구성된 경우 요청이 성공적으로 완료됩니다.
- 만든 스토리지 계정의 컨테이너에 학습 데이터 세트를 추가합니다.
- 사용자 지정 모델 타일을 선택하여 사용자 지정 프로젝트를 만듭니다. 이전 단계에서 만든 동일한 Document Intelligence 리소스 및 스토리지 계정을 선택해야 합니다.
- 이전 단계에서 업로드한 학습 데이터 세트가 있는 컨테이너를 선택합니다. 학습 데이터 세트가 폴더 내에 있는 경우 폴더 경로가 적절하게 설정되었는지 확인합니다.
- 필요한 권한이 있는 경우 Studio는 스토리지 계정에 액세스하는 데 필요한 CORS 설정을 지정합니다. 권한이 없는 경우 계속 진행하려면 CORS 설정이 Storage 계정에 구성되어 있는지 확인해야 합니다.
- Studio가 학습 데이터에 액세스하도록 구성되어 있는지 확인합니다. 레이블 지정 환경에서 문서를 볼 수 있는 경우 필요한 모든 연결이 설정된 것입니다.

이제 기본 보안 모델을 사용하여 Document Intelligence 솔루션을 빌드하는 데 필요한 모든 구성 요소 구현이 작동되고 있습니다.



이제 다음 단계를 완료합니다.

- Document Intelligence 리소스에서 관리 ID를 설정합니다.
- 특정 가상 네트워크 및 IP 주소의 트래픽만 제한하도록 스토리지 계정을 보호합니다.
- 스토리지 계정과 통신하도록 Document Intelligence 관리 ID를 구성합니다.
- Document Intelligence 리소스에 대한 퍼블릭 액세스를 사용하지 않도록 설정하고 특정 가상 네트워크 및 IP 주소에서만 액세스할 수 있도록 프라이빗 엔드포인트를 만듭니다.
- 선택한 가상 네트워크에서 스토리지 계정에 대한 프라이빗 엔드포인트를 추가합니다.
- 가상 네트워크 내에서 모델을 학습시키고 문서를 분석할 수 있는지 확인합니다.

Document Intelligence의 관리 ID 설정

Azure Portal의 Document Intelligence 리소스로 이동하고 **ID** 탭을 선택합니다. **시스템 할당** 관리 ID를 **켜기**로 전환하고 변경 내용을 저장합니다.

The screenshot shows the Azure Portal interface for managing a resource named "Form recognizer". On the left, there's a navigation menu with links for Overview, Activity log, Access control (IAM), Tags, Diagnose and solve problems, Resource Management, Identity, Cost analysis, Properties, Locks, Monitoring, Alerts, and Metrics. The "Identity" link is currently selected. At the top right, there are tabs for "System assigned" (which is selected) and "User assigned". Below the tabs, a note states: "A system assigned managed identity is restricted to one per resource and is tied to the lifecycle of this resource. You don't have to store any credentials in code. [Learn more about Managed identities](#)". There are buttons for Save, Discard, Refresh, and Got feedback?. A "Status" toggle switch is set to "On".

트래픽을 제한하도록 스토리지 계정 보호

Azure Portal에서 스토리지 계정의 네트워킹 탭으로 이동하여 보안 통신을 구성하기 시작합니다.

1. 방화벽 및 가상 네트워크의 공용 네트워크 액세스 목록에서 선택한 가상 네트워크 및 IP 주소에서 사용을 선택합니다.
2. 예외 목록에서 신뢰할 수 있는 서비스 목록의 Azure 서비스가 이 스토리지 계정에 액세스하도록 허용이 선택되었는지 확인합니다.
3. 변경 내용을 저장합니다.

The screenshot shows the 'Firewalls and virtual networks' section of the Azure Storage Account settings. On the left, there's a navigation menu with options like Overview, Activity log, Tags, 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, and Geo-replication. The 'Networking' tab is selected. At the top right, there are Save, Discard, and Refresh buttons. Below them, under 'Public network access', 'Enabled from selected virtual networks and IP addresses' is selected. A link to 'Configure network security for your storage accounts' is provided. Under 'Virtual networks', there are buttons for 'Add existing virtual network' and 'Add new virtual network'. A table lists 'Virtual Network', 'Subnet', 'Address range', and 'Endpoint Status'. It shows 'No network selected.' Under 'Firewall', there's a note about adding IP ranges and a checkbox for 'Add your client IP address'. An 'Address range' input field is present. Under 'Resource instances', it says 'Specify resource instances that will have access to your storage account based on their system-assigned managed identity.' There are dropdown menus for 'Resource type' and 'Instance name'. Under 'Exceptions', several checkboxes are available, with the first one ('Allow Azure services on the trusted services list to access this storage account') checked.

① 참고

공용 인터넷에서 스토리지 계정에 액세스할 수 없습니다.

Studio에서 사용자 지정 모델 레이블 지정 페이지를 새로 고치면 오류 메시지가 표시됩니다.

Document Intelligence에서 스토리지에 대한 액세스 사용

Document Intelligence 리소스가 학습 데이터 세트에 액세스할 수 있도록 하려면 [관리 ID](#)에 대한 역할 할당을 추가해야 합니다.

1. Azure Portal의 스토리지 계정 창에 있는 상태에서 왼쪽 탐색 모음의 **액세스 제어 (IAM)** 탭으로 이동합니다.
2. **역할 할당 추가** 단추를 선택합니다.

Add role assignment ...

Got feedback?

Role Members Review + assign

A role definition is a collection of permissions. You can use the built-in roles or you can create your own custom roles. [Learn more](#)

Type : All Category : All

Showing 4 of 40 roles

Name ↑↓	Description ↑↓
Storage Blob Data Contributor	Allows for read, write and delete access to Azure Storage blob containers and data
Storage Blob Data Owner	Allows for full access to Azure Storage blob containers and data, including assigning POSIX access control.
Storage Blob Data Reader	Allows for read access to Azure Storage blob containers and data
Storage Blob Delegator	Allows for generation of a user delegation key which can be used to sign SAS tokens

3. 역할 탭에서 Storage Blob 데이터 읽기 권한자 권한을 선택하고 다음을 선택합니다.

Storage account

Search (Ctrl+ /)

Add Download role assignments Edit columns Refresh Remove Got feedback?

Check access Role assignments Roles Deny assignments Classic administrators

My access
View my level of access to this resource.
[View my access](#)

Check access
Review the level of access a user, group, service principal, or managed identity has to this resource. [Learn more](#)

Find User, group, or service principal Managed identity
Search by name or email address

Grant access to this resource
Grant access to resources by assigning a role.
[Add role assignment](#) Learn more

View deny assignments
View the role assignments that have been denied access to specific actions at this scope.
[View](#) Learn more

4. 멤버 탭에서 관리 ID 옵션을 선택하고 + 멤버 선택을 선택합니다.

5. 관리 ID 선택 대화 상자 창에서 다음 옵션을 선택합니다.

- **구독.** 구독을 선택합니다.
- **관리 ID.** Form Recognizer를 선택합니다.
- **선택** 관리 ID로 사용하도록 설정한 Document Intelligence 리소스를 선택합니다.

The screenshot shows the 'Add role assignment' interface in the Azure portal. The 'Selected role' is set to 'Storage Blob Data Reader'. Under 'Assign access to', the 'Managed identity' option is selected. In the 'Members' section, there is a link to 'Select members'. On the right, a sidebar titled 'Select managed identities' lists a single item: 'Form recognizer (1)'. There is also a feedback link at the top.

6. 대화 상자 창을 닫습니다.

7. 마지막으로 **검토 + 할당**을 선택하여 변경 내용을 저장합니다.

좋습니다! 관리 ID를 사용하여 스토리지 계정에 연결하도록 Document Intelligence 리소스를 구성했습니다.

💡 팁

Document Intelligence Studio 를 사용하려고 하면 READ API 및 기타 미리 빌드된 모델에 문서를 처리하는 데 스토리지 액세스 권한이 필요하지 않다는 것을 확인할 수 있습니다. 그러나 Studio에서 스토리지 계정과 직접 통신할 수 없으므로 사용자 지정 모델을 학습시키려면 추가 구성이 필요합니다. 스토리지 계정의 **네트워킹** 탭에서 **클라이언트 IP 주소 추가**를 선택하여 IP 허용 목록을 통해 스토리지 계정에 액세스하도록 머신을 구성함으로써 스토리지 액세스를 사용하도록 설정할 수 있습니다.

VNET에서 액세스하기 위해 프라이빗 엔드포인트 구성

ⓘ 참고

- 리소스는 가상 네트워크에서만 액세스할 수 있습니다.
- 자동 레이블과 같은 Studio의 일부 Document Intelligence 기능을 사용하려면 Document Intelligence Studio에서 스토리지 계정에 액세스할 수 있어야 합니다.
- Document Intelligence 및 Storage 계정 리소스 모두에 대한 방화벽 허용 목록에 Studio IP 주소 20.3.165.95를 추가합니다. 문서 인텔리전스 스튜디오의 전용 IP 주소이며 안전하게 허용할 수 있습니다.

가상 네트워크에서 리소스에 연결할 때 프라이빗 엔드포인트를 추가하면 가상 네트워크에서 스토리지 계정과 Document Intelligence 리소스 둘 다에 액세스할 수 있습니다.

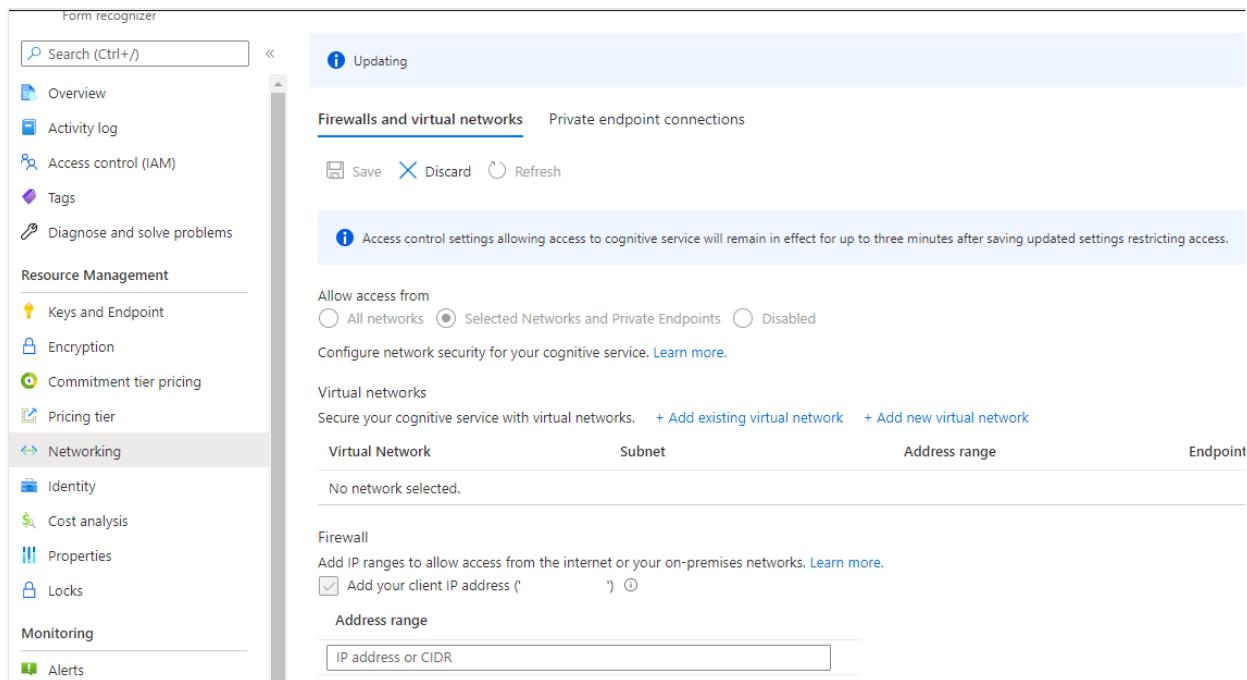
다음으로 가상 네트워크 내의 리소스 또는 네트워크를 통한 트래픽 라우터만 Document Intelligence 리소스 및 스토리지 계정에 액세스할 수 있도록 가상 네트워크를 구성합니다.

방화벽 및 가상 네트워크 사용

1. Azure Portal에서 Document Intelligence 리소스로 이동합니다.
2. 왼쪽 탐색 모음에서 **네트워킹** 탭을 선택합니다.
3. **방화벽 및 가상 네트워크** 탭에서 선택한 네트워킹 및 프라이빗 엔드포인트 옵션을 사용하도록 설정하고 저장을 선택합니다.

① 참고

Document Intelligence Studio 기능에 액세스하려고 하면 액세스 거부 메시지가 표시됩니다. 컴퓨터의 Studio에서 액세스를 사용하도록 설정하려면 **클라이언트 IP 주소 추가 확인란**을 선택하고 **저장**을 선택하여 액세스를 복원합니다.



The screenshot shows the Azure portal interface for managing a cognitive service. On the left, there's a sidebar with various resource management options like Overview, Activity log, Access control (IAM), Tags, and Networking. The Networking option is currently selected. The main pane is titled 'Firewalls and virtual networks' and shows a status message 'Updating'. It has tabs for 'Firewalls and virtual networks' and 'Private endpoint connections'. Below the tabs are buttons for Save, Discard, and Refresh. A note says 'Access control settings allowing access to cognitive service will remain in effect for up to three minutes after saving updated settings restricting access.' Under 'Allow access from', there are three radio button options: 'All networks' (unchecked), 'Selected Networks and Private Endpoints' (checked), and 'Disabled' (unchecked). A note below says 'Configure network security for your cognitive service. Learn more.' Under 'Virtual networks', it says 'Secure your cognitive service with virtual networks.' with buttons for '+ Add existing virtual network' and '+ Add new virtual network'. A table header shows columns for 'Virtual Network', 'Subnet', 'Address range', and 'Endpoint'. Below the table, it says 'No network selected.' Under 'Firewall', it says 'Add IP ranges to allow access from the internet or your on-premises networks. Learn more.' with a checked checkbox for 'Add your client IP address (' followed by an input field and a 'Save' button).

프라이빗 엔드포인트 구성

1. **프라이빗 엔드포인트 연결** 탭으로 이동하고 + **프라이빗 엔드포인트**를 선택합니다. **프라이빗 엔드포인트 만들기** 대화 상자 페이지로 이동됩니다.
2. **프라이빗 엔드포인트 만들기** 대화 상자 페이지에서 다음 옵션을 선택합니다.
 - **구독**. 청구 구독을 선택합니다.

- **리소스 그룹.** 해당 리소스 그룹을 선택합니다.
- **이름.** 프라이빗 엔드포인트에 대한 이름을 입력합니다.
- **지역.** 가상 네트워크와 동일한 지역을 선택합니다.
- **다음: 리소스를 선택합니다.**

Create a private endpoint ...

Basics **Resource** **Virtual Network** **Tags** **Review + create**

Use private endpoints to privately connect to a service or resource. Your private endpoint must be in the same region as your virtual network, but can be in a different region from the private link resource that you are connecting to. [Learn more](#)

Project details

Subscription * ⓘ Visual Studio Enterprise Subscription

Resource group * ⓘ work-related

Create new

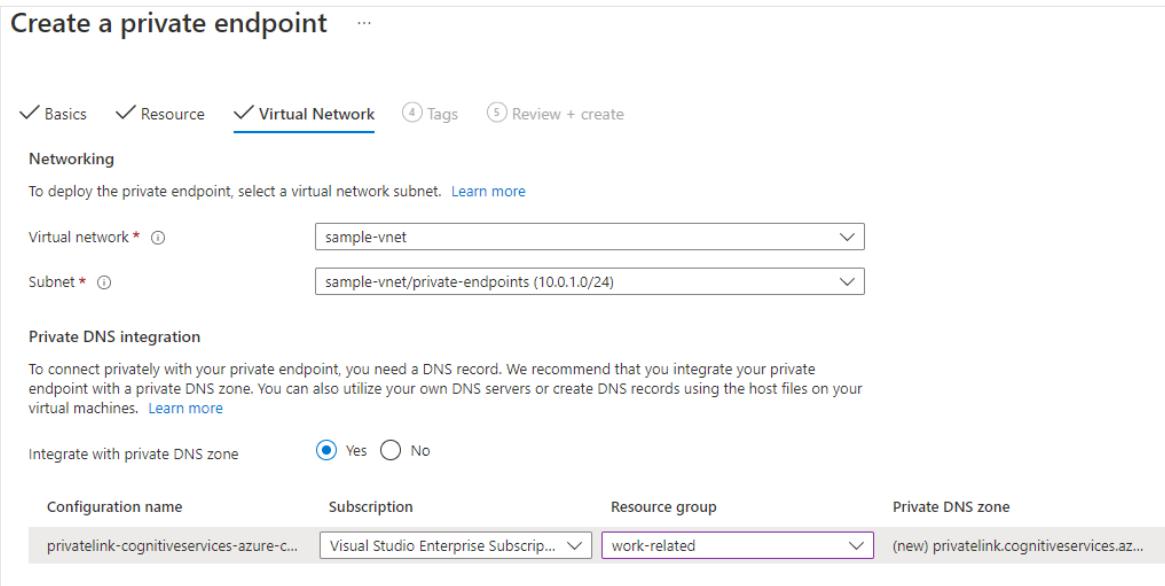
Instance details

Name * my-fr-endpoint

Region * Southeast Asia

가상 네트워크 구성

1. 리소스 탭에서 기본값을 적용하고 **다음: 가상 네트워크**를 선택합니다.
2. **Virtual Network** 탭에서 만든 가상 네트워크를 선택했는지 확인합니다.
3. 서브넷이 여러 개인 경우 프라이빗 엔드포인트를 연결할 서브넷을 선택합니다. **IP 주소를 동적으로 할당하려면 기본값을 적용합니다.**
4. **다음: DNS**를 선택합니다.
5. **프라이빗 DNS 영역과 통합**에 대해 기본값 예를 적용합니다.



6. 나머지 기본값을 수락하고 **다음: 태그**를 선택합니다.

7. 완료되면 **다음: 리뷰 + 만들기**를 클릭합니다.

모두 완료되었습니다! 이제 Document Intelligence 리소스는 가상 네트워크 및 IP 허용 목록의 임의 IP 주소에서만 액세스할 수 있습니다.

스토리지에 대한 프라이빗 엔드포인트 구성

Azure Portal의 **스토리지 계정**으로 이동합니다.

1. 왼쪽 탐색 메뉴에서 **네트워킹** 탭을 선택합니다.
2. **프라이빗 엔드포인트 연결** 탭을 선택합니다.
3. **+ 프라이빗 엔드포인트 추가**를 선택합니다.
4. 이름을 입력하고 가상 네트워크와 동일한 지역을 선택합니다.
5. **다음: 리소스**를 선택합니다.

Create a private endpoint ...

1 Basics 2 Resource 3 Virtual Network 4 Tags 5 Review + create

Use private endpoints to privately connect to a service or resource. Your private endpoint must be in the same region as your virtual network, but can be in a different region from the private link resource that you are connecting to. [Learn more](#)

Project details

Subscription * ⓘ Visual Studio Enterprise Subscription

Resource group * ⓘ work-related
[Create new](#)

Instance details

Name * my-stg-endpoint

Region * Southeast Asia

6. 리소스 탭의 대상 하위 리소스 목록에서 Blob을 선택합니다.

7. 다음: 가상 네트워크를 선택합니다.

Create a private endpoint ...

✓ Basics 2 Resource 3 Virtual Network 4 Tags 5 Review + create

Private Link offers options to create private endpoints for different Azure resources, like your private link service, a SQL server, or an Azure storage account. Select which resource you would like to connect to using this private endpoint. [Learn more](#)

Subscription Visual Studio Enterprise Subscription (083694cc-cace-414f-9d55-d6b9071db956)

Resource type Microsoft.Storage/storageAccounts

Resource vikurpadwork

Target sub-resource * ⓘ

- blob
- blob
- table
- queue
- file
- web
- dfs

8. 가상 네트워크 및 서브넷을 선택합니다. 이 서브넷의 모든 프라이빗 엔드포인트에 대해 네트워크 정책 사용이 선택되어 있고 동적으로 IP 주소 할당이 사용하도록 설정되어 있는지 확인합니다.

9. 다음: DNS를 선택합니다.

10. 프라이빗 DNS 영역과 통합에 대해 예가 사용하도록 설정되어 있는지 확인합니다.

11. 완료되면 다음: 태그를 선택합니다.

12. 완료되면 다음: 리뷰 + 만들기를 클릭합니다.

잘했습니다! 이제 관리 ID를 사용하도록 구성된 Document Intelligence 리소스와 스토리지가 모두 연결되어 있습니다.

① 참고

리소스는 가상 네트워크 및 허용된 IP에서만 액세스할 수 있습니다.

요청이 가상 네트워크에서 시작되거나 가상 네트워크를 통해 라우팅 되지 않는 한, Document Intelligence 리소스에 대한 Studio 액세스 및 분석 요청이 실패합니다.

배포 확인

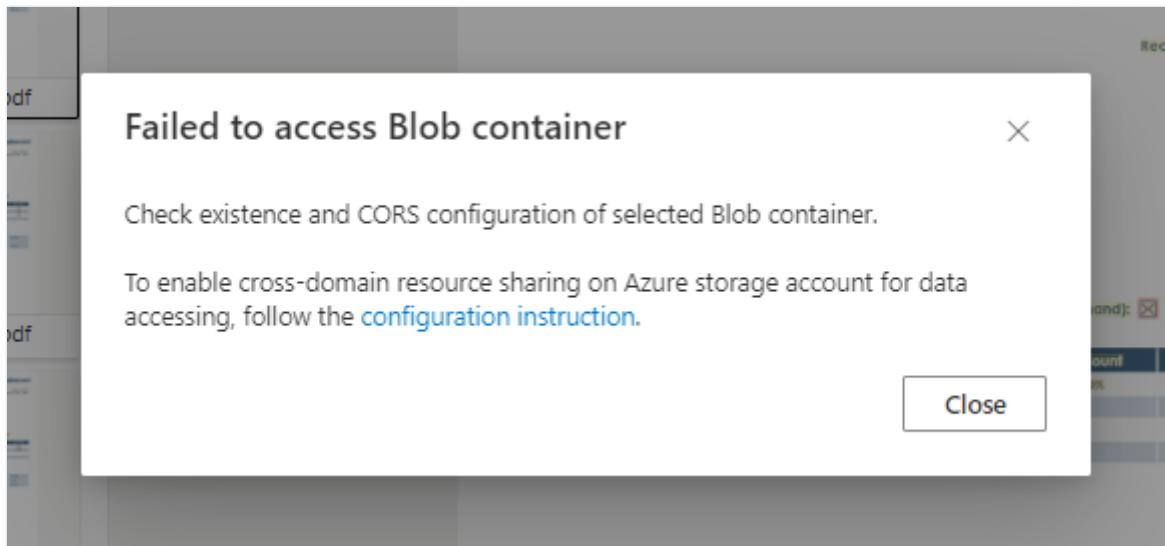
배포의 유효성을 검사하려면 가상 네트워크에 VM(가상 머신)을 배포하고 리소스에 연결할 수 있습니다.

1. 가상 네트워크에서 [Data Science VM](#) 을 구성합니다.
2. 데스크톱에서 VM에 원격으로 연결하고 브라우저 세션을 시작하여 Document Intelligence Studio에 액세스합니다.
3. 요청을 분석합니다. 그러면 학습 작업이 성공적으로 작동해야 합니다.

정말 간단하죠. 이제 관리 ID 및 프라이빗 엔드포인트를 사용하여 Document Intelligence 리소스에 대한 보안 액세스를 구성할 수 있습니다.

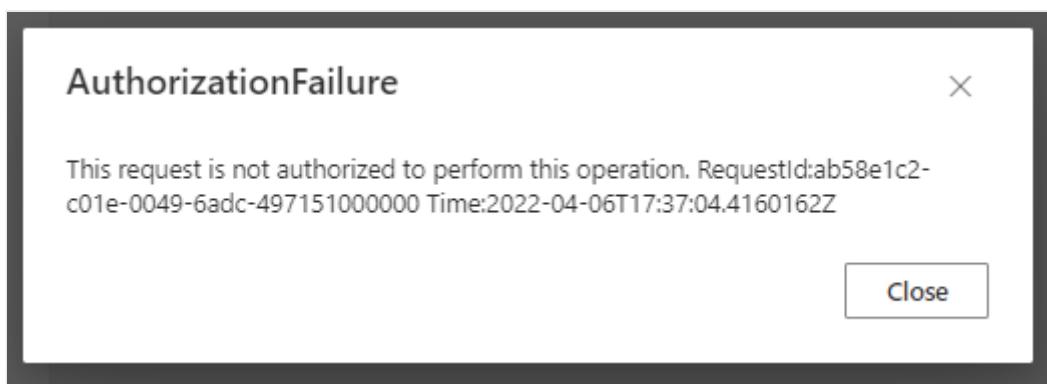
일반적인 오류 메시지

- Blob 컨테이너에 액세스하지 못했습니다.



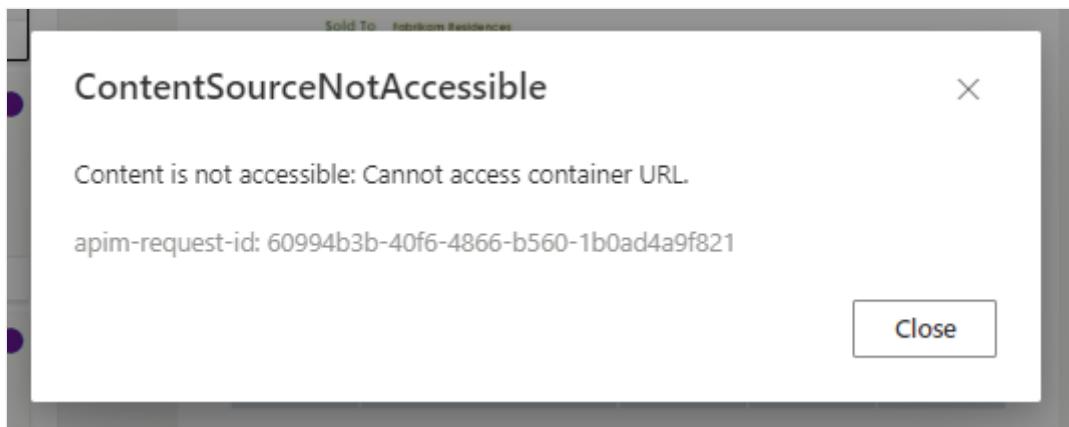
해결 방법:

1. [CORS를 구성합니다.](#)
 2. 클라이언트 컴퓨터가 동일한 VNET에 있는 Document Intelligence 리소스 및 스토리지 계정에 액세스할 수 있는지 확인하거나, Document Intelligence 리소스와 스토리지 계정 모두의 **네트워킹 > 방화벽과 가상 네트워크** 설정 페이지에서 클라이언트 IP 주소가 허용되는지 확인합니다.
- **AuthorizationFailure:**



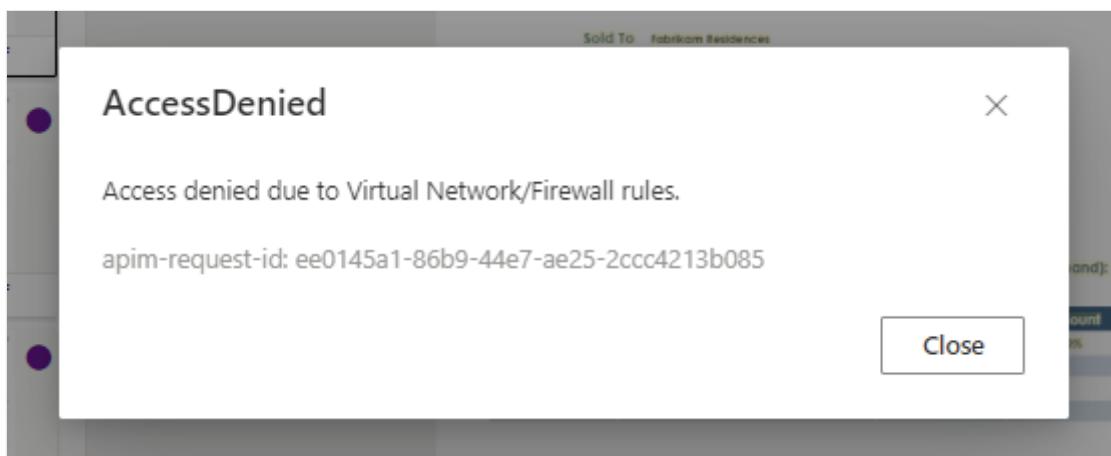
해결 방법: 클라이언트 컴퓨터가 동일한 VNET에 있는 Document Intelligence 리소스 및 스토리지 계정에 액세스할 수 있는지 확인하거나, Document Intelligence 리소스와 스토리지 계정 모두의 **네트워킹 > 방화벽과 가상 네트워크** 설정 페이지에서 클라이언트 IP 주소가 허용되는지 확인합니다.

- **ContentSourceNotAccessible:**



해결 방법: Document Intelligence 관리 ID에 Storage Blob 데이터 읽기 권한자 역할을 부여하고 네트워킹 탭에서 신뢰할 수 있는 서비스 액세스 또는 리소스 인스턴스 규칙을 사용하도록 설정했는지 확인합니다.

- AccessDenied:



해결 방법: Document Intelligence Studio에 액세스하는 컴퓨터와 Document Intelligence 서비스 간에 연결이 있는지 확인합니다. 예를 들어, Document Intelligence 서비스의 네트워킹 탭에 클라이언트 IP 주소를 추가해야 할 수 있습니다.

다음 단계

[웹앱에서 관리 ID를 사용하여 Azure Storage에 액세스](#)

미사용 데이터의 문서 인텔리전스 암호화

아티클 • 2023. 11. 15.

이 콘텐츠는 : v4.0(미리 보기) v3.1(GA) v3.0(GA) v2.1(GA)에 적용됩니다.

ⓘ 중요

- 이전 버전의 고객 관리형 키는 모델만 암호화했습니다. *릴리스부터 07/31/2023 모든 새 리소스는 고객 관리 키를 사용하여 모델과 문서 결과를 모두 암호화합니다. 모델과 데이터를 모두 암호화하도록 기존 서비스를 업그레이드하려면 고객 관리형 키를 사용하지 않도록 설정하고 다시 사용하도록 설정하기만 하면 됩니다.

Azure AI 문서 인텔리전스는 데이터를 클라우드에 유지할 때 자동으로 데이터를 암호화합니다. 문서 인텔리전스 암호화는 조직의 보안 및 규정 준수 약정을 충족하는 데 도움이 되도록 데이터를 보호합니다.

Azure AI 서비스 암호화 정보

데이터는 [FIPS 140-2](#) 규격 [256비트 AES](#) 암호화를 사용하여 암호화 및 암호 해독됩니다. 암호화 및 암호 해독은 투명하므로 암호화 및 액세스가 자동으로 관리됩니다. 사용자의 데이터는 기본적으로 안전합니다. 암호화를 활용하기 위해 코드나 애플리케이션을 수정할 필요가 없습니다.

암호화 키 관리 정보

기본적으로 구독은 Microsoft에서 관리하는 암호화 키를 사용합니다. 고객 관리형 키라고 하는 자체 키로 구독을 관리할 수도 있습니다. 고객 관리형 키를 사용하면 액세스 제어를 만들기, 회전, 사용하지 않도록 설정 및 취소하는 방식이 더 유연해집니다. 데이터를 보호하는 데 사용하는 암호화 키를 감사할 수도 있습니다. 구독에 대해 고객 관리형 키가 구성된 경우 이중 암호화가 제공됩니다. 이 두 번째 보호 계층을 사용하면 Azure Key Vault를 통해 암호화 키를 제어할 수 있습니다.

ⓘ 중요

고객 관리형 키는 2020년 5월 11일 이후에 생성된 리소스만 사용할 수 있습니다. 문서 인텔리전스와 함께 CMK를 사용하려면 새 문서 인텔리전스 리소스를 만들어야 합니다. 리소스가 생성되면 Azure Key Vault를 사용하여 관리 ID를 설정할 수 있습니다.

Azure Key Vault를 사용하는 고객 관리형 키

고객 관리형 키를 사용하는 경우 Azure Key Vault를 사용하여 저장해야 합니다. 사용자 고유의 키를 만들어 키 자격 증명 모음에 저장할 수도 있고, Azure Key Vault API를 사용하여 키를 생성할 수도 있습니다. Azure AI 서비스 리소스와 키 자격 증명 모음은 동일한 지역 및 동일한 Microsoft Entra 테넌트에 있어야 하지만 서로 다른 구독에 있을 수 있습니다. Key Vault에 대한 자세한 내용은 [Azure Key Vault](#)를 참조하세요.

새 Azure AI 서비스 리소스를 만들면 항상 Microsoft 관리형 키를 사용하여 암호화됩니다. 리소스를 만들 때 고객 관리형 키를 사용하도록 설정할 수 없습니다. 고객 관리형 키는 Key Vault에 저장됩니다. 키 자격 증명 모음은 Azure AI 서비스 리소스와 연결된 관리 ID에 키 권한을 부여하는 액세스 정책으로 프로비저닝되어야 합니다. 관리 ID는 고객 관리형 키에 필요한 가격 책정 계층을 사용하여 리소스를 만든 후에만 사용할 수 있습니다.

고객 관리형 키를 사용하도록 설정하면 Microsoft Entra ID의 기능인 시스템 할당 [관리 ID](#)도 사용할 수 있습니다. 시스템 할당 관리 ID를 사용하도록 설정하면 이 리소스가 Microsoft Entra ID에 등록됩니다. 등록된 후 관리 ID에는 고객 관리형 키를 설정하는 중에 선택한 키 자격 증명 모음에 대한 액세스 권한이 부여됩니다.

① 중요

시스템 할당 관리 ID를 사용하지 않도록 설정하면 키 자격 증명 모음에 대한 액세스가 제거되고 고객 키로 암호화된 모든 데이터에 더 이상 액세스할 수 없습니다. 이 데이터에 의존하는 모든 기능이 작동을 멈춥니다.

① 중요

관리 ID는 현재 디렉터리 간 시나리오를 지원하지 않습니다. Azure Portal에서 고객 관리형 키를 구성하면 관리 ID가 자동으로 배후에서 할당됩니다. 이후에 구독, 리소스 그룹 또는 리소스를 한 Microsoft Entra 디렉터리에서 다른 디렉터리로 이동하는 경우 리소스와 연결된 관리 ID가 새 테넌트로 전송되지 않으므로 고객 관리형 키가 더 이상 작동하지 않을 수 있습니다. 자세한 내용은 [FAQ에서 Microsoft Entra 디렉터리](#) 간에 구독 전송 및 Azure 리소스에 대한 관리 ID와 관련된 알려진 문제를 참조하세요.

Key Vault 구성

고객 관리형 키를 사용하는 경우 키 자격 증명 모음에서 **일시 제거 및 제거 안 함**이라는 두 가지 속성을 설정해야 합니다. 이러한 속성은 기본적으로 사용하도록 설정되지 않지만 Azure Portal, PowerShell 또는 Azure CLI를 사용하여 신규 또는 기존 키 자격 증명 모음에서 사용하도록 설정할 수 있습니다.

ⓘ 중요

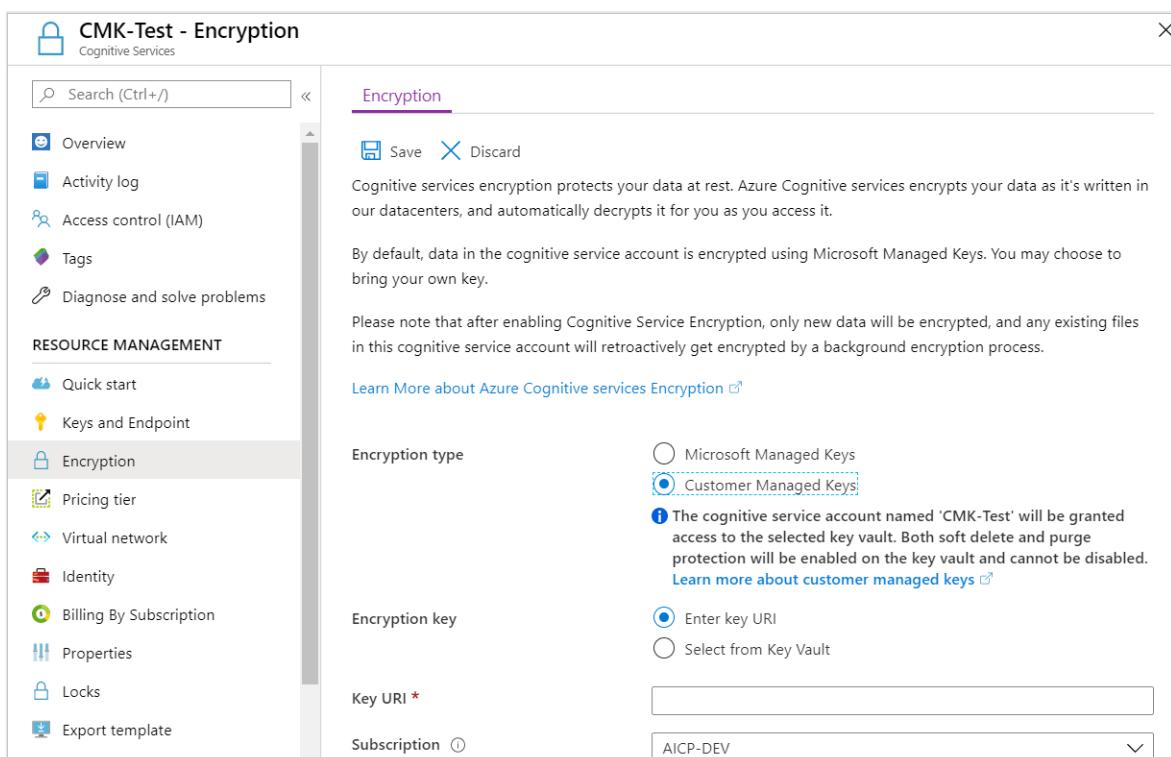
일시 제거 및 제거 안 함 속성이 사용하도록 설정되지 않은 상태에서 키를 제거하면 Azure AI 서비스 리소스의 데이터를 복구할 수 없습니다.

기존 키 자격 증명 모음에서 이러한 속성을 사용하도록 설정하는 방법에 대한 자세한 내용은 [일시 삭제 및 제거 보호를 통한 Azure Key Vault 복구 관리](#)를 참조하세요.

리소스에 대해 고객 관리형 키 사용

Azure Portal에서 고객 관리형 키를 사용하도록 설정하려면 다음 단계를 수행합니다.

1. Azure AI 서비스 리소스로 이동합니다.
2. 왼쪽에서 **암호화**를 선택합니다.
3. 다음 스크린샷과 같이 **암호화 형식**에서 **고객 관리형 키**를 선택합니다.



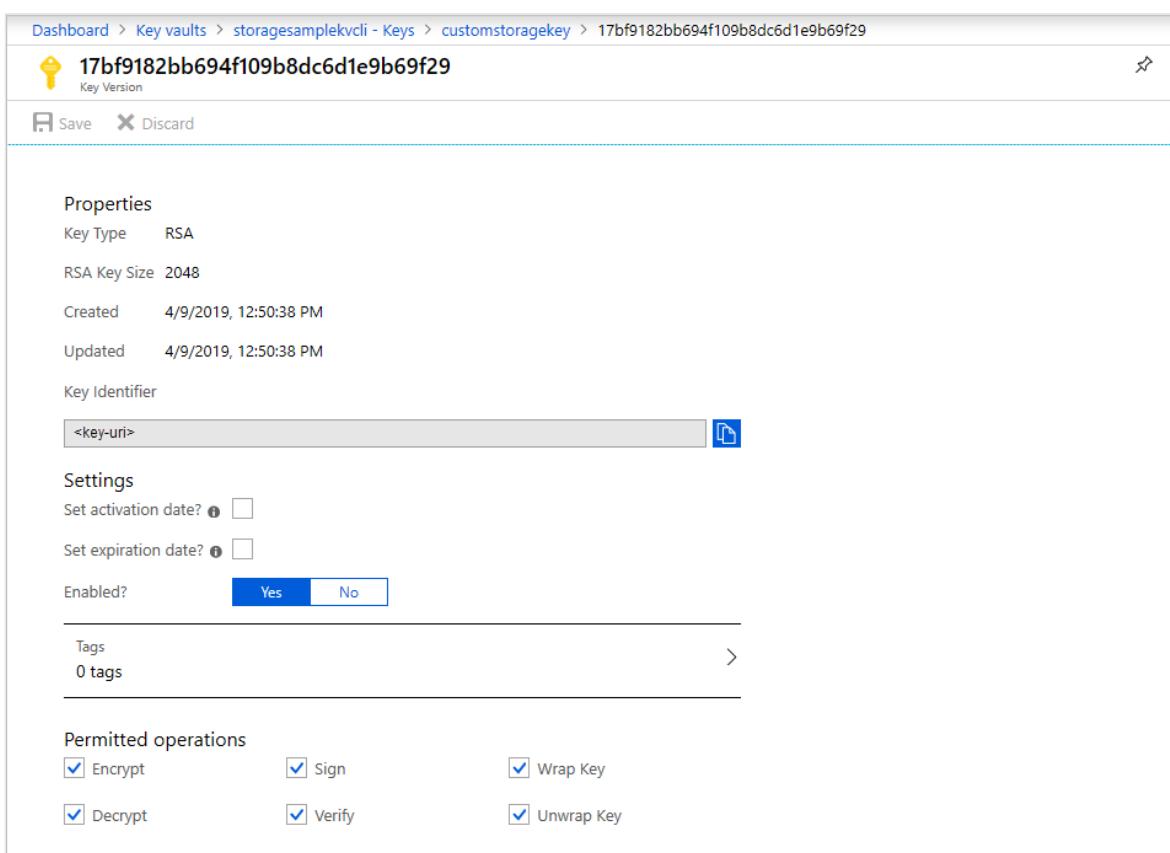
키 지정

고객 관리형 키를 사용하도록 설정한 후 Azure AI 서비스 리소스와 연결할 키를 지정할 수 있습니다.

키를 URI로 지정

키를 URI로 지정하려면 다음 단계를 수행합니다.

1. Azure Portal에서 키 자격 증명 모음으로 이동합니다.
2. 설정 아래에서 **키**를 선택합니다.
3. 원하는 키를 선택한 다음 해당 키를 선택하여 버전을 확인합니다. 해당 버전의 설정을 보려면 키 버전을 선택합니다.
4. URI를 제공하는 **키 식별자** 값을 복사합니다.



The screenshot shows the Azure Key Vault interface for managing keys. It displays a key named '17bf9182bb694f109b8dc6d1e9b69f29' with its properties: Key Type (RSA), RSA Key Size (2048), and creation and update times (4/9/2019, 12:50:38 PM). Below the properties is a 'Key Identifier' field containing '<key-uri>' with a copy icon. Under 'Settings', there are checkboxes for 'Set activation date?' and 'Set expiration date?'. The 'Enabled?' button is set to 'Yes'. In the 'Tags' section, it shows '0 tags'. At the bottom, under 'Permitted operations', several checkboxes are checked: Encrypt, Sign, Wrap Key, Decrypt, Verify, and Unwrap Key.

5. Azure AI 서비스 리소스로 돌아가서 **암호화**를 선택합니다.
6. **암호화 키**에서 **키 URI 입력**을 선택합니다.
7. 복사한 URI를 **키 URI** 상자에 붙여넣습니다.

The screenshot shows the 'Encryption' settings for a Cognitive Service account named 'CMK-Test'. The left sidebar lists various management options like Overview, Activity log, Access control (IAM), Tags, Diagnose and solve problems, and Resource Management (Quick start, Keys and Endpoint, Encryption, Pricing tier, Virtual network, Identity, Billing By Subscription, Properties, Locks, Export template). The 'Encryption' option is selected. The main pane displays the 'Encryption' configuration. It includes sections for 'Encryption type' (set to 'Customer Managed Keys'), 'Encryption key' (set to 'Enter key URI'), and 'Key URI' (containing '<key uri>'). A note states that after enabling encryption, only new data will be encrypted, and existing files will be retroactively encrypted. There's also a link to learn more about Azure Cognitive services Encryption.

8. 구독에서 키 자격 증명 모음이 포함된 구독을 선택합니다.

9. 변경 사항을 저장합니다.

키 자격 증명 모음에서 키 지정

키 자격 증명 모음에서 키를 지정하려면 먼저 키가 포함된 키 자격 증명 모음이 있는지 확인합니다. 그런 다음, 다음 단계를 수행합니다.

1. Azure AI 서비스 리소스로 이동한 다음 암호화를 선택합니다.
2. 암호화 키에서 Key Vault에서 선택을 선택합니다.
3. 사용하려는 키가 포함된 키 자격 증명 모음을 선택합니다.
4. 사용하려는 키를 선택합니다.

Microsoft Azure

Search resources, services, and docs (G+/-)

Home > CMKTest01-SB - Encryption > Select key from Azure Key Vault

Select key from Azure Key Vault

Subscription *	AICP-DEV
Key vault *	CMKTest-01SB Create new
Key *	CMKTest-01SB Create new
Version *	19fc5cfacbd34e47b373709c1e400902 Create new

5. 변경 사항을 저장합니다.

키 버전 업데이트

새 버전의 키를 만들 때 새 버전을 사용하도록 Azure AI 서비스 리소스를 업데이트합니다. 다음 단계를 수행합니다.

1. Azure AI 서비스 리소스로 이동한 다음 **암호화**를 선택합니다.
2. 새 키 버전의 URI를 입력합니다. 또는 키 자격 증명 모음을 선택한 다음 키를 다시 선택하여 버전을 업데이트할 수 있습니다.
3. 변경 사항을 저장합니다.

다른 키 사용

암호화에 사용하는 키를 변경하려면 다음 단계를 따릅니다.

1. Azure AI 서비스 리소스로 이동한 다음 **암호화**를 선택합니다.
2. 새 키의 URI를 입력합니다. 또는 키 자격 증명 모음을 선택한 다음 새 키를 선택할 수 있습니다.
3. 변경 사항을 저장합니다.

고객 관리형 키 순환

규정 준수 정책에 따라 Key Vault에서 고객 관리형 키를 회전할 수 있습니다. 키가 회전되면 새 키 URI를 사용하도록 Azure AI 서비스 리소스를 업데이트해야 합니다. Azure Portal에서 새 버전의 키를 사용하도록 리소스를 업데이트하는 방법을 알아보려면 키 버전 [업데이트를 참조하세요](#).

키를 회전해도 리소스의 데이터 재암호화는 트리거되지 않습니다. 사용자는 추가적인 작업을 할 필요가 없습니다.

고객 관리형 키에 대한 액세스 권한 철회

고객 관리형 키에 대한 액세스 권한을 철회하려면 PowerShell 또는 Azure CLI를 사용합니다. 자세한 내용은 [Azure Key Vault PowerShell](#) 또는 [Azure Key Vault CLI](#)를 참조하세요. Azure AI 서비스에서는 암호화 키에 액세스할 수 없기 때문에 액세스 권한을 취소하면 Azure AI 서비스 리소스의 모든 데이터에 대한 액세스가 효과적으로 차단됩니다.

고객 관리형 키 사용 안 함

고객 관리형 키를 사용하지 않도록 설정하면 Azure AI 서비스 리소스가 Microsoft 관리형 키로 암호화됩니다. 고객 관리형 키를 사용하지 않도록 설정하려면 다음 단계를 수행합니다.

1. Azure AI 서비스 리소스로 이동한 다음 **암호화**를 선택합니다.
2. 자체 키 사용 옆에 있는 확인란의 선택을 취소합니다.

다음 단계

- [Azure Key Vault에 대해 자세히 알아보기](#)

Azure AI 서비스에 대한 요청 인증

아티클 • 2023. 10. 24.

Azure AI 서비스에 대한 각 요청에는 인증 헤더가 포함되어야 합니다. 이 헤더는 서비스 또는 서비스 그룹에 대한 구독을 확인하는 데 사용되는 리소스 키 또는 인증 토큰을 전달합니다. 이 문서에서는 요청을 인증하는 세 가지 방법과 각각에 대한 요구 사항에 대해 알아봅니다.

- 단일 서비스 또는 다중 서비스 리소스 키로 인증
- 토큰으로 인증
- Microsoft Entra ID로 인증

필수 조건

요청하기 전에 Azure 계정과 Azure AI 서비스 구독이 필요합니다. 계정이 이미 있는 경우 다음 섹션으로 건너뜁니다. 계정이 없는 경우 몇 분 만에 설정을 완료할 수 있는 가이드가 있습니다. [다중 서비스 리소스 만들기](#).

계정을 생성 [한 후 Azure Portal](#)에서 리소스 키를 얻을 수 있습니다.

인증 헤더

Azure AI 서비스에 사용할 수 있는 인증 헤더를 빠르게 검토해 보겠습니다.

[+] 테이블 확장

헤더	설명
Ocp-Apim-Subscription-Key	이 헤더를 사용하여 특정 서비스에 대한 리소스 키 또는 다중 서비스 리소스 키로 인증합니다.
Ocp-Apim-Subscription-Region	이 헤더는 번역기 서비스 와 함께 다중 서비스 리소스 키를 사용할 때만 필요합니다. 이 헤더를 사용하여 리소스 지역을 지정합니다.
Authorization	액세스 토큰을 사용하는 경우 이 헤더를 사용합니다. 토큰 교환을 수행하는 단계는 다음 섹션에 자세히 설명되어 있습니다. 제공된 값은 다음 형식 <code>Bearer <TOKEN></code> 을 따릅니다.

단일 서비스 리소스 키로 인증

첫 번째 옵션은 번역기와 같은 특정 서비스에 대한 리소스 키를 사용하여 요청을 인증하는 것입니다. 키는 만든 각 리소스에 대해 Azure Portal에서 사용할 수 있습니다. 리소스 키를 사용하여 요청을 인증하려면 `Ocp-Apim-Subscription-Key` 헤더로 전달되어야 합니다.

이 샘플 요청은 `Ocp-Apim-Subscription-Key` 헤더를 사용하는 방법을 보여 줍니다. 이 샘플을 사용할 때 유효한 리소스 키를 포함해야 한다는 점을 명심하세요.

번역기 서비스에 대한 샘플 호출입니다.

CURL

```
curl -X POST 'https://api.cognitive.microsofttranslator.com/translate?api-version=3.0&from=en&to=de' \
-H 'Ocp-Apim-Subscription-Key: YOUR_SUBSCRIPTION_KEY' \
-H 'Content-Type: application/json' \
--data-raw '[{"text": "How much for the cup of coffee?" }]' | json_pp
```

다음 비디오에서는 Azure AI 서비스 키를 사용하는 방법을 보여줍니다.

다중 서비스 리소스 키로 인증

다중 서비스 리소스 키를 사용하여 요청을 인증할 수 있습니다. 주요 차이점은 다중 서비스 리소스 키가 특정 서비스에 연결되지 않고, 오히려 단일 키를 사용하여 여러 Azure AI 서비스에 대한 요청을 인증할 수 있다는 것입니다. 지역별 가용성, 지원되는 기능 및 가격에 대한 자세한 내용은 [Azure AI 서비스 가격 책정](#)을 참조하세요.

리소스 키는 각 요청에 `Ocp-Apim-Subscription-Key` 헤더로 제공됩니다.



지원되는 지역

다중 서비스 리소스 키를 사용하여 `api.cognitive.microsoft.com`에 요청할 때 URL에 지역을 포함해야 합니다. 예: `westus.api.cognitive.microsoft.com`

Azure AI 번역기와 함께 다중 서비스 리소스 키를 사용하는 경우 `Ocp-Apim-Subscription-Region` 헤더로 리소스 지역을 지정해야 합니다.

다중 서비스 인증은 다음 지역에서 지원됩니다.

- `australiaeast`
- `brazilsouth`
- `canadacentral`
- `centralindia`
- `eastasia`
- `eastus`
- `japaneast`
- `northeurope`
- `southcentralus`
- `southeastasia`
- `uksouth`
- `westcentralus`
- `westeurope`
- `westus`
- `westus2`
- `francecentral`
- `koreacentral`
- `northcentralus`
- `southafricanorth`
- `uaenorth`
- `switzerlandnorth`

샘플 요청

번역기 서비스에 대한 샘플 호출입니다.

cURL

```
curl -X POST 'https://api.cognitive.microsofttranslator.com/translate?api-version=3.0&from=en&to=de' \
-H 'Ocp-Apim-Subscription-Key: YOUR_SUBSCRIPTION_KEY' \
```

```
-H 'Ocp-Apim-Subscription-Region: YOUR_SUBSCRIPTION_REGION' \
-H 'Content-Type: application/json' \
--data-binary '[{"text": "How much for the cup of coffee?"}]' | json_pp
```

액세스 토큰으로 인증

일부 Azure AI 서비스는 액세스 토큰을 허용하고 경우에 따라 요구합니다. 현재 다음 서비스는 액세스 토큰을 지원합니다.

- Text Translation API
- 음성 서비스: 음성 텍스트 변환 API
- 음성 서비스: 텍스트 음성 변환 API

① 참고

QnA Maker도 인증 헤더를 사용하지만 엔드포인트 키가 필요합니다. 자세한 내용은 [QnA Maker: 기술 자료 답변을 확인하세요.](#)

⚠ 경고

액세스 토큰을 지원하는 서비스는 시간이 지남에 따라 변경될 수 있으므로 이 인증 방법을 사용하기 전에 서비스에 대한 API 참조를 확인합니다.

단일 서비스 및 다중 서비스 리소스 키 모두 인증 토큰으로 교환될 수 있습니다. 인증 토큰은 10분 동안 유효합니다. JWT(JSON Web Token) 형식으로 저장되며 [JWT 라이브러리](#)를 사용하여 프로그래밍 방식으로 쿼리할 수 있습니다.

액세스 토큰은 요청에 `Authorization` 헤더로 포함됩니다. 제공되는 토큰 값은 `Bearer` 뒤에 나와야 합니다(예: `Bearer YOUR_AUTH_TOKEN`).

샘플 요청

액세스 토큰에 대한 리소스 키를 교환하려면 다음 URL을 사용하세요: `https://YOUR-REGION.api.cognitive.microsoft.com/sts/v1.0/issueToken`.

cURL

```
curl -v -X POST \
"https://YOUR-REGION.api.cognitive.microsoft.com/sts/v1.0/issueToken" \
-H "Content-type: application/x-www-form-urlencoded" \
```

```
-H "Content-length: 0" \
-H "Ocp-Apim-Subscription-Key: YOUR_SUBSCRIPTION_KEY"
```

токен 교환을 지원하는 다중 서비스 지역은 다음과 같습니다.

- `australiaeast`
- `brazilsouth`
- `canadacentral`
- `centralindia`
- `eastasia`
- `eastus`
- `japaneast`
- `northeurope`
- `southcentralus`
- `southeastasia`
- `uksouth`
- `westcentralus`
- `westeurope`
- `westus`
- `westus2`

액세스 토큰을 받은 후에는 각 요청에서 이를 `Authorization` 헤더로 전달해야 합니다. 번역기 서비스에 대한 샘플 호출입니다.

cURL

```
curl -X POST 'https://api.cognitive.microsofttranslator.com/translate?api-
version=3.0&from=en&to=de' \
-H 'Authorization: Bearer YOUR_AUTH_TOKEN' \
-H 'Content-Type: application/json' \
--data-binary '[{"text": "How much for the cup of coffee?"}]' | json_pp
```

Microsoft Entra ID로 인증

ⓘ 중요

Microsoft Entra 인증은 항상 Azure 리소스의 사용자 지정 하위 기본 이름과 함께 사용해야 합니다. **지역 엔드포인트는 Microsoft Entra 인증을 지원하지 않습니다.**

이전 섹션에서는 단일 서비스 또는 다중 서비스 구독 키를 사용하여 Azure AI 서비스에 대해 인증하는 방법을 보여주었습니다. 이러한 키를 이용하면 빠르고 쉽게 개발을 시작할 수 있지만 [Azure RBAC\(역할 기반 액세스 제어\)](#)를 필요로 하는 더 복잡한 시나리오에서는 한계가 있습니다. Microsoft Entra ID를 사용하여 인증하는 데 필요한 항목을 살펴보겠습니다.

다음 섹션에서는 Azure Cloud Shell 환경 또는 Azure CLI를 사용하여 하위 도메인을 만들고, 역할을 할당하고, Azure AI 서비스를 호출하기 위한 전달자 토큰을 얻습니다. 문제가 발생하면 Azure Cloud Shell/Azure CLI의 각 명령에 대해 사용 가능한 모든 옵션이 포함된 링크가 각 섹션에 제공됩니다.

① 중요

조직에서 Microsoft Entra ID를 통해 인증을 수행하는 경우 조직의 사용자가 항상 Microsoft Entra ID **를 사용해야 하므로 로컬 인증**(키 인증)을 사용하지 않도록 설정해야 합니다.

사용자 지정 하위 도메인을 사용하여 리소스 만들기

첫 번째 단계는 사용자 지정 하위 기본 만드는 것입니다. 사용자 지정 하위 도메인 이름이 없는 기존 Azure AI 서비스 리소스를 사용하려면 [Azure AI 서비스 사용자 지정 하위 도메인](#)의 지침에 따라 리소스에 대한 사용자 지정 하위 도메인을 활성화하세요.

- 먼저 Azure Cloud Shell을 엽니다. 그런 다음, [구독](#)을 선택합니다.

PowerShell

```
Set-AzContext -SubscriptionName <SubscriptionName>
```

- 그런 다음 사용자 지정 하위 도메인을 사용하여 [Azure AI 서비스 리소스를 생성](#)합니다. 하위 도메인 이름은 전역적으로 고유해야 하며 특수 문자(예: ".", "!", ",")를 포함할 수 없습니다.

PowerShell

```
$account = New-AzCognitiveServicesAccount -ResourceGroupName <RESOURCE_GROUP_NAME> -name <ACCOUNT_NAME> -Type <ACCOUNT_TYPE> -SkuName <SUBSCRIPTION_TYPE> -Location <REGION> -CustomSubdomainName <UNIQUE_SUBDOMAIN>
```

- 성공하면 엔드포인트에 리소스에 고유한 하위 기본 이름이 표시됩니다.

서비스 주체에 역할 할당

이제 리소스와 연결된 사용자 지정 하위 기본 서비스 주체에 역할을 할당해야 합니다.

① 참고

Azure 역할 할당이 전파되는 데 최대 5분이 걸릴 수 있습니다.

- 먼저 Microsoft Entra 애플리케이션을 등록해 보겠습니다.

PowerShell

```
$SecureStringPassword = ConvertTo-SecureString -String <YOUR_PASSWORD>  
-AsPlainText -Force  
  
$app = New-AzureADApplication -DisplayName <APP_DISPLAY_NAME> -  
IdentifierUris <APP_URIS> -PasswordCredentials $SecureStringPassword
```

다음 단계에서 ApplicationId가 필요합니다.

- 다음으로 Microsoft Entra 애플리케이션에 대한 서비스 주체를 만들어야 합니다.

PowerShell

```
New-AzADServicePrincipal -ApplicationId <APPLICATION_ID>
```

① 참고

Azure Portal에서 애플리케이션을 등록하는 경우 이 단계가 완료됩니다.

- 마지막 단계는 "Cognitive Services 사용자" 역할을 서비스 주체(리소스로 범위 지정)에 할당하는 것입니다. 역할을 할당하면 서비스 주체에게 이 리소스에 대한 액세스 권한을 부여합니다. 동일한 서비스 주체에게 구독에서 여러 리소스에 대한 액세스 권한을 부여할 수 있습니다.

① 참고

서비스 주체의 ObjectId는 애플리케이션의 ObjectId가 아니라 사용됩니다. ACCOUNT_ID는 사용자가 만든 Azure AI 서비스 계정의 Azure 리소스 ID입니다. Azure Portal에서 리소스의 "속성"에서 Azure 리소스 ID를 찾을 수 있습니다.

Azure CLI

```
New-AzRoleAssignment -ObjectId <SERVICE_PRINCIPAL_OBJECTID> -Scope <ACCOUNT_ID> -RoleDefinitionName "Cognitive Services User"
```

샘플 요청

이 샘플에서는 암호가 서비스 주체를 인증하는 데 사용됩니다. 그런 다음, 제공된 토큰을 사용하여 Computer Vision API를 호출합니다.

1. TenantId 가져오기:

PowerShell

```
$context=Get-AzContext  
$context.Tenant.Id
```

2. 토큰 가져오기:

PowerShell

```
$tenantId = $context.Tenant.Id  
$clientId = $app.ApplicationId  
$clientSecret = "<YOUR_PASSWORD>"  
$resourceUrl = "https://cognitiveservices.azure.com/"  
  
$tokenEndpoint =  
"https://login.microsoftonline.com/$tenantId/oauth2/token"  
$body = @{  
    grant_type      = "client_credentials"  
    client_id      = $clientId  
    client_secret   = $clientSecret  
    resource       = $resourceUrl  
}  
  
$responseToken = Invoke-RestMethod -Uri $tokenEndpoint -Method Post -  
Body $body  
$accessToken = $responseToken.access_token
```

① 참고

스크립트에서 암호를 사용할 때마다 가장 안전한 옵션은 PowerShell 비밀 관리 모듈을 사용하고 Azure KeyVault와 같은 솔루션과 통합하는 것입니다.

3. Computer Vision API를 호출합니다.

PowerShell

```
$url = $account.Endpoint+"vision/v1.0/models"
$result = Invoke-RestMethod -Uri $url -Method Get -Headers
@{"Authorization"="Bearer $accessToken"} -Verbose
$result | ConvertTo-Json
```

또는 인증서를 사용하여 서비스 주체를 인증할 수 있습니다. 서비스 주체 외에도 다른 Microsoft Entra 애플리케이션을 통해 권한을 위임하여 사용자 보안 주체도 지원됩니다. 이 경우 암호나 인증서 대신 토큰을 획득할 때 사용자에게 2단계 인증을 요청하는 메시지가 표시됩니다.

관리 ID에 대한 액세스 권한 부여

Azure AI 서비스는 Azure 리소스에 대한 관리 ID를 사용하여 [Microsoft Entra 인증을 지원합니다](#). Azure 리소스에 대한 관리 ID는 Azure VM(가상 머신), 함수 앱, 가상 머신 확장 집합 및 기타 서비스에서 실행되는 애플리케이션에서 Microsoft Entra 자격 증명을 사용하여 Azure AI 서비스 리소스에 대한 액세스 권한을 부여할 수 있습니다. Microsoft Entra 인증과 함께 Azure 리소스에 대한 관리 ID를 사용하면 클라우드에서 실행되는 애플리케이션에 자격 증명을 저장하지 않아도 됩니다.

VM에서 관리 ID 사용

Azure 리소스에 대한 관리 ID를 사용하여 VM에서 Azure AI 서비스 리소스에 대한 액세스 권한을 부여하려면 먼저 VM에서 Azure 리소스에 대한 관리 ID를 사용하도록 설정해야 합니다. Azure 리소스의 관리 ID를 사용하도록 설정하는 방법을 알아보려면 다음을 참조하세요.

- [Azure Portal](#)
- [Azure PowerShell](#)
- [Azure CLI](#)
- [Azure Resource Manager 템플릿](#)
- [Azure Resource Manager 클라이언트 라이브러리](#)

관리 ID에 대한 자세한 내용은 [Azure 리소스의 관리 ID](#)를 참조하세요.

Azure 키 자격 증명 모음을 사용하여 자격 증명에 안전하게 액세스

[Azure Key Vault](#)를 사용하여 Azure AI 서비스 애플리케이션을 안전하게 개발할 수 있습니다. Key Vault를 사용하면 인증 자격 증명을 클라우드에 저장할 수 있으며 애플리케이션

에 보안 정보를 저장하지 않으므로 비밀이 실수로 유출될 가능성을 줄일 수 있습니다.

인증이 Microsoft Entra ID를 통해 수행됩니다. 권한 부여는 Azure RBAC(Azure 역할 기반 액세스 제어) 또는 Key Vault 액세스 정책을 통해 수행할 수 있습니다. Azure RBAC는 자격 증명 모음의 관리와 자격 증명 모음에 저장된 데이터에 액세스하는 데 모두 사용할 수 있으며, 키 자격 증명 모음 액세스 정책은 자격 증명 모음에 저장된 데이터에 액세스하려고 할 때만 사용할 수 있습니다.

참고 항목

- [Azure AI 서비스란?](#)
- [Azure AI 서비스 가격 ↗](#)
- [사용자 지정 하위 기본](#)

컨테이너 설치 및 실행

아티클 • 2024. 02. 28.

컨테이너에 대한 지원은 현재 모든 모델의 경우 문서 인텔리전스 버전 2022-08-31 (GA) 와 읽기 및 레이아웃 전용의 경우 2023-07-31 (GA)에서 사용할 수 있습니다.

- REST API 2022-08-31 (GA) ↗
- REST API 2023-07-31 (GA) ↗
- SDK 대상 지정 REST API 2022-08-31 (GA)
- SDK 대상 지정 REST API 2023-07-31 (GA)

✓ 지원되는 컨테이너 설명서는 [문서 인텔리전스 v3.0 컨테이너 설치 및 실행](#)을 참조하세요.

문서 인텔리전스 컨테이너 구성

아티클 • 2023. 12. 15.

컨테이너에 대한 지원은 현재 Document Intelligence 버전 2022-08-31 (GA) 에서만 사용 할 수 있습니다.

- REST API 2022-08-31 (GA) ↗
- SDK 대상 지정 REST API 2022-08-31 (GA)

✓ 지원되는 컨테이너 설명서는 문서 인텔리전스 v3.0 컨테이너 [구성을 참조하세요](#).

문서 인텔리전스 컨테이너 태그

아티클 • 2024. 02. 07.

컨테이너에 대한 지원은 현재 모든 모델 및 읽기 및 2023-07-31 (GA) 레이아웃에 대해서만 Document Intelligence 버전 2022-08-31 (GA)에서 사용할 수 있습니다.

- REST API 2022-08-31 (GA) ↗
- REST API 2023-07-31 (GA) ↗
- SDK 대상 지정 REST API 2022-08-31 (GA)
- SDK 대상 지정 REST API 2023-07-31 (GA)

✓ 지원되는 컨테이너 설명서는 문서 인텔리전스 컨테이너 이미지 태그를 참조하세요.

연결이 끊긴(오프라인) 환경의 컨테이너

아티클 • 2023. 12. 15.

컨테이너에 대한 지원은 현재 Document Intelligence 버전 2022-08-31 (GA)에서 사용할 수 있습니다.

- REST API 2022-08-31 (GA) ↗
- SDK 대상 지정 REST API 2022-08-31 (GA)

✓ 지원되는 컨테이너 설명서는 연결이 끊긴 환경의 문서 인텔리전스 v3.0 컨테이너를 [참조하세요](#).

문서 처리 모델

아티클 • 2024. 04. 14.

① 중요

- 문서 인텔리전스 공개 미리 보기 릴리스에서는 현재 개발 중인 기능에 대한 조기 액세스를 제공합니다.
- 기능, 방식 및 프로세스는 GA(일반 공급) 전에 사용자 피드백에 따라 변경될 수 있습니다.
- 문서 인텔리전스 클라이언트 라이브러리의 공개 미리 보기 버전은 기본적으로 REST API 버전 [2024-02-29-미리 보기](#)로 설정됩니다.
- 공개 미리 보기 버전 [2024-02-29-미리 보기](#)는 현재 다음 Azure 지역에서만 사용할 수 있습니다.
 - 미국 동부
 - 미국 서부2
 - 서유럽

이 콘텐츠의 적용 대상은 다음과 같습니다. ✓ v4.0(미리 보기) | 이전 버전: ✓ v3.1(GA) ✓ v3.0(GA) ✓ v2.1(GA)

Azure AI 문서 인텔리전스는 앱 및 흐름에 지능형 문서 처리를 추가할 수 있는 다양한 모델을 지원합니다. 미리 빌드된 도메인별 모델을 사용하거나 특정 비즈니스 요구 사항과 사용 사례에 맞게 조정된 사용자 지정 모델을 학습시킬 수 있습니다. 문서 인텔리전스는 REST API 또는 Python, C#, Java 및 JavaScript 클라이언트 라이브러리와 함께 사용할 수 있습니다.

모델 개요

다음 표에서는 각 현재 미리 보기 및 안정적인 API에 사용 가능한 모델을 보여 줍니다.

[+] 테이블 확장

모델 형식	모델	• 2024-02-29-미리 보기 글머리 기호(&b) 2023-10-31-미리 보기	2023-07-31(GA)	2022-08-31(GA)	v2.1(GA)
문서 분석 모델	읽음	✓	✓	✓	해당 없음

모델 형식	모델	• 2024-02-29-미리 보기 글머리 기호(&b) 2023-10-31-미리 보기	2023-07-31(GA)	2022-08-31(GA) ↗	v2.1(GA) ↗
문서 분석 모델	레이아웃	✓	✓	✓	✓
문서 분석 모델	일반 문서	레이아웃으로 이동**	✓	✓	해당 없음
미리 빌드된 모델	계약	✓	✓	해당 없음	해당 없음
미리 빌드된 모델	의료 보험 카드	✓	✓	✓	해당 없음
미리 빌드된 모델	ID 문서	✓	✓	✓	✓
미리 빌드된 모델	청구서	✓	✓	✓	✓
미리 빌드된 모델	영수증	✓	✓	✓	✓
미리 빌드된 모델	US 1040 세금*	✓	✓	해당 없음	해당 없음
미리 빌드된 모델	US 1098 세금*	✓	해당 없음	해당 없음	해당 없음
미리 빌드된 모델	US 1099 세금*	✓	해당 없음	해당 없음	해당 없음
미리 빌드된 모델	미국 W2 세금	✓	✓	✓	해당 없음
미리 빌드된 모델	미국 모기지 1003 URLA	✓	해당 없음	해당 없음	해당 없음
미리 빌드된 모델	미국 모기지 1008 요약	✓	해당 없음	해당 없음	해당 없음
미리 빌드된 모델	미국 모기지 마감 공시	✓	해당 없음	해당 없음	해당 없음
미리 빌드된 모델	결혼 인증서	✓	해당 없음	해당 없음	해당 없음
미리 빌드된 모델	신용 카드	✓	해당 없음	해당 없음	해당 없음
미리 빌드된 모델	명함	사용되지 않음	✓	✓	✓

모델 형식	모델	• 2024-02-29-미리 보기 글머리 기호(&b) 2023-10-31-미리 보기	2023-07-31(GA)	2022-08-31(GA)■	v2.1(GA)■
사용자 지정 분류 모델	사용자 지정 분류자	✓	✓	해당 없음	해당 없음
사용자 지정 추출 모델	사용자 지정 인공신경망	✓	✓	✓	해당 없음
Customextraction 모델	사용자 지정 템플릿	✓	✓	✓	✓
사용자 지정 추출 모델	사용자 지정 구성형	✓	✓	✓	✓
모든 모델	추가 기능	✓	✓	해당 없음	해당 없음

* - 하위 모델을 포함합니다. 지원되는 변형 및 하위 형식에 대한 모델 관련 정보를 참조하세요.

데이터 확장

추가 기능	추가 기능/무료	• 2024-02-29-미리 보기 • [2023-10-31-미리 보기] (/rest/api/aiservices/operation-groups? view=rest-aiservices-2024-02-29- preview&preserve-view=true	2023-07-31(GA)	2022-08-31(GA)■	v2.1(GA)■
Font 속성 추출	추가 기능	✓	✓	해당 없음	해당 없음
수식 추출	추가 기능	✓	✓	해당 없음	해당 없음
고해상도 추출	추가 기능	✓	✓	해당 없음	해당 없음
바코드 추출	Free	✓	✓	해당 없음	해당 없음
언어 감지	Free	✓	✓	해당 없음	해당 없음

추가 기능	추가 기능 / 무료	• 2024-02-29-미리 보기 • [2023-10-31-미리 보기] (/rest/api/aiservices/operation-groups? view=rest-aiservices-2024-02-29- preview&preserve-view=true	2023-07-31(GA)	2022-08-31(GA)	v2.1(GA) ↗
키 값 쌍	Free	✓	해당 없음	해당 없음	해당 없음
쿼리 필드	추가 기능 *	✓	해당 없음	해당 없음	해당 없음

모델 분석 기능

[+] 테이블 확장

Model ID	콘텐츠 추출	쿼리 필드	단락 역할	단락 활시	선택 표시	테이블 시	키/값 쌍	언어 코드	바코드 분석	문서 분석	수식 *	스타일 *	스릴 꼴	고해상도 *
prebuilt-read	✓						○	○		○	○	○		
prebuilt-layout	✓	✓	✓	✓	✓	✓			○	○		○	○	○
prebuilt-document	✓	✓	✓	✓	✓	✓	✓		○	○		○	○	○
prebuilt-businessCard	✓	✓									✓			
prebuilt-contract	✓	✓	✓	✓					○	○	✓	○	○	○
prebuilt-healthInsuranceCard.us	✓	✓							○	○	✓	○	○	○
prebuilt-idDocument	✓	✓							○	○	✓	○	○	○
prebuilt-invoice	✓	✓				✓	✓	○	○	○	✓	○	○	○
prebuilt-receipt	✓	✓							○	○	✓	○	○	○
prebuilt-marriageCertificate.us	✓	✓							○	○	✓	○	○	○
prebuilt-creditCard	✓	✓							○	○	✓	○	○	○
prebuilt-mortgage.us.1003	✓	✓							○	○	✓	○	○	○
prebuilt-mortgage.us.1008	✓	✓							○	○	✓	○	○	○

Model ID	콘텐츠 추출	쿼리 필드	단락 할	단락 표시	선택 표시	테이블	키/값 쌍	언어 쌍	바코드	문서 분석	수식 *	스타일 글꼴 *	고해상도 *
prebuilt-mortgage.us.closingDisclosure	✓	✓								○ ○ ✓ ○ ○ ○			
prebuilt-tax.us.w2	✓	✓				✓				○ ○ ✓ ○ ○ ○			
prebuilt-tax.us.1098	✓	✓				✓				○ ○ ✓ ○ ○ ○			
prebuilt-tax.us.1098E	✓	✓				✓				○ ○ ✓ ○ ○ ○			
prebuilt-tax.us.1098T	✓	✓				✓				○ ○ ✓ ○ ○ ○			
prebuilt-tax.us.1099(변형)	✓	✓				✓				○ ○ ✓ ○ ○ ○			
prebuilt-tax.us.1040(변형)	✓	✓								○ ○ ✓ ○ ○ ○			
{ customModelName }	✓	✓	✓	✓	✓	✓	✓			○ ○ ✓ ○ ○ ○			

✓ - 사용함

○ - 선택적

* - 프리미엄 기능에는 추가 비용이 발생합니다

추가 기능* - 쿼리 필드 가격은 다른 추가 기능과 다르게 책정됩니다. 자세한 내용은 [가격 책정](#) 을 참조하세요.

[+] 테이블 확장

모델	설명
문서 분석 모델	
OCR 읽기	단어, 위치 및 검색된 언어를 포함하여 인쇄 및 필기 텍스트를 추출합니다.
레이아웃 분석	테이블, 선택 표시, 제목, 섹션 제목 등과 같은 텍스트 및 문서 레이아웃 요소를 추출합니다.
미리 빌드된 모델	
의료 보험 카드	미국 의료 보험 카드에서 보험사, 회원, 처방전, 그룹 번호 및 기타 주요 정보를 추출하여 의료 프로세스를 자동화합니다.

모델	설명
미국 세금 문서 모델	U2 세금 양식을 처리하여 직원, 고용주, 임금 및 기타 정보를 추출합니다.
미국 모기지 문서 모델	미국 모기지 양식을 처리하여 대출자 대출 및 부동산 정보를 추출합니다.
계약	계약 및 당사자 세부 정보를 추출합니다.
청구서	청구서를 자동화합니다.
영수증	영수증에서 영수증 데이터를 추출합니다.
ID 문서(ID)	미국 운전면허증과 국가별 여권에서 ID(ID) 필드를 추출합니다.
명함	명함을 검사하여 주요 필드와 데이터를 애플리케이션으로 추출합니다.
사용자 지정 모델	<p>사용자 지정 모델(개요) 사용자 비즈니스와 관련된 양식 및 문서에서 데이터를 추출합니다. 사용자 지정 모델은 고객의 고유한 데이터 및 사용 사례를 학습합니다.</p> <p>사용자 지정 추출 모델</p> <ul style="list-style-type: none"> • 사용자 지정 템플릿 모델은 레이아웃 신호를 사용하여 문서에서 값을 추출하며, 정의된 시각적 템플릿을 사용하여 고도로 구조화된 문서에서 필드를 추출하는 데 적합합니다. • 사용자 지정 인공신경망 모델은 구조화된 문서, 반구조화된 문서 및 구조화되지 않은 문서에서 필드를 추출하기 위해 다양한 문서 유형에 대해 학습됩니다. <p>사용자 지정 분류 모델은 입력 파일의 각 페이지를 분류하여 문서를 식별할 수 있으며 입력 파일 내에서 여러 문서 또는 단일 문서의 여러 인스턴스를 식별할 수도 있습니다.</p> <p>구성형 모델 여러 사용자 지정 모델을 단일 모델로 결합하여 단일 구성 모델로 다양한 문서 형식 처리를 자동화합니다.</p>

명함 모델을 제외한 모든 모델에 대해 문서 인텔리전스는 이제 더 정교한 분석을 허용하는 추가 기능 기능을 지원합니다. 이러한 선택적 기능은 문서 추출 시나리오에 따라 활성화 및 비활성화할 수 있습니다. 2023-07-31(GA) 이상 API 버전에 사용할 수 있는 7가지 추가 기능입니다.

- [ocrHighResolution](#)
- [formulas](#)
- [styleFont](#)
- [barcodes](#)
- [languages](#)
- [keyValuePairs](#) (2024-02-29-미리 보기, 2023-10-31-미리 보기)

- queryFields (2024-02-29-미리 보기, 2023-10-31-미리 보기) Not available with the US.Tax models

모델 세부 정보

이 섹션에서는 각 모델에서 예상할 수 있는 출력에 대해 설명합니다. 추가 기능으로 대부분의 모델의 출력을 확장할 수 있습니다.

OCR 읽기



Read API는 줄, 단어, 위치, 감지된 언어, 필기 입력 스타일(감지된 경우)을 분석하고 추출합니다.

[문서 인텔리전스 스튜디오](#) 를 사용하여 처리한 문서 샘플:

Analyze API version: 2022-01-30-preview

Content
Result
Code

IRS-Unterstützung in IRS Katastrophenfällen

Von der US-Bundesregierung erklärttes Katastrophengebiet

Sie können den vollständigen oder teilweisen Verlust Ihres Wohngegenstums, Ihres Hausrats und Ihrer Kraftfahrzeuge infolge einer Beschädigung im Katastrophenfall auf Ihrer persönlichen Bundeseinkommenssteuererklärung absetzen. Wenn Sie im Steuerjahr unmittelbar vor dem Steuerjahr, in dem der Katastrophenfall eintrat, Steuern abgeführt haben, können Sie Ihren Verlust auf einem Formular 1040X (Amended U.S. Individual Income Tax Return – *Abgeänderte US-Einkommenssteuererklärung*) für das Vorjahr absetzen, anstatt zu warten, bis Sie Ihre Steuererklärung für das laufende Jahr einreichen. Auf diese Weise können Sie sich Ihre Steuern, die Sie in Verbindung mit Ihrer Steuererklärung des Vorjahrs bezahlt haben, teilweise oder ganz erstatten lassen.

Was das für Sie bedeutet

- Wenn Sie im vorausgegangenen Steuerjahr eine Bundeskommenssteuererklärung abgegeben und Bundessteuern bezahlt haben ...
- können Sie jetzt u. U. eine abgeänderte Steuererklärung einreichen (oder damit bis nächstes Jahr warten), um Ihren Verlust geltend zu machen und sich die bezahlten Steuern erstatten zu lassen.
- Sie müssen Ihre Abzüge auf Formular 1040, Aufstellung A, einzeln aufschlüsseln.

So machen Sie Ihren Verlust geltend

- Fertigen Sie eine Liste Ihres gesamten verloren gegangenen Besitzes an.
- Stellen Sie die Anschaffungskosten (bzw. die angepasste Steuerbemessungsgrundlage) fest.
- Stellen Sie den Verkehrswert jedes Besitzguts fest.
 - Dabei handelt es sich um den Betrag, zu dem das Besitzgut unmittelbar vor Eintreten des Katastrophenfalls hätte verkauft werden können.
- Stellen Sie den derzeitigen Wert – also nach Eintreten des Katastrophenfalls – fest.
- Stellen Sie Versicherungs- oder andere Erstattungsleistungen fest, die Sie bereits erhalten haben oder voraussichtlich erhalten werden.

So können Sie sich bei der Geltendmachung von Sachverlusten helfen lassen

- Besorgen Sie sich die IRS-Publikation 2194, *Disaster Resource Guide* (Hilfreiche Informationen und Materialien für Katastrophenfälle), für Einzelpersonen und Unternehmen.
- Besorgen Sie sich rechnergestellte Kopien Ihrer letztajährigen Steuererklärung vom IRS.
- Der IRS kann Ihnen beim Erstellen Ihrer abgeänderten Steuererklärungen behilflich sein.

So erhalten Sie weitere Informationen und Unterstützung

- IRS-Hotline für Hilfe im Katastrophenfall – 1-866-562-5227 (Montag bis Freitag von 7:00 bis 19:00 Uhr Ortszeit). Bitte treffen Sie vor dem Anruf bei der Hotline-Nummer bei Bedarf selbst Vorkehrungen für einen Dolmetscher.
- Besuchen Sie die IRS-Website unter www.irs.gov oder
- Wenden Sie sich an Ihren Steuerberater.

Publication 3067 EN-DE (Rev. 10-2017) Catalog Number 53671Z Department of the Treasury Internal Revenue Service www.irs.gov

자세히 알아보기: 읽기 모델

레이아웃 분석



레이아웃 분석 모델은 텍스트, 테이블, 선택 표시 및 제목, 섹션 머리글, 페이지 머리글, 페이지 바닥글 등과 같은 기타 구조 요소를 분석하고 추출합니다.

문서 인텔리전스 스튜디오 를 사용하여 처리한 문서 샘플:

The screenshot shows a Microsoft Word document with a news article titled "NEWS TODAY". The document includes a header with the date "Tuesday Sep 20, YYYY", a title "Mirjam Nilsson", and a subtitle "The scoop of the day The latest updates". There are several text blocks, a video placeholder, and a table. Callouts highlight specific features: "Video provides a powerful way to help you prove your point.", "To make your document look professionally produced, Word provides header, footer, cover page, and text box designs that complement each other.", "Themes and styles also help keep your document coordinated.", and "Save time in Word with new buttons that show up where you need them. To change". The right side of the screen displays the "Text" pane with the extracted structured data, including sections like "Text", "Tables", and "Selection marks", and specific items like "Issue #10", "Title", "Paragraph", and "SectionHeading".

자세히 알아보기: 레이아웃 모델

의료 보험 카드



의료 보험 카드 모델은 강력한 OCR(광학 인식) 기능과 딥 러닝 모델을 결합하여 미국 의료 보험 카드에서 주요 정보를 분석하고 추출합니다.

문서 인텔리전스 스튜디오 를 사용하여 처리한 미국 의료 보험 카드:

자세히 알아보기: 의료 보험 카드 모델

미국 세금 문서



미국 세금 문서 모델은 선택한 세금 문서 그룹에서 주요 필드와 품목을 분석하고 추출합니다. API는 전화 캡처 이미지, 스캔한 문서 및 디지털 PDF를 포함하여 다양한 형식과 품질의 영어 미국 세금 문서의 분석을 지원합니다. 현재 지원되는 모델은 다음과 같습니다.

테이블 확장

모델	설명	ModelID
미국 세금 W-2	과세 보상 세부 정보를 추출합니다.	prebuilt-tax.us.W-2
미국 세금 1040	모기지 이자 세부 정보를 추출합니다.	prebuilt-tax.us.1040(변형)
미국 세금 1098	모기지 이자 세부 정보를 추출합니다.	prebuilt-tax.us.1098(변형)
미국 세금 1099	고용주 이외의 출처에서 받은 소득을 추출합니다.	prebuilt-tax.us.1099(변형)

문서 인텔리전스 스튜디오 를 사용하여 처리한 W-2 문서 샘플:

Analyze API version: 2022-01-30-preview

a. Employee's Soc Sec No 123-45-6789	1 Wages, tips, other comp. 37160.56	2 Federal income tax withheld 3894.54			
b. Employer ID number (EIN) 98-7654321	3 Social security wages 37160.56	4 Social security tax withheld 2303.95			
c Employer's name, address and ZIP code CONTOSO LTD 123 MICROSOFT WAY REDMOND, WA 98765	5 Medicare wages and tips 37160.56	6 Medicare tax withheld 538.83			
d Control Number 000086242					
e Employee's name, address, and ZIP code ANGEL BROWN 4567 MAIN STREET BUFFALO, WA 12345					
f	7 Social security tips 302.30	8 Allocated tips 874.20			
g	10 Dependent care benefits 9873.20	11 Nonqualified plans 653.21			
h	13 Statutory employee x	12a Code See Inst. for box 12 DD 6939.68			
i	Retirement plan x	12b Code B 5432.00			
j	Third-party sick pay x	12c Code D 876.30			
k	PA 87654321	12d Code C 123.30			
l	WA 12345678	m	15 State Employer's state ID number 37160.56	16 State wages, tips, etc. 9631.20	17 State income tax 1032.30
n	18 Local wages, tips, etc. 37160.56	o	19 Local income tax 51.00	p	20 Locality name Cumberland Vly/Mdd
q	37160.56	r	594.54	s	E.Pennsboro/E.Pnns

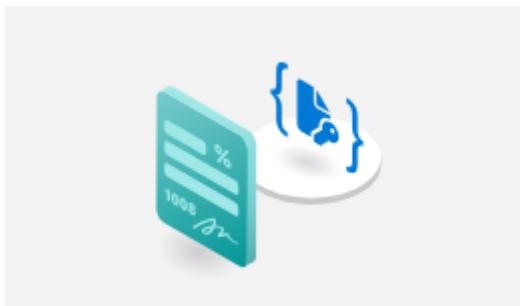
< 1 of 1 >

Fields Result Code

- DocType: tax.us.w2
- AdditionalInfo (4) #1
- AllocatedTips #1 100.00%
874.2
- ControlNumber #1 100.00%
000086242
- DependentCareBenefits #1 100.00%
9873.2
- Employee #1
- Employer #1
- FederalIncomeTaxWithheld #1 100.00%
3894.54
- LocalTaxInfos (2) #1
- MedicareTaxWithheld #1 100.00%
538.83
- MedicareWagesAndTips #1 100.00%

자세한 정보: 세금 문서 모델

미국 모기지 문서



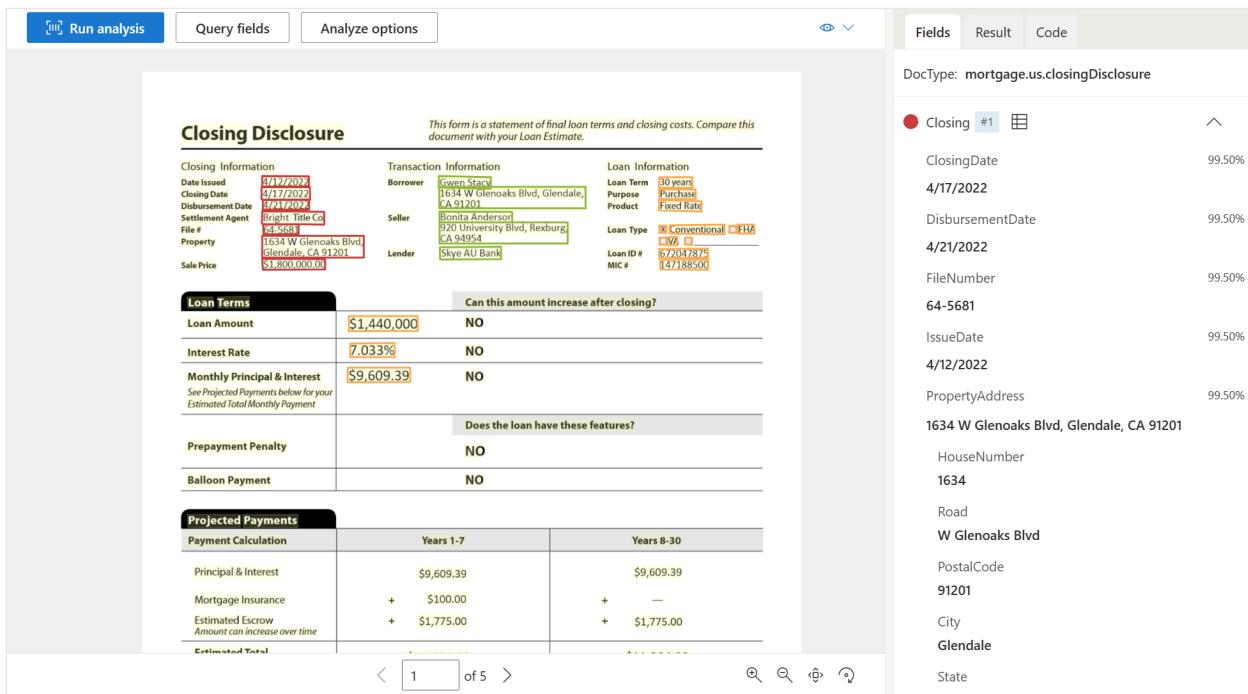
미국 모기지 문서 모델은 선택한 모기지 문서 그룹에서 대출자, 대출 및 부동산 정보를 포함한 주요 분야를 분석하고 추출합니다. API는 전화 캡처 이미지, 스캔한 문서 및 디지털 PDF를 포함하여 다양한 형식과 품질의 영어 미국 모기지 문서의 분석을 지원합니다. 현재 지원되는 모델은 다음과 같습니다.

테이블 확장

모델	설명	ModelID
1003 EULA(최종 사용자 사용권 계약)	대출, 대출자, 부동산 세부 정보를 추출합니다.	prebuilt-mortgage.us.1003
1008 요약 문서	대출자, 판매자, 부동산, 모기지 및 인수 세부 정보를 추출합니다.	prebuilt-mortgage.us.1008
마감 공시	마감, 거래 비용 및 대출 세부 정보	prebuilt-

모델	설명	ModelID
	를 추출합니다.	mortgage.us.closingDisclosure
결혼 증명서	공동 대출 신청자에 대한 결혼 정보 세부 정보를 추출합니다.	prebuilt-marriageCertificate
미국 세금 W-2	소득 확인을 위해 과세 보상 세부 정보를 추출합니다.	prebuilt-tax.us.W-2

문서 인텔리전스 스튜디오  를 사용하여 처리된 샘플 닫기 공개 문서:



The screenshot shows the Intelius Studio interface with two main panes. On the left, a 'Closing Disclosure' form is displayed. It includes sections for Closing Information (Date Issued: 4/17/2022, Closing Date: 4/17/2022, Disbursement Date: 4/21/2022, Settlement Agent: Bright Title Co., File # 64-5681, Property: 1634 W Glenoaks Blvd, Glendale, CA 91201, Sale Price: \$1,800,000.00), Transaction Information (Borrower: Gwen Stack, 1634 W Glenoaks Blvd, Glendale, CA 91201; Seller: Bonita Anderson, 920 University Blvd, Rexburg, ID 83495-4, Lender: Skye AU Bank), and Loan Information (Loan Term: 30 years, Purpose: Purchase, Product: Fixed Rate, Type: Conventional, MIC #: 672047875, MIC #: 147188500). Below these are sections for Loan Terms (Amount: \$1,440,000, Interest Rate: 7.033%, Monthly Principal & Interest: \$9,609.39), Prepayment Penalty, and Balloon Payment. A Projected Payments table shows payment calculations for Years 1-7 and Years 8-30. At the bottom, there is a page navigation bar showing page 1 of 5. On the right, the extracted data is shown in a table format under the 'Fields' tab, with columns for Fields, Result, and Code. The 'DocType' is identified as 'mortgage.us.closingDisclosure'. The results include fields like ClosingDate (99.50%), DisbursementDate (99.50%), FileNumber (99.50%), IssueDate (99.50%), PropertyAddress (99.50%), HouseNumber (1634), Road (W Glenoaks Blvd), PostalCode (91201), City (Glendale), and State (CA).

Fields	Result	Code
DocType	mortgage.us.closingDisclosure	
Closing	#1	
ClosingDate	99.50%	4/17/2022
DisbursementDate	99.50%	4/21/2022
FileNumber	99.50%	64-5681
IssueDate	99.50%	4/12/2022
PropertyAddress	99.50%	1634 W Glenoaks Blvd, Glendale, CA 91201
HouseNumber		1634
Road		W Glenoaks Blvd
PostalCode		91201
City		Glendale
State		CA

자세히 알아보기: 모기지 문서 모델

Contract



계약 모델은 당사자, 관할권, 계약 ID 및 타이틀을 포함한 체결한 계약에서 주요 필드 및 품목을 분석하고 추출합니다. 이 모델은 현재 영어 계약 문서를 지원합니다.

문서 인텔리전스 스튜디오를 사용하여 처리한 계약 샘플 :

Run analysis Analyze options Fields Content Result Code

DocType: contract

Jurisdictions #1

Clause

This Agreement shall be governed by and construed in accordance with the internal laws of the State of Washington applicable to agreements made and to be performed entirely within such state.

Region

Washington

Parties (2) #1

EffectiveDate #1 99.90% 15 day of October, 2022

ExecutionDate #1 99.90% 15 day of October, 2022

Title #1 99.90% WEB HOSTING AGREEMENT

WEB HOSTING AGREEMENT

This web Hosting Agreement is entered as of this 15 day of October 2022.

Effective Date by and between Contoso Corporation a Washington corporation having its principal place of business at 1 Microsoft Way, Redmond, Washington, 98058 Contoso and AdventureWorks Cycles a Washington corporation, having its principal place of business at 900 NW 76th Street, Suite 54, Bellevue, Washington, 98007 AdventureWorks.

This agreement shall void and nullify any and all previous agreements to this date between Contoso and AdventureWorks.

There shall be no additional fees of any kind paid to Contoso, other than those stated within this agreement for software usage and/or bandwidth usage. AdventureWorks agrees to pay Contoso \$0.01 (one cent) per access up to 400,000 accesses thereafter payment shall be \$0.005 (one-half cent) per access. AdventureWorks shall send this amount to Contoso by no later than Wednesday for accesses used from the previous week (Monday thru Sunday).

Contoso must provide a person(s) to correct any technical problems (Server being down or inaccessible) 24 hours per day, 7 days per week. This person(s) must be available by beeper or telephone. AdventureWorks shall provide this same 24 hour service at the broadcast location.

This Agreement shall be governed by and construed in accordance with the internal laws of the State of Washington applicable to agreements made and to be performed entirely within such state.

All parties have read and fully agree to all terms and conditions as set forth in this Web Hosting Agreement.

Contoso Corporation
By: Angel Brown
Title: CTO
Angel Brown

Adventure Works Cycle
By: Aaron Smith
Title: CEO
Aaron Smith

자세히 알아보기: 계약 모델

송장



청구서 모델은 청구서 처리를 자동화하여 고객 이름, 청구 주소, 기한, 지불액, 품목 및 기타 주요 데이터를 추출합니다. 현재 이 모델은 영어, 스페인어, 독일어, 프랑스어, 이탈리아어, 포르투갈어 및 네덜란드어 청구서를 지원합니다.

문서 인텔리전스 스튜디오 를 사용하여 처리한 청구서 샘플:

Analyze API version: 2021-09-30-preview

CONTOSO LTD.

INVOICE

INVOICE: #1
DUE DATE: 2021-09-30
CUSTOMER NAME: Microsoft Corp - Redmond
SERVICE PERIOD: 2021-09-01 - 2021-09-30
CUSTOMER ID: CID-12345

BILL TO: Microsoft Finance
SHIP TO: Microsoft Delivery
SERVICE ADDRESS: Microsoft Service Center
SALESPERSON: UNKNOWN
P.O. NUMBER: UNKNOWN
REQUISITIONER: UNKNOWN
SHIPPED VIA: UNKNOWN
F.O.B. POINT: UNKNOWN
TERMS: UNKNOWN

DATE	ITEM CODE	DESCRIPTION	QTY	UM	PRICE	TAX	AMOUNT
2021-09-01	ITEM1	Microsoft Product A	1	PC	\$1000	\$100	\$1100
2021-09-01	ITEM2	Microsoft Product B	1	PC	\$1000	\$100	\$1100
2021-09-01	ITEM3	Microsoft Product C	1	PC	\$1000	\$100	\$1100

SUBTOTAL: \$3300
SALES TAX: \$330
TOTAL: \$3630
PREVIOUS UNPAID BALANCE: \$0
AMOUNT DUE: \$3630

THANK YOU FOR YOUR BUSINESS!

REMIT TO: Microsoft Finance
123 Remit St
New York, NY, 10001

Values Result Code

- AmountDue #1 97.30%
- 610
- BillingAddress #1 94.70%
- 123 Bill St, Redmond WA, 98052
- BillingAddressRecipient #1 95.70%
- Microsoft Finance
- CustomerAddress #1 94.70%
- 123 Other St, Redmond WA, 98052
- CustomerAddressRecipient #1 95.60%
- Microsoft Corp
- CustomerId #1 96.40%
- CID-12345

자세히 알아보기: 청구서 모델

Receipt



영수증 모델을 사용하여 판매자 이름, 날짜, 항목, 수량, 인쇄 및 필기 영수증의 합계에 대한 판매 영수증을 검사합니다. 버전 v3.0은 단일 페이지 호텔 영수증 처리도 지원합니다.

[문서 인텔리전스 스튜디오](#) 를 사용하여 처리한 영수증 샘플.

Analyze API version: 2021-09-30-preview

Values	Result	Code
● Items (2) #1		
1 Name	Surface Pro 6	98.40%
Quantity		96.90%
1		
TotalPrice		98.50%
999		
2 Name	SurfacePen	96.90%
Quantity		97.00%
1		
TotalPrice		98.50%
99.99		
● Locale		98.70%

자세히 알아보기: 영수증 모델

ID 문서(ID)



ID 문서(ID) 모델을 사용하여 미국 운전 면허증(50개 주 전체 및 컬럼비아 특별구) 및 국가 별 여권의 전기 페이지(비자 및 기타 여행 문서 제외)를 처리하여 키 필드를 추출합니다.

[문서 인텔리전스 스튜디오](#) 를 사용하여 처리한 미국 운전면허증 샘플:

Analyze

Values Result Code

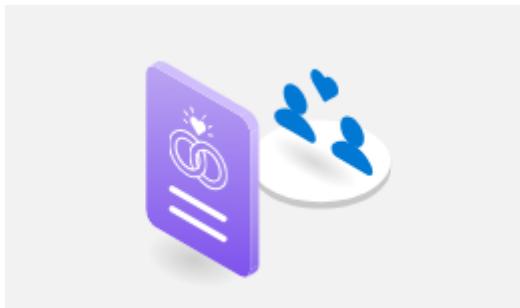
Address #1	123 STREET ADDRESS YOUR CITY WA 99999-1234	87.20%
CountryRegion	USA	99.50%
DateOfBirth #1	1958-01-06	98.90%
DateOfExpiration #1	2020-08-12	98.60%
DocumentNumber #1	WDLABCD456DG	97.50%
Endorsements #1	L	98.40%
FirstName #1	LIAM R.	84.60%
LastName #1	TALBOT	93.10%
Locale	en-US	100.00%

WA USA WASHINGTON DRIVER LICENSE FEDERAL LIMITS APPLY
4d LIC# WDLABCD456DG 9 CLASS B DONOR
1 TALBOT
2 LIAM R.
3 DOB 01/06/1958 4a ISS 01/06/2015
8 123 STREET ADDRESS YOUR CITY WA 99999-1234
15 SEX M 18 EYES BLU
16 HGT 5'-08" 17 WGT 165 lb
12 RESTRICTIONS 9a END L
4b EXP 08/12/2020
5 DDWDLABCD456DG1234567XX1101 Veteran
REV 07/01/2018

< 1 of 1 >

자세히 알아보기: ID 문서 모델

결혼 증명서



결혼 인증서 모델을 사용하여 미국 결혼 인증서를 처리하여 개인, 날짜 및 위치를 포함한 주요 필드를 추출합니다.

문서 인텔리전스 스튜디오 [를 사용하여 처리된 샘플 미국 결혼 인증서:](#)

The screenshot shows a Microsoft Power BI dashboard for analyzing a Michigan marriage license. The left side displays the original document with several fields redacted with a blue box. The right side shows the results of the entity extraction process:

- DocType:** marriageCertificate.us
- IssueDate #1:** March 22, 2017 (99.50%)
- MarriageDate #1:** 19th March AD. 2017 (98.60%)
- MarriagePlace #1:** Detroit Wayne MICHIGAN, (99.50%)
- Spouse1Address #1:** Detroit, Michigan (99.50%)
- Spouse1Age #1:** 30 (99.50%)
- Spouse1BirthPlace #1:** Detroit, Michigan (99.50%)

자세히 알아보기: ID 문서 모델

신용 카드



신용 카드 모델을 사용하여 신용 카드 및 직불 카드를 처리하여 키 필드를 추출합니다.

문서 인텔리전스 스튜디오 [를](#) 사용하여 처리된 **샘플 신용 카드**:

The screenshot shows a user interface for document analysis. At the top, there are buttons for "Run analysis", "Query fields", and "Analyze options". To the right, there are tabs for "Fields", "Result", and "Code".

Result Tab:

- DocType: creditCard
- CardHolderName #1: ADAM SMITH (99.50%)
- CardNumber #1: 5412 1234 5656 8888 (99.50%)
- CardVerificationValue #1: 123 (99.50%)
- CustomerServicePhoneNumbers (2) #1:
 - +1 200-345-6789
 - +1 200-000-8888
- ExpirationDate #1: 01/28 (99.50%)
- IssuingBank #1: Contoso Bank (99.50%)
- PaymentNetwork #1: mastercard (99.10%)

Code Tab:

```

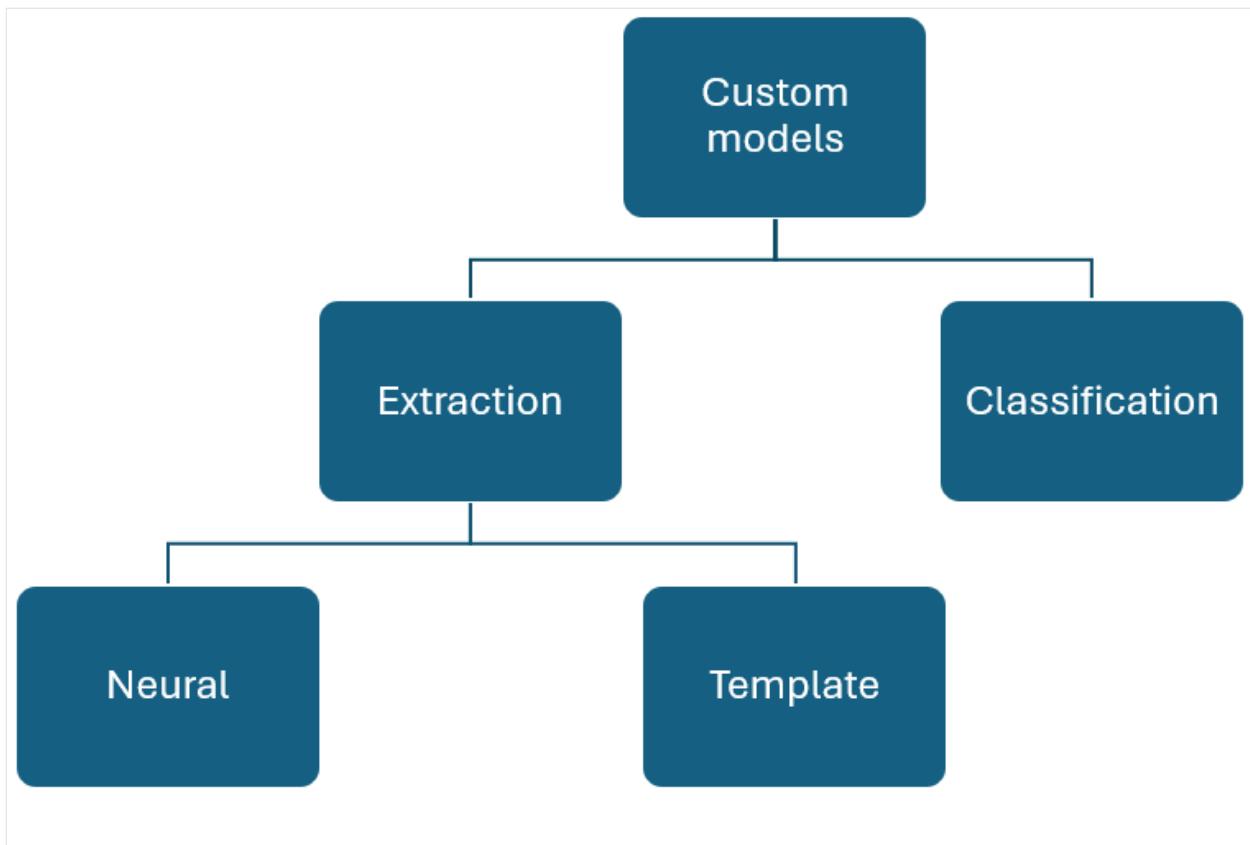
// Extracted code from the analysis results
const creditCard = {
  CardHolderName: "ADAM SMITH",
  CardNumber: "5412 1234 5656 8888",
  CardVerificationValue: "123",
  CustomerServicePhoneNumbers: ["+1 200-345-6789", "+1 200-000-8888"],
  ExpirationDate: "01/28",
  IssuingBank: "Contoso Bank",
  PaymentNetwork: "mastercard"
}
  
```

자세히 알아보기: ID 문서 모델

사용자 지정 모델



사용자 지정 모델은 크게 두 가지 유형으로 분류할 수 있습니다. 특정 문서 형식에서 정의된 스키마를 추출할 수 있는 "문서 형식" 및 사용자 지정 추출 모델의 분류를 지원하는 사용자 지정 분류 모델입니다.



사용자 지정 문서 모델은 사용자의 비즈니스에 특정한 양식 및 문서에서 데이터를 분석하고 추출합니다. 고유 콘텐츠 내의 양식 필드를 인식하고 키-값 쌍과 테이블 데이터를 추출하도록 학습되었습니다. 시작하려면 양식 형식의 한 가지 예제만 있으면 됩니다.

버전 v3.0 사용자 지정 모델은 템플릿 및 인공신경망 모델 모두에서 사용자 지정 템플릿(양식) 및 페이지 간 테이블에서 서명 검색을 지원합니다.

[문서 인텔리전스 스튜디오](#) 를 사용하여 처리한 사용자 지정 템플릿 샘플:

Label data

The screenshot shows the Microsoft Labeling interface. On the left, there are four thumbnail previews of invoices labeled 'Invoice_1.pdf', 'Invoice_2.pdf', 'Invoice_3.pdf', and 'Invoice_4.pdf'. In the center, a detailed view of 'Invoice_1.pdf' is shown, displaying the header 'Contoso' and 'Invoice For: Microsoft', the address '1 Redmond way Suite 6000 Redmond, WA 99243', and the body of the invoice with columns for 'Invoice Number', 'Invoice Date', 'Invoice Due Date', 'Charges', and 'VAT ID'. To the right, a 'Train' button is at the top, followed by a list of labeled data points:

- Receipt No: 2468 (Red circle)
- Sold To: Fabrikam Residences (Orange circle)
- ID #: 1197531 (Green circle)
- Live Delivery?: unselected (Dark Blue circle)
- Online Delivery?: selected (Light Blue circle)
- Video Delivery?: selected (Purple circle)

자세히 알아보기: 사용자 지정 모델

사용자 지정 추출



사용자 지정 추출 모델은 사용자 지정 템플릿 또는 사용자 지정 인공신경망의 두 가지 유형 중 하나일 수 있습니다. 사용자 지정 추출 모델을 만들려면 추출하려는 값으로 문서의 데이터 세트에 레이블을 지정하고 레이블이 지정된 데이터 세트를 모델에 학습시킵니다. 시작하려면 동일한 양식 또는 문서 유형의 5개 예제만 있으면 됩니다.

문서 인텔리전스 스튜디오 를 사용하여 처리한 사용자 지정 추출 샘플

Custom extraction models

Extract information from forms and documents with custom extraction models. Train a model by labeling as few as 5 example documents. (The same labeled dataset can train all types of custom extraction models.) Learn more about custom extraction models.

Template (Custom form) models

Template models work well when the target documents share a common visual layout. Training only takes a few minutes, and more than 100 languages are supported.

Request for Taxpayer Identification Number and Certification

Template models work well when the target documents share a common visual layout. Training only takes a few minutes, and more than 100 languages are supported.

Neural (Custom document) models

Neural models can flexibly handle both structured and unstructured documents. Training takes up to half an hour, and currently only English language documents are supported. The current version can extract inline field data and checkboxes. Neural models are available only in select regions. [Click here](#) for details.

HOUSE RENTAL AGREEMENT

This House Rental Agreement ("Agreement," "rental agreement," or "lease") is entered into between **Landlord** (**Landlord**) and **Tenant** (**Tenant**). If more than one person is named as Tenant they shall be jointly and severally liable and responsible under the terms of this Agreement. This lease Agreement involves a residential house, yard, and related facilities located at **Address** (**Address**) (the "premises"). The date of this Agreement is **Date**.

1. Landlord rents to Tenant, unfurnished, the premises on a month to month basis, terminable by either party at the end of any calendar month on at least 30 days notice to the other party. Tenant shall be entitled to possession of the premises and rent shall commence on **Apartment Address**. Tenant shall not assign, sublease, or allow anyone other than persons permitted under this lease to at any time be in possession of any portion of the premises. Landlord will provide five (\$5)

[자세한 정보: 사용자 지정 템플릿 모델](#)

[자세한 정보: 사용자 지정 인공신경망 모델](#)

사용자 지정 분류자



사용자 지정 분류 모델을 사용하면 추출 모델을 호출하기 전에 문서 유형을 식별할 수 있습니다. 분류 모델은 **2023-07-31 (GA)** API부터 사용할 수 있습니다. 사용자 지정 분류 모델을 학습시키려면 두 개 이상의 고유 클래스와 클래스당 최소 5개의 샘플이 필요합니다.

[자세한 정보: 사용자 지정 분류 모델](#)

구성된 모델

구성형 모델은 사용자 지정 모델 컬렉션을 가져와서 사용자의 양식 유형에서 빌드된 단일 모델에 할당하여 만듭니다. 단일 모델 ID로 호출되는 구성 모델에 여러 사용자 지정 모델을 할당할 수 있습니다. 구성된 단일 모델에 최대 200개의 학습시킨 사용자 지정 모델을 할당할 수 있습니다.

[문서 인텔리전스 스튜디오에서 구성된 모델 대화창](#)

Cognitive Services | Form Recognizer Studio - Preview

Custom Form

composed

Label data

Models

Test

Settings

Form Recognizer Studio > Custom Form > composed > Models

Compose Test Download Delete

Model ID	Model Description
8aa16866-16fe-44ca-b13a-8bfc6ad1d	
773fb140-f173-47a2-8aa9-a5fce1ceb	
4c493f98-87c3-4f6d-b0d8-3a1aab49e	

[자세히 알아보기: 사용자 지정 모델](#)

입력 요구 사항

- 최상의 결과를 위해 문서당 하나의 명확한 사진 또는 고품질 스캔을 제공합니다.
- 지원 파일 형식:

테이블 확장

모델	PDF	이미지: JPEG/JPG, PNG, BMP, TIFF, HEIF	Microsoft Office: Word(DOCX), Excel(XLSX), PowerPoint(PPTX) 및 HTML
읽음	✓	✓	✓
레이아웃	✓	✓	✓ (2024-02-29-미리 보기, 2023-10-31-미리 보기)
일반 문서	✓	✓	
사전 제작	✓	✓	
사용자 지정 추출	✓	✓	
사용자 지정 분류	✓	✓	✓ (2024-02-29-미리 보기)

- PDF 및 TIFF의 경우 최대 2000페이지를 처리할 수 있습니다(무료 계층 구독의 경우 처음 2페이지만 처리됨).
- 문서 분석을 위한 파일 크기는 유료(S0) 계층의 경우 500MB, 무료(F0) 계층의 경우 4MB입니다.
- 이미지 크기는 50 x 50픽셀에서 10,000 x 10,000픽셀 사이여야 합니다.
- PDF가 암호로 잠긴 경우에는 제출하기 전에 잠금을 해제해야 합니다.
- 추출할 텍스트의 최소 높이는 1024 x 768 픽셀 이미지의 경우 12픽셀입니다. 이 차원은 150DPI(인치당 도트 수)에서 약 8포인트 텍스트에 해당합니다.
- 사용자 지정 모델 학습의 경우 학습 데이터의 최대 페이지 수는 사용자 지정 템플릿 모델의 경우 500개, 사용자 지정 인공신경망 모델의 경우 50,000개입니다.
 - 사용자 지정 추출 모델 학습의 경우 학습 데이터의 총 크기는 템플릿 모델의 경우 50MB, 신경망 모델의 경우 1G-MB입니다.
 - 사용자 지정 분류 모델 학습의 경우 학습 데이터의 총 크기는 최대 10,000페이지인 1GB입니다.

① 참고

[샘플 레이블 지정 도구](#)는 BMP 파일 형식을 지원하지 않습니다. 이는 문서 인텔리전스 Service가 아닌 도구의 제한 사항입니다.

버전 마이그레이션

[문서 인텔리전스 v3.1 마이그레이션 가이드](#)를 참조하여 애플리케이션에서 문서 인텔리전스 v3.0을 사용하는 방법을 알아보세요.

다음 단계

- [문서 인텔리전스 스튜디오](#)를 사용하여 사용자 고유의 양식 및 문서를 처리해 보세요.
- [Document Intelligence 빠른 시작](#)을 완료하고 원하는 개발 언어로 문서 처리 앱 만들기를 시작해 보세요.

문서 인텔리전스 스튜디오

아티클 • 2024. 03. 05.

① 중요

- 문서 인텔리전스 공개 미리 보기 릴리스에서는 현재 개발 중인 기능에 대한 조기 액세스를 제공합니다.
- 기능, 방식 및 프로세스는 GA(일반 공급) 전에 사용자 피드백에 따라 변경될 수 있습니다.
- 문서 인텔리전스 클라이언트 라이브러리의 공개 미리 보기 버전은 기본적으로 REST API 버전 [2023-10-31-preview](#)로 설정됩니다.
- 공개 미리 보기 버전 [2023-10-31-preview](#)는 현재 다음 Azure 지역에서만 사용 할 수 있습니다.
 - 미국 동부
 - 미국 서부2
 - 서유럽

이 콘텐츠의 적용 대상은 다음과 같습니다. v4.0(미리 보기) | 이전 버전: v3.1(GA) v3.0(GA)

문서 인텔리전스 스튜디오 [는 문서 인텔리전스 서비스의 기능을 시각적으로 탐색, 이해 및 애플리케이션에 통합하기 위한 온라인 도구입니다. 문서 인텔리전스 스튜디오를 사용하여 다음을 수행합니다.](#)

- 문서 인텔리전스의 다양한 기능에 대해 자세히 알아봅니다.
- 문서 인텔리전스 리소스를 사용하여 샘플 문서에서 모델을 테스트하거나 자체 문서를 업로드합니다.
- 다양한 추가 기능 및 미리 보기 기능을 실험하여 필요에 맞게 출력을 조정합니다.
- 사용자 지정 분류 모델을 학습시켜 문서를 분류합니다.
- 사용자 지정 추출 모델을 학습시켜 문서에서 필드를 추출합니다.
- 애플리케이션에 통합할 언어별 SDK의 샘플 코드를 가져옵니다.

문서 분석 또는 미리 빌드된 모델을 사용하여 문서 분석을 시작하려면 [문서 인텔리전스 스튜디오 빠른 시작](#)을 사용하세요. 언어별 SDK 중 하나와 기타 빠른 시작을 사용하여 사용자 지정 모델을 빌드하고 애플리케이션에서 모델을 참조합니다.

다음 이미지는 문서 인텔리전스 스튜디오의 랜딩 페이지를 보여줍니다.

Extract text, tables, structure, key-value pairs, and named entities from documents.

**Read**

Extract printed and handwritten text from images and documents.

[Try it out](#)
**Layout**

Extract tables, check boxes, and text from forms and documents.

[Try it out](#)
**General documents**

Extract labels, values and entities such as names, dates, and amounts from documents.

[Try it out](#)
**General documents with query fields**

Extract labels, values and entities such as names, dates, and amounts from documents.

[Private preview – sign up.](#)
**Prebuilt models**

Extract data from unique document types using the following prebuilt models.

**Invoices**

Extract invoice ID, customer details, vendor details, ship to, bill to, total tax, subtotal, line items and more.

[Try it out](#)
**Receipts**

Extract time and date of the transaction, merchant information, amounts of taxes, totals and more.

[Try it out](#)
**Business cards**

Extract person name, job title, address, email, company, and phone numbers from business cards.

[Try it out](#)
**Identity documents**

Extract name, expiration date, machine readable zone, and more from passports and ID cards.

[Try it out](#)
**Health insurance cards**

Extract insurer, member, prescription, group number and more information from US health insurance cards.

[Try it out](#)
**W-2**

Extract employee, employer, wage information, etc. from US W-2 Tax Form.

[Try it out](#)
**1098**

Extract mortgage interest information from US 1098 Tax Form.

[Private preview – sign up.](#)
**1098-E**

Extract student loan information from US 1098-E Tax Form.

[Private preview – sign up.](#)
**1098-T**

Extract tuition information from US 1098-T Tax Form.

[Private preview – sign up.](#)
**Vaccination cards**

Extract card holder, health agency and vaccination records from US COVID-19 vaccination cards.

[Private preview – sign up.](#)
**Contracts**

Extract the title and signatory parties' information (including names, reference names, and addresses) from contracts.

[Private preview – sign up.](#)
**Custom models**

Train custom models to classify documents and extract text, structure and fields from your forms or documents.

**Custom extraction model**

Label and build a custom model to extract a specific schema from your forms and documents.

[Create new](#)
**Custom classification model**

Build a custom classification model to split and classify documents.

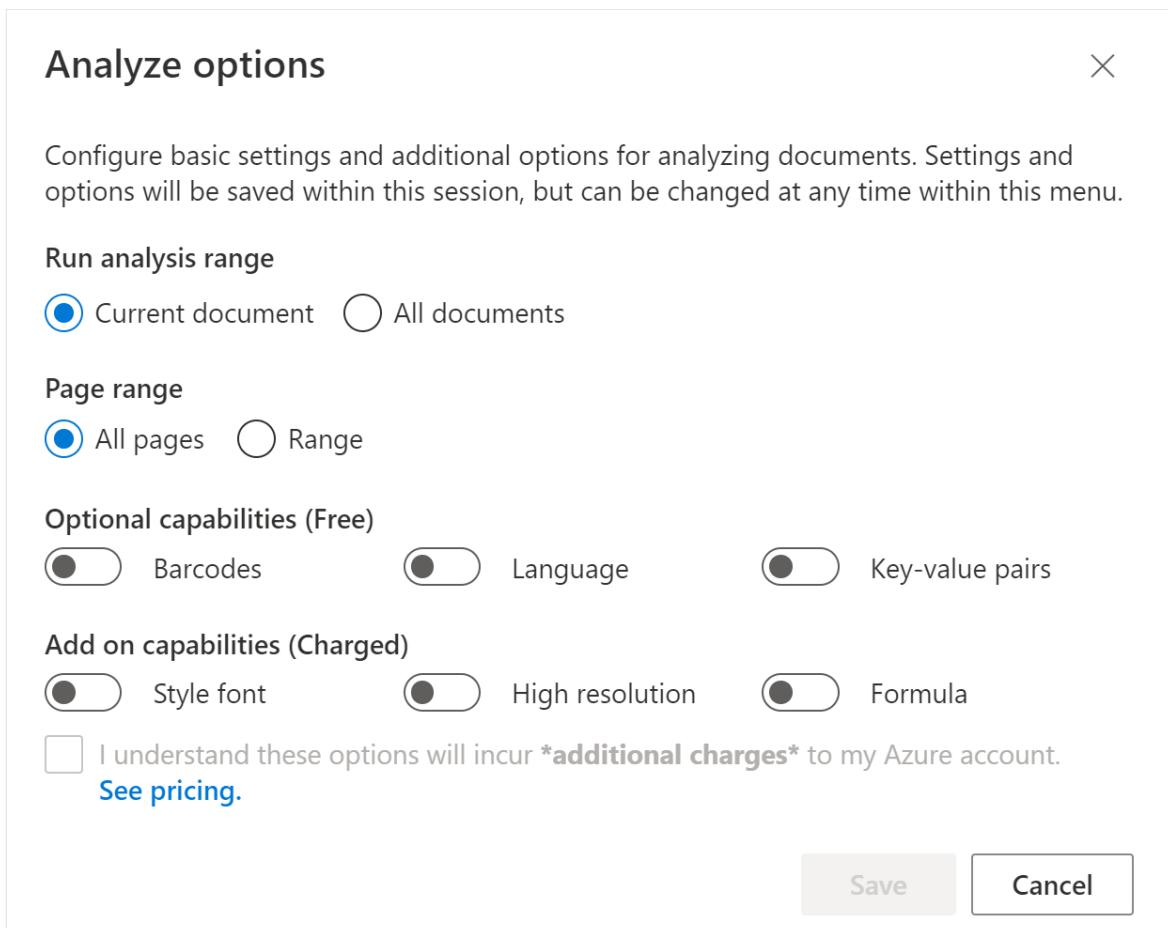
[Create new](#)

시작하기

스튜디오를 처음 방문하는 경우 [시작 가이드](#)에 따라 사용할 스튜디오를 설정합니다.

분석 옵션

- 문서 인텔리전스는 정교한 분석 기능을 지원합니다. 스튜디오에서 추가 기능 구성을 위한 하나의 진입점(분석 옵션 단추)을 쉽게 구성할 수 있습니다.
- 문서 추출 시나리오에 따라 분석 범위, 문서 페이지 범위, 선택적 탐지, 프리미엄 탐지 기능을 구성합니다.



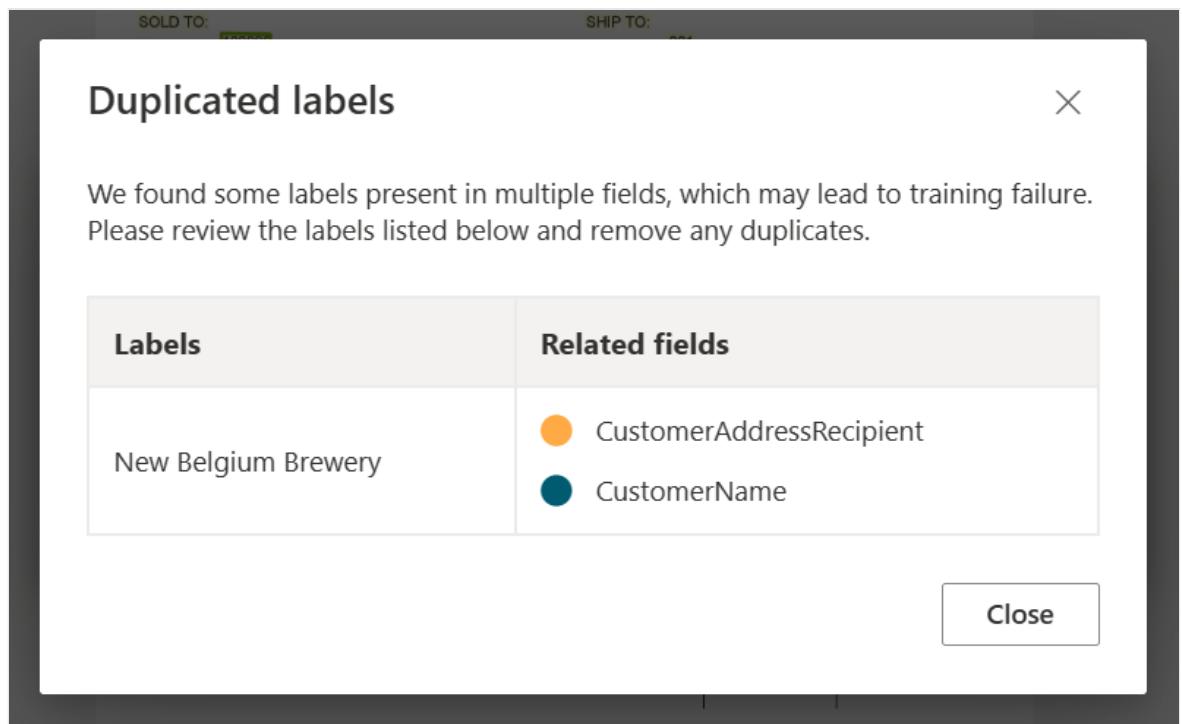
ⓘ 참고

문서 인텔리전스 스튜디오에서는 글꼴 추출이 시각화되지 않습니다. 하지만 JSON 출력의 스타일 섹션에서 글꼴 탐지 결과를 확인할 수 있습니다.

✓ 사전 제작된 모델 또는 자체 모델 중 하나를 사용하여 문서에 자동 레이블 지정

- 이제 사용자 지정 추출 모델 레이블 지정 페이지에서 문서 인텔리전스 서비스 미리 빌드된 모델 또는 학습시킨 모델 중 하나를 사용하여 문서에 자동으로 레이블을 지정할 수 있습니다.

- 일부 문서의 경우 자동 레이블 실행 후 레이블이 중복될 수 있습니다. 이후 레이블 지정 페이지에 중복 레이블이 없도록 레이블을 수정하시기 바랍니다.



✓ 자동 레이블 지정 표

- 사용자 지정 추출 모델 레이블 지정 페이지에서 이제 테이블에 수동으로 레이블을 지정할 필요 없이 문서의 테이블에 자동으로 레이블을 지정할 수 있습니다.

The screenshot shows the 'Label data' page in the Azure AI Document Intelligence Studio. On the right, a booking confirmation form is displayed with several fields highlighted by red boxes. These highlighted fields include:
 - 'Received from': 8556 Indian Summer Ave.
 - 'Name': Summer River
 - 'Phone': Tel: 3446823
 - 'Email': E-mail: summer@libertydelightful.com
 - 'Booking Confirmation – ORIGINAL': Our reference: 3456623
 - 'Summary': 45x72
 - 'Booking date': 05-Dec-2018
 - 'Contract No': 334566
 - 'Opt. A' and 'Opt. B' checkboxes
 - 'Export' and 'Common' buttons
 - A table with rows: Row 1 (18 Queen Street, Hoboken, NJ 07030), Row 2 (Flight, Round, Accent, Monkey, Route), and Row 3 (Ketch Harbour Ave., Peabody, MA 01960).
 - A green bar at the bottom with 'Booking Confirmation – ORIGINAL' and 'Export' buttons.

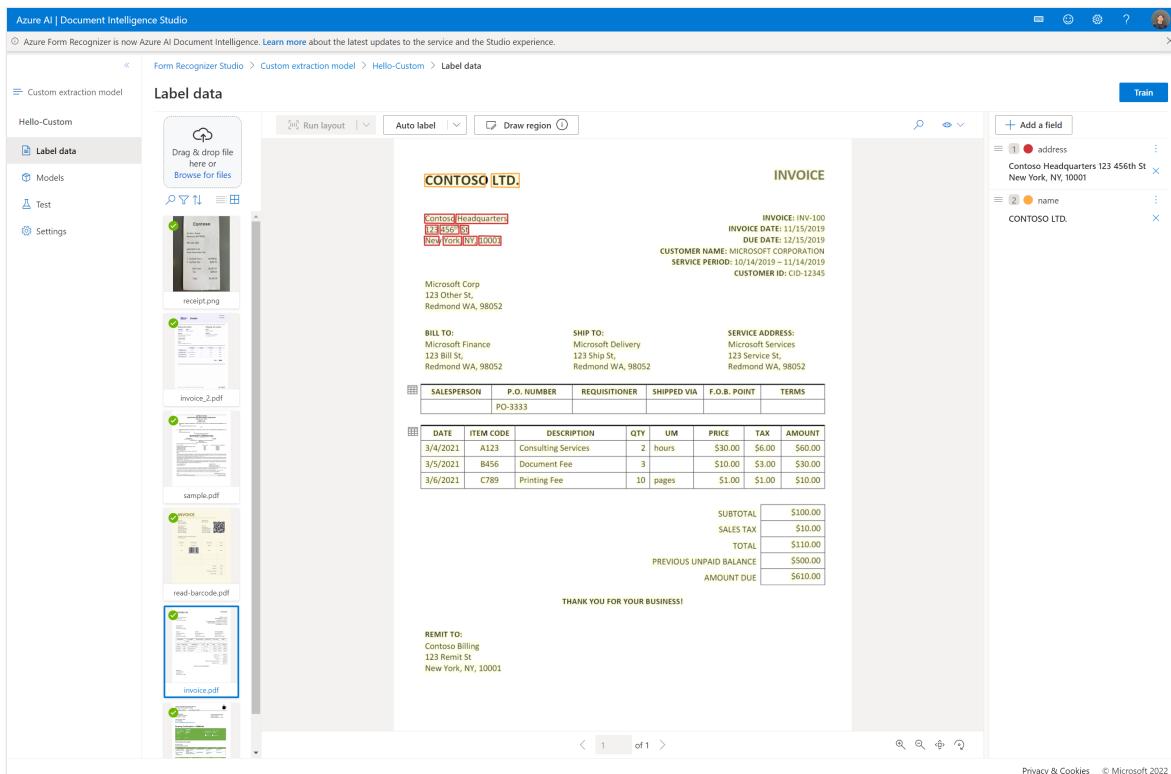
✓ 학습 데이터 세트에 직접 테스트 파일 추가

- 사용자 지정 추출 모델을 학습시킨 후에는 테스트 페이지를 활용하여 필요한 경우 테스트 문서를 학습 데이터 세트에 업로드하여 모델 품질을 향상시키세요.
- 일부 레이블에 대해 낮은 신뢰도 점수가 반환되면 레이블이 올바르게 지정되었는지 확인하세요. 그렇지 않은 경우 학습 데이터 세트에 추가하고 레이블을 다시 지정하여 모델 품질을 향상시키세요.

The screenshot shows the 'Test model' page in the Azure AI Document Intelligence Studio. On the right, a receipt document is shown with extracted text:
 - Address: 123 Main Street Redmond, WA 98052
 - Phone: 987-654-3210
 - Date: 6/10/2019 13:59
 - Sales Associate: Paul
 - Item 1: Surface Pro 6 (\$1,998.00)
 - Item 2: Surface Pen (\$299.97)
 - Sub-Total: \$2,297.97
 - Tax: \$218.31
 - Total: \$2,516.28
 To the right of the receipt, there is a results table:
 - Fields: receipt.png
 - Result: DocType: test-model
 - Code: address #1: 123 Main Street Redmond, WA 98052 (24.80%)
 - Code: name #1: Contoso (58.10%)

✓ 맞춤 프로젝트에서 문서 목록 옵션 및 필터를 활용하세요.

- 사용자 지정 추출 모델 레이블 지정 페이지에서 이제 검색, 필터링 및 정렬 기능을 사용하여 학습 문서를 쉽게 탐색할 수 있습니다.
- 그리드 보기를 활용하여 문서를 미리 보거나 목록 보기 사용하여 문서를 더 쉽게 스크롤할 수 있습니다.



✓ 프로젝트 공유

- 사용자 지정 추출 프로젝트를 쉽게 공유하세요. 자세한 내용은 [맞춤 모델과 프로젝트 공유](#)를 참조하세요.

문서 인텔리전스 모델 지원

- 읽기:** 문서 인텔리전스의 읽기 기능을 사용해 텍스트 줄, 단어, 감지된 언어 및 필기 스타일이 감지되면 추출해 보세요. [Studio 읽기 기능](#)으로 시작합니다. 샘플 문서 및 사용자 문서를 탐색합니다. 대화형 시각화 및 JSON 출력을 사용하여 기능의 작동 방식을 이해합니다. 자세히 알아보려면 [읽기 개요](#)를 참조하고 [레이아웃용 Python SDK 빠른 시작](#)을 시작하세요.
- 레이아웃:** 문서 인텔리전스의 레이아웃 기능을 사용해 텍스트, 표, 선택 표시 및 구조 정보를 추출해 보세요. [Studio 레이아웃 기능](#)으로 시작합니다. 샘플 문서 및 사용자 문서를 탐색합니다. 대화형 시각화 및 JSON 출력을 사용하여 기능의 작동 방식을 이해합니다. 자세히 알아보려면 [레이아웃 개요](#)를 참조하고 [레이아웃용 Python SDK 빠른 시작](#)을 시작하세요.

- **미리 빌드된 모델:** 문서 인텔리전스의 미리 빌드된 모델을 사용하면 자체 모델을 학습하고 빌드할 필요 없이 앱 및 흐름에 지능적인 문서 처리를 추가할 수 있습니다. 예를 들어 [Studio 청구서 기능](#)으로 시작합니다. 샘플 문서 및 사용자 문서를 탐색합니다. 대화형 시각화, 추출된 필드 목록 및 JSON 출력을 사용하여 기능의 작동 방식을 이해합니다. 자세히 알아보려면 [모델 개요](#)를 참조하고 [미리 빌드된 청구서용 Python SDK 빠른 시작](#)을 시작하세요.
- **사용자 지정 추출 모델:** 문서 인텔리전스의 맞춤 모델을 사용하면 데이터로 학습된 모델에서 양식과 문서에 맞게 맞춤화된 필드와 값을 추출할 수 있습니다. 독립 실행형 사용자 지정 모델을 만들거나 두 개 이상의 사용자 지정 모델을 결합하여 여러 양식 형식에서 데이터를 추출하는 구성된 모델을 만듭니다. [Studio 사용자 지정 모델 기능](#)으로 시작합니다. 도움말 마법사, 레이블 지정 인터페이스, 학습 단계 및 시각화를 사용하여 기능 작동 방식을 이해하세요. 샘플 문서를 사용하여 사용자 지정 모델을 테스트하고 반복하여 모델을 개선합니다. 자세한 내용은 [맞춤 모델 개요](#)를 참조하세요.
- **맞춤 분류 모델:** 문서 분류는 문서 인텔리전스에서 지원하는 새로운 시나리오입니다. 문서 분류자 API는 분류 및 분할 시나리오를 지원합니다. 분류 모델을 학습하여 애플리케이션이 지원하는 다양한 유형의 문서를 식별합니다. 분류 모델에 대한 입력 파일은 여러 문서를 포함할 수 있으며 연결된 페이지 범위 내에서 각 문서를 분류합니다. 자세한 내용은 [사용자 지정 분류 모델](#)을 참조하세요.
- **추가 기능:** 문서 인텔리전스는 이제 더욱 정교한 분석 기능을 지원합니다. 이러한 선택적 기능은 각 모델 페이지의 `Analze Options` 버튼을 사용하여 스튜디오에서 활성화 및 비활성화할 수 있습니다. `highResolution`, 수식, 글꼴, 바코드 추출 기능 등 4가지 추가 기능을 사용할 수 있습니다. 자세한 내용은 [추가 기능](#)을 참조하세요.

다음 단계

- 모델과 기능을 사용해 보려면 [문서 인텔리전스 스튜디오](#)를 방문하세요.
- [문서 인텔리전스 스튜디오 빠른 시작](#)으로 시작해 보세요.

사용자 지정 모델: 정확도 및 신뢰도 점수

아티클 • 2024. 03. 12.

이 콘텐츠의 적용 대상은 다음과 같습니다. ✓ v4.0(미리 보기) ✓ v3.1(GA) ✓ v3.0(GA)
✓ v2.1(GA)

① 참고

- 사용자 지정 신경망 모델은 학습 중에 정확도 점수를 제공하지 않습니다.
- 테이블, 표 행 및 표 셀의 신뢰도 점수는 사용자 지정 모델에 대한 2024-02-29-preview API 버전부터 사용할 수 있습니다.

사용자 지정 템플릿 모델은 학습 시 예상 정확도 점수를 생성합니다. 사용자 지정 모델을 사용하여 분석된 문서는 추출된 필드에 대한 신뢰도 점수를 생성합니다. 이 문서에서는 정확도 및 신뢰도 점수를 해석하고 이러한 점수를 사용하여 정확도 및 신뢰도 결과를 개선하는 모범 사례를 알아봅니다.

정확도 점수

`build(v3.0)` 또는 `train(v2.1)` 사용자 지정 모델 작업의 출력에는 예상 정확도 점수가 포함됩니다. 이 점수는 시각적으로 유사한 문서에서 레이블이 지정된 값을 정확하게 예측하는 모델의 기능을 나타냅니다. 정확도 값 범위는 0%(낮음)와 100%(높음) 사이의 백분율입니다. 예상 정확도는 레이블이 지정된 값을 예측하기 위해 학습 데이터의 몇 가지 다른 조합을 실행하여 계산됩니다.

문서 인텔리전스 스튜디오
학습된 사용자 지정 모델(청구서)

form-recognizer-studio

Field Name	Accuracy
Receipt No	95 %
Sold To	95 %
ID #	83.3 %
Live Delivery?	95 %
Online Delivery?	95 %
Video Delivery?	95 %

신뢰도 점수

① 참고

- 이제 테이블, 행 및 셀 신뢰도 점수가 2024-02-29-preview API 버전에 포함됩니다.
- 사용자 지정 모델의 표 셀에 대한 신뢰도 점수는 2024-02-29-preview API부터 API에 추가됩니다.

문서 인텔리전스 분석 결과는 예측된 단어, 키-값 쌍, 선택 표시, 지역 및 서명에 대한 예상 신뢰도를 반환합니다. 현재 모든 문서 필드가 신뢰도 점수를 반환하는 것은 아닙니다.

필드 신뢰도는 예측이 정확할 0에서 1 사이의 예상 확률을 나타냅니다. 예를 들어 신뢰도 값이 0.95(95%)이면 예측이 20번 중 19번 정확할 가능성이 있음을 나타냅니다. 정확도가 중요한 시나리오의 경우 신뢰도를 사용하여 예측을 자동으로 수락할지 아니면 사람이 검토하도록 플래그를 지정할지 결정할 수 있습니다.

문서 인텔리전스 스튜디오

분석된 청구서 미리 빌드된 청구서 모델

----- Analyzing invoice -----

Analyzed document has doc type prebuilt:invoice with confidence : 1.00

Vendor Name: CONTOSO LTD., confidence: 0.96

Vendor address: 123 456th St New York, NY, 10001, confidence: 0.95

Customer Name: MICROSOFT CORPORATION, confidence: 0.95

Customer Address Recipient: Microsoft Corp, confidence: 0.96

Invoice ID: INV-100, confidence: 0.98

Invoice Date: 2019-11-15, confidence: 0.98

Invoice Total: 110.00, confidence: 0.97

Invoice Items:

Unit Price: 1.000000, confidence: 0.68

Description: Test for 23 fields, confidence: 0.90s

Quantity: 1.000000, confidence: 0.88

사용자 지정 모델의 정확도 및 신뢰도 점수 해석

사용자 지정 모델에서 신뢰도 점수를 해석할 때 모델에서 반환된 모든 신뢰도 점수를 고려해야 합니다. 먼저 모든 신뢰도 점수 목록으로 시작해 보겠습니다.

- 문서 형식 신뢰도 점수:** 문서 형식 신뢰도는 분석된 문서가 학습 데이터 세트의 문서와 비슷하게 표시된 지표입니다. 문서 형식 신뢰도가 낮으면 분석된 문서의 서식 파일 또는 구조적 변형을 나타냅니다. 문서 형식 신뢰도를 높이려면 특정 변형으로 문서에 레이블을 지정하고 학습 데이터 세트에 추가합니다. 모델이 다시 학습되면 해당 변형 클래스를 처리하는 것이 더 잘 준비되어야 합니다.
- 필드 수준 신뢰도:** 추출된 각 래블 필드에는 연결된 신뢰도 점수가 있습니다. 이 점수는 추출된 값의 위치에 대한 모델의 신뢰도를 반영합니다. 신뢰도를 평가하는 동안 추출된 결과에 대한 포괄적인 신뢰도를 생성하려면 기본 추출 신뢰도도 확인해야 합니다. 필드 형식에 따라 텍스트 추출 또는 선택 표시에 대한 OCR 결과를 평가하여 필드에 대한 복합 신뢰도 점수를 생성합니다.
- Word 신뢰도 점수:** 문서 내에서 추출된 각 단어에는 연결된 신뢰도 점수가 있습니다. 점수는 전사의 신뢰도를 나타냅니다. 페이지 배열에는 단어 배열이 포함되며, 각 단어에는 연결된 범위와 신뢰도가 있습니다. 추출된 사용자 지정 필드의 범위는 추출된 단어의 범위와 일치합니다.
- 선택 표시 신뢰도 점수:** 페이지 배열에는 선택 표시 배열도 포함되며, 각 선택 표시에는 삭제 표시 및 선택 상태 검색의 신뢰도를 나타내는 신뢰도 점수가 있습니다. 레

이블이 지정된 필드가 선택 표시인 경우 선택 표시 신뢰도와 결합된 사용자 지정 필드 선택 신뢰도는 필드가 올바르게 추출된 전체 신뢰도의 정확한 표현입니다.

다음 표에서는 정확도와 신뢰도 점수를 모두 해석하여 사용자 지정 모델의 성능을 측정하는 방법을 보여줍니다.

테이블 확장

정확도	신뢰도	결과
높음	높음	<ul style="list-style-type: none">모델은 레이블이 지정된 키 및 문서 형식으로 잘 작동합니다.균형 잡힌 학습 데이터 세트가 있습니다.
높음	낮음	<ul style="list-style-type: none">분석된 문서가 학습 데이터 세트와 다르게 표시됩니다.이 모델은 레이블이 지정된 문서가 5개 이상 있는 재학습을 통해 이점을 얻을 수 있습니다.이러한 결과는 학습 데이터 세트와 분석된 문서 간의 형식 변형을 나타낼 수도 있습니다. <p>새 모델을 추가하는 것이 좋습니다.</p>
낮음	높음	<ul style="list-style-type: none">이 결과는 거의 없습니다.낮은 정확도 점수를 얻으려면 레이블이 지정된 데이터를 더 추가하거나 시각적으로 고유한 문서를 여러 모델로 분할합니다.
낮음	낮음	<ul style="list-style-type: none">레이블이 지정된 데이터를 더 추가합니다.시각적으로 고유한 문서를 여러 모델로 분할합니다.

표, 행 및 셀 신뢰도

API에 테이블, 행 및 셀 신뢰도 [2024-02-29-preview](#) 를 추가하여 표, 행 및 셀 점수를 해석하는 데 도움이 되는 몇 가지 일반적인 질문은 다음과 같습니다.

Q: 셀에 대해 높은 신뢰도 점수를 볼 수 있지만 행의 신뢰도 점수는 낮습니까?

A: 예. 서로 다른 수준의 테이블 신뢰도(셀, 행 및 테이블)는 해당 특정 수준에서 예측의 정확성을 캡처하기 위한 것입니다. 다른 누락이 있는 행에 속하는 올바르게 예측된 셀은 셀 신뢰도가 높지만 행 신뢰도는 낮아야 합니다. 마찬가지로 다른 행에 문제가 있는 테이블의 올바른 행은 행 신뢰도가 높은 반면 테이블의 전반적인 신뢰도는 낮습니다.

Q: 셀이 병합될 때 예상되는 신뢰도 점수는 무엇인가요? 병합으로 인해 식별된 열 수가 변경되므로 점수는 어떻게 영향을 받나요?

A: 테이블 형식에 관계없이 병합된 셀에 대한 기대는 신뢰도 값이 낮아야 한다는 것입니다. 또한 누락된 셀(인접한 셀과 병합되었기 때문에)에도 `NULL` 신뢰도가 낮은 값이 있어

야 합니다. 이러한 값이 얼마나 낮을지는 학습 데이터 세트에 따라 달라지며, 점수가 낮은 병합된 셀과 누락된 셀의 일반적인 추세는 유지되어야 합니다.

Q: 값이 선택 사항인 경우 신뢰도 점수는 무엇인가요? 값이 누락된 경우 값과 높은 신뢰도 점수를 가진 `NULL` 셀을 예상해야 하나요?

A: 학습 데이터 세트가 셀의 선택성을 나타내는 경우 모델이 학습 집합에 값이 표시되는 빈도를 파악하여 유추 중에 예상되는 사항을 파악하는 데 도움이 됩니다. 이 기능은 예측의 신뢰도를 계산하거나 전혀 예측하지 않을 때 사용됩니다(`NULL`). 학습 집합에서도 대부분 비어 있는 누락 값에 대한 신뢰도가 높은 빈 필드가 필요합니다.

Q: 필드가 선택 사항이며 존재하지 않거나 누락되지 않은 경우 신뢰도 점수는 어떻게 영향을 받나요? 신뢰도 점수가 그 질문에 답할 것이라는 기대는 무엇입니까?

A: 행에서 값이 누락되면 셀 `NULL`에 값과 신뢰도가 할당됩니다. 여기서 높은 신뢰도 점수는 모델 예측(값이 없음)이 정확할 가능성이 더 높다는 것을 의미해야 합니다. 반면, 낮은 점수는 모델에서 더 많은 불확실성을 신호해야 합니다(따라서 누락되는 값과 같은 오류의 가능성).

Q: 행이 여러 페이지로 분할된 다중 페이지 테이블을 추출할 때 셀 신뢰도 및 행 신뢰도에 대한 기대는 어떻게 되나요?

A: 셀 신뢰도가 높고 행 신뢰도가 분할되지 않은 행보다 낮을 수 있습니다. 학습 데이터 집합의 분할 행 비율은 신뢰도 점수에 영향을 줄 수 있습니다. 일반적으로 분할 행은 테이블의 다른 행과 다르게 보입니다(따라서 모델은 정확하지 않습니다).

Q: 페이지 경계에서 클린 끝과 시작 행이 있는 페이지 간 테이블의 경우 신뢰도 점수가 페이지에서 일관된다고 가정하는 것이 옳습니까?

A: 예. 행은 문서(또는 페이지)에 있는 위치에 관계없이 모양과 내용이 비슷하게 표시되므로 해당 신뢰도 점수는 일치해야 합니다.

Q: 새 신뢰도 점수를 활용하는 가장 좋은 방법은 무엇인가요?

A: 위에서 아래로 시작하는 모든 수준의 테이블 신뢰도를 살펴봅니다. 먼저 테이블의 신뢰도를 전체적으로 검사 다음 행 수준으로 드릴다운하고 개별 행을 살펴보고 마지막으로 셀 수준 신뢰도를 살펴봅니다. 테이블 유형에 따라 다음과 같은 몇 가지 사항을 확인할 수 있습니다.

고정 테이블의 경우 셀 수준 신뢰도는 이미 사물의 정확성에 대한 상당한 정보를 캡처합니다. 즉, 단순히 각 셀을 살펴보고 신뢰도를 확인하면 예측의 품질을 결정하는 데 충분할 수 있습니다. 동적 테이블의 경우 수준은 서로 위에 빌드되므로 위에서 아래로 가는 방법이 더 중요합니다.

높은 모델 정확도 보장

문서의 시각적 구조의 차이는 모델의 정확도에 영향을 미칩니다. 분석된 문서가 학습에 사용된 문서와 다른 경우 보고된 정확도가 일치하지 않을 수 있습니다. 문서 세트는 사람이 볼 때는 비슷하게 보이지만 AI 모델에게는 다르게 보일 수 있다는 점을 기억해야 합니다. 다음은 가장 높은 정확도를 이용한 모델 학습의 모범 사례 목록입니다. 이러한 지침에 따라 분석 중에 정확도 및 신뢰도 점수가 높은 모델을 생성하고 사람이 검토하도록 플래그가 지정된 문서 수를 줄여야 합니다.

- 문서의 모든 변형이 학습 데이터 세트에 포함되도록 해야 합니다. 변형에는 여러 형식이 포함됩니다(예: 디지털 PDF와 스캔된 PDF).
- 모델이 두 유형의 PDF 문서를 모두 분석할 것으로 예상되는 경우 각 형식의 샘플을 학습 데이터 세트에 5개 이상 추가합니다.
- 시각적으로 고유한 문서 형식을 구분하여 다른 모델을 학습시킵니다.
 - 일반적으로 사용자가 입력한 값을 모두 제거하고 문서가 비슷한 경우 기존 모델에 학습 데이터를 더 추가해야 합니다.
 - 문서가 서로 다른 경우 학습 데이터를 다른 폴더로 분할하고 각 변형에 대한 모델을 학습시킵니다. 그런 다음, 다양한 변형을 단일 모델로 작성할 수 있습니다.
- 불필요한 레이블이 없는지 확인합니다.
- 서명 및 지역 레이블 지정에 주변 텍스트가 포함되지 않는지 확인합니다.

다음 단계

[사용자 지정 모델을 만드는 방법 알아보기](#)

문서 API 응답 분석

아티클 • 2024. 03. 08.

이 콘텐츠의 적용 대상은 다음과 같습니다. ✓ v4.0(미리 보기) ✓ v3.1(GA) ✓ v3.0(GA)

이 문서에서는 응답의 `AnalyzeDocument` 일부로 반환된 다양한 개체와 애플리케이션에서 문서 분석 API 응답을 사용하는 방법을 살펴보겠습니다.

문서 분석 요청

문서 인텔리전스 API는 이미지, PDF 및 기타 문서 파일을 분석하여 다양한 콘텐츠, 레이아웃, 스타일 및 의미 요소를 추출하고 검색합니다. 분석 작업은 비동기 API입니다. 문서를 제출하면 완료를 위해 폴링할 URL이 포함된 **Operation-Location** 헤더가 반환됩니다. 분석 요청이 성공적으로 완료되면 응답에는 [모델 데이터 추출](#)에 설명된 요소가 포함됩니다.

응답 요소

- 콘텐츠 요소는 문서에서 추출된 기본 텍스트 요소입니다.
- 레이아웃 요소는 콘텐츠 요소를 구조 단위로 그룹화합니다.
- 스타일 요소는 콘텐츠 요소의 글꼴 및 언어를 설명합니다.
- 의미 체계 요소는 지정된 콘텐츠 요소에 의미를 할당합니다.

모든 콘텐츠 요소는 페이지 번호(`1`-인덱스 지정)로 지정된 페이지에 따라 그룹화됩니다. 또한 의미 체계적으로 연속적인 요소를 함께 배열하는 읽기 순서로 정렬됩니다. 이는 줄 또는 열 경계를 넘더라도 적용됩니다. 단락 및 기타 레이아웃 요소 간의 읽기 순서가 모호한 경우, 서비스는 일반적으로 왼쪽에서 오른쪽, 위에서 아래로 콘텐츠를 반환합니다.

① 참고

현재 문서 인텔리전스는 페이지 경계를 넘는 읽는 순서를 지원하지 않습니다. 선택 표시가 주변 단어 내에 배치되지 않습니다.

최상위 콘텐츠 속성은 모든 콘텐츠 요소의 연결을 읽기 순서로 포함합니다. 모든 요소는 이 콘텐츠 문자열 내의 범위를 통해 읽기 순서로 위치가 지정됩니다. 일부 요소의 콘텐츠는 항상 연속적이지는 않습니다.

분석 응답

각 API에 대한 분석 응답은 서로 다른 개체를 반환합니다. API 응답에는 해당하는 경우 구성 요소 모델의 요소가 포함되어 있습니다.

테이블 확장

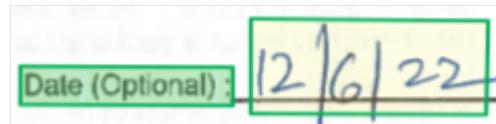
응답 콘텐츠	설명	API
pages	입력 문서의 각 페이지에서 인식된 단어, 줄 및 범위입니다.	읽기 모델, 레이아웃 모델, 일반 문서 모델, 미리 빌드된 모델, 사용자 지정 모델
paragraphs	단락으로 인식된 콘텐츠입니다.	읽기 모델, 레이아웃 모델, 일반 문서 모델, 미리 빌드된 모델, 사용자 지정 모델
styles	식별된 텍스트 요소 속성입니다.	읽기 모델, 레이아웃 모델, 일반 문서 모델, 미리 빌드된 모델, 사용자 지정 모델
languages	추출된 텍스트의 각 범위와 연결된 식별된 언어입니다.	읽음
tables	문서에서 식별되고 추출된 테이블 형식 콘텐츠입니다. 테이블은 미리 학습된 레이아웃 모델로 식별된 테이블과 관련됩니다. 테이블로 테이블이 지정된 콘텐츠는 문서 개체에 구조화된 필드로 추출됩니다.	레이아웃 모델, 일반 문서 모델, 청구서 모델, 사용자 지정 모델
수치	문서에서 식별 및 추출된 그림(차트, 이미지)은 복잡한 정보를 이해하는 데 도움이 되는 시각적 표현을 제공합니다.	레이아웃 모델
섹션	문서에서 식별되고 추출된 계층적 문서 구조입니다. 해당 요소(단락, 표, 그림)가 연결된 섹션 또는 하위 섹션입니다.	레이아웃 모델
keyValuePairs	미리 학습된 모델에서 인식되는 키-값 쌍입니다. 키는 관련 값이 있는 문서의 텍스트 범위입니다.	일반 문서 모델, 청구서 모델
documents	인식된 필드는 문서 목록 내의 <code>fields</code> 사전에 반환됩니다.	미리 빌드된 모델, 사용자 지정 모델.

각 API에서 반환되는 개체에 대한 자세한 내용은 [모델 데이터 추출](#)을 참조하세요.

요소 속성

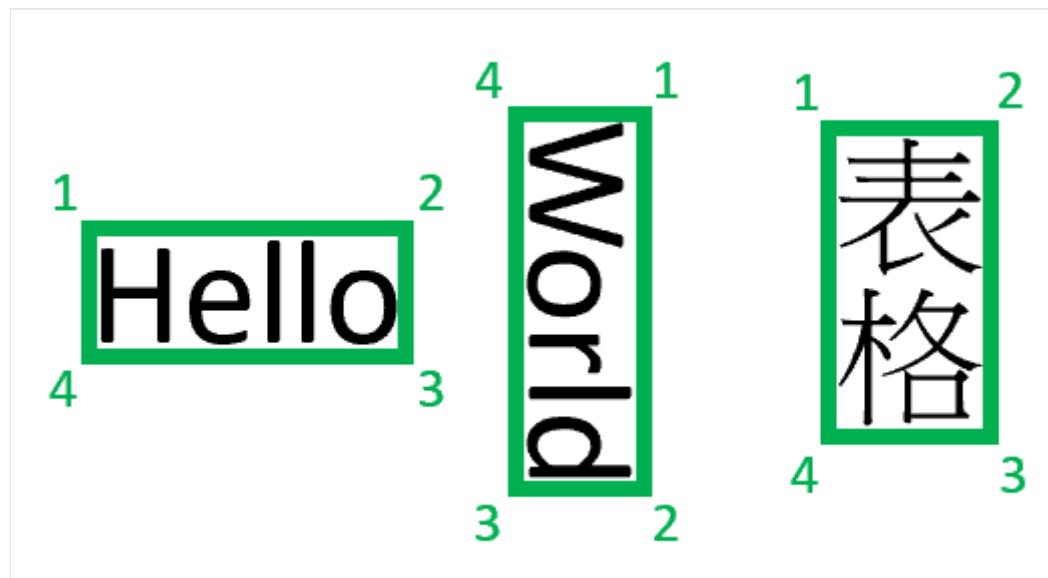
범위

범위는 전체 읽기 순서에서 각 요소의 논리적 위치를 지정하며, 각 범위는 최상위 콘텐츠 문자열 속성에서의 문자 오프셋 및 길이를 지정합니다. 기본적으로 문자 오프셋 및 길이는 사용자 인식 문자 단위(grapheme clusters 또는 텍스트 요소라고도 함)로 반환됩니다. 다른 문자 단위를 사용하는 다양한 개발 환경을 수용하려면 사용자는 유니코드 코드 포인트(Python 3) 또는 UTF16 코드 단위(Java, JavaScript, .NET)에서도 범위 오프셋 및 길이를 반환하는 `stringIndexIndex` 쿼리 매개 변수를 지정할 수 있습니다. 자세한 내용은 [다국어/이모지 지원](#)을 참조하세요.



경계 영역

경계 영역은 파일에 있는 각 요소의 시작적 위치를 설명합니다. 요소가 시각적으로 연속되지 않거나 여러 페이지(테이블)에 걸쳐 있는 경우 대부분의 요소 위치는 경계 영역의 배열을 통해 설명됩니다. 각 영역은 페이지 번호(1 인덱싱됨) 및 경계 다각형을 지정합니다. 경계 다각형은 요소의 자연 방향을 기준으로 왼쪽에서 시계 방향으로 이어지는 포인트 시퀀스로 설명됩니다. 사분면의 경우 플롯 포인트는 왼쪽 위, 오른쪽 위, 오른쪽 아래 및 왼쪽 아래 모서리입니다. 각 점은 단위 속성에 지정된 페이지 단위의 x, y 좌표를 나타냅니다. 일반적으로 이미지의 측정 단위는 픽셀이며 PDF는 인치를 사용합니다.



① 참고

현재 문서 인텔리전스는 4개의 꼭지점 사변형만 경계 다각형으로 반환합니다. 이후 버전에서는 곡선 또는 직사각형이 아닌 이미지와 같이 더 복잡한 셰이프를 설명할 수 있도록 다른 포인트 수를 반환하게 될 수 있습니다. 경계 영역은 렌더링된 파일에

만 적용됩니다. 파일이 렌더링되지 않았다면 경계 영역이 반환되지 않습니다. 현재 docx/xlsx/pptx/html 형식의 파일은 렌더링되지 않습니다.

콘텐츠 요소

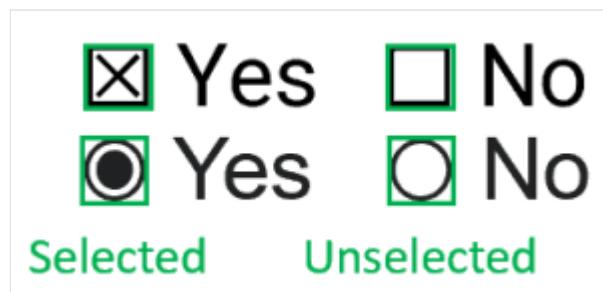
Word

단어는 문자 시퀀스로 구성된 콘텐츠 요소입니다. 문서 인텔리전스를 사용하면 단어는 인접한 문자의 시퀀스로 정의되며, 각 단어는 공백으로 구분됩니다. 단어 사이에 공백 구분 기호를 사용하지 않는 언어의 경우 각 문자는 의미 체계 단어 단위를 나타내지 않더라도 별도의 단어로 반환됩니다.

The Microsoft logo is shown in a large, bold, black font. The word "Microsoft" is followed by its Chinese translation "(微软)" and "Corp.". All text is enclosed within a thick green rectangular border.

선택 표시

선택 영역 표시는 선택 영역의 상태를 나타내는 시각적 문자 모양을 대변하는 콘텐츠 요소입니다. 확인란은 흔히 사용되는 선택 표시의 한 형태입니다. 하지만 라디오 단추 또는 시각적 형식의 박스형 셀로 나타날 수도 있습니다. 선택 표시의 상태를 선택하거나 선택 취소하면 다른 시각적 표현으로 상태를 나타낼 수 있습니다.



레이아웃 요소

줄

줄은 시각적 공간으로 구분된 연속 콘텐츠 요소의 순서가 지정된 시퀀스이거나, 단어 사이에 공백 구분 기호가 없는 언어의 경우 곧바로 인접한 요소입니다. 동일한 가로 평면(행)에 있지만 둘 이상의 시각적 공간으로 구분된 콘텐츠 요소는 대부분 여러 개의 줄로 분할됩니다. 때때로 이 기능은 의미 체계적으로 연속된 콘텐츠를 별도의 줄로 분할하기도

하지만, 텍스트 콘텐츠 표현을 여러 열이나 셀로 분할할 수 있도록 합니다. 세로 쓰기의 줄은 세로 방향으로 검색됩니다.

To make your document look professionally produced, Word provides header, footer, cover page, and text box designs that complement each other. For example, you can add a matching cover page, header, and sidebar.

단락

단락은 논리적 단위를 형성하는 순서가 지정된 줄 시퀀스입니다. 일반적으로 줄 간에는 동일한 공통 맞춤과 간격이 사용됩니다. 단락은 들여쓰기, 추가 간격 또는 글머리 기호/번호 매기기를 통해 구분되는 경우가 많습니다. 콘텐츠는 단일 단락에만 할당할 수 있습니다. 선택 단락은 문서의 기능 역할과도 연결될 수 있습니다. 현재 지원되는 역할에는 페이지 머리글, 페이지 바닥글, 페이지 번호, 제목, 섹션 제목 및 각주 등이 있습니다.

Video provides a powerful way to help you prove your point. When you click Online Video, you can paste in the embed code for the video you want to add. You can also type a keyword to search online for the video that best fits your document.

Themes and styles also help keep your document coordinated. When you click Design and choose a new Theme, the pictures, charts, and SmartArt graphics change to match your new theme. When you apply styles, your headings change to match the new theme.

To make your document look professionally produced, Word provides header, footer, cover page, and text box designs that

Save time in Word with new buttons that show up where you need them. To change

페이지

페이지는 콘텐츠를 그룹화한 것으로 일반적으로 용지 한쪽에 해당합니다. 렌더링된 페이지는 지정된 단위의 너비와 높이를 특징으로 합니다. 일반적으로 이미지는 픽셀을 사용하고 PDF는 인치를 사용합니다. 각도 속성은 회전할 수 있는 페이지의 전체 텍스트 각도를 단위로 설명합니다.

① 참고

Excel과 같은 스프레드시트의 경우 각 시트가 페이지에 매핑됩니다. PowerPoint와 같은 프레젠테이션의 경우 각 슬라이드가 페이지에 매핑됩니다. HTML 또는 Word 문서와 같이 렌더링되지 않고 페이지의 네이티브 개념이 없는 파일 형식의 경우 파일의 기본 콘텐츠는 단일 페이지로 간주됩니다.

테이블

테이블은 콘텐츠를 그리드 레이아웃의 셀 그룹으로 구성합니다. 행과 열은 눈금선, 색상 밴딩 또는 더 큰 간격으로 시각적으로 구분될 수 있습니다. 테이블 셀의 위치는 행 및 열

인덱스를 통해 지정됩니다. 셀은 여러 행과 열에 걸쳐 있을 수 있습니다.

위치 및 스타일에 따라 셀은 일반 콘텐츠, 행 머리글, 열 머리글, 스텁 헤드 또는 설명으로 분류될 수 있습니다.

- 행 머리글 셀은 일반적으로 행의 다른 셀을 설명하는 행의 첫 번째 셀입니다.
- 열 머리글 셀은 일반적으로 열의 다른 셀을 설명하는 열의 첫 번째 셀입니다.
- 행 또는 열에는 계층적 콘텐츠를 설명하는 여러 머리글 셀이 포함될 수 있습니다.
- 스텁 헤드 셀은 일반적으로 첫 번째 행과 첫 번째 열 위치에 있는 셀입니다. 비어 있을 수도 있고, 동일한 행/열의 머리글 셀에 있는 값을 설명할 수도 있습니다.
- 설명 셀은 일반적으로 테이블의 맨 위쪽 또는 아래쪽 영역에 표시되며 전체 테이블 내용을 설명합니다. 그러나 테이블을 섹션으로 나누기 위해 테이블 중간에 표시되는 경우도 있습니다. 일반적으로 설명 셀은 단일 행의 여러 셀에 걸쳐 있습니다.
- 테이블 캡션은 테이블을 설명하는 콘텐츠를 지정합니다. 테이블에는 연결된 캡션과 일련의 각주가 있을 수 있습니다. 설명 셀과 달리 캡션은 일반적으로 그리드 레이아웃 외부에 있습니다. 표 각주가 표 안의 내용에 주석을 달며 표 표 아래에 각주 기호가 표시되는 경우가 많습니다.

레이아웃 테이블은 테이블 형식 데이터에서 추출된 문서 필드와 다릅니다. 레이아웃 테이블은 콘텐츠의 의미 체계를 고려하지 않고 문서의 테이블 형식 시각적 콘텐츠에서 추출됩니다. 실제로 일부 레이아웃 테이블은 오로지 시각적 레이아웃을 위해 설계되었으며, 구조화된 데이터를 포함하지 않는 경우도 있습니다. 영수증의 항목별 세부 정보와 같이 다양한 시각적 레이아웃이 있는 문서에서 구조화된 데이터를 추출하는 방법의 경우 일반적으로 대규모의 후처리를 진행해야 합니다. 정규화된 필드 이름이 있는 구조화된 필드에 행 또는 열 머리글을 매핑하는 것이 필수입니다. 문서 유형에 따라 미리 빌드된 모델을 사용하거나 사용자 지정 모델을 학습시켜 이러한 구조화된 콘텐츠를 추출합니다. 그 결과로 얻은 정보는 문서 필드로 공개됩니다. 이러한 학습된 모델은 테이블 형식이 아닌 머리글 및 구조화된 데이터가 없는 표 형식 데이터(예: 이력서의 경력 섹션) 또한 처리할 수 있습니다.

(In millions, except earnings per share)	2021	2020	2019
Year Ended June 30,			
Net income available for common shareholders (A)	\$ 61,271	\$ 44,281	\$ 39,240
Weighted average outstanding shares of common stock (B)	7,547	7,610	7,673
Dilutive effect of stock-based awards	61	73	80
Common stock and common stock equivalents (C)	7,608	7,683	7,753
Earnings Per Share			
Basic (A/B)	\$ 8.12	\$ 5.82	\$ 5.11
Diluted (A/C)	\$ 8.05	\$ 5.76	\$ 5.06

수치

문서의 그림(차트, 이미지)은 텍스트 콘텐츠를 보완하고 향상시키는 데 중요한 역할을 하며 복잡한 정보를 이해하는 데 도움이 되는 시각적 표현을 제공합니다. 레이아웃 모델에서 검색된 그림 개체에는 그림의 경계를 윤곽을 그리는 페이지 번호 및 다각형 좌표를 포함하여 문서 페이지에 있는 그림의 공간 위치) `spans` 와 같은 `boundingRegions` 주요 속성이 있습니다(그림과 관련된 텍스트 범위를 자세히 설명하고 문서 텍스트 내에서 오프셋과 길이를 지정합니다. 이 연결은 그림을 관련 텍스트 컨텍스트 `elements` 와 연결하고,(그림과 관련되거나 설명되는 문서 내의 텍스트 요소 또는 단락에 대한 식별자) 및 `caption` 있는 경우 도움이 됩니다.

```
JSON

{
  "figures": [
    {
      "boundingRegions": [],
      "spans": [],
      "elements": [
        "/paragraphs/15",
        ...
      ],
      "caption": {
        "content": "Here is a figure with some text",
        "boundingRegions": [],
        "spans": [],
        "elements": [
          "/paragraphs/15"
        ]
      }
    }
  ]
}
```

섹션

계층적 문서 구조는 광범위한 문서를 구성, 이해 및 처리하는 데 중요한 역할을 합니다. 이 접근 방식은 이해력을 높이고 탐색을 용이하게 하며 정보 검색을 개선하기 위해 긴 문서를 의미상 분할하는 데 매우 중요합니다. 문서 생성 AI에서 [RAG\(검색 보강 세대\)](#) 의 출현은 계층적 문서 구조 분석의 중요성을 강조합니다. 레이아웃 모델은 각 섹션 내의 섹션과 개체의 관계를 식별하는 출력의 섹션 및 하위 섹션을 지원합니다. 계층 구조는 각 섹션에 `elements` 기본.

```
JSON

{
  "sections": [
    {
      "spans": []
    }
  ]
}
```

```
    "elements": [
        "/paragraphs/0",
        "/sections/1",
        "/sections/2",
        "/sections/5"
    ],
},
...
}
```

양식 필드(키 값 쌍)

양식 필드는 필드 레이블(키) 및 값으로 구성됩니다. 필드 레이블은 일반적으로 필드의 의미를 설명하는 설명 텍스트 문자열입니다. 값의 왼쪽에 표시되는 경우가 많지만 값 위쪽이나 아래쪽에 표시될 수도 있습니다. 필드 값에는 특정 필드 인스턴스의 콘텐츠 값이 포함됩니다. 같은 단어, 선택 표시 및 기타 콘텐츠 요소로 구성될 수 있습니다. 채워지지 않은 양식 필드의 경우 비어 있을 수도 있습니다. 특수 형식의 양식 필드에는 필드 레이블이 오른쪽에 있는 선택 표시 값이 있습니다. 문서 필드는 일반 양식 필드와 비슷하면서도 다른 개념입니다. 일반 양식 필드의 필드 레이블(키)은 문서에 표시되어야 합니다. 따라서 일반적으로 영수증의 가맹점 이름과 같은 정보는 캡처할 수 없습니다. 문서 필드에는 레이블이 지정되고 키를 추출하지 않습니다. 문서 필드는 추출된 값만 레이블이 지정된 키에 매핑합니다. 자세한 내용은 [문서 필드](#)를 참조하세요.

A screenshot of a digital form field. The label above the field is '7. PROPOSED START DATE'. The input field contains the date '2/1/2022' in a large, bold, black font. The entire field is highlighted with a green border.

스타일 요소

스타일

스타일 요소는 텍스트 콘텐츠에 적용할 글꼴 스타일을 설명합니다. 콘텐츠는 전역 콘텐츠 속성에 대한 범위를 통해 지정됩니다. 현재 감지되는 유일한 글꼴 스타일은 텍스트가 필기로 작성된 것인지 여부뿐입니다. 앞으로 다른 스타일이 추가되면 충돌하지 않는 여러 스타일 개체를 통해 텍스트를 설명할 수 있습니다. 압축성을 높이기 위해, 특정 글꼴 스타일을 (동일한 신뢰도로) 공유하는 모든 텍스트는 단일 스타일 개체를 통해 설명됩니다.

A screenshot of a digital form field. The label to the left of the field is 'Date (Optional)'. The input field contains the date '12/16/22' in a large, bold, black font. The entire field is highlighted with a green border.

JSON

```
{  
    "confidence": 1,  
    "spans": [  
        {  
            "offset": 2402,  
            "length": 7  
        }  
    ],  
    "isHandwritten": true  
}
```

언어

언어 요소는 전역 콘텐츠 속성에 대한 범위를 통해 지정된 콘텐츠로 감지된 언어를 설명합니다. 감지된 언어는 [BCP-47 언어 태그](#)를 통해 지정되어 기본 언어와 선택적 스크립트 및 지역 정보를 나타냅니다. 예를 들어 영어와 중국어 번체는 각각 "en" 및 zh-Hant로 인식됩니다. 영국 영어의 지역별 맞춤법 차이로 인해 텍스트가 en-GB로 감지될 수도 있습니다. 언어 요소는 주요 언어(예: 숫자)가 없는 텍스트를 다루지 않습니다.

의미 체계 요소

① 참고

여기서 설명하는 의미 체계 요소는 문서 인텔리전스 미리 빌드된 모델에 적용됩니다. 사용자 지정 모델은 다른 데이터 표현을 반환할 수 있습니다. 예를 들어 사용자 지정 모델에서 반환되는 날짜 및 시간은 표준 ISO 8601 서식과 다른 패턴으로 표시될 수 있습니다.

문서

문서는 의미 체계상 완전한 하나의 단위입니다. PDF 파일 안에 세금 양식이 여러 개 있거나, 단일 페이지 안에 영수증이 여러 개 있는 경우처럼 파일에 여러 문서가 포함될 수 있습니다. 그러나 파일 내의 문서 순서는 전달되는 정보에 근본적인 영향을 미치지 않습니다.

① 참고

현재 문서 인텔리전스는 단일 페이지에 여러 문서를 지원하지 않습니다.

문서 형식은 시각적 개체 템플릿이나 레이아웃과 관계없이 구조화된 스키마로 표현되는 공통된 의미 체계 필드 집합을 공유하는 문서를 설명합니다. 예를 들어 "영수증" 유형의 모든 문서에는 가맹점 이름, 거래 날짜 및 거래 합계가 포함될 수 있지만, 레스토랑 영수증과 호텔 영수증은 모양이 다른 경우가 많습니다.

문서 요소에는 감지된 문서 형식의 의미 체계 스키마에 의해 지정된 필드 중에서 인식된 필드의 목록이 포함됩니다.

- 문서 필드는 추출하거나 유추할 수 있습니다. 추출된 필드는 추출된 콘텐츠로 표시되며, 해석 가능한 경우 필요에 따라 정규화된 값으로 표시됩니다.
- 유추된 필드에는 콘텐츠 속성이 없으며 해당 필드의 값으로만 표시됩니다.
- 배열 필드에는 콘텐츠 속성이 포함되지 않습니다. 이 콘텐츠는 배열 요소의 콘텐츠와 연결할 수 있습니다.
- 개체 필드에는 추출된 하위 필드의 상위 집합일 수 있는 개체를 나타내는 전체 콘텐츠를 지정하는 콘텐츠 속성이 포함되어 있습니다.

문서 형식의 의미 체계 스키마는 문서 형식에 포함될 수 있는 필드를 통해 설명되어 있습니다. 각 필드 스키마는 정식 이름과 값 형식을 통해 지정됩니다. 필드 값 형식에는 기본(예: 문자열), 복합(예: 주소) 및 구조적(예: 배열, 개체) 형식이 있습니다. 필드 값 형식은 감지된 콘텐츠를 정규화 표현으로 변환하기 위해 수행되는 의미 체계 정규화 또한 지정합니다. 정규화는 로캘에 따라 달라질 수 있습니다.

기본 형식

[+] 테이블 확장

필드 값 형식	설명	정규화된 표현	예제(필드 콘텐츠 -> 값)
string	일반 텍스트	콘텐츠와 동일	MerchantName: "Contoso" → "Contoso"
date	날짜	ISO 8601 - YYYY-MM-DD	InvoiceDate: "2022년 5월 7일" → "2022-05-07"
time	Time	ISO 8601 - hh:mm:ss	TransactionTime: "오후 9시 45분" → "21:45:00"
phoneNumber	전화번호	E.164 - +{CountryCode}{SubscriberNumber}	WorkPhone: "(800) 555-7676" → "+18005557676"
countryRegion	국가/지역	ISO 3166-1 alpha-3	CountryRegion: "미국" → "USA"

필드 값 형식	설명	정규화된 표현	예제(필드 콘텐츠 -> 값)
selectionMark	선택됨	"signed" 또는 "unsigned"	AcceptEula: <input checked="" type="checkbox"/> → "selected"
signature	서명됨	콘텐츠와 동일	LendeeSignature: {signature} → "signed"
번호	부동 소수점 수	부동 소수점 수	수량: "1.20" → 1.2
정수	정수	64비트 부호 있는 숫자	개수: "123" → 123
부울 값	부울 값	true/false	IsStatutoryEmployee: <input checked="" type="checkbox"/> → true

복합 형식

- 통화: 통화 단위(선택 사항)가 있는 통화 금액입니다. 값이며, 예를 들면 `InvoiceTotal: $123.45` 입니다.

JSON

```
{
  "amount": 123.45,
  "currencySymbol": "$"
}
```

- 주소: 구문 분석된 주소입니다. 예: `ShipToAddress: 123 Main St., Redmond, WA 98052`

JSON

```
{
  "poBox": "PO Box 12",
  "houseNumber": "123",
  "streetName": "Main St.",
  "city": "Redmond",
  "state": "WA",
  "postalCode": "98052",
  "countryRegion": "USA",
  "streetAddress": "123 Main St."
}
```

구조적 형식

- 배열: 형식이 동일한 필드의 목록

JSON

```
"Items": {  
    "type": "array",  
    "valueArray": [  
  
    ]  
}
```

- **개체:** 형식이 다를 수 있는 하위 필드의 명명된 목록

JSON

```
"InvoiceTotal": {  
    "type": "currency",  
    "valueCurrency": {  
        "currencySymbol": "$",  
        "amount": 110  
    },  
    "content": "$110.00",  
    "boundingRegions": [  
        {  
            "pageNumber": 1,  
            "polygon": [  
                7.3842,  
                7.465,  
                7.9181,  
                7.465,  
                7.9181,  
                7.6089,  
                7.3842,  
                7.6089  
            ]  
        }  
    ],  
    "confidence": 0.945,  
    "spans": [  
        {  
            "offset": 806,  
            "length": 7  
        }  
    ]  
}
```

다음 단계

- Document Intelligence Studio를 사용하여 사용자 고유의 양식 및 문서를 처리해 보세요.

- Document Intelligence [빠른 시작](#)을 완료하고 원하는 개발 언어로 문서 처리 앱 만들기를 시작해 보세요.

문서 인텔리전스 읽기 모델

아티클 • 2024. 04. 14.

① 중요

- 문서 인텔리전스 공개 미리 보기 릴리스에서는 현재 개발 중인 기능에 대한 조기 액세스를 제공합니다.
- 기능, 방식 및 프로세스는 GA(일반 공급) 전에 사용자 피드백에 따라 변경될 수 있습니다.
- 문서 인텔리전스 클라이언트 라이브러리의 공개 미리 보기 버전은 기본적으로 REST API 버전 [2024-02-29-preview](#)로 설정됩니다.
- 공개 미리 보기 버전 [2024-02-29-preview](#)는 현재 다음 Azure 지역에서만 사용할 수 있습니다.
 - 미국 동부
 - 미국 서부2
 - 서유럽

이 콘텐츠의 적용 대상은 다음과 같습니다. v4.0(미리 보기) | 이전 버전: v3.1(GA) v3.0(GA)

① 참고

레이블, 도로 표지판 및 포스터와 같은 외부 이미지에서 텍스트를 추출하려면 사용자 환경 시나리오에 OCR을 쉽게 포함할 수 있도록 성능이 향상된 동기 API를 사용하여 일반 문서가 아닌 이미지에 최적화된 [Azure AI 이미지 분석 v4.0 읽기](#) 기능을 사용합니다.

문서 인텔리전스의 OCR(광학 인식) 읽기 모델은 PDF 문서 및 스캔한 이미지에서 인쇄 및 필기 텍스트를 읽고 추출하는 Azure AI Vision보다 높은 해상도로 실행됩니다. Microsoft Word, Excel, PowerPoint 및 HTML 문서에서 텍스트를 추출하기 위한 지원도 포함되어 있습니다. 단락, 텍스트 줄, 단어, 위치 및 언어를 검색합니다. 읽기 모델은 사용자 지정 모델 외에도 레이아웃, 일반 문서, 청구서, 영수증, ID 문서, 건강보험증, W2와 같은 다른 문서 인텔리전스 미리 빌드된 모델에 대한 기본 OCR 엔진입니다.

문서용 OCR이란?

문서용 OCR(광학 인식)은 여러 파일 형식 및 전역 언어로 된 텍스트가 많은 대형 문서에 최적화되어 있습니다. 더 작고 조밀한 텍스트를 더 잘 처리하기 위해 문서 이미지의 고해상도 검사, 단락 검색 및 채울 수 있는 양식 관리와 같은 기능이 포함되어 있습니다. OCR 기능에는 단일 문자 상자와 같은 고급 시나리오와 청구서, 영수증 및 기타 미리 빌드된 시나리오에서 일반적으로 발견되는 주요 필드의 정확한 추출도 포함됩니다.

개발 옵션

문서 인텔리전스 v4.0(2024-02-29-미리 보기 2023-10-31-preview)은 다음 도구, 애플리케이션 및 라이브러리를 지원합니다.

[+] 테이블 확장

기능	리소스	Model ID
OCR 모델 읽기	<ul style="list-style-type: none">문서 인텔리전스 스튜디오REST APIC# SDKPython SDKJava SDKJavaScript SDK	prebuilt-read

입력 요구 사항

- 최상의 결과를 위해 문서당 하나의 명확한 사진 또는 고품질 스캔을 제공합니다.
- 지원 파일 형식:

[+] 테이블 확장

모델	PDF	이미지: JPEG/JPG, PNG, BMP, TIFF, HEIF	Microsoft Office: Word(DOCX), Excel(XLSX), PowerPoint(PPTX) 및 HTML
읽음	✓	✓	✓
레이아웃	✓	✓	✓ (2024-02-29-preview, 2023-10-31-preview)
일반 문서	✓	✓	
사전 제작	✓	✓	
사용자 지정 추출	✓	✓	

모델	PDF	이미지: JPEG/JPG, PNG, BMP, TIFF, HEIF	Microsoft Office: Word(DOCX), Excel(XLSX), PowerPoint(PPTX) 및 HTML
사용자 지정 분류	✓	✓	✓ (2024-02-29-preview)

- PDF 및 TIFF의 경우 최대 2000페이지를 처리할 수 있습니다(무료 계층 구독의 경우 처음 2페이지만 처리됨).
- 문서 분석을 위한 파일 크기는 유료(S0) 계층의 경우 500MB, 무료(F0) 계층의 경우 4MB입니다.
- 이미지 크기는 50 x 50픽셀에서 10,000 x 10,000픽셀 사이여야 합니다.
- PDF가 암호로 잠긴 경우에는 제출하기 전에 잠금을 해제해야 합니다.
- 추출할 텍스트의 최소 높이는 1024 x 768 픽셀 이미지의 경우 12픽셀입니다. 이 차원은 150DPI(인치당 도트 수)에서 약 8포인트 텍스트에 해당합니다.
- 사용자 지정 모델 학습의 경우 학습 데이터의 최대 페이지 수는 사용자 지정 템플릿 모델의 경우 500개, 사용자 지정 인공신경망 모델의 경우 50,000개입니다.
 - 사용자 지정 추출 모델 학습의 경우 학습 데이터의 총 크기는 템플릿 모델의 경우 50MB, 신경망 모델의 경우 1G-MB입니다.
 - 사용자 지정 분류 모델 학습의 경우 학습 데이터의 총 크기는 최대 10,000페이지 인 1GB입니다.

읽기 모델 시작

문서 인텔리전스 스튜디오를 사용하여 양식 및 문서에서 텍스트를 추출해 보세요. 다음 자산이 필요합니다.

- Azure 구독은 [무료로 만들 수](#) 있습니다.
- Azure Portal의 [Document Intelligence 인스턴스](#)입니다. 무료 가격 책정 계층(F0)을 사용하여 서비스를 시도할 수 있습니다. 리소스가 배포된 후 [리소스로 이동](#)을 선택하여 키 및 엔드포인트를 가져옵니다.

A screenshot of the Azure portal interface. On the left, there's a sidebar with various service links like Overview, Activity log, Access control (IAM), Tags, Diagnose and solve problems, Resource Management, Keys and Endpoint (which is selected and highlighted with a red box), Encryption, Pricing tier, Networking, Identity, Cost analysis, Properties, Locks, Monitoring, Automation, and Help. The main content area shows a note about API keys, followed by sections for 'Show Keys' (KEY 1 and KEY 2), 'Location/Region' (set to westus2), and 'Endpoint' (https://contoso-di.cognitiveservices.azure.com/). A blue 'Regenerate Key1' and 'Regenerate Key2' button is located at the top right of the main content area.

① 참고

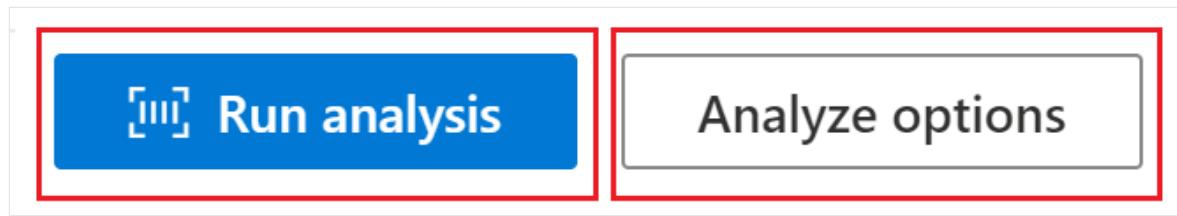
현재 문서 인텔리전스 스튜디오는 Microsoft Word, Excel, PowerPoint 및 HTML 파일 형식을 지원하지 않습니다.

문서 인텔리전스 스튜디오에서 처리된 샘플 문서

A screenshot of the Microsoft Document Intelligence Analyze API interface. At the top, there are buttons for 'Read' (disabled), 'Service resource:' (dropdown), '+ Add' (button), and 'Analyze' (button, highlighted with a red box). Below these are two document thumbnails: 'read-vs...aper.png' and 'read-german.png'. The main content area has a heading 'Introduction' with a detailed description of Visual Studio 2019 licensing. Below it is a section titled 'Visual Studio 2019 Licensing Overview' with information on purchasing licenses for users and the DevOps environment. At the bottom, there's a note about Visual Studio subscription options and a Microsoft logo.

1. 문서 인텔리전스 스튜디오 홈페이지에서 읽기를 선택합니다.
2. 샘플 문서를 분석하거나 자체 파일을 업로드할 수 있습니다.

3. 분석 실행 버튼을 선택하고 필요한 경우 분석 옵션을 구성합니다.



문서 인텔리전스 스튜디오를 사용해 보세요 .

지원되는 언어 및 로캘

지원되는 언어의 전체 목록은 [언어 지원 - 문서 분석 모델](#) 페이지를 참조하세요.

데이터 추출

① 참고

Microsoft Word 및 HTML 파일은 v3.1 이상 버전에서 지원됩니다. PDF 및 이미지와 비교하여 아래 기능은 지원되지 않습니다.

- 각 페이지 개체에는 각도, 너비/높이 및 단위가 없습니다.
- 검색된 각 개체에 대해 경계 다각형 또는 경계 영역이 없습니다.
- 페이지 범위(pages)는 매개 변수로 지원되지 않습니다.
- lines 개체가 없습니다.

페이지

페이지 컬렉션은 문서 내의 페이지 목록입니다. 각 페이지는 문서 내에서 순차적으로 표시되며 페이지가 회전되는지 여부를 나타내는 방향 각도와 너비와 높이(픽셀 단위)를 포함합니다. 모델 출력의 페이지 단위는 다음과 같이 계산됩니다.

테이블 확장

파일 형식	컴퓨팅된 페이지 단위	전체 페이지 수
이미지(JPEG/JPG, PNG, BMP, HEIF)	각 이미지 = 1페이지 단위	총 이미지
PDF	PDF의 각 페이지 = 1페이지 단위	PDF의 총 페이지 수

파일 형식	컴퓨팅된 페이지 단위	전체 페이지 수
TIFF	TIFF의 각 이미지 = 1페이지 단위	TIFF의 총 이미지
Word(DOCX)	최대 3,000자 = 1페이지 단위, 포함되거나 연결된 이미지가 지원되지 않음	각각 최대 3,000자 의 총 페이지
Excel(XLSX)	개별 워크시트 = 1페이지 단위, 포함되거나 연결된 이미지가 지원되지 않음	전체 워크시트
PowerPoint(PPTX)	개별 슬라이드 = 1페이지 단위, 포함되거나 연결된 이미지가 지원되지 않음	전체 슬라이드
HTML	최대 3,000자 = 1페이지 단위, 포함되거나 연결된 이미지가 지원되지 않음	각각 최대 3,000자 의 총 페이지

JSON

```
"pages": [
  {
    "pageNumber": 1,
    "angle": 0,
    "width": 915,
    "height": 1190,
    "unit": "pixel",
    "words": [],
    "lines": [],
    "spans": []
  }
]
```

텍스트 추출을 위한 페이지 선택

대규모의 다중 페이지 PDF 문서의 경우 `pages` 쿼리 매개 변수를 사용하여 텍스트 추출을 위한 특정 페이지 번호 또는 페이지 범위를 지정합니다.

단락

문서 인텔리전스의 OCR 읽기 모델은 `paragraphs` 컬렉션에서 식별된 모든 텍스트 블록을 `analyzeResults` 아래의 최상위 개체로 추출합니다. 이 컬렉션의 각 항목은 텍스트 블록을 나타내며 추출된 텍스트를 `content` 및 경계 `polygon` 좌표로 포함합니다. `span` 정보는 문서의 전체 텍스트가 포함된 최상위 `content` 속성 내의 텍스트 조각을 가리킵니다.

JSON

```
"paragraphs": [
  {
```

```
        "spans": [],
        "boundingRegions": [],
        "content": "While healthcare is still in the early stages of its AI
journey, we are seeing pharmaceutical and other life sciences organizations
making major investments in AI and related technologies.\\" TOM LAWRY |
National Director for AI, Health and Life Sciences | Microsoft"
    }
]
```

텍스트 줄 및 단어

OCR 읽기 모델은 인쇄 및 필기 스타일 텍스트를 `lines` 및 `words`로 추출합니다. 모델은 추출된 단어에 대한 경계 `polygon` 좌표와 `confidence`를 출력합니다. `styles` 컬렉션에는 연결된 텍스트를 가리키는 범위와 함께 선에 대한 필기 스타일이 포함됩니다(탐지된 경우). 이 기능은 [지원되는 필기 언어](#)에 적용됩니다.

Microsoft Word, Excel, PowerPoint 및 HTML의 경우 문서 인텔리전스 읽기 모델 v3.1 이상 버전에서는 포함된 모든 텍스트를 있는 그대로 추출합니다. 텍스트는 단어와 단락으로 추출됩니다. 포함된 이미지는 지원되지 않습니다.

JSON

```
"words": [
  {
    "content": "While",
    "polygon": [],
    "confidence": 0.997,
    "span": {}
  },
],
"lines": [
  {
    "content": "While healthcare is still in the early stages of its AI
journey, we",
    "polygon": [],
    "spans": []
  }
]
```

텍스트 줄에 대한 필기 스타일

응답에는 신뢰도 점수와 함께 각 텍스트 줄이 필기 스타일인지 여부를 분류하는 것이 포함됩니다. 자세한 내용은 [필기 언어 지원](#)을 참조하세요. 다음 예제에서는 JSON 코드 조각 예제를 보여줍니다.

JSON

```
"styles": [  
  {  
    "confidence": 0.95,  
    "spans": [  
      {  
        "offset": 509,  
        "length": 24  
      }  
      "isHandwritten": true  
    ]  
  }  
]
```

글꼴/스타일 추가 기능을 사용하도록 설정하면 글꼴/스타일 결과도 `styles` 개체의 일부로 가져옵니다.

다음 단계

문서 인텔리전스 빠른 시작 완료하기:

- ✓ REST API
- ✓ C# SDK
- ✓ Python SDK
- ✓ Java SDK
- ✓ JavaScript

REST API를 살펴봅니다.

[문서 인텔리전스 API v4.0](#)

문서 인텔리전스 레이아웃 모델

아티클 • 2024. 03. 10.

① 중요

- 문서 인텔리전스 공개 미리 보기 릴리스에서는 현재 개발 중인 기능에 대한 조기 액세스를 제공합니다.
- 기능, 방식 및 프로세스는 GA(일반 공급) 전에 사용자 피드백에 따라 변경될 수 있습니다.
- Document Intelligence 클라이언트 라이브러리의 공개 미리 보기 버전은 기본적으로 REST API 버전 [2024-02-29-preview로 설정됩니다](#).
- 공개 미리 보기 버전 [2024-02-29-preview](#) 는 현재 다음 Azure 지역에서만 사용할 수 있습니다.
 - 미국 동부
 - 미국 서부2
 - 서유럽

이 콘텐츠의 적용 대상은 다음과 같습니다. ✓ v4.0(미리 보기) | 이전 버전: ✓ v3.1(GA) ✓ v3.0(GA) ✓ v2.1(GA)

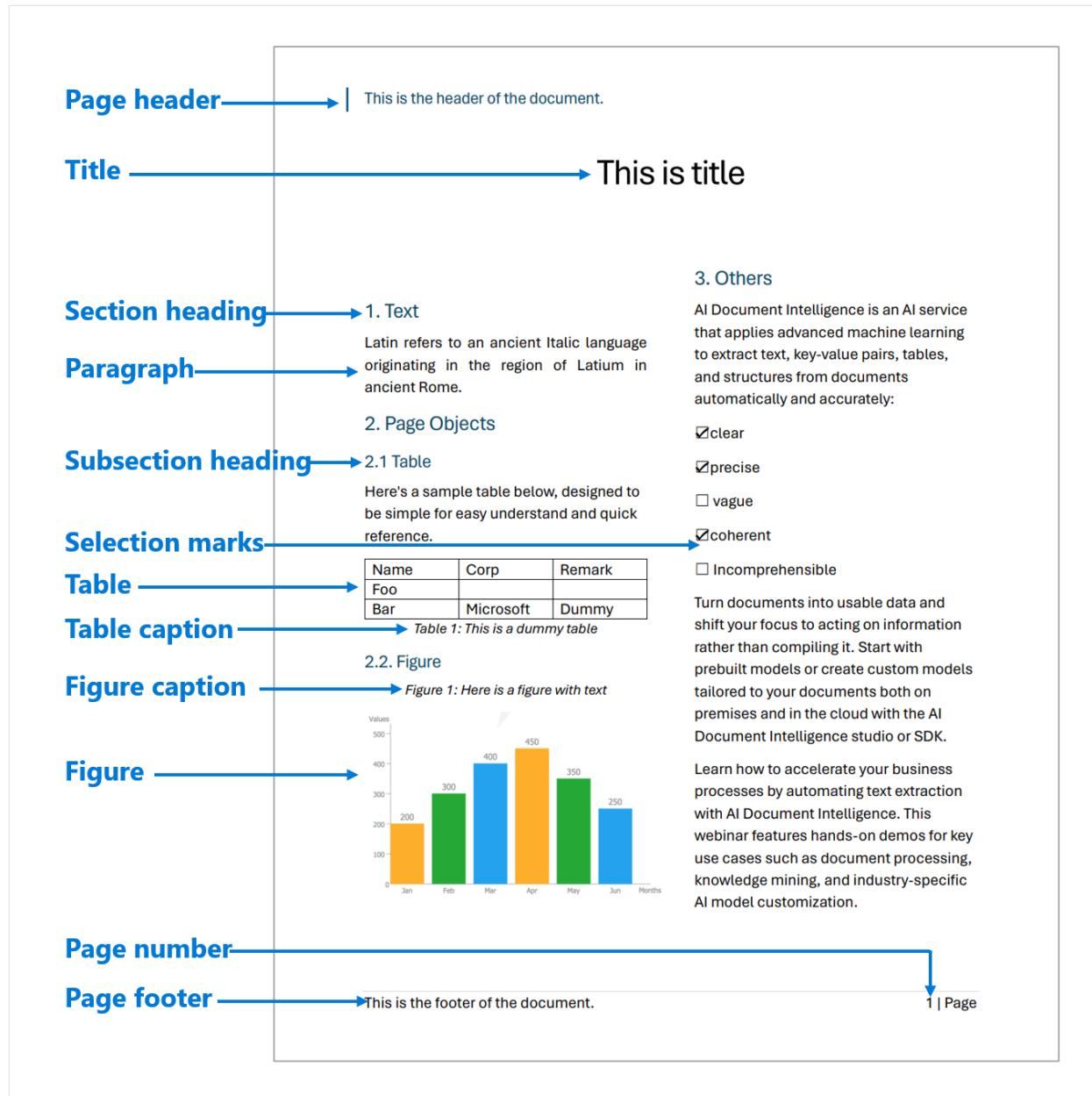
문서 인텔리전스 레이아웃 모델은 문서 인텔리전스 클라우드에서 사용할 수 있는 고급 기계 학습 기반 문서 분석 API입니다. 이를 통해 사용자는 다양한 형식으로 문서를 가져와 문서의 정형 데이터 표현을 반환할 수 있습니다. 이 버전은 강력한 [OCR\(광학 문자 인식\)](#) 기능을 딥 러닝 모델과 결합하여 텍스트, 테이블, 선택 표시 및 문서 구조를 추출합니다.

문서 레이아웃 분석

문서 구조 레이아웃 분석은 문서를 분석하여 관심 영역과 해당 상호 관계를 추출하는 프로세스입니다. 목표는 페이지에서 텍스트 및 구조 요소를 추출하여 더 나은 의미 체계 이해 모델을 빌드하는 것입니다. 문서 레이아웃에는 두 가지 유형의 역할이 있습니다.

- 기하학적 역할:** 텍스트, 테이블, 그림 및 선택 표시는 기하학적 역할의 예입니다.
- 논리적 역할:** 제목, 제목 및 바닥글은 텍스트의 논리적 역할의 예입니다.

다음 그림에서는 샘플 페이지의 이미지에 있는 일반적인 구성 요소를 보여 줍니다.



개발 옵션

Document Intelligence v4.0(2024-02-29-preview, 2023-10-31-preview)은 다음 도구, 애플리케이션 및 라이브러리를 지원합니다.

테이블 확장

기능	리소스	Model ID
레이아웃 모델	<ul style="list-style-type: none"> 문서 인텔리전스 스튜디오 REST API C# SDK Python SDK Java SDK JavaScript SDK 	prebuilt-layout

입력 요구 사항

- 최상의 결과를 위해 문서당 하나의 명확한 사진 또는 고품질 스캔을 제공합니다.
- 지원 파일 형식:

 테이블 확장

모델	PDF	이미지: JPEG/JPG, PNG, BMP, TIFF, HEIF	Microsoft Office: Word(DOCX), Excel(XLSX), PowerPoint(PPTX) 및 HTML
읽음	✓	✓	✓
레이아웃	✓	✓	✓ (2024-02-29-preview, 2023-10-31-preview)
일반 문서	✓	✓	
사전 제작	✓	✓	
사용자 지정	✓	✓	

- PDF 및 TIFF의 경우 최대 2000페이지를 처리할 수 있습니다(무료 계층 구독의 경우 처음 2페이지만 처리됨).
- 문서 분석을 위한 파일 크기는 유료(S0) 계층의 경우 500MB, 무료(F0) 계층의 경우 4MB입니다.
- 이미지 크기는 50 x 50픽셀에서 10,000 x 10,000픽셀 사이여야 합니다.
- PDF가 암호로 잠긴 경우에는 제출하기 전에 잠금을 해제해야 합니다.
- 추출할 텍스트의 최소 높이는 1024 x 768 픽셀 이미지의 경우 12픽셀입니다. 이 차원은 150DPI(인치당 도트 수)에서 약 8포인트 텍스트에 해당합니다.
- 사용자 지정 모델 학습의 경우 학습 데이터의 최대 페이지 수는 사용자 지정 템플릿 모델의 경우 500개, 사용자 지정 인공신경망 모델의 경우 50,000개입니다.
 - 사용자 지정 추출 모델 학습의 경우 학습 데이터의 총 크기는 템플릿 모델의 경우 50MB, 신경망 모델의 경우 1G-MB입니다.
 - 사용자 지정 분류 모델 학습의 경우 학습 데이터의 총 크기는 최대 10,000페이지 인 1GB 입니다.

레이아웃 모델 시작

문서 인텔리전스를 사용하여 텍스트, 테이블, 표 머리글, 선택 표시 및 구조 정보를 포함한 데이터를 문서에서 추출하는 방법을 알아보세요. 다음 리소스가 필요합니다.

- Azure 구독 - 무료로 [구독을 만들 수 있습니다](#).
- Azure Portal의 [Document Intelligence 인스턴스](#)입니다. 무료 가격 책정 계층(F0)을 사용하여 서비스를 시도할 수 있습니다. 리소스가 배포된 후 [리소스로 이동](#)을 선택하여 키 및 엔드포인트를 가져옵니다.

The screenshot shows the Azure portal interface for a 'Contoso-DI' resource. On the left, there's a navigation sidebar with various options like Overview, Activity log, Access control (IAM), Tags, Diagnose and solve problems, Resource Management, Keys and Endpoint (which is selected and highlighted with a red box), Encryption, Pricing tier, Networking, Identity, Cost analysis, Properties, Locks, Monitoring, Automation, and Help. The main content area has a title 'Contoso-DI | Keys and Endpoint'. It includes a note about securely storing keys and two key fields: 'KEY 1' and 'KEY 2', both of which are highlighted with red boxes. Below that is a 'Location/Region' dropdown set to 'westus2' and an 'Endpoint' field containing the URL 'https://contoso-di.cognitiveservices.azure.com/'. There are also 'Regenerate Key1' and 'Regenerate Key2' buttons at the top right of the main content area.

① 참고

문서 인텔리전스 스튜디오는 v3.0 API 이상 버전에서 사용할 수 있습니다.

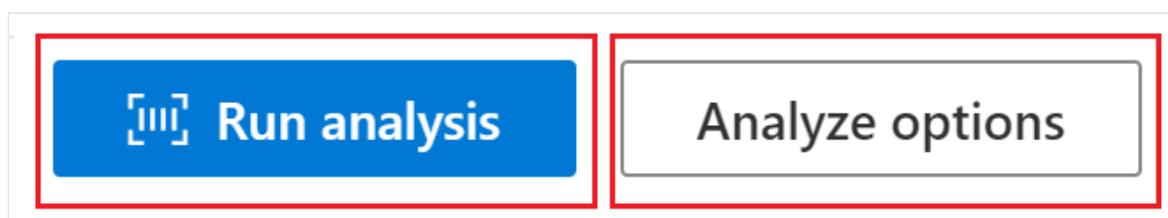
문서 인텔리전스 스튜디오에서 처리된 샘플 문서

The screenshot shows the Microsoft Word ribbon with the 'Text' tab selected. The main document area displays a news article with sections such as 'The scoop of the day' and 'Content'. A sidebar on the right lists analysis results under categories like 'Text', 'Tables', 'Selection marks', and 'Title'.

1. Document Intelligence Studio 홈페이지에서 레이아웃을 선택합니다.

2. 샘플 문서를 분석하거나 자체 파일을 업로드할 수 있습니다.

3. 분석 실행 버튼을 선택하고 필요한 경우 분석 옵션을 구성합니다.



Document Intelligence Studio 를 사용해 보세요.

지원되는 언어 및 로캘

지원되는 언어의 전체 목록은 [언어 지원 - 문서 분석 모델](#) 페이지를 참조하세요.

데이터 추출

레이아웃 모델은 문서에서 텍스트, 선택 표시, 표, 단락 및 단락 형식(`roles`)을 추출합니다.

① 참고

버전 및 2023-10-31-preview 이후 버전 2024-02-29-preview에서는 Microsoft Office(DOCX, XLSX, PPTX) 및 HTML 파일을 지원합니다. 다음 기능은 지원되지 않습니다.

- 각 페이지 개체에는 각도, 너비/높이 및 단위가 없습니다.
- 검색된 각 개체에 대해 경계 다각형 또는 경계 영역이 없습니다.
- 페이지 범위(pages)는 매개 변수로 지원되지 않습니다.
- 개체가 없습니다 lines .

페이지

페이지 컬렉션은 문서 내의 페이지 목록입니다. 각 페이지는 문서 내에서 순차적으로 표시되며 페이지가 회전되는지 여부를 나타내는 방향 각도와 너비와 높이(픽셀 단위)를 포함합니다. 모델 출력의 페이지 단위는 다음과 같이 계산됩니다.

테이블 확장

파일 형식	컴퓨팅된 페이지 단위	전체 페이지 수
이미지(JPEG/JPG, PNG, BMP, HEIF)	각 이미지 = 1페이지 단위	총 이미지
PDF	PDF의 각 페이지 = 1페이지 단위	PDF의 총 페이지 수
TIFF	TIFF의 각 이미지 = 1페이지 단위	TIFF의 총 이미지
Word(DOCX)	최대 3,000자 = 1페이지 단위, 포함되거나 연결된 이미지가 지원되지 않음	각각 최대 3,000자 의 총 페이지
Excel(XLSX)	개별 워크시트 = 1페이지 단위, 포함되거나 연결된 이미지가 지원되지 않음	전체 워크시트
PowerPoint(PPTX)	개별 슬라이드 = 1페이지 단위, 포함되거나 연결된 이미지가 지원되지 않음	전체 슬라이드
HTML	최대 3,000자 = 1페이지 단위, 포함되거나 연결된 이미지가 지원되지 않음	각각 최대 3,000자 의 총 페이지

JSON

```
"pages": [
  {
    "pageNumber": 1,
    "angle": 0,
    "width": 915,
    "height": 1190,
    "unit": "pixel",
    "words": [],
    "lines": [],
    "spans": []
  }
]
```

```
        }  
    ]
```

문서에서 선택한 페이지를 추출합니다.

대규모의 다중 페이지 문서의 경우 `pages` 쿼리 매개 변수를 사용하여 텍스트 추출을 위한 특정 페이지 번호 또는 페이지 범위를 지정합니다.

단락

레이아웃 모델은 `paragraphs` 컬렉션에서 식별된 모든 텍스트 블록을 `analyzeResults` 아래의 최상위 개체로 추출합니다. 이 컬렉션의 각 항목은 텍스트 블록을 나타내며 추출된 텍스트를 `content` 및 경계 `polygon` 좌표로 포함합니다. `span` 정보는 문서의 전체 텍스트가 포함된 최상위 `content` 속성 내의 텍스트 조각을 가리킵니다.

JSON

```
"paragraphs": [  
  {  
    "spans": [],  
    "boundingRegions": [],  
    "content": "While healthcare is still in the early stages of its AI  
journey, we are seeing pharmaceutical and other life sciences organizations  
making major investments in AI and related technologies.\\" TOM LAWRY |  
National Director for AI, Health and Life Sciences | Microsoft"  
  }  
]
```

단락 역할

새로운 기계 학습 기반 페이지 개체 검색은 제목, 섹션 머리글, 페이지 머리글, 페이지 바닥글 등과 같은 논리적 역할을 추출합니다. 문서 인텔리전스 레이아웃 모델은 모델이 예측하는 특수 역할 또는 형식을 사용하여 `paragraphs` 컬렉션의 특정 텍스트 블록을 할당합니다. 보다 풍부한 의미 체계 분석을 위해 추출된 콘텐츠의 레이아웃을 이해하는 데 도움이 되도록 구조화되지 않은 문서와 함께 사용하는 것이 가장 좋습니다. 지원되는 단락 역할은 다음과 같습니다.

[+] 테이블 확장

예측된 역할	설명	지원되는 파일 형식
<code>title</code>	페이지의 기본 제목	pdf, image, docx, pptx, xlsx, html

예측된 역할	설명	지원되는 파일 형식
sectionHeading	페이지에 있는 하나 이상의 부제목	pdf, image, docx, xlsx, html
footnote	페이지 아래쪽에 있는 텍스트	pdf, 이미지
pageHeader	페이지 위쪽 에지에 있는 텍스트	pdf, image, docx
pageFooter	페이지 아래쪽 에지에 있는 텍스트	pdf, image, docx, pptx, html
pageNumber	페이지 번호	pdf, 이미지

JSON

```
{
  "paragraphs": [
    {
      "spans": [],
      "boundingRegions": [],
      "role": "title",
      "content": "NEWS TODAY"
    },
    {
      "spans": [],
      "boundingRegions": [],
      "role": "sectionHeading",
      "content": "Mirjam Nilsson"
    }
  ]
}
```

텍스트, 선 및 단어

문서 인텔리전스의 문서 레이아웃 모델은 인쇄 및 필기 스타일 텍스트를 `lines` 과(와) `words` (으)로 추출합니다. `styles` 컬렉션에는 연결된 텍스트를 가리키는 범위와 함께 선에 대한 필기 스타일이 포함됩니다(탐지된 경우). 이 기능은 [지원되는 필기 언어](#)에 적용됩니다.

Microsoft Word, Excel, PowerPoint 및 HTML의 경우 문서 인텔리전스 버전 2024-02-29-preview 및 2023-10-31-preview 레이아웃 모델은 포함된 모든 텍스트를 있는 그대로 추출합니다. 텍스트는 단어와 단락으로 추출됩니다. 포함된 이미지는 지원되지 않습니다.

JSON

```
"words": [
  {
    "content": "While",
```

```
        "polygon": [],
        "confidence": 0.997,
        "span": {}
    },
],
"lines": [
{
    "content": "While healthcare is still in the early stages of its AI
journey, we",
    "polygon": [],
    "spans": [],
}
]
```

텍스트 줄에 대한 필기 스타일

응답에는 신뢰도 점수와 함께 각 텍스트 줄이 필기 스타일인지 여부를 분류하는 것이 포함됩니다. 자세한 내용은 필기 언어 지원을 [참조하세요](#). 다음 예제에서는 JSON 코드 조각 예제를 보여줍니다.

JSON

```
"styles": [
{
    "confidence": 0.95,
    "spans": [
    {
        "offset": 509,
        "length": 24
    }
    "isHandwritten": true
]
}
```

글꼴/스타일 추가 기능[도 사용하도록 설정](#)하면 개체의 `styles` 일부로 글꼴/스타일 결과가 표시됩니다.

선택 표시

또한 레이아웃 모델은 문서에서 선택 표시를 추출합니다. 추출된 선택 표시는 각 페이지의 `pages` 컬렉션 내에 표시됩니다. 여기에는 경계 `polygon`, `confidence` 및 선택 영역 `state (selected/unselected)`가 포함됩니다. 텍스트 표현(즉, `:selected:` 및 `:unselected:`)도 시작 인덱스(`offset`)로 포함되며 `length` 문서의 전체 텍스트를 포함하는 최상위 속성을 `content` 참조합니다.

JSON

```
{  
  "selectionMarks": [  
    {  
      "state": "unselected",  
      "polygon": [],  
      "confidence": 0.995,  
      "span": {  
        "offset": 1421,  
        "length": 12  
      }  
    }  
  ]  
}
```

테이블

테이블 추출은 일반적으로 테이블 형식이 지정된 대량의 데이터를 포함하는 문서를 처리하기 위한 주요 요구 사항입니다. 레이아웃 모델은 JSON 출력의 `pageResults` 섹션에서 테이블을 추출합니다. 추출된 테이블 정보에는 열 및 행 수, 행 범위 및 열 범위가 포함됩니다. 경계 다각형이 있는 각 셀은 영역이 `columnHeader`로 인식되는지 여부에 관계없이 정보와 함께 출력됩니다. 모델은 순환되는 테이블 추출을 지원합니다. 각 테이블 셀에는 행 및 열 인덱스와 경계 다각형 좌표가 포함됩니다. 셀 텍스트의 경우 모델은 시작 인덱스 (`offset`)가 포함된 `span` 정보를 출력합니다. 또한 모델은 문서의 전체 텍스트를 포함하는 최상위 수준 콘텐츠 내에서 `length`를 출력합니다.

① 참고

입력 파일이 XLSX인 경우 테이블이 지원되지 않습니다.

JSON

```
{  
  "tables": [  
    {  
      "rowCount": 9,  
      "columnCount": 4,  
      "cells": [  
        {  
          "kind": "columnHeader",  
          "rowIndex": 0,  
          "columnIndex": 0,  
          "columnSpan": 4,  
          "content": "(In millions, except earnings per share)",  
          "boundingRegions": [],  
          "spans": []  
        },  
        {  
          "kind": "cell",  
          "rowIndex": 0,  
          "columnIndex": 1,  
          "columnSpan": 1,  
          "content": "Revenue",  
          "boundingRegions": [{"x": 1421, "y": 1421, "x2": 1421, "y2": 1433}, {"x": 1421, "y": 1433, "x2": 1433, "y2": 1433}, {"x": 1421, "y": 1433, "x2": 1421, "y2": 1445}, {"x": 1421, "y": 1445, "x2": 1433, "y2": 1445}],  
          "spans": []  
        },  
        {  
          "kind": "cell",  
          "rowIndex": 0,  
          "columnIndex": 2,  
          "columnSpan": 1,  
          "content": "Profit",  
          "boundingRegions": [{"x": 1421, "y": 1421, "x2": 1421, "y2": 1433}, {"x": 1421, "y": 1433, "x2": 1433, "y2": 1433}, {"x": 1421, "y": 1433, "x2": 1421, "y2": 1445}, {"x": 1421, "y": 1445, "x2": 1433, "y2": 1445}],  
          "spans": []  
        },  
        {  
          "kind": "cell",  
          "rowIndex": 0,  
          "columnIndex": 3,  
          "columnSpan": 1,  
          "content": "EPS",  
          "boundingRegions": [{"x": 1421, "y": 1421, "x2": 1421, "y2": 1433}, {"x": 1421, "y": 1433, "x2": 1433, "y2": 1433}, {"x": 1421, "y": 1433, "x2": 1421, "y2": 1445}, {"x": 1421, "y": 1445, "x2": 1433, "y2": 1445}],  
          "spans": []  
        }  
      ]  
    }  
  ]  
}
```

```
        ]
    }
]
```

markdown 형식으로 출력

레이아웃 API는 추출된 텍스트를 markdown 형식으로 출력할 수 있습니다.

`outputContentFormat=markdown` 을 사용하여 markdown에서 출력 형식을 지정합니다.

markdown 콘텐츠는 `content` 섹션의 일부로 출력됩니다.

JSON

```
"analyzeResult": {
  "apiVersion": "2024-02-29-preview",
  "modelId": "prebuilt-layout",
  "contentFormat": "markdown",
  "content": "# CONTOSO LTD...",
}
```

수치

문서의 그림(차트, 이미지)은 텍스트 콘텐츠를 보완하고 향상시키는데 중요한 역할을 하며 복잡한 정보를 이해하는 데 도움이 되는 시각적 표현을 제공합니다. 레이아웃 모델에서 검색된 그림 개체에는 그림의 경계를 윤곽을 그리는 페이지 번호 및 다각형 좌표를 포함하여 문서 페이지에 있는 그림의 공간 위치) `spans` 와 같은 `boundingRegions` 주요 속성이 있습니다(그림과 관련된 텍스트 범위를 자세히 설명하고 문서 텍스트 내에서 오프셋과 길이를 지정합니다. 이 연결은 그림을 관련 텍스트 컨텍스트 `elements` 와 연결하고,(그림과 관련되거나 설명되는 문서 내의 텍스트 요소 또는 단락에 대한 식별자) 및 `caption` 있는 경우 도움이 됩니다.

JSON

```
{
  "figures": [
    {
      "boundingRegions": [],
      "spans": [],
      "elements": [
        "/paragraphs/15",
        ...
      ],
      "caption": {
        "text": "Figure 1: A line graph showing sales trends over time."}
```

```
        "content": "Here is a figure with some text",
        "boundingRegions": [],
        "spans": [],
        "elements": [
            "/paragraphs/15"
        ]
    }
}
]
```

섹션

계층적 문서 구조 분석은 광범위한 문서를 구성, 이해 및 처리하는 데 중요한 역할을 합니다. 이 접근 방식은 이해력을 높이고 탐색을 용이하게 하며 정보 검색을 개선하기 위해 긴 문서를 의미상 분할하는 데 매우 중요합니다. 문서 생성 AI에서 [RAG\(검색 보강 세대\)](#)의 출현은 계층적 문서 구조 분석의 중요성을 강조합니다. 레이아웃 모델은 각 섹션 내의 섹션과 개체의 관계를 식별하는 출력의 섹션 및 하위 섹션을 지원합니다. 계층 구조는 각 섹션에 `elements` 기본 출력을 사용하여 [markdown 형식](#) 을 사용하여 markdown에서 섹션 및 하위 섹션을 쉽게 가져올 수 있습니다.

JSON

```
{
    "sections": [
        {
            "spans": [],
            "elements": [
                "/paragraphs/0",
                "/sections/1",
                "/sections/2",
                "/sections/5"
            ]
        },
        ...
    }
```

다음 단계

- 문서 인텔리전스 스튜디오 [를](#) 사용하여 자체 양식과 문서를 처리하는 방법을 알아 봅니다.
- Document Intelligence 빠른 시작[을](#) 완료하고 원하는 개발 언어로 문서 처리 앱 만들기를 시작해 보세요.

문서 인텔리전스 추가 기능

아티클 • 2024. 03. 04.

① 중요

- 문서 인텔리전스 공개 미리 보기 릴리스에서는 현재 개발 중인 기능에 대한 조기 액세스를 제공합니다.
- 기능, 방식 및 프로세스는 GA(일반 공급) 전에 사용자 피드백에 따라 변경될 수 있습니다.
- 문서 인텔리전스 클라이언트 라이브러리의 공개 미리 보기 버전은 기본적으로 REST API 버전 [2023-10-31-preview](#)로 설정됩니다.
- 공개 미리 보기 버전 [2023-10-31-preview](#)는 현재 다음 Azure 지역에서만 사용 가능합니다.
 - 미국 동부
 - 미국 서부2
 - 서유럽

이 콘텐츠 적용 대상: v4.0(미리 보기) | 이전 버전: v3.1(GA)

문서 인텔리전스는 더욱 정교한 모듈식 분석 기능을 지원합니다. 추가 기능을 사용하여 문서에서 추출된 더 많은 기능이 포함되도록 결과를 확장합니다. 일부 추가 기능에서는 추가 비용이 발생합니다. 이러한 선택적 기능을 문서 추출 시나리오에 따라 사용하거나 사용하지 않도록 설정할 수 있습니다. 기능을 사용하도록 설정하려면 연결된 기능 이름을 `features` 쿼리 문자열 속성에 추가합니다. 쉼표로 구분된 기능 목록을 제공하여 요청에서 추가 기능을 2개 이상 사용하도록 설정할 수 있습니다. 다음 추가 기능은 [2023-07-31 \(GA\)](#) 이상 릴리스에 사용 가능합니다.

- `ocrHighResolution`
- `formulas`
- `styleFont`
- `barcodes`
- `languages`

② 참고

일부 추가 기능만 모든 모델에서 지원됩니다. 자세한 내용은 [모델 데이터 추출을 참조하세요.](#)

다음 추가 기능은 `2023-10-31-preview` 이상 릴리스에 사용 가능합니다.

- [keyValuePairs](#)
- [queryFields](#)

① 참고

2023-10-30-preview API의 쿼리 필드 구현은 마지막 미리 보기 릴리스와 다릅니다. 새 구현은 비용이 적게 들고 구조화된 문서에서 잘 작동합니다.

테이블 확장

추가 기능	추가 기능/ 무료	2023-10-31- preview	2023-07- 31(GA)	2022-08- 31(GA) ↗	v2.1(GA) ↗
Font 속성 추출	추가 기능	✓	✓	해당 없음	해당 없음
수식 추출	추가 기능	✓	✓	해당 없음	해당 없음
고해상도 추 출	추가 기능	✓	✓	해당 없음	해당 없음
바코드 추출	Free	✓	✓	해당 없음	해당 없음
언어 감지	Free	✓	✓	해당 없음	해당 없음
키 값 쌍	Free	✓	해당 없음	해당 없음	해당 없음
쿼리 필드	추가 기능*	✓	해당 없음	해당 없음	해당 없음

추가 기능* - 쿼리 필드 가격은 다른 추가 기능과 다르게 책정됩니다. 자세한 내용은 [가격 책정](#)을 참조하세요.

고해상도 추출

엔지니어링 드로잉과 같은 대형 문서에서 작은 텍스트를 인식하는 작업은 어려운 일입니다. 텍스트가 다른 그래픽 요소와 혼합되어 있는 경우가 많으며 글꼴, 크기 및 방향이 다양합니다. 또한 텍스트를 별도의 부분으로 나누거나 다른 기호와 연결할 수 있습니다. 이제 문서 인텔리전스는 `ocr.highResolution` 기능을 사용하여 이런 유형의 문서에서 콘텐츠

추출을 지원합니다. 이 추가 기능 기능을 사용하도록 설정하면 A1/A2/A3 문서에서 콘텐츠 추출 품질이 향상됩니다.

REST API

REST

```
https://{{your  
resource}}.cognitiveservices.azure.com/documentintelligence/documentModels/pr  
ebuilt-layout:analyze?api-version=2023-10-31-  
preview&features=ocrHighResolution
```

수식 추출

`ocr.formula` 기능은 `formulas` 컬렉션의 수학적 수식과 같은 식별된 모든 수식을 `content` 아래의 최상위 개체로 추출합니다. `content` 내부에서 검색된 수식은 `:formula:`로 표시됩니다. 이 컬렉션의 각 항목은 수식 형식을 `inline` 또는 `display`로 포함하고 해당 `polygon` 좌표와 함께 LaTeX 표현을 `value`로 포함하는 수식을 나타냅니다. 처음에는 수식이 각 페이지의 끝에 표시됩니다.

① 참고

`confidence` 점수는 하드 코딩됩니다.

JSON

```
"content": ":formula:",  
"pages": [  
  {  
    "pageNumber": 1,  
    "formulas": [  
      {  
        "kind": "inline",  
        "value": "\frac { \partial a } { \partial b }",  
        "polygon": [...],  
        "span": {...},  
        "confidence": 0.99  
      },  
      {  
        "kind": "display",  
        "value": "y = a \times b + a \times c",  
        "polygon": [...],  
        "span": {...},  
        "confidence": 0.99  
      }  
    ]  
  ]
```

```
        ]  
    }  
]
```

REST API

REST

```
https://{{your  
resource}}.cognitiveservices.azure.com/documentintelligence/documentModels/pr  
ebuilt-layout:analyze?api-version=2023-10-31-preview&features=formulas
```

Font 속성 추출

`ocr.font` 기능은 `styles` 컬렉션에서 추출된 텍스트의 모든 글꼴 속성을 `content` 아래의 최상위 개체로 추출합니다. 각 스타일 개체는 단일 글꼴 속성, 적용되는 텍스트 범위 및 해당 신뢰도 점수를 지정합니다. 기존 스타일 속성은 텍스트의 글꼴의 경우 `similarFontFamily`, 기울임꼴 및 일반 스타일의 경우 `fontStyle`, 굵게 또는 일반 스타일의 경우 `fontWeight`, 텍스트 색의 경우 `color`, 텍스트 경계 상자의 색의 경우 `backgroundColor`와 같은 더 많은 글꼴 속성으로 확장됩니다.

JSON

```
"content": "Foo bar",  
"styles": [  
  {  
    "similarFontFamily": "Arial, sans-serif",  
    "spans": [ { "offset": 0, "length": 3 } ],  
    "confidence": 0.98  
  },  
  {  
    "similarFontFamily": "Times New Roman, serif",  
    "spans": [ { "offset": 4, "length": 3 } ],  
    "confidence": 0.98  
  },  
  {  
    "fontStyle": "italic",  
    "spans": [ { "offset": 1, "length": 2 } ],  
    "confidence": 0.98  
  },  
  {  
    "fontWeight": "bold",  
    "spans": [ { "offset": 2, "length": 3 } ],  
    "confidence": 0.98  
  },  
  {  
    "color": "#FF0000",  
    "spans": [ { "offset": 3, "length": 1 } ],  
    "confidence": 0.98  
  }]
```

```
        "spans": [ { "offset": 4, "length": 2 } ],
        "confidence": 0.98
    },
    {
        "backgroundColor": "#00FF00",
        "spans": [ { "offset": 5, "length": 2 } ],
        "confidence": 0.98
    }
]
```

REST API

REST

```
https://{{your
resource}}.cognitiveservices.azure.com/documentintelligence/documentModels/pr
ebuilt-layout:analyze?api-version=2023-10-31-preview&features=styleFont
```

바코드 속성 추출

`ocr.barcode` 기능은 `barcodes` 컬렉션에서 식별된 모든 바코드를 `content` 아래의 최상위 개체로 추출합니다. `content` 내에서 검색된 바코드는 `:barcode:`로 표시됩니다. 이 컬렉션의 각 항목은 바코드를 나타내며 바코드 형식을 `kind`로, 포함된 바코드 콘텐츠를 `value`와 `polygon` 좌표로 포함합니다. 처음에는 바코드가 각 페이지의 끝에 표시됩니다. `confidence`는 1로 하드 코딩됩니다.

지원되는 바코드 유형

[+] 테이블 확장

바코드 유형	예제
QR Code	
Code 39	 0123456789ABC

바코드 유형 예제

Code 93



Code 128



UPC (UPC-A &
UPC-E)



PDF417



EAN-8



EAN-13

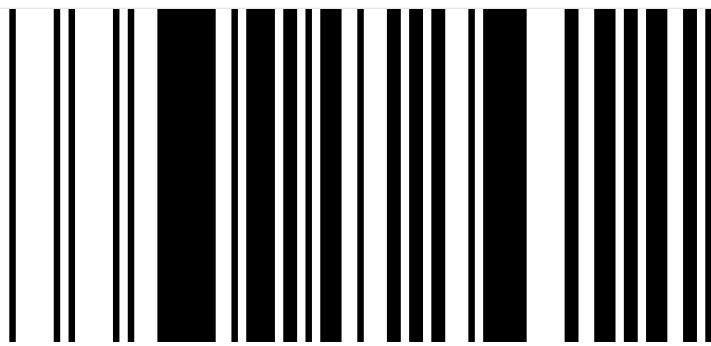


Codabar



바코드 유형 예제

Databar



Databar

Expanded



ITF



Data Matrix



REST API

REST

```
https://{{your  
resource}}.cognitiveservices.azure.com/documentintelligence/documentModels/pr  
ebuilt-layout:analyze?api-version=2023-10-31-preview&features=barcodes
```

언어 감지

`languages` 기능을 `analyzeResult` 요청에 추가하면 `analyzeResult`의 `languages` 컬렉션에 서 `confidence`와 함께 텍스트 줄마다 검색된 기본 언어가 예측됩니다.

JSON

```
"languages": [
  {
    "spans": [
      {
        "offset": 0,
        "length": 131
      }
    ],
    "locale": "en",
    "confidence": 0.7
  },
]
```

REST API

REST

```
https://{{your
resource}}.cognitiveservices.azure.com/documentintelligence/documentModels/pr
ebuilt-layout:analyze?api-version=2023-10-31-preview&features=languages
```

키-값 쌍

이전 API 버전에서 미리 빌드된 문서 모델은 양식과 문서에서 키-값 쌍을 추출했습니다. `keyValuePairs` 기능이 미리 빌드된 레이아웃에 추가되면서 이제 레이아웃 모델에서 같은 결과를 생성합니다.

키-값 쌍은 레이블 또는 키 및 이와 관련된 응답 또는 값을 식별하는 문서 내의 특정 범위입니다. 구조화된 양식에서 이러한 쌍은 레이블 및 사용자가 해당 필드에 입력한 값일 수 있습니다. 구조화되지 않은 문서에서는 단락의 텍스트를 기준으로 계약이 실행된 날짜일 수 있습니다. AI 모델은 다양한 문서 유형, 형식 및 구조를 기반으로 식별 가능한 키와 값을 추출하도록 학습되었습니다.

모델이 연결된 값이 없는 키가 존재하는 것을 감지하거나 선택적 필드를 처리할 때는 키가 격리되어 있을 수도 있습니다. 예를 들어 경우에 따라 양식에서 중간 이름 필드를 비워둘 수 있습니다. 키-값 쌍은 문서에 포함된 텍스트 범위입니다. 동일한 값이 다른 방식으로 설명되는 문서(예: 고객/사용자)가 있는 경우 연관된 키는 컨텍스트에 따라 고객 또는 사용자가 됩니다.

REST API

REST

```
https://{{your  
resource}}.cognitiveservices.azure.com/documentintelligence/documentModels/pr  
ebuilt-layout:analyze?api-version=2023-10-31-preview&features=keyValuePairs
```

쿼리 필드

쿼리 필드는 미리 빌드된 모델에서 추출된 스키마를 확장하거나 키 이름이 변수일 때 특정 키 이름을 정의하는 추가 기능입니다. 쿼리 필드를 사용하려면 기능을 `queryFields`로 설정하고 `queryFields` 속성에 쉼표로 구분된 필드 이름 목록을 제공합니다.

- 이제 문서 인텔리전스에서 쿼리 필드 추출을 지원합니다. 쿼리 필드 추출을 사용하면 추가 학습 없이도 쿼리 요청을 사용하여 추출 프로세스에 필드를 추가할 수 있습니다.
- 미리 빌드된 또는 사용자 지정 모델의 스키마를 확장하거나 레이아웃의 출력으로 몇 가지 필드를 추출해야 하는 경우에 쿼리 필드를 사용합니다.
- 쿼리 필드는 프리미엄 추가 기능입니다. 최상의 결과를 위해 여러 단어 필드 이름에 카멜 표기법 또는 파스칼 표기법 필드 이름을 사용하여 추출하려는 필드를 정의합니다.
- 쿼리 필드는 필드를 요청당 최대 20개까지 지원합니다. 문서에 필드 값이 포함되어 있으면 필드와 값이 반환됩니다.
- 이 릴리스에는 이전 구현보다 가격이 저렴하고 유효성을 검사해야 하는 쿼리 필드 기능이 새롭게 구현되어 있습니다.

① 참고

현재 2023-10-31-preview API부터 `us.tax.*` 모델(W2, 1098대 및 1099대 모델)을 제외한 이후 릴리스에서 레이아웃 및 미리 빌드된 모델과 함께 Document Intelligence Studio 쿼리 필드 추출을 사용할 수 있습니다.

쿼리 필드 추출

쿼리 필드 추출을 위해 추출하려는 필드를 지정하면 문서 인텔리전스가 그에 따라 문서를 분석합니다. 예를 들면 다음과 같습니다.

- 문서 인텔리전스 Studio [Studio](#)에서 계약을 처리하는 경우 2023-10-31-preview 버전을 사용하세요.

Layout

API version: 2023-10-31 (Preview) ▾

Service resource: fr-preview-demo [Edit](#)

Run analysis

Query fields

Analyze options

Drag & drop file here or
Browse for files or
Fetch from URL

- analyze document 요청의 일부로 Party1, Party2, TermsOfUse, PaymentTerms, PaymentDate 및 TermEndDate"와 같은 필드 레이블 목록을 전달할 수 있습니다.

Azure AI | Document Intelligence Studio

○ Azure Form Recognizer is now Azure AI Document Intelligence. [Learn more](#) about the latest updates to the service and the Studio experience.

Document Intelligence Studio > Layout

Layout

API version: 2023-10-31 (Preview) ▾ Service resource: fr-preview-demo

Run analysis

Query fields

Add fields you want to extract to the list and select the ones to be used in Analyze operations. Press Save to update all edits in the list. [Learn more about query fields](#).

Party1

Party2

TermsOfUse

PaymentTerms

PaymentDate

TermEndDate

Maximum of 6/20 fields

I acknowledge that using Query Fields will incur usage to my account. See [pricing](#).

Delete all

Save

Cancel

- 문서 인텔리전스는 필드 데이터를 분석 및 추출하고 구조화된 JSON 출력 값을 반환 할 수 있습니다.
- 쿼리 필드 외에도 응답에는 텍스트, 테이블, 선택 표시 및 기타 관련 데이터가 포함 됩니다.

REST API

REST

<https://{{your-resource}}.cognitiveservices.azure.com/documentintelligence/documentModels/pr>

ebuilt-layout:analyze?api-version=2023-10-31-preview&features=queryFields&queryFields=TERMS

다음 단계

자세히 알아보기: [모델 읽기](#) [레이아웃 모델](#).

SDK 샘플: [python](#)

문서 인텔리전스 쿼리 필드 추출

아티클 • 2024. 04. 08.

**문서 인텔리전스는 이제 쿼리 필드를 지원합니다. 이를 통해 미리 빌드된 모델의 스키마를 확장하여 필요한 특정 필드를 추출할 수 있습니다. 쿼리 필드를 레이아웃에 추가하여 양식이나 문서의 구조 외에 필드를 추출할 수도 있습니다.

① 참고

문서 인텔리전스 스튜디오 쿼리 필드 추출은 현재 UX.Tax 미리 빌드된 모델을 제외하고 레이아웃 및 미리 빌드된 모델에서 사용할 수 있습니다.

쿼리 필드 또는 키-값 쌍

쿼리 필드와 키-값 쌍은 수행하는 기능이 비슷합니다. 어떤 기능을 선택할지 결정할 때 알아야 할 몇 가지 차이점이 있습니다.

- 키-값 쌍은 레이아웃 및 청구서 모델에서만 사용할 수 있습니다. 미리 빌드된 모델의 스키마를 확장하려는 경우 쿼리 필드를 사용하세요.
- 추출할 특정 필드를 모르거나 필드 수가 큰 경우(20보다 큼) 키-값 쌍이 더 나은 솔루션일 수 있습니다.
- 키-값 쌍은 양식이나 문서에 있는 키와 값을 추출하므로 키 변형에 대비해야 합니다. 예: 키 `First Name` 또는 `Given Name`. 쿼리 필드의 경우 사용자는 키를 정의하고, 모델은 해당 값만 추출합니다.
- 필요한 값을 문서에서 키-값 쌍으로 설명할 수 없는 경우 쿼리 필드를 사용하세요. 계약의 계약 날짜를 예로 들 수 있습니다.

쿼리 필드 추출을 위해 추출하려는 필드를 지정하면 문서 인텔리전스가 그에 따라 문서를 분석합니다. 예를 들면 다음과 같습니다.

- 문서 인텔리전스 스튜디오 [에서 계약을 처리하는 경우 2024-02-29-preview, 2023-10-31-preview 이상의 API 버전을 사용하세요.](#)

Layout

API version: 2023-10-31 (Preview) ▾

Service resource: fr-preview-demo [Edit](#)

Run analysis

Query fields

Analyze options

Drag & drop file here or
Browse for files or
Fetch from URL

- AnalyzeDocument 요청의 일부로 Party1, Party2, TermsOfUse, PaymentTerms, PaymentDate 및 TermEndDate"와 같은 필드 레이블 목록을 전달할 수 있습니다.

Azure AI | Document Intelligence Studio

○ Azure Form Recognizer is now Azure AI Document Intelligence. [Learn more](#) about the latest updates to the service and the Studio experience.

Document Intelligence Studio > Layout

Layout

API version: 2023-10-31 (Preview) ▾ Service resource: fr-preview-dem

Run analysis

Query fields

Add fields you want to extract to the list and select the ones to be used in Analyze operations. Press Save to update all edits in the list. [Learn more about query fields](#).

Party1

Party2

TermsOfUse

PaymentTerms

PaymentDate

TermEndDate

Maximum of 6/20 fields

I acknowledge that using Query Fields will incur usage to my account. See [pricing](#).

Delete all

Save

Cancel

- 응답에는 쿼리 필드 외에도 모델 출력이 포함됩니다. 각 모델이 추출한 기능 또는 스키마 목록은 [모델 분석 기능](#)을 참조하세요.

쿼리 필드 REST API 요청**

일반 문서 모델과 쿼리 필드 기능을 사용하고 사용자 지정 모델을 학습시킬 필요 없이 추출 프로세스에 필드를 추가합니다.

HTTP

```
POST https://[endpoint]/documentintelligence/documentModels/prebuilt-
layout:analyze?api-version=2024-02-29-
preview&features=queryFields&queryFields=Terms,PaymentDate HTTP/1.1
Host: *.cognitiveservices.azure.com
Content-Type: application/json
Ocp-Apim-Subscription-Key:

{
    "urlSource": "https://raw.githubusercontent.com/Azure-Samples/cognitive-
services-REST-api-samples/master/curl/form-recognizer/sample-layout.pdf"
}
```

다음 단계

문서 인텔리전스 Studio 빠른 시작을 사용해 보세요.

다른 추가 기능에 대해 알아보기

Azure AI 문서 인텔리전스를 사용한 검색 증강 생성

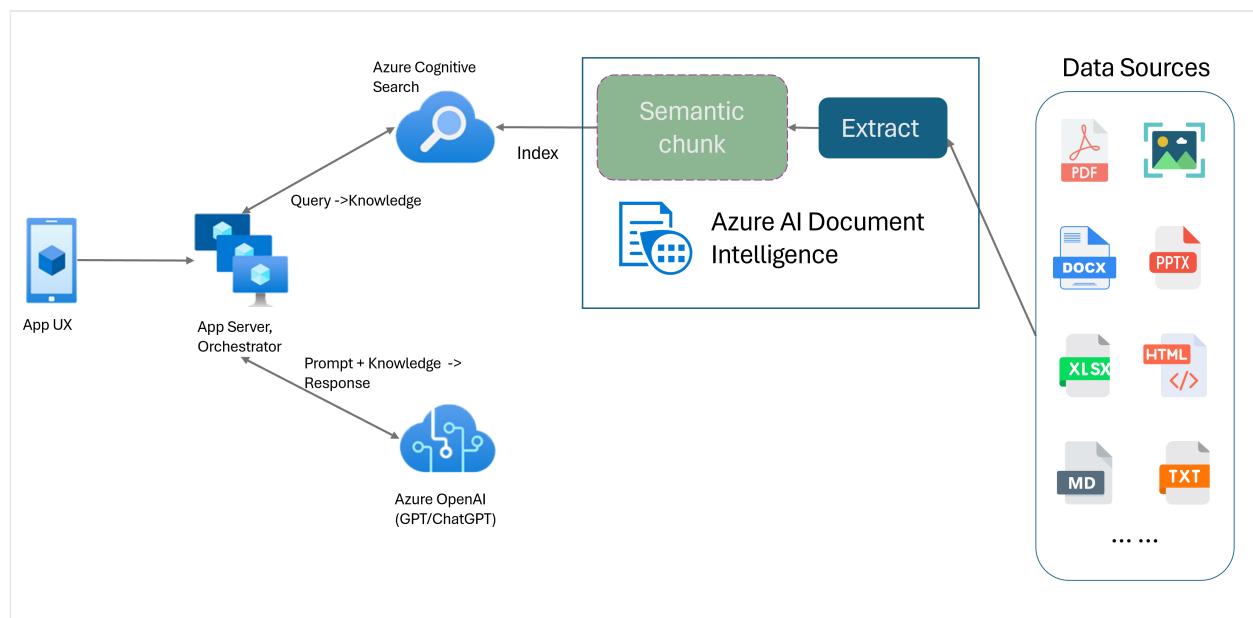
아티클 • 2024. 03. 08.

이 콘텐츠의 적용 대상은 다음과 같습니다. ✅ v4.0(미리 보기)

소개

RAG(검색 증강 생성)은 ChatGPT와 같은 미리 학습된 LLM(대규모 언어 모델)을 외부 데이터 검색 시스템과 결합하여 원래 학습 데이터 외부의 새 데이터를 통합하는 향상된 응답을 생성하는 디자인 패턴입니다. 애플리케이션에 정보 검색 시스템을 추가하면 문서와 채팅하고, 매력적인 콘텐츠를 생성하고, Azure OpenAI 모델의 강력한 데이터 성능에 액세스할 수 있습니다. 또한 응답을 작성할 때 LLM에서 사용하는 데이터의 제어가 향상됩니다.

문서 인텔리전스 [레이아웃 모델](#)은 고급 기계 학습 기반 문서 분석 API입니다. 레이아웃 모델은 고급 콘텐츠 추출 및 문서 구조 분석 기능을 위한 포괄적인 솔루션을 제공합니다. 레이아웃 모델을 사용하면 텍스트 및 구조 요소를 쉽게 추출하여 텍스트의 큰 본문을 임의의 분할이 아닌 의미 체계 콘텐츠에 따른 더 작고 의미 있는 청크로 나눌 수 있습니다. 추출된 정보를 Markdown 형식으로 편리하게 출력할 수 있으므로 제공된 구성 요소를 기반으로 의미 체계 청킹 전략을 정의할 수 있습니다.



의미 체계 청킹

긴 문장은 NLP(자연어 처리) 애플리케이션이 처리하기 어렵습니다. 문장이 여러 절, 복합 명사 또는 동사구, 관계절, 괄호 그룹화로 구성된 경우 특히 그렇습니다. 인간 관찰자와 마

찬가지로 NLP 시스템도 제시된 모든 종속성을 성공적으로 추적해야 합니다. 의미 체계 청킹의 목표는 문장 표현의 의미상 일관된 조각을 찾는 것입니다. 그런 다음 이러한 조각을 독립적으로 처리하고, 정보, 해석 또는 의미 체계 관련성의 손실 없이 의미 체계 표현으로 다시 결합할 수 있습니다. 텍스트의 내재된 의미는 청킹 프로세스의 지침으로 사용됩니다.

텍스트 데이터 청킹 전략은 RAG 응답 및 성능을 최적화하는 데 중요한 역할을 합니다. 고정 크기와 의미 체계는 두 가지 고유한 청킹 방법입니다.

- **고정 크기 청킹.** 오늘날 RAG에서 사용되는 대부분의 청킹 전략은 청크라고 하는 고정 크기의 텍스트 세그먼트를 기반으로 합니다. 고정 크기 청킹은 로그와 데이터처럼 강력한 의미 체계 구조가 없는 텍스트에서 빠르고 쉽고 효과적입니다. 하지만 의미 체계 이해와 정확한 컨텍스트가 필요한 텍스트에는 권장되지 않습니다. 창의 고정 크기 특성으로 인해 단어, 문장 또는 단락이 끊어져 이해가 방해되고 정보와 이해의 흐름이 중단될 수 있습니다.
- **의미 체계 청킹.** 이 방법은 의미 체계 이해를 기반으로 텍스트를 청크로 나눕니다. 나누기 경계는 문장 제목에 초점을 맞추고 중요한 계산 알고리즘으로 복잡한 리소스를 사용합니다. 하지만 각 청크 내에서 의미 체계 일관성을 유지한다는 고유한 이점이 있습니다. 텍스트 요약, 감정 분석, 문서 분류 작업에 유용합니다.

문서 인텔리전스 레이아웃 모델을 사용한 의미 체계 청킹

Markdown은 구조화되고 형식이 지정된 마크업 언어이며 RAG(검색 증강 생성)에서 의미 체계 청킹이 가능하도록 하는 인기 있는 입력입니다. [레이아웃 모델](#)에서 Markdown 콘텐츠를 사용하여 단락 경계를 기반으로 문서를 분할하고, 테이블의 특정 청크를 만들고, 청킹 전략을 미세 조정하여 생성된 응답의 품질을 향상시킬 수 있습니다.

레이아웃 모델 사용의 이점

- **간소화된 처리.** 단일 API 호출로 디지털 및 스캔된 PDF, 이미지, Office 파일(docx, xlsx, pptx), HTML과 같은 다양한 문서 형식을 구문 분석할 수 있습니다.
- **확장성 및 AI 품질** 레이아웃 모델은 OCR(광학 인식), 테이블 추출, [문서 구조 분석](#)에서 확장성이 높습니다. 이 모델은 [309개의 인쇄 언어와 12개의 필기 언어](#)를 지원하여 AI 기능에 기반한 고품질 결과를 보장합니다.
- **LLM(대규모 언어 모델) 호환성.** 레이아웃 모델 Markdown 형식 출력은 LLM 친화적이며, 워크플로에 원활하게 통합할 수 있습니다. 문서의 테이블을 Markdown 형식으

로 전환하고 문서를 구문 분석하는 데 드는 광범위한 노력을 방지하여 LLM 이해를 높일 수 있습니다.

문서 인텔리전스 스튜디오를 사용하여 처리되고 레이아웃 모델을 사용하여 Markdown으로 출력된 텍스트 이미지

The screenshot shows a Microsoft Word document with various sections labeled by arrows pointing to specific parts of the interface:

- Title**: Points to the title bar.
- Page header**: Points to the page header "Latest news and bulletin update".
- Section heading**: Points to the section heading "The scoop of the day".
- Paragraph**: Points to a paragraph of text.
- Picture caption**: Points to a picture caption.
- Page number**: Points to the page number "Page XX".

Below the document, the generated Markdown code is shown:

```
<!-- PageHeader="Tuesday, Sep 20, YYYY" -->
NEWS TODAY

<!-- PageHeader="Latest news and bulletin updates" --> <!--
PageHeader="Issue \#10" -->
Mirjam Nilsson

The scoop of the day
The latest updates to get you through the day

Video provides a powerful way to help you prove your point. When you click Online Video, you can paste in the embed code for the video you want to add. You can also type a keyword to search for the video that best fits your document.

Themes and styles also help keep your document coordinated. When you click Design and choose a new theme, Word provides header, footer, cover page, and text box designs that complement each other. For example, you can add a matching cover page, header, and sidebar.

Click Insert and then choose the elements you want from the different galleries.

Themes and styles also help keep your document coordinated. When you click Design and choose a new theme, Word provides header, footer, cover page, and text box designs that complement each other. For example, you can add a matching cover page, header, and sidebar.

Themes and styles also help keep your document coordinated. When you click Design and choose a new theme, Word provides header, footer, cover page, and text box designs that complement each other. For example, you can add a matching cover page, header, and sidebar.

Themes and styles also help keep your document coordinated. When you click Design and choose a new theme, Word provides header, footer, cover page, and text box designs that complement each other. For example, you can add a matching cover page, header, and sidebar.

Themes and styles also help keep your document coordinated. When you click Design and choose a new theme, Word provides header, footer, cover page, and text box designs that complement each other. For example, you can add a matching cover page, header, and sidebar.
```

레이아웃 모델을 사용하여 문서 인텔리전스 스튜디오로 처리된 텍스트 이미지

NOTE 2 — EARNINGS PER SHARE

Basic earnings per share ("EPS") is computed based on the weighted average number of shares of common stock outstanding during the period. Diluted EPS is computed based on the weighted average number of shares of common stock plus the effect of dilutive potential common shares outstanding during the period using the treasury stock method. Dilutive potential common shares include outstanding stock options and stock awards.

The components of basic and diluted EPS were as follows:

(In millions, except earnings per share)	2021	2020	2019
Year Ended June 30,			
Net income available for common shareholders (A)	\$ 61,271	\$ 44,281	\$ 39,240
Weighted average outstanding shares of common stock (B)	7,547	7,610	7,673
Dilutive effect of stock-based awards	61	73	80
Common stock and common stock equivalents (C)	7,608	7,683	7,753
Earnings Per Share			
Basic (A/B)	\$ 8.12	\$ 5.82	\$ 5.11
Diluted (A/C)	\$ 8.05	\$ 5.76	\$ 5.06

Table



(In millions, except earnings per share)	2021	2020	2019
Year Ended June 30,			
Net income available for common shareholders (A)	\$ 61,271	\$ 44,281	\$ 39,240
Weighted average outstanding shares of common stock (B)	7,547	7,610	7,673
Dilutive effect of stock-based awards	61	73	80
Common stock and common stock equivalents (C)	7,608	7,683	7,753
Earnings Per Share			
Basic (A/B)	\$ 8.12	\$ 5.82	\$ 5.11
Diluted (A/C)	\$ 8.05	\$ 5.76	\$ 5.06

시작하기

문서 인텔리전스 레이아웃 모델 2024-02-29-preview 및 2023-10-31-preview 는 다음 개발 옵션을 지원합니다.

- Document Intelligence Studio [↗](#).
- REST API.
- .NET • Java • JavaScript • Python 프로그래밍 언어 SDK(클라이언트 라이브러리).

시작할 준비가 되셨나요?

문서 인텔리전스 스튜디오

문서 인텔리전스 스튜디오 빠른 시작에 따라 시작할 수 있습니다. 다음으로 제공된 샘플 코드를 사용하여 문서 인텔리전스 기능을 자체 애플리케이션에 통합할 수 있습니다.

- 레이아웃 모델 [로](#) 시작하세요. 스튜디오에서 RAG를 사용하려면 다음 분석 옵션을 선택해야 합니다.

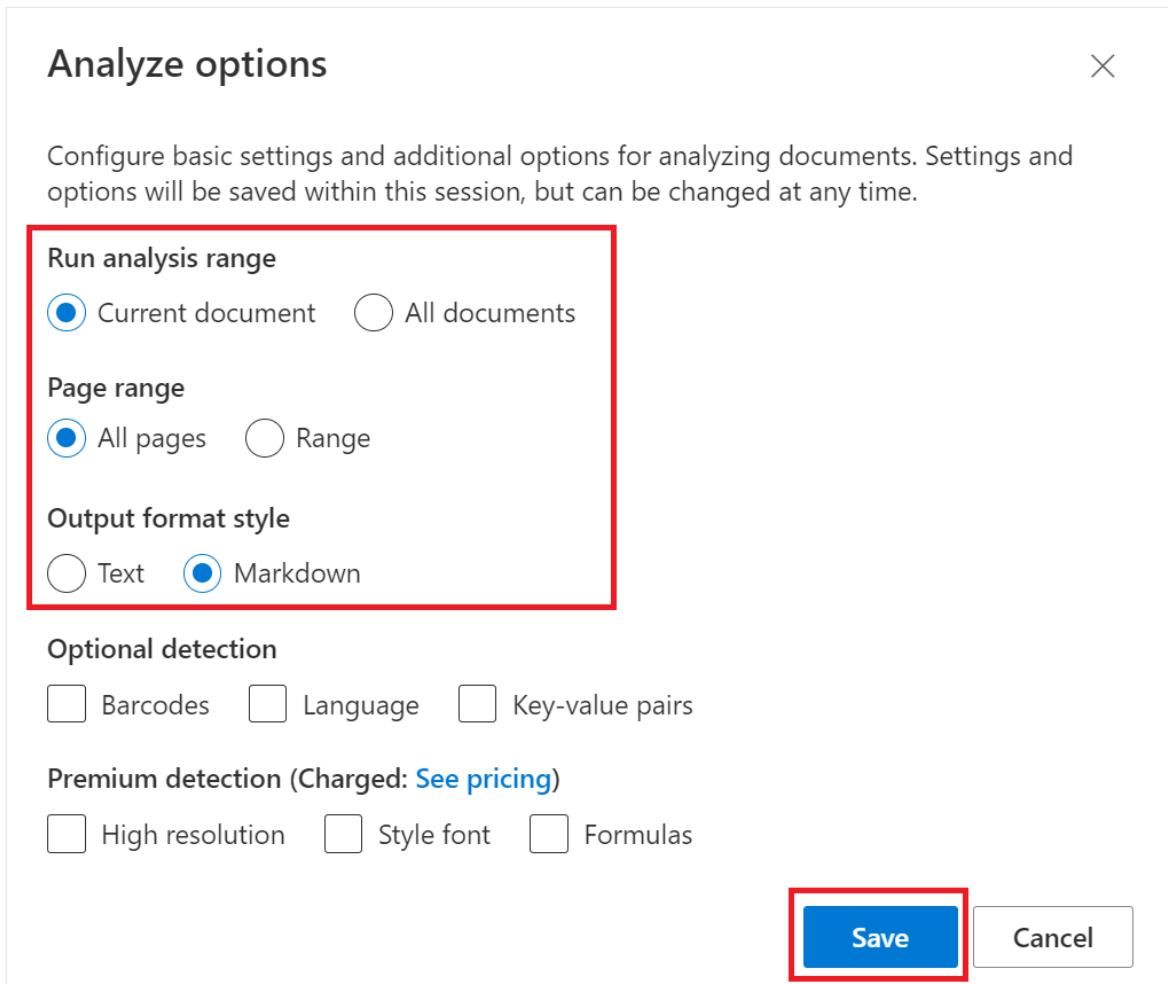
****Required****

- 현재 문서에 → 분석 범위를 실행합니다.
- 페이지 범위 → **모든 페이지**
- Markdown을 → 출력 형식 스타일입니다.

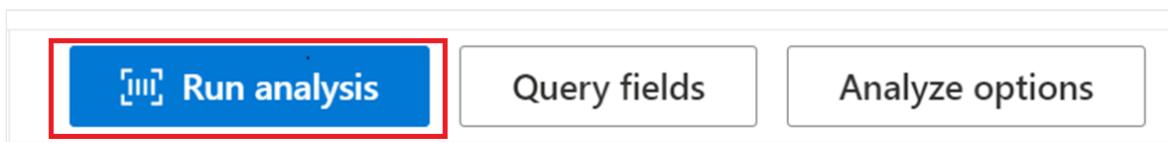
****Optional****

- 관련 선택적 감지 매개 변수를 선택할 수도 있습니다.

- 저장을 선택합니다.



- 출력을 보려면 **분석 실행** 단추를 선택합니다.



SDK 또는 REST API

- 기본 프로그래밍 언어 SDK 또는 REST API에 대한 문서 인텔리전스 빠른 시작을 따를 수 있습니다. 레이아웃 모델을 사용하여 문서에서 콘텐츠와 구조를 추출합니다.
- Markdown 출력 형식의 문서를 분석하기 위해 GitHub 리포지토리에서 코드 샘플과 팁을 확인할 수도 있습니다.
 - [Python ↗](#)
 - [JavaScript ↗](#)
 - [Java ↗](#)
 - [.NET ↗](#)

의미 체계 청킹을 사용하여 문서 채팅 빌드

- [Azure OpenAI on Your Data](#)를 사용하면 문서에서 지원되는 채팅을 실행할 수 있습니다. Azure OpenAI on Your Data는 문서 인텔리전스 레이아웃 모델을 적용하여 테이블과 단락을 기반으로 긴 텍스트를 청킹함으로써 문서를 추출하고 구문 분석합니다. GitHub 리포지토리에 있는 [Azure OpenAI 샘플 스크립트 ↗](#)를 사용하여 청킹 전략을 사용자 지정할 수도 있습니다.
- Azure AI 문서 인텔리전스는 이제 문서 로더 중 하나로 [LangChain ↗](#)과 통합되었습니다. 이를 사용하여 쉽게 데이터를 로드하고 Markdown 형식으로 출력할 수 있습니다. 자세한 내용은 LangChain에서 Azure AI 문서 인텔리전스를 문서 로더로, Azure Search를 검색기로 사용하는 RAG 패턴의 간단한 데모를 보여 주는 [샘플 코드 ↗](#)를 참조하세요.
- 데이터 솔루션 가속기 [코드 샘플 ↗](#)과의 채팅은 엔드 투 엔드 기준선 RAG 패턴 샘플을 보여 줍니다. 이 샘플은 Azure AI Search를 검색기로 사용하고 Azure AI 문서 인텔리전스를 문서 로딩과 의미 체계 청킹에 사용합니다.

사용 사례

문서에서 특정 섹션을 찾는 경우 의미 체계 청킹을 사용하여 섹션 머리글을 기반으로 문서를 더 작은 척크로 나누면 원하는 섹션을 빠르고 쉽게 찾는데 도움이 됩니다.

Python

```
# Using SDK targeting 2024-02-29-preview or 2023-10-31-preview, make sure
your resource is in one of these regions: East US, West US2, West Europe
# pip install azure-ai-documentintelligence==1.0.0b1
# pip install langchain langchain-community azure-ai-documentintelligence
```

```

from azure.ai.documentintelligence import DocumentIntelligenceClient

endpoint = "https://<my-custom-subdomain>.cognitiveservices.azure.com/"
key = "<api_key>"

from langchain_community.document_loaders import
AzureAIDocumentIntelligenceLoader
from langchain.text_splitter import MarkdownHeaderTextSplitter

# Initiate Azure AI Document Intelligence to load the document. You can
either specify file_path or url_path to load the document.
loader = AzureAIDocumentIntelligenceLoader(file_path=<path to your file>,
api_key = key, api_endpoint = endpoint, api_model="prebuilt-layout")
docs = loader.load()

# Split the document into chunks base on markdown headers.
headers_to_split_on = [
    ("#", "Header 1"),
    ("##", "Header 2"),
    ("###", "Header 3"),
]
text_splitter =
MarkdownHeaderTextSplitter(headers_to_split_on=headers_to_split_on)

docs_string = docs[0].page_content
splits = text_splitter.split_text(docs_string)
splits

```

다음 단계

- Azure AI 문서 인텔리전스에 대해 자세히 알아봅니다.
- 문서 인텔리전스 스튜디오 [☞](#)를 사용하여 자체 양식과 문서를 처리하는 방법을 알아봅니다.
- Document Intelligence 빠른 시작을 완료하고 원하는 개발 언어로 문서 처리 앱 만들기를 시작해 보세요.

문서 인텔리전스 신용 카드 모델

아티클 • 2024. 04. 14.

이 콘텐츠의 적용 대상은 다음과 같습니다. v4.0(미리 보기)

문서 인텔리전스 신용/직불 카드 모델은 강력한 광학 문자 인식(OCR) 기능을 사용하여 신용 카드와 직불 카드에서 주요 필드를 분석하고 추출합니다. 신용 카드와 직불 카드는 휴대폰으로 캡처한 이미지, 스캔한 문서, 디지털 PDF 등 형식과 품질이 다양할 수 있습니다. API는 문서 텍스트를 분석합니다. 카드 번호, 발급 은행 및 만료 날짜와 같은 주요 정보를 추출하고 구조화된 JSON 데이터 표현을 반환합니다. 이 모델은 현재 영어 문서 형식을 지원합니다.

자동화된 카드 처리

자동 신용/직불 카드 처리는 은행 카드에서 주요 필드를 추출하는 프로세스입니다. 지금 까지 은행 카드 분석 프로세스는 수동으로 수행되었으므로 시간이 많이 걸렸습니다. 은행 카드에서 주요 데이터를 정확하게 추출하는 것은 일반적으로 계약 자동화 프로세스에서 가장 중요한 단계 중 첫 번째이자 가장 중요한 단계 중 하나입니다.

개발 옵션

문서 인텔리전스 v4.0(2024-02-29-preview)은 다음 도구, 애플리케이션, 라이브러리를 지원합니다.

테이블 확장

기능	리소스	Model ID
계약 모델	<ul style="list-style-type: none">문서 인텔리전스 스튜디오REST APIC# SDKPython SDKJava SDKJavaScript SDK	prebuilt-creditCard

입력 요구 사항

- 최상의 결과를 위해 문서당 하나의 명확한 사진 또는 고품질 스캔을 제공합니다.
- 지원 파일 형식:

모델	PDF	이미지: JPEG/JPG, PNG, BMP, TIFF, HEIF	Microsoft Office: Word(DOCX), Excel(XLSX), PowerPoint(PPTX) 및 HTML
읽음	✓	✓	✓
레이아웃	✓	✓	✓ (2024-02-29-preview, 2023-10-31-preview)
일반 문서	✓	✓	
사전 제작	✓	✓	
사용자 지정 추출	✓	✓	
사용자 지정 분류	✓	✓	✓ (2024-02-29-preview)

- PDF 및 TIFF의 경우 최대 2000페이지를 처리할 수 있습니다(무료 계층 구독의 경우 처음 2페이지만 처리됨).
- 문서 분석을 위한 파일 크기는 유료(S0) 계층의 경우 500MB, 무료(F0) 계층의 경우 4MB입니다.
- 이미지 크기는 50 x 50픽셀에서 10,000 x 10,000픽셀 사이여야 합니다.
- PDF가 암호로 잠긴 경우에는 제출하기 전에 잠금을 해제해야 합니다.
- 추출할 텍스트의 최소 높이는 1024 x 768 픽셀 이미지의 경우 12픽셀입니다. 이 차원은 150DPI(인치당 도트 수)에서 약 8포인트 텍스트에 해당합니다.
- 사용자 지정 모델 학습의 경우 학습 데이터의 최대 페이지 수는 사용자 지정 템플릿 모델의 경우 500개, 사용자 지정 인공신경망 모델의 경우 50,000개입니다.
 - 사용자 지정 추출 모델 학습의 경우 학습 데이터의 총 크기는 템플릿 모델의 경우 50MB, 신경망 모델의 경우 1G-MB입니다.
 - 사용자 지정 분류 모델 학습의 경우 학습 데이터의 총 크기는 최대 10,000페이지인 1GB입니다.

신용 카드 데이터 추출 사용해 보기

신용/직불 카드 서비스에서 데이터 추출이 작동하는 방식을 확인하려면 다음 리소스가 필요합니다.

- Azure 구독은 무료로 만들 수 있습니다.
- Azure Portal의 Document Intelligence 인스턴스입니다. 무료 가격 책정 계층(F0)을 사용하여 서비스를 시도할 수 있습니다. 리소스가 배포된 후 리소스로 이동을 선택하여 키 및 엔드포인트를 가져옵니다.

Home > Contoso-DI

Contoso-DI | Keys and Endpoint

Search | Regenerate Key1 | Regenerate Key2

- Overview
- Activity log
- Access control (IAM)
- Tags
- Diagnose and solve problems
- Resource Management
 - Keys and Endpoint**
 - Encryption
 - Pricing tier
 - Networking
 - Identity
 - Cost analysis
 - Properties
 - Locks
- Monitoring
- Automation
- Help

These keys are used to access your Azure AI service API. Do not share your keys. Store them securely—for example, using Azure Key Vault. We also recommend regenerating these keys regularly. Only one key is necessary to make an API call. When regenerating the first key, you can use the second key for continued access to the service.

Show Keys

KEY 1	Copy
KEY 2	Copy
Location/Region	westus2	Copy
Endpoint	https://contoso-di.cognitiveservices.azure.com/	Copy

Document Intelligence Studio

1. 문서 인텔리전스 스튜디오 홈페이지에서 **신용/직불 카드**를 선택합니다.
2. 샘플 세금 문서를 분석하거나 자체 파일을 업로드할 수 있습니다.
3. **분석 실행** 버튼을 선택하고 필요한 경우 **분석 옵션**을 구성합니다.



[Document Intelligence Studio 사용해 보기](#)

지원되는 언어 및 로캘

지원되는 언어의 전체 목록은 [언어 지원 - 미리 빌드된 모델 페이지](#)를 참조하세요.

필드 추출

다음은 JSON 출력 응답의 계약에서 추출된 필드입니다.

[\[+\] 테이블 확장](#)

이름	형식	설명	예제 출력
CardNumber	문자열	카드의 고유 식별자	4275 0000 0000 0000
IssuingBank	문자열	카드를 발급한 은행의 이름	Woodgrove Bank
PaymentNetwork	문자열	카드 거래를 처리하는 결제 네트워크	VISA
CardHolderName	문자열	카드를 소유한 사람의 이름	JOHN SMITH
CardHolderCompanyName	문자열	카드가 연결된 회사의 이름	Contoso, Ltd.
ValidDate	날짜	유효 기간 시작	01/16
ExpirationDate	날짜	만료 날짜	01/19
CardVerificationValue	문자열	카드 보안 코드(CVV)	764
CustomerServicePhoneNumbers	Array	지원 번호 목록	+1 (555) 123-4567

추출된 은행 카드 키-값 쌍 및 품목은 JSON 출력의 `documentResults` 섹션에 있습니다.

다음 단계

- 문서 인텔리전스 스튜디오 [\[↗\]](#)를 사용하여 사용자 고유의 양식 및 문서를 처리해 보세요.
- Document Intelligence 빠른 시작을 완료하고 원하는 개발 언어로 문서 처리 앱 만들기를 시작해 보세요.

문서 인텔리전스 모기지 문서 모델

아티클 • 2024. 04. 15.

이 콘텐츠의 적용 대상은 다음과 같습니다. v4.0(미리 보기)

문서 인텔리전스 모기지 모델은 강력한 OCR(광학 인식) 기능과 딥 러닝 모델을 사용하여 모기지 문서에서 주요 필드를 분석하고 추출합니다. 모기지 문서 형식과 품질은 다양할 수 있습니다. API는 모기지 문서를 분석하고 구조화된 JSON 데이터 표현을 반환합니다. 모델은 현재 영어 문서만 지원합니다.

지원되는 문서 유형:

- 1003 EULA(최종 사용자 사용권 계약)
- 양식 1008
- 모기지 마감 공시

개발 옵션

문서 인텔리전스 v4.0(2024-02-29-미리 보기)은 다음 도구, 애플리케이션, 라이브러리를 지원합니다.

테이블 확장

기능	리소스	Model ID
모기지 모델	<ul style="list-style-type: none">문서 인텔리전스 스튜디오REST APIC# SDKPython SDKJava SDKJavaScript SDK	<ul style="list-style-type: none">prebuilt-mortgage.us.1003prebuilt-mortgage.us.1008prebuilt-mortgage.us.closingDisclosure

입력 요구 사항

- 최상의 결과를 위해 문서당 하나의 명확한 사진 또는 고품질 스캔을 제공합니다.
- 지원 파일 형식:

테이블 확장

모델	PDF	이미지: JPEG/JPG, PNG, BMP, TIFF, HEIF	Microsoft Office: Word(DOCX), Excel(XLSX), PowerPoint(PPTX) 및 HTML
읽음	✓	✓	✓
레이아웃	✓	✓	✓ (2024-02-29-미리 보기, 2023-10-31-미리 보기)
일반 문서	✓	✓	
사전 제작	✓	✓	
사용자 지정 추출	✓	✓	
사용자 지정 분류	✓	✓	✓ (2024-02-29-미리 보기)

- PDF 및 TIFF의 경우 최대 2000페이지를 처리할 수 있습니다(무료 계층 구독의 경우 처음 2페이지만 처리됨).
- 문서 분석을 위한 파일 크기는 유료(S0) 계층의 경우 500MB, 무료(F0) 계층의 경우 4MB입니다.
- 이미지 크기는 50 x 50픽셀에서 10,000 x 10,000픽셀 사이여야 합니다.
- PDF가 암호로 잠긴 경우에는 제출하기 전에 잠금을 해제해야 합니다.
- 추출할 텍스트의 최소 높이는 1024 x 768 픽셀 이미지의 경우 12픽셀입니다. 이 차원은 150DPI(인치당 도트 수)에서 약 8포인트 텍스트에 해당합니다.
- 사용자 지정 모델 학습의 경우 학습 데이터의 최대 페이지 수는 사용자 지정 템플릿 모델의 경우 500개, 사용자 지정 인공신경망 모델의 경우 50,000개입니다.
 - 사용자 지정 추출 모델 학습의 경우 학습 데이터의 총 크기는 템플릿 모델의 경우 50MB, 신경망 모델의 경우 1G-MB입니다.
 - 사용자 지정 분류 모델 학습의 경우 학습 데이터의 총 크기는 최대 10,000페이지인 1GB입니다.

모기지 문서 데이터 추출 시도

모기지 문서 서비스에서 데이터 추출이 작동하는 방식을 확인하려면 다음 리소스가 필요합니다.

- Azure 구독은 [무료로 만들 수](#) 있습니다.

- Azure Portal의 Document Intelligence 인스턴스입니다. 무료 가격 책정 계층(F0)을 사용하여 서비스를 시도할 수 있습니다. 리소스가 배포된 후 리소스로 이동을 선택하여 키 및 엔드포인트를 가져옵니다.

Home > Contoso-DI

Contoso-DI | Keys and Endpoint

Search Regenerate Key1 Regenerate Key2

- Overview
- Activity log
- Access control (IAM)
- Tags
- Diagnose and solve problems
- Resource Management
- Keys and Endpoint**
- Encryption
- Pricing tier
- Networking
- Identity
- Cost analysis
- Properties
- Locks
- Monitoring
- Automation
- Help

These keys are used to access your Azure AI service API. Do not share your keys. Store them securely—for example, using Azure Key Vault. We also recommend regenerating these keys regularly. Only one key is necessary to make an API call. When regenerating the first key, you can use the second key for continued access to the service.

Show Keys

KEY 1	Copy
KEY 2	Copy
Location/Region	westus2	Copy
Endpoint	https://contoso-di.cognitiveservices.azure.com/	Copy

Document Intelligence Studio

1. 문서 인텔리전스 스튜디오 홈페이지에서 **모기지**를 선택합니다.
2. 샘플 모기지 문서를 분석하거나 사용자 고유의 파일을 업로드할 수 있습니다.
3. **분석 실행** 버튼을 선택하고 필요한 경우 **분석 옵션**을 구성합니다.



[Document Intelligence Studio 사용해 보기](#)

지원되는 언어 및 로캘

지원되는 언어의 전체 목록은 [언어 지원 - 미리 빌드된 모델 페이지](#)를 참조하세요.

필드 추출 1003 EULA(최종 사용자 사용권 계약)

다음은 JSON 출력 응답의 1003 EULA 양식에서 추출된 필드입니다.

[+] 테이블 확장

속성	형식	설명	예제 출력
LenderLoanNumber	문자열	대출 번호 또는 범용 대출 식별자	10Bx939c5543TqA1144M999143X38
AgencyCaseNumber	문자열	에이전시 사례 번호	115894
빌어 쓰는 사람	Object	이름, SSN, 생년월일 등 차용자의 ID 표식을 포함하는 개체입니다.	
공동 대출자	Object	공동 대출자의 이름과 서명된 날짜가 포함된 개체입니다.	
CurrentEmployment	Object	고용주 이름, 고용주 전화 번호, 고용주 주소 등 현재 고용에 대한 정보를 포함하는 개체입니다.	
대출	Object	금액, 목적 유형, 재융자 유형 등 대출 정보를 포함하는 개체입니다.	
속성	개체	주소, 단위 수, 값 등 속성에 대한 정보를 포함하는 개체입니다.	

추출된 1003 EULA 키-값 쌍 및 품목은 JSON 출력의 `documentResults` 섹션에 있습니다.

필드 추출 양식 1008

다음은 JSON 출력 응답의 1008 양식에서 추출된 필드입니다.

[+] 테이블 확장

속성	형식	설명	예제 출력
빌어 쓰는 사람	Object	이름 및 대출자 수를 포함하여 대출자에 대한 정보를 포함하는 개체입니다.	

속성	형식	설명	예제 출력
사람		체입니다.	
속성	Object	주소, 점유율 상태, 판매 가격 등 속성에 대한 정보를 포함하는 개체입니다.	
모기지	Object	대출 유형, 상각 유형, 대출 목적 유형 등 모기지에 대한 정보를 포함하는 개체입니다.	
인수	Object	보험업자 이름, 평가자 이름, 대출자 소득 등 인수 정보에 대한 정보가 포함된 개체입니다.	
판매자	Object	이름, 주소, 번호 등 판매자에 대한 정보를 포함하는 개체입니다.	

추출된 양식 1008 키-값 쌍 및 줄 항목은 JSON 출력의 `documentResults` 섹션에 있습니다.

현장 추출 모기지 마감 공시

다음은 JSON 출력 응답의 모기지 마감 공시 양식에서 추출된 필드입니다.

[+] 테이블 확장

속성	형식	설명	예제 출력
닫기	Object	발급 날짜, 마감 날짜, 지출 날짜 등 닫는 정보에 대한 정보가 들어 있는 개체입니다.	
트랜잭션	Object	대출자 이름, 대출자 주소, 판매자 이름 등 거래 정보에 대한 정보를 포함하는 개체입니다.	
대출	Object	용어, 목적, 제품 등 대출 정보를 포함하는 개체입니다.	

추출된 모기지 마감 공시 키-값 쌍 및 품목은 JSON 출력의 `documentResults` 섹션에 있습니다.

다음 단계

- 문서 인텔리전스 스튜디오 [\[로고\]](#)를 사용하여 사용자 고유의 양식 및 문서를 처리해 보세요.
- Document Intelligence 빠른 시작을 완료하고 원하는 개발 언어로 문서 처리 앱 만들기를 시작해 보세요.

문서 인텔리전스 결혼 인증서 모델

아티클 • 2024. 04. 12.

이 콘텐츠의 적용 대상은 다음과 같습니다. v4.0(미리 보기)

문서 인텔리전스 결혼 인증서 모델은 강력한 광학 문자 인식(OCR) 기능을 사용하여 결혼 인증서에서 주요 필드를 분석하고 추출합니다. 결혼 인증서는 휴대폰으로 캡처한 이미지, 스캔한 문서, 디지털 PDF 등 형식과 품질이 다양할 수 있습니다. API는 문서 텍스트를 분석합니다. 배우자 이름, 발행일 및 결혼 장소와 같은 주요 정보를 추출합니다. 구조화된 JSON 데이터 표현을 반환합니다. 이 모델은 현재 영어 문서 형식을 지원합니다.

자동화된 결혼 인증서 처리

자동화된 결혼 인증서 처리는 결혼 인증서에서 키 필드를 추출하는 프로세스입니다. 지금 까지 결혼 증명서 분석 프로세스는 수동으로 수행되었으므로 시간이 많이 걸렸습니다. 결혼 인증서에서 핵심 데이터를 정확하게 추출하는 것은 일반적으로 결혼 인증서 자동화 프로세스에서 첫 번째이자 가장 중요한 단계 중 하나입니다.

개발 옵션

문서 인텔리전스 v4.0(2024-02-29-preview)은 다음 도구, 애플리케이션, 라이브러리를 지원합니다.

테이블 확장

기능	리소스	Model ID
계약 모델	<ul style="list-style-type: none">문서 인텔리전스 스튜디오REST APIC# SDKPython SDKJava SDKJavaScript SDK	prebuilt-marriageCertificate.us

입력 요구 사항

- 최상의 결과를 위해 문서당 하나의 명확한 사진 또는 고품질 스캔을 제공합니다.
- 지원 파일 형식:

모델	PDF	이미지: JPEG/JPG, PNG, BMP, TIFF, HEIF	Microsoft Office: Word(DOCX), Excel(XLSX), PowerPoint(PPTX) 및 HTML
읽음	✓	✓	✓
레이아웃	✓	✓	✓ (2024-02-29-preview, 2023-10-31-preview)
일반 문서	✓	✓	
사전 제작	✓	✓	
사용자 지정 추출	✓	✓	
사용자 지정 분류	✓	✓	✓ (2024-02-29-preview)

- PDF 및 TIFF의 경우 최대 2000페이지를 처리할 수 있습니다(무료 계층 구독의 경우 처음 2페이지만 처리됨).
- 문서 분석을 위한 파일 크기는 유료(S0) 계층의 경우 500MB, 무료(F0) 계층의 경우 4MB입니다.
- 이미지 크기는 50 x 50픽셀에서 10,000 x 10,000픽셀 사이여야 합니다.
- PDF가 암호로 잠긴 경우에는 제출하기 전에 잠금을 해제해야 합니다.
- 추출할 텍스트의 최소 높이는 1024 x 768 픽셀 이미지의 경우 12픽셀입니다. 이 차원은 150DPI(인치당 도트 수)에서 약 8포인트 텍스트에 해당합니다.
- 사용자 지정 모델 학습의 경우 학습 데이터의 최대 페이지 수는 사용자 지정 템플릿 모델의 경우 500개, 사용자 지정 인공신경망 모델의 경우 50,000개입니다.
 - 사용자 지정 추출 모델 학습의 경우 학습 데이터의 총 크기는 템플릿 모델의 경우 50MB, 신경망 모델의 경우 1G-MB입니다.
 - 사용자 지정 분류 모델 학습의 경우 학습 데이터의 총 크기는 최대 10,000페이지인 1GB입니다.

결혼 인증서 문서 데이터 추출 시도

결혼 인증서 카드 서비스에서 데이터 추출이 작동하는 방식을 확인하려면 다음 리소스가 필요합니다.

- Azure 구독은 무료로 만들 수 있습니다.
- Azure Portal의 Document Intelligence 인스턴스입니다. 무료 가격 책정 계층(F0)을 사용하여 서비스를 시도할 수 있습니다. 리소스가 배포된 후 리소스로 이동을 선택하여 키 및 엔드포인트를 가져옵니다.

Home > Contoso-DI

Contoso-DI | Keys and Endpoint

Search | Regenerate Key1 | Regenerate Key2

- Overview
- Activity log
- Access control (IAM)
- Tags
- Diagnose and solve problems
- Resource Management
 - Keys and Endpoint**
 - Encryption
 - Pricing tier
 - Networking
 - Identity
 - Cost analysis
 - Properties
 - Locks
- Monitoring
- Automation
- Help

These keys are used to access your Azure AI service API. Do not share your keys. Store them securely—for example, using Azure Key Vault. We also recommend regenerating these keys regularly. Only one key is necessary to make an API call. When regenerating the first key, you can use the second key for continued access to the service.

Show Keys

KEY 1	[REDACTED]	Copy
KEY 2	[REDACTED]	Copy
Location/Region	westus2	Copy
Endpoint	https://contoso-di.cognitiveservices.azure.com/	Copy

Document Intelligence Studio

1. 문서 인텔리전스 스튜디오 홈페이지에서 결혼 인증서를 선택합니다.
2. 샘플 결혼 인증서를 분석하거나 자체 파일을 업로드할 수 있습니다.
3. **분석 실행** 버튼을 선택하고 필요한 경우 **분석 옵션**을 구성합니다.



[Document Intelligence Studio 사용해 보기](#)

지원되는 언어 및 로캘

지원되는 언어의 전체 목록은 [언어 지원 - 미리 빌드된 모델 페이지](#)를 참조하세요.

필드 추출

다음은 JSON 출력 응답의 결혼 인증서에서 추출된 필드입니다.

[+] 테이블 확장

속성	형식	설명	예제 출력
Spouse1FirstName	문자열	배우자 1의 이름	Wesley
Spouse1MiddleName	문자열	배우자 1의 중간 이름	13.
Spouse1LastName	문자열	배우자 1의 성	Perry
Spouse1Age	정수	배우자 1의 나이	26
Spouse1BirthDate	날짜	배우자 1의 생년 월일	1997년 11월 16일
Spouse1Address	주소	배우자 1의 주소	4292 Don Jackson Lane, Bloomfield Township, Michigan 48302
Spouse1BirthPlace	문자열	배우자 1의 출생 지	미시간
Spouse2FirstName	문자열	배우자 2의 이름	Beth
Spouse2MiddleName	문자열	배우자 2의 중간 이름	18.
Spouse2LastName	문자열	배우자 2의 성	Mason
Spouse2Age	정수	배우자 2의 나이	23
Spouse2BirthDate	날짜	배우자 2의 생년 월일	2000년 7월 22일
Spouse2Address	주소	배우자 2의 주소	2671 Comfort Court, Madison, Wisconsin 53704
Spouse2BirthPlace	문자열	배우자 2의 출생 지	위스콘신
DocumentNumber	문자열	문서 번호	01976/202

속성	형식	설명	예제 출력
IssueDate	날짜	인증서 발급 날짜	2023년 10월 10일
IssuePlace	문자열	인증서 발급지	2398 Echo Lane, Hastings, Michigan 49058
MarriageDate	날짜	결혼 날짜	2023년 10월 10일
MarriagePlace	문자열	결혼 장소	105 Coal Street, Galloway, Wisconsin 54432

추출된 결혼 인증서 키-값 쌍 및 항목은 JSON 출력의 `documentResults` 섹션에 있습니다.

다음 단계

- 문서 인텔리전스 스튜디오 [를 사용하여 사용자 고유의 양식 및 문서를 처리해 보세요.](#)
- [Document Intelligence 빠른 시작](#)을 완료하고 원하는 개발 언어로 문서 처리 앱 만들기를 시작해 보세요.

Document Intelligence 계약 모델

아티클 • 2024. 04. 11.

① 중요

- 문서 인텔리전스 공개 미리 보기 릴리스에서는 현재 개발 중인 기능에 대한 조기 액세스를 제공합니다.
- 기능, 방식 및 프로세스는 GA(일반 공급) 전에 사용자 피드백에 따라 변경될 수 있습니다.
- 문서 인텔리전스 클라이언트 라이브러리의 공개 미리 보기 버전은 기본적으로 REST API 버전 [2024-02-29-미리 보기](#)로 설정됩니다.
- 공개 미리 보기 버전 [2024-02-29-미리 보기](#)는 현재 다음 Azure 지역에서만 사용할 수 있습니다.
 - 미국 동부
 - 미국 서부2
 - 서유럽

이 콘텐츠의 적용 대상은 다음과 같습니다. v4.0(미리 보기) | 이전 버전: v3.1(GA)

Document Intelligence 계약 모델은 강력한 OCR(광학 인식) 기능을 사용하여 중요한 계약 엔터티의 선택된 그룹에서 키 필드와 품목을 분석하고 추출합니다. 계약은 휴대폰으로 캡처한 이미지, 스캔한 문서, 디지털 PDF 등의 다양한 형식과 품질이 가능합니다. API는 문서 텍스트를 분석합니다. 당사자, 관할권, 계약 ID, 제목과 같은 주요 정보를 추출합니다. 구조화된 JSON 데이터 표현을 반환합니다. 이 모델은 현재 영어 문서 형식을 지원합니다.

자동화된 계약 처리

자동화된 계약 처리는 문서에서 주요 계약 필드를 추출하는 프로세스입니다. 지금까지 계약 분석 프로세스는 수동으로 수행되었으므로 시간이 많이 걸립니다. 계약에서 키 데이터를 정확하게 추출하는 것은 일반적으로 계약 자동화 프로세스에서 가장 중요한 첫 번째 단계 중 하나입니다.

개발 옵션

문서 인텔리전스 v4.0(2024-02-29-미리 보기)은 다음 도구, 애플리케이션, 라이브러리를 지원합니다.

기능	리소스	Model ID
계약 모델	<ul style="list-style-type: none">문서 인텔리전스 스튜디오 ↗REST APIC# SDKPython SDKJava SDKJavaScript SDK	prebuilt-contract

입력 요구 사항

- 최상의 결과를 위해 문서당 하나의 명확한 사진 또는 고품질 스캔을 제공합니다.
- 지원 파일 형식:

모델	PDF	이미지: JPEG/JPG, PNG, BMP, TIFF, HEIF	Microsoft Office: Word(DOCX), Excel(XLSX), PowerPoint(PPTX) 및 HTML
읽음	✓	✓	✓
레이아웃	✓	✓	✓ (2024-02-29-미리 보기, 2023-10-31-미리 보기)
일반 문서	✓	✓	
사전 제작	✓	✓	
사용자 지정 추출	✓	✓	
사용자 지정 분류	✓	✓	✓ (2024-02-29-미리 보기)

- PDF 및 TIFF의 경우 최대 2000페이지를 처리할 수 있습니다(무료 계층 구독의 경우 처음 2페이지만 처리됨).
- 문서 분석을 위한 파일 크기는 유료(S0) 계층의 경우 500MB, 무료(F0) 계층의 경우 4MB입니다.
- 이미지 크기는 50 x 50픽셀에서 10,000 x 10,000픽셀 사이여야 합니다.
- PDF가 암호로 잠긴 경우에는 제출하기 전에 잠금을 해제해야 합니다.

- 추출할 텍스트의 최소 높이는 1024 x 768 픽셀 이미지의 경우 12픽셀입니다. 이 차원은 150DPI(인치당 도트 수)에서 약 8포인트 텍스트에 해당합니다.
- 사용자 지정 모델 학습의 경우 학습 데이터의 최대 페이지 수는 사용자 지정 템플릿 모델의 경우 500개, 사용자 지정 인공신경망 모델의 경우 50,000개입니다.
 - 사용자 지정 추출 모델 학습의 경우 학습 데이터의 총 크기는 템플릿 모델의 경우 50MB, 신경망 모델의 경우 1G-MB입니다.
 - 사용자 지정 분류 모델 학습의 경우 학습 데이터의 총 크기는 최대 10,000페이지인 1GB입니다.

계약 문서 데이터 추출 시도

고객 정보, 공급업체 세부 정보, 품목을 포함한 데이터가 계약에서 추출되는 방법을 확인합니다. 다음 리소스가 필요합니다.

- Azure 구독은 [무료로 만들 수](#) 있습니다.
- Azure Portal의 [Document Intelligence 인스턴스](#)입니다. 무료 가격 책정 계층(F0)을 사용하여 서비스를 시도할 수 있습니다. 리소스가 배포된 후 [리소스로 이동](#)을 선택하여 키 및 엔드포인트를 가져옵니다.

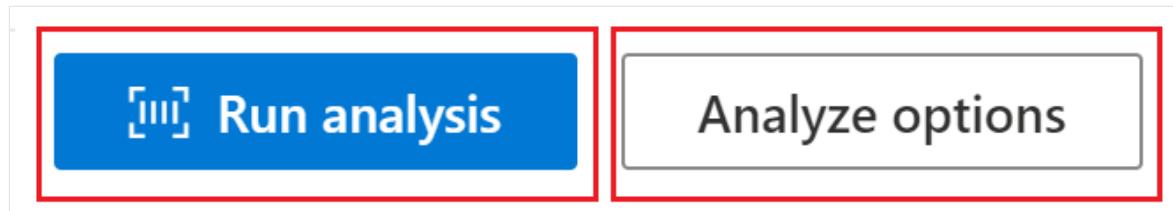
The screenshot shows the Azure portal interface for managing a Document Intelligence instance. The main title is "Contoso-DI | Keys and Endpoint". On the left, there's a sidebar with various service management options like Overview, Activity log, Access control (IAM), Tags, and Resource Management. Under Resource Management, the "Keys and Endpoint" option is selected and highlighted with a red box. The main content area contains instructions about keys, a "Show Keys" button, and input fields for KEY 1, KEY 2, Location/Region (set to westus2), and Endpoint (set to https://contoso-di.cognitiveservices.azure.com/). The Endpoint field is also highlighted with a red box.

Document Intelligence Studio

1. 문서 인텔리전스 스튜디오 홈페이지에서 **세금 문서**를 선택합니다.

2. 샘플 세금 문서를 분석하거나 자체 파일을 업로드할 수 있습니다.

3. **분석 실행** 버튼을 선택하고 필요한 경우 **분석 옵션**을 구성합니다.



지원되는 언어 및 로캘

지원되는 언어의 전체 목록은 [언어 지원 - 미리 빌드된 모델 페이지](#)를 참조하세요.

필드 추출

다음은 JSON 출력 응답의 계약에서 추출된 필드입니다.

[+] 테이블 확장

이름	형식	설명	예제 출력
타이틀	문자 열	계약 제목	서비스 계약
ContractId	문자 열	계약 제목	AB12956
파티	배열	법적 당사자 목록	
ExecutionDate	날짜	모든 당사자가 계약에 모두 서명하고 합의한 날짜	On this twenty-third day of February two thousand and twenty two
ExpirationDate	날짜	계약 효력이 종료되는 날짜	1년
RenewalDate	날짜	계약을 갱신해야 하는 날짜	On this twenty-third day of February two thousand and twenty two
관할권	배열	관할권 목록	

추출된 계약 키-값 쌍 및 품목은 JSON 출력의 `documentResults` 섹션에 있습니다.

다음 단계

- 문서 인텔리전스 스튜디오 [를 사용하여 사용자 고유의 양식 및 문서를 처리해 보세요.](#)
- Document Intelligence 빠른 시작을 완료하고 원하는 개발 언어로 문서 처리 앱 만들기를 시작해 보세요.

문서 인텔리전스 미국 세금 문서 모델

아티클 • 2024. 04. 15.

ⓘ 중요

- 문서 인텔리전스 공개 미리 보기 릴리스에서는 현재 개발 중인 기능에 대한 조기 액세스를 제공합니다.
- 기능, 방식 및 프로세스는 GA(일반 공급) 전에 사용자 피드백에 따라 변경될 수 있습니다.
- 문서 인텔리전스 클라이언트 라이브러리의 공개 미리 보기 버전은 기본적으로 REST API 버전 [2024-02-29-미리 보기](#)로 설정됩니다.
- 공개 미리 보기 버전 [2024-02-29-미리 보기](#)는 현재 다음 Azure 지역에서만 사용할 수 있습니다.
- 미국 동부**
- 미국 서부2**
- 서유럽**

이 콘텐츠 적용 대상: v4.0(미리 보기) | 이전 버전: v3.1(GA)

문서 인텔리전스 계약 모델은 강력한 광학 문자 인식(OCR) 기능을 사용하여 선택된 세금 문서 그룹에서 주요 필드와 항목을 분석하고 추출합니다. 세금 문서는 전화로 캡처한 이미지, 스캔한 문서, 디지털 PDF 등 다양한 형식과 품질을 가질 수 있습니다. API는 문서 텍스트를 분석합니다. 고객 이름, 청구서 수신 주소, 만기일, 결제 금액 등 주요 정보를 추출합니다. 구조화된 JSON 데이터 표현을 반환합니다. 이 모델은 현재 특정 영어 세금 문서 형식을 지원합니다.

지원되는 문서 유형:

- W-2
- 1098
- 1098-E
- 1098-T
- 1099 및 변형(A, B, C, CAP, DIV, G, H, INT, K, LS, LTC, MISC, NEC, OID, PATR, Q, QA, R, S, SA, SB)
- 1040 및 변형 (일정 1, 일정 2, 일정 3, 일정 8812, 일정 A, 일정 B, 일정 C, 일정 D, 일정 E, 일정 EIC, 일정 F, 일정 H, 일정 J, 일정 R, 일정 SE 및 일정 선임)

자동화된 세금 문서 처리

자동화된 세금 문서 처리는 세금 문서에서 주요 필드를 추출하는 프로세스입니다. 과거에는 세금 문서가 수동으로 처리되었습니다. 이 모델을 사용하면 세금 시나리오를 쉽게 자동화할 수 있습니다.

개발 옵션

문서 인텔리전스 v4.0(2023-10-31-preview)은 다음 도구, 애플리케이션, 라이브러리를 지원합니다.

기능	리소스	Model ID
미국 세금 양식 모델	<ul style="list-style-type: none"> 문서 인텔리전스 스튜디오 ↗ REST API C# SDK Python SDK Java SDK JavaScript SDK 	<ul style="list-style-type: none"> prebuilt-tax.us.W-2 prebuilt-tax.us.1098 prebuilt-tax.us.1098E prebuilt-tax.us.1098T prebuilt-tax.us.1099A prebuilt-tax.us.1099B prebuilt-tax.us.1099C prebuilt-tax.us.1099CAP prebuilt-tax.us.1099DIV prebuilt-tax.us.1099G prebuilt-tax.us.1099H prebuilt-tax.us.1099INT prebuilt-tax.us.1099K prebuilt-tax.us.1099LS prebuilt-tax.us.1099LTC prebuilt-tax.us.1099MISC prebuilt-tax.us.1099NEC prebuilt-tax.us.1099OID prebuilt-tax.us.1099PATR prebuilt-tax.us.1099Q prebuilt-tax.us.1099QA prebuilt-tax.us.1099R prebuilt-tax.us.1099S prebuilt-tax.us.1099SA prebuilt-tax.us.1099SB prebuilt-tax.us.1040 prebuilt-tax.us.1040Schedule1 prebuilt-tax.us.1040Schedule2 prebuilt-tax.us.1040Schedule3 prebuilt-tax.us.1040Schedule8812 prebuilt-tax.us.1040ScheduleA prebuilt-tax.us.1040ScheduleB prebuilt-tax.us.1040ScheduleC prebuilt-tax.us.1040ScheduleD prebuilt-tax.us.1040ScheduleE prebuilt-tax.us.1040ScheduleEIC prebuilt-tax.us.1040ScheduleF prebuilt-tax.us.1040ScheduleH prebuilt-tax.us.1040ScheduleJ prebuilt-tax.us.1040ScheduleR prebuilt-tax.us.1040ScheduleSE prebuilt-tax.us.1040Senior

입력 요구 사항

- 최상의 결과를 위해 문서당 하나의 명확한 사진 또는 고품질 스캔을 제공합니다.
- 지원 파일 형식:

모델	PDF	이미지:		Microsoft Office: Word(DOCX), Excel(XLSX), PowerPoint(PPTX) 및 HTML
		JPEG/JPG, PNG, BMP, TIFF, HEIF		
읽음	✓	✓		✓
레이아웃	✓	✓		✓ (2024-02-29-미리 보기, 2023-10-31-미리 보기)
일반 문서	✓	✓		
사전 제작	✓	✓		
사용자 지정 추출	✓	✓		
사용자 지정 분류	✓	✓		✓ (2024-02-29-미리 보기)

- PDF 및 TIFF의 경우 최대 2000페이지를 처리할 수 있습니다(무료 계층 구독의 경우 처음 2페이지만 처리됨).
- 문서 분석을 위한 파일 크기는 유료(S0) 계층의 경우 500MB, 무료(F0) 계층의 경우 4MB입니다.
- 이미지 크기는 50 x 50픽셀에서 10,000 x 10,000픽셀 사이여야 합니다.
- PDF가 암호로 잠긴 경우에는 제출하기 전에 잠금을 해제해야 합니다.
- 추출할 텍스트의 최소 높이는 1024 x 768 픽셀 이미지의 경우 12픽셀입니다. 이 차원은 150DPI(인치당 도트 수)에서 약 8포인트 텍스트에 해당합니다.
- 사용자 지정 모델 학습의 경우 학습 데이터의 최대 페이지 수는 사용자 지정 템플릿 모델의 경우 500개, 사용자 지정 인공신경망 모델의 경우 50,000개입니다.
 - 사용자 지정 추출 모델 학습의 경우 학습 데이터의 총 크기는 템플릿 모델의 경우 50MB, 신경망 모델의 경우 1G-MB입니다.
 - 사용자 지정 분류 모델 학습의 경우 학습 데이터의 총 크기는 최대 10,000페이지인 1GB입니다.

세금 문서 데이터 추출을 사용해 보세요

고객 정보, 공급업체 세부 정보 및 라인 항목을 포함한 데이터가 청구서에서 추출되는 방법을 확인합니다. 다음 리소스가 필요합니다.

- Azure 구독은 [무료로 만들 수](#) 있습니다.
- Azure Portal의 [Document Intelligence 인스턴스](#)입니다. 무료 가격 책정 계층(F0)을 사용하여 서비스를 시도할 수 있습니다. 리소스가 배포된 후 [리소스로 이동](#)을 선택하여 키 및 엔드포인트를 가져옵니다.

Home > Contoso-DI

Contoso-DI | Keys and Endpoint

Search Regenerate Key1 Regenerate Key2

Overview Activity log Access control (IAM) Tags Diagnose and solve problems Resource Management Keys and Endpoint (highlighted)

Encryption Pricing tier Networking Identity Cost analysis Properties Locks Monitoring Automation Help

KEY 1
***** (red box)
KEY 2
***** (red box)
Location/Region (westus2)
Endpoint https://contoso-di.cognitiveservices.azure.com/ (red box)

문서 인텔리전스 스튜디오

1. Document Intelligence Studio 홈페이지 [홈페이지](#)에서 지원되는 세금 문서 모델을 선택합니다.
2. 샘플 세금 문서를 분석하거나 자체 파일을 업로드할 수 있습니다.
3. 분석 실행 버튼을 선택하고 필요한 경우 분석 옵션을 구성합니다.



[Document Intelligence Studio 사용해 보기](#)

지원되는 언어 및 로캘

지원되는 언어의 전체 목록은 [언어 지원 - 미리 빌드된 모델](#) 페이지를 참조하세요.

필드 추출 W-2

다음은 JSON 출력 응답의 W-2 세금 양식에서 추출된 필드입니다.

테이블 확장

이름	형식	설명	예제 출력	부양 가족
<code>W-2FormVariant</code>	문자열	IR W-2 형태 변형. 이 필드는 다음 값 중 하나를 가질 수 있습니다: <code>W-2</code> , <code>W-2AS</code> , <code>W-2CM</code> , <code>W-2GU</code> 또는 <code>W-2VI</code>	W-2	
<code>TaxYear</code>	숫자	양식 과세연도	2021	
<code>W2Copy</code>	문자열	이 사본과 관련된 인쇄된 지침과 함께 W-2 세금 사본 버전		사본 A—사회보장국용
<code>Employee</code>	개체	사회 보장 번호, 이름 및 주소를 포함하는 개체		
<code>ControlNumber</code>	string	W-2 제어 번호. IRS W-2 필드 d	0AB12 D345 7890	
<code>Employer</code>	Object	고용주 ID 번호, 이름 및 주소를 포함하는 개체		
<code>WagesTipsAndOtherCompensation</code>	숫자	임금, 팁, 기타 보상 금액은 USD로 표시됩니다. IRS W-2 필드 1	1234567.89	
<code>FederalIncomeTaxWithheld</code>	숫자	연방 소득세 원천징수 금액(USD)입니다. IRS W-2 필드 2	1234567.89	
<code>SocialSecurityWages</code>	숫자	사회 보장 임금 금액(USD). IRS W-2 필드 3	1234567.89	
<code>SocialSecurityTaxWithheld</code>	숫자	사회 보장세 원천징수 금액(USD). IRS W-2 필드 4	1234567.89	
<code>MedicareWagesAndTips</code>	숫자	메디케어 임금과 팁 금액(USD) IRS W-2 필드 5	1234567.89	
<code>MedicareTaxWithheld</code>	숫자	메디케어 세금 원천징수 금액(USD). IRS W-2 필드 6	1234567.89	
<code>SocialSecurityTips</code>	숫자	사회보장 팁 금액(USD). IRS W-2 필드 7	1234567.89	
<code>AllocatedTips</code>	숫자	할당된 팁(USD). IRS W-2 필드 8	1234567.89	
<code>VerificationCode</code>	숫자	W-2 인증 코드. IRS W-2 필드 9	1234567.89	
<code>DependentCareBenefits</code>	숫자	부양 가족 돌봄 혜택 금액(USD). IRS W-2 필드 10	1234567.89	
<code>NonQualifiedPlans</code>	숫자	비적격 계획 금액(USD). IRS W-2 필드 11	1234567.89	
<code>IsStatutoryEmployee</code>	문자열	IRS W-2 필드 13의 일부입니다. true 또는 false일 수 있습니다.	true	

이름	형식	설명	예제 출력	부 양 가 족
IsRetirementPlan	문자 열	IRS W-2 필드 13의 일부입니다. true 또는 false일 수 있습니다.	true	
IsThirdPartySickPay	문자 열	IRS W-2 필드 13의 일부입니다. true 또는 false일 수 있습니다.	true	
Other	문자 열	IRS W-2 필드 14의 내용	SICK LV는 SBJT의 임금을 \$511/일 한도 1356으로 책정합니다.	
StateTaxInfos	배열	주세 관련 정보입니다. IRS W-2 필드 15~17의 내용		
LocaleTaxInfos	배열	지방세 관련 정보입니다. IRS W-2 필드 18~20의 내용		

필드 추출 1098

다음은 JSON 출력 응답의 1098 세금 양식에서 추출된 필드입니다. 1098-T 및 1098-E 양식도 지원됩니다.

[\[+\] 테이블 확장](#)

이름	형식	설명	예제 출력
TaxYear	숫자	양식 과세연도	2021
빌어 쓰는 사람	Object	차입자의 TIN, 이름, 주소 및 계좌 번호를 포함하는 개체	
대주	Object	대출 기관의 TIN, 이름, 주소 및 전화번호 가 포함된 개체	
MortgageInterest	숫자	지불자/대출자로부터 받은 모기지 이자 금액(상자 1)	1,234,567.89
OutstandingMortgagePrincipal	숫자	미납 모기지 원금(상자 2)	1,234,567.89
MortgageOriginationDate	날짜	모기지 발생일(박스 3)	2022-01-01
OverpaidInterestRefund	숫자	초과 지급된 이자 환불 금액(박스 4)	1,234,567.89
MortgageInsurancePremium	숫자	모기지 보험료 금액(박스 5)	1,234,567.89
PointsPaid	숫자	주 거주지 구입 시 지급되는 포인트(박 스 6)	1,234,567.89
IsPropertyAddressSameAsBorrower	문자 열	모기지를 담보하는 부동산의 주소가 지 불인/대출자의 우편 주소(상자 7)와 동일 합니까?	true

이름	형식	설명	예제 출력
PropertyAddress	문자열 (박스 8)	모기지 담보 부동산의 주소 또는 설명	123 Main St., Redmond WA 98052
MortgagedPropertiesCount	숫자	저당 잡힌 부동산 수(박스 9)	1
기타	문자열	지불자에게 보고할 추가 정보(상자 10)	
RealEstateTax	숫자	부동산 세금(상자 1)	1,234,567.89
AdditionalAssessment	문자열	부동산에 대한 추가 평가(상자 10)	1,234,567.89
MortgageAcquisitionDate	날짜	모기지 취득 날짜(박스 11)	2022-01-01

필드 추출 1099-NEC

다음은 JSON 출력 응답의 1099-nec 세금 양식에서 추출된 필드입니다. 1099의 다른 변형도 지원됩니다.

[\[+\] 테이블 확장](#)

이름	형식	설명	예제 출력
TaxYear	문자열	양식 1099-NEC에서 추출된 과세 연도입니다.	2021
Payer	Object	납부자의 TIN, 이름, 주소, 전화번호가 포함된 개체	
Recipient	Object	수신자의 TIN, 이름, 주소, 계정번호가 포함된 개체	
Box1	번호	양식 1099-NEC에서 추출된 상자 1.	123456
Box2	부울 값	양식 1099-NEC에서 추출된 상자 2.	true
Box4	번호	양식 1099-NEC에서 추출된 상자 4.	123456
StateTaxesWithheld	배열	양식 1099-NEC에서 추출된 주세 원천징수(상자 5, 6, 7)	

필드 추출 1040 세금 양식

다음은 JSON 출력 응답의 1040 세금 양식에서 추출된 필드입니다. 1040의 다른 변형도 지원됩니다.

[\[+\] 테이블 확장](#)

속성	형식	설명	예제 출력
TaxPayer	Object	납세자의 정보(예: SSN, 성 및 주소)가 들어 있는 개체입니다.	
Spouse	Object	SSN, 성, 이름 및 이니셜 이름과 같은 배우자	

속성	형식	설명	예제 출력	
		의 정보를 포함하는 개체입니다.		
Dependents	배열	이름, SSN 및 크레딧 유형과 같은 정보를 포함하는 종속 항목 목록이 포함된 배열입니다.		
ThirdPartyDesignee	개체	타사 디자인 담당자에 대한 정보가 들어 있는 개체입니다.		
SignatureDetails	개체	전화 번호 및 전자 메일과 같은 서명자 정보를 포함하는 개체입니다.		
PaidPreparer	개체	준비 관리자에 대한 정보가 들어 있는 개체입니다.		
FilingStatus	문자 열	값은 noSelection, single, marriedFilingJointly, marriedFilingSeparately, headOfHousehold, qualifyingSurvivingSpouse 또는 multiSelection 중 하나일 수 있습니다.	single	
FilingStatusDetails	개체	제출 상태에 대한 정보가 들어 있는 개체입니다.		
NameOfSpouseOrQualifyingPerson	문자 열	양식 1040에서 추출된 배우자 또는 적격자의 이름입니다.	John Smith	
PresidentialElectionCampaign	문자 열	값은 noSelection, 납세자, 배우자 또는 multiSelection 중 하나일 수 있습니다.	납세자	
PresidentialElectionCampaignDetails	개체	대통령 선거 운동에 대한 세부 정보가 들어 있는 개체입니다.		
DigitalAssets	문자 열	값은 noSelection, 예, 아니요 또는 multiSelection 중 하나일 수 있습니다.	예	
DigitalAssetsDetails	개체	디지털 자산에 대한 세부 정보가 들어 있는 개체입니다.		
ClaimStatus	문자 열	값은 noSelection, taxpayerAsDependent, spouseAsDependent, spouseItemizesSeparatelyOrDualStatusAlien 또는 multiSelection 중 하나일 수 있습니다.	taxpayerAsDependent	
ClaimStatusDetails	개체	클레임 상태에 대한 세부 정보가 들어 있는 개체입니다.		
TaxpayerAgeBlindness	문자 열	값은 noSelection, above64, blind 또는 multiSelection 중 하나일 수 있습니다.	above64	
TaxPayerAgeBlindnessDetails	개체	납세자 연령 실명에 대한 세부 정보를 포함하는 개체입니다.		
SpouseAgeBlindness	문자 열	값은 noSelection, above64, blind 또는 multiSelection 중 하나일 수 있습니다.	above64	

속성	형식	설명	예제 출력
TaxPayerAgeBlindnessDetails	개체	배우자 연령 실명에 대한 세부 정보를 포함하는 개체입니다.	
MoreThanFourDependents	부울 값	양식 1040에서 추출된 4개 이상의 종속성입니다.	true
Box1a	번호	Box 1a 1040에서 추출되었습니다.	123456
제공된 JSON 구조에 따라 요청된 것과 동일한 테이블 형식으로 변환하면 결과는 다음과 같습니다.			
Box1b	번호	Box 1b 1040에서 추출되었습니다.	123456
Box1c	번호	Box 1c 1040에서 추출되었습니다.	123456
Box1d	번호	Box 1d 1040에서 추출되었습니다.	123456
Box1e	번호	Box 1e 1040에서 추출되었습니다.	123456
Box1f	번호	Box 1f 1040에서 추출되었습니다.	123456
Box1g	번호	Box 1g 1040에서 추출되었습니다.	123456
Box1h	번호	Box 1h 1040에서 추출되었습니다.	123456
Box1i	번호	Box 1i 1040에서 추출되었습니다.	123456
Box1z	번호	Box 1z 1040에서 추출되었습니다.	123456
Box2a	번호	Box 2a 1040에서 추출되었습니다.	123456
Box2b	번호	Box 2b 1040에서 추출되었습니다.	123456
Box3a	번호	Box 3a 1040에서 추출되었습니다.	123456
Box3b	번호	Box 3b 1040에서 추출되었습니다.	123456
Box4a	번호	Box 4a 1040에서 추출되었습니다.	123456
Box4b	번호	Box 4b 1040에서 추출되었습니다.	123456
Box5a	번호	Box 5a 1040에서 추출되었습니다.	123456
Box5b	번호	Box 5b 1040에서 추출되었습니다.	123456
Box6a	번호	Box 6a 1040에서 추출되었습니다.	123456
Box6b	번호	Box 6b 1040에서 추출되었습니다.	123456
Box6cCheckbox	부울 값	Box 6c 1040에서 추출된 확인 상자입니다.	true
Box7Checkbox	부울 값	상자 7 1040에서 추출된 확인 상자입니다.	true

속성	형식	설명	예제 출력
Box7	번호	상자 7 1040에서 추출되었습니다.	123456
Box8	번호	상자 8 1040에서 추출되었습니다.	123456
Box9	번호	상자 9 1040에서 추출되었습니다.	123456
Box10	번호	상자 10 1040에서 추출되었습니다.	123456
Box11	번호	상자 11 1040에서 추출되었습니다.	123456
Box12	번호	상자 12 1040에서 추출되었습니다.	123456
Box13	번호	상자 13 1040에서 추출되었습니다.	123456
Box14	번호	상자 14 1040에서 추출되었습니다.	123456
Box15	번호	상자 15 1040에서 추출되었습니다.	123456
Box16FromForm	string	값은 noSelection, 8814, 4972, 기타 또는 multiSelection 중 하나일 수 있습니다.	8814
Box16FromFormDetails	개체	Box 16에 대한 세부 정보가 포함된 개체	
Box16OtherFormNumber	string	Box 16 1040에서 추출된 기타 양식 번호입니다.	8888
Box16	번호	상자 16 1040에서 추출되었습니다.	123456
Box17	번호	상자 17 1040에서 추출되었습니다.	123456
Box18	번호	상자 18 1040에서 추출되었습니다.	123456
Box19	번호	상자 19 1040에서 추출되었습니다.	123456
Box20	번호	상자 20 1040에서 추출되었습니다.	123456
Box21	번호	상자 21 1040에서 추출되었습니다.	123456
Box22	번호	상자 22 1040에서 추출되었습니다.	123456
Box23	번호	상자 23 1040에서 추출되었습니다.	123456
Box24	번호	상자 24 1040에서 추출되었습니다.	123456
Box25a	번호	Box 25a 1040에서 추출되었습니다.	123456
Box25b	번호	Box 25b 1040에서 추출되었습니다.	123456
Box25c	번호	Box 25c 1040에서 추출되었습니다.	123456
Box25d	번호	Box 25d 1040에서 추출되었습니다.	123456
Box26	번호	상자 26 1040에서 추출되었습니다.	123456
Box27	번호	상자 27 1040에서 추출되었습니다.	123456
Box28	번호	상자 28 1040에서 추출되었습니다.	123456

속성	형식	설명	예제 출력
Box29	번호	상자 29 1040에서 추출되었습니다.	123456
Box31	번호	상자 31 1040에서 추출되었습니다.	123456
Box32	번호	상자 32 1040에서 추출되었습니다.	123456
Box33	번호	상자 33 1040에서 추출되었습니다.	123456
Box34	번호	상자 34 1040에서 추출되었습니다.	123456
Box35Checkbox	부울 값	상자 35 1040에서 추출된 확인 상자입니다.	true
Box35a	번호	Box 35a 1040에서 추출되었습니다.	123456
Box35b	번호	Box 35b 1040에서 추출되었습니다.	123456
Box35c	string	값은 noSelection, checking, savings 또는 multiSelection 중 하나일 수 있습니다.	검사
Box35cDetails	개체	Box 35c에 대한 세부 정보가 포함된 개체	
Box35d	번호	Box 35d 1040에서 추출되었습니다.	123456
Box36	번호	상자 36 1040에서 추출되었습니다.	123456
Box37	번호	상자 37 1040에서 추출되었습니다.	123456
Box38	번호	상자 38 1040에서 추출되었습니다.	123456
HasAssignedThirdPartyDesignee	string	값은 noSelection, 예, 아니요 또는 multiSelection 중 하나일 수 있습니다.	예
HasAssignedThirdPartyDesigneeDetails	개체	할당된 타사 디자인 담당자에 대해 선택된 항목에 대한 정보를 포함하는 개체	

추출된 세금 문서 키-값 쌍 및 품목은 JSON 출력의 `documentResults` 섹션에 있습니다.

다음 단계

- 문서 인텔리전스 스튜디오 [를](#) 사용하여 사용자 고유의 양식 및 문서를 처리해 보세요.
- Document Intelligence 빠른 시작을 완료하고 원하는 개발 언어로 문서 처리 앱 만들기를 시작해 보세요.

Document Intelligence 건강 보험 카드 모델

아티클 • 2024. 04. 11.

① 중요

- 문서 인텔리전스 공개 미리 보기 릴리스에서는 현재 개발 중인 기능에 대한 초기 액세스를 제공합니다.
- 기능, 방식 및 프로세스는 GA(일반 공급) 전에 사용자 피드백에 따라 변경될 수 있습니다.
- 문서 인텔리전스 클라이언트 라이브러리의 공개 미리 보기 버전은 기본적으로 REST API 버전 [2024-02-29-미리 보기](#)로 설정됩니다.
- 공개 미리 보기 버전 [2024-02-29-미리 보기](#)는 현재 다음 Azure 지역에서만 사용할 수 있습니다.
 - 미국 동부**
 - 미국 서부2**
 - 서유럽**

이 콘텐츠의 적용 대상은 다음과 같습니다. v4.0(미리 보기) | 이전 버전: v3.1(GA) v3.0(GA)

Document Intelligence 건강 보험 카드 모델은 강력한 OCR(광학 인식) 기능과 딥 러닝 모델을 결합하여 미국 건강 보험 카드에서 주요 정보를 분석하고 추출합니다. 의료 보험 카드는 건강 검진을 위한 핵심 문서이며 환자 교육, 금융 적용 정보, 현금없는 지불 및 보험 청구 처리를 위해 디지털로 분석할 수 있습니다. 의료 보험 카드 모델은 의료 카드 이미지를 분석하고, 보험사, 회원, 처방전 및 그룹 번호와 같은 주요 정보를 추출하고, 구조화된 JSON 표현을 반환합니다. 의료 보험 카드는 휴대폰으로 캡처한 이미지, 스캔한 문서, 디지털 PDF 등 형식과 품질로 제공될 수 있습니다.

Document Intelligence Studio를 사용하여 처리된 샘플 건강 보험 카드

The screenshot shows a medical insurance card from PREMERA Blue Cross. Key details include:

- Member:** ANGEL BROWN
- Group #:** 1000000
- Rx Group #:** BCAAXY7
- Rx BIN#:** 987654
- ABC:** 123456789
- Prefix:** Identification #
- Suffix:** 01
- Medical Network:** HERITAGE
- Premera Dental:** YES
- Premera Vision:** YES
- HEALTH SAVINGS PLAN:** Shared In and Out of Network
- Deductible:** \$1,500
- Coinsurance Max:** \$1,000
- Note:** Rx and Medical Cost Shares are Shared
- PPO:** PPO logo

The Analyze tool's results pane shows the following extracted fields:

Field	Value	Confidence (%)
GroupNumber	#1	80.00%
	1000000	
IdNumber	#1	87.00%
	Number	
	123456789	
Insurer	#1	87.00%
	PREMERA	
Member	#1	100.00%
	Name	
	ANGEL BROWN	
PrescriptionInfo	#1	80.00%
	RxBIN	

개발 옵션

문서 인텔리전스 v4.0(2024-02-29-미리 보기 2023-10-31-미리 보기)은 다음 도구, 애플리케이션 및 라이브러리를 지원합니다.

테이블 확장

기능	리소스	Model ID
의료 보험 카드 모델	<ul style="list-style-type: none"> 문서 인텔리전스 스튜디오 REST API C# SDK Python SDK Java SDK JavaScript SDK 	prebuilt-healthInsuranceCard.us

입력 요구 사항

- 최상의 결과를 위해 문서당 하나의 명확한 사진 또는 고품질 스캔을 제공합니다.
- 지원 파일 형식:

테이블 확장

모델	PDF	이미지: JPEG/JPG, PNG, BMP, TIFF, HEIF	Microsoft Office: Word(DOCX), Excel(XLSX), PowerPoint(PPTX) 및 HTML
읽음	✓	✓	✓

모델	PDF	이미지: JPEG/JPG, PNG, BMP, TIFF, HEIF	Microsoft Office: Word(DOCX), Excel(XLSX), PowerPoint(PPTX) 및 HTML
레이아웃	✓	✓	✓ (2024-02-29-미리 보기, 2023-10-31-미리 보기)
일반 문서	✓	✓	
사전 제작	✓	✓	
사용자 지정 추출	✓	✓	
사용자 지정 분류	✓	✓	✓ (2024-02-29-미리 보기)

- PDF 및 TIFF의 경우 최대 2000페이지를 처리할 수 있습니다(무료 계층 구독의 경우 처음 2페이지만 처리됨).
- 문서 분석을 위한 파일 크기는 유료(S0) 계층의 경우 500MB, 무료(F0) 계층의 경우 4MB입니다.
- 이미지 크기는 50 x 50픽셀에서 10,000 x 10,000픽셀 사이여야 합니다.
- PDF가 암호로 잠긴 경우에는 제출하기 전에 잠금을 해제해야 합니다.
- 추출할 텍스트의 최소 높이는 1024 x 768 픽셀 이미지의 경우 12픽셀입니다. 이 차원은 150DPI(인치당 도트 수)에서 약 8포인트 텍스트에 해당합니다.
- 사용자 지정 모델 학습의 경우 학습 데이터의 최대 페이지 수는 사용자 지정 템플릿 모델의 경우 500개, 사용자 지정 인공신경망 모델의 경우 50,000개입니다.
 - 사용자 지정 추출 모델 학습의 경우 학습 데이터의 총 크기는 템플릿 모델의 경우 50MB, 신경망 모델의 경우 1G-MB입니다.
 - 사용자 지정 분류 모델 학습의 경우 학습 데이터의 총 크기는 최대 10,000페이지인 1GB입니다.

Document Intelligence Studio 사용해 보기

Document Intelligence Studio를 사용하여 건강 보험 카드에서 데이터가 추출되는 방법을 참조하세요. 다음 리소스가 필요합니다.

- Azure 구독은 [무료 만들 수](#) 있습니다.

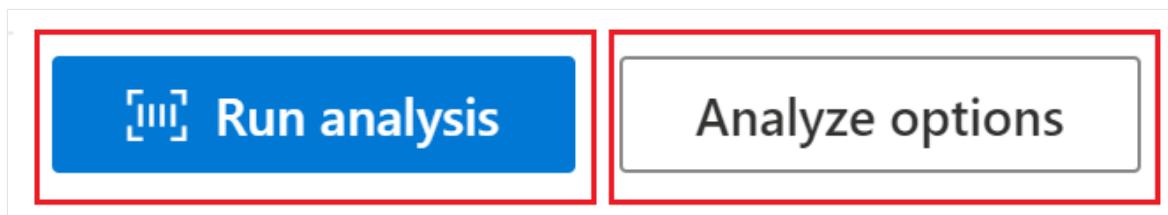
- Azure Portal의 Document Intelligence 인스턴스 [입니다](#). 무료 가격 책정 계층(F0)을 사용하여 서비스를 시도할 수 있습니다. 리소스가 배포된 후 리소스로 이동을 선택하여 키 및 엔드포인트를 가져옵니다.

The screenshot shows the Azure Portal interface for the 'Contoso-DI' resource. The left sidebar has a 'Keys and Endpoint' section highlighted with a red box. The main content area shows 'KEY 1' and 'Endpoint' fields, both of which are also highlighted with red boxes. A note at the top right says: 'These keys are used to access your Azure AI service API. Do not share your keys. Store them securely—for example, using Azure Key Vault. We also recommend regenerating these keys regularly. Only one key is necessary to make an API call. When regenerating the first key, you can use the second key for continued access to the service.'

① 참고

Document Intelligence Studio는 API 버전 v3.0과 함께 사용할 수 있습니다.

1. [Document Intelligence Studio 홈페이지](#)에서 건강 보험 카드를 선택합니다.
2. 샘플 의료 보험 카드 문서를 분석하거나 **+** 추가 단추를 선택하여 자체 샘플을 업로드합니다.
3. **분석 실행** 버튼을 선택하고 필요한 경우 **분석 옵션**을 구성합니다.



[문서 인텔리전스 스튜디오 사용해 보기](#)

지원되는 언어 및 로캘

지원되는 언어의 전체 목록은 [언어 지원 - 미리 빌드된 모델 페이지](#)를 참조하세요.

필드 추출

[+] 테이블 확장

필드	형식	Description	예시
Insurer	string	의료 보험 공급자 이름	PREMERA BLUE CROSS
Member	object		
Member.Name	string	멤버 이름	ANGEL BROWN
Member.BirthDate	date	회원 생년월일	01/06/1958
Member.Employer	string	회원 이름 고용주	Microsoft
Member.Gender	string	회원 성별	M
Member.IdNumberSuffix	string	일부 의료 보험 카드에 표시되는 식별 번호 접미사	01
Dependents	array	가능한 경우 멤버 자격 접미사 값별로 정렬되는 종속 항목 목록이 포함된 배열	
Dependents.*	object		
Dependents.*.Name	string	종속 항목 이름	01
IdNumber	object		
IdNumber.Prefix	string	일부 의료 보험 카드에 표시되는 식별 번호 접두사	ABC
IdNumber.Number	string	식별 번호	123456789
GroupNumber	string	보험 그룹 번호	1000000
PrescriptionInfo	object		
PrescriptionInfo.Issuer	string	ANSI IIN(발급자 식별 번호)	(80840) 300-11908-77

필드	형식	Description	예시
PrescriptionInfo.RxBIN	string	처방전 발급 BIN 번호	987654
PrescriptionInfo.RxPCN	string	처방전 프로세서 제어 번호	63200305
PrescriptionInfo.RxGrp	string	처방전 그룹 번호	BCAAXYZ
PrescriptionInfo.RxId	string	처방전 식별 번호입니다. 없는 경우 기본 값은 멤버 자격 ID 번호입니다.	P97020065
PrescriptionInfo.RxPlan	string	처방전 플랜 번호	A1
Pbm	string	플랜에 대한 Pharmacy Benefit Manager	CVS CAREMARK
EffectiveDate	date	플랜이 적용되는 날짜	08/12/2012
Copays	array	Co-Pay 혜택 목록이 포함된 배열	
Copays.*	object		
Copays.*.Benefit	string	Co-Pay 보조금 이름	공제 가능
Copays.*.Amount	currency	Co-Pay 필요 금액	\$1,500
Payer	object		
Payer.Id	string	지급인 ID 번호	89063
Payer.Address	address	지급인 주소	123 Service St., Redmond WA, 98052
Payer.PhoneNumber	phoneNumber	지급인 전화 번호	+1 (987) 213-5674
Plan	object		
Plan.Number	string	플랜 번호	456
Plan.Name	string	플랜 이름 - Medicaid 인 경우 - > medicaid(모두 소문자)	의료비 절감 플랜

필드	형식	Description	예시
Plan.Type	string	플랜 유형	PPO
MedicareMedicaidInfo	object		
MedicareMedicaidInfo.Id	string	Medicare 또는 Medicaid 번호	1AB2-CD3-EF45
MedicareMedicaidInfo.PartAEffectiveDate	date	Medicare Part A의 유효 날짜	01-01-2023
MedicareMedicaidInfo.PartBEffectiveDate	date	Medicare Part B의 유효 날짜	01-01-2023

마이그레이션 가이드 및 REST API v3.1

- 애플리케이션 및 워크플로에서 v3.1 버전을 사용하는 방법을 알아보려면 [Document Intelligence v3.1 마이그레이션 가이드](#)를 따르세요.
- v3.1 버전 및 새 기능에 대해 자세히 알아보려면 [REST API](#)를 살펴보세요.

다음 단계

- [문서 인텔리전스 스튜디오](#)를 사용하여 사용자 고유의 양식 및 문서를 처리해 보세요.
- [Document Intelligence 빠른 시작](#)을 완료하고 원하는 개발 언어로 문서 처리 앱 만들기를 시작해 보세요.

문서 인텔리전스 청구서 모델

아티클 • 2024. 04. 11.

① 중요

- 문서 인텔리전스 공개 미리 보기 릴리스에서는 현재 개발 중인 기능에 대한 조기 액세스를 제공합니다.
- 기능, 방식 및 프로세스는 GA(일반 공급) 전에 사용자 피드백에 따라 변경될 수 있습니다.
- 문서 인텔리전스 클라이언트 라이브러리의 공개 미리 보기 버전은 기본적으로 REST API 버전 [2024-02-29-미리 보기](#)로 설정됩니다.
- 공개 미리 보기 버전 [2024-02-29-미리 보기](#)는 현재 다음 Azure 지역에서만 사용할 수 있습니다.
 - 미국 동부
 - 미국 서부2
 - 서유럽

이 콘텐츠의 적용 대상은 다음과 같습니다. v4.0(미리 보기) | 이전 버전: v3.1(GA) v3.0(GA) v2.1(GA)

문서 인텔리전스 청구서 모델은 강력한 OCR(광학 인식) 기능을 사용하여 판매 청구서, 공과금 청구서 및 구매 주문서의 주요 필드와 항목을 분석하고 추출합니다. 청구서는 휴대폰으로 캡처한 이미지, 스캔한 문서, 디지털 PDF 등 형식과 품질이 다양할 수 있습니다. API는 청구서 텍스트를 분석하고, 고객 이름, 청구 주소, 만기일, 만기금액과 같은 주요 정보를 추출하고, 구조화된 JSON 데이터 표현을 반환합니다. 이 모델은 현재 27개 언어로 청구서를 지원합니다.

지원되는 문서 유형:

- 송장
- 공과금 청구서
- 판매 주문
- 구매 주문

자동화된 청구서 처리

자동화된 청구서 처리는 청구 계정 문서에서 주요 지급 계정 필드를 추출하는 프로세스입니다. 추출된 데이터에는 검토 및 결제를 위해 AP(지급 계정) 워크플로와 통합된 청구

서의 항목이 포함됩니다. 지금까지 지급 계정 프로세스는 수동으로 수행되어 시간이 많이 걸렸습니다. 청구서에서 핵심 데이터를 정확하게 추출하는 것은 일반적으로 청구서 자동화 프로세스에서 첫 번째이자 가장 중요한 단계 중 하나입니다.

[문서 인텔리전스 스튜디오](#)로 처리된 샘플 청구서:

The screenshot shows a Microsoft Power BI Analyze interface. On the left, there is a preview of an 'INVOICE' document from 'CONTOSO LTD.' containing various fields like Customer Name, Service Period, and Item Details. On the right, a 'Values' pane displays the extracted data with confidence scores. The results include:

- AmountDue #1: 97.30% (610)
- BillingAddress #1: 94.70% (123 Bill St, Redmond WA, 98052)
- BillingAddressRecipient #1: 95.70% (Microsoft Finance)
- CustomerAddress #1: 94.70% (123 Other St, Redmond WA, 98052)
- CustomerAddressRecipient #1: 95.60% (Microsoft Corp)
- CustomerId #1: 96.40% (CID-12345)
- CustomerName #1: 94.90% (MICROSOFT CORPORATION)
- DueDate #1: 97.30% (2019-12-15)
- InvoiceDate #1: 97.20% (2019-11-15)

개발 옵션

문서 인텔리전스 v4.0(2024-02-29-미리 보기 2023-10-31-미리 보기)은 다음 도구, 애플리케이션 및 라이브러리를 지원합니다.

[\[+\] 테이블 확장](#)

기능	리소스	Model ID
청구서 모델	<ul style="list-style-type: none"> 문서 인텔리전스 스튜디오 REST API C# SDK Python SDK Java SDK JavaScript SDK 	prebuilt-invoice

입력 요구 사항

- 최상의 결과를 위해 문서당 하나의 명확한 사진 또는 고품질 스캔을 제공합니다.
- 지원 파일 형식:

모델	PDF	이미지: JPEG/JPG, PNG, BMP, TIFF, HEIF	Microsoft Office: Word(DOCX), Excel(XLSX), PowerPoint(PPTX) 및 HTML
읽음	✓	✓	✓
레이아웃	✓	✓	✓ (2024-02-29-미리 보기, 2023-10-31-미리 보기)
일반 문서	✓	✓	
사전 제작	✓	✓	
사용자 지정 추출	✓	✓	
사용자 지정 분류	✓	✓	✓ (2024-02-29-미리 보기)

- PDF 및 TIFF의 경우 최대 2000페이지를 처리할 수 있습니다(무료 계층 구독의 경우 처음 2페이지만 처리됨).
- 문서 분석을 위한 파일 크기는 유료(S0) 계층의 경우 500MB, 무료(F0) 계층의 경우 4MB입니다.
- 이미지 크기는 50 x 50픽셀에서 10,000 x 10,000픽셀 사이여야 합니다.
- PDF가 암호로 잠긴 경우에는 제출하기 전에 잠금을 해제해야 합니다.
- 추출할 텍스트의 최소 높이는 1024 x 768 픽셀 이미지의 경우 12픽셀입니다. 이 차원은 150DPI(인치당 도트 수)에서 약 8포인트 텍스트에 해당합니다.
- 사용자 지정 모델 학습의 경우 학습 데이터의 최대 페이지 수는 사용자 지정 템플릿 모델의 경우 500개, 사용자 지정 인공신경망 모델의 경우 50,000개입니다.
 - 사용자 지정 추출 모델 학습의 경우 학습 데이터의 총 크기는 템플릿 모델의 경우 50MB, 신경망 모델의 경우 1G-MB입니다.
 - 사용자 지정 분류 모델 학습의 경우 학습 데이터의 총 크기는 최대 10,000페이지인 1GB입니다.

청구서 모델 데이터 추출

고객 정보, 공급업체 세부 정보 및 라인 항목을 포함한 데이터가 청구서에서 추출되는 방법을 확인합니다. 다음 리소스가 필요합니다.

- Azure 구독은 [무료로 만들 수](#) 있습니다.
- Azure Portal의 [Document Intelligence 인스턴스](#)입니다. 무료 가격 책정 계층(F0)을 사용하여 서비스를 시도할 수 있습니다. 리소스가 배포된 후 [리소스로 이동](#)을 선택하여 키 및 엔드포인트를 가져옵니다.

Home > Contoso-DI

Contoso-DI | Keys and Endpoint

Search | Regenerate Key1 | Regenerate Key2

Overview | Activity log | Access control (IAM) | Tags | Diagnose and solve problems | Resource Management | Keys and Endpoint | Encryption | Pricing tier | Networking | Identity | Cost analysis | Properties | Locks | Monitoring | Automation | Help

These keys are used to access your Azure AI service API. Do not share your keys. Store them securely—for example, using Azure Key Vault. We also recommend regenerating these keys regularly. Only one key is necessary to make an API call. When regenerating the first key, you can use the second key for continued access to the service.

Show Keys

KEY 1

KEY 2

Location/Region westus2

Endpoint https://contoso-di.cognitiveservices.azure.com/

1. 문서 인텔리전스 스튜디오 홈페이지에서 **청구서**를 선택합니다.
2. 샘플 청구서를 분석하거나 자체 파일을 업로드할 수 있습니다.
3. **분석 실행** 버튼을 선택하고 필요한 경우 **분석 옵션**을 구성합니다.



문서 인텔리전스 스튜디오를 사용해 보세요.

지원되는 언어 및 로캘

지원되는 언어의 전체 목록은 [언어 지원 - 미리 빌드된 모델 페이지](#)를 참조하세요.

필드 추출

이름	형식	설명	표준화된 출력
CustomerName	문자 열	청구서가 청구된 고객	
고객 ID	문자 열	고객 참조 ID	
PurchaseOrder	문자 열	구매 주문 참조 번호	
InvoiceId	문자 열	이 특정 청구서의 ID(많은 경우 '청구서 번호')	
InvoiceDate	날짜	청구서가 발행된 날짜	yyyy-mm- dd
DueDate	날짜	이 청구서의 지불 기일	yyyy-mm- dd
VendorName	문자 열	공급업체 이름	
VendorTaxId	문자 열	공급업체와 연결된 납세자 번호	
VendorAddress	문자 열	공급업체 우편 주소	
VendorAddressRecipient	문자 열	VendorAddress와 관련된 이름	
CustomerAddress	문자 열	고객의 우편 주소	
CustomerTaxId	문자 열	고객과 연결된 납세자 번호	
CustomerAddressRecipient	문자 열	CustomerAddress와 관련된 이름	
BillingAddress	문자 열	고객의 명시적 청구 주소	
BillingAddressRecipient	문자 열	BillingAddress와 관련된 이름	
ShippingAddress	문자 열	고객의 명시적 배송 주소	

이름	형식	설명	표준화된 출력
ShippingAddressRecipient	문자 열	ShippingAddress와 관련된 이름	
PaymentTerm	문자 열	청구서에 대한 지불 조건	
소계합계	숫자	이 청구서에서 식별된 소계 필드	정수
TotalTax	숫자	이 청구서에서 식별된 세금 총액 필드	정수
InvoiceTotal	숫자 (USD)	이 청구서와 관련된 새 요금 총액	정수
AmountDue	숫자 (USD)	공급업체에 지불할 총액	정수
ServiceAddress	문자 열	고객의 명시적 서비스 주소 또는 건물 주소	
ServiceAddressRecipient	문자 열	ServiceAddress와 관련된 이름	
RemittanceAddress	문자 열	고객의 명시적 송금 주소 또는 지불 주소	
RemittanceAddressRecipient	문자 열	RemittanceAddress와 관련된 이름	
ServiceStartDate	날짜	서비스 기간의 시작 날짜(예: 유ти리티 청구 서비스 기간)	yyyy-mm- dd
ServiceEndDate	날짜	서비스 기간의 종료 날짜(예: 유ти리티 청구 서비스 기간)	yyyy-mm- dd
PreviousUnpaidBalance	숫자	이전에 미지불된 명시적 잔액	정수
통화 코드	문자 열	추출된 금액과 연결된 통화 코드	
KVKNumber(NL 전용)	문자 열	네덜란드에 등록된 기업의 고유 식별자	12345678
PaymentDetails	배열	IBAN, SWIFT, BPay(AU) 같은 결제 옵션 세부 정보를 보유하는 배열	
TotalDiscount	숫자	청구서에 적용된 총 할인	정수
TaxItems	Array	CGST, IGST, SGST 같은 추가 세금 정보를 보 유하는 배열입니다. 이 품목은 현재 독일(de),	

이름	형식	설명	표준화된 출력
		스페인(es), 포르투갈(pt) 및 영어 캐나다(en-CA)로 캘에서만 사용할 수 있습니다.	

품목

다음은 JSON 출력 응답의 청구서에서 추출된 품목입니다. 다음 출력에서는 이 [샘플 청구서](#)를 사용합니다.

 테이블 확장

속성	형식	설명	텍스트(개별 항목 1)	값(표준화된 출력)
아이템	문자열	개별 항목의 전체 문자열 텍스트 줄	2021/3/4 A123 컨설팅 서비스 \$30.00 10% \$60.00	
금액	번호	개별 항목의 금액	\$60.00	100
설명	문자열	청구서 개별 항목의 텍스트 설명	컨설팅 서비스	컨설팅 서비스
수량	숫자	이 청구서 개별 항목의 수량	2	2
단가	숫자	이 항목의 1개 단위의 정가 및 총 가(청구서의 청구서 총액 설정에 따라 다름)	\$30.00	30
ProductCode	문자열	특정 개별 항목과 관련된 제품 코드, 제품 번호 또는 SKU	A123	
단위	문자열	개별 항목의 단위(예: kg, lb 등)	Hours	
날짜	날짜	각 개별 항목에 해당하는 날짜이며, 개별 항목이 배송된 날짜인 경우가 많음	2021/3/4	2021-03-04
세금	숫자	각 개별 항목과 관련된 세금이며, 가능한 값으로는 세액 및 세금 Y/N이 있음	10.00	

속성	형식	설명	텍스트(개별 항목 1)	값(표준화된 출력)
TaxRate	숫자	각 개별 항목과 관련된 세율입니다.	10%	

추출된 청구서 키-값 쌍 및 품목은 JSON 출력의 `documentResults` 섹션에 있습니다.

다음 단계

- 문서 인텔리전스 스튜디오 [☞](#)를 사용하여 사용자 고유의 양식 및 문서를 처리해 보세요.
- Document Intelligence 빠른 시작을 완료하고 원하는 개발 언어로 문서 처리 앱 만들기를 시작해 보세요.

문서 인텔리전스 영수증 모델

아티클 • 2024. 04. 15.

① 중요

- 문서 인텔리전스 공개 미리 보기 릴리스에서는 현재 개발 중인 기능에 대한 조기 액세스를 제공합니다.
- 기능, 방식 및 프로세스는 GA(일반 공급) 전에 사용자 피드백에 따라 변경될 수 있습니다.
- 문서 인텔리전스 클라이언트 라이브러리의 공개 미리 보기 버전은 기본적으로 REST API 버전 [2024-02-29-미리 보기](#)로 설정됩니다.
- 공개 미리 보기 버전 [2024-02-29-미리 보기](#)는 현재 다음 Azure 지역에서만 사용할 수 있습니다.
 - 미국 동부
 - 미국 서부2
 - 서유럽

이 콘텐츠의 적용 대상은 다음과 같습니다. v4.0(미리 보기) | 이전 버전: v3.1(GA) v3.0(GA) v2.1(GA)

문서 인텔리전스 영수증 모델은 강력한 OCR(광학 문자 인식) 기능과 딥 러닝 모델을 결합하여 판매 영수증에서 주요 정보를 분석하고 추출합니다. 영수증은 인쇄 및 필기 영수증을 포함하여 다양한 형식과 품질일 수 있습니다. API는 가맹점 이름, 가맹점 전화 번호, 거래 날짜, 세금 및 거래 합계와 같은 주요 정보를 추출하고 구조화된 JSON 데이터를 반환합니다.

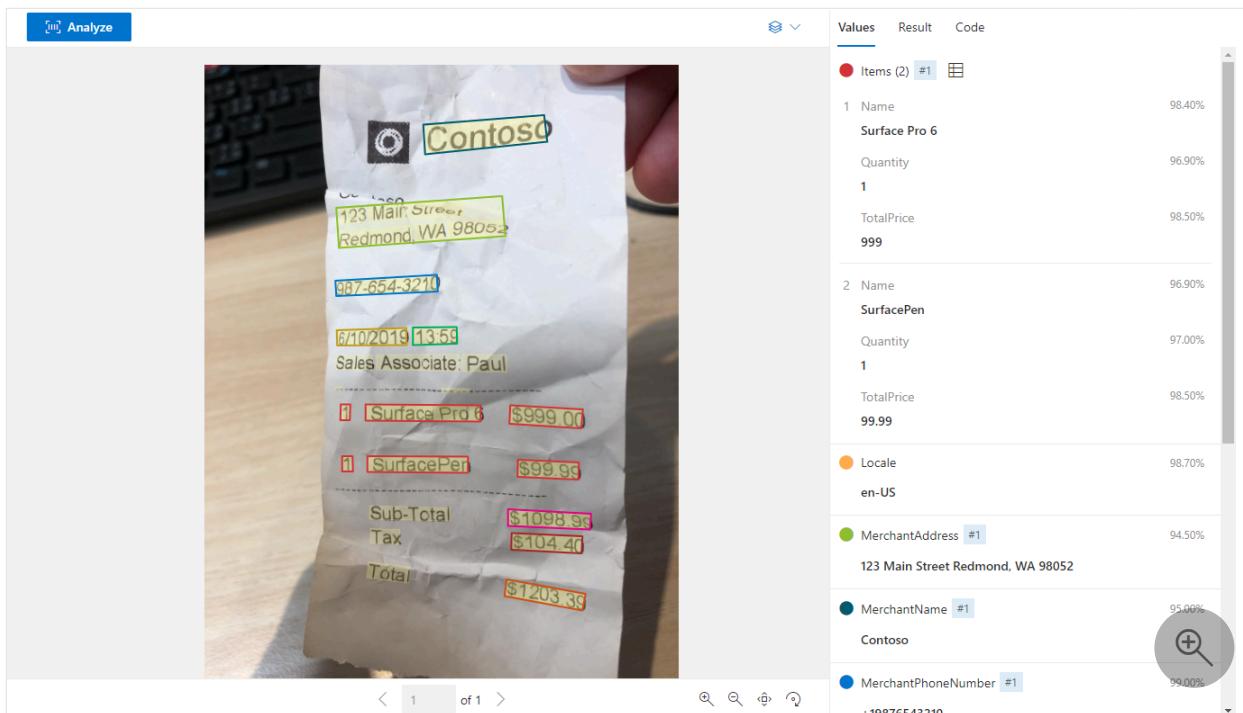
지원되는 영수증 유형:

- 열
- 신용 카드
- 가스
- 주차
- 호텔
- 운송
- 통신
- Entertainment
- 항공편
- 학습
- 약국/의료

영수증 데이터 추출

영수증 디지털화는 스캔, 사진 촬영, 인쇄 사본을 포함한 다양한 유형의 영수증을 간소화된 다운스트림 처리를 위해 디지털 형식으로 변환하는 것을 포함합니다. 예로는 비용 관리, 소비자 행동 분석, 세금 자동화 등이 있습니다. OCR(광학 문자 인식) 기술이 적용된 문서 인텔리전스를 사용하면 이러한 다양한 영수증 형식에서 데이터를 추출하고 해석할 수 있습니다. 문서 인텔리전스 처리는 변환 프로세스를 간소화하지만 필요한 시간과 노력을 크게 줄여 효율적인 데이터 관리 및 검색을 용이하게 합니다.

문서 인텔리전스 스튜디오 로 처리된 샘플 영수증:



The screenshot shows the Microsoft Document Intelligence Studio interface. On the left, there is a preview of a physical receipt from 'Contoso' located at '123 Main Street Redmond, WA 98052'. The receipt includes a phone number '987-654-3210', a date '6/10/2019 13:59', and a sales associate 'Paul'. It lists two items: 'Surface Pro 6' and 'SurfacePen', both priced at '\$999.00'. The total is calculated as Sub-Total '\$1098.95' plus Tax '\$104.40', resulting in a Total of '\$1203.39'. On the right, the 'Values' tab of the results pane is selected, displaying the extracted data with confidence scores. The results include:

Field	Value	Confidence Score
Name	Surface Pro 6	98.40%
Quantity	1	96.90%
TotalPrice	\$999	98.50%
Name	SurfacePen	96.90%
Quantity	1	97.00%
TotalPrice	\$99.99	98.50%
Locale	en-US	98.70%
MerchantAddress	123 Main Street Redmond, WA 98052	94.50%
MerchantName	Contoso	95.00%
MerchantPhoneNumber	+19876543210	99.00%

개발 옵션

문서 인텔리전스 v4.0(2024-02-29-미리 보기 2023-10-31-미리 보기)은 다음 도구, 애플리케이션 및 라이브러리를 지원합니다.

테이블 확장

기능	리소스	Model ID
영수증 모델	<ul style="list-style-type: none">문서 인텔리전스 스튜디오REST APIC# SDKPython SDKJava SDKJavaScript SDK	prebuilt-receipt

입력 요구 사항

- 최상의 결과를 위해 문서당 하나의 명확한 사진 또는 고품질 스캔을 제공합니다.
- 지원 파일 형식:

[+] 테이블 확장

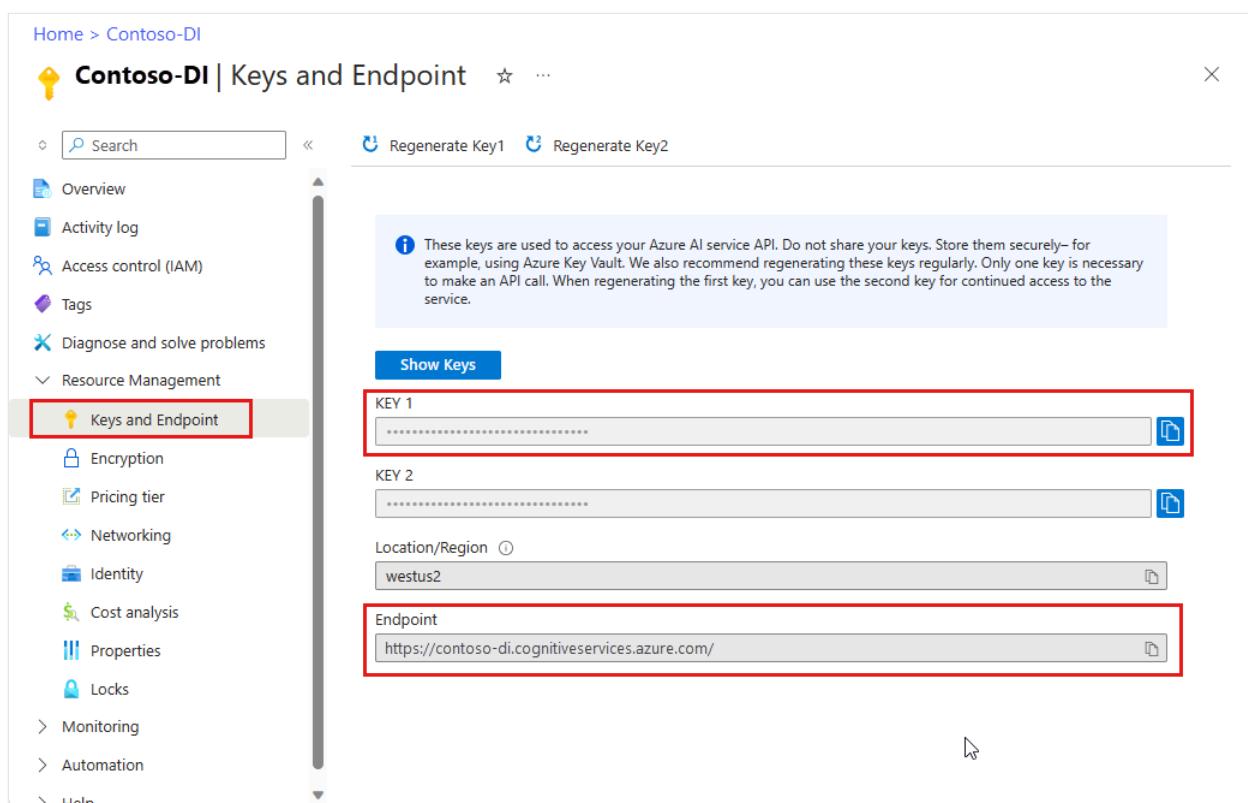
모델	PDF	이미지: JPEG/JPG, PNG, BMP, TIFF, HEIF	Microsoft Office: Word(DOCX), Excel(XLSX), PowerPoint(PPTX) 및 HTML
읽음	✓	✓	✓
레이아웃	✓	✓	✓ (2024-02-29-미리 보기, 2023-10-31-미리 보기)
일반 문서	✓	✓	
사전 제작	✓	✓	
사용자 지정 추출	✓	✓	
사용자 지정 분류	✓	✓	✓ (2024-02-29-미리 보기)

- PDF 및 TIFF의 경우 최대 2000페이지를 처리할 수 있습니다(무료 계층 구독의 경우 처음 2페이지만 처리됨).
- 문서 분석을 위한 파일 크기는 유료(S0) 계층의 경우 500MB, 무료(F0) 계층의 경우 4MB입니다.
- 이미지 크기는 50 x 50픽셀에서 10,000 x 10,000픽셀 사이여야 합니다.
- PDF가 암호로 잠긴 경우에는 제출하기 전에 잠금을 해제해야 합니다.
- 추출할 텍스트의 최소 높이는 1024 x 768 픽셀 이미지의 경우 12픽셀입니다. 이 차원은 150DPI(인치당 도트 수)에서 약 8포인트 텍스트에 해당합니다.
- 사용자 지정 모델 학습의 경우 학습 데이터의 최대 페이지 수는 사용자 지정 템플릿 모델의 경우 500개, 사용자 지정 인공신경망 모델의 경우 50,000개입니다.
 - 사용자 지정 추출 모델 학습의 경우 학습 데이터의 총 크기는 템플릿 모델의 경우 50MB, 신경망 모델의 경우 1G-MB입니다.
 - 사용자 지정 분류 모델 학습의 경우 학습 데이터의 총 크기는 최대 10,000페이지인 1GB입니다.

영수증 모델 데이터 추출

문서 인텔리전스가 거래 시간 및 날짜, 판매자 정보, 영수증의 총액 등의 데이터를 추출하는 방법을 알아보세요. 다음 리소스가 필요합니다.

- Azure 구독은 [무료로 만들 수](#) 있습니다.
- Azure Portal의 [Document Intelligence 인스턴스](#)입니다. 무료 가격 책정 계층(F0)을 사용하여 서비스를 시도할 수 있습니다. 리소스가 배포된 후 [리소스로 이동](#)을 선택하여 키 및 엔드포인트를 가져옵니다.

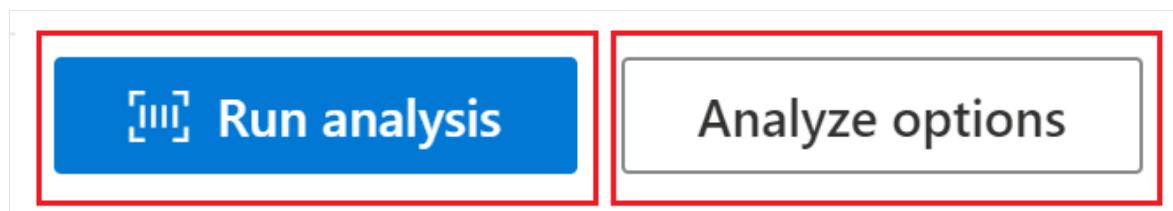


The screenshot shows the Azure portal interface for a 'Contoso-DI' document intelligence instance. The left sidebar lists various service management options, with 'Keys and Endpoint' highlighted by a red box. The main content area displays the configuration for the AI service, including two API keys (KEY 1 and KEY 2), the location set to 'westus2', and the endpoint URL 'https://contoso-di.cognitiveservices.azure.com/'. A note at the top of the page cautions against sharing the keys securely, such as using Azure Key Vault, and recommends regular key regeneration.

① 참고

문서 인텔리전스 스튜디오는 v3.1 및 v3.0 API 및 이후 버전에서 사용할 수 있습니다.

1. 문서 인텔리전스 스튜디오 홈페이지에서 **영수증**을 선택합니다.
2. 샘플 영수증을 분석하거나 자체 파일을 업로드할 수 있습니다.
3. **분석 실행** 버튼을 선택하고 필요한 경우 **분석 옵션**을 구성합니다.



문서 인텔리전스 스튜디오를 사용해 보세요

지원되는 언어 및 로캘

지원되는 언어의 전체 목록은 [언어 지원 - 미리 빌드된 모델 페이지](#)를 참조하세요.

필드 추출

문서 인텔리전스 v3.0 이상 버전에는 몇 가지 새로운 기능이 도입되었습니다. 열 영수증 외에도 **영수증** 모델은 단일 페이지 호텔 영수증 처리 및 모든 영수증 형식에 대한 세금 세부 정보 추출을 지원합니다.

Document Intelligence v4.0 이상 버전에서는 열 및 호텔 영수증에 대한 모든 가격 관련 필드에 대한 통화 지원이 도입되었습니다.

Receipt

[+] 테이블 확장

필드	형식	Description	예시
MerchantName	string	영수증을 발급한 판매자의 이름	Contoso
MerchantPhoneNumber	phoneNumber	판매자 전화 번호 나열	987-654-3210
MerchantAddress	address	판매자의 주소 나열	123 Main St. Redmond, Washington 98052
Total	number	영수증의 총 트랜잭션 합계	\$14.34
TransactionDate	date	영수증이 발급된 날짜	2019년 6월 6일
TransactionTime	time	영수증이 발급된 시간	오후 4시 49분
Subtotal	number	영수증의 소계(보통 세금이 적용되기 전)	\$12.34
TotalTax	number	영수증에 대한 세금(보통 판매 세금 또는 이에 상응하는 금액)	\$2.00
Tip	number	구매자에 의해 포함된 팁	\$1.00
Items	array		
Items.*	object	추출된 품목	1 Surface Pro 6

필드	형식	Description	예시
			\$999.00 \$999.00
Items.*.TotalPrice	number	품목에 대한 총 가격	\$999.00
Items.*.Description	string	항목 설명	Surface Pro 6
Items.*.Quantity	number	각 항목의 수량	1
Items.*.Price	number	각 항목 단위의 개별 가격	\$999.00
Items.*.ProductCode	string	특정 개별 항목과 관련된 제품 코드, 제품 번호 또는 SKU	A123
Items.*.QuantityUnit	string	각 항목의 수량 단위	
TaxDetails	array		
TaxDetails.*	object	추출된 품목	1 Surface Pro 6 \$999.00 \$999.00
TaxDetails.*.Amount	currency	세금 세부 금액	\$999.00

receipt.retailMeal

[+] 테이블 확장

필드	형식	Description	예示
MerchantName	string	영수증을 발급한 판매자의 이름	Contoso
MerchantPhoneNumber	phoneNumber	판매자 전화 번호 나열	987-654-3210
MerchantAddress	address	판매자의 주소 나열	123 Main St. Redmond, Washington 98052
Total	number	영수증의 총 트랜잭션 합계	\$14.34
TransactionDate	date	영수증이 발급된 날짜	2019년 6월 6일
TransactionTime	time	영수증이 발급된 시간	오후 4시 49분
Subtotal	number	영수증의 소계(보통 세금이 적용되기 전)	\$12.34
TotalTax	number	영수증에 대한 세금(보통 판매)	\$2.00

필드	형식	Description	예시
		세금 또는 이에 상응하는 금액)	
Tip	number	구매자에 의해 포함된 팁	\$1.00
Items	array		
Items.*	object	추출된 품목	1 Surface Pro 6 \$999.00 \$999.00
Items.*.TotalPrice	number	품목에 대한 총 가격	\$999.00
Items.*.Description	string	항목 설명	Surface Pro 6
Items.*.Quantity	number	각 항목의 수량	1
Items.*.Price	number	각 항목 단위의 개별 가격	\$999.00
Items.*.ProductCode	string	특정 개별 항목과 관련된 제품 코드, 제품 번호 또는 SKU	A123
Items.*.QuantityUnit	string	각 항목의 수량 단위	
TaxDetails	array		
TaxDetails.*	object	추출된 품목	1 Surface Pro 6 \$999.00 \$999.00
TaxDetails.*.Amount	currency	세금 세부 금액	\$999.00

receipt.creditCard

[+] 테이블 확장

필드	형식	Description	예시
MerchantName	string	영수증을 발급한 판매자의 이름	Contoso
MerchantPhoneNumber	phoneNumber	판매자 전화 번호 나열	987-654-3210
MerchantAddress	address	판매자의 주소 나열	123 Main St. Redmond, Washington 98052
Total	number	영수증의 총 트랜잭션 합계	\$14.34

필드	형식	Description	예시
TransactionDate	date	영수증이 발급된 날짜	2019년 6월 6일
TransactionTime	time	영수증이 발급된 시간	오후 4시 49분
Subtotal	number	영수증의 소계(보통 세금이 적용되기 전)	\$12.34
TotalTax	number	영수증에 대한 세금(보통 판매 세금 또는 이에 상응하는 금액)	\$2.00
Tip	number	구매자에 의해 포함된 팁	\$1.00
Items	array		
Items.*	object	추출된 품목	1 Surface Pro 6 \$999.00 \$999.00
Items.*.TotalPrice	number	품목에 대한 총 가격	\$999.00
Items.*.Description	string	항목 설명	Surface Pro 6
Items.*.Quantity	number	각 항목의 수량	1
Items.*.Price	number	각 항목 단위의 개별 가격	\$999.00
Items.*.ProductCode	string	특정 개별 항목과 관련된 제품 코드, 제품 번호 또는 SKU	A123
Items.*.QuantityUnit	string	각 항목의 수량 단위	
TaxDetails	array		
TaxDetails.*	object	추출된 품목	1 Surface Pro 6 \$999.00 \$999.00
TaxDetails.*.Amount	currency	세금 세부 금액	\$999.00

receipt.gas

[+] 테이블 확장

필드	형식	Description	예시
MerchantName	string	영수증을 발급한 판매자의 이름	Contoso

필드	형식	Description	예시
MerchantPhoneNumber	phoneNumber	판매자 전화 번호 나열	987-654-3210
MerchantAddress	address	판매자의 주소 나열	123 Main St. Redmond, Washington 98052
Total	number	영수증의 총 트랜잭션 합계	\$14.34
TransactionDate	date	영수증이 발급된 날짜	2019년 6월 6일
TransactionTime	time	영수증이 발급된 시간	오후 4시 49분
Subtotal	number	영수증의 소계(보통 세금이 적용되기 전)	\$12.34
TotalTax	number	영수증에 대한 세금(보통 판매 세금 또는 이에 상응하는 금액)	\$2.00
Tip	number	구매자에 의해 포함된 팁	\$1.00
Items	array		
Items.*	object	추출된 품목	1 Surface Pro 6 \$999.00 \$999.00
Items.*.TotalPrice	number	품목에 대한 총 가격	\$999.00
Items.*.Description	string	항목 설명	Surface Pro 6
Items.*.Quantity	number	각 항목의 수량	1
Items.*.Price	number	각 항목 단위의 개별 가격	\$999.00
Items.*.ProductCode	string	특정 개별 항목과 관련된 제품 코드, 제품 번호 또는 SKU	A123
Items.*.QuantityUnit	string	각 항목의 수량 단위	
TaxDetails	array		
TaxDetails.*	object	추출된 품목	1 Surface Pro 6 \$999.00 \$999.00
TaxDetails.*.Amount	currency	세금 세부 금액	\$999.00

receipt.parking

필드	형식	Description	예시
MerchantName	string	영수증을 발급한 판매자의 이름	Contoso
MerchantPhoneNumber	phoneNumber	판매자 전화 번호 나열	987-654-3210
MerchantAddress	address	판매자의 주소 나열	123 Main St. Redmond, Washington 98052
Total	number	영수증의 총 트랜잭션 합계	\$14.34
TransactionDate	date	영수증이 발급된 날짜	2019년 6월 6일
TransactionTime	time	영수증이 발급된 시간	오후 4시 49분
Subtotal	number	영수증의 소계(보통 세금이 적용되기 전)	\$12.34
TotalTax	number	영수증에 대한 세금(보통 판매 세금 또는 이에 상응하는 금액)	\$2.00
Tip	number	구매자에 의해 포함된 팁	\$1.00
Items	array		
Items.*	object	추출된 품목	1 Surface Pro 6 \$999.00 \$999.00
Items.*.TotalPrice	number	품목에 대한 총 가격	\$999.00
Items.*.Description	string	항목 설명	Surface Pro 6
Items.*.Quantity	number	각 항목의 수량	1
Items.*.Price	number	각 항목 단위의 개별 가격	\$999.00
Items.*.ProductCode	string	특정 개별 항목과 관련된 제품 코드, 제품 번호 또는 SKU	A123
Items.*.QuantityUnit	string	각 항목의 수량 단위	
TaxDetails	array		
TaxDetails.*	object	추출된 품목	1 Surface Pro 6 \$999.00 \$999.00

필드	형식	Description	예시
TaxDetails.*.Amount	currency	세금 세부 금액	\$999.00

receipt.hotel

[+] 테이블 확장

필드	형식	Description	예시
MerchantName	string	영수증을 발급한 판매자의 이름	Contoso
MerchantPhoneNumber	phoneNumber	판매자 전화 번호 나열	987-654-310
MerchantAddress	address	판매자의 주소 나열	123 Main St. Redmond, Washington 98052
Total	number	영수증의 총 트랜잭션 합계	\$14.34
ArrivalDate	date	도착 날짜	27Mar21
DepartureDate	date	출발 날짜	28Mar21
Currency	string	영수증 금액의 통화 단위(ISO 4217) 또는 여러 값이 있는 경우 'MIXED'	USD
MerchantAliases	array		
MerchantAliases.*	string	가맹점의 대체 이름	Contoso (R)
Items	array		
Items.*	object	추출된 품목	1 Surface Pro 6 \$999.00 \$999.00
Items.*.TotalPrice	number	품목에 대한 총 가격	\$999.00
Items.*.Description	string	항목 설명	객실 요금
Items.*.Date	date	항목 날짜	27Mar21
Items.*.Category	string	항목 범주	방

다음 단계

- 문서 인텔리전스 스튜디오 [\[링크\]](#)를 사용하여 사용자 고유의 양식 및 문서를 처리해 보세요.
- Document Intelligence 빠른 시작을 완료하고 원하는 개발 언어로 문서 처리 앱 만들기를 시작해 보세요.

문서 인텔리전스 ID 문서 모델

아티클 • 2024. 04. 11.

ⓘ 중요

- 문서 인텔리전스 공개 미리 보기 릴리스에서는 현재 개발 중인 기능에 대한 조기 액세스를 제공합니다.
- 기능, 방식 및 프로세스는 GA(일반 공급) 전에 사용자 피드백에 따라 변경될 수 있습니다.
- 문서 인텔리전스 클라이언트 라이브러리의 공개 미리 보기 버전은 기본적으로 REST API 버전 [2024-02-29-미리 보기](#)로 설정됩니다.
- 공개 미리 보기 버전 [2024-02-29-미리 보기](#)는 현재 다음 Azure 지역에서만 사용할 수 있습니다.
 - 미국 동부
 - 미국 서부2
 - 서유럽

이 콘텐츠의 적용 대상은 다음과 같습니다. v4.0(미리 보기) | 이전 버전: v3.1(GA) v3.0(GA) v2.1(GA)

문서 인텔리전스 ID(IDentity Document) 모델은 OCR(광학 문자 인식)과 딥 러닝 모델을 결합하여 ID로부터 핵심 정보를 분석하고 추출합니다. API는 ID 문서(다음 포함)를 분석하고, 구조화된 JSON 데이터 표현을 반환합니다.

- 전 세계 여권 수첩, 여권 카드
- 미국, 유럽, 인도, 캐나다 및 오스트레일리아의 운전 면허증
- 미국 신분증, 거주 허가증(녹색 카드), 사회 보장 카드, 군인 ID
- 유럽 신분증, 거주 허가증
- 인도 PAN 카드, 아드하르 카드
- 캐나다 신분증, 거주 허가증(메이플 카드)
- 오스트레일리아 사진 카드, 키 패스 ID(디지털 버전 포함)

ID 문서 처리

ID 문서 처리에는 수동으로 또는 OCR 기반 기술을 사용하여 ID 문서에서 데이터를 추출하는 작업이 포함됩니다. ID 문서 처리는 ID 증명이 필요한 비즈니스 작업에서 중요한 단계입니다. 예를 들어, 은행 및 기타 금융 기관의 고객 확인, 대출 신청, 병원 진료, 클레임 처리, 관광 산업 등이 있습니다. 개인은 비즈니스에서 서비스 및 혜택을 제공하기 전에 효율적으로 확인할 수 있도록 운전 면허증, 여권 및 기타 유사한 문서를 통해 몇 가지 신원 증명을 제공합니다.

문서 인텔리전스에서 처리된 [미국 운전 면허증 샘플](#)

Analyze

Values	Result	Code
Address #1	123 STREET ADDRESS YOUR CITY WA 99999-1234	87.20%
CountryRegion	USA	99.50%
DateOfBirth #1	1958-01-06	98.90%
DateOfExpiration #1	2020-08-12	98.60%
DocumentNumber	WDLABCD456DG	97.50%
Endorsements #1	L	98.40%
FirstName #1	LIAM R.	84.60%
LastName #1	TALBOT	93.10%
Locale	en-US	100.00%

개발 옵션

문서 인텔리전스 v4.0(2024-02-29-미리 보기 2023-10-31-미리 보기)은 다음 도구, 애플리케이션 및 라이브러리를 지원합니다.

□ 테이블 확장

기능	리소스	Model ID
ID 문서 모델	<ul style="list-style-type: none"> 문서 인텔리전스 스튜디오 REST API C# SDK Python SDK Java SDK JavaScript SDK 	prebuilt-idDocument

입력 요구 사항

- 최상의 결과를 위해 문서당 하나의 명확한 사진 또는 고품질 스캔을 제공합니다.
- 지원 파일 형식:

□ 테이블 확장

모델	PDF	이미지: JPEG/JPG, PNG, BMP, TIFF, HEIF	Microsoft Office: Word(DOCX), Excel(XLSX), PowerPoint(PPTX) 및 HTML
읽음	✓	✓	✓
레이아웃	✓	✓	✓ (2024-02-29-미리 보기, 2023-10-31-미리 보기)
일반 문서	✓	✓	

모델	PDF	이미지: JPEG/JPG, PNG, BMP, TIFF, HEIF	Microsoft Office: Word(DOCX), Excel(XLSX), PowerPoint(PPTX) 및 HTML
사전 제작	✓	✓	
사용자 지정 추출	✓	✓	
사용자 지정 분류	✓	✓	✓ (2024-02-29-미리 보기)

- PDF 및 TIFF의 경우 최대 2000페이지를 처리할 수 있습니다(무료 계층 구독의 경우 처음 2페이지만 처리됨).
- 문서 분석을 위한 파일 크기는 유료(S0) 계층의 경우 500MB, 무료(F0) 계층의 경우 4MB입니다.
- 이미지 크기는 50 x 50픽셀에서 10,000 x 10,000픽셀 사이여야 합니다.
- PDF가 암호로 잠긴 경우에는 제출하기 전에 잠금을 해제해야 합니다.
- 추출할 텍스트의 최소 높이는 1024 x 768 픽셀 이미지의 경우 12픽셀입니다. 이 차원은 150DPI(인치당 도트 수)에서 약 8포인트 텍스트에 해당합니다.
- 사용자 지정 모델 학습의 경우 학습 데이터의 최대 페이지 수는 사용자 지정 템플릿 모델의 경우 500개, 사용자 지정 인공신경망 모델의 경우 50,000개입니다.
 - 사용자 지정 추출 모델 학습의 경우 학습 데이터의 총 크기는 템플릿 모델의 경우 50MB, 신경망 모델의 경우 1G-MB입니다.
 - 사용자 지정 분류 모델 학습의 경우 학습 데이터의 총 크기는 최대 10,000페이지인 1GB입니다.

ID 문서 모델 데이터 추출

ID 문서에서 이름, 생년월일, 만료 날짜를 포함한 데이터를 추출합니다. 다음 리소스가 필요합니다.

- Azure 구독은 [무료로 만들 수](#) 있습니다.
- Azure Portal의 [Document Intelligence 인스턴스](#)입니다. 무료 가격 책정 계층(F0)을 사용하여 서비스를 시도 할 수 있습니다. 리소스가 배포된 후 [리소스로 이동](#)을 선택하여 키 및 앤드포인트를 가져옵니다.

The screenshot shows the Azure portal interface for managing a Cognitive Services resource. The left sidebar has a tree view with 'Keys and Endpoint' selected. The main content area shows the following details:

- Show Keys** button
- KEY 1**: A red box highlights the first key value.
- KEY 2**: A red box highlights the second key value.
- Location/Region**: westus2
- Endpoint**: https://contoso-di.cognitiveservices.azure.com/ (highlighted by a red box)

① 참고

문서 인텔리전스 스튜디오는 v3.1 및 v3.0 API 및 이후 버전에서 사용할 수 있습니다.

1. 문서 인텔리전스 스튜디오 홈페이지에서 신분 문서를 선택합니다.
2. 샘플 청구서를 분석하거나 자체 파일을 업로드할 수 있습니다.
3. **분석 실행** 버튼을 선택하고 필요한 경우 **분석 옵션**을 구성합니다.



[문서 인텔리전스 스튜디오 사용해 보기](#)

지원되는 문서 유형

[데이터 확장](#)

지역	문서 형식
전 세계	여권 수첩, 여권 카드
미국	운전 면허증, 신분증, 거주 허가증(녹색 카드), 사회 보장 카드, 군인 ID
유럽	운전 면허증, 신분증, 거주 허가증
인도	운전 면허증, PAN 카드, 아드하르 카드
캐나다	운전 면허증, 신분증, 거주 허가증(메이플 카드)

지역	문서 형식
오스트레일리아	운전 면허증, 사진 카드, 키 패스 ID(디지털 버전 포함)

필드 추출

다음은 문서 유형별로 추출된 필드입니다. 문서 인텔리전스 ID 모델 `prebuilt-idDocument` 은(는) `documents.*.fields`에서 다음 필드를 추출합니다. json 출력에는 문서, 단어, 선 및 스타일에 추출된 모든 텍스트가 포함됩니다.

idDocument.driverLicense

 테이블 확장

필드	형식	Description	예시
<code>CountryRegion</code>	<code>countryRegion</code>	국가 또는 지역 코드	미국
<code>Region</code>	<code>string</code>	시 또는 도	워싱턴
<code>DocumentNumber</code>	<code>string</code>	운전 면허증 번호	WDLABCD456DG
<code>DocumentDiscriminator</code>	<code>string</code>	운전 면허증 문서 판별자	12645646464554646456464544
<code>FirstName</code>	<code>string</code>	지정된 이름 및 중간 이니셜(해당하는 경우)	LIAM R.
<code>LastName</code>	<code>string</code>	Surname	TALBOT
<code>Address</code>	<code>address</code>	주소	123 STREET ADDRESS YOUR CITY WA 99999-1234
<code>DateOfBirth</code>	<code>date</code>	생년월일	01/06/1958
<code>DateOfExpiration</code>	<code>date</code>	만료 날짜	08/12/2020
<code>DateOfIssue</code>	<code>date</code>	이슈 날짜	08/12/2012
<code>EyeColor</code>	<code>string</code>	눈 색	파랑
<code>HairColor</code>	<code>string</code>	모발 색	Brown
<code>Height</code>	<code>string</code>	Height	5'11"
<code>Weight</code>	<code>string</code>	가중치	185LB
<code>Sex</code>	<code>string</code>	성별	M
<code>Endorsements</code>	<code>string</code>	보증	L
<code>Restrictions</code>	<code>string</code>	제한 사항	B
<code>VehicleClassifications</code>	<code>string</code>	차량 분류	D

idDocument.passport

 테이블 확장

필드	형식	Description	예시
<code>DocumentNumber</code>	<code>string</code>	여권 번호	340020013

idDocument.nationalIdentityCard

 테이블 확장

필드	형식	Description	예시
CountryRegion	countryRegion	국가 또는 지역 코드	미국
Region	string	시 또는 도	워싱턴
DocumentNumber	string	국가 ID 카드 번호	WDLABCD456DG
DocumentDiscriminator	string	국가 ID 카드 문서 판별자	12645646464554646456464544
FirstName	string	지정된 이름 및 중간 이니셜(해당하는 경우)	LIAM R.
LastName	string	Surname	TALBOT
Address	address	주소	123 STREET ADDRESS YOUR CITY WA 99999-1234
DateOfBirth	date	생년월일	01/06/1958
DateOfExpiration	date	만료 날짜	08/12/2020
DateOfIssue	date	이슈 날짜	08/12/2012
EyeColor	string	눈 색	파란색
HairColor	string	모발 색	갈색
Height	string	Height	5'11"
Weight	string	가중치	185LB
Sex	string	성별	M

idDocument.residencePermit

 테이블 확장

필드	형식	Description	예시
CountryRegion	countryRegion	국가 또는 지역 코드	미국
DocumentNumber	string	거주 허가증 번호	WDLABCD456DG
FirstName	string	지정된 이름 및 중간 이니셜(해당하는 경우)	LIAM R.
LastName	string	Surname	TALBOT
DateOfBirth	date	생년월일	01/06/1958
DateOfExpiration	date	만료 날짜	08/12/2020
DateOfIssue	date	이슈 날짜	08/12/2012
Sex	string	성별	M
PlaceOfBirth	string	출생 장소	독일

필드	형식	Description	예시
Category	string	허용 범주	DV2
Address	string	주소	123 STREET ADDRESS YOUR CITY WA 99999-1234

idDocument.usSocialSecurityCard

[+] 테이블 확장

필드	형식	Description	예시
DocumentNumber	string	사회 보장 카드 번호	WDLABCD456DG
FirstName	string	지정된 이름 및 중간 이니셜(해당하는 경우)	LIAM R.
LastName	string	Surname	TALBOT
DateOfIssue	date	이슈 날짜	08/12/2012

idDocument

[+] 테이블 확장

필드	형식	Description	예시
Address	address	주소	123 STREET ADDRESS YOUR CITY WA 99999-1234
DocumentNumber	string	운전 면허증 번호	WDLABCD456DG
FirstName	string	지정된 이름 및 중간 이니셜(해당하는 경우)	LIAM R.
LastName	string	Surname	TALBOT
DateOfBirth	date	생년월일	01/06/1958
DateOfExpiration	date	만료 날짜	08/12/2020

다음 단계

- 문서 인텔리전스 스튜디오 [\[↗\]](#)를 사용하여 사용자 고유의 양식 및 문서를 처리해 보세요.
- Document Intelligence 빠른 시작을 완료하고 원하는 개발 언어로 문서 처리 앱 만들기를 시작해 보세요.

문서 인텔리전스 쿼리 필드 추출

아티클 • 2024. 04. 08.

**문서 인텔리전스는 이제 쿼리 필드를 지원합니다. 이를 통해 미리 빌드된 모델의 스키마를 확장하여 필요한 특정 필드를 추출할 수 있습니다. 쿼리 필드를 레이아웃에 추가하여 양식이나 문서의 구조 외에 필드를 추출할 수도 있습니다.

① 참고

문서 인텔리전스 스튜디오 쿼리 필드 추출은 현재 UX.Tax 미리 빌드된 모델을 제외하고 레이아웃 및 미리 빌드된 모델에서 사용할 수 있습니다.

쿼리 필드 또는 키-값 쌍

쿼리 필드와 키-값 쌍은 수행하는 기능이 비슷합니다. 어떤 기능을 선택할지 결정할 때 알아야 할 몇 가지 차이점이 있습니다.

- 키-값 쌍은 레이아웃 및 청구서 모델에서만 사용할 수 있습니다. 미리 빌드된 모델의 스키마를 확장하려는 경우 쿼리 필드를 사용하세요.
- 추출할 특정 필드를 모르거나 필드 수가 큰 경우(20보다 큼) 키-값 쌍이 더 나은 솔루션일 수 있습니다.
- 키-값 쌍은 양식이나 문서에 있는 키와 값을 추출하므로 키 변형에 대비해야 합니다. 예: 키 `First Name` 또는 `Given Name`. 쿼리 필드의 경우 사용자는 키를 정의하고, 모델은 해당 값만 추출합니다.
- 필요한 값을 문서에서 키-값 쌍으로 설명할 수 없는 경우 쿼리 필드를 사용하세요. 계약의 계약 날짜를 예로 들 수 있습니다.

쿼리 필드 추출을 위해 추출하려는 필드를 지정하면 문서 인텔리전스가 그에 따라 문서를 분석합니다. 예를 들면 다음과 같습니다.

- 문서 인텔리전스 스튜디오 [에서 계약을 처리하는 경우 2024-02-29-preview, 2023-10-31-preview 이상의 API 버전을 사용하세요.](#)

Layout

API version: 2023-10-31 (Preview) ▾

Service resource: fr-preview-demo [Edit](#)

Run analysis

Query fields

Analyze options

Drag & drop file here or
Browse for files or
Fetch from URL

- AnalyzeDocument 요청의 일부로 Party1, Party2, TermsOfUse, PaymentTerms, PaymentDate 및 TermEndDate"와 같은 필드 레이블 목록을 전달할 수 있습니다.

Azure AI | Document Intelligence Studio

○ Azure Form Recognizer is now Azure AI Document Intelligence. [Learn more](#) about the latest updates to the service and the Studio experience.

Document Intelligence Studio > Layout

Layout

API version: 2023-10-31 (Preview) ▾ Service resource: fr-preview-dem

Run analysis

Query fields

Add fields you want to extract to the list and select the ones to be used in Analyze operations. Press Save to update all edits in the list. [Learn more about query fields](#).

Party1

Party2

TermsOfUse

PaymentTerms

PaymentDate

TermEndDate

Maximum of 6/20 fields

I acknowledge that using Query Fields will incur usage to my account. See [pricing](#).

Delete all

Save

Cancel

- 응답에는 쿼리 필드 외에도 모델 출력이 포함됩니다. 각 모델이 추출한 기능 또는 스키마 목록은 [모델 분석 기능](#)을 참조하세요.

쿼리 필드 REST API 요청**

일반 문서 모델과 쿼리 필드 기능을 사용하고 사용자 지정 모델을 학습시킬 필요 없이 추출 프로세스에 필드를 추가합니다.

HTTP

```
POST https://[endpoint]/documentintelligence/documentModels/prebuilt-
layout:analyze?api-version=2024-02-29-
preview&features=queryFields&queryFields=Terms,PaymentDate HTTP/1.1
Host: *.cognitiveservices.azure.com
Content-Type: application/json
Ocp-Apim-Subscription-Key:

{
  "urlSource": "https://raw.githubusercontent.com/Azure-Samples/cognitive-
  services-REST-api-samples/master/curl/form-recognizer/sample-layout.pdf"
}
```

다음 단계

문서 인텔리전스 Studio 빠른 시작을 사용해 보세요.

다른 추가 기능에 대해 알아보기

문서 인텔리전스 사용자 지정 모델

아티클 • 2024. 03. 12.

① 중요

- 문서 인텔리전스 공개 미리 보기 릴리스에서는 현재 개발 중인 기능에 대한 조기 액세스를 제공합니다.
- 기능, 방식 및 프로세스는 GA(일반 공급) 전에 사용자 피드백에 따라 변경될 수 있습니다.
- Document Intelligence 클라이언트 라이브러리의 공개 미리 보기 버전은 기본적으로 REST API 버전 [2024-02-29-preview로 설정됩니다](#).
- 공개 미리 보기 버전 [2024-02-29-preview](#) 는 현재 다음 Azure 지역에서만 사용할 수 있습니다.
 - 미국 동부
 - 미국 서부2
 - 서유럽

이 콘텐츠의 적용 대상은 다음과 같습니다. ✓ v4.0(미리 보기) | 이전 버전: ✓ v3.1(GA) ✓ v3.0(GA) ✓ v2.1(GA)

문서 인텔리전스는 고급 기계 학습 기술을 사용하여 문서를 식별하고, 양식과 문서에서 정보를 탐지하여 추출하고, 추출된 데이터를 정형 JSON 출력으로 반환합니다. 문서 인텔리전스를 사용하면 문서 분석 모델, 미리 빌드/미리 학습된 모델 또는 학습된 독립 실행형 사용자 지정 모델을 사용할 수 있습니다.

이제 사용자 지정 모델에는 추출 모델을 [호출하기 전에 문서 형식을 식별해야 하는 시나리오에 대한 사용자 지정 분류 모델이 포함됩니다](#). 분류자 모델은 [2023-07-31 \(GA\)](#) API부터 사용할 수 있습니다. 분류 모델을 사용자 지정 추출 모델과 페어링하여 비즈니스와 관련된 양식 및 문서에서 필드를 분석하고 추출하여 문서 처리 솔루션을 만들 수 있습니다. 독립 실행형 사용자 지정 추출 모델을 결합하여 [구성된 모델](#)을 만들 수 있습니다.

사용자 지정 문서 모델 형식

사용자 지정 문서 모델은 [사용자 지정 템플릿](#) 또는 사용자 지정 양식과 [사용자 지정 인공 신경망](#) 또는 사용자 지정 문서 모델이라는 두 가지 유형 중 하나일 수 있습니다. 두 모델의 레이블 지정 및 학습 프로세스는 동일하지만 모델은 다음과 같은 차이점이 있습니다.

사용자 지정 추출 모델

사용자 지정 추출 모델을 만들려면 추출하려는 값으로 문서의 데이터 세트에 레이블을 지정하고 레이블이 지정된 데이터 세트를 모델에 학습시킵니다. 시작하려면 동일한 양식 또는 문서 유형의 5개 예제만 있으면 됩니다.

사용자 지정 인공신경망 모델

① 중요

버전 4.0 - 2024-02-29-preview API부터 사용자 지정 신경망 모델은 이제 겹치는 필드와 테이블, 행 및 셀 수준 신뢰도를 지원합니다.

사용자 지정 인공신경망(사용자 지정 문서) 모델은 대규모 문서 컬렉션에 대해 학습된 딥러닝 모델 및 기본 모델을 사용합니다. 그런 다음, 레이블이 지정된 데이터 세트를 사용하여 모델을 학습시킬 때 이 모델을 미세 조정하거나 데이터에 맞게 조정합니다. 사용자 지정 인공신경망 모델은 필드를 추출하기 위해 정형, 반정형 및 비정형 문서를 지원합니다. 사용자 지정 인공신경망 모델은 현재 영어 문서를 지원합니다. 두 모델 유형 중에서 선택할 때 기능 요구 사항을 충족하는지 확인하기 위한 인공신경망 모델로 시작합니다. 사용자 지정 문서 모델에 대한 자세한 내용은 [인공신경망 모델](#)을 참조하세요.

사용자 지정 템플릿 모델

사용자 지정 템플릿 또는 사용자 지정 양식 모델은 일관된 시각적 템플릿을 사용하여 레이블이 지정된 데이터를 추출합니다. 문서의 시각적 구조의 차이는 모델의 정확도에 영향을 미칩니다. 설문지 또는 애플리케이션과 같은 구조화된 양식은 일관된 시각적 템플릿의 예입니다.

학습 집합은 서식 및 레이아웃이 정적이고 문서 인스턴스 간에 일정한 구조화된 문서로 구성됩니다. 사용자 지정 템플릿 모델은 키-값 쌍, 선택 표시, 테이블, 서명 필드 및 지역을 지원합니다. 템플릿 모델에는 [지원되는 언어](#)로 문서를 학습시킬 수 있습니다. 자세한 내용은 [사용자 지정 템플릿 모델](#)을 참조하세요.

문서 및 추출 시나리오의 언어가 사용자 지정 인공신경망 모델을 지원하는 경우 정확도를 높이기 위해 템플릿 모델보다 사용자 지정 신경망 모델을 사용하는 것이 좋습니다.

💡 팁

학습 문서가 일관된 시각적 템플릿을 제공하는지 확인하려면 집합의 각 양식에서 사용자가 입력한 모든 데이터를 제거합니다. 빈 양식이 모양이 동일한 경우 일관된 시각적 템플릿을 나타냅니다.

자세한 내용은 [사용자 지정 모델의 정확도 및 신뢰도 해석 및 개선을 참조하세요.](#)

입력 요구 사항

- 최상의 결과를 위해 문서당 하나의 명확한 사진 또는 고품질 스캔을 제공합니다.
- 지원 파일 형식:

 테이블 확장

모델	PDF	이미지: jpeg/jpg, png, bmp, tiff, heif	Microsoft Office: Word(docx), Excel(xlsx), PowerPoint(pptx)
읽기	✓	✓	✓
레이아웃	✓	✓	✓ (2024-02-29-preview, 2023-10-31-preview 이상)
일반 문서	✓	✓	
사전 제작	✓	✓	
사용자 지정 추출	✓	✓	
사용자 지정 분류	✓	✓	✓

* Microsoft Office 파일은 현재 다른 모델 또는 버전에서 지원되지 않습니다.

- PDF 및 TIFF의 경우 최대 2,000페이지를 처리할 수 있습니다(무료 계층 구독에서는 처음 두 페이지만 처리됨).
- 문서 분석을 위한 파일 크기는 유료(S0) 계층의 경우 500MB, 무료(F0) 계층의 경우 4MB입니다.
- 이미지 크기는 50 x 50픽셀에서 10,000 x 10,000픽셀 사이여야 합니다.
- PDF가 암호로 잠긴 경우에는 제출하기 전에 잠금을 해제해야 합니다.
- 추출할 텍스트의 최소 높이는 1024 x 768 픽셀 이미지의 경우 12픽셀입니다. 이 차원은 인치당 150개 점의 약 8-point 텍스트에 해당합니다.
- 사용자 지정 모델 학습의 경우 학습 데이터의 최대 페이지 수는 사용자 지정 템플릿 모델의 경우 500개, 사용자 지정 인공신경망 모델의 경우 50,000개입니다.

- 사용자 지정 추출 모델 학습의 경우 학습 데이터의 총 크기는 템플릿 모델의 경우 50MB, 신경망 모델의 경우 1G-MB입니다.
- 사용자 지정 분류 모델 학습의 경우 학습 데이터의 총 크기는 최대 10,000페이지인 1GB 입니다.

빌드 모드

사용자 지정 모델 빌드 작업은 템플릿 및 인공신경망 사용자 지정 모델에 대한 지원을 추가합니다. 이전 버전의 REST API 및 클라이언트 라이브러리는 이제 템플릿 모드라고 하는 단일 빌드 모드만 지원했습니다.

- 템플릿 모델은 동일한 기본 페이지 구조가 동일한(시각적 모양이 균일한) 문서 또는 문서 내 요소의 상대 위치가 동일한 문서만 허용합니다.
- 인공신경망 모델은 정보가 동일하지만 페이지 구조가 다른 문서를 지원합니다. 이러한 문서의 예로는 동일한 정보를 공유하지만 회사마다 모양이 다를 수 있는 미국 W2 양식이 있습니다. 인공신경망 모델은 현재 영어 텍스트만 지원합니다.

이 표에서는 GitHub의 빌드 모드 프로그래밍 언어 SDK 참조 및 코드 샘플에 대한 링크를 제공합니다.

[] 테이블 확장

프로그래밍 언어	SDK 참조	코드 샘플
C#/.NET	DocumentBuildMode 구조체	Sample_BuildCustomModelAsync.cs ↗
Java	DocumentBuildMode 클래스	BuildModel.java ↗
JavaScript	DocumentBuildMode 유형	buildModel.js ↗
Python	DocumentBuildMode 열거형	sample_build_model.py ↗

모델 기능 비교

다음 표에서는 사용자 지정 템플릿과 사용자 지정 인공신경망 기능을 비교합니다.

[] 테이블 확장

기능	사용자 지정 템플릿(양식)	사용자 지정 인공신경망(문서)
문서 구조	템플릿, 양식 및 정형	정형, 반정형 및 비정형

기능	사용자 지정 템플릿(양식)	사용자 지정 인공신경망(문서)
학습 시간	1~5분	20분~1시간
데이터 추출	키-값 쌍, 테이블, 선택 표시, 좌표 및 서명	키-값 쌍, 선택 표시 및 테이블
겹치는 필드	지원되지 않음	지원됨
문서 변형	각 변형당 모델 필요	모든 변형에 단일 모델 사용
언어 지원	다중 언어 지원	영어(스페인어, 프랑스어, 독일어, 이탈리아어 및 네덜란드 어 지원 미리 보기 지원)

사용자 지정 분류 모델

문서 분류는 문서 인텔리전스 2023-07-31 (v3.1 GA) API에서 지원하는 새로운 시나리오입니다. 문서 분류자 API는 분류 및 분할 시나리오를 지원합니다. 분류 모델을 학습하여 애플리케이션이 지원하는 다양한 유형의 문서를 식별합니다. 분류 모델에 대한 입력 파일은 여러 문서를 포함할 수 있으며 연결된 페이지 범위 내에서 각 문서를 분류합니다. 자세한 내용은 사용자 지정 분류 모델을 참조하세요.

① 참고

2024-02-29-preview 이제 API 버전 문서 분류부터 Office 문서 유형 분류가 지원됩니다. 이 API 버전에서는 분류 모델에 대한 충분한 학습 도입했습니다.

사용자 지정 모델 도구

문서 인텔리전스 v3.1 이상 모델은 다음 도구, 애플리케이션, 프로그램 및 라이브러리를 지원합니다.

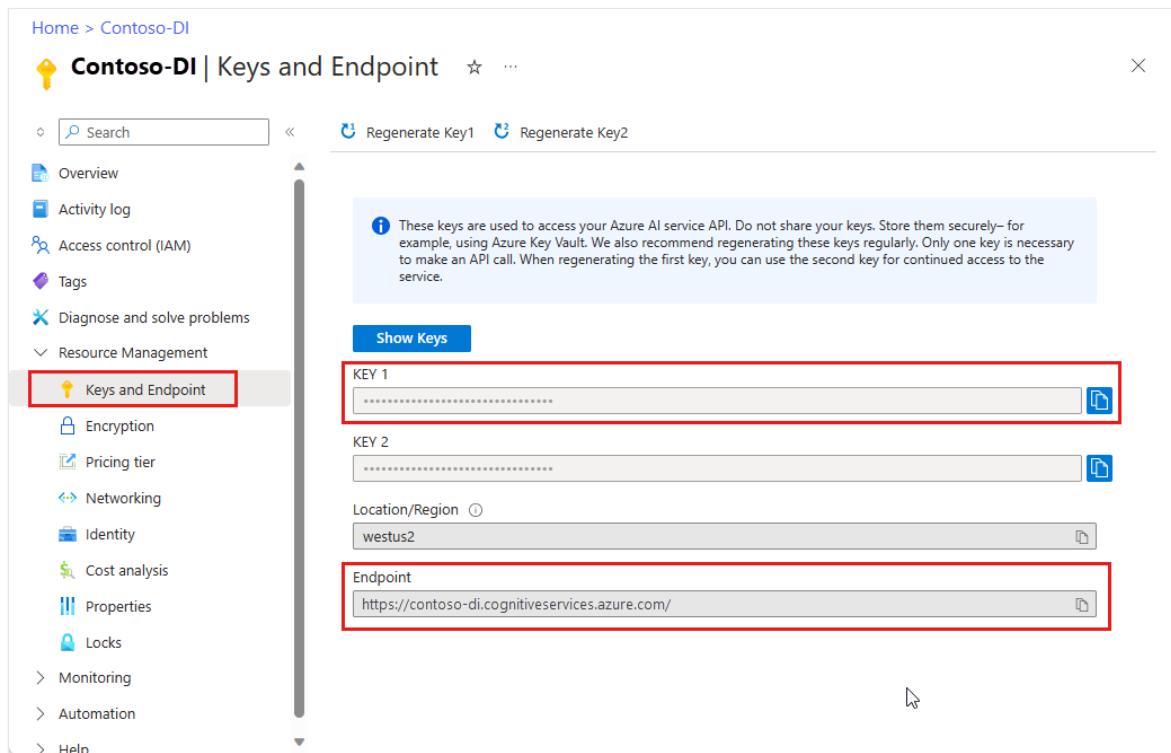
데이터 확장

기능	리소스	Model ID
사용자 지정 모델	<ul style="list-style-type: none">문서 인텔리전스 스튜디오REST APIC# SDKPython SDK	custom-model-id

사용자 지정 모델 빌드

사용자 지정 모델을 사용하여 특정 또는 고유한 문서에서 데이터를 추출합니다. 다음 리소스가 필요합니다.

- Azure 구독 [무료로 하나를 만들 수 있습니다](#).
- Azure Portal의 [Document Intelligence 인스턴스](#)입니다. 무료 가격 책정 계층(F0)을 사용하여 서비스를 시도할 수 있습니다. 리소스가 배포된 후 **리소스로 이동**을 선택하여 키 및 엔드포인트를 가져옵니다.



문서 인텔리전스 스튜디오

① 참고

문서 인텔리전스 스튜디오는 v3.1 및 v3.0 API와 함께 사용할 수 있습니다.

1. **문서 인텔리전스 스튜디오** 홈페이지에서 **사용자 지정 추출 모델**을 선택합니다.
2. **내 프로젝트** 아래에서 **프로젝트 만들기**를 선택합니다.
3. 프로젝트 세부 정보 필드를 완료합니다.
4. **스토리지 계정 및 Blob 컨테이너**를 추가하여 학습 데이터 원본을 연결하는 방식으로 서비스 리소스를 구성합니다.

5. 프로젝트를 검토하고 만듭니다.

6. 샘플 문서를 추가하여 사용자 지정 모델에 레이블을 지정하고, 빌드하고, 테스트합니다.

Document Intelligence Studio 사용해 보기

첫 번째 사용자 지정 추출 모델을 만드는 자세한 연습은 사용자 지정 추출 모델을 [만드는 방법을 참조하세요](#).

사용자 지정 모델 추출 요약

이 표에서는 지원되는 데이터 추출 영역을 비교합니다.

[+] 테이블 확장

모델	양식 필드	선택 표시	정형 필드(레이블)	서명	지역 레이블 지정	겹치는 필드
사용자 지정 템플릿	✓	✓	✓	✓	✓	n/a
사용자 지정 인공신경망	✓	✓	✓	n/a	*	✓ (2024-02-29-preview)

테이블 기호:

✓ -지원됨

**n/a- 현재 사용할 수 없음;

*-모델에 따라 다르게 동작합니다. 템플릿 모델을 사용하면 학습 시 합성 데이터가 생성됩니다. 인공신경망 모델을 사용하면 지역에서 인식된 종료 텍스트가 선택됩니다.

💡 팁

두 모델 유형 중에서 선택할 때 기능 요구 사항을 충족하는 경우 사용자 지정 인공신경망 모델로 시작합니다. 사용자 지정 인공신경망 모델에 대한 자세한 내용은 [사용자 지정 인공신경망](#)을 참조하세요.

사용자 지정 모델 개발 옵션

다음 표에서는 연결된 도구 및 클라이언트 라이브러리에서 사용할 수 있는 기능에 대해 설명합니다. 모범 사례는 여기에 나열된 호환되는 도구를 사용하는 것입니다.

Document type	REST API	SDK	모델 레이블 지정 및 테스트
사용자 지정 템플릿 v 4.0 v3.1 v3.0	Document Intelligence 3.1	Document Intelligence SDK	Document Intelligence Studio ↗
사용자 지정 인공신경망 v4.0 v3.1 v3.0	Document Intelligence 3.1	Document Intelligence SDK	Document Intelligence Studio ↗
사용자 지정 양식 v2.1	문서 인텔리전스 2.1 GA API ↗	문서 인텔리전스 SDK	샘플 레이블 지정 도구 ↗

① 참고

3.0 API를 사용하여 학습된 사용자 지정 템플릿 모델은 OCR 엔진의 향상된 기능으로 인해 2.1 API보다 몇 가지 개선 사항이 있습니다. 2.1 API를 사용하여 사용자 지정 템플릿 모델을 학습시키는 데 사용된 데이터 세트는 여전히 3.0 API를 사용하여 새 모델을 학습시키는 데 사용할 수 있습니다.

- 최상의 결과를 위해 문서당 하나의 명확한 사진 또는 고품질 스캔을 제공합니다.
- 지원되는 파일 형식은 JPEG/JPG, PNG, BMP, TIFF 및 PDF(텍스트 포함 또는 검사)입니다. 텍스트 포함 PDF는 문자 추출 및 위치에 오류가 발생할 가능성을 제거하는 데 가장 좋습니다.
- PDF 및 TIFF 파일의 경우 최대 2,000페이지를 처리할 수 있습니다. 무료 계층 구독이 있는 경우 처음 두 페이지만 처리됩니다.
- 파일 크기는 유료(S0) 계층의 경우 500MB 미만, 무료(F0) 계층의 경우 4MB 미만이어야 합니다.
- 이미지 차원은 50 x 50 픽셀에서 10,000 x 10,000 픽셀 사이여야 합니다.
- PDF 크기는 Legal 또는 A3 용지 크기 이하에 해당하는 최대 17 x 17인치입니다.
- 학습 데이터 세트의 총 크기는 500페이지 이하입니다.
- PDF가 암호로 잠긴 경우에는 제출하기 전에 잠금을 해제해야 합니다.

💡 팁

학습 데이터:

- 가능하면 이미지 기반 문서 대신 텍스트 기반 PDF 문서를 사용합니다. 스캔한 PDF는 이미지로 처리됩니다.
- 문서당 하나의 양식 인스턴스만 제공합니다.
- 채워진 양식의 경우 모든 필드가 채워진 예제를 사용합니다.
- 각 필드에 서로 다른 값이 있는 양식을 사용합니다.
- 양식 이미지의 품질이 낮은 경우 더 큰 데이터 세트를 사용합니다. 예를 들어 10~15개의 이미지를 사용합니다.

지원되는 언어 및 로캘

[지원되는 모든 언어 목록은 언어 지원 - 사용자 지정 모델 페이지](#)를 참조하세요.

다음 단계

- Document Intelligence Studio를 사용하여 사용자 고유의 [양식 및 문서를 처리해보세요.](#)
- [Document Intelligence 빠른 시작](#)을 완료하고 원하는 개발 언어로 문서 처리 앱 만들기를 시작해 보세요.

문서 인텔리전스 사용자 지정 모델 수명 주기

아티클 • 2024. 03. 06.

이 콘텐츠의 적용 대상은 다음과 같습니다. ✓ v4.0(미리 보기) ✓ v3.1(GA)

v3.1(GA) 이상 API를 사용하면 사용자 지정 모델에 3.1 API 이상으로 학습된 각 모델에 대해 설정되는 expirationDateTime 속성이 도입됩니다. 사용자 지정 모델은 Layout API 버전의 API 버전과 모델 빌드 작업의 API 버전에 따라 달라집니다. 최상의 결과를 얻으려면 모든 분석 요청에 대해 모델이 학습된 API 버전을 계속 사용하세요. 이 지침은 추출 및 분류 모델을 포함한 모든 문서 인텔리전스 사용자 지정 모델에 적용됩니다.

GA API 버전으로 학습된 모델

v3.1 API를 사용하면 사용자 지정 모델에 새로운 모델 만료 속성이 도입됩니다. 모델 만료는 GA API를 사용하여 모델을 빌드하는 모든 요청에 대해 모델이 빌드된 날짜로부터 2년으로 설정됩니다. 만료 날짜가 지난 후에도 모델을 계속 사용하려면 현재 GA API 버전으로 모델을 학습해야 합니다.

미리 보기 API 버전으로 학습된 모델

미리 보기 API 버전을 사용하는 빌드 요청의 경우 만료 날짜는 모델이 빌드된 날짜로부터 3개월로 설정됩니다. 미리 보기 API로 학습된 모델은 프로덕션에서 사용해서는 안 되며 해당 GA API 버전을 사용할 수 있게 되면 다시 학습해야 합니다. 미리 보기 API 버전과 GA API 버전 간의 호환성이 항상 유지되는 것은 아닙니다. 미리 보기 API가 사용 중지되면 미리 보기 API 버전으로 학습된 모델을 사용할 수 없게 됩니다.

Preview API는 일반적으로 해당 GA API를 사용할 수 있게 된 후 3개월 이내에 만료됩니다.

모델 만료 날짜 보기

GET 모델 API는 `expirationDateTime` 속성을 포함한 모델 세부정보를 반환합니다.

rest

```
GET /documentModels/{customModelId}?api-version={apiVersion}
{
    "modelId": "{customModelId}",
    "description": "{customModelDescription}",
    "createdDateTime": "2021-09-24T12:54:35Z",
```

```
"expirationDateTime": "2023-01-01T00:00:00Z",
"apiVersion": "2023-07-31",
"docTypes": { ... }
}
```

모델 재교육

최신 API 버전으로 모델을 다시 학습하려면 학습 데이터 세트에 있는 문서의 레이아웃 결과가 모델 빌드 요청의 API 버전과 일치하는지 확인하세요. 예를 들어, v3.1:2023-07-31 API 버전으로 모델을 빌드하려는 경우 학습 데이터세트의 해당 *.ocr.json 파일도 v3.1:2023-07-31 API 버전으로 생성되어야 합니다. ocr.json 파일은 학습 데이터 세트에서 레이아웃을 실행하여 생성됩니다. 레이아웃 결과 버전의 유효성을 검사하려면 ocr.json 문서의 `analyzeResult`에서 `apiVersion` 속성을 확인하세요.

다음 단계

사용자 지정 모델을 만들고 구성하는 방법을 알아봅니다.

[사용자 지정 모델 빌드](#)

[사용자 지정 모델 작성](#)

Document Intelligence 사용자 지정 템플릿 모델

아티클 • 2024. 03. 10.

ⓘ 중요

- 문서 인텔리전스 공개 미리 보기 릴리스에서는 현재 개발 중인 기능에 대한 초기 액세스를 제공합니다.
- 기능, 방식 및 프로세스는 GA(일반 공급) 전에 사용자 피드백에 따라 변경될 수 있습니다.
- Document Intelligence 클라이언트 라이브러리의 공개 미리 보기 버전은 기본적으로 REST API 버전 [2024-02-29-preview로 설정됩니다](#).
- 공개 미리 보기 버전 [2024-02-29-preview](#) 는 현재 다음 Azure 지역에서만 사용할 수 있습니다.
 - 미국 동부
 - 미국 서부2
 - 서유럽

이 콘텐츠의 적용 대상은 다음과 같습니다. v4.0(미리 보기) | 이전 버전: v3.1(GA) v3.0(GA) v2.1(GA)

사용자 지정 템플릿(이전의 사용자 지정 양식)은 문서에서 레이블이 지정된 키-값 쌍, 선택 표시, 테이블, 지역, 서명을 정확하게 추출하는 학습하기 쉬운 문서 모델입니다. 템플릿 모델은 레이아웃 신호를 사용하여 문서에서 값을 추출하고 정의된 시각적 템플릿을 사용하여 고도로 구조화된 문서에서 필드를 추출하는 데 적합합니다.

사용자 지정 템플릿 모델은 사용자 지정 신경망 모델과 동일한 레이블 지정 형식 및 전략을 공유하며 더 많은 필드 형식 및 언어를 지원합니다.

모델 기능

사용자 지정 템플릿 모델은 키-값 쌍, 선택 표시, 테이블, 서명 필드, 선택된 지역을 지원합니다.

테이블 확장

양식 필드	선택 표시	테이블 형식 필드(테이블)	서명	선택 영역
지원됨	지원	지원	지원	지원됨

표 형식 필드

API 버전 2022-06-30-preview 이상의 릴리스에서 사용자 지정 템플릿 모델은 페이지 간 테이블 형식 필드(테이블)에 대한 지원을 추가합니다.

- 여러 페이지에 걸쳐 있는 테이블에 레이블을 지정하려면 단일 테이블의 여러 페이지에 걸쳐 테이블의 각 행에 레이블을 지정합니다.
- 가장 좋은 방법은 데이터 세트에 예상되는 변형의 몇 가지 샘플이 포함되어 있는지 확인하는 것입니다. 예를 들어 전체 테이블이 단일 페이지에 있고 문서에서 이러한 변형이 예상되는 경우 테이블이 두 개 이상의 페이지에 걸쳐 있는 샘플을 포함합니다.

표 형식 필드는 표로 인식되지 않는 문서 내에서 반복되는 정보를 추출할 때도 유용합니다. 예를 들어 이력서에서 반복되는 업무 환경 섹션에 레이블을 지정하고 테이블 형식 필드로 추출할 수 있습니다.

변형 처리

템플릿 모델은 정의된 시각적 템플릿을 사용하며 템플릿을 변경하면 정확도가 낮아질 수 있습니다. 이러한 인스턴스에서 각 템플릿의 샘플을 5개 이상 포함하도록 학습 데이터 세트를 분할하고 각 변형에 대한 모델을 학습시킵니다. 그런 다음, 모델을 단일 엔드포인트로 구성할 수 있습니다. 디지털 PDF 문서 및 이미지와 같은 미묘한 변형의 경우 동일한 학습 데이터 세트에 각 형식의 예제를 5개 이상 포함하는 것이 가장 좋습니다.

입력 요구 사항

- 최상의 결과를 위해 문서당 하나의 명확한 사진 또는 고품질 스캔을 제공합니다.
- 지원 파일 형식:

 테이블 확장

모델	PDF	이미지: JPEG/JPG, PNG, BMP, TIFF, HEIF	Microsoft Office: Word(DOCX), Excel(XLSX), PowerPoint(PPTX) 및 HTML
읽음	✓	✓	✓

모델	PDF	이미지: JPEG/JPG, PNG, BMP, TIFF, HEIF	Microsoft Office: Word(DOCX), Excel(XLSX), PowerPoint(PPTX) 및 HTML
레이아웃	✓	✓	✓ (2024-02-29-preview, 2023-10-31-preview 이상)
일반 문서	✓	✓	
사전 제작	✓	✓	
사용자 지정	✓	✓	

* Microsoft Office 파일은 현재 다른 모델 또는 버전에서 지원되지 않습니다.

- PDF 및 TIFF의 경우 최대 2000페이지를 처리할 수 있습니다(무료 계층 구독의 경우 처음 2페이지만 처리됨).
- 문서 분석을 위한 파일 크기는 유료(S0) 계층의 경우 500MB, 무료(F0) 계층의 경우 4MB입니다.
- 이미지 크기는 50 x 50픽셀에서 10,000 x 10,000픽셀 사이여야 합니다.
- PDF가 암호로 잠긴 경우에는 제출하기 전에 잠금을 해제해야 합니다.
- 추출할 텍스트의 최소 높이는 1024 x 768 픽셀 이미지의 경우 12픽셀입니다. 이 차원은 150DPI(인치당 도트 수)에서 약 8포인트 텍스트에 해당합니다.
- 사용자 지정 모델 학습의 경우 학습 데이터의 최대 페이지 수는 사용자 지정 템플릿 모델의 경우 500개, 사용자 지정 인공신경망 모델의 경우 50,000개입니다.
- 사용자 지정 추출 모델 학습의 경우 학습 데이터의 총 크기는 템플릿 모델의 경우 50MB, 신경망 모델의 경우 1G-MB입니다.
- 사용자 지정 분류 모델 학습의 경우 학습 데이터의 총 크기는 최대 10,000페이지인 1GB입니다.

모델 학습

사용자 지정(템플릿) 모델은 일반적으로 [v4.0 API](#)에서 사용할 수 있습니다. 새 프로젝트로 시작하거나 레이블이 지정된 기준 데이터 세트가 있는 경우 Document Intelligence Studio와 함께 v3.1 또는 v3.0 API를 사용하여 사용자 지정 템플릿 모델을 학습합니다.

모델	REST API	SDK	모델 레이블 지정 및 테스트
사용자 지정 템플릿	v3.1 API	Document Intelligence SDK	Document Intelligence Studio ↗

v3.0 이상 API를 사용하면 모델은 학습시키는 빌드 작업이 새 `buildMode` 속성을 지원하고, 사용자 지정 템플릿 모델을 학습하기 위해 `buildMode`를 `template`으로 설정합니다.

REST

`https://{{endpoint}}/documentintelligence/documentModels:build?api-version=2024-02-29-preview`

```
{  
    "modelId": "string",  
    "description": "string",  
    "buildMode": "template",  
    "azureBlobSource":  
    {  
        "containerUrl": "string",  
        "prefix": "string"  
    }  
}
```

지원되는 언어 및 로캘

지원되는 언어의 전체 목록은 [언어 지원 - 사용자 지정 모델 페이지](#)를 참조하세요.

다음 단계

사용자 지정 모델을 만들고 구성하는 방법을 알아봅니다.

[사용자 지정 모델 빌드](#)[사용자 지정 모델 작성](#)

문서 인텔리전스 사용자 지정 신경망 모델

아티클 • 2024. 04. 14.

① 중요

- 문서 인텔리전스 공개 미리 보기 릴리스에서는 현재 개발 중인 기능에 대한 조기 액세스를 제공합니다.
- 기능, 방식 및 프로세스는 GA(일반 공급) 전에 사용자 피드백에 따라 변경될 수 있습니다.
- 문서 인텔리전스 클라이언트 라이브러리의 공개 미리 보기 버전은 기본적으로 REST API 버전 [2024-02-29-preview](#)로 설정됩니다.
- 공개 미리 보기 버전 [2024-02-29-preview](#)는 현재 다음 Azure 지역에서만 사용 할 수 있습니다.
 - 미국 동부
 - 미국 서부2
 - 서유럽

이 콘텐츠의 적용 대상은 다음과 같습니다. ✓ v4.0(미리 보기) | 이전 버전: ✓ v3.1(GA) ✓ v3.0(GA)

사용자 지정 신경망 문서 모델 또는 신경망 모델은 레이아웃 및 언어 기능을 결합하여 문서에서 레이블이 지정된 필드를 정확하게 추출하는 딥 러닝된 모델 유형입니다. 기본 사용자 지정 신경망 모델은 구조화된 문서, 반구조화된 문서, 구조화되지 않은 문서에서 필드를 추출하는 데 적합한 다양한 문서 형식에 대해 학습됩니다. 사용자 지정 신경망 모델은 [v3.0 이상 모델](#)에서 사용할 수 있습니다. 아래 표에는 각 범주의 일반적인 문서 유형이 나와 있습니다.

[+] 테이블 확장

문서	예제
구조적	설문 조사, 설문지
반구조적	청구서, 구매 주문
비구조적	계약, 편지

사용자 지정 신경망 모델은 [사용자 지정 템플릿](#) 모델과 동일한 레이블 지정 형식 및 전략을 공유합니다. 현재 사용자 지정 신경망 모델은 사용자 지정 템플릿 모델에서 지원하는 필드 형식의 하위 집합만 지원합니다.

모델 기능

① 중요

API 버전 2024-02-29-preview부터 사용자 지정 신경망 모델은 겹치는 필드 및 테이블 셀 신뢰도에 대한 지원을 추가합니다.

사용자 지정 인공신경망 모델은 현재 키-값 쌍 및 선택 표시 및 구조화된 필드(테이블)만 지원하며, 향후 릴리스에는 서명에 대한 지원이 포함됩니다.

테이블 확장

양식 필드	선택 표시	표 형식 필드	서명	지역	겹치는 필드
지원됨	지원	지원됨	지원되지 않음	지원됨 ¹	지원됨 ²

¹ 사용자 지정 인공신경망 모델의 지역 레이블은 지정된 지역에 대한 레이아웃 API의 결과를 사용합니다. 이 기능은 값이 없으면 학습 시 텍스트가 생성되는 템플릿 모델과 다릅니다. ² 겹치는 필드는 REST API 버전 2024-02-29-preview부터 지원됩니다. 겹치는 필드에는 몇 가지 제한이 있습니다. 자세한 내용은 [겹치는 필드](#)를 참조하세요.

빌드 모드

사용자 지정 모델 빌드 작업은 '템플릿' 및 '인공신경망' 사용자 지정 모델을 지원합니다. 이전 버전의 REST API 및 클라이언트 라이브러리는 현재 템플릿 모드라고 알려진 단일 빌드 모드만 지원했습니다.

인공신경망 모델은 정보가 동일하지만 페이지 구조가 다른 문서를 지원합니다. 이러한 문서의 예로는 동일한 정보를 공유하지만 회사마다 모양이 다를 수 있는 미국 W2 양식이 있습니다. 자세한 내용은 사용자 지정 모델 빌드 모드를 [참조하세요](#).

지원되는 언어 및 로캘

지원되는 모든 언어 목록은 [언어 지원 - 사용자 지정 모델](#) 페이지를 참조하세요.

겹치는 필드

API 버전 2024-02-29-preview 이상 릴리스와 함께 사용자 지정 신경망 모델은 겹치는 필드를 지원합니다.

겹치는 필드를 사용하려면 데이터 세트에 예상되는 겹침이 있는 샘플을 하나 이상 포함해야 합니다. 겹침에 레이블을 지정하려면 **영역 레이블 지정**을 사용하여 각 필드에 대한 콘텐츠 범위(겹침 포함)를 각각 지정합니다. 필드 선택 영역이 있는 겹침 레이블 지정(값 강조 표시)은 필드 겹침을 나타내는 유일한 지원되는 레이블 지정 도구이므로 스튜디오에서 실패합니다. 겹침 지원에는 다음이 포함됩니다.

- 전체 겹침. 동일한 토큰 집합은 서로 다른 두 필드에 대해 레이블이 지정됩니다.
- 부분 겹침. 일부 토큰은 두 필드에 모두 속하지만 한 필드 또는 다른 필드의 일부인 토큰만 있습니다.

겹치는 필드에는 몇 가지 제한이 있습니다.

- 모든 토큰 또는 단어는 두 개의 필드로만 레이블을 지정할 수 있습니다.
- 테이블의 겹치는 필드는 테이블 행에 걸쳐 있을 수 없습니다.
- 데이터 세트의 하나 이상의 샘플에 해당 필드에 대한 겹치는 레이블이 포함된 경우에만 겹치는 필드를 인식할 수 있습니다.

겹치는 필드를 사용하려면 데이터 세트에 겹침 레이블을 지정하고 API 버전 `2024-02-29-preview` 이상을 사용하여 모델을 학습시킵니다.

테이블 형식 필드는 표, 행 및 셀 신뢰도를 추가합니다.

API 버전 `2022-06-30-preview` 이상 릴리스와 함께 사용자 지정 신경망 모델은 표 형식 필드(테이블)를 지원합니다.

- API 버전 2022-08-31 이상으로 학습된 모델은 표 형식 필드 레이블을 허용합니다.
- API 버전 2022-06-30-preview 이상을 사용하여 사용자 지정 신경망 모델로 분석된 문서는 테이블 전체에서 집계된 표 형식 필드를 생성합니다.
- 결과는 분석 작업 후에 반환되는 `analyzeResult` 객체의 `documents` 배열에서 찾을 수 있습니다.

표 형식 필드는 기본적으로 **크로스 페이지 테이블**을 지원합니다.

- 여러 페이지에 걸쳐 있는 테이블에 레이블을 지정하려면 단일 테이블의 여러 페이지에 걸쳐 테이블의 각 행에 레이블을 지정합니다.
- 가장 좋은 방법은 데이터 세트에 예상되는 변형의 몇 가지 샘플이 포함되어 있는지 확인하는 것입니다. 예를 들어, 전체 테이블이 단일 페이지에 있고 테이블이 둘 이상의 페이지에 걸쳐 있는 샘플을 포함합니다.

표 형식 필드는 표로 인식되지 않는 문서 내에서 반복되는 정보를 추출할 때도 유용합니다. 예를 들어 이력서에서 반복되는 업무 환경 섹션에 레이블을 지정하고 테이블 형식 필

드로 추출할 수 있습니다.

테이블 형식 필드는 2024-02-29-preview API로 시작하는 **테이블, 행 및 셀 신뢰도**를 제공합니다.

- 고정 또는 동적 테이블은 다음 요소에 대한 신뢰도 지원을 추가합니다.
 - 테이블 신뢰도는 전체 테이블이 얼마나 정확하게 인식되는지에 대한 측정값입니다.
 - 행 신뢰도, 개별 행 인식의 측정값입니다.
 - 셀 신뢰도, 개별 셀 인식의 측정값입니다.
- 권장되는 방법은 먼저 테이블부터 시작하여, 행 다음으로 셀로 하향식으로 정확도를 검토하는 것입니다.

표, 행 및 셀 신뢰도에 대한 자세한 내용은 [신뢰도 및 정확도 점수](#)를 참조하세요.

지원되는 지역

2022년 10월 18일부터 문서 인텔리전스 사용자 지정 인공신경망 모델 학습은 추후 공지가 있을 때까지 다음 Azure 지역에서만 사용할 수 있습니다.

- 오스트레일리아 동부
- 브라질 남부
- 캐나다 중부
- 인도 중부
- 미국 중부
- 동아시아
- 미국 동부
- 미국 동부2
- 프랑스 중부
- 일본 동부
- 미국 중남부
- 동남아시아
- 영국 남부
- 서유럽
- 미국 서부2
- US Gov 애리조나
- US Gov 버지니아

💡 **팁**

나열된 일부 지역 중 하나에서 학습된 모델을 다른 지역에 **복사**하여 적절하게 사용 할 수 있습니다.

REST API 또는 [문서 인텔리전스 스튜디오](#)를 사용하여 모델을 다른 지역으로 복사 합니다.

입력 요구 사항

- 최상의 결과를 위해 문서당 하나의 명확한 사진 또는 고품질 스캔을 제공합니다.
- 지원 파일 형식:

[+] 테이블 확장

모델	PDF	이미지: jpeg/jpg, png, bmp, tiff, heif	Microsoft Office: Word(docx), Excel(xlsx), PowerPoint(pptx) 및 HTML
읽음	✓	✓	✓
레이아웃	✓	✓	✓ (2024-02-29-preview, 2023-10-31-preview 이상)
일반 문서	✓	✓	
사전 제작	✓	✓	
사용자 지정 인 공신경망	✓	✓	

* Microsoft Office 파일은 현재 다른 모델 또는 버전에서 지원되지 않습니다.

- PDF 및 TIFF의 경우 최대 2,000페이지를 처리할 수 있습니다(무료 계층 구독의 경우 처음 2페이지만 처리됨).
- 문서 분석을 위한 파일 크기는 유료(S0) 계층의 경우 500MB, 무료(F0) 계층의 경우 4MB입니다.
- 이미지 크기는 50 x 50픽셀에서 10,000 x 10,000픽셀 사이여야 합니다.
- PDF가 암호로 잠긴 경우에는 제출하기 전에 잠금을 해제해야 합니다.
- 추출할 텍스트의 최소 높이는 1024 x 768 픽셀 이미지의 경우 12픽셀입니다. 이 차 원은 150 인치당 도트 수에서 약 8포인트 텍스트에 해당합니다.

- 사용자 지정 모델 학습의 경우 학습 데이터의 최대 페이지 수는 사용자 지정 템플릿 모델의 경우 500개, 사용자 지정 인공신경망 모델의 경우 50,000개입니다.
- 사용자 지정 추출 모델 학습의 경우 학습 데이터의 총 크기는 템플릿 모델의 경우 50MB, 신경망 모델의 경우 1G-MB입니다.
- 사용자 지정 분류 모델 학습의 경우 학습 데이터의 총 크기는 최대 10,000페이지인 1GB입니다.

모범 사례

사용자 지정 신경망 모델은 몇 가지 다른 방법으로 사용자 지정 템플릿 모델과 다릅니다. 사용자 지정 템플릿 또는 모델은 일관된 시각적 템플릿을 사용하여 레이블이 지정된 데이터를 추출합니다. 사용자 지정 인공신경망 모델은 필드를 추출하기 위해 정형, 반정형 및 비정형 문서를 지원합니다. 두 가지 모델 형식 중에서 선택할 때 신경망 모델로 시작하여 기능적 요구 사항을 지원하는지 확인하기 위해 테스트합니다.

변형 처리

사용자 지정 신경망 모델은 단일 문서 형식의 다양한 형식에서 일반화할 수 있습니다. 문서 형식의 모든 변형에 대해 단일 모델을 만드는 것이 좋습니다. 학습 데이터 세트에 서로 다른 각 변형에 대해 레이블이 지정된 샘플을 5개 이상 추가합니다.

필드 이름 지정

데이터에 레이블을 지정할 때 값과 관련된 필드에 레이블을 지정하면 추출된 키-값 쌍의 정확도가 향상됩니다. 예를 들어 공급업체 ID를 포함하는 필드 값의 경우 필드 이름을 `supplier_id`로 지정하는 것이 좋습니다. 필드 이름은 문서의 언어여야 합니다.

연속 값 레이블 지정

값 토큰/한 필드의 단어 중 하나여야 합니다.

- 다른 필드와 인터리빙하지 않고 자연 읽기 순서로 연속 시퀀스로
- 다른 필드를 포함하지 않는 지역에서

대표 데이터

학습 사례의 값은 다양하고 대표적이어야 합니다. 예를 들어 필드 이름이 날짜인 경우 이 필드의 값은 날짜여야 합니다. 임의 문자열과 같은 가상 값은 모델 성능에 영향을 줄 수 있습니다.

현재 제한 사항

- 사용자 지정 신경망 모델은 페이지 경계를 넘어 분할된 값을 인식하지 않습니다.
- 사용자 지정 템플릿 모델에 레이블이 지정된 데이터 세트를 사용하여 사용자 지정 신경망 모델을 학습시키는 경우 지원되지 않는 사용자 지정 신경망 필드 형식은 무시됩니다.
- 사용자 지정 신경망 모델은 매월 빌드 작업 20번으로 제한됩니다. 제한을 늘려야 하는 경우 지원 요청을 업니다. 자세한 내용은 [문서 인텔리전스 서비스 할당량 및 제한](#)을 참조하세요.

모델 학습

사용자 지정 신경망 모델은 [v3.0 이상 모델](#)에서 사용할 수 있습니다.

[+] 테이블 확장

문서 형식	REST API	SDK	모델 레이블 지정 및 테스트
사용자 지정 문서	Document Intelligence 3.1	Document Intelligence SDK	Document Intelligence Studio

모델을 학습시키는 빌드 작업은 새 `buildMode` 속성을 지원하고, 사용자 지정 신경망 모델을 학습하고, `buildMode`를 `neural`로 설정합니다.

```
https://{{endpoint}}/documentintelligence/documentModels:build?api-version=2024-02-29-preview
```

```
{
  "modelId": "string",
  "description": "string",
  "buildMode": "neural",
  "azureBlobSource": {
    "containerUrl": "string",
    "prefix": "string"
  }
}
```

다음 단계

사용자 지정 모델을 만들고 구성하는 방법을 알아봅니다.

[사용자 지정 모델 빌드](#)

[사용자 지정 모델 작성](#)

문서 인텔리전스 사용자 지정 분류 모델

아티클 • 2024. 03. 10.

① 중요

- 문서 인텔리전스 공개 미리 보기 릴리스에서는 현재 개발 중인 기능에 대한 조기 액세스를 제공합니다.
- 기능, 방식 및 프로세스는 GA(일반 공급) 전에 사용자 피드백에 따라 변경될 수 있습니다.
- Document Intelligence 클라이언트 라이브러리의 공개 미리 보기 버전은 기본적으로 REST API 버전 [2024-02-29-preview로 설정됩니다](#).
- 공개 미리 보기 버전 [2024-02-29-preview](#) 는 현재 다음 Azure 지역에서만 사용할 수 있습니다.
 - 미국 동부
 - 미국 서부2
 - 서유럽

이 콘텐츠의 적용 대상은 다음과 같습니다. v4.0(미리 보기) | 이전 버전: v3.1(GA)

① 중요

- 2024-02-29-preview API, 사용자 지정 분류 모델은 분석 프로세스 중에 기본적으로 문서를 분할하지 않습니다.
- 이전 릴리스의 동작을 유지하려면 명시적으로 `splitMode` 속성을 자동으로 설정해야 합니다. `splitMode`에 대한 기본값은 `none`입니다.
- 입력 파일에 여러 문서가 포함된 경우 `splitMode`를 `auto`로 설정하여 분할을 사용하도록 설정해야 합니다.

Azure AI 문서 인텔리전스는 지능형 문서 처리 솔루션을 빌드할 수 있는 클라우드 기반 Azure AI 서비스입니다. 문서 인텔리전스 API는 이미지, PDF 및 기타 문서 파일을 분석하여 다양한 콘텐츠, 레이아웃, 스타일 및 의미 체계 요소를 추출하고 검색합니다.

사용자 지정 분류 모델은 레이아웃 및 언어 기능을 결합하여 애플리케이션 내에서 처리하는 문서를 정확하게 검색하고 식별하는 딥 러닝 모델 형식입니다. 사용자 지정 분류 모델은 입력 파일의 분류를 한 번에 한 페이지씩 수행하여 문서를 식별하며 입력 파일 내에서 여러 문서 또는 단일 문서의 여러 인스턴스를 식별할 수도 있습니다.

모델 기능

① 참고

- 2024-02-29-preview API부터 사용자 지정 설명 모델은 충분 학습을 지원합니다. 기존 클래스에 새 샘플을 추가하거나 기존 분류자를 참조하여 새 클래스를 추가할 수 있습니다.

사용자 지정 분류 모델은 단일 또는 여러 파일 문서를 분석하여 학습된 문서 형식이 입력 파일 내에 포함되어 있는지 여부를 식별할 수 있습니다. 현재 지원되는 시나리오는 다음과 같습니다.

- 하나의 문서를 포함하는 단일 파일. 예를 들어 대출 신청 양식이 있습니다.
- 여러 문서를 포함하는 단일 파일. 예를 들어 대출 신청 양식, 급여 명세서 및 은행 명세서를 포함하는 대출 신청 패키지가 있습니다.
- 동일한 문서의 여러 인스턴스를 포함하는 단일 파일. 예를 들어 스캔한 청구서의 컬렉션이 있습니다.

✓ 사용자 지정 분류자를 학습시키려면 `two` 이상의 고유 클래스와 클래스당 최소 `five` 문서 샘플이 필요합니다. 모델 응답에는 식별된 문서의 각 클래스에 대한 페이지 범위가 포함됩니다.

✓ 허용되는 최대 클래스 수는 `500`입니다. 클래스당 허용되는 최대 문서 샘플 수는 `100`입니다.

모델은 입력 문서의 각 페이지를 레이블이 지정된 데이터 세트의 클래스 중 하나로 분류합니다. 애플리케이션에 대한 임계값을 설정하려면 응답의 신뢰도 점수를 사용합니다.

증분 학습

사용자 지정 모델을 사용하면 학습 데이터 세트에 대한 액세스를 기본 기존 클래스에 대한 새 샘플로 분류자를 업데이트하거나 새 클래스를 추가해야 합니다. 이제 분류자 모델은 기존 분류자를 참조하고 기존 클래스에 대한 새 샘플을 추가하거나 샘플이 포함된 새 클래스를 추가할 수 있는 증분 학습을 지원합니다. 증분 교육을 사용하면 변화하는 비즈니스 요구에 맞게 데이터 보존이 어려운 시나리오와 분류자를 업데이트해야 하는 시나리오를 사용할 수 있습니다. 증분 학습은 API 버전 `2024-02-29-preview` 이상을 사용하여 학습된 모델에서 지원됩니다.

① 중요

증분 학습은 동일한 API 버전으로 학습된 모델에서만 지원됩니다. 모델을 확장하려는 경우 원래 모델이 학습된 API 버전을 사용하여 모델을 확장합니다. 증분 학습은 API 버전 2024-02-29-preview 이상에서만 지원됩니다.

증분 학습을 수행하려면 원래 모델 ID를 .로 `baseClassifierId` 제공해야 합니다. 증분 학습을 사용하는 방법에 대한 자세한 내용은 [증분 학습을 참조하세요](#).

Office 문서 유형 지원

이제 PDF, 이미지, Word, PowerPoint 및 Excel을 비롯한 다양한 형식으로 문서 형식을 인식하도록 분류자를 학습시킬 수 있습니다. 학습 데이터 세트를 어셈블할 때 지원되는 형식의 문서를 추가할 수 있습니다. 분류자는 특정 형식에 명시적으로 레이블을 지정할 필요가 없습니다. 모범 사례로, 모델의 전반적인 정확도를 향상시키기 위해 학습 데이터 세트에 각 형식의 샘플이 하나 이상 있는지 확인합니다.

사용자 지정 분류 및 구성된 모델 비교

사용자 지정 분류 모델은 일부 시나리오에서 [구성된 모델](#)을 대체할 수 있지만 주의해야 할 몇 가지 차이점이 있습니다.

테이블 확장

기능	사용자 지정 분류자 프로세스	구성된 모델 프로세스
추출 모델 처리를 위해 학습된 형식 중 하나에 속하는 알 수 없는 형식의 단일 문서를 분석합니다.	<ul style="list-style-type: none">여러 번의 호출이 필요합니다.문서 클래스를 기반으로 분류 모델을 호출합니다. 이 단계에서는 추출 모델 분석을 호출하기 전에 신뢰도 기반 검사를 허용합니다.추출 모델을 호출합니다.	<ul style="list-style-type: none">입력 문서 형식에 해당하는 모델을 포함하는 구성된 모델에 대한 단일 호출이 필요합니다.
추출 모델 처리를 위해 학습된 여러 형식에 속하는 알 수 없는 형식의 단일 문서를 분석합니다.	<ul style="list-style-type: none">여러 번의 호출이 필요합니다.추출을 위해 지정된 형식과 일치하지 않는 문서를 무시하는 분류자를 호출합니다.추출 모델을 호출합니다.	<ul style="list-style-type: none">구성된 모델에 대한 단일 호출이 필요합니다.작성된 모델은 문서를 무시할 수 없습니다.
추출 모델 처리를 위해 학습된 형식 중 하나에 속하는 알려진 형식 또는 알 수 없는 형식의	<ul style="list-style-type: none">여러 번의 호출이 필요합니다.입력 파일에서 식별된 각 문서에 대해 추출 모델을 호출합	<ul style="list-style-type: none">구성된 모델에 대한 단일 호출이 필요합니다.작성 모델은 문서의 첫 번째 인스턴스에서 구성 요소 모델을

기능	사용자 지정 분류자 프로세스	구성된 모델 프로세스
여러 문서가 포함된 파일을 분석합니다.	니다. • 추출 모델을 호출합니다.	한 번 호출합니다. • 재기본 문서는 무시됩니다.

언어 지원

이제 다양한 언어의 문서에 대해 분류 모델을 학습시킬 수 있습니다. 전체 목록은 [지원되는 언어](#)를 참조하세요.

입력 요구 사항

지원 파일 형식:

[+] 테이블 확장

모델	PDF	이미지: jpeg/jpg, png, bmp, tiff, heif	Microsoft Office: Word(docx), Excel(xlxs), PowerPoint(pptx)
읽기	✓	✓	✓
레이아웃	✓	✓	✓ (2024-02-29-preview, 2023-10-31-preview 이상)
일반 문서	✓	✓	
사전 제작	✓	✓	
사용자 지정 추출	✓	✓	
사용자 지정 분류	✓	✓	✓

- 최상의 결과를 위해 문서 형식당 다섯 개의 명확한 사진 또는 고품질 스캔을 제공하세요.
- PDF 및 TIFF의 경우 최대 2,000페이지를 처리할 수 있습니다(무료 계층 구독에서는 처음 두 페이지만 처리됨).
- 문서 분석을 위한 파일 크기는 유료(S0) 계층의 경우 500MB, 무료(F0) 계층의 경우 4MB입니다.
- 이미지 크기는 50 x 50픽셀에서 10,000 x 10,000픽셀 사이여야 합니다.

- PDF가 암호로 잠긴 경우에는 제출하기 전에 잠금을 해제해야 합니다.
- 추출할 텍스트의 최소 높이는 1024 x 768 픽셀 이미지의 경우 12픽셀입니다. 이 차원은 인치당 150개 점(DPI)의 약 8-point 텍스트에 해당합니다.
- 사용자 지정 모델 학습의 경우 학습 데이터의 최대 페이지 수는 사용자 지정 템플릿 모델의 경우 500개, 사용자 지정 인공신경망 모델의 경우 50,000개입니다.
- 사용자 지정 추출 모델 학습의 경우 학습 데이터의 총 크기는 템플릿 모델의 경우 50MB, 신경망 모델의 경우 1G-MB입니다.
- 사용자 지정 분류 모델 학습의 경우 학습 데이터의 총 크기는 1GB이며 최대 10,000 페이지입니다.

문서 분할

파일에 문서가 두 개 이상 있는 경우 분류자는 입력 파일에 포함된 다양한 문서 형식을 식별할 수 있습니다. 분류자 응답에는 파일에 포함된 식별된 각 문서 형식의 페이지 범위가 포함됩니다. 이 응답에는 동일한 문서 형식의 여러 인스턴스가 포함될 수 있습니다.

이제 분석 작업에는 분할 동작을 세부적으로 제어할 수 있는 `splitMode` 속성이 포함됩니다.

- 전체 입력 파일을 분류를 위한 단일 문서로 처리하려면 `splitMode`를 `none`으로 설정합니다. 이렇게 하면 서비스는 전체 입력 파일에 대해 하나의 클래스만 반환합니다.
- 입력 파일의 각 페이지를 분류하려면 `splitMode`를 `perPage`로 설정합니다. 서비스는 각 페이지를 개별 문서로 분류하려고 시도합니다.
- `splitMode`를 `auto`로 설정하면 서비스는 문서 및 연결된 페이지 범위를 식별합니다.

모범 사례

사용자 지정 분류 모델을 학습하려면 클래스당 최소 5개의 샘플이 필요합니다. 클래스가 비슷한 경우 추가 학습 샘플을 추가하면 모델 정확도가 향상됩니다.

분류자는 각 문서를 클래스 중 하나에 할당하려 시도합니다. 학습 데이터 세트의 일부인 클래스에 없는 문서 형식을 모델이 볼 것으로 예상하는 경우 분류 점수의 임계값을 설정하거나 문서 형식의 몇 가지 대표적인 샘플을 `"other"` 클래스에 추가해야 합니다. 클래스를 `"other"` 추가하면 불필요한 문서가 분류자 품질에 영향을 주지 않습니다.

모델 학습

사용자 지정 분류 모델은 v4.0:2024-02-29-preview 및 v3.1:2023-07-31(GA) API에서 지원됩니다. [문서 인텔리전스 스튜디오](#)는 사용자 지정 분류자를 대화형으로 학습시키는 코드 없는 사용자 인터페이스를 제공합니다. 시작하려면 [방법 가이드](#)를 따르세요.

REST API를 사용할 때 폴더별로 문서를 구성하는 경우 요청의 `azureBlobSource` 속성을 사용하여 분류 모델을 학습시킬 수 있습니다.

JSON

```
https://{{endpoint}}/documentintelligence/documentClassifiers:build?api-version=2024-02-29-preview

{
    "classifierId": "demo2.1",
    "description": "",
    "docTypes": {
        "car-maint": {
            "azureBlobSource": {
                "containerUrl": "SAS URL to container",
                "prefix": "sample1/car-maint/"
            }
        },
        "cc-auth": {
            "azureBlobSource": {
                "containerUrl": "SAS URL to container",
                "prefix": "sample1/cc-auth/"
            }
        },
        "deed-of-trust": {
            "azureBlobSource": {
                "containerUrl": "SAS URL to container",
                "prefix": "sample1/deed-of-trust/"
            }
        }
    }
}
```

또는 파일의 플랫 목록이 있거나 각 폴더 내에서 몇 개의 선택 파일만 사용하여 모델을 학습시킬 계획인 경우 `azureBlobFileListSource` 속성을 사용하여 모델을 학습시킬 수 있습니다. 이 단계에서는 [JSON 줄](#) 형식의 `file list`가 필요합니다. 각 클래스에 대해 학습을 위해 제출할 파일 목록이 있는 새 파일을 추가합니다.

JSON

```
{
    "classifierId": "demo2",
    "description": "",
    "docTypes": {
```

```

"car-maint": {
    "azureBlobFileListSource": {
        "containerUrl": "SAS URL to container",
        "fileList": "{path to dataset root}/car-maint.jsonl"
    }
},
"cc-auth": {
    "azureBlobFileListSource": {
        "containerUrl": "SAS URL to container",
        "fileList": "{path to dataset root}/cc-auth.jsonl"
    }
},
"deed-of-trust": {
    "azureBlobFileListSource": {
        "containerUrl": "SAS URL to container",
        "fileList": "{path to dataset root}/deed-of-trust.jsonl"
    }
}
}
}

```

예를 들어 파일 목록 `car-maint.jsonl`에는 다음 파일이 포함됩니다.

JSON

```
{"file":"classifier/car-maint/Commercial Motor Vehicle - Adatum.pdf"}
{"file":"classifier/car-maint/Commercial Motor Vehicle - Fincher.pdf"}
 {"file":"classifier/car-maint/Commercial Motor Vehicle - Lamna.pdf"}
 {"file":"classifier/car-maint/Commercial Motor Vehicle - Liberty.pdf"}
 {"file":"classifier/car-maint/Commercial Motor Vehicle - Trey.pdf"}
```

모델 응답

문서 분류 모델을 사용하여 입력 파일을 분석합니다.

JSON

```
https://{{endpoint}}/documentintelligence/documentClassifiers/{{classifier}}:analyze?api-version=2024-02-29-preview
```

응답에는 응답의 문서 섹션에 연결된 페이지 범위가 있는 식별된 문서가 포함됩니다.

JSON

```
{
    ...
}
```

```
"documents": [
  {
    "docType": "formA",
    "boundingRegions": [
      { "pageNumber": 1, "polygon": [...] },
      { "pageNumber": 2, "polygon": [...] }
    ],
    "confidence": 0.97,
    "spans": []
  },
  {
    "docType": "formB",
    "boundingRegions": [
      { "pageNumber": 3, "polygon": [...] }
    ],
    "confidence": 0.97,
    "spans": []
  },
  ...
]
```

다음 단계

사용자 지정 분류 모델을 만드는 방법을 알아봅니다.

[사용자 지정 분류 모델 빌드](#)

[사용자 지정 모델 개요](#)

증분 분류자 교육

아티클 • 2024. 03. 08.

이 콘텐츠의 적용 대상은 다음과 같습니다. v4.0(미리 보기)

Azure AI 문서 인텔리전스는 지능형 문서 처리 솔루션을 빌드할 수 있는 클라우드 기반 Azure AI 서비스입니다. 문서 인텔리전스 API는 이미지, PDF 및 기타 문서 파일을 분석하여 다양한 콘텐츠, 레이아웃, 스타일 및 의미 체계 요소를 추출하고 검색합니다.

[문서 인텔리전스 사용자 지정 분류 모델](#)은 레이아웃 및 언어 기능을 결합하여 애플리케이션 내에서 처리하는 문서를 정확하게 검색하고 식별하는 딥 러닝 모델 유형입니다. 사용자 지정 분류 모델은 입력 파일의 분류를 한 번에 한 페이지씩 수행하여 문서를 식별하며 입력 파일 내에서 여러 문서 또는 단일 문서의 여러 인스턴스를 식별할 수도 있습니다.

문서 인텔리전스 문서 분류자는 파일에서 알려진 문서 유형을 식별합니다. 여러 문서 형식으로 입력 파일을 처리하거나 문서 형식을 모르는 경우 분류자를 사용하여 문서를 식별합니다. 기존 클래스에 대한 새 템플릿을 추가하거나, 인식할 새 문서 형식을 추가하거나, 분류자 신뢰도가 낮으면 분류자를 주기적으로 업데이트해야 합니다. 일부 시나리오에서는 분류자를 학습시키는 데 사용된 원래 문서 집합을 더 이상 가질 수 없습니다. 증분 학습을 사용하면 이제 레이블이 지정된 새 샘플만 사용하여 분류자를 업데이트할 수 있습니다.

① 참고

증분 학습은 사용자 지정 모델이 아닌 문서 분류자 모델에만 적용됩니다.

증분 학습은 사용자 지정 분류자의 품질을 향상하려는 경우에 유용합니다. 기존 클래스에 대한 새 학습 샘플을 추가하면 기존 문서 형식에 대한 모델의 신뢰도가 향상됩니다. 예를 들어 기존 양식의 새 버전이 추가되거나 새 문서 형식이 있는 경우 예를 들어 애플리케이션에서 새 문서 형식을 유효한 입력으로 지원하기 시작할 때입니다.

증분 학습 시작

- 증분 학습은 새 API 엔드포인트를 도입하지 않습니다.
- `documentClassifiers:build` 요청 페이로드는 증분 학습을 지원하도록 수정됩니다.
- 증분 학습을 수행하면 기존 분류자를 그대로 두고 새 분류자 모델을 만듭니다.
- 새 분류자는 새로 제공된 샘플과 함께 모든 문서 샘플 및 이전 분류자의 형식을 갖습니다. 애플리케이션이 새로 학습된 분류자에서 작동하도록 업데이트되었는지 확인

해야 합니다.

① 참고

분류자 복사 작업은 현재 사용할 수 없습니다.

증분 분류자 빌드 요청 만들기

증분 분류자 빌드 요청은 분류 문서 빌드 요청과 유사하지만 새 `baseClassifierId` 속성을 포함합니다. `baseClassifierId` 확장하려는 기존 분류자로 설정됩니다. 또한 샘플 집합의 `docTypes` 다양한 문서 형식에 대해 제공해야 합니다. `baseClassifier`에 존재하는 샘플을 제공 `docType` 하면 요청에 제공된 샘플이 기본 분류자를 학습할 때 제공된 샘플에 추가됩니다. 증분 학습에 추가된 새 `docType` 값은 새 분류자만 추가됩니다. 샘플을 다시 지정하는 프로세스는 변경되지 기본. 자세한 내용은 [분류자 모델 학습을 참조하세요](#).

샘플 POST 요청

증분 문서 분류자 빌드에 대한 샘플 `POST` 요청

`POST {your-endpoint}/documentintelligence/documentClassifiers:build?api-version=2024-02-29-preview`

JSON

```
{  
    "classifierId": "myAdaptedClassifier",  
    "description": "Classifier description",  
    "baseClassifierId": "myOriginalClassifier",  
    "docTypes": {  
        "formA": {  
            "azureBlobSource": {  
                "containerUrl": "https://myStorageAccount.blob.core.windows.net/myContainer?mySasToken",  
                "prefix": "formADocs/"  
            }  
        },  
        "formB": {  
            "azureBlobFileListSource": {  
                "containerUrl": "https://myStorageAccount.blob.core.windows.net/myContainer?mySasToken",  
                "fileList": "formB.jsonl"  
            }  
        }  
    }  
}
```

POST 응답

모든 문서 인텔리전스 API는 비동기이며 반환된 작업 위치를 폴링하면 빌드 작업에 대한 상태 제공합니다. 분류자는 학습 속도가 빠르며 분류자는 1~2분 안에 사용할 수 있습니다.

성공적으로 완료된 경우

- 성공한 `POST` 메서드는 서비스에서 요청을 생성했음을 나타내는 응답 코드를 반환 `202 OK` 합니다.
- 번역된 문서는 대상 컨테이너에 있습니다.
- 요청은 `POST` 를 비롯한 `Operation-Location` 응답 헤더도 반환합니다. 이 헤더의 값에는 비동기 작업의 상태 가져와 동일한 리소스 구독 키가 있는 요청을 사용하여 결과를 검색하기 위해 쿼리할 수 있는 `GET` 값이 포함 `resultId` 됩니다.

샘플 GET 요청

증분 문서 분류자의 결과를 검색하기 위한 샘플 `GET` 요청

```
GET {your-
endpoint}/documentintelligence/documentClassifiers/{classifierId}/analyzeResults/{re
sultId}?api-version=2024-02-29-preview
```

JSON

```
{
  "classifierId": "myAdaptedClassifier",
  "description": "Classifier description",
  "createdDateTime": "2022-07-30T00:00:00Z",
  "expirationDateTime": "2023-01-01T00:00:00Z",
  "apiVersion": "2024-02-29-preview",

  "baseClassifierId": "myOriginalClassifier",

  "docTypes": {
    "formA": {
      "azureBlobSource": {
        "containerUrl":
        "https://myStorageAccount.blob.core.windows.net/myContainer",
        "prefix": "formADocs/"
      }
    },
    "formB": {
      "azureBlobFileListSource": {
        "containerUrl":
        "https://myStorageAccount.blob.core.windows.net/myContainer",
        "fileList": "formB.json"
      }
    }
  }
}
```

```
        }  
    }  
}
```

GET 응답

증분 학습된 분류자의 응답은 `GET` 표준 분류자 `GET` 응답과 다릅니다. 증분 학습된 분류자는 지원되는 모든 문서 형식을 반환하지 않습니다. 증분 학습 단계 및 확장된 기본 분류자에서 추가되거나 업데이트된 문서 형식을 반환합니다. 문서 형식의 전체 목록을 얻으려면 기본 분류자를 나열해야 합니다. 기본 분류자를 삭제해도 증분 학습 분류자의 사용에는 영향을 주지 않습니다.

제한

- 증분 학습은 기본 분류자와 증분 학습된 분류자를 모두 동일한 API 버전에서 학습하는 경우에만 작동합니다. 결과적으로 증분 학습된 분류자는 기본 분류자로 모델 수명 [주기가 동일합니다](#).
- 증분 분류자의 학습 데이터 세트 크기 제한은 다른 분류자 모델과 동일합니다. 적용 가능한 제한[의 전체 목록은 서비스 제한을 참조하세요](#).

다음 단계

- 문서 분류에 대한 [자세한 정보](#)

모범 사례: 레이블이 지정된 데이터 세트 생성

아티클 • 2024. 03. 05.

이 콘텐츠의 적용 대상은 다음과 같습니다. ✓ v4.0(미리 보기) | 이전 버전: ✓ v3.1(GA) ✓ v3.0(GA)

사용자 지정 모델(템플릿 및 신경망)에는 모델을 학습하기 위해 최소 5개 문서의 레이블이 지정된 데이터 세트가 필요합니다. 레이블이 지정된 데이터 세트의 품질은 학습된 모델의 정확도에 영향을 미칩니다. 이 가이드는 다양한 데이터 세트를 조합하여 정확도가 높은 모델을 생성하는 방법에 대해 자세히 알아보고 문서에 레이블을 지정하는 모범 사례를 제공합니다.

레이블이 지정된 데이터 세트의 구성 요소 이해

레이블이 지정된 데이터 세트는 다음과 같은 여러 파일로 구성됩니다.

- 샘플 문서 집합(일반적으로 PDF 또는 이미지)을 제공합니다. 모델을 학습하려면 최소 5개의 문서가 필요합니다.
- 또한 레이블 지정 프로세스는 다음 파일을 생성합니다.
 - 첫 번째 필드가 추가되면 `fields.json` 파일이 만들어집니다. 전체 학습 데이터 세트에 대해 하나의 `fields.json` 파일이 있으며 필드 목록에는 필드 이름과 연결된 하위 필드 및 형식이 포함됩니다.
 - Studio는 [레이아웃 API](#)를 통해 각 문서를 실행합니다. 데이터 세트의 각 샘플 파일에 대한 레이아웃 응답이 `{file}.ocr.json`으로 추가됩니다. 레이아웃 응답은 텍스트의 특정 범위에 레이블이 지정될 때 필드 레이블을 생성하는 데 사용됩니다.
 - 문서에서 필드에 레이블이 지정되면 `{file}.labels.json` 파일이 만들어지거나 업데이트됩니다. 레이블 파일에는 사용자가 특정 필드의 값으로 추가하는 각 텍스트 범위에 대한 레이아웃 출력의 텍스트 범위 및 관련 다각형이 포함되어 있습니다.

동영상: 사용자 지정 레이블 팁 및 포인터

- 다음 동영상은 더 높은 정확도로 사용자 지정 모델을 빌드하는 데 도움이 되는 두 가지 프레젠테이션 중 첫 번째입니다(두 번째 프레젠테이션에서는 [문서 레이블 지정](#)

모범 사례를 살펴봅니다).

- 여기에서는 균형 잡힌 데이터 세트를 만들고 레이블을 지정할 올바른 문서를 선택하는 방법을 살펴봅니다. 이 과정을 통해 더 높은 품질의 모델을 얻을 수 있습니다.

[https://www.microsoft.com/ko-kr/videoplayer/embed/RWWHru?
postJs||Msg=true&autoCaptions=ko-kr](https://www.microsoft.com/ko-kr/videoplayer/embed/RWWHru?postJs||Msg=true&autoCaptions=ko-kr)

균형 잡힌 데이터 세트 만들기

레이블 지정을 시작하기 전에 문서의 몇 가지 다른 샘플을 살펴보고 레이블이 지정된 데이터 세트에서 사용할 샘플을 식별하는 것이 좋습니다. 균형 잡힌 데이터 세트는 문서에 대해 예상되는 모든 일반적인 변형을 나타냅니다. 균형 잡힌 데이터 세트를 만들면 가능한 가장 높은 정확도를 가진 모델이 만들어집니다. 고려해야 할 몇 가지 예는 다음과 같습니다.

- 문서 형식:** 디지털 문서와 검사한 문서를 모두 분석하려는 경우 학습 데이터 세트에 각 형식의 몇 가지 예를 추가합니다.
- 변형(템플릿 모델):** 데이터 세트를 폴더로 분할하고 각 변형에 대해 모델을 학습시키는 것이 좋습니다. 구조 또는 레이아웃을 포함하는 모든 변형은 다른 모델로 분할되어야 합니다. 그런 다음 개별 모델을 하나의 [구성된 모델](#)로 구성할 수 있습니다.
- 변형(신경망 모델):** 데이터 세트에 약 15개 이하의 관리 가능한 변형 집합이 있는 경우 각 변형의 몇 가지 샘플로 단일 데이터 세트를 만들어 단일 모델을 학습합니다. 템플릿 변형 수가 15개보다 크면 여러 모델을 학습시키고 함께 [구성](#)합니다.
- 테이블:** 행 수가 가변적인 테이블이 포함된 문서의 경우 학습 데이터 세트가 행 수가 다른 문서도 나타내는지 확인합니다.
- 다중 페이지 테이블:** 테이블이 여러 페이지에 걸쳐 있는 경우 단일 테이블에 레이블을 지정합니다. 예상되는 변형이 표시된 학습 데이터 세트에 문서를 추가합니다. 단일 페이지에만 테이블이 있는 문서와 레이블이 지정된 모든 행이 있는 두 페이지 이상에 걸쳐 있는 테이블이 있는 문서입니다.
- 선택 필드:** 데이터 세트에 옵션 필드가 있는 문서가 포함된 경우 학습 데이터 세트에 옵션이 표시된 문서가 몇 개 있는지 유효성을 검사합니다.

필드를 식별하여 시작

시간을 내어 데이터 세트에서 레이블을 지정하려는 각 필드를 식별합니다. 선택적 필드에 주의합니다. 지원되는 형식과 가장 일치하는 레이블로 필드를 정의합니다.

다음 지침을 사용하여 필드를 정의합니다.

- 사용자 지정 신경망 모델의 경우 의미 체계로 관련된 필드 이름을 사용합니다. 예를 들어, 추출되는 값이 `Effective Date` 인 경우 `date1`과 같은 일반 이름이 아닌 `effective_date` 또는 `EffectiveDate`로 이름을 지정합니다.
- 이상적으로는 파스칼 또는 카멜 대/소문자로 필드 이름을 지정합니다.
- 값이 시각적으로 반복되는 구조의 일부이고 단일 값만 필요한 경우 테이블로 레이블을 지정하고 후처리 중에 필요한 값을 추출합니다.
- 여러 페이지에 걸쳐 있는 테이블 형식 필드의 경우 필드를 단일 테이블로 정의하고 레이블을 지정합니다.

① 참고

사용자 지정 신경망 모델은 사용자 지정 템플릿 모델과 동일한 레이블 지정 형식 및 전략을 공유합니다. 현재 사용자 지정 신경망 모델은 사용자 지정 템플릿 모델에서 지원하는 필드 형식의 하위 집합만 지원합니다.

모델 기능

사용자 지정 신경망 모델은 현재 키-값 쌍, 구조화된 필드(테이블) 및 선택 표시만 지원합니다.

▣ 테이블 확장

모델 유형	양식 필드	선택 표시	표 형식 필드	서명	지역
사용자 지정 인공신경망	✓ 지원됨	✓ 지원됨	✓ 지원됨	지원되지 않음	✓ 지원됨 ¹
사용자 지정 템플릿	✓ 지원됨	✓ 지원됨	✓ 지원됨	✓ 지원됨	✓ 지원됨

¹ 지역 레이블 지정 구현은 템플릿과 신경망 모델 간에 다릅니다. 템플릿 모델의 경우 레이블이 지정된 영역에서 텍스트를 찾을 수 없는 경우 학습 프로세스는 학습 시 가상 데이터를 삽입합니다. 신경망 모델을 사용하면 가상 텍스트가 삽입되지 않고 인식된 텍스트가 그대로 사용됩니다.

표 형식 필드

테이블 형식 필드(테이블)는 API 버전 `2022-06-30-preview`부터 사용자 지정 신경망 모델에서 지원됩니다. API 버전 `2022-06-30-preview` 이상으로 학습된 모델은 테이블 형식 필

드 레이블을 허용하고 API 버전 2022-06-30-preview 이상이 있는 모델로 분석된 문서는 `analyzeResult` 개체 결과의 `documents` 섹션 내 출력에 테이블 형식 필드를 생성합니다.

표 형식 필드는 기본적으로 **크로스 페이지 테이블**을 지원합니다. 여러 페이지에 걸쳐 있는 테이블에 레이블을 지정하려면 단일 테이블의 여러 페이지에 걸쳐 테이블의 각 행에 레이블을 지정합니다. 가장 좋은 방법은 데이터 세트에 예상되는 변형의 몇 가지 샘플이 포함되어 있는지 확인하는 것입니다. 예를 들어, 전체 테이블이 단일 페이지에 있는 샘플과 둘 이상의 페이지에 걸쳐 있는 테이블 샘플을 모두 포함합니다.

표 형식 필드는 표로 인식되지 않는 문서 내에서 반복되는 정보를 추출할 때도 유용합니다. 예를 들어 이력서에서 반복되는 업무 환경 섹션에 레이블을 지정하고 테이블 형식 필드로 추출할 수 있습니다.

① 참고

레이블이 지정된 테이블 필드는 응답의 `documents` 섹션의 일부로 추출됩니다. 응답에는 레이아웃 모델이 문서에서 추출한 테이블이 포함된 `tables` 섹션도 포함되어 있습니다. 필드에 테이블로 레이블을 지정한 경우 응답의 문서 섹션에서 필드를 찾습니다.

레이블 지정 지침

- **레이블 지정 값이 필요합니다.** 주변 텍스트를 포함하지 마세요. 예를 들어, 확인란에 레이블을 지정할 때 문서에서 예 또는 아니요 텍스트에 레이블을 지정하는 대신 `selectionYes` 및 `selectionNo`와 같이 확인란 선택을 나타내도록 필드 이름을 지정 합니다.
- **인터리빙 필드 값 제공 안 함** 한 필드의 단어 및/또는 영역의 값은 다른 필드와 인터리빙하지 않고 자연스럽게 읽는 순서로 연속된 시퀀스이거나 다른 필드를 포함하지 않는 지역에 있어야 합니다.
- **일관된 레이블 지정.** 값이 문서의 여러 컨텍스트에 표시되는 경우 문서 전체에서 동일한 컨텍스트를 일관되게 선택하여 값에 레이블을 지정합니다.
- **시각적으로 반복되는 데이터.** 테이블은 명시적 테이블뿐만 아니라 시각적으로 반복되는 정보 그룹을 지원합니다. 명시적 테이블은 분석된 문서의 테이블 섹션에서 레이아웃 출력의 일부로 식별되며 테이블로 레이블을 지정할 필요가 없습니다. 정보가 시각적으로 반복되고 레이아웃 응답의 일부로 테이블로 식별되지 않는 경우에만 테이블 필드에 레이블을 지정합니다. 이력서의 반복되는 업무 환경 섹션을 예로 들 수 있습니다.

- 지역 레이블 지정(사용자 지정 템플릿). 특정 지역에 레이블을 지정하면 값이 없을 때 값을 정의할 수 있습니다. 값이 선택 사항인 경우 레이블이 지정되지 않은 지역에 몇 개의 샘플 문서를 남겨 두어야 합니다. 지역에 레이블을 지정할 때 레이블과 함께 주변 텍스트를 포함하지 마세요.

다음 단계

- 사용자 지정 모델을 학습합니다.

[모델을 훈련하는 방법](#)

- 사용자 지정 템플릿 모델에 대해 자세히 알아보세요.

[사용자 지정 템플릿 모델](#)

- 사용자 지정 신경망 모델에 대해 자세히 알아보세요.

[사용자 지정 신경망 모델](#)

- REST API를 봅니다.

[문서 인텔리전스 API v4.0:2023-10-31-preview](#)

[문서 인텔리전스 API v3.1:2023-07-31\(GA\)](#)

레이블이 지정된 데이터 세트를 빌드하기 위한 팁

아티클 • 2024. 03. 10.

이 콘텐츠의 적용 대상은 다음과 같습니다. ✓ v4.0(미리 보기) | 이전 버전: ✓ v3.1(GA) ✓ v3.0(GA)

이 문서에서는 Document Intelligence Studio에서 사용자 지정 모델 데이터 세트에 레이블을 지정하는 가장 좋은 방법을 강조합니다. 문서에 레이블이 많거나, 길거나, 구조가 다양한 경우 문서에 레이블을 지정하려면 오랜 시간이 걸릴 수 있습니다. 이러한 팁은 문서에 더욱 효율적으로 레이블을 지정하는 데 도움이 됩니다.

동영상: 사용자 지정 레이블 모범 사례

- 다음 비디오는 사용자 지정 모델을 더 정확히 빌드하는 데 도움이 되는 프레젠테이션 2개 중 두 번째입니다(첫 번째 프레젠테이션에서는 [균형 잡힌 데이터 집합을 만드는 방법](#)을 살펴봅니다).
- 선택한 문서에 레이블을 지정하는 모범 사례를 검토합니다. 레이블을 일관적이고 의미상 관련이 있도록 지정하면 모델 성능이 향상됩니다.

<https://www.microsoft.com/ko-kr/videoplayer/embed/RE5fZKB?postJsIMsg=true&autoCaptions=ko-kr>

Search

레이블을 지정할 특정 단어를 찾아야 하지만 문서 어디쯤 있는지 모르는 경우를 대비해, 이제 Studio에 인스턴스 검색 상자가 포함됩니다. 단어나 구를 검색하고 문서의 특정 섹션으로 이동하여 항목에 레이블을 지정하기만 하면 됩니다.

테이블 자동 레이블 지정

테이블에 행이 많거나 텍스트가 조밀한 경우 레이블 지정이 어려울 수 있습니다. 레이아웃 테이블이 필요한 결과를 추출하면 해당 결과를 사용하고 레이블 지정 프로세스는 건너뛰면 됩니다. 레이아웃 테이블이 필요한 것은 아니라 해도, 레이아웃이 추출한 값에서 테이블 필드를 생성하는 것으로 프로세스를 시작할 수 있습니다. 먼저 페이지에서 테이블 아이콘을 선택하고 자동 레이블 단추를 선택합니다. 그런 다음 필요에 따라 값을 편집할 수 있습니다. 자동 레이블은 현재 단일 페이지 테이블만 지원합니다.

Shift 선택

범위에 있는 각 단어를 표시하는 대신 넓은 텍스트 범위에 레이블을 지정하는 경우, 단어를 선택할 때 Shift 키를 누르고 있으면 더 빠르게 레이블을 지정할 수 있으며 텍스트 범위에 있는 그 어떤 단어도 놓치지 않을 수 있습니다.

지역 레이블 지정

넓은 텍스트 범위에 레이블을 지정하는 두 번째 옵션은 영역 레이블 지정을 사용하는 것입니다. 지역 레이블 지정을 사용하는 **OCR** 경우 결과는 학습 시간에 값으로 채워집니다. Shift 선택과 영역 레이블 지정의 차이점은 Shift를 사용하여 레이블을 지정하면 시각적 피드백이 제공된다는 것뿐입니다.

겹치는 필드 레이블 지정

필드와 표 셀에는 겹치는 필드가 지원됩니다. 분석 결과에 겹치는 필드가 포함될 것으로 예상되는 경우 특정 필드가 레이블이 겹치는 학습 데이터 세트에 샘플을 하나 이상 추가해야 합니다. 겹치는 필드에 레이블을 지정하려면 지역 레이블 지정 기능을 사용하여 각 필드에 대한 지역을 선택합니다. 전체 및 부분 중복이 모두 지원됩니다. 문서의 모든 단어는 두 필드에 대해서만 레이블을 지정할 수 있습니다.

필드 하위 유형

필드를 만들 때는 올바른 하위 유형을 선택하여 후처리를 최소화하세요. 예를 들어 날짜의 경우 `dd-mm-yyyy` 형식으로 값을 추출하려면 **dmy** 옵션을 선택하세요.

다음 단계

- 사용자 지정 레이블 지정에 대해 자세히 알아보세요.

[사용자 지정 레이블](#)

- 사용자 지정 템플릿 모델에 대해 자세히 알아보세요.

[사용자 지정 템플릿 모델](#)

- 사용자 지정 신경망 모델에 대해 자세히 알아보세요.

[사용자 지정 신경망 모델](#)

문서 인텔리전스로 구성된 사용자 지정 모델

아티클 • 2024. 03. 08.

① 중요

- 문서 인텔리전스 공개 미리 보기 릴리스에서는 현재 개발 중인 기능에 대한 초기 액세스를 제공합니다.
- 기능, 방식 및 프로세스는 GA(일반 공급) 전에 사용자 피드백에 따라 변경될 수 있습니다.
- Document Intelligence 클라이언트 라이브러리의 공개 미리 보기 버전은 기본적으로 REST API 버전 [2024-02-29-preview로 설정됩니다](#).
- 공개 미리 보기 버전 [2024-02-29-preview](#) 는 현재 다음 Azure 지역에서만 사용할 수 있습니다.
 - 미국 동부
 - 미국 서부2
 - 서유럽

이 콘텐츠의 적용 대상은 다음과 같습니다. ✓v4.0(미리 보기) | 이전 버전: ✓v3.1(GA) ✓v3.0(GA) ✓v2.1(GA)

구성형 모델. 구성형 모델은 사용자 지정 모델 컬렉션을 가져와서 사용자의 양식 유형에서 빌드된 단일 모델에 할당하여 만듭니다. 구성형 모델을 사용하여 분석용으로 문서가 제출될 때 서비스는 분류를 수행해서 제출된 문서를 가장 잘 나타내는 사용자 지정 모델을 결정합니다.

구성형 모델을 선택하면 단일 모델 ID로 호출된 구성형 모델에 여러 사용자 지정 모델을 할당할 수 있습니다. 이 방법은 여러 모델을 학습시키고 비슷한 양식 유형을 분석하도록 그룹화하려는 경우에 유용합니다. 예를 들어 구성형 모델은 물자, 장비 및 가구 구매 주문을 분석하도록 학습된 사용자 지정 모델을 포함할 수 있습니다. 적절한 모델을 수동으로 선택하는 대신, 구성형 모델을 사용하여 각 분석 및 추출에 적합한 사용자 지정 모델을 결정할 수 있습니다.

- `Custom form` 및 `Custom template` 모델을 단일 구성 모델로 함께 구성할 수 있습니다.
- 모델 구성 작업을 사용하면 최대 200개까지 학습된 사용자 지정 모델을 단일 구성형 모델에 할당할 수 있습니다. 구성된 모델로 문서를 분석하기 위해 문서 인텔리전

스는 먼저 제출된 양식을 분류하고 가장 일치하는 할당 모델을 선택한 후 결과를 반환합니다.

- **사용자 지정 템플릿 모델**의 경우 사용자 지정 템플릿의 변형 또는 다른 양식 유형을 사용해서 구성형 모델을 만들 수 있습니다. 이 작업은 들어오는 양식이 여러 템플릿 중 하나에 속할 수 있을 때 유용합니다.
- 응답에는 문서 분석을 위해 사용된 구성형 모델을 나타내는 `docType` 속성이 포함됩니다.
- `Custom neural` 모델의 경우 단일 문서 형식의 다양한 변형을 단일 학습 데이터 세트에 추가하고 사용자 지정 신경망 모델을 학습하는 것이 가장 좋습니다. 모델 작성은 분석을 위해 제출되는 다양한 형식의 문서가 있는 시나리오에 가장 적합합니다.

사용자 지정 분류 모델이 도입되어, 분석 전에 **구성형 모델** 또는 **분류 모델**을 명시적 단계로 사용하도록 선택할 수 있습니다. 분류 또는 구성된 모델을 사용해야 하는 경우에 대한 자세한 내용은 [사용자 지정 분류 모델](#)을 참조하세요.

모델 제한 구성

① 참고

사용자 지정 인공신경망 모델과 함께 구성될 수 있는 모델의 호환성에는 몇 가지 제한이 있습니다.

구성형 모델 호환성

[+] 테이블 확장

사용자 지정 모델 유형	v2.1 및 v2.0으로 학습된 모델	사용자 지정 템플릿 모델 v3.0	사용자 지정 신경망 모델 3.0	사용자 지정 신경망 모델 v3.1
버전 2.1 및 v2.0으로 학습된 모델	지원됨	지원됨	지원되지 않음	지원되지 않음
사용자 지정 템플릿 모델 v3.0	지원됨	지원됨	지원되지 않음	지원되지 않음
사용자 지정 템플릿 모델 v3.0	지원되지 않음	지원되지 않음	지원되지 않음	지원되지 않음
사용자 지정 템플릿 모델 v3.1	지원되지 않음	지원되지 않음	지원되지 않음	지원되지 않음

사용자 지정 모델 유형	v2.1 및 v2.0으로 학습된 모델	사용자 지정 템플릿 모델 v3.0	사용자 지정 신경망 모델 3.0	사용자 지정 신경망 모델 v3.1
사용자 지정 신경망 모델 v3.0	지원되지 않음	지원되지 않음	지원됨	지원됨
사용자 지정 신경망 모델 v3.1	지원되지 않음	지원되지 않음	지원됨	지원됨

- API의 이전 버전(v2.1 이하)으로 학습된 모델을 구성하려면 동일한 레이블이 지정된 데이터 세트를 사용해서 v3.0 API로 모델을 학습합니다. 이렇게 추가하면 v2.1 모델을 다른 모델과 함께 구성할 수 있습니다.
- API의 v2.1을 사용하여 구성된 모델은 계속 지원되며 업데이트가 필요하지 않습니다.
- 사용자 지정 모델의 경우 구성할 수 있는 최대 수는 100개입니다.

개발 옵션

Document Intelligence v4.0:2023-02-29-preview 는 다음 도구, 애플리케이션 및 라이브러리를 지원합니다.

테이블 확장

기능	리소스
사용자 지정 모델	<ul style="list-style-type: none"> 문서 인텔리전스 스튜디오 REST API C# SDK Java SDK JavaScript SDK Python SDK
구성형 모델	<ul style="list-style-type: none"> 문서 인텔리전스 스튜디오 REST API C# SDK Java SDK JavaScript SDK Python SDK

다음 단계

사용자 지정 모델을 만들고 구성하는 방법을 알아봅니다.

사용자 지정 모델 빌드

사용자 지정 모델 작성

Azure Form Recognizer client library for .NET - version 4.1.0

Article • 08/11/2023

Note: on July 2023, the Azure Cognitive Services Form Recognizer service was renamed to Azure AI Document Intelligence. Any mentions to Form Recognizer or Document Intelligence in documentation refer to the same Azure service.

Azure AI Document Intelligence is a cloud service that uses machine learning to analyze text and structured data from your documents. It includes the following main features:

- Layout - Extract text, selection marks, table structures, styles, and paragraphs, along with their bounding region coordinates from documents.
- General document - Analyze key-value pairs in addition to general layout from documents.
- Read - Read information about textual elements, such as page words and lines in addition to text language information.
- Prebuilt - Analyze data from certain types of common documents using prebuilt models. Supported documents include receipts, invoices, business cards, identity documents, US W2 tax forms, and more.
- Custom analysis - Build custom document models to analyze text, field values, selection marks, table structures, styles, and paragraphs from documents. Custom models are built with your own data, so they're tailored to your documents.
- Custom classification - Build custom classifier models that combine layout and language features to accurately detect and identify documents you process within your application.

[Source code](#) | [Package \(NuGet\)](#) | [API reference documentation](#) | [Product documentation](#) | [Samples](#)

Getting started

Install the package

Install the Azure Form Recognizer client library for .NET with [NuGet](#):

.NET CLI

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

Note: This version of the client library defaults to the 2023-07-31 version of the service.

This table shows the relationship between SDK versions and supported API versions of the service:

SDK version	Supported API version of service
4.1.0	2.0, 2.1, 2022-08-31, 2023-07-31
4.0.0	2.0, 2.1, 2022-08-31
3.1.X	2.0, 2.1
3.0.X	2.0

Note: Starting with version 4.0.0, a new set of clients were introduced to leverage the newest features of the Document Intelligence service. Please see the [Migration Guide](#) for detailed instructions on how to update application code from client library version 3.1.X or lower to the latest version. Additionally, see the [Changelog](#) for more detailed information. The table below describes the relationship of each client and its supported API version(s):

API version	Supported clients
2023-07-31	DocumentAnalysisClient and DocumentModelAdministrationClient
2022-08-31	DocumentAnalysisClient and DocumentModelAdministrationClient
2.1	FormRecognizerClient and FormTrainingClient
2.0	FormRecognizerClient and FormTrainingClient

Prerequisites

- An [Azure subscription](#).
- A [Cognitive Services or Form Recognizer resource](#) to use this package.

Create a Cognitive Services or Form Recognizer resource

Document Intelligence supports both [multi-service and single-service access](#). Create a Cognitive Services resource if you plan to access multiple cognitive services under a single endpoint/key. For Document Intelligence access only, create a Form Recognizer resource. Please note that you will need a single-service resource if you intend to use [Azure Active Directory authentication](#).

You can create either resource using:

- Option 1: [Azure Portal](#).
- Option 2: [Azure CLI](#).

Below is an example of how you can create a Form Recognizer resource using the CLI:

PowerShell

```
# Create a new resource group to hold the Form Recognizer resource
# If using an existing resource group, skip this step
az group create --name <your-resource-name> --location <location>
```

PowerShell

```
# Create the Form Recognizer resource
az cognitiveservices account create \
    --name <resource-name> \
    --resource-group <resource-group-name> \
    --kind FormRecognizer \
    --sku <sku> \
    --location <location> \
    --yes
```

For more information about creating the resource or how to get the location and sku information see [here](#).

Authenticate the client

In order to interact with the Document Intelligence service, you'll need to create an instance of the [DocumentAnalysisClient](#) class. An **endpoint** and **credential** are necessary to instantiate the client object.

Get the endpoint

You can find the endpoint for your Form Recognizer resource using the [Azure Portal](#) or [Azure CLI](#):

PowerShell

```
# Get the endpoint for the Form Recognizer resource
az cognitiveservices account show --name "<resource-name>" --resource-group
"<resource-group-name>" --query "properties.endpoint"
```

Either a regional endpoint or a custom subdomain can be used for authentication. They are formatted as follows:

```
Regional endpoint: https://<region>.api.cognitive.microsoft.com/
Custom subdomain: https://<resource-name>.cognitiveservices.azure.com/
```

A regional endpoint is the same for every resource in a region. A complete list of supported regional endpoints can be consulted [here](#). Please note that regional endpoints do not support AAD authentication.

A custom subdomain, on the other hand, is a name that is unique to the Form Recognizer resource. They can only be used by [single-service resources](#).

Get the API Key

The API key can be found in the [Azure Portal](#) or by running the following Azure CLI command:

PowerShell

```
az cognitiveservices account keys list --name "<resource-name>" --resource-
group "<resource-group-name>"
```

Create DocumentAnalysisClient with AzureKeyCredential

Once you have the value for the API key, create an `AzureKeyCredential`. With the endpoint and key credential, you can create the [DocumentAnalysisClient](#):

C#

```
string endpoint = "<endpoint>";
string apiKey = "<apiKey>";
var credential = new AzureKeyCredential(apiKey);
var client = new DocumentAnalysisClient(new Uri(endpoint), credential);
```

Create DocumentAnalysisClient with Azure Active Directory Credential

`AzureKeyCredential` authentication is used in the examples in this getting started guide, but you can also authenticate with Azure Active Directory using the [Azure Identity library](#). Note that regional endpoints do not support AAD authentication. Create a custom subdomain for your resource in order to use this type of authentication.

To use the [DefaultAzureCredential](#) provider shown below, or other credential providers provided with the Azure SDK, please install the `Azure.Identity` package:

.NET CLI

```
dotnet add package Azure.Identity
```

You will also need to [register a new AAD application](#) and [grant access](#) to Document Intelligence by assigning the `"Cognitive Services User"` role to your service principal.

Set the values of the client ID, tenant ID, and client secret of the AAD application as environment variables: `AZURE_CLIENT_ID`, `AZURE_TENANT_ID`, `AZURE_CLIENT_SECRET`.

C#

```
string endpoint = "<endpoint>";
var client = new DocumentAnalysisClient(new Uri(endpoint), new
DefaultAzureCredential());
```

Key concepts

DocumentAnalysisClient

`DocumentAnalysisClient` provides operations for:

- Analyzing input documents using prebuilt and custom models through the `AnalyzeDocument` and `AnalyzeDocumentFromUri` APIs.
- Detecting and identifying custom input documents with the `ClassifyDocument` and `ClassifyDocumentFromUri` APIs.

Sample code snippets are provided to illustrate using a `DocumentAnalysisClient` [here](#). More information about analyzing documents, including supported features, locales, and document types can be found in the [service documentation](#).

DocumentModelAdministrationClient

`DocumentModelAdministrationClient` provides operations for:

- Building custom models to analyze specific fields you specify by labeling your custom documents. A `DocumentModelDetails` instance is returned indicating the document type(s) the model can analyze, the fields it can analyze for each document type, as well as the estimated confidence for each field. See the [service documentation](#) for a more detailed explanation.
- Compose a model from a collection of existing models.
- Managing models created in your account.
- Copying a custom model from one Form Recognizer resource to another.
- Listing build operations or getting specific operations created within the last 24 hours.
- Building and managing document classification models to accurately detect and identify documents you process within your application.

See examples for [Build a Custom Model](#), [Manage Models](#), and [Build a Document Classifier](#).

Please note that models and classifiers can also be built using a graphical user interface such as the [Document Intelligence Studio](#).

Long-Running Operations

Because analyzing documents and building models take time, these operations are implemented as [long-running operations](#). Long-running operations consist of an initial request sent to the service to start an operation, followed by polling the service at intervals to determine whether the operation has completed or failed, and if it has succeeded, to get the result.

For long running operations in the Azure SDK, the client exposes a method that returns an `Operation<T>` object. You can set its parameter `waitFor` to `WaitUntil.Completed` to wait for the operation to complete and obtain its result; or set it to `WaitUntil.Started` if you just want to start the operation and consume the result later. A sample code snippet is provided to illustrate using long-running operations [below](#).

Thread safety

We guarantee that all client instance methods are thread-safe and independent of each other ([guideline](#)). This ensures that the recommendation of reusing client instances is always safe, even across threads.

Additional concepts

[Client options ↗](#) | [Accessing the response ↗](#) | [Handling failures ↗](#) | [Diagnostics ↗](#) | [Mocking](#) | [Client lifetime ↗](#)

Examples

The following section provides several code snippets illustrating common patterns used in the Form Recognizer .NET API. Most of the snippets below make use of asynchronous service calls, but keep in mind that the Azure.AI.FormRecognizer package supports both synchronous and asynchronous APIs.

Async examples

- [Extract Layout](#)
- [Use the Prebuilt General Document Model](#)
- [Use the Prebuilt Read Model](#)
- [Use Prebuilt Models](#)
- [Build a Custom Model](#)
- [Analyze Custom Documents](#)
- [Manage Models](#)
- [Build a Document Classifier](#)
- [Classify a Document](#)

Sync examples

- [Manage Models Synchronously](#)

Note that these samples use SDK version [4.1.0](#). For version 3.1.1 or lower, see [Form Recognizer Samples for V3.1.X ↗](#).

Extract Layout

Extract text, selection marks, table structures, styles, and paragraphs, along with their bounding region coordinates from documents.

C#

```
Uri fileUri = new Uri("<fileUri>");

AnalyzeDocumentOperation operation = await
    client.AnalyzeDocumentFromUriAsync(WaitUntil.Completed, "prebuilt-layout",
```

```

fileUri);
AnalyzeResult result = operation.Value;

foreach (DocumentPage page in result.Pages)
{
    Console.WriteLine($"Document Page {page.PageNumber} has
{page.Lines.Count} line(s), {page.Words.Count} word(s),");
    Console.WriteLine($"and {page.SelectionMarks.Count} selection
mark(s).");

    for (int i = 0; i < page.Lines.Count; i++)
    {
        DocumentLine line = page.Lines[i];
        Console.WriteLine($" Line {i} has content: '{line.Content}'.");

        Console.WriteLine($" Its bounding polygon (points ordered
clockwise):");

        for (int j = 0; j < line.BoundingPolygon.Count; j++)
        {
            Console.WriteLine($" Point {j} => X:
{line.BoundingPolygon[j].X}, Y: {line.BoundingPolygon[j].Y}");
        }
    }

    for (int i = 0; i < page.SelectionMarks.Count; i++)
    {
        DocumentSelectionMark selectionMark = page.SelectionMarks[i];

        Console.WriteLine($" Selection Mark {i} is
{selectionMark.State}.");
        Console.WriteLine($" Its bounding polygon (points ordered
clockwise):");

        for (int j = 0; j < selectionMark.BoundingPolygon.Count; j++)
        {
            Console.WriteLine($" Point {j} => X:
{selectionMark.BoundingPolygon[j].X}, Y:
{selectionMark.BoundingPolygon[j].Y}");
        }
    }
}

Console.WriteLine("Paragraphs:");

foreach (DocumentParagraph paragraph in result.Paragraphs)
{
    Console.WriteLine($" Paragraph content: {paragraph.Content}");

    if (paragraph.Role != null)
    {
        Console.WriteLine($" Role: {paragraph.Role}");
    }
}

```

```

foreach (DocumentStyle style in result.Styles)
{
    // Check the style and style confidence to see if text is handwritten.
    // Note that value '0.8' is used as an example.

    bool isHandwritten = style.IsHandwritten.HasValue && style.IsHandwritten
== true;

    if (isHandwritten && style.Confidence > 0.8)
    {
        Console.WriteLine($"Handwritten content found:");

        foreach (DocumentSpan span in style.Spans)
        {
            Console.WriteLine($" Content:
{result.Content.Substring(span.Index, span.Length)}");
        }
    }
}

Console.WriteLine("The following tables were extracted:");

for (int i = 0; i < result.Tables.Count; i++)
{
    DocumentTable table = result.Tables[i];
    Console.WriteLine($" Table {i} has {table.RowCount} rows and
{table.ColumnCount} columns.");

    foreach (DocumentTableCell cell in table.Cells)
    {
        Console.WriteLine($" Cell ({cell.RowIndex}, {cell.ColumnIndex})
has kind '{cell.Kind}' and content: '{cell.Content}'.");
    }
}

```

For more information and samples see [here ↗](#).

Use the Prebuilt General Document Model

Analyze text, selection marks, table structures, styles, paragraphs, and key-value pairs from documents using the prebuilt general document model.

C#

```

Uri fileUri = new Uri("<fileUri>");

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

```

```

Console.WriteLine("Detected key-value pairs:");

foreach (DocumentKeyValuePair kvp in result.KeyValuePairs)
{
    if (kvp.Value == null)
    {
        Console.WriteLine($"  Found key with no value:
'{kvp.Key.Content}'");
    }
    else
    {
        Console.WriteLine($"  Found key-value pair: '{kvp.Key.Content}' and
'{kvp.Value.Content}'");
    }
}

foreach (DocumentPage page in result.Pages)
{
    Console.WriteLine($"Document Page {page.PageNumber} has
{page.Lines.Count} line(s), {page.Words.Count} word(s),");
    Console.WriteLine($"and {page.SelectionMarks.Count} selection
mark(s).");

    for (int i = 0; i < page.Lines.Count; i++)
    {
        DocumentLine line = page.Lines[i];
        Console.WriteLine($"  Line {i} has content: '{line.Content}'.");

        Console.WriteLine($"    Its bounding polygon (points ordered
clockwise):");

        for (int j = 0; j < line.BoundingPolygon.Count; j++)
        {
            Console.WriteLine($"      Point {j} => X:
{line.BoundingPolygon[j].X}, Y: {line.BoundingPolygon[j].Y}");
        }
    }

    for (int i = 0; i < page.SelectionMarks.Count; i++)
    {
        DocumentSelectionMark selectionMark = page.SelectionMarks[i];

        Console.WriteLine($"  Selection Mark {i} is
{selectionMark.State}.");
        Console.WriteLine($"  Its bounding polygon (points ordered
clockwise):");

        for (int j = 0; j < selectionMark.BoundingPolygon.Count; j++)
        {
            Console.WriteLine($"      Point {j} => X:
{selectionMark.BoundingPolygon[j].X}, Y:
{selectionMark.BoundingPolygon[j].Y}");
        }
    }
}

```

```

foreach (DocumentStyle style in result.Styles)
{
    // Check the style and style confidence to see if text is handwritten.
    // Note that value '0.8' is used as an example.

    bool isHandwritten = style.IsHandwritten.HasValue && style.IsHandwritten
== true;

    if (isHandwritten && style.Confidence > 0.8)
    {
        Console.WriteLine($"Handwritten content found:");

        foreach (DocumentSpan span in style.Spans)
        {
            Console.WriteLine($"  Content:
{result.Content.Substring(span.Index, span.Length)}");
        }
    }
}

Console.WriteLine("The following tables were extracted:");

for (int i = 0; i < result.Tables.Count; i++)
{
    DocumentTable table = result.Tables[i];
    Console.WriteLine($"  Table {i} has {table.RowCount} rows and
{table.ColumnCount} columns.");

    foreach (DocumentTableCell cell in table.Cells)
    {
        Console.WriteLine($"    Cell ({cell.RowIndex}, {cell.ColumnIndex})
has kind '{cell.Kind}' and content: '{cell.Content}'.");
    }
}

```

For more information and samples see [here](#).

Use the Prebuilt Read Model

Analyze textual elements, such as page words and lines, styles, paragraphs, and text language information from documents using the prebuilt read model.

C#

```

Uri fileUri = new Uri("<fileUri>");

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

```

```

Console.WriteLine("Detected languages:");

foreach (DocumentLanguage language in result.Languages)
{
    Console.WriteLine($"  Found language with locale '{language.Locale}' and
confidence {language.Confidence}.");
}

foreach (DocumentPage page in result.Pages)
{
    Console.WriteLine($"Document Page {page.PageNumber} has
{page.Lines.Count} line(s), {page.Words.Count} word(s),");
    Console.WriteLine($"and {page.SelectionMarks.Count} selection
mark(s).");

    for (int i = 0; i < page.Lines.Count; i++)
    {
        DocumentLine line = page.Lines[i];
        Console.WriteLine($"  Line {i} has content: '{line.Content}'.");

        Console.WriteLine($"    Its bounding polygon (points ordered
clockwise):");

        for (int j = 0; j < line.BoundingPolygon.Count; j++)
        {
            Console.WriteLine($"      Point {j} => X:
{line.BoundingPolygon[j].X}, Y: {line.BoundingPolygon[j].Y}");
        }
    }
}

foreach (DocumentStyle style in result.Styles)
{
    // Check the style and style confidence to see if text is handwritten.
    // Note that value '0.8' is used as an example.

    bool isHandwritten = style.IsHandwritten.HasValue && style.IsHandwritten
== true;

    if (isHandwritten && style.Confidence > 0.8)
    {
        Console.WriteLine($"Handwritten content found:");

        foreach (DocumentSpan span in style.Spans)
        {
            Console.WriteLine($"  Content:
{result.Content.Substring(span.Index, span.Length)}");
        }
    }
}

```

For more information and samples see [here ↗](#).

Use Prebuilt Models

Analyze data from certain types of common documents using prebuilt models provided by the Document Intelligence service.

For example, to analyze fields from an invoice, use the prebuilt Invoice model provided by passing the `prebuilt-invoice` model ID into the `AnalyzeDocumentAsync` method:

C#

```
string filePath = "<filePath>";

using var stream = new FileStream(filePath, FileMode.Open);

AnalyzeDocumentOperation operation = await
client.AnalyzeDocumentAsync(WaitUntil.Completed, "prebuilt-invoice",
stream);
AnalyzeResult result = operation.Value;

// To see the list of all the supported fields returned by service and its
corresponding types for the
// prebuilt-invoice model, consult:
// https://aka.ms/azsdk/formrecognizer/invoicefieldschema

for (int i = 0; i < result.Documents.Count; i++)
{
    Console.WriteLine($"Document {i}:");

    AnalyzedDocument document = result.Documents[i];

    if (document.Fields.TryGetValue("VendorName", out DocumentField
vendorNameField))
    {
        if (vendorNameField.FieldType == DocumentFieldType.String)
        {
            string vendorName = vendorNameField.ValueAsString();
            Console.WriteLine($"Vendor Name: '{vendorName}', with confidence
{vendorNameField.Confidence}");
        }
    }

    if (document.Fields.TryGetValue("CustomerName", out DocumentField
customerNameField))
    {
        if (customerNameField.FieldType == DocumentFieldType.String)
        {
            string customerName = customerNameField.ValueAsString();
            Console.WriteLine($"Customer Name: '{customerName}', with
confidence {customerNameField.Confidence}");
        }
    }
}
```

```

if (document.Fields.TryGetValue("Items", out DocumentField itemsField))
{
    if (itemsField.FieldType == DocumentFieldType.List)
    {
        foreach (DocumentField itemField in itemsField.Value.AsList())
        {
            Console.WriteLine("Item:");

            if (itemField.FieldType == DocumentFieldType.Dictionary)
            {
                IReadOnlyDictionary<string, DocumentField> itemFields =
itemField.Value.AsDictionary();

                if (itemFields.TryGetValue("Description", out
DocumentField itemDescriptionField))
                {
                    if (itemDescriptionField.FieldType ==
DocumentFieldType.String)
                    {
                        string itemDescription =
itemDescriptionField.ValueAsString();

                        Console.WriteLine($" Description:
'{itemDescription}', with confidence {itemDescriptionField.Confidence}");
                    }
                }

                if (itemFields.TryGetValue("Amount", out DocumentField
itemAmountField))
                {
                    if (itemAmountField.FieldType ==
DocumentFieldType.Currency)
                    {
                        CurrencyValue itemAmount =
itemAmountField.Value.AsCurrency();

                        Console.WriteLine($" Amount:
'{itemAmount.Symbol}{itemAmount.Amount}', with confidence
{itemAmountField.Confidence}");
                    }
                }
            }
        }
    }
}

if (document.Fields.TryGetValue("SubTotal", out DocumentField
subTotalField))
{
    if (subTotalField.FieldType == DocumentFieldType.Currency)
    {
        CurrencyValue subTotal = subTotalField.Value.AsCurrency();
        Console.WriteLine($"Sub Total: '{subTotal.Symbol}
{subTotal.Amount}', with confidence {subTotalField.Confidence}");
    }
}

```

```

    }

    if (document.Fields.TryGetValue("TotalTax", out DocumentField
totalTaxField))
    {
        if (totalTaxField.FieldType == DocumentFieldType.Currency)
        {
            CurrencyValue totalTax = totalTaxField.Value.AsCurrency();
            Console.WriteLine($"Total Tax: '{totalTax.Symbol}
{totalTax.Amount}', with confidence {totalTaxField.Confidence}");
        }
    }

    if (document.Fields.TryGetValue("InvoiceTotal", out DocumentField
invoiceTotalField))
    {
        if (invoiceTotalField.FieldType == DocumentFieldType.Currency)
        {
            CurrencyValue invoiceTotal =
invoiceTotalField.Value.AsCurrency();
            Console.WriteLine($"Invoice Total: '{invoiceTotal.Symbol}
{invoiceTotal.Amount}', with confidence {invoiceTotalField.Confidence}");
        }
    }
}

```

You are not limited to invoices! There are a couple of prebuilt models to choose from, each of which has its own set of supported fields. More information about the supported document types can be found in the [service documentation](#).

For more information and samples, see [here](#).

Build a Custom Model

Build a custom model on your own document type. The resulting model can be used to analyze values from the types of documents it was built on.

C#

```

// For this sample, you can use the training documents found in the
`trainingFiles` folder.
// Upload the documents to your storage container and then generate a
container SAS URL. Note
// that a container URI without SAS is accepted only when the container is
public or has a
// managed identity configured.
//
// For instructions to set up documents for training in an Azure Blob
Storage Container, please see:
// https://aka.ms/azsdk/formrecognizer/buildcustommodel

```

```

Uri blobContainerUri = new Uri("<blobContainerUri>");
var client = new DocumentModelAdministrationClient(new Uri(endpoint), new
AzureKeyCredential(apiKey));

// We are selecting the Template build mode in this sample. For more
information about the available
// build modes and their differences, please see:
// https://aka.ms/azsdk/formrecognizer/buildmode

BuildDocumentModelOperation operation = await
client.BuildDocumentModelAsync(WaitUntil.Completed, blobContainerUri,
DocumentBuildMode.Template);
DocumentModelDetails model = operation.Value;

Console.WriteLine($" Model Id: {model.ModelId}");
Console.WriteLine($" Created on: {model.CreatedOn}");

Console.WriteLine(" Document types the model can recognize:");
foreach (KeyValuePair<string, DocumentTypeDetails> documentType in
model.DocumentTypes)
{
    Console.WriteLine($"     Document type: {documentType.Key} which has the
following fields:");
    foreach (KeyValuePair<string, DocumentFieldSchema> schema in
documentType.Value.FieldSchema)
    {
        Console.WriteLine($"         Field: {schema.Key} with confidence
{documentType.Value.FieldConfidence[schema.Key]}");
    }
}

```

For more information and samples see [here ↗](#).

Analyze Custom Documents

Analyze text, field values, selection marks, and table structures, styles, and paragraphs from custom documents, using models you built with your own document types.

C#

```

string modelId = "<modelId>";
Uri fileUri = new Uri("<fileUri>");

AnalyzeDocumentOperation operation = await
client.AnalyzeDocumentFromUriAsync(WaitUntil.Completed, modelId, fileUri);
AnalyzeResult result = operation.Value;

Console.WriteLine($"Document was analyzed with model with ID:
{result.ModelId}");

```

```

foreach (AnalyzedDocument document in result.Documents)
{
    Console.WriteLine($"Document of type: {document.DocumentType}");

    foreach (KeyValuePair<string, DocumentField> fieldKvp in
document.Fields)
    {
        string fieldName = fieldKvp.Key;
        DocumentField field = fieldKvp.Value;

        Console.WriteLine($"Field '{fieldName}': ");

        Console.WriteLine($"  Content: '{field.Content}'");
        Console.WriteLine($"  Confidence: '{field.Confidence}'");
    }
}

```

For more information and samples see [here ↗](#).

Manage Models

Manage the models stored in your account.

C#

```

var client = new DocumentModelAdministrationClient(new Uri(endpoint), new
AzureKeyCredential(apiKey));

// Check number of custom models in the Form Recognizer resource, and the
maximum number of custom models that can be stored.
ResourceDetails resourceDetails = await client.GetResourceDetailsAsync();
Console.WriteLine($"Resource has {resourceDetails.CustomDocumentModelCount}
custom models.");
Console.WriteLine($"It can have at most
{resourceDetails.CustomDocumentModelLimit} custom models.");

// List the first ten or fewer models currently stored in the resource.
AsyncPageable<DocumentModelSummary> models =
client.GetDocumentModelsAsync();

int count = 0;
await foreach (DocumentModelSummary modelSummary in models)
{
    Console.WriteLine($"Custom Model Summary:");
    Console.WriteLine($"  Model Id: {modelSummary.ModelId}");
    if (string.IsNullOrEmpty(modelSummary.Description))
        Console.WriteLine($"  Model description:
{modelSummary.Description}");
    Console.WriteLine($"  Created on: {modelSummary.CreatedOn}");
    if (++count == 10)
        break;
}

```

```

// Create a new model to store in the resource.
Uri blobContainerUri = new Uri("<blobContainerUri>");
BuildDocumentModelOperation operation = await
client.BuildDocumentModelAsync(WaitUntil.Completed, blobContainerUri,
DocumentBuildMode.Template);
DocumentModelDetails model = operation.Value;

// Get the model that was just created.
DocumentModelDetails newCreatedModel = await
client.GetDocumentModelAsync(model.ModelId);

Console.WriteLine($"Custom Model with Id {newCreatedModel.ModelId} has the
following information:");

Console.WriteLine($"  Model Id: {newCreatedModel.ModelId}");
if (string.IsNullOrEmpty(newCreatedModel.Description))
    Console.WriteLine($"  Model description:
{newCreatedModel.Description}");
Console.WriteLine($"  Created on: {newCreatedModel.CreatedOn}");

// Delete the model from the resource.
await client.DeleteDocumentModelAsync(newCreatedModel.ModelId);

```

For more information and samples see [here](#).

Manage Models Synchronously

Manage the models stored in your account with a synchronous API.

C#

```

var client = new DocumentModelAdministrationClient(new Uri(endpoint), new
AzureKeyCredential(apiKey));

// Check number of custom models in the Form Recognizer resource, and the
maximum number of custom models that can be stored.
ResourceDetails resourceDetails = client.GetResourceDetails();
Console.WriteLine($"Resource has {resourceDetails.CustomDocumentModelCount}
custom models.");
Console.WriteLine($"It can have at most
{resourceDetails.CustomDocumentModelLimit} custom models.");

// List the first ten or fewer models currently stored in the resource.
Pageable<DocumentModelSummary> models = client.GetDocumentModels();

foreach (DocumentModelSummary modelSummary in models.Take(10))
{
    Console.WriteLine($"Custom Model Summary:");
    Console.WriteLine($"  Model Id: {modelSummary.ModelId}");
    if (string.IsNullOrEmpty(modelSummary.Description))
        Console.WriteLine($"  Model description:
{modelSummary.Description}");
}

```

```

{modelSummary.Description}");  

    Console.WriteLine($"  Created on: {modelSummary.CreatedOn}");  

}  
  

// Create a new model to store in the resource.  
  

Uri blobContainerUri = new Uri("<blobContainerUri>");  

BuildDocumentModelOperation operation =  

client.BuildDocumentModel(WaitUntil.Completed, blobContainerUri,  

DocumentBuildMode.Template);  

DocumentModelDetails model = operation.Value;  
  

// Get the model that was just created.  

DocumentModelDetails newCreatedModel =  

client.GetDocumentModel(model.ModelId);  
  

Console.WriteLine($"Custom Model with Id {newCreatedModel.ModelId} has the  

following information:");  
  

Console.WriteLine($"  Model Id: {newCreatedModel.ModelId}");  

if (string.IsNullOrEmpty(newCreatedModel.Description))  

    Console.WriteLine($"  Model description:  

{newCreatedModel.Description}");  

Console.WriteLine($"  Created on: {newCreatedModel.CreatedOn}");  
  

// Delete the created model from the resource.  

client.DeleteDocumentModel(newCreatedModel.ModelId);

```

Build a Document Classifier

Build a document classifier by uploading custom training documents.

C#

```

// For this sample, you can use the training documents found in the  

`classifierTrainingFiles` folder.  

// Upload the documents to your storage container and then generate a  

container SAS URL. Note  

// that a container URI without SAS is accepted only when the container is  

public or has a  

// managed identity configured.  

//  

// For instructions to set up documents for training in an Azure Blob  

Storage Container, please see:  

// https://aka.ms/azsdk/formrecognizer/buildclassifiermodel  
  

Uri trainingFilesUri = new Uri("<trainingFilesUri>");  

var client = new DocumentModelAdministrationClient(new Uri(endpoint), new  

AzureKeyCredential(apiKey));  
  

var sourceA = new BlobContentSource(trainingFilesUri) { Prefix = "IRS-1040-  

A/train" };

```

```

var sourceB = new BlobContentSource(trainingFilesUri) { Prefix = "IRS-1040-B/train" };

var documentTypes = new Dictionary<string, ClassifierDocumentTypeDetails>()
{
    { "IRS-1040-A", new ClassifierDocumentTypeDetails(sourceA) },
    { "IRS-1040-B", new ClassifierDocumentTypeDetails(sourceB) }
};

BuildDocumentClassifierOperation operation = await
client.BuildDocumentClassifierAsync(WaitUntil.Completed, documentTypes);
DocumentClassifierDetails classifier = operation.Value;

Console.WriteLine($" Classifier Id: {classifier.ClassifierId}");
Console.WriteLine($" Created on: {classifier.CreatedOn}");

Console.WriteLine(" Document types the classifier can recognize:");
foreach (KeyValuePair<string, ClassifierDocumentTypeDetails> documentType in
classifier.DocumentTypes)
{
    Console.WriteLine($"      {documentType.Key}");
}

```

For more information and samples see [here ↗](#).

Classify a Document

Use document classifiers to accurately detect and identify documents you process within your application.

C#

```

string classifierId = "<classifierId>";
Uri fileUri = new Uri("<fileUri>");

ClassifyDocumentOperation operation = await
client ClassifyDocumentFromUriAsync(WaitUntil.Completed, classifierId,
fileUri);
AnalyzeResult result = operation.Value;

Console.WriteLine($"Document was classified by classifier with ID:
{result.ModelId}");

foreach (AnalyzedDocument document in result.Documents)
{
    Console.WriteLine($"Document of type: {document.DocumentType}");
}

```

For more information and samples see [here ↗](#).

Troubleshooting

General

When you interact with the Form Recognizer client library using the .NET SDK, errors returned by the service will result in a `RequestFailedException` with the same HTTP status code returned by the [REST API](#) request.

For example, if you submit a receipt image with an invalid `Uri`, a `400` error is returned, indicating "Bad Request".

C#

```
try
{
    AnalyzeDocumentOperation operation = await
    client.AnalyzeDocumentFromUriAsync(WaitUntil.Completed, "prebuilt-receipt",
    new Uri("http://invalid.uri"));
}
catch (RequestFailedException e)
{
    Console.WriteLine(e.ToString());
}
```

You will notice that additional information is logged, like the client request ID of the operation.

Message:

```
Azure.RequestFailedException: Service request failed.
Status: 400 (Bad Request)
ErrorCode: InvalidRequest
```

Content:

```
{"error":{"code":"InvalidRequest","message":"Invalid
request.","innererror":{"code":"InvalidContent","message":"The file is
corrupted or format is unsupported. Refer to documentation for the list of
supported formats."}}}
```

Headers:

```
Transfer-Encoding: chunked
x-envoy-upstream-service-time: REDACTED
apim-request-id: REDACTED
Strict-Transport-Security: REDACTED
X-Content-Type-Options: REDACTED
Date: Fri, 01 Oct 2021 02:55:44 GMT
Content-Type: application/json; charset=utf-8
```

Error codes and messages raised by the Document Intelligence service can be found in the [service documentation](#).

For more details about common issues, see our [troubleshooting guide](#).

Setting up console logging

The simplest way to see the logs is to enable the console logging. To create an Azure SDK log listener that outputs messages to console use the `AzureEventSourceListener.CreateConsoleLogger` method.

C#

```
// Setup a listener to monitor logged events.  
using AzureEventSourceListener listener =  
    AzureEventSourceListener.CreateConsoleLogger();
```

To learn more about other logging mechanisms see [Diagnostics Samples](#).

Next steps

Samples showing how to use the Form Recognizer library are available in this GitHub repository. Samples are provided for each main functional area:

- [Extract the layout of a document](#)
- [Analyze with the prebuilt general document model](#)
- [Analyze with the prebuilt read model](#)
- [Analyze a document with a custom model](#)
- [Analyze a document with a prebuilt model](#)
- [Build a custom model](#)
- [Manage models](#)
- [\[Classify a document\]\[classify_a_document\]](#)
- [Build a document classifier](#)
- [Get and List document model operations](#)
- [Compose a model](#)
- [Copy a custom model between Form Recognizer resources](#)
- [Mock a client for testing using the Moq library](#)

Note that these samples use SDK version `4.1.0`. For version 3.1.1 or lower, see [Form Recognizer Samples for V3.1.X](#).

Contributing

This project welcomes contributions and suggestions. Most contributions require you to agree to a Contributor License Agreement (CLA) declaring that you have the right to, and actually do, grant us the rights to use your contribution. For details, visit cla.microsoft.com.

When you submit a pull request, a CLA-bot will automatically determine whether you need to provide a CLA and decorate the PR appropriately (e.g., label, comment). Simply follow the instructions provided by the bot. You will only need to do this once across all repos using our CLA.

This project has adopted the [Microsoft Open Source Code of Conduct](#). For more information see the [Code of Conduct FAQ](#) or contact opencode@microsoft.com with any additional questions or comments.

Azure Document Intelligence client library for Java - version 4.1.5

Article • 02/21/2024

Azure Document Intelligence ([previously known as Form Recognizer](#)) is a cloud service that uses machine learning to analyze text and structured data from your documents. It includes the following main features:

- Layout - Extract text, table structures, and selection marks, along with their bounding region coordinates, from documents.
- Document - Analyze entities, key-value pairs, tables, and selection marks from documents using the general prebuilt document model.
- Prebuilt - Analyze data from certain types of common documents (such as receipts, invoices, business cards, identity documents or US W2 tax forms) using prebuilt models.
- Custom - Build custom models to extract text, field values, selection marks, and table data from documents. Custom models are built with your own data, so they're tailored to your documents.
- Read - Read information about textual elements, such as page words and lines in addition to text language information.
- Classifiers - Build custom classifiers to categorize documents into predefined classes.

[Source code](#) | [Package \(Maven\)](#) | [API reference documentation](#) | [Product Documentation](#) | [Samples](#)

Getting started

Prerequisites

- A [Java Development Kit \(JDK\)](#), version 8 or later.
- [Azure Subscription](#)
- [Cognitive Services or Form Recognizer account](#) to use this package.

Include the Package

Include the BOM file

Do include the azure-sdk-bom to your project to take dependency on GA version of the library. In the following snippet, replace the {bom_version_to_target} placeholder with the version number. To learn more about the BOM, see the [AZURE SDK BOM README](#).

XML

```
<dependencyManagement>
  <dependencies>
    <dependency>
      <groupId>com.azure</groupId>
      <artifactId>azure-sdk-bom</artifactId>
      <version>{bom_version_to_target}</version>
      <type>pom</type>
      <scope>import</scope>
    </dependency>
  </dependencies>
</dependencyManagement>
```

Then, include the direct dependency in the dependencies' section without the version tag.

XML

```
<dependencies>
  <dependency>
    <groupId>com.azure</groupId>
    <artifactId>azure-ai-formrecognizer</artifactId>
  </dependency>
</dependencies>
```

Include direct dependency

If you want to take dependency on a particular version of the library that is not present in the BOM, add the direct dependency to your project as follows.

XML

```
<dependency>
  <groupId>com.azure</groupId>
  <artifactId>azure-ai-formrecognizer</artifactId>
  <version>4.1.5</version>
</dependency>
```

Note: This version of the client library defaults to the "2023-07-31" version of the service.

This table shows the relationship between SDK versions and supported API versions of the service:

[+] Expand table

SDK version	Supported API version of service
3.0.x	2.0
3.1.X - 3.1.12	2.0, 2.1 (default)
4.0.0	2.0, 2.1, 2022-08-31 (default)
4.1.0	2.0, 2.1, 2022-08-31, 2023-07-31 (default)

Note: Starting with version 4.0.X, a new set of clients were introduced to leverage the newest features of the Form Recognizer service. Please see the [Migration Guide](#) for detailed instructions on how to update application code from client library version 3.1.X or lower to the latest version. For more information, see [Changelog](#). The below table describes the relationship of each client and its supported API version(s):

[+] Expand table

API version	Supported clients
2023-07-31	DocumentAnalysisClient and DocumentModelAdministrationClient
2022-08-31	DocumentAnalysisClient and DocumentModelAdministrationClient
2.1	FormRecognizerClient and FormTrainingClient
2.0	FormRecognizerClient and FormTrainingClient

Create a Form Recognizer resource

Form Recognizer supports both [multi-service and single-service access](#). Create a Cognitive Service's resource if you plan to access multiple cognitive services under a single endpoint/key. For Form Recognizer access only, create a Form Recognizer resource.

You can create either resource using the

Option 1: [Azure portal](#)

Option 2: [Azure CLI](#)

Below is an example of how you can create a Form Recognizer resource using the CLI:

Bash

```
# Create a new resource group to hold the Form Recognizer resource -  
# if using an existing resource group, skip this step  
az group create --name <your-resource-group> --location <location>
```

Bash

```
# Create Form Recognizer  
az cognitiveservices account create \  
    --name <your-form-recognizer-resource-name> \  
    --resource-group <your-resource-group> \  
    --kind FormRecognizer \  
    --sku <sku> \  
    --location <location> \  
    --yes
```

Authenticate the client

In order to interact with the Form Recognizer service, you will need to create an instance of the Document Analysis client. Both the asynchronous and synchronous clients can be created by using `DocumentAnalysisClientBuilder`. Invoking `buildClient()` will create the synchronous client, while invoking `buildAsyncClient` will create its asynchronous counterpart.

You will need an `endpoint`, and a `key` to instantiate a client object.

Looking up the endpoint

You can find the `endpoint` for your Form Recognizer resource in the [Azure portal](#), or [Azure CLI](#).

Bash

```
# Get the endpoint for the resource  
az cognitiveservices account show --name "resource-name" --resource-group  
"resource-group-name" --query "endpoint"
```

Create a Document Analysis client using AzureKeyCredential

To use `AzureKeyCredential` authentication, provide the `key` as a string to the [AzureKeyCredential](#). This key can be found in the [Azure portal](#) in your created Form Recognizer resource, or by running the following Azure CLI command to get the key from the Form Recognizer resource:

Bash

```
az cognitiveservices account keys list --resource-group <your-resource-group-name> --name <your-resource-name>
```

Use the API key as the credential parameter to authenticate the client:

Java

```
DocumentAnalysisClient documentAnalysisClient = new DocumentAnalysisClientBuilder()
    .credential(new AzureKeyCredential("{key}"))
    .endpoint("{endpoint}")
    .buildClient();
```

Java

```
DocumentModelAdministrationClient client =
    new DocumentModelAdministrationClientBuilder()
        .credential(new AzureKeyCredential("{key}"))
        .endpoint("{endpoint}")
        .buildClient();
```

Create a Document Analysis client with Azure Active Directory credential

Azure SDK for Java supports an Azure Identity package, making it easy to get credentials from Microsoft identity platform.

Authentication with AAD requires some initial setup:

- Add the Azure Identity package

XML

```
<dependency>
    <groupId>com.azure</groupId>
    <artifactId>azure-identity</artifactId>
    <version>1.10.0</version>
</dependency>
```

- Register a new Azure Active Directory application
- Grant access to Form Recognizer by assigning the "Cognitive Services User" role to your service principal.

After the setup, you can choose which type of [credential](#) from `azure-identity` to use. As an example, [DefaultAzureCredential](#) can be used to authenticate the client: Set the values of the client ID, tenant ID, and client secret of the AAD application as environment variables: `AZURE_CLIENT_ID`, `AZURE_TENANT_ID`, `AZURE_CLIENT_SECRET`.

Authorization is easiest using [DefaultAzureCredential](#). It finds the best credential to use in its running environment. For more information about using Azure Active Directory authorization with Form Recognizer, see [the associated documentation](#).

Java

```
DocumentAnalysisClient documentAnalysisClient = new  
DocumentAnalysisClientBuilder()  
    .endpoint("{endpoint}")  
    .credential(new DefaultAzureCredentialBuilder().build())  
    .buildClient();
```

Key concepts

DocumentAnalysisClient

The [DocumentAnalysisClient](#) and [DocumentAnalysisAsyncClient](#) provide both synchronous and asynchronous operations for analyzing input documents using custom and prebuilt models through the `beginAnalyzeDocument` and `beginAnalyzeDocumentFromUrl` methods. See a full list of supported models [here](#).

Sample code snippets to illustrate using a `DocumentAnalysisClient` [here](#). More information about analyzing documents, including supported features, locales, and document types can be found [here](#).

DocumentModelAdministrationClient

The [DocumentModelAdministrationClient](#) and [DocumentModelAdministrationAsyncClient](#) provide both synchronous and asynchronous operations

- Build custom document analysis models to analyze text content, fields, and values found in your custom documents. See example [Build a document model](#). A

`DocumentModelDetails` is returned indicating the document types that the model can analyze, along with the fields and schemas it will extract.

- Managing models created in your account by building, listing, deleting, and see the limit of custom models your account. See example [Manage models](#).
- Copying a custom model from one Form Recognizer resource to another.
- Creating a composed model from a collection of existing built models.
- Listing document model operations associated with the Form Recognizer resource.

Sample code snippets are provided to illustrate using a `DocumentModelAdministrationClient` [here](#).

Long-running operations

Long-running operations are operations that consist of an initial request sent to the service to start an operation, followed by polling the service at intervals to determine whether the operation has completed or failed, and if it has succeeded, to get the result.

Methods that build models, analyze values from documents, or copy and compose models are modeled as long-running operations. The client exposes a `begin<MethodName>` method that returns a `SyncPoller` or `PollerFlux` instance. Callers should wait for the operation to be completed by calling `getFinalResult()` on the returned operation from the `begin<MethodName>` method. Sample code snippets are provided to illustrate using long-running operations [below](#).

Examples

The following section provides several code snippets covering some of the most common Form Recognizer tasks, including:

- [Extract Layout](#)
- [Use a General Document Model](#)
- [Use Prebuilt Models](#)
- [Build a Document Model](#)
- [Analyze Documents using a Custom Model](#)
- [Manage Your Models](#)
- [Classify a document](#)

Extract Layout

Extract text, table structures, and selection marks like radio buttons and check boxes, along with their bounding box coordinates from documents without the need to build a

model.

Java

```
// analyze document layout using file input stream
File layoutDocument = new File("local/file_path/filename.png");
Path filePath = layoutDocument.toPath();
BinaryData layoutDocumentData = BinaryData.fromFile(filePath, (int) layoutDocument.length());

SyncPoller<OperationResult, AnalyzeResult> analyzeLayoutResultPoller =
    documentAnalysisClient.beginAnalyzeDocument("prebuilt-layout",
layoutDocumentData);

AnalyzeResult analyzeLayoutResult =
analyzeLayoutResultPoller.getFinalResult();

// pages
analyzeLayoutResult.getPages().forEach(documentPage -> {
    System.out.printf("Page has width: %.2f and height: %.2f, measured with
unit: %s%n",
        documentPage.getWidth(),
        documentPage.getHeight(),
        documentPage.getUnit());

    // lines
    documentPage.getLines().forEach(documentLine ->
        System.out.printf("Line '%s' is within a bounding box %s.%n",
            documentLine.getContent(),
            documentLine.getBoundingPolygon().toString()));

    // selection marks
    documentPage.getSelectionMarks().forEach(documentSelectionMark ->
        System.out.printf("Selection mark is '%s' and is within a bounding
box %s with confidence %.2f.%n",
            documentSelectionMark.getSelectionMarkState().toString(),
            documentSelectionMark.getBoundingPolygon().toString(),
            documentSelectionMark.getConfidence()));
});

// tables
List<DocumentTable> tables = analyzeLayoutResult.getTables();
for (int i = 0; i < tables.size(); i++) {
    DocumentTable documentTable = tables.get(i);
    System.out.printf("Table %d has %d rows and %d columns.%n", i,
documentTable.getRowCount(),
        documentTable.getColumnCount());
    documentTable.getCells().forEach(documentTableCell -> {
        System.out.printf("Cell '%s', has row index %d and column index
%d.%n", documentTableCell.getContent(),
            documentTableCell.getRowIndex(),
            documentTableCell.getColumnIndex());
    });
}
```

```
        System.out.println();
    }
```

Use a General Document Model

Analyze key-value pairs, tables, styles, and selection marks from documents using the general document model provided by the Form Recognizer service. Select the General Document Model by passing modelId="prebuilt-document" into the beginAnalyzeDocumentFromUrl method as follows:

Java

```
String documentUrl = "{document-url}";
String modelId = "prebuilt-document";
SyncPoller<OperationResult, AnalyzeResult> analyzeDocumentPoller =
    documentAnalysisClient.beginAnalyzeDocumentFromUrl(modelId,
documentUrl);

AnalyzeResult analyzeResult = analyzeDocumentPoller.getFinalResult();

for (int i = 0; i < analyzeResult.getDocuments().size(); i++) {
    final AnalyzedDocument analyzedDocument =
analyzeResult.getDocuments().get(i);
    System.out.printf("----- Analyzing document %d -----%n", i);
    System.out.printf("Analyzed document has doc type %s with confidence :
%.2f%n",
        analyzedDocument.getDocType(), analyzedDocument.getConfidence());
}

analyzeResult.getPages().forEach(documentPage -> {
    System.out.printf("Page has width: %.2f and height: %.2f, measured with
unit: %s%n",
        documentPage.getWidth(),
        documentPage.getHeight(),
        documentPage.getUnit());

    // lines
    documentPage.getLines().forEach(documentLine ->
        System.out.printf("Line '%s' is within a bounding box %s.%n",
            documentLine.getContent(),
            documentLine.getBoundingPolygon().toString()));

    // words
    documentPage.getWords().forEach(documentWord ->
        System.out.printf("Word '%s' has a confidence score of %.2f.%n",
            documentWord.getContent(),
            documentWord.getConfidence()));
});

// tables
List<DocumentTable> tables = analyzeResult.getTables();
```

```

for (int i = 0; i < tables.size(); i++) {
    DocumentTable documentTable = tables.get(i);
    System.out.printf("Table %d has %d rows and %d columns.%n", i,
documentTable.getRowCount(),
        documentTable.getColumnCount());
    documentTable.getCells().forEach(documentTableCell -> {
        System.out.printf("Cell '%s', has row index %d and column index
%d.%n",
            documentTableCell.getContent(),
            documentTableCell.getRowIndex(),
documentTableCell.getColumnIndex());
    });
    System.out.println();
}

// Key-value
analyzeResult.getKeyValuePairs().forEach(documentKeyValuePair -> {
    System.out.printf("Key content: %s%n",
documentKeyValuePair.getKey().getContent());
    System.out.printf("Key content bounding region: %s%n",
        documentKeyValuePair.getKey().getBoundingRegions().toString());

    System.out.printf("Value content: %s%n",
documentKeyValuePair.getValue().getContent());
    System.out.printf("Value content bounding region: %s%n",
        documentKeyValuePair.getValue().getBoundingRegions().toString());
});

```

Use Prebuilt Models

Extract fields from select document types such as receipts, invoices, business cards, and identity documents using prebuilt models provided by the Form Recognizer service.

Supported prebuilt models are:

- Analyze receipts using the `prebuilt-receipt` model (fields recognized by the service can be found [here](#)).
- Analyze business cards using the `prebuilt-businessCard` model (fields recognized by the service can be found [here](#)).
- Analyze invoices using the `prebuilt-invoice` model (fields recognized by the service can be found [here](#)).
- Analyze identity documents using the `prebuilt-idDocuments` model (fields recognized by the service can be found [here](#)).
- Analyze US W2 tax forms using the `prebuilt-tax.us.w2` model. [Supported fields](#).

For example, to analyze fields from a sales receipt, into the `beginAnalyzeDocumentFromUrl` method:

Java

```
String receiptUrl = "https://raw.githubusercontent.com/Azure/azure-sdk-for-java/main/sdk/formrecognizer"
    + "/azure-ai-formrecognizer/src/samples/resources/sample-documents/receipts/contoso-allinone.jpg";

SyncPoller<OperationResult, AnalyzeResult> analyzeReceiptPoller =
    documentAnalysisClient.beginAnalyzeDocumentFromUrl("prebuilt-receipt",
receiptUrl);

AnalyzeResult receiptResults = analyzeReceiptPoller.getFinalResult();

for (int i = 0; i < receiptResults.getDocuments().size(); i++) {
    AnalyzedDocument analyzedReceipt = receiptResults.getDocuments().get(i);
    Map<String, DocumentField> receiptFields = analyzedReceipt.getFields();
    System.out.printf("----- Analyzing receipt info %d -----%n",
i);
    DocumentField merchantNameField = receiptFields.get("MerchantName");
    if (merchantNameField != null) {
        if (DocumentFieldType.STRING == merchantNameField.getType()) {
            String merchantName = merchantNameField.getValueAsString();
            System.out.printf("Merchant Name: %s, confidence: %.2f%n",
                merchantName, merchantNameField.getConfidence());
        }
    }

    DocumentField merchantPhoneNumberField =
receiptFields.get("MerchantPhoneNumber");
    if (merchantPhoneNumberField != null) {
        if (DocumentFieldType.PHONE_NUMBER ==
merchantPhoneNumberField.getType()) {
            String merchantAddress =
merchantPhoneNumberField.getValueAsPhoneNumber();
            System.out.printf("Merchant Phone number: %s, confidence:
%.2f%n",
                merchantAddress, merchantPhoneNumberField.getConfidence());
        }
    }

    DocumentField transactionDateField =
receiptFields.get("TransactionDate");
    if (transactionDateField != null) {
        if (DocumentFieldType.DATE == transactionDateField.getType()) {
            LocalDate transactionDate =
transactionDateField.getValueAsDate();
            System.out.printf("Transaction Date: %s, confidence: %.2f%n",
                transactionDate, transactionDateField.getConfidence());
        }
    }

    DocumentField receiptItemsField = receiptFields.get("Items");
    if (receiptItemsField != null) {
        System.out.printf("Receipt Items: %n");
```

```

    if (DocumentFieldType.LIST == receiptItemsField.getType()) {
        List<DocumentField> receiptItems =
receiptItemsField.getValueAsList();
        receiptItems.stream()
            .filter(receiptItem -> DocumentFieldType.MAP ==
receiptItem.getType())
            .map(documentField -> documentField.getValueAsMap())
            .forEach(documentFieldMap -> documentFieldMap.forEach((key,
documentField) -> {
            if ("Name".equals(key)) {
                if (DocumentFieldType.STRING ==
documentField.getType()) {
                    String name = documentField.getValueAsString();
                    System.out.printf("Name: %s, confidence:
%.2fs%n",
                        name, documentField.getConfidence());
                }
            }
            if ("Quantity".equals(key)) {
                if (DocumentFieldType.DOUBLE ==
documentField.getType()) {
                    Double quantity =
documentField.getValueAsDouble();
                    System.out.printf("Quantity: %f, confidence:
%.2f%n",
                        quantity, documentField.getConfidence());
                }
            }
        }));
    }
}

```

For more information and samples using prebuilt models, see:

- [Business Cards ↗](#)
- [Identity Documents ↗](#)
- [Invoices ↗](#)
- [Receipts sample ↗](#)

Build a document model

Build a machine-learned model on your own document type. The resulting model will be able to analyze values from the types of documents it was built on. Provide a container SAS url to your Azure Storage Blob container where you're storing the training documents. See details on setting this up in the [service quickstart documentation](#).

Note

You can use the [Form Recognizer Studio preview](#) for creating a labeled file for your training forms. More details on setting up a container and required file structure can be found in the [here](#).

Java

```
// Build custom document analysis model
String blobContainerUrl = "{SAS_URL_of_your_container_in_blob_storage}";
// The shared access signature (SAS) Url of your Azure Blob Storage
// container with your forms.
String prefix = "{blob_name_prefix}";
SyncPoller<OperationResult, DocumentModelDetails> buildOperationPoller =
    documentModelAdminClient.beginBuildDocumentModel(blobContainerUrl,
        DocumentModelBuildMode.TEMPLATE,
        prefix,
        new BuildDocumentModelOptions().setModelId("my-build-
model").setDescription("model desc"),
        Context.NONE);

DocumentModelDetails documentModelDetails =
buildOperationPoller.getFinalResult();

// Model Info
System.out.printf("Model ID: %s%n", documentModelDetails.getModelId());
System.out.printf("Model Description: %s%n",
documentModelDetails.getDescription());
System.out.printf("Model created on: %s%n%n",
documentModelDetails.getCreatedOn());
documentModelDetails.getDocumentTypes().forEach((key, documentTypeDetails) -
> {
    System.out.printf("Document type: %s%n", key);
    documentTypeDetails.getFieldSchema().forEach((name, documentFieldSchema)
-> {
        System.out.printf("Document field: %s%n", name);
        System.out.printf("Document field type: %s%n",
documentFieldSchema.getType().toString());
        System.out.printf("Document field confidence: %.2f%n",
documentTypeDetails.getFieldConfidence().get(name));
    });
});
```

Analyze Documents using a Custom Model

Analyze the key/value pairs and table data from documents. These models are built with your own data, so they're tailored to your documents. You should only analyze documents of the same doc type that the custom model was built on.

Java

```

String documentUrl = "{document-url}";
String modelId = "{custom-built-model-ID}";
SyncPoller<OperationResult, AnalyzeResult> analyzeDocumentPoller =
    documentAnalysisClient.beginAnalyzeDocumentFromUrl(modelId,
documentUrl);

AnalyzeResult analyzeResult = analyzeDocumentPoller.getFinalResult();

for (int i = 0; i < analyzeResult.getDocuments().size(); i++) {
    final AnalyzedDocument analyzedDocument =
analyzeResult.getDocuments().get(i);
    System.out.printf("----- Analyzing custom document %d -----%n",
i);
    System.out.printf("Analyzed document has doc type %s with confidence :
%.2f%n",
        analyzedDocument.getDocType(), analyzedDocument.getConfidence());
    analyzedDocument.getFields().forEach((key, documentField) -> {
        System.out.printf("Document Field content: %s%n",
documentField.getContent());
        System.out.printf("Document Field confidence: %.2f%n",
documentField.getConfidence());
        System.out.printf("Document Field Type: %s%n",
documentField.getType());
        System.out.printf("Document Field found within bounding region:
%s%n",
            documentField.getBoundingRegions().toString());
    });
}

analyzeResult.getPages().forEach(documentPage -> {
    System.out.printf("Page has width: %.2f and height: %.2f, measured with
unit: %s%n",
        documentPage.getWidth(),
        documentPage.getHeight(),
        documentPage.getUnit());

    // lines
    documentPage.getLines().forEach(documentLine ->
        System.out.printf("Line '%s' is within a bounding box %s.%n",
            documentLine.getContent(),
            documentLine.getBoundingPolygon().toString()));

    // words
    documentPage.getWords().forEach(documentWord ->
        System.out.printf("Word '%s' has a confidence score of %.2f.%n",
            documentWord.getContent(),
            documentWord.getConfidence()));
});

// tables
List<DocumentTable> tables = analyzeResult.getTables();
for (int i = 0; i < tables.size(); i++) {
    DocumentTable documentTable = tables.get(i);
    System.out.printf("Table %d has %d rows and %d columns.%n", i,

```

```

documentTable.getRowCount(),
    documentTable.getColumnCount());
documentTable.getCells().forEach(documentTableCell -> {
    System.out.printf("Cell '%s', has row index %d and column index
%d.%n",
        documentTableCell.getContent(),
        documentTableCell.getRowIndex(),
documentTableCell.getColumnIndex());
});
System.out.println();
}

```

Manage your models

Manage the models in your Form Recognizer account.

Java

```

AtomicReference<String> modelId = new AtomicReference<>();

// First, we see how many models we have, and what our limit is
ResourceDetails resourceDetails =
documentModelAdminClient.getResourceDetails();
System.out.printf("The resource has %s models, and we can have at most %
models",
    resourceDetails.getCustomDocumentModelCount(),
resourceDetails.getCustomDocumentModelLimit());

// Next, we get a paged list of all of our models
PagedIterable<DocumentModelSummary> customDocumentModels =
documentModelAdminClient.listDocumentModels();
System.out.println("We have following models in the account:");
customDocumentModels.forEach(documentModelSummary -> {
    System.out.printf("Model ID: %s%n", documentModelSummary.getModelId());
    modelId.set(documentModelSummary.getModelId());

    // get custom document analysis model info
    DocumentModelDetails documentModel =
documentModelAdminClient.getDocumentModel(documentModelSummary.getModelId())
;
    System.out.printf("Model ID: %s%n", documentModel.getModelId());
    System.out.printf("Model Description: %s%n",
documentModel.getDescription());
    System.out.printf("Model created on: %s%n",
documentModel.getCreatedOn());
    documentModel.getDocumentTypes().forEach((key, documentTypeDetails) -> {
        documentTypeDetails.getFieldSchema().forEach((field,
documentFieldSchema) -> {
            System.out.printf("Field: %s", field);
            System.out.printf("Field type: %s",
documentFieldSchema.getType());
            System.out.printf("Field confidence: %.2f",

```

```
documentTypeDetails.getFieldConfidence().get(field));
    });
});
});

// Delete Model
documentModelAdminClient.deleteDocumentModel(modelId.get());
```

Classify a document

The Form Recognizer service supports custom document classifiers that can classify documents into a set of predefined categories based on a training data set. Documents can be classified with a custom classifier using the `beginClassifyDocument` or `beginClassifyDocumentFromUrl` method of `DocumentAnalysisClient`. The following sample shows how to classify a document using a custom classifier:

Java

```
String documentUrl = "{file_source_url}";
String classifierId = "{custom_trained_classifier_id}";

documentAnalysisClient.beginClassifyDocumentFromUrl(classifierId,
documentUrl, Context.NONE)
    .getFinalResult()
    .getDocuments()
    .forEach(analyzedDocument -> System.out.printf("Doc Type: %s%n",
analyzedDocument.getDocType()));
```

For more detailed examples, refer to [samples](#).

Troubleshooting

General

Form Recognizer clients raise `HttpResponseException` [exceptions](#). For example, if you try to provide an invalid file source URL an `HttpResponseException` would be raised with an error indicating the failure cause. In the following code snippet, the error is handled gracefully by catching the exception and display the additional information about the error.

Java

```
try {
    documentAnalysisClient.beginAnalyzeDocumentFromUrl("prebuilt-receipt",
```

```
"invalidSourceUrl");  
} catch (HttpResponseException e) {  
    System.out.println(e.getMessage());  
    // Do something with the exception  
}
```

Enable client logging

Azure SDKs for Java offer a consistent logging story to help aid in troubleshooting application errors and expedite their resolution. The logs produced will capture the flow of an application before reaching the terminal state to help locate the root issue. View the [logging](#) wiki for guidance about enabling logging.

Default HTTP Client

All client libraries by default use the Netty HTTP client. Add the above dependency to automatically configure the client library to use the Netty HTTP client. Configuring or changing the HTTP client is detailed in the [HTTP clients](#) wiki.

Next steps

The following section provides several code snippets illustrating common patterns used in the Form Recognizer API. These code samples show common scenario operations with the Azure Form Recognizer client library.

- Analyze business card from a URL: [AnalyzeBusinessCardFromUrl](#)
- Analyze identity documents from a URL: [AnalyzeIdentityDocumentsFromUrl](#)
- Analyze invoice from a URL: [AnalyzeInvoiceFromUrl](#)
- Analyze receipts: [AnalyzeReceipts](#)
- Analyze receipts from a URL: [AnalyzeReceiptsFromUrl](#)
- Extract layout: [AnalyzeLayout](#)
- Analyze custom documents from a URL: [AnalyzeCustomDocumentFromUrl](#)
- Build a model: [BuildModel](#)
- Manage custom models: [ManageCustomModels](#)
- Copy a model between Form Recognizer resources: [CopyModel](#)
- Create a composed model from a collection of custom-built models: [ComposeModel](#)
- Get/List document model operations associated with the Form Recognizer resource: [GetOperation](#)
- Build a document classifier : [BuildDocumentClassifier](#)

Async APIs

All the examples shown so far have been using synchronous APIs, but we provide full support for async APIs as well. You'll need to use `DocumentAnalysisAsyncClient`

Java

```
DocumentAnalysisAsyncClient documentAnalysisAsyncClient = new  
DocumentAnalysisClientBuilder()  
    .credential(new AzureKeyCredential("{key}"))  
    .endpoint("{endpoint}")  
    .buildAsyncClient();
```

- Analyze business card from a URL: [AnalyzeBusinessCardFromUrlAsync ↗](#)
- Analyze identity documents from a URL: [AnalyzeIdentityDocumentsFromUrlAsync ↗](#)
- Analyze invoice: [AnalyzeInvoiceAsync ↗](#)
- Analyze receipts: [AnalyzeReceiptsAsync ↗](#)
- Analyze receipts from a URL: [AnalyzeReceiptsFromUrlAsync ↗](#)
- Extract layout from a URL: [AnalyzeLayoutFromUrlAsync ↗](#)
- Analyze custom documents: [AnalyzeCustomDocumentAsync ↗](#)
- Build a document model: [BuildModelAsync ↗](#)
- Manage custom models: [ManageCustomModelsAsync ↗](#)
- Copy a document model between Form Recognizer resources: [CopyModelAsync ↗](#)
- Create a composed document model from a collection of custom-built models: [ComposeModelAsync ↗](#)
- Get/List document model operations associated with the Form Recognizer resource: [GetOperationAsync ↗](#)
- Build a document classifier : [BuildDocumentClassifierAsync ↗](#)

Additional documentation

See the [Sample README ↗](#) for several code snippets illustrating common patterns used in the Form Recognizer Java SDK. For more extensive documentation on Azure Cognitive Services Form Recognizer, see the [Form Recognizer documentation ↗](#).

Contributing

This project welcomes contributions and suggestions. Most contributions require you to agree to a [Contributor License Agreement \(CLA\) ↗](#) declaring that you have the right to, and actually do, grant us the rights to use your contribution.

When you submit a pull request, a CLA-bot will automatically determine whether you need to provide a CLA and decorate the PR appropriately (e.g., label, comment). Simply follow the instructions provided by the bot. You will only need to do this once across all repos using our CLA.

This project has adopted the [Microsoft Open Source Code of Conduct](#). For more information, see the [Code of Conduct FAQ](#) or contact opencode@microsoft.com with any additional questions or comments.

 **Collaborate with us on GitHub**

The source for this content can be found on GitHub, where you can also create and review issues and pull requests. For more information, see [our contributor guide](#).



Azure SDK for Java feedback

Azure SDK for Java is an open source project. Select a link to provide feedback:

 [Open a documentation issue](#)

 [Provide product feedback](#)

Python용 Azure Form Recognizer 클라이언트 라이브러리 - 버전 3.3.2

아티클 • 2023. 11. 09.

Azure Document Intelligence(이전에는 Form Recognizer라고 [함](#))는 기계 학습을 사용하여 문서에서 텍스트 및 구조화된 데이터를 분석하는 클라우드 서비스입니다. 여기에는 다음과 같은 기본 기능이 포함됩니다.

- 레이아웃 - 문서에서 콘텐츠 및 구조(예: 단어, 선택 표시, 테이블)를 추출합니다.
- 문서 - 문서의 일반 레이아웃 외에도 키-값 쌍을 분석합니다.
- 읽기 - 문서에서 페이지 정보를 읽습니다.
- 미리 빌드 - 미리 빌드된 모델을 사용하여 선택한 문서 유형(예: 영수증, 청구서, 명함, ID 문서, 미국 W-2 세금 문서 등)에서 공통 필드 값을 추출합니다.
- 사용자 지정 - 사용자 고유의 데이터에서 사용자 지정 모델을 빌드하여 문서에서 일반 레이아웃 외에도 맞춤형 필드 값을 추출합니다.
- 분류자 - 레이아웃 및 언어 기능을 결합하여 애플리케이션 내에서 처리하는 문서를 정확하게 검색하고 식별하는 사용자 지정 분류 모델을 빌드합니다.
- 추가 기능 - 바코드/QR 코드, 수식, 글꼴/스타일 등을 추출하거나 선택적 매개 변수가 있는 큰 문서에 대해 고해상도 모드를 사용하도록 설정합니다.

[소스 코드](#) | [패키지\(PyPI\)](#) | [패키지\(Conda\)](#) | [API 참조 설명서](#) | [제품 설명서](#) | [샘플](#)

시작

필수 구성 요소

- 이 패키지를 사용하려면 Python 3.7 이상이 필요합니다.
- 이 패키지를 사용하려면 [Azure 구독](#)과 [Cognitive Services](#) 또는 [Form Recognizer 리소스](#)가 있어야 합니다.

패키지 설치

[pip](#)를 사용하여 Python용 Azure Form Recognizer 클라이언트 라이브러리를 설치합니다.

Bash

```
pip install azure-ai-formrecognizer
```

참고: 이 버전의 클라이언트 라이브러리는 기본적으로 서비스 버전으로 설정됩니다

2023-07-31 .

다음 표에서는 SDK 버전 및 지원되는 API 버전의 서비스 간 관계를 보여 줍니다.

SDK 버전	지원되는 API 버전의 서비스
3.3.X - 최신 GA 릴리스	2.0, 2.1, 2022-08-31, 2023-07-31(기본값)
3.2.X	2.0, 2.1, 2022-08-31(기본값)
3.1.X	2.0, 2.1(기본값)
3.0.0	2.0

참고: 버전 3.2.x부터 Document Intelligence 서비스의 최신 기능을 활용하기 위해 새 클라이언트 집합이 도입되었습니다. 클라이언트 라이브러리 버전 이하에서 최신 버전 3.1.x 으로 애플리케이션 코드를 업데이트하는 방법에 대한 자세한 지침은 [마이그레이션 가이드](#) 를 참조하세요. 또한 자세한 내용은 [Changelog](#) 를 참조하세요. 아래 표에서는 각 클라이언트와 지원되는 API 버전의 관계를 설명합니다.

API 버전	지원되는 클라이언트
2023-07-31	DocumentAnalysisClient 및 DocumentModelAdministrationClient
2022-08-31	DocumentAnalysisClient 및 DocumentModelAdministrationClient
2.1	FormRecognizerClient 및 FormTrainingClient
2.0	FormRecognizerClient 및 FormTrainingClient

Cognitive Services 또는 Form Recognizer 리소스 만들기

Document Intelligence는 [다중 서비스 및 단일 서비스 액세스를](#) 모두 지원합니다. 단일 엔드포인트/키에서 여러 Cognitive Services에 액세스하려는 경우 Cognitive Services 리소스를 만듭니다. 문서 인텔리전스 액세스에만 Form Recognizer 리소스를 만듭니다. [Azure Active Directory 인증](#)을 사용하려는 경우 단일 서비스 리소스가 필요합니다.

다음을 사용하여 리소스를 만들 수 있습니다.

- 옵션 1: [Azure Portal](#).
- 옵션 2: [Azure CLI](#).

다음은 CLI를 사용하여 Form Recognizer 리소스를 만드는 방법의 예입니다.

PowerShell

```
# Create a new resource group to hold the Form Recognizer resource
# if using an existing resource group, skip this step
az group create --name <your-resource-name> --location <location>
```

PowerShell

```
# Create form recognizer
az cognitiveservices account create \
    --name <your-resource-name> \
    --resource-group <your-resource-group-name> \
    --kind FormRecognizer \
    --sku <sku> \
    --location <location> \
    --yes
```

리소스를 만드는 방법 또는 위치 및 sku 정보를 가져오는 방법에 대한 자세한 내용은 여기를 참조 [하세요](#).

클라이언트 인증

Document Intelligence 서비스와 상호 작용하려면 클라이언트의 instance 만들어야 합니다. 클라이언트 개체를 인스턴스화하려면 [엔드포인트 및 자격 증명](#) 이 필요합니다.

엔드포인트 가져오기

Azure Portal 또는 Azure CLI를 사용하여 Form Recognizer 리소스에 대한 엔드포인트를 찾을 수 있습니다.

Bash

```
# Get the endpoint for the Form Recognizer resource
az cognitiveservices account show --name "resource-name" --resource-group
"resource-group-name" --query "properties.endpoint"
```

지역 엔드포인트 또는 사용자 지정 하위 도메인을 인증에 사용할 수 있습니다. 형식은 다음과 같이 지정됩니다.

```
Regional endpoint: https://<region>.api.cognitive.microsoft.com/
Custom subdomain: https://<resource-name>.cognitiveservices.azure.com/
```

지역 엔드포인트는 지역의 모든 리소스에 대해 동일합니다. 지원되는 지역 엔드포인트의 전체 목록은 [여기에서](#) 참조할 수 있습니다. 지역 엔드포인트는 AAD 인증을 지원하지 않습니다.

반면에 사용자 지정 하위 도메인은 Form Recognizer 리소스에 고유한 이름입니다. [단일 서비스 리소스](#) 에서만 사용할 수 있습니다.

API 키 가져오기

API 키는 [Azure Portal](#) 에서 또는 다음 Azure CLI 명령을 실행하여 찾을 수 있습니다.

Bash

```
az cognitiveservices account keys list --name "<resource-name>" --resource-group "<resource-group-name>"
```

AzureKeyCredential을 사용하여 클라이언트 만들기

API 키를 매개 변수로 `credential` 사용하려면 [AzureKeyCredential](#)의 instance 키를 문자열로 전달합니다.

Python

```
from azure.core.credentials import AzureKeyCredential
from azure.ai.formrecognizer import DocumentAnalysisClient

endpoint = "https://<my-custom-subdomain>.cognitiveservices.azure.com/"
credential = AzureKeyCredential("<api_key>")
document_analysis_client = DocumentAnalysisClient(endpoint, credential)
```

Azure Active Directory 자격 증명을 사용하여 클라이언트 만들기

`AzureKeyCredential` 인증은 이 시작 가이드의 예제에서 사용되지만 [azure-identity](#) 라이브러리를 사용하여 Azure Active Directory로 인증할 수도 있습니다. 지역 엔드포인트는 AAD 인증을 지원하지 않습니다. 이러한 유형의 인증을 사용하기 위해 리소스에 대한 [사용자 지정 하위 도메인](#) 이름을 만듭니다.

아래에 표시된 [DefaultAzureCredential](#) 형식 또는 Azure SDK와 함께 제공되는 기타 자격 증명 형식을 사용하려면 패키지를 설치 `azure-identity` 하세요.

```
pip install azure-identity
```

또한 새 AAD 애플리케이션을 등록하고 서비스 주체에게 역할을 할당하여 "Cognitive Services User" 문서 인텔리전스에 대한 액세스 권한을 부여해야 합니다.

완료되면 AAD 애플리케이션의 클라이언트 ID, 테넌트 ID 및 클라이언트 암호 값을 환경 변수 AZURE_CLIENT_ID, AZURE_TENANT_ID, AZURE_CLIENT_SECRET 설정합니다.

Python

```
"""DefaultAzureCredential will use the values from these environment
variables: AZURE_CLIENT_ID, AZURE_TENANT_ID, AZURE_CLIENT_SECRET
"""

from azure.ai.formrecognizer import DocumentAnalysisClient
from azure.identity import DefaultAzureCredential

endpoint = os.environ["AZURE_FORM_RECOGNIZER_ENDPOINT"]
credential = DefaultAzureCredential()

document_analysis_client = DocumentAnalysisClient(endpoint, credential)
```

주요 개념

DocumentAnalysisClient

DocumentAnalysisClient 는 및 begin_analyze_document_from_url API를 통해 미리 빌드된 및 사용자 지정 모델을 사용하여 입력 문서를 분석하기 begin_analyze_document 위한 작업을 제공합니다. 매개 변수를 model_id 사용하여 분석할 모델 유형을 선택합니다. 지원되는 모델의 전체 목록은 여기를 참조 [하세요](#). 또한는 DocumentAnalysisClient 및 begin_classify_document_from_url API를 통해 begin_classify_document 문서를 분류하기 위한 작업을 제공합니다. 사용자 지정 분류 모델은 입력 파일의 각 페이지를 분류하여 문서를 식별할 수 있으며, 입력 파일 내에서 여러 문서 또는 단일 문서의 여러 인스턴스를 식별할 수도 있습니다.

여기에서 DocumentAnalysisClient 예제를 사용하는 방법을 설명하기 위해 샘플 코드 조각 제공됩니다. 지원되는 기능, 로캘 및 문서 형식을 포함한 문서 분석에 대한 자세한 내용은 [서비스 설명서](#)에서 확인할 수 있습니다.

DocumentModelAdministrationClient

DocumentModelAdministrationClient 는 다음에 대한 작업을 제공합니다.

- 사용자 지정 문서에 레이블을 지정하여 지정한 특정 필드를 분석하는 사용자 지정 모델을 빌드합니다. 모델이 분석할 수 있는 문서 유형과 각 필드에 대한 예상 신뢰도

를 나타내는가 `DocumentModelDetails` 반환됩니다. 자세한 설명은 [서비스 설명서](#)를 참조하세요.

- 기존 모델 컬렉션에서 구성된 모델을 만듭니다.
- 계정에서 생성된 모델을 관리합니다.
- 작업을 나열하거나 지난 24시간 이내에 만든 특정 모델 작업을 가져옵니다.
- 하나의 Form Recognizer 리소스에서 다른 리소스로 사용자 지정 모델을 복사합니다.
- 사용자 지정 분류 모델을 빌드하고 관리하여 애플리케이션 내에서 처리하는 문서를 분류합니다.

[Document Intelligence Studio](#) 와 같은 그래픽 사용자 인터페이스를 사용하여 모델을 빌드할 수도 있습니다.

샘플 코드 조각은 여기에서 `DocumentModelAdministrationClient` 사용하는 방법을 설명하기 위해 제공됩니다.

장기 실행 작업

장기 실행 작업은 작업을 시작하기 위해 서비스로 전송된 초기 요청으로 구성된 작업이며, 그 다음에는 간격으로 서비스를 폴링하여 작업이 완료되었는지 또는 실패했는지, 성공했는지 여부를 확인하여 결과를 가져옵니다.

문서를 분석하거나, 모델을 빌드하거나, 모델을 복사/작성하는 메서드는 장기 실행 작업으로 모델링됩니다. 클라이언트는 또는 `AsyncLROPoller`를 `begin_<method-name>` 반환하는 메서드를 `LROPoller` 노출합니다. 호출자는 메서드에서 `begin_<method-name>` 반환된 폴러 개체를 호출 `result()` 하여 작업이 완료되기를 기다려야 합니다. 샘플 코드 조각은 아래의 장기 실행 작업 를 사용하여 설명하기 위해 제공됩니다.

예제

다음 섹션에서는 다음을 포함하여 가장 일반적인 문서 인텔리전스 작업 중 일부를 다루는 몇 가지 코드 조각을 제공합니다.

- [레이아웃 추출](#)
- [일반 문서 모델 사용](#)
- [미리 빌드된 모델 사용](#)
- [사용자 지정 모델 빌드](#)
- [사용자 지정 모델을 사용하여 문서 분석](#)
- [모델 관리](#)
- [문서 분류](#)
- [추가 기능](#)

레이아웃 추출

문서에서 경계 영역 좌표와 함께 텍스트, 선택 표시, 텍스트 스타일 및 테이블 구조를 추출합니다.

Python

```
from azure.core.credentials import AzureKeyCredential
from azure.ai.formrecognizer import DocumentAnalysisClient

endpoint = os.environ["AZURE_FORM_RECOGNIZER_ENDPOINT"]
key = os.environ["AZURE_FORM_RECOGNIZER_KEY"]

document_analysis_client = DocumentAnalysisClient(
    endpoint=endpoint, credential=AzureKeyCredential(key)
)
with open(path_to_sample_documents, "rb") as f:
    poller = document_analysis_client.begin_analyze_document(
        "prebuilt-layout", document=f
    )
result = poller.result()

for idx, style in enumerate(result.styles):
    print(
        "Document contains {} content".format(
            "handwritten" if style.is_handwritten else "no handwritten"
        )
    )

for page in result.pages:
    print("----Analyzing layout from page #{}----".format(page.page_number))
    print(
        "Page has width: {} and height: {}, measured with unit: {}".format(
            page.width, page.height, page.unit
        )
    )

    for line_idx, line in enumerate(page.lines):
        words = line.get_words()
        print(
            "...Line # {} has word count {} and text '{}' within bounding
polygon '{}'".format(
                line_idx,
                len(words),
                line.content,
                line.polygon,
            )
        )

        for word in words:
            print(
                ".....Word '{}' has a confidence of {}".format(
                    word.content, word.confidence
                )
            )
```

```

        )
    )

    for selection_mark in page.selection_marks:
        print(
            "...Selection mark is '{}' within bounding polygon '{}' and has
a confidence of {}".format(
                selection_mark.state,
                selection_mark.polygon,
                selection_mark.confidence,
            )
        )
    )

for table_idx, table in enumerate(result.tables):
    print(
        "Table # {} has {} rows and {} columns".format(
            table_idx, table.row_count, table.column_count
        )
    )
    for region in table.bounding_regions:
        print(
            "Table # {} location on page: {} is {}".format(
                table_idx,
                region.page_number,
                region.polygon,
            )
        )
    for cell in table.cells:
        print(
            "...Cell[{}][{}] has content '{}'".format(
                cell.row_index,
                cell.column_index,
                cell.content,
            )
        )
        for region in cell.bounding_regions:
            print(
                "...content on page {} is within bounding polygon
'{}'.format(
                    region.page_number,
                    region.polygon,
                )
            )
    )

print("-----")

```

일반 문서 모델 사용

Document Intelligence 서비스에서 제공하는 일반 문서 모델을 사용하여 문서에서 키-값 쌍, 테이블, 스타일 및 선택 표시를 분석합니다. 메서드에 전달 `model_id="prebuilt-document"` `begin_analyze_document` 하여 일반 문서 모델을 선택합니다.

Python

```
from azure.core.credentials import AzureKeyCredential
from azure.ai.formrecognizer import DocumentAnalysisClient

endpoint = os.environ["AZURE_FORM_RECOGNIZER_ENDPOINT"]
key = os.environ["AZURE_FORM_RECOGNIZER_KEY"]

document_analysis_client = DocumentAnalysisClient(
    endpoint=endpoint, credential=AzureKeyCredential(key)
)
with open(path_to_sample_documents, "rb") as f:
    poller = document_analysis_client.begin_analyze_document(
        "prebuilt-document", document=f
    )
result = poller.result()

for style in result.styles:
    if style.is_handwritten:
        print("Document contains handwritten content: ")
        print(",".join([result.content[span.offset:span.offset + span.length] for span in style.spans]))

print("----Key-value pairs found in document----")
for kv_pair in result.key_value_pairs:
    if kv_pair.key:
        print(
            "Key '{}' found within '{}' bounding regions".format(
                kv_pair.key.content,
                kv_pair.key.bounding_regions,
            )
        )
    if kv_pair.value:
        print(
            "Value '{}' found within '{}' bounding regions\n".format(
                kv_pair.value.content,
                kv_pair.value.bounding_regions,
            )
        )

for page in result.pages:
    print("----Analyzing document from page #{}----"
        .format(page.page_number))
    print(
        "Page has width: {} and height: {}, measured with unit: {}".format(
            page.width, page.height, page.unit
        )
    )
    for line_idx, line in enumerate(page.lines):
        words = line.get_words()
        print(
            "...Line # {} has {} words and text '{}' within bounding polygon '{}'".
            format(
                line_idx, len(words), line.text, line.bounding_polygon
            )
        )
```

```

        line_idx,
        len(words),
        line.content,
        line.polygon,
    )
)

for word in words:
    print(
        ".....Word '{}' has a confidence of {}".format(
            word.content, word.confidence
        )
    )

for selection_mark in page.selection_marks:
    print(
        "...Selection mark is '{}' within bounding polygon '{}' and has
a confidence of {}".format(
            selection_mark.state,
            selection_mark.polygon,
            selection_mark.confidence,
        )
    )

for table_idx, table in enumerate(result.tables):
    print(
        "Table # {} has {} rows and {} columns".format(
            table_idx, table.row_count, table.column_count
        )
    )
    for region in table.bounding_regions:
        print(
            "Table # {} location on page: {} is {}".format(
                table_idx,
                region.page_number,
                region.polygon,
            )
        )
    for cell in table.cells:
        print(
            "...Cell[{}][{}] has content '{}'".format(
                cell.row_index,
                cell.column_index,
                cell.content,
            )
        )
        for region in cell.bounding_regions:
            print(
                "...content on page {} is within bounding polygon
'{}'.\n".format(
                    region.page_number,
                    region.polygon,
                )
            )
print("-----")

```

- 모델에서 제공하는 기능에 대한 자세한 내용은 여기를 [prebuilt-document 참조하세요](#).

미리 빌드된 모델 사용

문서 인텔리전스 서비스에서 제공하는 미리 빌드된 모델을 사용하여 영수증, 청구서, 명함, ID 문서 및 미국 W-2 세금 문서와 같은 일부 문서 유형에서 필드를 추출합니다.

예를 들어 판매 영수증에서 필드를 분석하려면 메서드에 전달 `model_id="prebuilt-receipt"` `begin_analyze_document` 하여 제공된 미리 빌드된 영수증 모델을 사용합니다.

Python

```
from azure.core.credentials import AzureKeyCredential
from azure.ai.formrecognizer import DocumentAnalysisClient

endpoint = os.environ["AZURE_FORM_RECOGNIZER_ENDPOINT"]
key = os.environ["AZURE_FORM_RECOGNIZER_KEY"]

document_analysis_client = DocumentAnalysisClient(
    endpoint=endpoint, credential=AzureKeyCredential(key)
)
with open(path_to_sample_documents, "rb") as f:
    poller = document_analysis_client.begin_analyze_document(
        "prebuilt-receipt", document=f, locale="en-US"
    )
receipts = poller.result()

for idx, receipt in enumerate(receipts.documents):
    print(f"-----Analysis of receipt #{idx + 1}-----")
    print(f"Receipt type: {receipt.doc_type if receipt.doc_type else 'N/A'}")
    merchant_name = receipt.fields.get("MerchantName")
    if merchant_name:
        print(
            f"Merchant Name: {merchant_name.value} has confidence: "
            f"{merchant_name.confidence}"
        )
    transaction_date = receipt.fields.get("TransactionDate")
    if transaction_date:
        print(
            f"Transaction Date: {transaction_date.value} has confidence: "
            f"{transaction_date.confidence}"
        )
    if receipt.fields.get("Items"):
        print("Receipt items:")
        for idx, item in enumerate(receipt.fields.get("Items").value):
            print(f"...Item #{idx + 1}")
            item_description = item.value.get("Description")
```

```

    if item_description:
        print(
            f".....Item Description: {item_description.value} has
confidence: "
            f"{item_description.confidence}"
        )
    item_quantity = item.value.get("Quantity")
    if item_quantity:
        print(
            f".....Item Quantity: {item_quantity.value} has
confidence: "
            f"{item_quantity.confidence}"
        )
    item_price = item.value.get("Price")
    if item_price:
        print(
            f".....Individual Item Price: {item_price.value} has
confidence: "
            f"{item_price.confidence}"
        )
    item_total_price = item.value.get("TotalPrice")
    if item_total_price:
        print(
            f".....Total Item Price: {item_total_price.value} has
confidence: "
            f"{item_total_price.confidence}"
        )
    subtotal = receipt.fields.get("Subtotal")
    if subtotal:
        print(f"Subtotal: {subtotal.value} has confidence:
{subtotal.confidence}")
    tax = receipt.fields.get("TotalTax")
    if tax:
        print(f"Total tax: {tax.value} has confidence: {tax.confidence}")
    tip = receipt.fields.get("Tip")
    if tip:
        print(f"Tip: {tip.value} has confidence: {tip.confidence}")
    total = receipt.fields.get("Total")
    if total:
        print(f"Total: {total.value} has confidence: {total.confidence}")
print("-----")

```

영수증으로 제한되지 않습니다! 미리 빌드된 몇 가지 모델 중에서 선택할 수 있으며, 각 모델에는 지원되는 필드 집합이 있습니다. 여기에서 지원되는 다른 미리 빌드된 모델을 참조 [하세요](#).

사용자 지정 모델 빌드

사용자 고유의 문서 형식으로 사용자 지정 모델을 빌드합니다. 결과 모델을 사용하여 학습된 문서 형식의 값을 분석할 수 있습니다. 학습 문서를 저장하는 Azure Storage Blob 컨테이너에 컨테이너 SAS URL을 제공합니다.

컨테이너 및 필수 파일 구조 설정에 대한 자세한 내용은 [서비스 설명서](#)에서 확인할 수 있습니다.

Python

```
from azure.ai.formrecognizer import (
    DocumentModelAdministrationClient,
    ModelBuildMode,
)
from azure.core.credentials import AzureKeyCredential

endpoint = os.environ["AZURE_FORM_RECOGNIZER_ENDPOINT"]
key = os.environ["AZURE_FORM_RECOGNIZER_KEY"]
container_sas_url = os.environ["CONTAINER_SAS_URL"]

document_model_admin_client = DocumentModelAdministrationClient(
    endpoint, AzureKeyCredential(key)
)
poller = document_model_admin_client.begin_build_document_model(
    ModelBuildMode.TEMPLATE,
    blob_container_url=container_sas_url,
    description="my model description",
)
model = poller.result()

print(f"Model ID: {model.model_id}")
print(f"Description: {model.description}")
print(f"Model created on: {model.created_on}")
print(f"Model expires on: {model.expires_on}")
print("Doc types the model can recognize:")
for name, doc_type in model.doc_types.items():
    print(
        f"Doc Type: '{name}' built with '{doc_type.build_mode}' mode which
has the following fields:"
    )
    for field_name, field in doc_type.field_schema.items():
        print(
            f"Field: '{field_name}' has type '{field['type']}' and
confidence score "
            f"{doc_type.field_confidence[field_name]}"
        )
    )
```

사용자 지정 모델을 사용하여 문서 분석

문서 필드, 테이블, 선택 표시 등을 분석합니다. 이러한 모델은 사용자 고유의 데이터로 학습되므로 문서에 맞게 조정됩니다. 최상의 결과를 위해 사용자 지정 모델이 빌드된 것과 동일한 문서 형식의 문서만 분석해야 합니다.

Python

```
from azure.core.credentials import AzureKeyCredential
from azure.ai.formrecognizer import DocumentAnalysisClient

endpoint = os.environ["AZURE_FORM_RECOGNIZER_ENDPOINT"]
key = os.environ["AZURE_FORM_RECOGNIZER_KEY"]
model_id = os.getenv("CUSTOM_BUILT_MODEL_ID", custom_model_id)

document_analysis_client = DocumentAnalysisClient(
    endpoint=endpoint, credential=AzureKeyCredential(key)
)

# Make sure your document's type is included in the list of document types
# the custom model can analyze
with open(path_to_sample_documents, "rb") as f:
    poller = document_analysis_client.begin_analyze_document(
        model_id=model_id, document=f
    )
result = poller.result()

for idx, document in enumerate(result.documents):
    print(f"-----Analyzing document #{idx + 1}-----")
    print(f"Document has type {document.doc_type}")
    print(f"Document has document type confidence {document.confidence}")
    print(f"Document was analyzed with model with ID {result.model_id}")
    for name, field in document.fields.items():
        field_value = field.value if field.value else field.content
        print(
            f".....found field of type '{field.value_type}' with value
'{field_value}' and with confidence {field.confidence}"
        )

# iterate over tables, lines, and selection marks on each page
for page in result.pages:
    print(f"\nLines found on page {page.page_number}")
    for line in page.lines:
        print(f"...Line '{line.content}'")
    for word in page.words:
        print(f"...Word '{word.content}' has a confidence of
{word.confidence}")
    if page.selection_marks:
        print(f"\nSelection marks found on page {page.page_number}")
        for selection_mark in page.selection_marks:
            print(
                f"...Selection mark is '{selection_mark.state}' and has a
confidence of {selection_mark.confidence}"
            )

for i, table in enumerate(result.tables):
    print(f"\nTable {i + 1} can be found on page:")
    for region in table.bounding_regions:
        print(f"....{region.page_number}")
    for cell in table.cells:
        print(
            f"....Cell[{cell.row_index}][{cell.column_index}] has text
{cell.content}"
```

```
'{cell.content}'  
    )  
print("-----")
```

또는 문서 URL을 사용하여 메서드를 사용하여 `begin_analyze_document_from_url` 문서를 분석할 수도 있습니다.

Python

```
document_url = "<url_of_the_document>"  
poller =  
document_analysis_client.begin_analyze_document_from_url(model_id=model_id,  
document_url=document_url)  
result = poller.result()
```

모델 관리

계정에 연결된 사용자 지정 모델을 관리합니다.

Python

```
from azure.ai.formrecognizer import DocumentModelAdministrationClient  
from azure.core.credentials import AzureKeyCredential  
from azure.core.exceptions import ResourceNotFoundError  
  
endpoint = "https://<my-custom-subdomain>.cognitiveservices.azure.com/"  
credential = AzureKeyCredential("<api_key>")  
  
document_model_admin_client = DocumentModelAdministrationClient(endpoint,  
credential)  
  
account_details = document_model_admin_client.get_resource_details()  
print("Our account has {} custom models, and we can have at most {} custom  
models".format(  
    account_details.custom_document_models.count,  
    account_details.custom_document_models.limit  
))  
  
# Here we get a paged list of all of our models  
models = document_model_admin_client.list_document_models()  
print("We have models with the following ids: {}".format(  
    ", ".join([m.model_id for m in models])  
))  
  
# Replace with the custom model ID from the "Build a model" sample  
model_id = "<model_id from the Build a Model sample>"  
  
custom_model =  
document_model_admin_client.get_document_model(model_id=model_id)  
print("Model ID: {}".format(custom_model.model_id))
```

```

print("Description: {}".format(custom_model.description))
print("Model created on: {} \n".format(custom_model.created_on))

# Finally, we will delete this model by ID
document_model_admin_client.delete_document_model(model_id=custom_model.mode
l_id)

try:

    document_model_admin_client.get_document_model(model_id=custom_model.model_i
d)
except ResourceNotFoundError:
    print("Successfully deleted model with id
{}".format(custom_model.model_id))

```

추가 기능

Document Intelligence는 보다 정교한 분석 기능을 지원합니다. 이러한 선택적 기능은 문서 추출 시나리오에 따라 사용하도록 설정하고 사용하지 않도록 설정할 수 있습니다.

다음 추가 기능 기능은 2023-07-31(GA) 이상 릴리스에 사용할 수 있습니다.

- barcode/QR 코드 ↗
- 수식 ↗
- 글꼴/스타일 ↗
- 고해상도 모드 ↗
- language ↗

일부 추가 기능에서는 추가 요금이 발생합니다. 가격 책정을

<https://azure.microsoft.com/pricing/details/ai-document-intelligence/> 참조하세요.

문제 해결

일반

Form Recognizer 클라이언트 라이브러리는 [Azure Core](#) ↗에 정의된 예외를 발생합니다. 문서 인텔리전스 서비스에서 발생한 오류 코드 및 메시지는 [서비스 설명서](#) ↗에서 찾을 수 있습니다.

로깅

이 라이브러리는 로깅에 표준 [로깅](#) ↗ 라이브러리를 사용합니다.

HTTP 세션(URL, 헤더 등)에 대한 기본 정보는 수준에서 기록됩니다 [INFO](#).

`DEBUG` 요청/응답 본문 및 수정되지 않은 헤더를 포함한 자세한 수준 로깅은 클라이언트에서 또는 키워드(keyword) 인수를 `logging_enable` 사용하여 작업별로 사용하도록 설정할 수 있습니다.

여기에 예제가 포함된 전체 SDK 로깅 설명서를 참조 [하세요](#).

선택적 구성

선택적 키워드(keyword) 인수는 클라이언트 및 작업별 수준에서 전달할 수 있습니다. [azure-core 참조 설명서](#) 에서는 재시도, 로깅, 전송 프로토콜 등에 사용할 수 있는 구성에 대해 설명합니다.

다음 단계

추가 샘플 코드

Form Recognizer Python API에서 사용되는 일반적인 패턴을 보여 주는 여러 코드 조각은 [샘플 추가 정보](#) 를 참조하세요.

추가 설명서

Azure AI 문서 인텔리전스에 대한 자세한 설명서는 [docs.microsoft.com 대한 문서 인텔리전스 설명서를 참조하세요](#).

참여

이 프로젝트에 대한 기여와 제안을 환영합니다. 대부분의 경우 기여하려면 권한을 부여하며 실제로 기여를 사용할 권한을 당사에 부여한다고 선언하는 CLA(기여자 라이선스 계약)에 동의해야 합니다. 자세한 내용은 [cla.microsoft.com](#).

끌어오기 요청을 제출하면 CLA-bot은 CLA를 제공하고 PR을 적절하게 데코레이팅해야 하는지 여부를 자동으로 결정합니다(예: 레이블, 설명). 봇에서 제공하는 지침을 따르기만 하면 됩니다. 이 작업은 CLA를 사용하여 모든 리포지토리에서 한 번만 수행하면 됩니다.

이 프로젝트에는 [Microsoft Open Source Code of Conduct](#) (Microsoft 오픈 소스 준수 사항)가 적용됩니다. 자세한 내용은 [Code of Conduct FAQ](#) (규정 FAQ)를 참조하세요. 또는 추가 질문이나 의견은 opencode@microsoft.com으로 문의하세요.

JavaScript용 Azure AI Document Intelligence 클라이언트 라이브러리 - 버전 5.0.0

아티클 • 2023. 08. 16.

Azure AI 문서 인텔리전스 는 기계 학습을 사용하여 문서에서 텍스트 및 구조화된 데이터를 분석하는 클라우드 서비스입니다. 여기에는 다음과 같은 기본 기능이 포함됩니다.

- 레이아웃 - 문서에서 경계 영역 좌표와 함께 텍스트, 테이블 구조 및 선택 표시를 추출합니다.
- 문서 - 미리 빌드된 일반 문서 모델을 사용하여 문서에서 엔터티, 키-값 쌍, 테이블 및 선택 표시를 분석합니다.
- 읽기 - 텍스트 언어 정보 외에도 페이지 단어 및 줄과 같은 텍스트 요소에 대한 정보를 읽습니다.
- 미리 빌드 - 미리 빌드된 모델을 사용하여 특정 유형의 일반 문서(예: 영수증, 청구서, 명함 또는 ID 문서)의 데이터를 분석합니다.
- 사용자 지정 - 문서에서 텍스트, 필드 값, 선택 표시 및 테이블 데이터를 추출하는 사용자 지정 모델을 빌드합니다. 사용자 지정 모델은 사용자 고유의 데이터로 빌드되므로 문서에 맞게 조정됩니다.
- 분류자 - 문서를 미리 정의된 클래스로 분류하는 사용자 지정 분류자를 빌드합니다.

[소스 코드](#) | [패키지\(NPM\)](#) | [API 참조 설명서](#) | [제품 설명서](#) | [샘플](#)

참고

문서 인텔리전스 서비스는 이전에 "Azure Form Recognizer"로 알려졌습니다. 이러한 서비스는 하나와 동일하며 `@azure/ai-form-recognizer` JavaScript 패키지는 Azure AI Document Intelligence 서비스에 대한 Azure SDK 패키지입니다. 작성 시 Azure FORM RECOGNIZER Azure AI Document Intelligence로의 이름 바꾸기가 진행 중이므로 "Form Recognizer" 및 "문서 인텔리전스"를 경우에 따라 서로 바꿔 사용할 수 있습니다.

`@azure/ai-form-recognizer` 패키지를 설치합니다.

를 사용하여 JavaScript용 Azure Document Intelligence 클라이언트 라이브러리를 npm 설치합니다.

Bash

```
npm install @azure/ai-form-recognizer
```

시작

JavaScript

```
const { DocumentAnalysisClient } = require("@azure/ai-form-recognizer");
const { DefaultAzureCredential } = require("@azure/identity");

const fs = require("fs");

const credential = new DefaultAzureCredential();
const client = new DocumentAnalysisClient(
    "https://<resource name>.cognitiveservices.azure.com",
    credential
);

// Document Intelligence supports many different types of files.
const file = fs.createReadStream("path/to/file.jpg");
const poller = await client.beginAnalyzeDocument("<model ID>", file);

const { pages, tables, styles, keyValuePairs, entities, documents } = await
poller.pollUntilDone();
```

현재 지원되는 환경

- Node.js의 LTS 버전 ↗
- 최신 버전의 Safari, Chrome, Edge, Firefox.

자세한 내용은 [지원 정책](#) ↗ 을 참조하세요.

필수 구성 요소

- Azure 구독 ↗
- Cognitive Services 또는 Form Recognizer 리소스. 리소스를 만들어야 하는 경우 Azure Portal ↗ 또는 Azure CLI를 사용할 수 있습니다.

Form Recognizer 리소스 만들기

참고: 작성 당시 Azure Portal 여전히 리소스를 "Form Recognizer" 리소스로 참조합니다. 나중에 "문서 인텔리전스" 리소스로 업데이트될 수 있습니다. 지금은 다음 설명서에서 "Form Recognizer" 이름을 사용합니다.

Document Intelligence는 [다중 서비스 및 단일 서비스 액세스](#)를 모두 지원합니다. 단일 앤드포인트/키에서 여러 Cognitive Services에 액세스하려는 경우 Cognitive Services 리소스를 만듭니다. Form Recognizer 리소스를 Form Recognizer 액세스 전용으로 만듭니다.

를 사용하여 리소스를 만들 수 있습니다.

옵션 1:[Azure Portal](#)

옵션 2:[Azure CLI](#).

다음은 CLI를 사용하여 Form Recognizer 리소스를 만드는 방법의 예입니다.

Bash

```
# Create a new resource group to hold the Form Recognizer resource -  
# if using an existing resource group, skip this step  
az group create --name my-resource-group --location westus2
```

Azure CLI를 사용하는 경우 및 <your-resource-name> 을 고유한 이름으로 바꿔 <your-resource-group-name> 니다.

Bash

```
az cognitiveservices account create --kind FormRecognizer --resource-group  
<your-resource-group-name> --name <your-resource-name> --sku <your-sku-name>  
--location <your-location>
```

클라이언트 만들기 및 인증

Document Intelligence 서비스와 상호 작용하려면 또는 을

`DocumentAnalysisClient` `DocumentModelAdministrationClient` 선택하고 이 유형의 instance 만들어야 합니다. 다음 예제에서는 를 사용합니다 `DocumentAnalysisClient`. Document Intelligence API에 액세스하기 위한 클라이언트 instance 만들려면 Form Recognizer 리소스 및 `credential`의 가 필요합니다 `endpoint`. 클라이언트는 리소스의 API 키와 함께 또는 Azure Active Directory RBAC를 `TokenCredential` 사용하여 클라이언트에 권한을 부여하는 를 사용할 `AzureKeyCredential` 수 있습니다.

Azure Portal에서 또는 아래 [Azure CLI](#) 코드 조각을 사용하여 Form Recognizer 리소스에 대한 엔드포인트를 찾을 수 있습니다.

Bash

```
az cognitiveservices account show --name <your-resource-name> --resource-group  
<your-resource-group-name> --query "properties.endpoint"
```

API 키 사용

Azure Portal [↗](#) 을 사용하여 Form Recognizer 리소스로 이동하고 API 키를 검색하거나 아래 Azure CLI 코드 조각을 사용합니다.

참고: API 키를 "구독 키" 또는 "구독 API 키"라고도 합니다.

Bash

```
az cognitiveservices account keys list --resource-group <your-resource-group-name> --name <your-resource-name>
```

API 키와 엔드포인트가 있으면 다음과 같이 사용할 수 있습니다.

JavaScript

```
const { DocumentAnalysisClient, AzureKeyCredential } = require("@azure/ai-form-recognizer");

const client = new DocumentAnalysisClient("<endpoint>", new AzureKeyCredential("<API key>"));
```

Azure Active Directory 사용

API 키 권한 부여는 대부분의 예제에서 사용되지만 Azure ID 라이브러리 [↗](#) 를 사용하여 Azure Active Directory로 클라이언트를 인증할 수도 있습니다. 아래에 표시된 DefaultAzureCredential [↗](#) 공급자 또는 Azure SDK와 함께 제공되는 기타 자격 증명 공급자를 사용하려면 패키지를 설치 `@azure/identity` 하세요.

Bash

```
npm install @azure/identity
```

서비스 주체를 사용하여 인증하려면 [AAD 애플리케이션을 등록](#) 하고 서비스 주체에 역할을 할당하여 `"Cognitive Services User"` 서비스에 대한 액세스 권한을 부여해야 합니다 (참고: 와 같은 `"Owner"` 다른 역할은 필요한 권한을 부여하지 않으며 예제 및 샘플 코드를 실행하는 데만 `"Cognitive Services User"` 충분함).

AAD 애플리케이션의 클라이언트 ID, 테넌트 ID 및 클라이언트 암호 값을 환경 변수로 설정합니다. `AZURE_CLIENT_ID`, `AZURE_TENANT_ID` `AZURE_CLIENT_SECRET`.

JavaScript

```
const { DocumentAnalysisClient } = require("@azure/ai-form-recognizer");
const { DefaultAzureCredential } = require("@azure/identity");
```

```
const client = new DocumentAnalysisClient("<endpoint>", new DefaultAzureCredential());
```

주요 개념

DocumentAnalysisClient

`DocumentAnalysisClient` 는 사용자 지정 및 미리 빌드된 모델을 사용하여 입력 문서를 분석하기 위한 작업을 제공합니다. 다음과 같은 세 가지 메서드가 있습니다.

- `beginAnalyzeDocument` - 모델 ID로 지정된 사용자 지정 또는 미리 빌드된 모델을 사용하여 입력 문서 파일 스트림에서 데이터를 추출합니다. 모든 리소스 및 해당 모델 ID/출력에서 지원되는 미리 빌드된 모델에 대한 자세한 내용은 [모델의 서비스 설명서를](#) 참조하세요.
- `beginAnalyzeDocumentFromUrl` 와 동일한 함수 `beginAnalyzeDocument` 를 수행하지만 파일 스트림 대신 공개적으로 액세스할 수 있는 파일 URL을 제출합니다.

DocumentModelAdministrationClient

`DocumentModelAdministrationClient` 는 리소스에서 모델을 관리(만들기, 읽기, 나열 및 삭제)하는 작업을 제공합니다.

- `beginBuildDocumentModel` 는 사용자 고유의 학습 데이터 집합에서 새 문서 모델을 만드는 작업을 시작합니다. 만든 모델은 사용자 지정 스키마에 따라 필드를 추출할 수 있습니다. 학습 데이터는 Azure Storage 컨테이너에 있고 특정 규칙에 따라 구성되어야 합니다. 학습 데이터 집합에 레이블을 적용하는 방법에 대한 자세한 설명은 [학습 데이터 집합 만들기에 대한 서비스의 설명서를](#) 참조하세요.
- `beginComposeDocumentModel` 는 여러 모델을 단일 모델로 구성하는 작업을 시작합니다. 사용자 지정 양식 인식에 사용되는 경우 새 구성 모델은 먼저 입력 문서의 분류를 수행하여 가장 적합한 하위 모델 중 어느 것을 결정합니다.
- `beginCopyModelTo` 는 사용자 지정 모델을 한 리소스에서 다른 리소스(또는 동일한 리소스로)로 복사하는 작업을 시작합니다. 메서드를 `copyAuthorization` 사용하여 생성할 수 있는 대상 리소스의 가 `getCopyAuthorization` 필요합니다.
- `getResourceDetails` 는 사용자 지정 모델 수 및 리소스에서 지원할 수 있는 최대 모델 수와 같은 리소스의 제한에 대한 정보를 검색합니다.
- `getDocumentModel`, `listDocumentModels` 및 `deleteDocumentModel` 리소스에서 모델 관리를 사용하도록 설정합니다.
- `getOperation` 진행 `listOperations` 중이거나 실패한 작업에서도 모델 만들기 작업의 상태 볼 수 있습니다. 작업은 24시간 동안 유지됩니다.

Document Intelligence 서비스의 그래픽 사용자 인터페이스인 Document Intelligence Studio[↗]를 사용하여 모델을 만들 수도 있습니다.

모델을 빌드하는 데 사용하는 `DocumentModelAdministrationClient` 방법을 보여 주는 샘플 코드 조각은 "모델 빌드" 예제 섹션에서 아래에 나와 있습니다.

장기 실행 작업

LLO(장기 실행 작업)는 작업을 시작하기 위해 서비스로 전송된 초기 요청으로 구성된 작업이며, 작업이 완료되었는지 여부와 실패 또는 성공 여부를 결정하기 위해 특정 간격으로 결과를 폴링합니다. 궁극적으로 LRO는 오류와 함께 실패하거나 결과를 생성합니다.

Azure AI 문서 인텔리전스에서 모델 복사 및 작성과 함께 모델을 만드는 작업과 분석/데이터 추출 작업은 LLO입니다. SDK 클라이언트는 개체를 반환 `Promise<PollerLike>` 하는 비동 `begin<operation-name>` 기 메서드를 제공합니다. 개체는 `PollerLike` 서비스의 인프라에서 비동기적으로 실행되는 작업을 나타내며 프로그램은 메서드에서 반환된 폴러에서 `begin<operation-name>` 메서드를 호출하고 대기하여 `pollUntilDone` 작업이 완료 되기를 기다릴 수 있습니다. 다음 섹션에서 장기 실행 작업을 사용하는 방법을 설명하기 위해 샘플 코드 조각이 제공됩니다.

예제

다음 섹션에서는 Document Intelligence 클라이언트 라이브러리에 사용되는 일반적인 패턴을 보여 주는 여러 JavaScript 코드 조각을 제공합니다.

- 모델 ID를 사용하여 문서 분석
- 미리 빌드된 문서 모델 사용
- 미리 빌드된 "레이아웃" 사용
- 미리 빌드된 "문서" 사용
- 미리 빌드된 "읽기" 사용
- 모델 작성
- 모델 관리

모델 ID를 사용하여 문서 분석

메서드는 `beginAnalyzeDocument` 문서에서 필드 및 테이블 데이터를 추출할 수 있습니다. 분석은 사용자 고유의 데이터로 학습된 사용자 지정 모델 또는 서비스에서 제공하는 미리 빌드된 모델을 사용할 수 있습니다(아래 [미리 빌드된 모델 사용](#) 참조). 사용자 지정 모델은 사용자 고유의 문서에 맞게 조정되므로 모델의 문서 형식 중 하나와 동일한 구조의 문서와만 사용해야 합니다(구성된 모델과 같이 여러 개 있을 수 있음).

JavaScript

```
const { DocumentAnalysisClient, AzureKeyCredential } = require("@azure/ai-form-recognizer");

const fs = require("fs");

async function main() {
    const endpoint = "<cognitive services endpoint>";
    const apiKey = "<api key>";
    const modelId = "<model id>";
    const path = "<path to a document>";

    const readStream = fs.createReadStream(path);

    const client = new DocumentAnalysisClient(endpoint, new
AzureKeyCredential(apiKey));
    const poller = await client.beginAnalyzeDocument(modelId, readStream, {
        onProgress: ({ status }) => {
            console.log(`status: ${status}`);
        },
    });

    // There are more fields than just these three
    const { documents, pages, tables } = await poller.pollUntilDone();

    console.log("Documents:");
    for (const document of documents || []) {
        console.log(`Type: ${document.docType}`);
        console.log("Fields:");
        for (const [name, field] of Object.entries(document.fields)) {
            console.log(
                `Field ${name} has value '${field.value}' with a confidence score of
${field.confidence}`
            );
        }
    }
    console.log("Pages:");
    for (const page of pages || []) {
        console.log(`Page number: ${page.pageNumber}
(${page.width}x${page.height} ${page.unit})`);
    }

    console.log("Tables:");
    for (const table of tables || []) {
        console.log(`- Table (${table.columnCount}x${table.rowCount})`);
        for (const cell of table.cells) {
            console.log(`  - cell (${cell.RowIndex},${cell.ColumnIndex})
"${cell.content}"`);
        }
    }
}

main().catch((err) => {
```

```
    console.error("The sample encountered an error:", err);
});
```

URL에서 문서 분석

읽기 가능한 스트림을 제공하는 대신 메서드를 사용하여 공개적으로 액세스할 수 있는 URL을 `beginAnalyzeDocumentFromUrl` 제공할 수 있습니다. "공개적으로 액세스할 수 있음"은 URL 원본이 서비스의 인프라에서 액세스할 수 있어야 했음을 의미합니다(즉, 헤더 또는 인증서 기반 비밀을 사용하는 개인 인트라넷 URL 또는 URL은 Document Intelligence 서비스에서 URL에 액세스할 수 있어야 하므로 작동하지 않음). 그러나 URL 자체는 쿼리 매개 변수에 SAS 토큰을 포함하는 Azure Storage Blob URL과 같은 비밀을 인코딩할 수 있습니다.

미리 빌드된 문서 모델 사용

또한 이 메서드는 `beginAnalyzeDocument` 문서 인텔리전스 서비스에서 제공하는 미리 빌드된 모델을 사용하여 영수증, 청구서, 명함, ID 문서 등과 같은 특정 유형의 일반 문서에서 필드를 추출할 수 있습니다. 미리 빌드된 모델은 모델 ID 문자열(사용자 지정 문서 모델과 동일하며 아래의 [다른 미리 빌드된 모델](#) 섹션 참조)으로 제공되거나 개체를 `DocumentModel` 사용할 수 있습니다. 를 `DocumentModel` 사용하는 경우 JavaScript용 Document Intelligence SDK는 모델의 스키마를 기반으로 추출된 문서에 훨씬 더 강력한 TypeScript 형식을 제공하며 JavaScript 명명 규칙을 사용하도록 변환됩니다.

현재 서비스 API 버전(2022-08-31)에 대한 예제 `DocumentModel` 개체는 [샘플 디렉터리](#)에서 [prebuilt](#) 찾을 수 있습니다. 다음 예제에서는 해당 디렉터리의 [`prebuilt-receipt.ts`] 파일에서 를 사용합니다 `PrebuiltReceiptModel`.

기반 분석의 `DocumentModel` 기본 이점은 TypeScript 형식 제약 조건이 더 강력하기 때문에 다음 샘플은 ECMAScript 모듈 구문을 사용하여 TypeScript로 작성됩니다.

TypeScript

```
import { DocumentAnalysisClient, AzureKeyCredential } from "@azure/ai-form-recognizer";

// Copy the file from the above-linked sample directory so that it can be imported in this module
import { PrebuiltReceiptModel } from "./prebuilt/prebuilt-receipt";

import fs from "fs";

async function main() {
    const endpoint = "<cognitive services endpoint>";
    const apiKey = "<api key>";
```

```

const path = "<path to your receipt document>"; // pdf/jpeg/png/tiff
formats

const readStream = fs.createReadStream(path);

const client = new DocumentAnalysisClient(endpoint, new
AzureKeyCredential(apiKey));

// The PrebuiltReceiptModel `DocumentModel` instance encodes both the
model ID and a stronger return type for the operation
const poller = await client.beginAnalyzeDocument(PrebuiltReceiptModel,
readStream, {
    onProgress: ({ status }) => {
        console.log(`status: ${status}`);
    },
});

const {
    documents: [receiptDocument],
} = await poller.pollUntilDone();

// The fields of the document constitute the extracted receipt data.
const receipt = receiptDocument.fields;

if (receipt === undefined) {
    throw new Error("Expected at least one receipt in analysis result.");
}

console.log(`Receipt data (${receiptDocument.docType})`);
console.log(" Merchant Name:", receipt.merchantName?.value);

// The items of the receipt are an example of a `DocumentArrayValue`
if (receipt.items !== undefined) {
    console.log("Items:");
    for (const { properties: item } of receipt.items.values) {
        console.log("- Description:", item.description?.value);
        console.log(" Total Price:", item.totalPrice?.value);
    }
}

console.log(" Total:", receipt.total?.value);
}

main().catch((err) => {
    console.error("The sample encountered an error:", err);
});

```

또는 위에서 설명한 것처럼 더 강력한 반환 형식을 생성하는 를 사용하는

`PrebuiltReceiptModel` 대신 미리 빌드된 영수증의 모델 ID("미리 빌드된 영수증")를 사용 할 수 있지만 문서 필드는 TypeScript에 강하게 입력되지 않으며 필드 이름은 일반적으로 "camelCase" 대신 "PascalCase"에 있습니다.

기타 미리 빌드된 모델

당신은 영수증에 국한되지 않습니다! 몇 가지 미리 빌드된 모델 중에서 선택할 수 있으며 더 많은 모델을 선택할 수 있습니다. 미리 빌드된 각 모델에는 지원되는 고유한 필드 집합이 있습니다.

- 영수증(위와 같이) 또는 미리 빌드된 영수증 모델 ID "prebuilt-receipt"를 사용합니다 [PrebuiltReceiptModel](#).
- 또는 모델 ID "prebuilt-businessCard"를 사용하는 [PrebuiltBusinessCardModel](#) 명함.
- 송장, 사용 [PrebuiltInvoiceModel](#) 또는 해당 모델 ID "prebuilt-invoice".
- 또는 모델 ID "prebuilt-idDocument"를 사용하는 [PrebuiltIdDocumentModel](#) ID 문서(예: 운전 면허증 및 여권)입니다.
- W2 Tax Forms(미국) 또는 모델 ID "prebuilt-tax.us.w2"를 사용합니다 [PrebuiltTaxUsW2Model](#).
- [[PrebuiltHealthInsuranceCardUsModel](#)][samples-prebuilt-healthinsurancecard.us] 또는 모델 ID "prebuilt-healthInsuranceCard.us"를 사용하여 건강 보험 카드(미국)

위의 미리 빌드된 각 모델은 (모델 필드 스키마의 추출된 인스턴스)를 생성 [documents](#) 합니다. 필드 스키마가 없으므로 를 생성 [documents](#)하지 않는 세 가지 미리 빌드된 모델도 있습니다. 아래에 이 계정과 키의 예제가 나와 있습니다.

- 페이지 및 테이블과 같은 OCR(기본 레이아웃) 요소에 대한 정보를 추출하는 미리 빌드된 레이아웃 모델(아래 [의 "레이아웃" 사용](#) 참조)입니다.
- 미리 빌드된 일반 문서 모델(아래 [의 "문서" 미리 빌드](#) 사용 참조)은 레이아웃 모델에서 생성된 정보에 키-값 쌍(레이블이 지정된 요소와 같은 페이지 요소 간의 연결 지시)을 추가합니다.
- 미리 빌드된 읽기 모델(아래 [의 "읽기" 미리 빌드](#) 사용 참조)은 문서 언어에 대한 정보와 함께 페이지 단어 및 줄과 같은 텍스트 요소만 추출합니다.

이러한 모든 모델의 필드에 대한 자세한 내용은 [미리 빌드된 사용 가능한 모델의 서비스 설명서를 참조하세요](#).

미리 빌드된 모든 모델의 필드는 의 메서드(모델 ID [DocumentModelAdministrationClient](#))를 사용하여 [getDocumentModel](#) 프로그래밍 방식으로 액세스하고 결과에서 필드를 검사할 [docTypes](#) 수도 있습니다.

미리 빌드된 "레이아웃" 사용

모델은 "prebuilt-layout" 입력 문서의 텍스트 콘텐츠 내에서 경계 영역 및 범위와 함께 페이지(텍스트 단어/선 및 선택 표시로 구성됨), 테이블 및 시각적 텍스트 스타일과 같은 문서의 기본 요소만 추출합니다. 이 모델을 호출하는 강력한 [PrebuiltLayoutModel](#) 형식

`DocumentModel`의 instance 제공하거나 항상 모델 ID "prebuilt-layout" 를 직접 사용할 수 있습니다.

기반 분석의 `DocumentModel` 기본 이점은 TypeScript 형식 제약 조건이 더 강력하기 때문에 다음 샘플은 ECMAScript 모듈 구문을 사용하여 TypeScript로 작성됩니다.

TypeScript

```
import { DocumentAnalysisClient, AzureKeyCredential } from "@azure/ai-form-recognizer";

// Copy the above-linked `DocumentModel` file so that it may be imported in this module.
import { PrebuiltLayoutModel } from "./prebuilt/prebuilt-layout";

import fs from "fs";

async function main() {
    const endpoint = "<cognitive services endpoint>";
    const apiKey = "<api key>";
    const path = "<path to a document>"; // pdf/jpeg/png/tiff formats

    const readStream = fs.createReadStream(path);

    const client = new DocumentAnalysisClient(endpoint, new AzureKeyCredential(apiKey));
    const poller = await client.beginAnalyzeDocument(PrebuiltLayoutModel, readStream);
    const { pages, tables } = await poller.pollUntilDone();

    for (const page of pages || []) {
        console.log(`- Page ${page.pageNumber}: (${page.width}x${page.height}) ${page.unit}`);
    }

    for (const table of tables || []) {
        console.log(`- Table (${table.columnCount}x${table.rowCount})`);
        for (const cell of table.cells) {
            console.log(`  cell [${cell.RowIndex},${cell.ColumnIndex}] "${cell.content}"`);
        }
    }
}

main().catch((err) => {
    console.error("The sample encountered an error:", err);
});
```

미리 빌드된 "문서" 사용

모델은 "prebuilt-document" 레이아웃 추출 메서드에서 생성된 속성 외에도 키-값 쌍(레이블이 지정된 필드와 같은 페이지 요소 간 연결)에 대한 정보를 추출합니다. 이 미리 빌드된(일반) 문서 모델은 문서 인텔리전스 서비스의 이전 반복에서 레이블 정보 없이 학습된 사용자 지정 모델과 유사한 기능을 제공하지만 이제는 다양한 문서에서 작동하는 미리 빌드된 모델로 제공됩니다. 이 모델을 호출하는 강력한 [PrebuiltDocumentModel](#) 형식 `DocumentModel` 의 instance 제공하거나 항상 모델 ID "prebuilt-document" 를 직접 사용할 수 있습니다.

기반 분석의 `DocumentModel` 기본 이점은 TypeScript 형식 제약 조건이 더 강력하기 때문에 다음 샘플은 ECMAScript 모듈 구문을 사용하여 TypeScript로 작성됩니다.

TypeScript

```
import { DocumentAnalysisClient, AzureKeyCredential } from "@azure/ai-form-recognizer";

// Copy the above-linked `DocumentModel` file so that it may be imported in this module.
import { PrebuiltDocumentModel } from "./prebuilt/prebuilt-document";

import fs from "fs";

async function main() {
    const endpoint = "<cognitive services endpoint>";
    const apiKey = "<api key>";
    const path = "<path to a document>"; // pdf/jpeg/png/tiff formats

    const readStream = fs.createReadStream(path);

    const client = new DocumentAnalysisClient(endpoint, new AzureKeyCredential(apiKey));
    const poller = await client.beginAnalyzeDocument(PrebuiltDocumentModel, readStream);

    // `pages`, `tables` and `styles` are also available as in the "layout" example above, but for the sake of this
    // example we won't show them here.
    const { keyValuePairs } = await poller.pollUntilDone();

    if (!keyValuePairs || keyValuePairs.length <= 0) {
        console.log("No key-value pairs were extracted from the document.");
    } else {
        console.log("Key-Value Pairs:");
        for (const { key, value, confidence } of keyValuePairs) {
            console.log(`- Key : ${key.content}`);
            console.log(` Value: ${value?.content ?? '<undefined>'}`);
        }
    }
}
```

```
main().catch((err) => {
    console.error("The sample encountered an error:", err);
});
```

미리 빌드된 "읽기" 사용

이 `"prebuilt-read"` 모델은 단어 및 단락과 같은 문서에서 텍스트 정보를 추출하고 해당 텍스트의 언어 및 쓰기 스타일(예: 필기 및 형식 집합)을 분석합니다. 이 모델을 호출하는 강력한 [PrebuiltReadModel](#) 형식 `DocumentModel`의 instance 제공하거나 항상 모델 ID `"prebuilt-read"`를 직접 사용할 수 있습니다.

기반 분석의 `DocumentModel` 기본 이점은 TypeScript 형식 제약 조건이 더 강력하기 때문에 다음 샘플은 ECMAScript 모듈 구문을 사용하여 TypeScript로 작성됩니다.

TypeScript

```
import { DocumentAnalysisClient, AzureKeyCredential } from "@azure/ai-form-
recognizer";

// Copy the above-linked `DocumentModel` file so that it may be imported in
this module.
import { PrebuiltReadModel } from "./prebuilt/prebuilt-read";

// See the samples directory for a definition of this helper function.
import { getTextOfSpans } from "./utils";

import fs from "fs";

async function main() {
    const endpoint = "<cognitive services endpoint>";
    const apiKey = "<api key>";
    const path = "<path to a document>"; // pdf/jpeg/png/tiff formats

    const readStream = fs.createReadStream(path);

    const client = new DocumentAnalysisClient(endpoint, new
AzureKeyCredential(apiKey));
    const poller = await client.beginAnalyzeDocument(PrebuiltReadModel,
readStream);

    // The "prebuilt-read" model (`beginReadDocument` method) only extracts
information about the textual content of the
    // document, such as page text elements, text styles, and information
about the language of the text.
    const { content, pages, languages } = await poller.pollUntilDone();

    if (!pages || pages.length <= 0) {
        console.log("No pages were extracted from the document.");
    } else {
        console.log("Pages:");
    }
}
```

```

for (const page of pages) {
    console.log("- Page", page.pageNumber, `(unit: ${page.unit})`);
    console.log(` ${page.width}x${page.height}, angle: ${page.angle}`);
    console.log(
        ` ${page.lines && page.lines.length} lines, ${page.words &&
page.words.length} words`;
    );

    if (page.lines && page.lines.length > 0) {
        console.log(" Lines:");
        for (const line of page.lines) {
            console.log(` - ${line.content}`);
        }
    }
}

if (!languages || languages.length <= 0) {
    console.log("No language spans were extracted from the document.");
} else {
    console.log("Languages:");
    for (const languageEntry of languages) {
        console.log(
            ` - Found language: ${languageEntry.locale} (confidence:
${languageEntry.confidence})`
        );
        for (const text of getTextOfSpans(content, languageEntry.spans)) {
            const escapedText = text.replace(/\r?\n/g, "\\n").replace(/"/g,
'\\\"');
            console.log(` - ${escapedText}`);
        }
    }
}

main().catch((error) => {
    console.error("An error occurred:", error);
    process.exit(1);
});

```

문서 분류

Document Intelligence 서비스는 학습 데이터 집합을 기반으로 문서를 미리 정의된 범주 집합으로 분류할 수 있는 사용자 지정 문서 분류자를 지원합니다. 문서는 의 `DocumentAnalysisClient` 메서드를 사용하여 사용자 지정 분류자를 사용하여 `beginClassifyDocument` 분류할 수 있습니다. 위와 마찬가지로 `beginAnalyzeDocument` 이 메서드는 `beginClassifyDocumentFromUrl` 분류할 문서가 포함된 파일 또는 스트림을 허용하며, 문서에 공개적으로 액세스할 수 있는 URL을 대신 허용하는 해당 파일이 있습니다.

다음 샘플에서는 사용자 지정 분류자를 사용하여 문서를 분류하는 방법을 보여줍니다.

JavaScript

```
const { AzureKeyCredential, DocumentAnalysisClient } = require("@azure/ai-form-recognizer");

async function main() {
    const endpoint = "<endpoint>";
    const credential = new AzureKeyCredential("<api key>");

    const documentUrl =
        "https://raw.githubusercontent.com/Azure/azure-sdk-for-js/main/sdk/formrecognizer/ai-form-recognizer/assets/invoice/Invoice_1.pdf";

    const client = new DocumentAnalysisClient(endpoint, credential);

    const poller = await client.beginClassifyDocumentFromUrl("<classifier id>", documentUrl);

    const result = await poller.pollUntilDone();

    if (result.documents === undefined || result.documents.length === 0) {
        throw new Error("Failed to extract any documents.");
    }

    for (const document of result.documents) {
        console.log(`Extracted a document with type '${document.docType}' on page ${document.boundingRegions?.[0].pageNumber} (confidence: ${document.confidence})`);
    }
}

main().catch((error) => {
    console.error("An error occurred:", error);
    process.exit(1);
});
```

사용자 지정 분류자를 학습하는 방법에 대한 자세한 내용은 다음 섹션의 끝에 있는 [분류자 학습 섹션](#)을 참조하세요.

모델 빌드

또한 SDK는 클래스를 사용하여 모델 만들기를 `DocumentModelAdministrationClient` 지원합니다. 레이블이 지정된 학습 데이터에서 모델을 빌드하면 사용자 고유의 문서에 대해 학습된 새 모델이 만들어지고, 결과 모델은 해당 문서의 구조에서 값을 인식할 수 있습니다. 모델 빌드 작업은 학습 문서를 보유하는 Azure Storage Blob 컨테이너에 SAS로 인코

ding된 URL을 허용합니다. Document Intelligence 서비스의 인프라는 컨테이너의 파일을 읽고 해당 내용을 기반으로 모델을 만듭니다. 학습 데이터 컨테이너를 만들고 구성하는 방법에 대한 자세한 내용은 [모델 빌드를 위한 Document Intelligence 서비스의 설명서](#)를 참조하세요.

프로그래밍 방식 모델 만들기를 위해 이러한 방법을 제공하지만 Document Intelligence 서비스 팀은 웹에서 모델을 만들고 관리할 수 있는 대화형 웹 애플리케이션 [인 Document Intelligence Studio](#)를 만들었습니다.

예를 들어 다음 프로그램은 SAS로 인코딩된 URL을 사용하여 기존 Azure Storage 컨테이너에 사용자 지정 문서 모델을 빌드합니다.

JavaScript

```
const {
    DocumentModelAdministrationClient,
    AzureKeyCredential,
} = require("@azure/ai-form-recognizer");

async function main() {
    const endpoint = "<cognitive services endpoint>";
    const apiKey = "<api key>";
    const containerSasUrl = "<SAS url to the blob container storing training documents>";

    const client = new DocumentModelAdministrationClient(endpoint, new
AzureKeyCredential(apiKey));

    // You must provide the model ID. It can be any text that does not start
    // with "prebuilt-".
    // For example, you could provide a randomly generated GUID using the
    // "uuid" package.
    // The second parameter is the SAS-encoded URL to an Azure Storage
    // container with the training documents.
    // The third parameter is the build mode: one of "template" (the only mode
    // prior to 4.0.0-beta.3) or "neural".
    // See https://aka.ms/azsdk/formrecognizer/buildmode for more information
    // about build modes.
    const poller = await client.beginBuildDocumentModel("<model ID>",
    containerSasUrl, "template", {
        // The model description is optional and can be any text.
        description: "This is my new model!",
        onProgress: ({ status }) => {
            console.log(`operation status: ${status}`);
        },
    });
    const model = await poller.pollUntilDone();

    console.log("Model ID:", model.modelId);
    console.log("Description:", model.description);
    console.log("Created:", model.createdOn);
```

```

// A model may contain several document types, which describe the possible
object structures of fields extracted using
// this model

console.log("Document Types:");
for (const [docType, { description, fieldSchema: schema }] of
Object.entries(
model.docTypes ?? {}
)) {
  console.log(` - Name: "${docType}"`);
  console.log(`   Description: "${description}"`);

  // For simplicity, this example will only show top-level field names
  console.log("   Fields:");

  for (const [fieldName, fieldSchema] of Object.entries(schema)) {
    console.log(`     - "${fieldName}" (${fieldSchema.type})`);
    console.log(`       ${fieldSchema.description ?? "<no description>"}`);
  }
}
}

main().catch((err) => {
  console.error("The sample encountered an error:", err);
});

```

사용자 지정 분류자는 가 아닌 `beginBuildDocumentModel` 메서드를 사용하여

`beginBuildDocumentClassifier` 비슷한 방식으로 빌드됩니다. 입력 학습 데이터가 약간 다른 형식으로 제공되기 때문에 사용자 지정 분류자를 빌드하는 방법에 대한 자세한 내용은 [빌드 분류자 샘플을](#) 참조하세요. 사용자 지정 분류자를 위한 학습 데이터 집합을 빌드하는 방법에 대한 자세한 내용은 [문서 인텔리전스 서비스 설명서를 참조하세요](#).

모델 관리

`DocumentModelAdministrationClient` 모델 액세스 및 나열을 위한 여러 방법도 제공합니다. 다음 예제에서는 리소스의 모델을 반복하는 방법을 보여 줍니다(리소스의 사용자 지정 모델과 모든 리소스에 공통되는 미리 빌드된 모델 모두 포함), ID로 모델을 가져오고 모델을 삭제합니다.

JavaScript

```

const {
  DocumentModelAdministrationClient,
  AzureKeyCredential,
} = require("@azure/ai-form-recognizer");

async function main() {
  const endpoint = "<cognitive services endpoint>";
  const apiKey = "<api key>";

```

```

const client = new DocumentModelAdministrationClient(endpoint, new
AzureKeyCredential(apiKey));

    // Produces an async iterable that supports paging
(`PagedAsyncIterableIterator`). The `listDocumentModels` method will only
    // iterate over model summaries, which do not include detailed schema
information. Schema information is only returned
    // from `getDocumentModel` as part of the full model information.
const models = client.listDocumentModels();
let i = 1;
for await (const summary of models) {
    console.log(`Model ${i++}:`, summary);
}

    // The iterable is paged, and the application can control the flow of
paging if needed
i = 1;
for await (const page of client.listDocumentModels().byPage()) {
    for (const summary of page) {
        console.log(`Model ${i++}`, summary);
    }
}

    // We can also get a full ModelInfo by ID. Here we only show the basic
information. See the documentation and the
    // `getDocumentModel` sample program for information about the `docTypes`
field, which contains the model's document type
    // schemas.
const model = await client.getDocumentModel("<model ID>");
console.log("ID", model.modelId);
console.log("Created:", model.createdOn);
console.log("Description: ", model.description ?? "<none>");

    // A model can also be deleted by its model ID. Once it is deleted, it
CANNOT be recovered.
const modelIdToDelete = "<model ID that should be deleted forever>";
await client.deleteDocumentModel(modelIdToDelete);
}

main().catch((err) => {
    console.error("The sample encountered an error:", err);
});

```

유사한 메서드 `listDocumentClassifiers` 및 `getDocumentClassifier` 는 사용자 지정 분류자를 삭제하는 것 외에도 `deleteDocumentClassifier` 사용자 지정 분류자에 대한 정보를 나열하고 가져오는 데 사용할 수 있습니다.

문제 해결

문제 해결에 대한 지원은 [문제 해결 가이드를 참조하세요 ↗](#).

로깅

로깅을 사용하도록 설정하면 실패에 대한 유용한 정보를 파악하는데 도움이 될 수 있습니다. HTTP 요청 및 응답 로그를 보려면 `AZURE_LOG_LEVEL` 환경 변수를 `info`로 설정합니다. 또는 `@azure/logger`에서 `setLogLevel`을 호출하여 런타임에 로깅을 사용하도록 설정할 수 있습니다.

JavaScript

```
const { setLogLevel } = require("@azure/logger");

setLogLevel("info");
```

로그를 사용하는 방법에 대한 자세한 내용은 [@azure/logger package docs](#)를 참조하세요.

다음 단계

위의 "예제" 섹션에 표시되지 않는 몇 가지 기능 및 메서드(예: 모델 복사 및 작성, 모델 관리 작업 나열 및 모델 삭제)를 포함하여 이 라이브러리를 사용하는 방법을 보여 주는 자세한 코드 샘플은 [샘플](#) 디렉터리를 살펴보세요.

참여

이 라이브러리에 기여하려면 [기여 가이드](#)를 참조하여 코드를 빌드하고 테스트하는 방법에 대해 자세히 알아보세요.

GitHub에서 Microsoft와 공동 작업

이 콘텐츠의 원본은 GitHub에서 찾을 수 있으며, 여기서 문제와 끌어오기 요청을 만들고 검토할 수도 있습니다. 자세한 내용은 [참여자 가이드](#)를 참조하세요.



Azure SDK for JavaScript 피드백

Azure SDK for JavaScript은(는) 오픈 소스 프로젝트입니다. 다음 링크를 선택하여 피드백을 제공해 주세요.

 설명서 문제 열기

 제품 사용자 의견 제공

Transparency note and use cases for Document Intelligence

Article • 03/12/2024

What is a transparency note?

An AI system includes not only the technology, but also the people who will use it, the people who will be affected by it, and the environment in which it is deployed. Creating a system that is fit for its intended purpose requires an understanding of how the technology works, its capabilities and limitations, and how to achieve the best performance.

Microsoft provides *transparency notes* to help you understand how our AI technology works. This includes the choices system owners can make that influence system performance and behavior, and the importance of thinking about the whole system, including the technology, the people, and the environment. You can use transparency notes when developing or deploying your own system, or share them with the people who will use or be affected by your system.

Transparency notes are part of a broader effort at Microsoft to put our AI principles into practice. To find out more, see [Microsoft's AI principles](#).

The basics of Document Intelligence

Introduction

[Document Intelligence](#) is accessed via a set of APIs and allows developers to easily extract text, structure, and fields from their documents. It is composed of features like:

- Read for text extraction.
- Layout and General Documents for structural insights and general key-values and entities such as names, places, and things.
- Prebuilt models for specific document types like invoices, receipts, business cards, W2s, and IDs.
- Custom models for building models specific to your document types.

Document Intelligence supports one or more languages and locales for each of the features, as listed in the [Supported languages](#) article.

Key terms

[+] [Expand table](#)

Term	Definition
Read	This feature extracts text lines, words, and their locations from images and documents, along with other information such as detected languages.
Layout	This feature extracts text, selection marks, and table structure (the row and column numbers associated with the text). See Document Intelligence Layout .
General Documents	Analyze documents and associate values to keys and entries to tables that it discovers. For more information, see Document Intelligence General Documents .
Prebuilt models	Prebuilt models are document-specific models for unique form types. These models don't require custom training before use. For example, the prebuilt invoice model extracts key fields from invoices. For more information, see Document Intelligence prebuilt invoice model .
Custom models	Document Intelligence allows you to train a custom model that's tailored to your forms and documents. This model extracts text, key-value pairs, selection marks, and table data. Custom models can be improved with human feedback by applying human review, updating the labels, and retraining the model by using the API.
Confidence value	All Get Analysis Results operations return confidence values in the range between 0 and 1 for all extracted words and key-value mappings. This value represents the service's estimate of how many times it correctly extracts the word out of 100 or correctly maps the key-value pairs. For example, a word that's estimated to be extracted correctly 82% of the time results in a confidence value of 0.82.
Add-on features	Document Intelligence offers a set of add-on features to extend the results to include more elements from your documents. Some add-on features incur an extra cost and can be enabled and disabled depending on the scenario of the document extraction. We currently offer high resolution, formula, styleFont, barcodes, languages, keyValuePairs, and queryFields extraction capabilities. For more information, see Document Intelligence Add-on capabilities .

Capabilities

System behavior

Azure AI Document Intelligence is a cloud-based Azure AI service that's built by using optical character recognition (OCR), Text Analytics, and Custom Text from Azure AI services. OCR is used to extract typeface and handwritten text documents. Document

Intelligence uses OCR to detect and extract information from forms and documents supported by AI to provide more structure and information to the text extraction.

Use cases

Intended uses

Document Intelligence includes features that enable customers from various industries to extract data from their documents. The following scenarios are examples of appropriate use cases:

- **Accounts payable:** A company can increase the efficiency of its accounts payable clerks by using the prebuilt invoice model and custom forms to speed up invoice data entry with a human in the loop. The prebuilt invoice model can extract key fields, such as *Invoice Total* and *Shipping Address*.
- **Insurance form processing:** A customer can train a model by using custom forms to extract a key-value pair in insurance forms and then feeds the data to their business flow to improve the accuracy and efficiency of their process. For their unique forms, customers can build their own model that extracts key values by using custom forms. These extracted values then become actionable data for various workflows within their business.
- **Bank form processing:** A bank can use the prebuilt ID model and custom forms to speed up the data entry for "know your customer" documentation, or to speed up data entry for a mortgage packet. If a bank requires their customers to submit personal identification as part of a process, the prebuilt ID model can extract key values, such as *Name* and *Document Number*, speeding up the overall time for data entry.
- **Robotic process automation (RPA):** Using the custom extraction model, customers can extract specific data needed from various types of documents. The key-value pair extracted can then be entered into various systems such as databases, or CRM systems, through RPA, replacing manual data entry. Customers can also use custom classification model to categorize documents based on their content and file them in proper location. As such, an organized set of data extracted from the custom model can be an essential first step to document RPA scenarios for businesses that handle large volumes of documents regularly.

Considerations when choosing other use cases

Consider the following factors when you choose a use case:

- **Carefully consider applying human review when sensitive data or scenarios are involved:** It's important to include a human in the loop for a manual review when you're dealing with high-stakes scenarios (e.g. affecting someone's consequential rights) or sensitive data. Machine learning models aren't perfect. Consider carefully when to include a manual review step for certain workflows. For example, identity verification at a port of entry such as airports should include human oversight.
- **Carefully consider when using for awarding or denying of benefits:** Document Intelligence was not designed or evaluated for the award or denial of benefits, and use in these scenarios may have unintended consequences. These scenarios include:
 - **Medical insurance:** This would include using healthcare records and medical prescriptions as the basis for decisions on insurance reward or denial.
 - **Loan approvals:** These include applications for new loans or refinancing of existing ones.
- **Carefully consider the supported document types and locales:** Prebuilt models have a predefined list of supported fields and are built for specific locales. Be sure to carefully check the officially supported locales and document types to ensure the best results. For example, see [Document Intelligence prebuilt receipt locales](#).

Limitations

Technical limitations, operational factors, and ranges

Prebuilt model limitations

Document Intelligence prebuilt models are used for processing specific document types and are pretrained on thousands of forms. This capability allows developers to get started and get results within minutes, with no training data or labeling required. For prebuilt models, it's important to note the list of input requirements, supported document types, and locales for each prebuilt model for optimal results. For example, refer to the prebuilt Invoice input requirements.

Custom model limitations

Document Intelligence custom models are trained using your own training data so that the model can train to your specific forms and documents. This capability is heavily dependent on the way you label the data, as well as the type of training data set you provide. For custom models, it's important to note the limits of training data set size,

document page limits, and minimum number of samples needed for each type of document.

The [Service limits](#) page contains more information on Azure AI Document Intelligence service quotas and limits for all pricing tiers. It also contains model limitations and best practices for model usage and avoiding request throttling.

Feature support

See the [Analysis features table](#) for a list of the different operations that Document Intelligence models can perform.

System performance

Accuracy

Text is composed of lines and words at the foundational level and entities such as names, prices, amounts, company names, and products at the document understanding level.

Word-level accuracy

A popular measure of accuracy for OCR is word error rate (WER), or how many words were incorrectly output in the extracted results. The lower the WER, the higher the accuracy.

WER is defined as:

$$WER = \frac{S + D + I}{N} = \frac{S + D + I}{S + D + C}$$

Where:

[] Expand table

Term	Definition	Example
S	Count of incorrect words ("substituted") in the output.	"Velvet" gets extracted as "Veivet" because "l" is detected as "i."
D	Count of missing ("deleted") words in the output.	For the text "Company Name: Microsoft," Microsoft isn't extracted because it's handwritten or hard to read.

Term	Definition	Example
I	Count of nonexistent ("inserted") words in the output.	"Department" gets incorrectly segmented into three words as "Dep arm ent." In this case, the result is one deleted word and three inserted words.
C	Count of correctly extracted words in the output.	All words that are correctly extracted.
N	Count of total words in the reference ($N=S+D+C$) excluding I because those words were missing from the original reference and were incorrectly predicted as present.	Consider an image with the sentence, "Microsoft, headquartered in Redmond, WA announced a new product called Velvet for finance departments." Assume the OCR output is ", headquartered in Redmond, WA announced a new product called Veivet for finance dep arm ents." In this case, S (Velvet) = 1, D (Microsoft) = 1, I (dep arm ents) = 3, C (11), and N = S + D + C = 13. Therefore, WER = $(S + D + I) / N = 5 / 13 = 0.38$ or 38% (out of 100).

Using a confidence value

As covered in an earlier section, the service provides a confidence value for each predicted word in the OCR output. Customers use this value to calibrate custom thresholds for their content and scenarios to route the content for straight-through processing or forwarding to the human-in-the-loop process. The resulting measurements determine the scenario-specific accuracy.

OCR system performance implications can vary by scenarios where the OCR technology is applied. We'll review a few examples to illustrate that concept.

- **Medical device compliance:** In this first example, a multinational pharmaceutical company with a diverse product portfolio of patents, devices, medications, and treatments needs to analyze FDA-compliant product label information and analysis results documents. The company might prefer a low confidence value threshold for applying human-in-the-loop because the cost of incorrectly extracted data can have significant impact for consumers and fines from regulatory agencies.
- **Image and documents processing:** In this second example, a company performs insurance and loan application processing. The customer using OCR might prefer a medium confidence value threshold because the automated text extraction is combined downstream with other information inputs and human-in-the-loop steps for a holistic review of applications.
- **Content moderation:** For a large volume of e-commerce catalog data imported from suppliers at scale, the customer might prefer a high confidence value threshold with high accuracy because even a small percentage of falsely flagged

content can generate a lot of overhead for their human review teams and suppliers.

Document and entity-level accuracy

At the document level, for example, in the case of an invoice or receipt, an error of only one character in the entire document might be rated insignificant. But if that error is in the text that represents the paid amount, the entire invoice or receipt might get flagged as incorrect.

Another useful metric is the entity error rate (EER). It's the percentage of incorrectly extracted entities, such as names, prices, amounts, and phone numbers, out of the total number of the corresponding entities in one or more documents. For example, for a total of 30 words representing 10 names, 2 incorrect words out of 30 equals 0.06 (6%) WER. But if that results in 2 names out of 10 as incorrect, the Name EER is 0.20 (20%), which is much higher than the WER.

Measuring both WER and EER is a useful exercise to get a full perspective on document understanding accuracy.

Best practices for improving system performance

Consider the following points about limitations and performance:

The service supports images and documents. For the allowable limits for number of pages, image sizes, paper sizes, and file sizes, see [What is Document Intelligence?](#).

- Many variables can affect the accuracy of the OCR results upon which Document Intelligence depends. These variables include document scan quality, resolution, contrast, light conditions, rotation, and text attributes such as size, color, and density. For example, we recommend that the image be at least 50 x 50 pixels. Refer to the product specifications and test the service on your documents to validate the fit for your situation.
- Note the limitations of each service with regard to currently supported inputs, languages and locales, and document types. For example, refer to the [Layout supported languages](#).

Best practices to improve custom model quality

When you're using the Document Intelligence custom model, you provide your own training data so that the model can train to your specific forms and documents. The

following list uses the custom form model type to share starter tips for improving your model quality.

- For filled-in forms, use examples that have all of their fields filled in.
- Use forms with real-world values that you expect to see for each field.
- If your form images are of lower quality, use a larger data set (at least 10-15 images, for example).

For a full guide and input requirements, see [Build a training data set for a custom model](#).

Evaluation of Document Intelligence

Document Intelligence's performance will vary depending on the real-world solutions for which it's implemented. To ensure optimal performance in their scenarios, customers should conduct their own evaluations. The service provides a confidence value in the range between 0 and 1 for each extracted word and key-value mapping. Customers should run a pilot or a proof of concept representing their use case to understand the range of confidence values and the extraction quality from Document Intelligence. They can then estimate the confidence value thresholds for the results to be either sent for straight-through processing (STP) or reviewed by a human. For example, the customer might submit results with confidence values greater than or equal to .80 for straight-through processing and apply human review to results with confidence values less than .80.

Evaluating and integrating Document Intelligence for your use

Microsoft wants to help you responsibly develop and deploy solutions that use Document Intelligence. We're taking a principled approach to upholding personal agency and dignity by considering the AI systems' fairness, reliability and safety, privacy and security, inclusiveness, transparency, and human accountability. These considerations are in line with our commitment to developing Responsible AI.

When you're getting ready to deploy AI-powered products or features, the following activities help to set you up for success:

- **Understand what it can do:** Fully assess the potential of Document Intelligence to understand its capabilities and limitations. Understand how it will perform in your particular scenario and context. For example, if you're using the prebuilt invoice

model, test with real-world invoices from your business processes to analyze and benchmark the results against your existing process metrics.

- **Respect an individual's right to privacy:** Only collect data and information from individuals for lawful and justifiable purposes. Only use data and information that you have consent to use for this purpose.
- **Legal review:** Obtain appropriate legal review, particularly if you plan to use it in sensitive or high-risk applications. Understand what restrictions you might need to work within, and your responsibility to resolve any issues that might come up in the future.
- **Human-in-the-loop:** Keep a human in the loop, and include human oversight as a consistent pattern area to explore. This means ensuring constant human oversight of the AI-powered product or feature and to maintain the role of humans in decision-making. Ensure that you can have real-time human intervention in the solution to prevent harm. A human in the loop enables you to manage situations when Document Intelligence does not perform as required.
- **Security:** Ensure your solution is secure and that it has adequate controls to preserve the integrity of your content and prevent unauthorized access.

Recommendations for preserving privacy

A successful privacy approach empowers individuals with information and provides controls and protection to preserve their privacy.

- If Document Intelligence is part of a solution designed to incorporate personally identifiable information (PII), think carefully about whether and how to record that data. Follow applicable national and regional regulations on privacy and sensitive data.
- Privacy managers should consider the retention policies on the extracted text and values, and the underlying documents or images of those documents. The retention policies will be tied to the intended use of each application.

Learn more about responsible AI

- Microsoft AI principles ↗
- Microsoft responsible AI resources ↗
- Microsoft Azure Learning courses on responsible AI

Learn more about Document Intelligence

- Document Intelligence overview
- Data, privacy, and security for Document Intelligence

Data, privacy, and security for Document Intelligence

Article • 07/18/2023

This article provides details regarding how Document Intelligence processes your data. Document Intelligence is designed with compliance, privacy, and security in mind. However, you are responsible for its use and the implementation of this technology. It's your responsibility to comply with all applicable laws and regulations in your jurisdiction.

How does Document Intelligence process data?

Authenticate (with subscription or API keys)

The most common way to authenticate access to Document Intelligence is by using the customer's Document Intelligence API key. Each request to the service URL must include an authentication header. This header passes along an API key (or token if applicable), which is used to validate your subscription for a service or group of services. For more information, see [Authenticate requests to Azure AI services](#).

Secure data in transit (for scanning)

All Azure AI services endpoints, including the Document Intelligence API URLs, use HTTPS URLs for encrypting data during transit. The client operating system needs to support Transport Layer Security (TLS) 1.2 for calling the endpoints. For more information, see [Azure AI services security](#).

Encrypts input data for processing

The incoming data is processed in the same region where the Document Intelligence resource was created. When you submit your documents to a Document Intelligence operation, it starts the process of analyzing the document to extract all text and identify structure and key values in a document. Your data and results are then temporarily encrypted and stored in Azure Storage.

Retrieve the results

The "Get Analyze Results" operation is authenticated against the same API key that was used to call the "Analyze" operation to ensure no other customer can access your data.

It returns the analysis job completion status, When the status shows as completed, the operation also returns the extracted results in JSON format.

Data stored by Document Intelligence

For all analysis: To facilitate asynchronous analysis and checking the completion status and returning the extracted results to the customer upon completion, the data and extracted results are stored temporarily in Azure Storage in the same region. All customers in the same region share the temporary storage. The customer's data is logically isolated from other customers with their Azure subscription and API credentials.

For customer trained models: The Custom model feature allows customers to build custom models from training data stored in customer's Azure blob storage locations. The interim outputs after analysis and labeling are stored in the same location. The trained custom models are stored in Azure storage in the same region and logically isolated with their Azure subscription and API credentials.

Deletes data: For all features, the input data and results are deleted within 24 hours and not used for any other purpose. For customer trained models, the customers can delete their models and associated metadata at any time by using the API.

To learn more about privacy and security commitments, see the [Microsoft Trust Center](#).

문서 인텔리전스 Logic Apps 워크플로 만들기

아티클 • 2023. 11. 15.

이 콘텐츠는 : v4.0(미리 보기) | 이전 버전: v3.1(GA) v3.0(GA)에 적용됩니다.

① 중요

이 자습서와 논리 앱 문서 인텔리전스 커넥터는 문서 인텔리전스 REST API v3.0 이상을 대상으로 합니다.

Azure Logic Apps는 한 줄의 코드도 작성하지 않고 워크플로를 자동화하는 데 사용할 수 있는 클라우드 기반 플랫폼입니다. 플랫폼을 사용하면 Microsoft 및 타사 애플리케이션을 앱, 데이터, 서비스 및 시스템과 쉽게 통합할 수 있습니다. Logic App은 워크플로를 개발하려는 경우에 만드는 Azure 리소스입니다. Logic App에서 수행할 수 있는 작업의 몇 가지 예는 다음과 같습니다.

- 비즈니스 프로세스 및 워크플로를 시작적으로 만듭니다.
- 워크플로를 SaaS(Software as a Service) 및 엔터프라이즈 애플리케이션과 통합합니다.
- EAI(엔터프라이즈 애플리케이션 통합), B2B(기업 간) 및 EDI(전자 데이터 교환) 작업을 자동화합니다.

자세한 내용은 [Logic Apps 개요](#)를 참조하세요.

이 자습서에서는 다음 작업을 자동화하기 위해 논리 앱 커넥터 흐름을 빌드하는 방법을 보여줍니다.

- ✓ 청구서가 OneDrive 폴더에 추가된 시기를 검색합니다.
- ✓ 문서 인텔리전스의 사전 빌드 청구서 모델을 사용하여 청구서를 처리합니다.
- ✓ 청구서에서 추출한 정보를 미리 지정된 이메일 주소로 보냅니다.

Microsoft OneDrive 계정 또는 Microsoft ShareDrive 사이트의 파일을 사용하는 워크플로를 선택하세요.

필수 조건

이 자습서를 완료하려면 다음 리소스가 필요합니다.

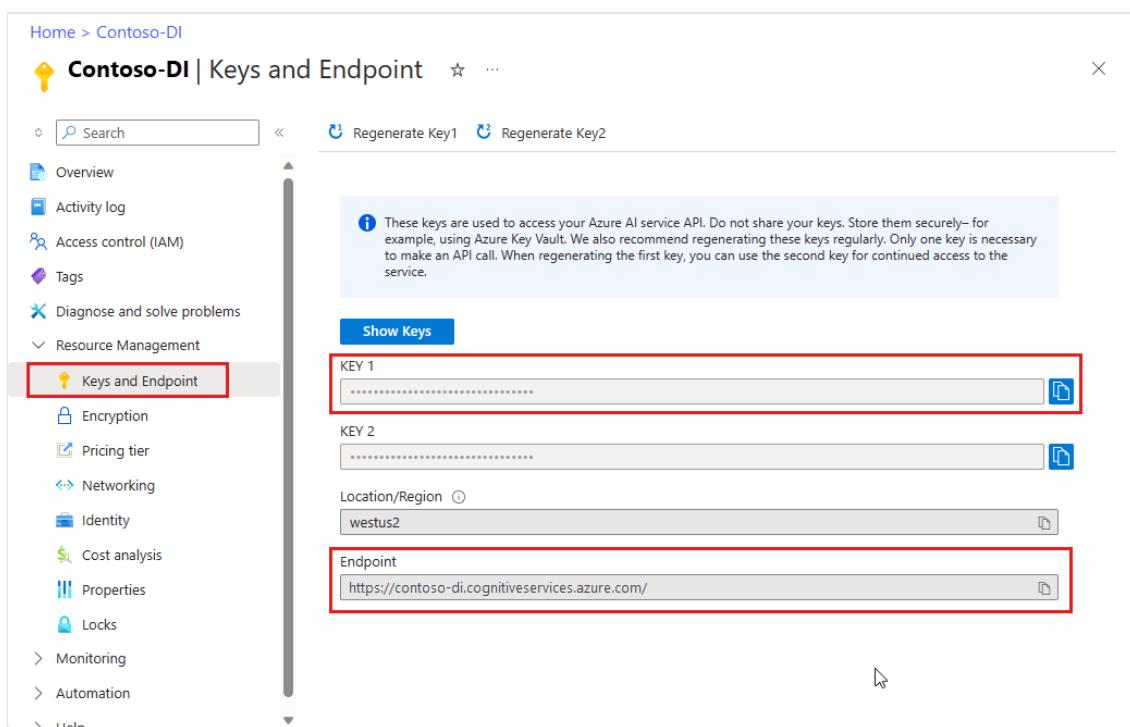
- Azure 구독. [체험 Azure 구독을 만들](#) 수 있습니다.

- 무료 OneDrive 또는 비즈니스용 OneDrive 클라우드 저장소 계정

① 참고

- OneDrive는 개인 스토리지입니다.
- 비즈니스용 OneDrive는 Office 365의 일부이며 조직을 위해 설계되었습니다. 모든 작업 파일을 저장, 공유 및 동기화할 수 있는 클라우드 저장소를 제공합니다.

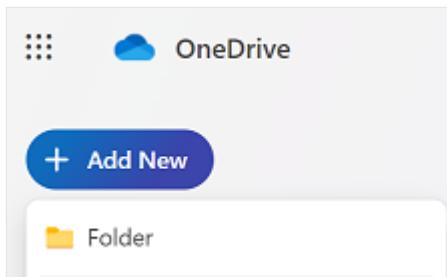
- 무료 Outlook 온라인 또는 Office 365 이메일 계정**
- Logic App을 테스트하기 위한 샘플 청구서. 이 자습서에 대한 샘플 청구서 문서를 다운로드하여 사용할 수 있습니다.
- 문서 인텔리전스 리소스. Azure를 구독하고 나면 Azure portal에서 문서 인텔리전스 리소스를 생성하여 키와 엔드포인트를 가져옵니다. 기존 문서 인텔리전스 리소스가 있는 경우 리소스 페이지로 직접 이동하세요. 체험 가격 책정 계층(F0)을 통해 서비스를 사용해 보고, 나중에 프로덕션을 위한 유료 계층으로 업그레이드할 수 있습니다.
 - 리소스가 배포된 후 리소스로 이동을 선택합니다. Azure Portal의 리소스에서 키 및 엔드포인트 값을 복사하여 Microsoft 메모장과 같은 편리한 위치에 붙여넣습니다. 애플리케이션을 문서 인텔리전스 API에 연결하려면 키와 엔드포인트 값이 필요합니다. 자세한 내용은 문서 인텔리전스 리소스 만들기를 참조하세요.



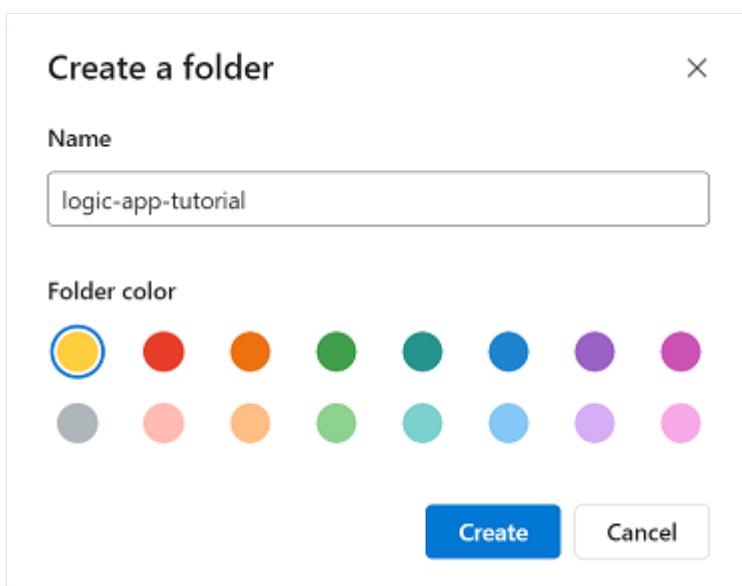
OneDrive 폴더 만들기

Logic App을 만들려면 먼저 OneDrive 폴더를 설정해야 합니다.

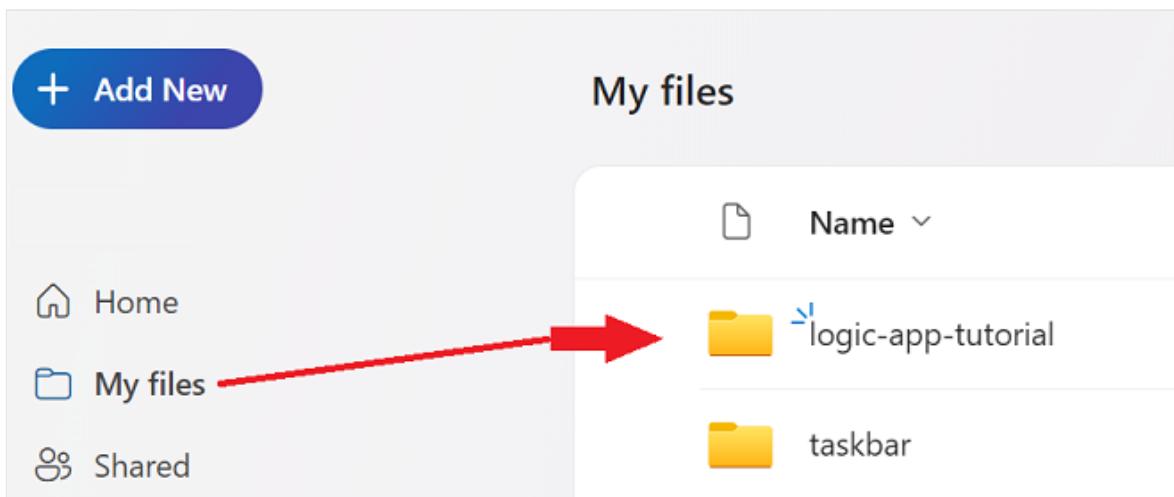
1. [OneDrive](#) 또는 [비즈니스용 OneDrive](#) 홈페이지에 로그인합니다.
2. 왼쪽 상단 사이드바에서 **+** 새로 추가 버튼을 선택하고 **폴더**를 선택합니다.



3. 새 폴더의 이름을 입력하고, **만들기**를 선택합니다.



4. 파일에 새 폴더가 표시됩니다.

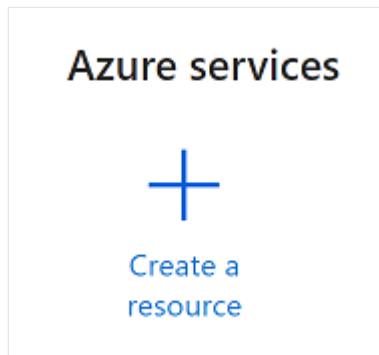


5. 이제 OneDrive 사용이 완료되었습니다.

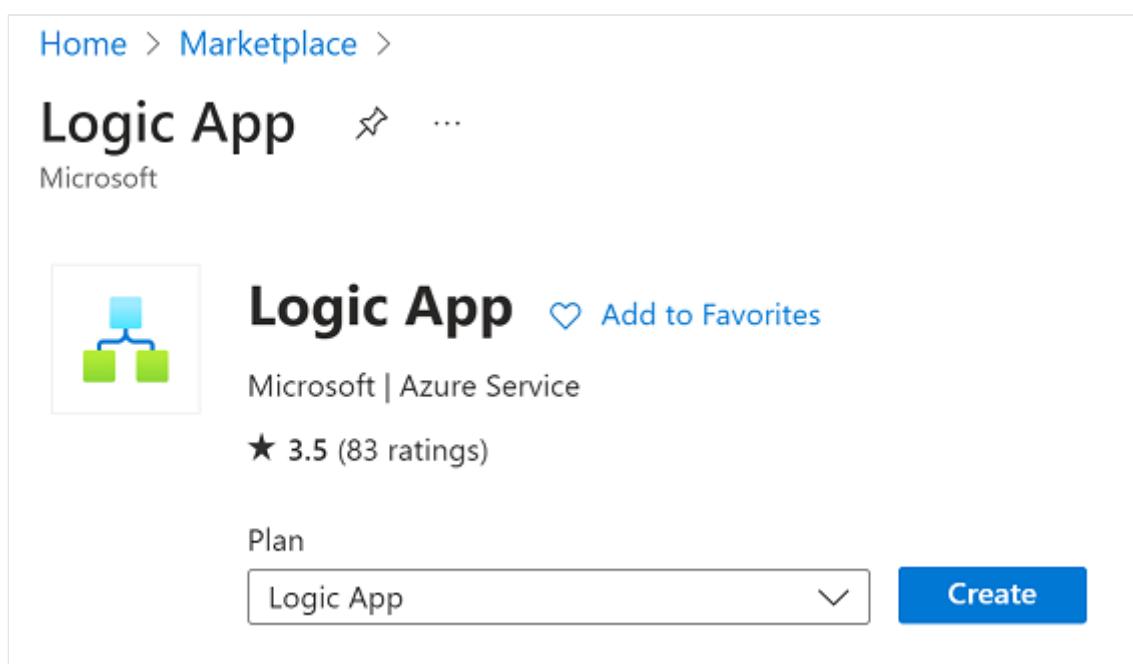
Logic App 리소스 만들기

이 시점에서 문서 인텔리전스 리소스와 OneDrive 폴더가 모두 설정되어 있어야 합니다.
이제 Logic App 리소스를 만들어야 합니다.

1. Azure Portal 로 이동합니다.
2. Azure 홈페이지에서  리소스 만들기를 선택합니다.



3. 검색 창에서 Logic App을 검색하여 선택합니다.
4. 만들기 단추를 선택합니다.



Home > Marketplace >

Logic App  

Microsoft

 **Logic App**  Add to Favorites

Microsoft | Azure Service

★ 3.5 (83 ratings)

Plan

Logic App  **Create**

5. 다음으로 Logic App 만들기 필드를 다음 값으로 채웁니다.
- **구독**. 현재 구독을 선택합니다.
 - **리소스 그룹**. 리소스가 포함된 Azure 리소스 그룹입니다. 문서 인텔리전스 리소스에 대해 동일한 리소스 그룹을 선택합니다.
 - **유형**. 사용량을 선택합니다. 사용량 리소스 종류는 글로벌 다중 테넌트 Azure Logic Apps에서 실행되며 사용량 청구 모델을 사용합니다.

- **논리 앱 이름.** 리소스 이름을 입력합니다. 설명이 포함된 이름(예: *YourNameLogicApp*)을 사용하는 것이 좋습니다.
- **게시.** 워크플로를 선택합니다.
- **지역.** 지역을 선택합니다.
- **로그 분석 사용.** 이 프로젝트의 경우 **아니요**를 선택합니다.
- **플랜 유형.** **사용량**을 선택합니다. 사용량 리소스 종류는 글로벌 다중 테넌트 Azure Logic Apps에서 실행되며 **사용량 청구 모델**을 사용합니다.
- **영역 중복.** **사용 중지됨**을 선택합니다.

6. 완료되면 다음 이미지와 비슷한 이미지가 표시됩니다(리소스 그룹, 논리 앱 이름 및 지역은 다를 수 있음). 이러한 값을 확인한 후 왼쪽 아래 모서리에서 **검토 + 만들기**를 선택합니다.

Home > Create a resource > Logic App >

Create Logic App

Create a logic app, which lets you group workflows as a logical unit for easier management, deployment and sharing of resources. Workflows let you connect your business-critical apps and services with Azure Logic Apps, automating your workflows without writing a single line of code.

Project Details

Select a subscription to manage deployed resources and costs. Use resource groups like folders to organize and manage all your resources.

Subscription *	<input type="text" value="your-subscription"/>
Resource Group *	<input type="text" value="your-resource-group"/> Create new

Instance Details

Type *

<input checked="" type="radio"/> Consumption	<input type="radio"/> Standard
<small>Looking for the classic consumption create experience? Click here</small>	

Logic App name *

Region *

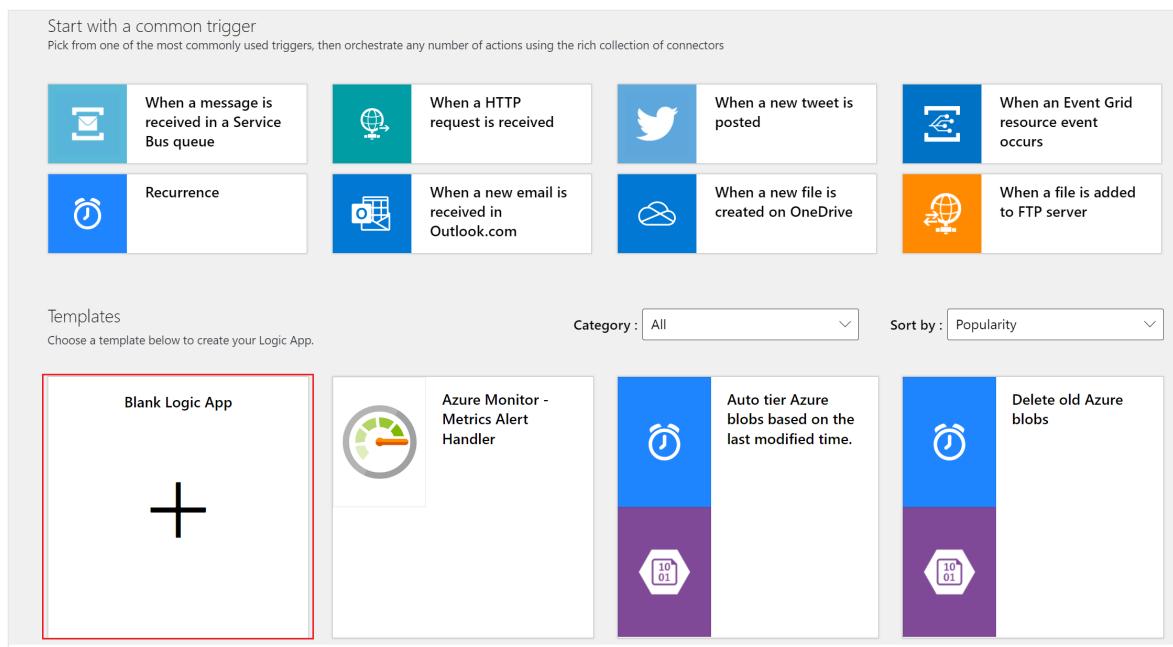
Enable log analytics *

<input type="radio"/> Yes	<input checked="" type="radio"/> No
---------------------------	-------------------------------------

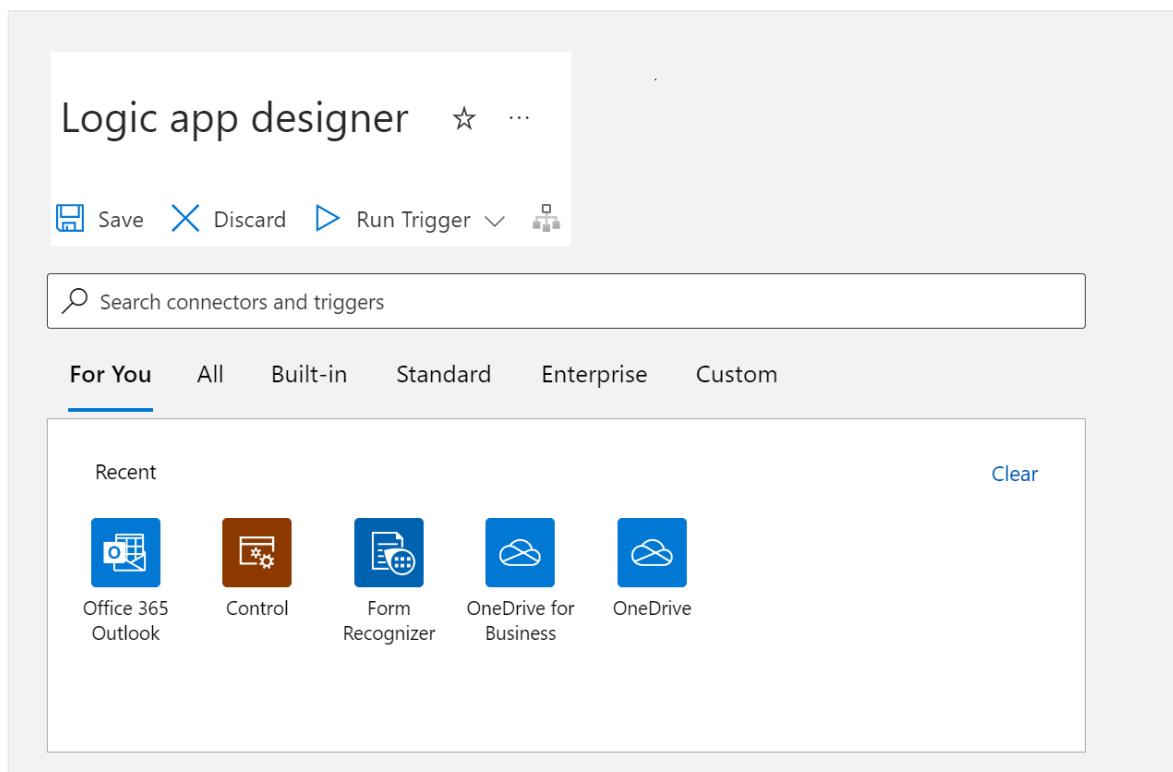
Review + create < Previous Next : Tags >

7. 간단한 유효성 검사가 실행됩니다. 성공적으로 완료되면 왼쪽 아래 모서리에서 **만들기**를 선택합니다.
8. 다음으로, **배포 진행 중**이라는 화면으로 리디렉션됩니다. Azure에 배포할 약간의 시간을 허용합니다. 몇 분이 걸릴 수 있습니다. 배포가 완료되면 **배포가 완료되었습니다**라는 배너가 표시됩니다. 이 화면이 표시되면 **리소스로 이동**을 선택합니다.

9. 마지막으로 Logic Apps 디자이너 페이지로 리디렉션됩니다. 홈 화면에서 사용할 수 있는 Logic Apps를 간략하게 소개하는 짧은 비디오가 있습니다. 논리 앱 디자인을 시작할 준비가 되면 템플릿 섹션에서 빈 논리 앱 버튼을 선택하세요.



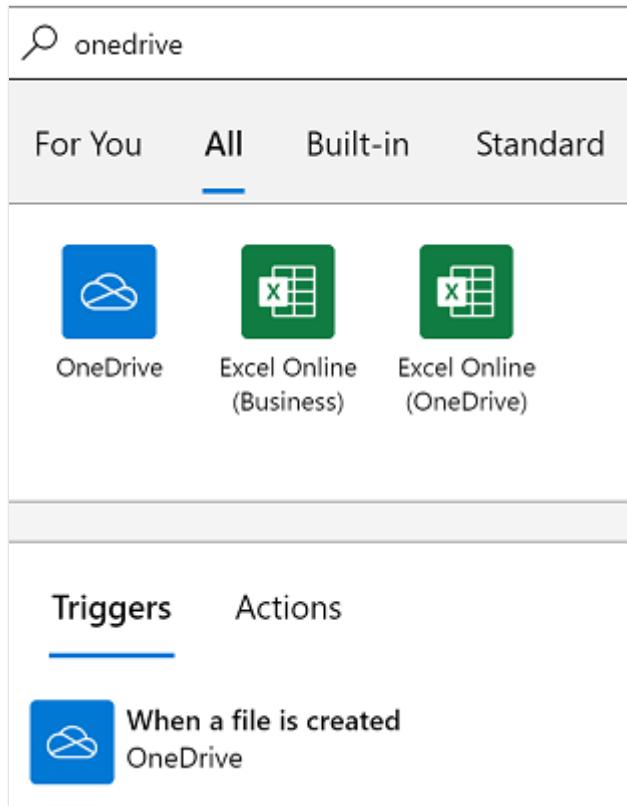
10. 다음 이미지와 유사한 화면이 표시됩니다. 이제 Logic App 디자인 및 구현을 시작할 준비가 되었습니다.



자동화 흐름 만들기

이제 논리 앱 커넥터 리소스를 설정하고 구성했으므로 자동화 흐름을 만들고 테스트해 보겠습니다.

1. 검색 창에서 OneDrive 또는 비즈니스용 OneDrive를 검색하여 선택합니다. 그런 다음 파일이 생성될 때 트리거를 선택합니다.



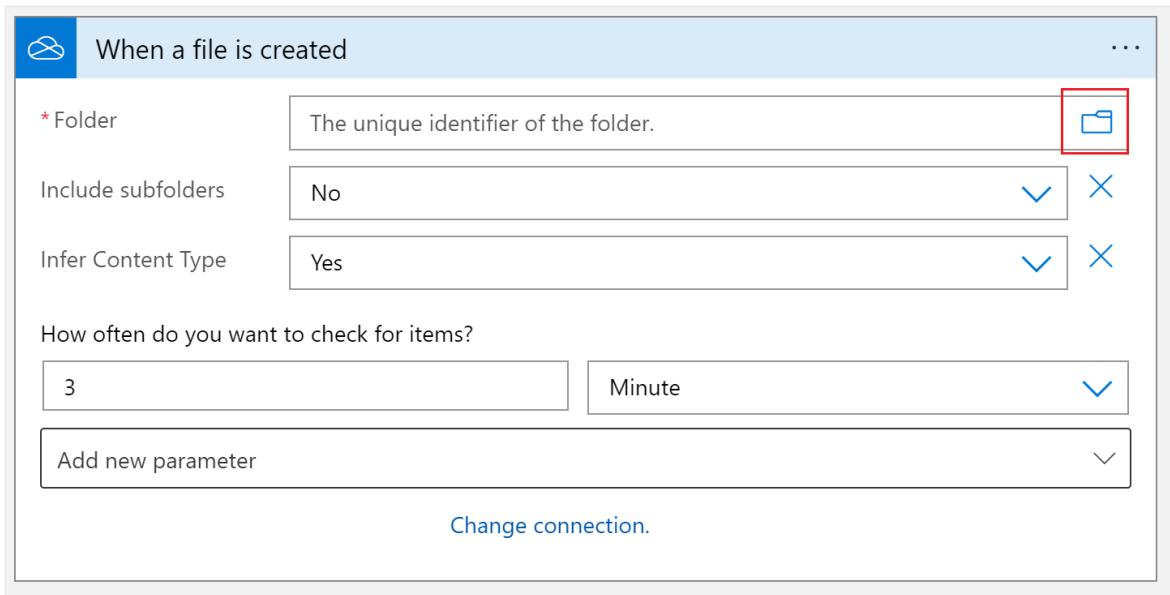
2. 다음으로 OneDrive 계정에 로그인하라는 팝업 창이 나타납니다. 로그인을 선택하고, 프롬프트에 따라 계정을 연결합니다.

💡 팁

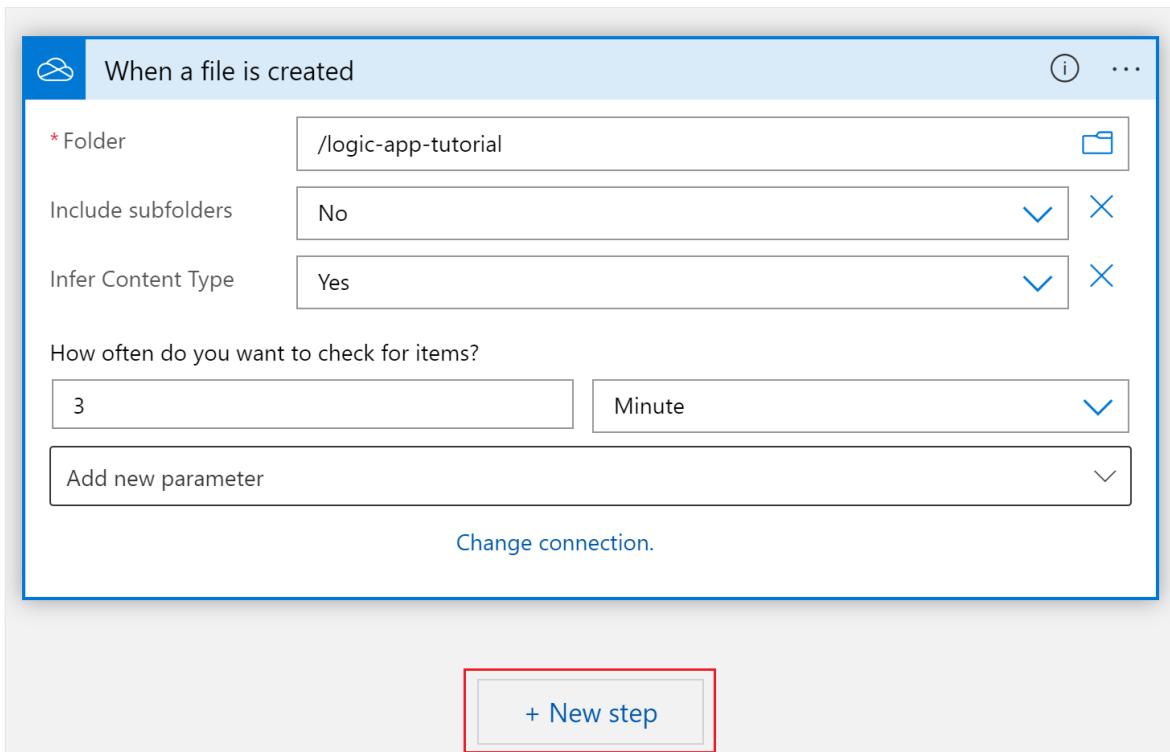
Office 365 계정을 사용하여 OneDrive 커넥터에 로그인하려고 하면 **@MICROSOFT.COM 계정을 사용하여 여기에 로그인할 수 없습니다.**라는 오류가 표시될 수 있습니다.

- OneDrive가 Office 365 계정이 아닌 Outlook.com 또는 Microsoft Live 계정으로 액세스할 수 있는 개인용 클라우드 기반 스토리지이므로 이 오류가 발생합니다.
- Office 365 계정을 사용하려는 경우 비즈니스용 OneDrive 커넥터를 사용할 수 있습니다. 비즈니스용 OneDrive 계정에서 이 프로젝트에 대한 **OneDrive 폴더를 만들었는지** 확인합니다.

3. 계정이 연결되면 이전에 OneDrive 또는 비즈니스용 OneDrive 계정에서 만든 폴더를 선택합니다. 다른 기본값은 그대로 둡니다.



4. 다음으로 새 단계를 워크플로에 추가합니다. 새로 생성된 OneDrive 노드 아래에서 새 단계 버튼을 선택합니다.

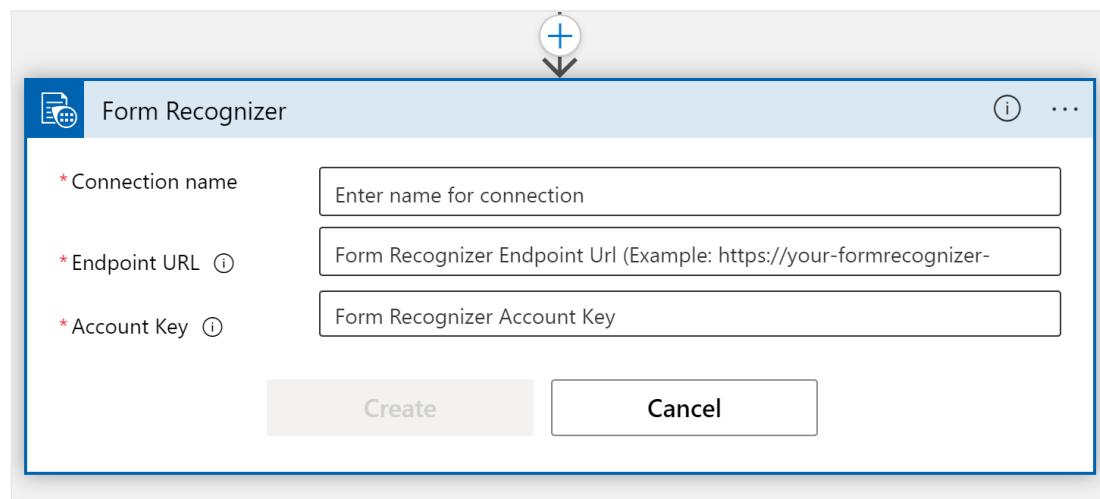


5. 논리 앱 디자이너 보기의 새 노드가 추가됩니다. **작업 선택** 검색창에서 **Form Recognizer**(문서 인텔리전스 출시 예정)를 검색하고 목록에서 **사전 빌드 또는 사용자 지정 모델에 대한 문서 분석(v3.0 API)**을 선택합니다.

The screenshot shows the Microsoft Logic Apps Designer interface. At the top, there are two tabs: 'Triggers' and 'Actions'. The 'Actions' tab is selected, indicated by an underline. Below the tabs, there is a list of actions categorized under 'Form Recognizer'. The actions are: 'Analyze Business Card', 'Analyze Custom Form', 'Analyze Document for Prebuilt or Custom models (v3.0 API)' (which is highlighted with a red border), 'Analyze ID Document', 'Analyze Invoice', and 'Analyze Layout'. Each action has a small icon to its left and a help icon (info symbol) to its right.

6. 이제 연결을 생성하는 창이 표시됩니다. 특히 문서 인텔리전스 리소스를 Logic Apps 디자이너 스튜디오에 연결합니다.

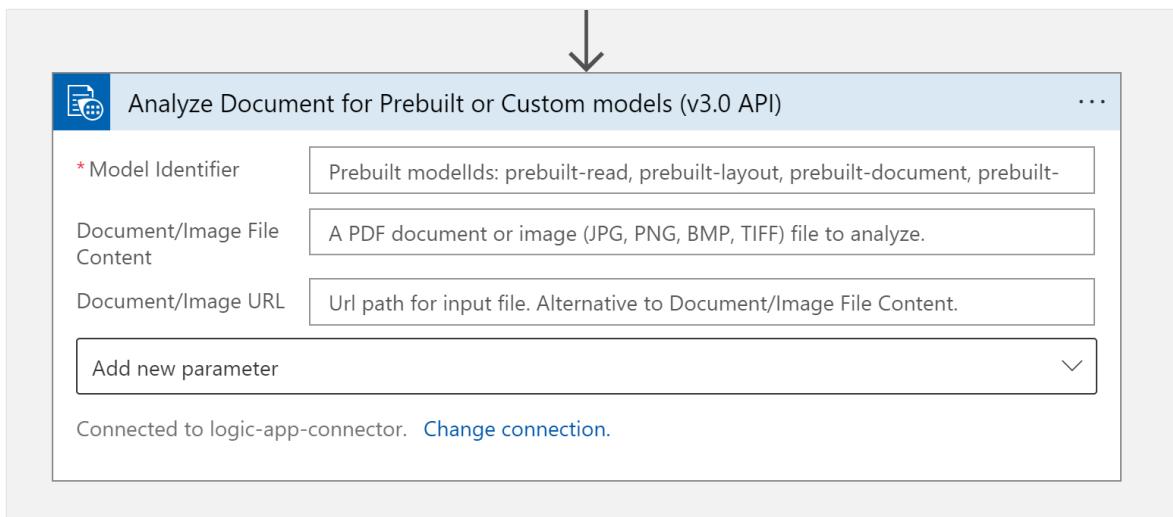
- **연결 이름**을 입력합니다. 기억하기 쉬운 이름이어야 합니다.
- 이전에 복사한 문서 인텔리전스 리소스 **엔드포인트 URL**과 **계정 키**를 입력합니다. 이전에 이 단계를 건너뛰었거나 문자열을 잃어버린 경우 문서 인텔리전스 리소스로 돌아가서 다시 복사할 수 있습니다. 완료되면 **만들기**를 선택합니다.



① 참고

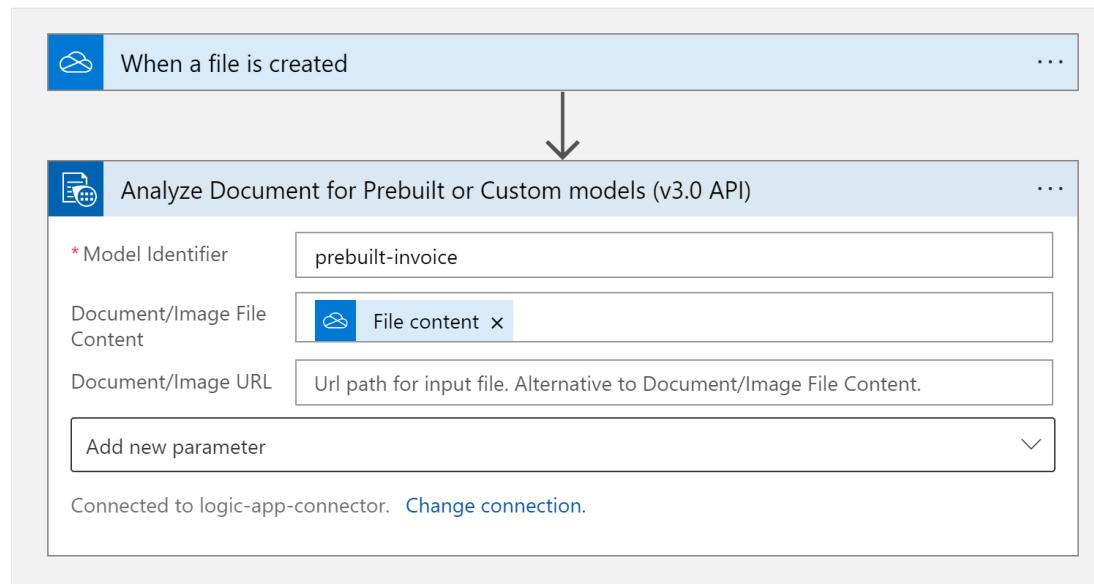
이미 자격 증명으로 로그인한 경우 이전 단계를 건너뜁니다.

7. 다음으로, 사전 빌드된 모델이나 사용자 지정 모델에 대한 문서 분석(v3.0 API) 커넥터에 대한 선택 매개 변수 창이 표시됩니다.



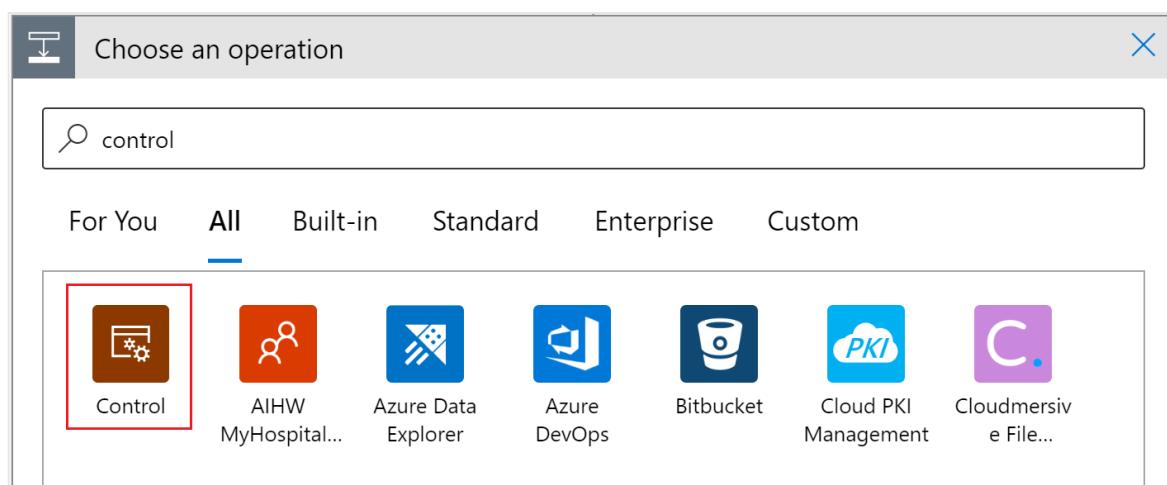
8. 다음과 같이 필드를 완성하세요.

- 모델 식별자.** 호출할 모델을 지정하세요. 이 경우 사전 빌드된 청구서 모델을 호출하므로 **prebuilt-invoice**를 입력하세요.
- 문서/이미지 파일 콘텐츠.** 이 필드를 선택합니다. 동적 콘텐츠 팝업이 나타납니다. 그렇지 않은 경우 입력란 아래에 있는 **동적 콘텐츠 추가** 버튼을 선택하고 **파일 콘텐츠**를 선택합니다. 이 단계는 기본적으로 분석할 파일을 문서 인텔리전스의 사전 빌드된 청구서 모델로 보내는 것입니다. **파일 콘텐츠 배지가 문서/이미지 파일 콘텐츠** 필드에 표시되면 이 단계를 올바르게 완료한 것입니다.
- 문서/이미지 URL.** 이미 OneDrive 폴더에서 직접 파일 콘텐츠를 가리키고 있으므로 이 프로젝트에서는 이 필드를 건너뛰세요.
- 새 매개 변수를 추가합니다.** 이 프로젝트에서는 이 필드를 건너뛰세요.

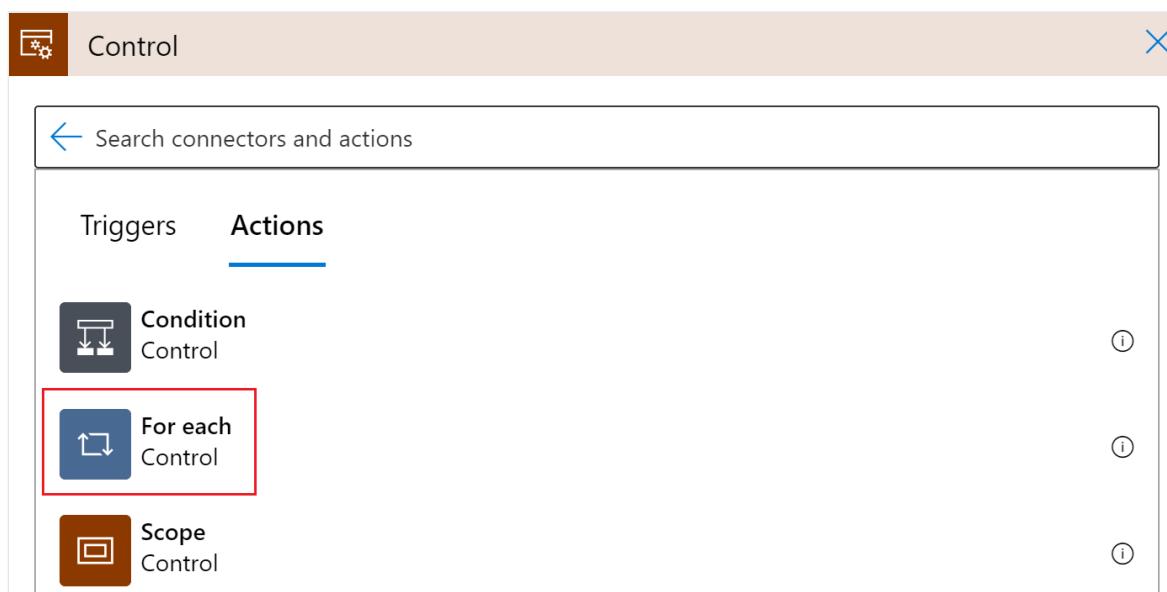


9. 몇 가지 단계를 더 추가해야 합니다. 다시 한 번, **+ 새 단계** 단추를 선택하여 다른 작업을 추가합니다.

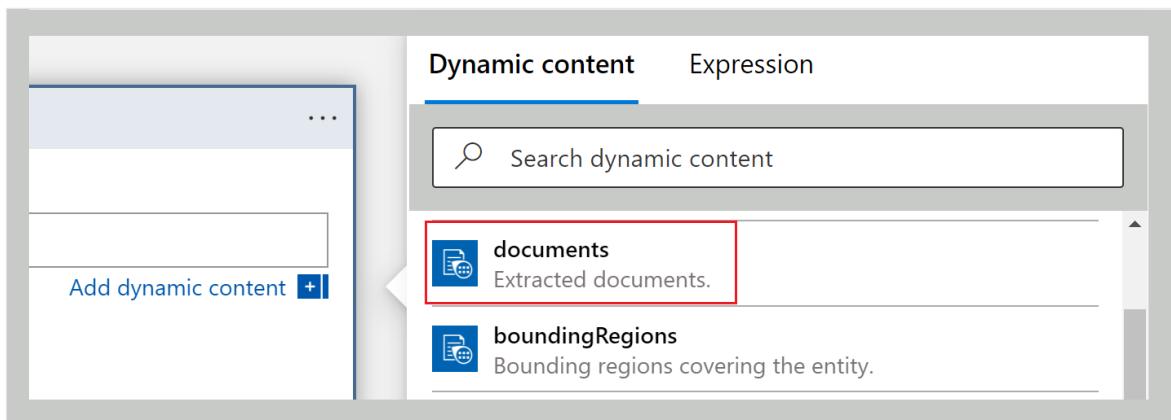
10. **작업 선택** 검색창에 컨트롤을 입력하고 **컨트롤 타일**을 선택합니다.



11. 아래로 스크롤하여 **컨트롤** 목록에서 각 컨트롤에 대해 타일을 선택합니다.



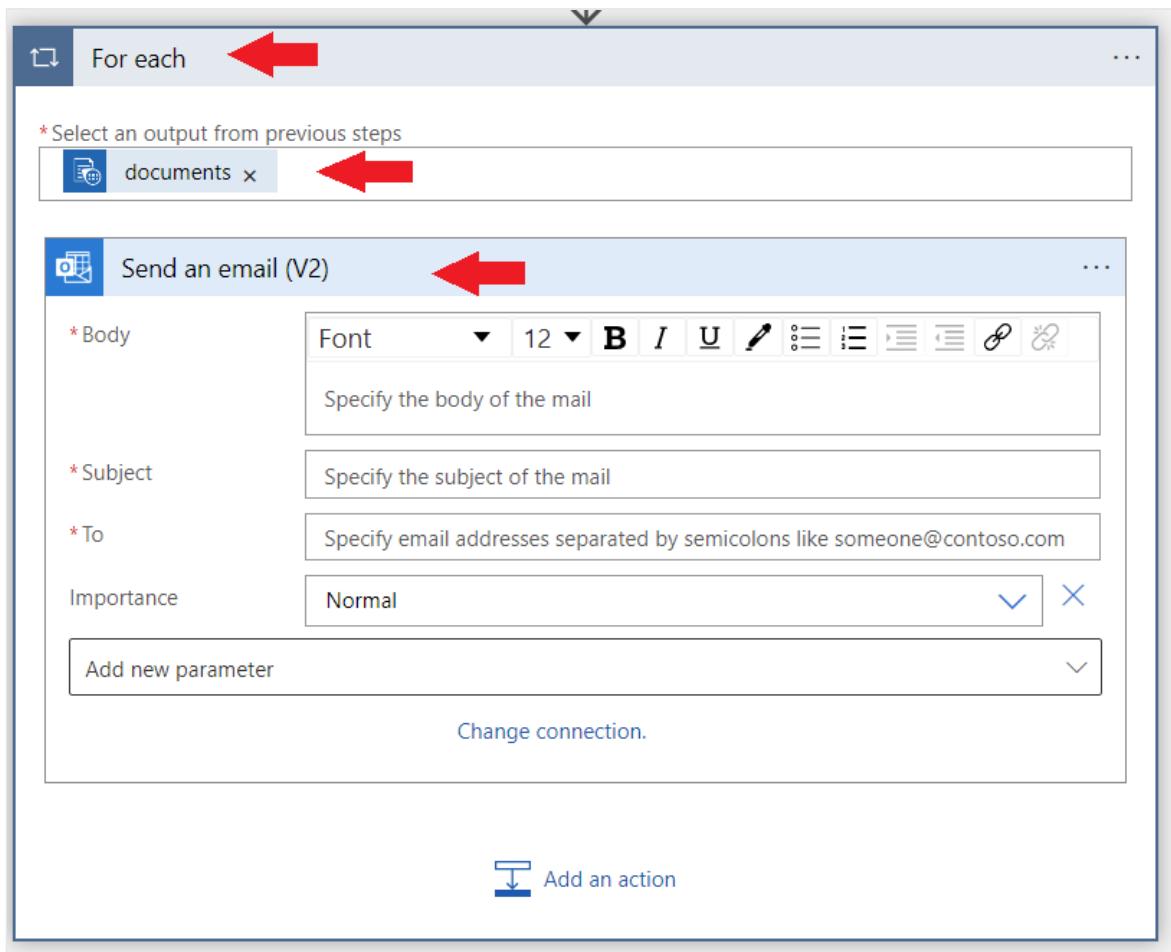
12. 각 단계 창에는 이전 단계의 결과 선택이라는 필드가 있습니다. 이 필드를 선택합니다. 동적 콘텐츠 팝업이 나타납니다. 그렇지 않은 경우 입력란 아래에 있는 동적 콘텐츠 추가 버튼을 선택하고 문서를 선택합니다.



13. 이제 각 단계 창에서 작업 추가를 선택하세요.

14. 작업 선택 검색창에 Outlook을 입력하고 Outlook.com(개인용) 또는 Office 365 Outlook(업무용)을 선택합니다.

15. 작업 목록에서 이메일 보내기(V2)를 찾을 때까지 아래로 스크롤하고 이 작업을 선택합니다.



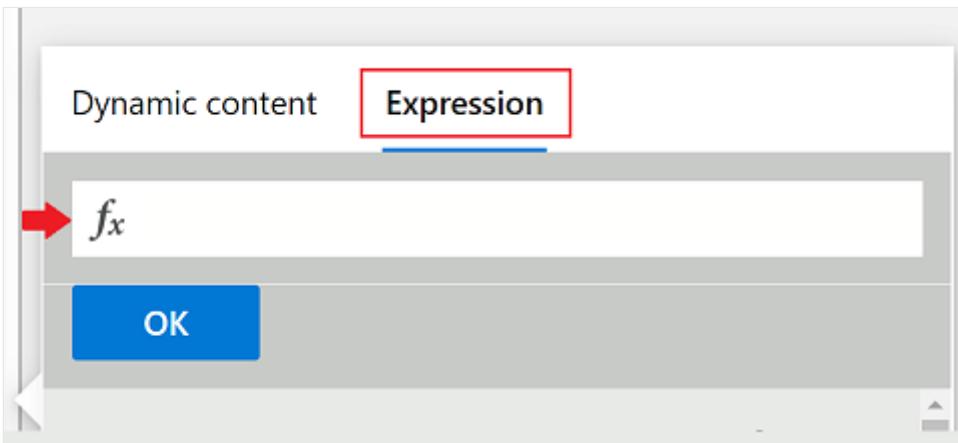
16. OneDrive와 마찬가지로 Outlook 또는 Office 365 Outlook 계정에 로그인하라는 메시지가 표시됩니다. 로그인하면 문서 인텔리전스가 청구서에서 추출하는 동적 콘텐츠로 전자 메일의 형식을 지정할 창이 표시됩니다.

17. 다음 식을 사용하여 일부 필드를 완성하겠습니다.

Power Apps 수식

```
items('For_each')?['fields']?[FIELD-NAME]?['content']
```

18. 특정 필드에 액세스하려면 **동적 콘텐츠 추가** 버튼을 선택하고 **식** 탭을 선택합니다.



19. **fx** 상자에 위 수식을 복사하여 붙여넣고 **FIELD-NAME**을 추출하려는 필드 이름으로 바꿉니다. 사용 가능한 필드의 전체 목록은 해당 API의 개념 페이지를 참조하세요. 이 경우에는 [사전 빌드된 청구서 모델 필드 추출 값](#)을 사용합니다.

20. 거의 완료되었습니다! 다음 필드를 다음과 같이 변경합니다.

- **받는 사람**: 개인 이메일 주소, 비즈니스 이메일 주소 또는 액세스 권한이 있는 다른 이메일 주소를 입력합니다.
- **주제**: **청구서 발신처**를 입력하고 다음 식을 추가합니다.

Power Apps 수식

```
items('For_each')?['fields']?[VendorName]?['content']
```

- **본문** 청구서에 대한 특정 정보를 추가합니다.
 - **청구서 ID**:를 입력하고 이전과 동일한 방법을 사용하여 다음 식을 추가합니다.

Power Apps 수식

```
items('For_each')?['fields']?['InvoiceId']?['content']
```

- 새 줄에 **청구서 기한**:을 입력하고 다음 식을 추가합니다.

Power Apps 수식

```
items('For_each')?['fields']?['DueDate']?['content']
```

- **미결제 금액**:을 입력하고 다음 식을 추가합니다.

Power Apps 수식

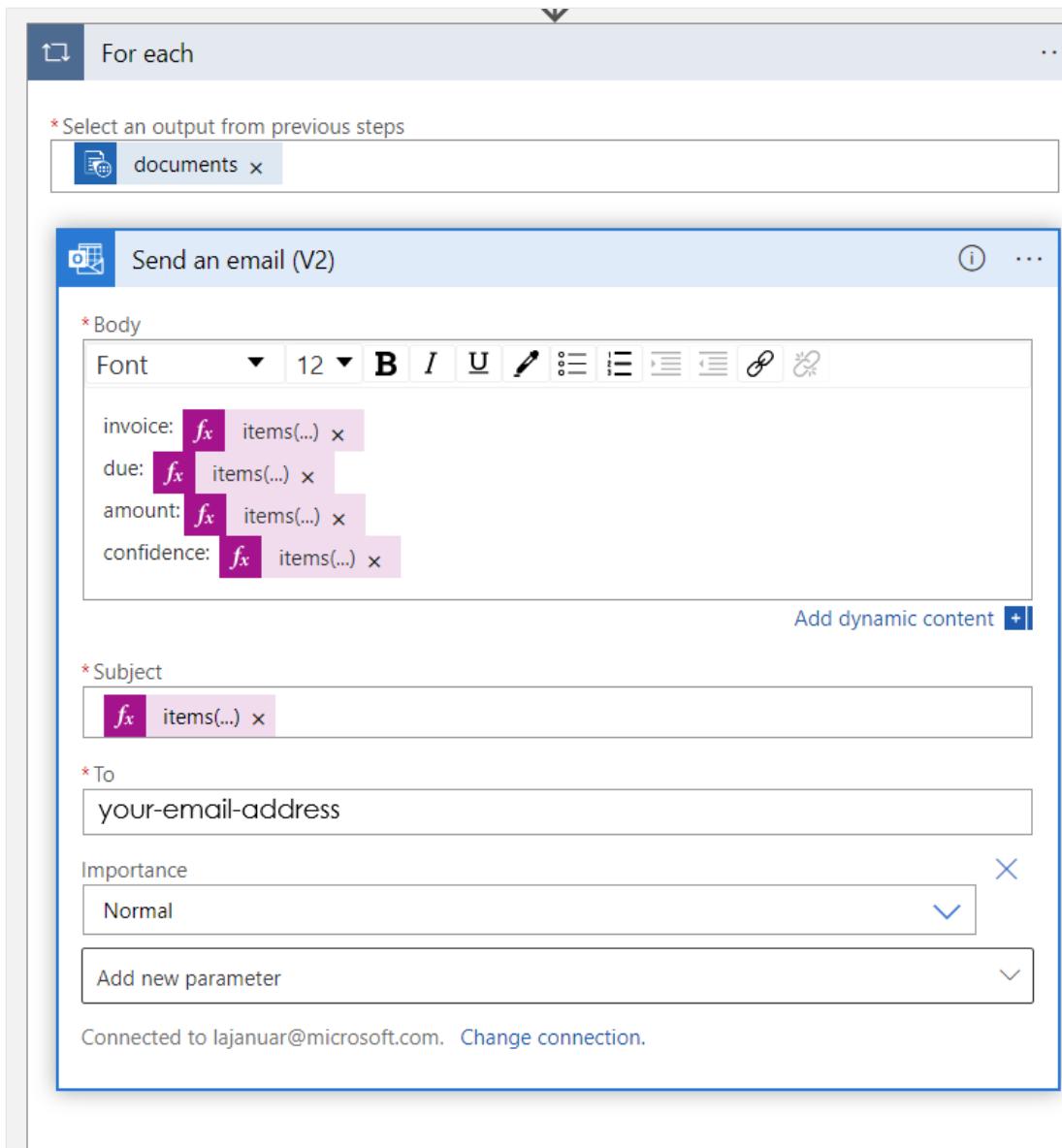
```
items('For_each')?['fields']?['AmountDue']?['content']
```

- 마지막으로, 미결제 금액이 중요한 숫자이므로 이 추출에 대한 신뢰도 점수도 이메일로 보내려고 합니다. 이를 수행하려면 **미결제 금액(신뢰도)**:을 입력하고 다음 식을 추가하세요.

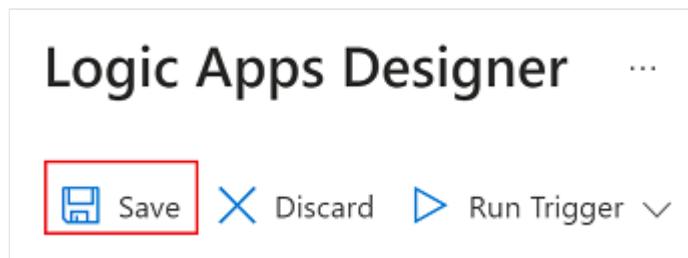
Power Apps 수식

```
items('For_each')?['fields']?['AmountDue']?['confidence']
```

- 완료되면 창은 다음 이미지와 유사하게 표시됩니다.



21. 왼쪽 위 모서리에서 저장을 선택합니다.



① 참고

- 이 현재 버전은 PDF당 하나의 청구서만 반환합니다.
- 나중에 PDF에서 둘 이상의 청구서를 반환할 수 있는 출력 형식을 활성화하려면 이메일 보내기 작업에 "각 루프마다"가 필요합니다.

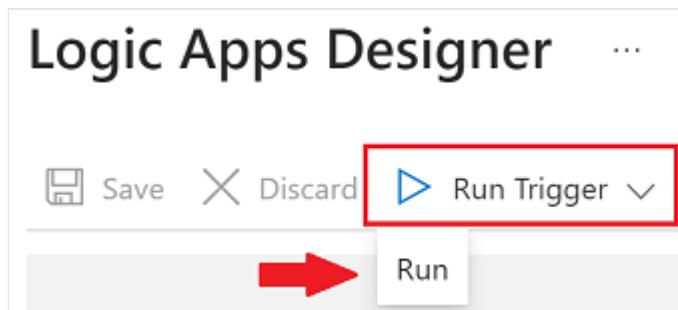
자동화 흐름 테스트

먼저 수행한 작업을 빠르게 검토한 후에 흐름을 테스트해 보겠습니다.

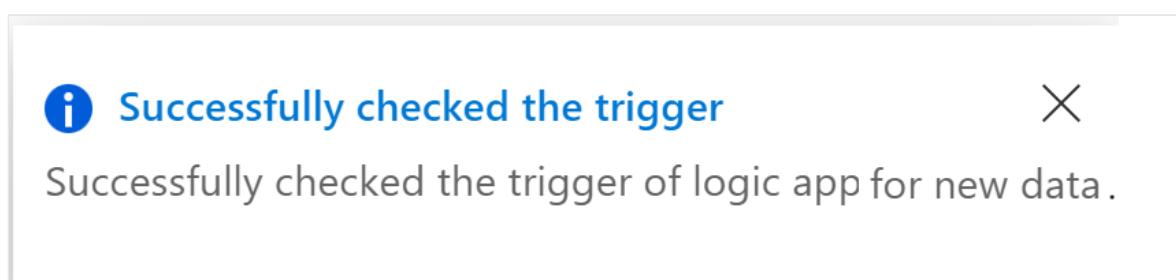
- ✓ 이 시나리오에서는 트리거를 만들었습니다. OneDrive 계정의 미리 지정된 폴더에 파일이 생성되면 트리거가 활성화됩니다.
- ✓ 흐름에 문서 인텔리전스 작업을 추가했습니다. 이 시나리오에서는 청구서 API를 사용하여 OneDrive 폴더의 청구서를 자동으로 분석하기로 결정했습니다.
- ✓ 흐름에 Outlook.com 작업을 추가했습니다. 분석된 청구서 데이터 중 일부를 미리 결정된 이메일 주소로 보냈습니다.

이제 흐름이 만들어졌으므로 마지막으로 수행해야 하는 작업은 해당 흐름을 테스트하고 예상되는 동작을 수행하는지 확인하는 것입니다.

1. 논리 앱을 테스트하려면 먼저 새 탭을 열고 이 자습서 시작 부분에서 설정한 OneDrive 폴더로 이동합니다. 이 파일을 [샘플 청구서](#)라는 OneDrive 폴더에 추가합니다.
2. Logic Apps 디자이너 탭으로 돌아가서 **트리거 실행** 단추를 선택하고, 드롭다운 메뉴에서 **실행**을 선택합니다.



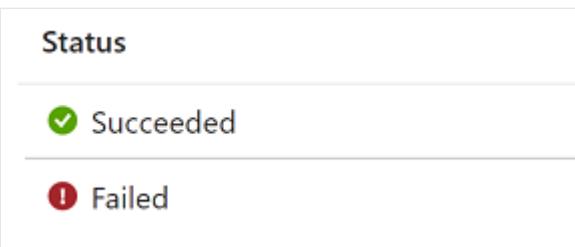
3. 트리거가 성공했음을 나타내는 메시지가 오른쪽 상단에 표시됩니다.



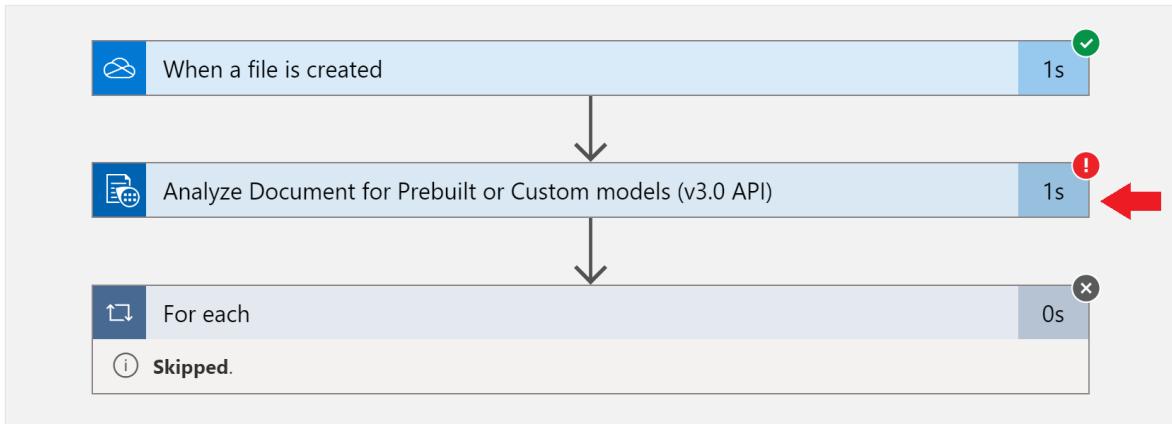
4. 왼쪽 위 모서리에 있는 앱 이름 링크를 선택하여 논리 앱 개요 페이지로 이동합니다.



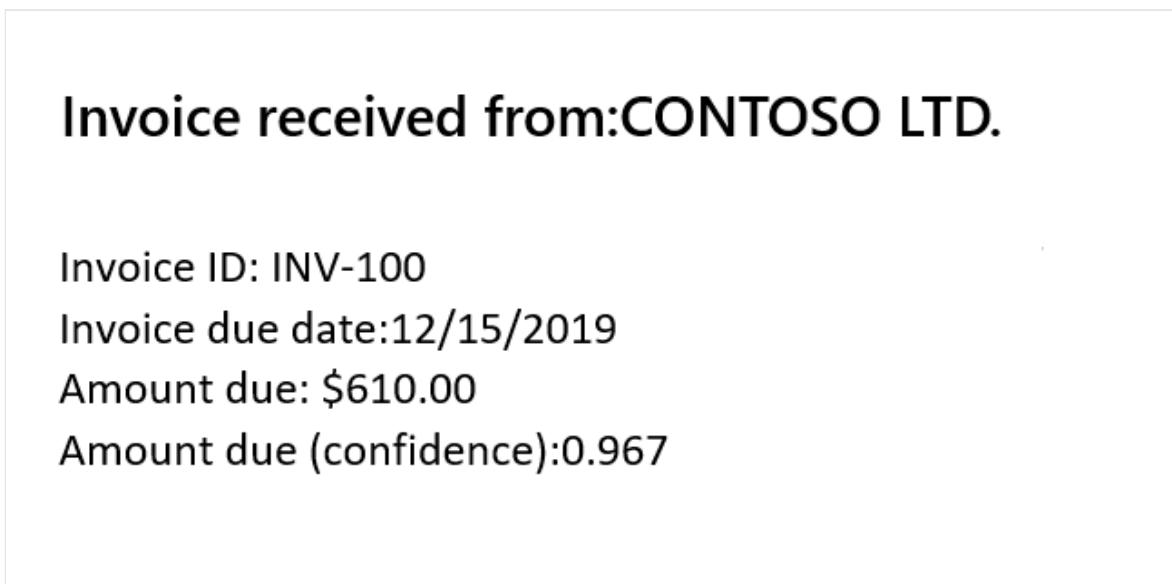
5. 상태를 확인하여 실행이 성공했는지 실패했는지 확인하세요. 상태 표시기를 선택하여 어떤 단계가 성공했는지 확인할 수 있습니다.



6. 실행이 실패한 경우 실패한 단계를 확인하여 올바른 정보를 입력했는지 확인하세요.



7. 성공적으로 실행되면 이메일을 확인하세요. 지정한 정보가 포함된 새 이메일이 있습니다.



8. 사용을 중지한 후에는 Logic App을 [사용하지 않도록 설정하거나 삭제해야 합니다.](#)



축하합니다! 이 자습서를 공식적으로 완료했습니다.

다음 단계

[문서 인텔리전스 모델에 대해 자세히 알아보기](#)

문서 처리 사용자 지정 모델 만들기

아티클 • 2024. 02. 20.

요구 사항 검토 후 문서 처리 모델 작성을 시작할 수 있습니다.

AI Builder에 로그인

1. [Power Apps](#) 또는 [Power Automate](#)에 로그인합니다.

2. 왼쪽 창에서 ... 더보기 > AI 허브를 선택합니다.

3. AI 기능 검색에서 AI 모델을 선택합니다.

(선택 사항) 쉽게 액세스할 수 있도록 AI 모델을 메뉴에 영구적으로 유지하려면 핀 아이콘을 선택합니다.

4. 문서에서 사용자 지정 정보 추출을 선택합니다.

5. 사용자 지정 모델 만들기를 선택합니다.

6. 단계별 마법사는 문서에서 추출하려는 모든 데이터를 나열하도록 요청하여 프로세스를 설명합니다. 사용자 고유의 문서를 사용하여 모델을 만들려면 동일한 레이아웃을 사용하는 다섯 가지 이상의 예제가 있는지 확인합니다. 그렇지 않은 경우에는 [샘플 데이터를 사용](#)하여 모델을 만들 수 있습니다.

7. 학습을 선택합니다.

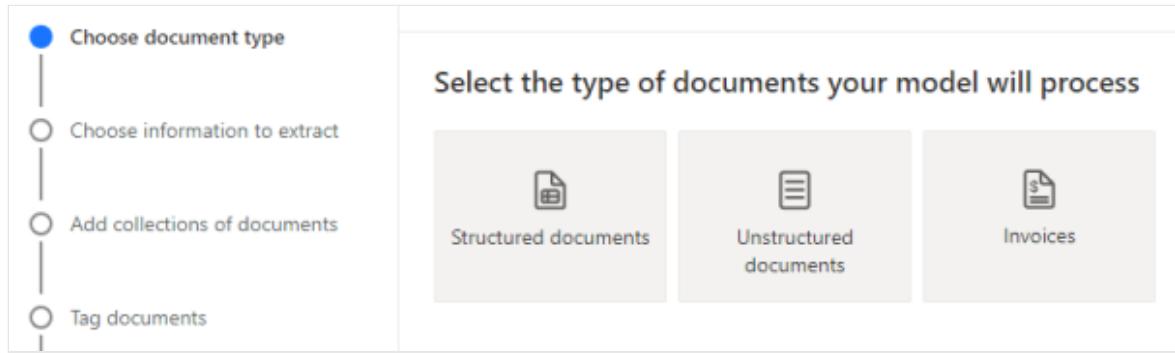
8. 빠른 테스트를 선택하여 모델을 테스트합니다.

문서의 유형 선택

문서 유형 선택 단계에서 데이터 추출을 자동화하기 위해 AI 모델을 구축하려는 문서 유형을 선택합니다. 다음과 같은 세 가지 옵션이 있습니다.

- 구조화 및 반구조화된 문서.** 구조화 및 반구조화 문서는 지정된 레이아웃에 대해 필드, 테이블, 확인란 및 기타 항목이 유사한 위치에서 찾을 수 있는 문서입니다. 구조화 및 반구조화 문서의 예로는 송장, 구매 주문서, 배송 주문서, 세금 문서 등이 있습니다.
- 비구조적 및 자유 양식 문서.** 구조화되지 않은 문서는 정해진 구조가 없는 문서로, 일반적으로 다양한 단락 수가 있는 무료 문서입니다. 비구조적 문서의 예로는 계약서, 작업 명세서, 서신 등이 있습니다.

- **송장**, **기본** 또는 제대로 추출되지 않은 문서 샘플 외에 추출할 새 필드를 추가하여 사전 제작된 송장 처리 모델의 동작을 강화합니다.



추출할 정보 정의

추출할 정보 선택 화면에서 모델이 추출하도록 학습시키려는 필드, 테이블 및 확인란을 정의합니다. +**추가** 버튼을 눌러 정의를 시작합니다.

Add

ents. For ts.

Text field

Number field

Date field

Checkbox

Table

Mortgage Statement

Recipient's / Lender's
Contoso, Ltd.
4567 Main St.
Buffalo, NY 90852

Recipient's / Lender's TIN 750-01-1829	Payer's/Borrower's TIN 306-14-5298
1.Mortgage interest \$ 13,345.34	
2.Outstanding mortgage \$ 764,321.33	3.Mortgage origination date 10/08/2019
4.Refund of overpaid \$ 15.01	5.Mortgage insurance \$ 1,222.05
6. Address 1025 Fifth St. Sunnyvale, CA 27673	

Overview

Use this to extract printed or handwritten text from your documents.
For example: Name, Address, Phone number.

Next Cancel

1. 각 **텍스트** 필드에 대해 모델에서 사용할 필드의 이름을 제공합니다.

2. 각 **숫자** 필드에 대해 모델에서 사용할 필드의 이름을 제공합니다.

또한 소수 구분 기호로 점(.) 또는 쉼표(,) 형식을 정의합니다.

3. 각 날짜 필드에 대해 모델에서 사용할 필드의 이름을 제공합니다.

또한 (연, 월, 일) 또는 (월, 일, 연) 또는 (일, 월, 연) 형식을 정의합니다

4. 각 확인란에 대해 모델에서 사용할 확인란의 이름을 제공합니다.

문서에서 확인할 수 있는 각 항목에 대해 별도의 확인란을 정의합니다.

5. 각 테이블에 대해 테이블 이름을 제공합니다.

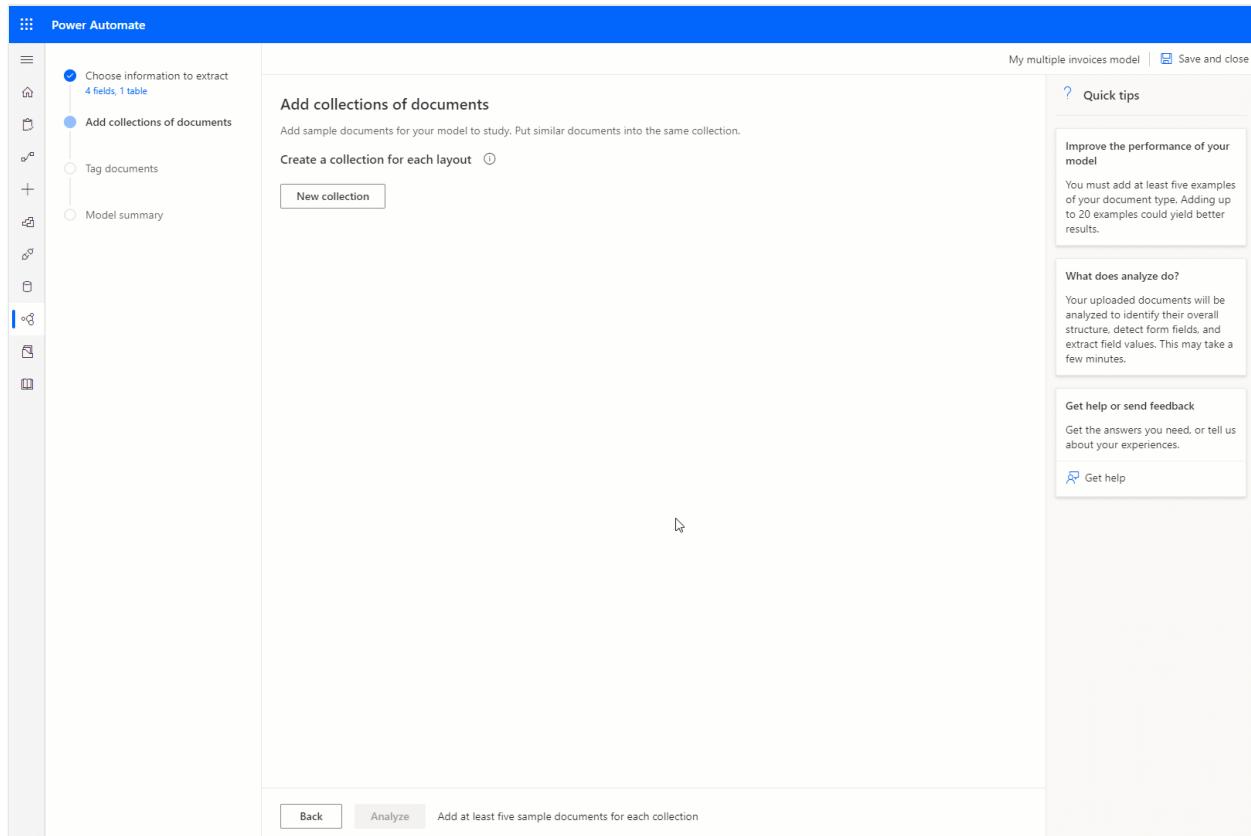
또한 모델이 추출해야 하는 다른 열을 정의합니다.

① 참고

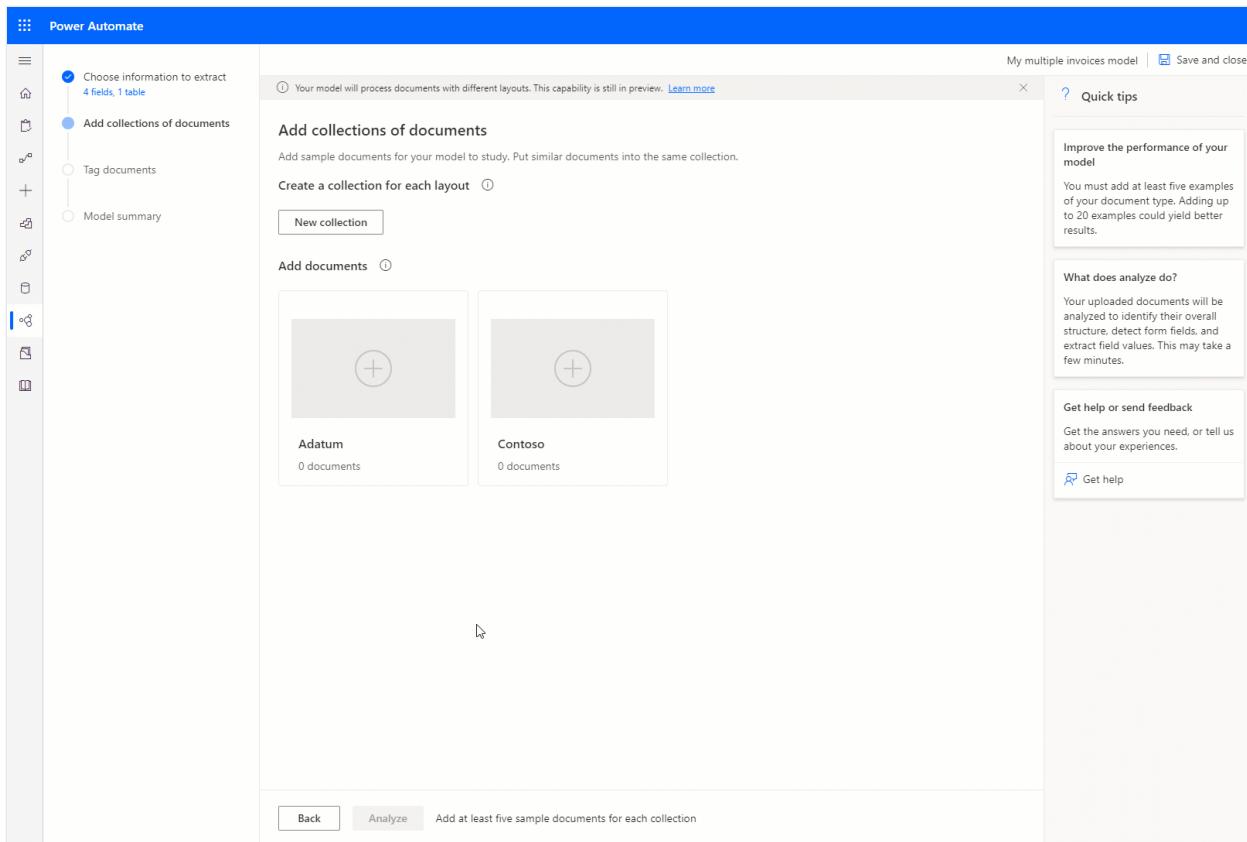
사용자 지정 송장 모델에는 편집할 수 없는 기본 필드가 함께 제공됩니다.

컬렉션별로 문서 그룹화

컬렉션은 레이아웃이 같은 문서의 그룹입니다. 모델에서 처리할 문서 레이아웃만큼 컬렉션을 만듭니다. 예를 들어 각각 고유한 송장 템플릿이 있는 두 개의 서로 다른 공급 업체의 송장을 처리하는 AI 모델을 구축하는 경우 두 개의 컬렉션을 만듭니다.



생성하는 각 컬렉션에 대해 컬렉션당 최소 5개의 샘플 문서를 업로드해야 합니다. 현재 JPG, PNG 및 PDF 파일 형식의 파일이 허용됩니다.



① 참고

모델당 최대 200개의 컬렉션을 만들 수 있습니다.

다음 단계

[문서 처리 모델의 문서에 태그 지정](#)

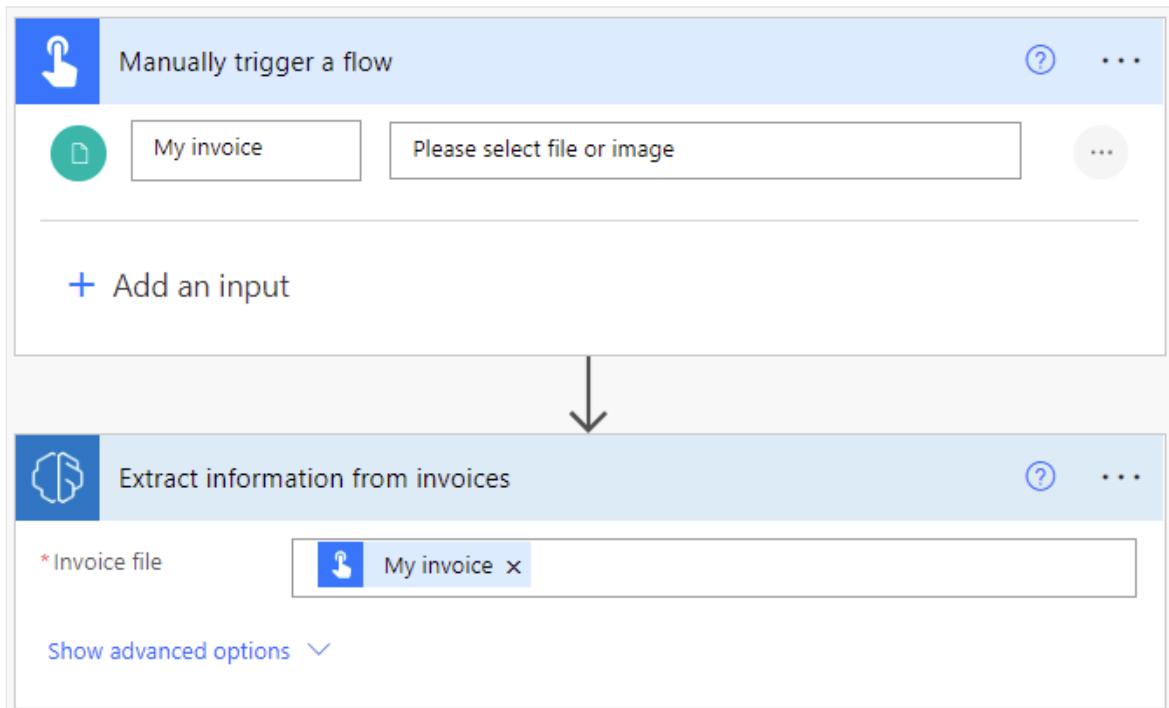
참조 항목

[학습: AI Builder로 사용자 지정 문서 처리\(모듈\)](#)

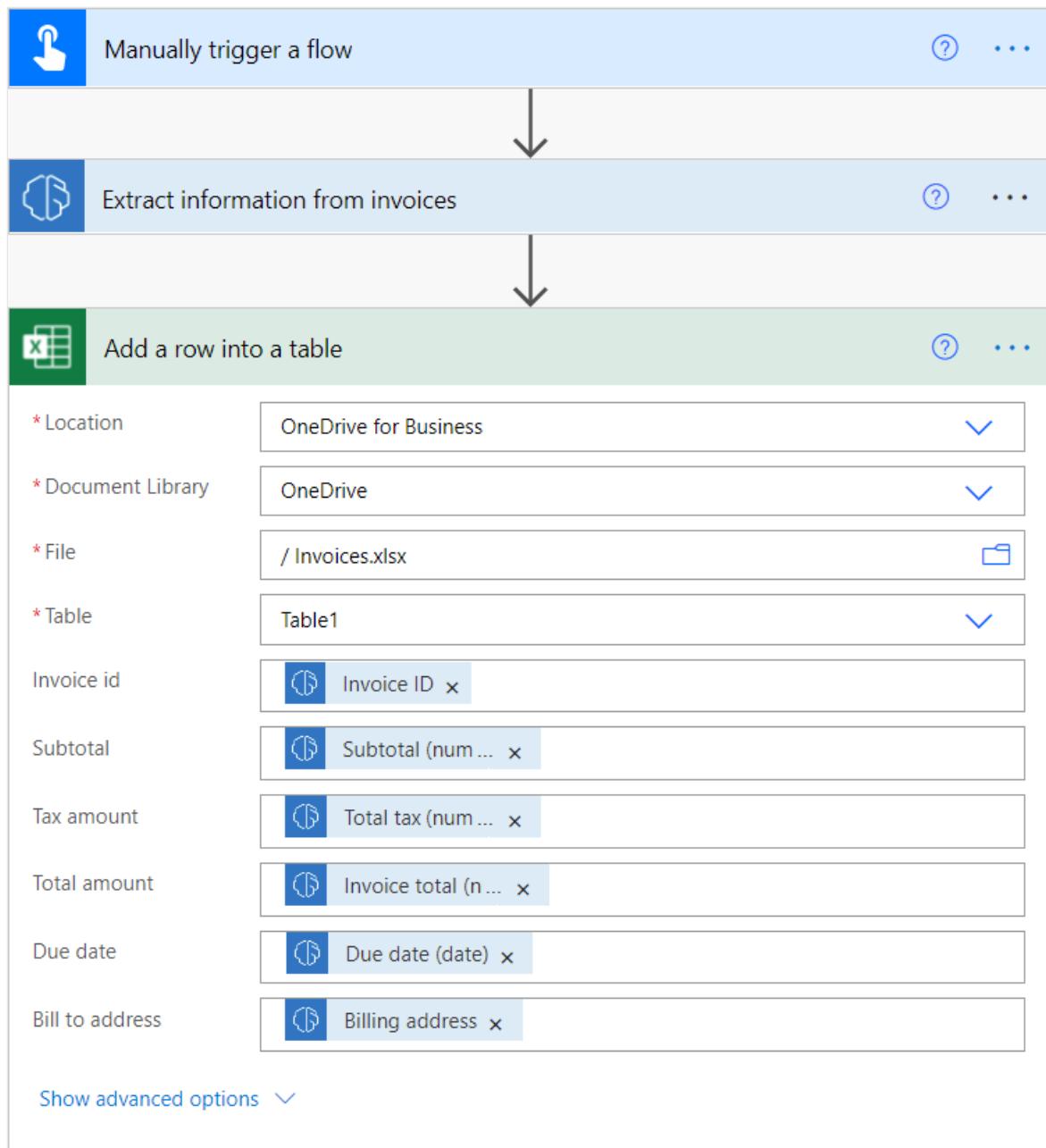
Power Automate에서 송장 처리 미리 빌드된 모델 사용

아티클 • 2023. 03. 16.

1. Power Automate [에](#) 로그인합니다.
2. 왼쪽 창에서 **내 흐름**을 선택한 다음 **새 흐름**>**인스턴트 클라우드 흐름**을 선택합니다.
3. 흐름에 이름을 지정한 다음 **이 흐름을 트리거하는 방법** 아래에서 **수동으로 흐름 트리거**를 선택하고 **만들기**를 선택합니다.
4. **수동으로 흐름 트리거**를 확장한 다음 입력 유형으로 **+입력>파일**을 추가를 선택합니다.
5. **파일 콘텐츠를 내 송장(제목이라고도 함)**으로 바꿉니다.
6. **+새 단계>AI Builder**를 선택한 다음, 작업 목록에서 **송장에서 정보 추출**을 선택합니다.
7. **송장 파일** 입력의 트리거에서 **내 송장**을 지정합니다.



8. 연속 작업으로 **모델 출력**의 송장 값을 사용할 수 있습니다.



축하합니다! AI Builder 송장 처리 모델을 사용하는 흐름을 만들었습니다. 오른쪽 위에서 저장을 선택한 다음, 테스트를 선택하여 흐름을 시험해 보십시오.

페이지 범위

대형 문서의 경우 처리할 페이지 범위를 지정할 수 있습니다.

Manually trigger a flow

Extract information from invoices

* Invoice file: My invoice

Pages: Enter page range (Ex: 1 or 3-5). Only returns results of a single invoice.

Hide advanced options ^

페이지 매개 변수에서 페이지 값이나 페이지 범위를 입력할 수 있습니다. 예: 1 또는 3-5.

① 참고

송장이 하나만 포함된 큰 문서가 있는 경우 **페이지 매개변수를 사용하여 송장을 목표로 하여 모델 예측 비용을 줄이고 성능을 향상하는** 것이 좋습니다. 그러나 페이지 범위에는 올바른 데이터를 반환하는 작업에 대한 고유 송장이 포함되어야 합니다.

예: 문서의 2페이지에 첫 번째 송장이 있고 3페이지와 4페이지에 걸쳐서 두 번째 송장이 있습니다.

- 페이지 범위 2를 입력하면 첫 번째 송장의 데이터를 반환합니다.
- 페이지 범위 3-4를 입력하면 두 번째 송장의 데이터만 반환합니다.
- 페이지 범위 2-4를 입력하면 첫 번째 및 두 번째 송장의 일부 데이터가 반환됩니다(피해야 함).

매개 변수

입력

[+] 테이블 확장

이름	필요함	종류	설명
영수증 파일	네	파일	처리할 송장 파일
페이지	아니요	문자열	처리할 페이지 범위

출력

[[테이블 확장]]

이름	종류	정의
미결제 금액(텍스트)	문자열	송장에 기재된 미결제 금액
미결제 금액(숫자)	부동	표준 숫자 형식의 미결제 금액. 예: 1234.98
미결제 금액에 대한 신뢰도	부동	모델의 예측 신뢰도 수준. 0(낮은 신뢰도) 및 1(높은 신뢰도) 사이의 점수.
청구 주소	문자열	청구 주소
청구 주소에 대한 신뢰도	부동	모델의 예측 신뢰도 수준. 0(낮은 신뢰도) 및 1(높은 신뢰도) 사이의 점수.
청구 주소의 받는 사람	문자열	청구 주소의 받는 사람
청구 주소의 받는 사람에 대한 신뢰도	부동	모델의 예측 신뢰도 수준. 0(낮은 신뢰도) 및 1(높은 신뢰도) 사이의 점수.
고객 주소	문자열	고객 주소
고객 주소에 대한 신뢰도	부동	모델의 예측 신뢰도 수준. 0(낮은 신뢰도) 및 1(높은 신뢰도) 사이의 점수.
고객 주소의 받는 사람	문자열	고객 주소의 받는 사람
고객 주소의 받는 사람에 대한 신뢰도	부동	모델의 예측 신뢰도 수준. 0(낮은 신뢰도) 및 1(높은 신뢰도) 사이의 점수.
고객 ID	문자열	고객 ID
고객 ID에 대한 신뢰도	부동	모델의 예측 신뢰도 수준. 0(낮은 신뢰도) 및 1(높은 신뢰도) 사이의 점수.
고객 이름	문자열	고객 이름
고객 이름에 대한 신뢰도	부동	모델의 예측 신뢰도 수준. 0(낮은 신뢰도) 및 1(높은 신뢰도) 사이의 점수.
기한(텍스트)	문자열	송장에 기재된 기한
기한(날짜)		표준 날짜 형식의 기한. 예: 2019-05-31T00:00:00Z
기한에 대한 신뢰도	부동	모델의 예측 신뢰도 수준. 0(낮은 신뢰도) 및 1(높은 신뢰도) 사이의 점수.
송장 날짜(텍스트)	문자열	송장에 기재된 송장 날짜

이름	종류	정의
송장 날짜(날짜)	날짜	표준 날짜 형식의 송장 날짜. 예: 2019-05-31T00:00:00Z
송장 날짜에 대한 신뢰도	부동	모델의 예측 신뢰도 수준. 0(낮은 신뢰도) 및 1(높은 신뢰도) 사이의 점수.
송장 ID	문자열	송장 ID
송장 ID에 대한 신뢰도	부동	모델의 예측 신뢰도 수준. 0(낮은 신뢰도) 및 1(높은 신뢰도) 사이의 점수.
송장 합계(텍스트)	문자열	송장에 기재된 송장 합계
송장 합계(숫자)	부동	표준 날짜 형식의 송장 합계. 예: 2019-05-31T00:00:00Z
송장 합계에 대한 신뢰도	부동	모델의 예측 신뢰도 수준. 0(낮은 신뢰도) 및 1(높은 신뢰도) 사이의 점수.
구매 주문	문자열	구매 주문
구매 주문에 대한 신뢰도	부동	모델의 예측 신뢰도 수준. 0(낮은 신뢰도) 및 1(높은 신뢰도) 사이의 점수.
송금 주소	문자열	송금 주소
송금 주소에 대한 신뢰도	부동	모델의 예측 신뢰도 수준. 0(낮은 신뢰도) 및 1(높은 신뢰도) 사이의 점수.
송금 주소의 받는 사람	문자열	송금 주소의 받는 사람
송금 주소의 받는 사람에 대한 신뢰도	부동	모델의 예측 신뢰도 수준. 0(낮은 신뢰도) 및 1(높은 신뢰도) 사이의 점수.
서비스 주소	문자열	서비스 주소
서비스 주소에 대한 신뢰도	부동	모델의 예측 신뢰도 수준. 0(낮은 신뢰도) 및 1(높은 신뢰도) 사이의 점수.
서비스 주소의 받는 사람	문자열	서비스 주소의 받는 사람
서비스 주소의 받는 사람에 대한 신뢰도	부동	모델의 예측 신뢰도 수준. 0(낮은 신뢰도) 및 1(높은 신뢰도) 사이의 점수.
배송지 주소	문자열	배송지 주소
배송지 주소에 대한 신뢰도	부동	모델의 예측 신뢰도 수준. 0(낮은 신뢰도) 및 1(높은 신뢰도) 사이의 점수.
배송지 주소의 받는 사람	문자열	배송지 주소의 받는 사람
배송지 주소의 받는 사람에 대한 신뢰도	부동	모델의 예측 신뢰도 수준. 0(낮은 신뢰도) 및 1(높은 신뢰도) 사이의 점수.

이름	종류	정의
소계(텍스트)	문자열	송장에 기재된 소계
소계(번호)	부동	표준 숫자 형식의 소계. 예: 1234.98
소계에 대한 신뢰도	부동	모델의 예측 신뢰도 수준. 0(낮은 신뢰도) 및 1(높은 신뢰도) 사이의 점수.
총 세금(텍스트)	문자열	송장에 기재된 총 세금
총 세금(숫자)	부동	표준 숫자 형식의 총 세금. 예: 1234.98
총 세금에 대한 신뢰도	부동	모델의 예측 신뢰도 수준. 0(낮은 신뢰도) 및 1(높은 신뢰도) 사이의 점수.
공급업체 주소	문자열	공급업체 주소
공급업체 주소에 대한 신뢰도	부동	모델의 예측 신뢰도 수준. 0(낮은 신뢰도) 및 1(높은 신뢰도) 사이의 점수.
공급업체 주소의 받는 사람	문자열	공급업체 주소의 받는 사람
공급업체 주소의 받는 사람에 대한 신뢰도	부동	모델의 예측 신뢰도 수준. 0(낮은 신뢰도) 및 1(높은 신뢰도) 사이의 점수.
공급업체 이름	문자열	공급업체 이름
공급업체 이름에 대한 신뢰도	부동	모델의 예측 신뢰도 수준. 0(낮은 신뢰도) 및 1(높은 신뢰도) 사이의 점수.
검색한 텍스트	문자열	송장에서 OCR을 실행하여 인식된 텍스트 줄. 텍스트 목록의 일부로 반환됩니다.
감지한 텍스트의 페이지 번호	integer	인식된 텍스트 줄이 있는 페이지. 텍스트 목록의 일부로 반환됩니다.

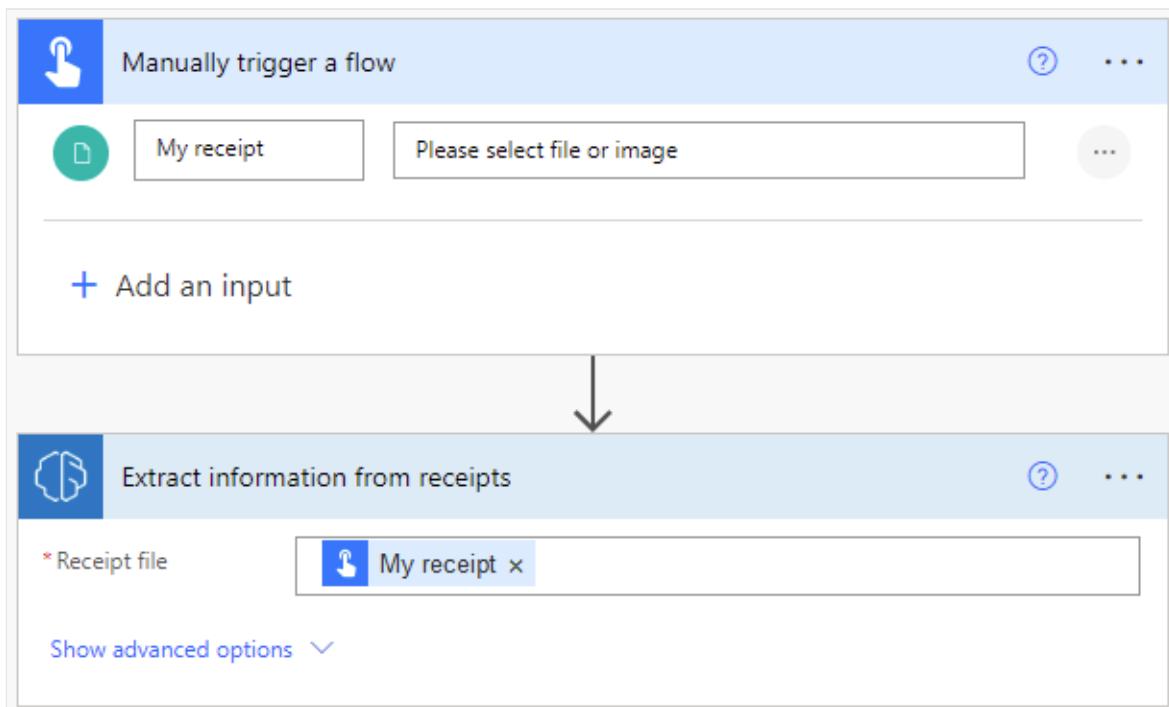
관련 항목

[송장 처리 개요](#)

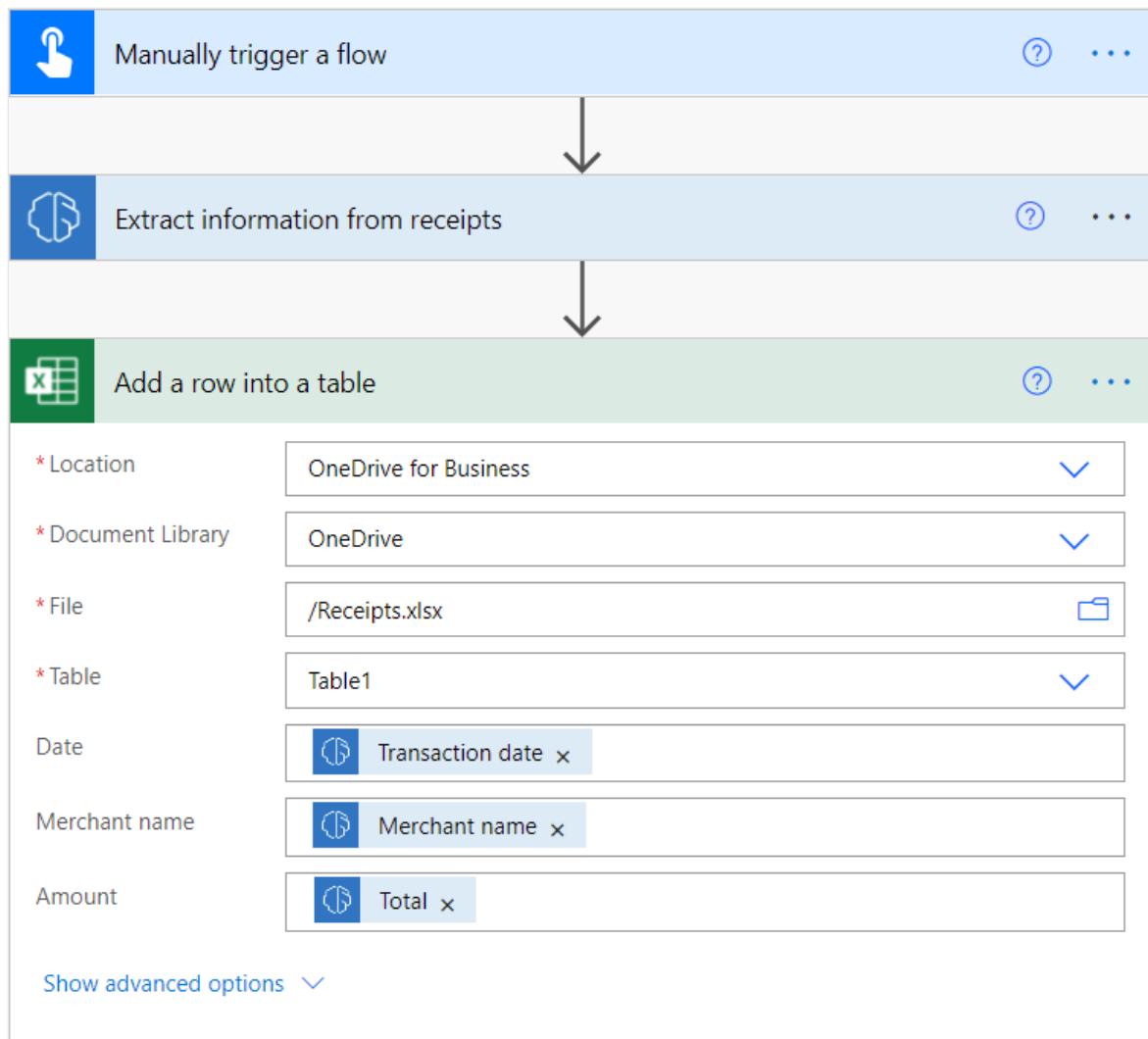
Power Automate에서 영수증 처리 미리 빌드된 모델 사용

아티클 • 2023. 03. 16.

1. Power Automate [에](#) 로그인합니다.
2. 왼쪽 창에서 **내 흐름**을 선택한 다음 **새 흐름**>**인스턴트 클라우드 흐름**을 선택합니다.
3. 흐름에 이름을 지정한 다음 **이 흐름을 트리거하는 방법** 아래에서 **수동으로 흐름 트리거**를 선택하고 **만들기**를 선택합니다.
4. **수동으로 흐름 트리거**를 확장한 다음 입력 유형으로 **+입력>파일**을 추가를 선택합니다.
5. **파일 콘텐츠를 내 영수증(제목이라고도 함)**으로 바꿉니다.
6. **+새 단계>AI Builder**를 선택한 다음, 작업 목록에서 **영수증에서 정보 추출**을 선택합니다.
7. **영수증 파일** 입력을 선택한 다음 **동적 콘텐츠 목록에서 내 영수증**을 선택합니다.



8. 연속 작업에서 아래 **모델 출력** 섹션의 영수증 값을 사용할 수 있습니다.



① 참고

영수증 값이 문자열로 반환됩니다. 숫자로 조작하려면 **float** 또는 **int** 변환 기능을 사용할 수 있습니다.

축하합니다! AI Builder 영수증 처리 모델을 사용하는 흐름을 만들었습니다. 오른쪽 위에서 **저장**을 선택한 다음, **테스트**를 선택하여 흐름을 시험해 보십시오.

페이지 범위

대형 문서의 경우 처리할 페이지 범위를 지정할 수 있습니다.

페이지 매개 변수에서 페이지 값이나 페이지 범위를 입력할 수 있습니다. 예: 1 또는 3-5.

① 참고

영수증이 하나만 포함된 큰 문서가 있는 경우 **페이지 매개변수를 사용하여 영수증을 목표로 하여 모델 예측 비용을 줄이고 성능을 향상하는** 것이 좋습니다. 다만, 페이지 범위 내에서 **첫 번째 영수증의 데이터만 반환되며** 다중 페이지 영수증은 지원되지 않습니다.

예: 문서의 2페이지에 첫 번째 영수증이 있고 3페이지와 4페이지에 걸쳐서 두 번째 영수증이 있습니다.

- 값 2를 입력하면 첫 번째 영수증의 데이터를 반환합니다.
- 값 3-4를 입력하면 두 번째 영수증의 첫 페이지의 데이터만 반환됩니다.
- 값 2-4를 입력하면 두 번째 영수증의 데이터가 아니라 첫 번째 영수증의 데이터만 반환됩니다.

매개 변수

입력

[+] 테이블 확장

이름	필요함	종류	설명
영수증 파일	네	문자열	처리할 영수증 파일
페이지	아니요	문자열	처리할 페이지 범위

출력

 테이블 확장

이름	종류	설명
가맹점 이름	문자열	가맹점 이름
가맹점 주소	string	가맹점 주소
가맹점 전화 번호	string	가맹점 전화 번호
거래일	string	거래일
거래 시간	string	거래 시간
구매 품목 이름	string	구매 품목 이름. 항목 목록의 일부로 반환됩니다.
구매 품목 수량	string	구매 품목 수량. 항목 목록의 일부로 반환됩니다.
구매 품목 가격	string	구매 품목 가격. 항목 목록의 일부로 반환됩니다.
구매 품목 총 가격	string	구매 품목 총 가격. 항목 목록의 일부로 반환됩니다.
소계	string	소계
세금	string	세금
팁	string	팁
합계	string	합계
가맹점 이름에 대한 신뢰도	부동	모델의 검색 신뢰도
가맹점 주소에 대한 신뢰도	부동	모델의 검색 신뢰도
가맹점 전화 번호에 대한 신뢰도	부동	모델의 검색 신뢰도
거래일에 대한 신뢰도	부동	모델의 검색 신뢰도
거래 시간에 대한 신뢰도	부동	모델의 검색 신뢰도
구매 품목 이름에 대한 신뢰도	부동	모델의 검색 신뢰도. 항목 목록의 일부로 반환됩니다.
구매 품목 수량에 대한 신뢰도	부동	모델의 검색 신뢰도. 항목 목록의 일부로 반환됩니다.
구매 품목 가격에 대한 신뢰도	부동	모델의 검색 신뢰도. 항목 목록의 일부로 반환됩니다.

이름	종류	설명
구매 품목 총 가격에 대한 신뢰도	부동	모델의 검색 신뢰도. 항목 목록의 일부로 반환됩니다.
소계에 대한 신뢰도	부동	모델의 검색 신뢰도
세금에 대한 신뢰도	부동	모델의 검색 신뢰도
팁에 대한 신뢰도	부동	모델의 검색 신뢰도
총계에 대한 신뢰도	부동	모델의 검색 신뢰도
감지한 텍스트	string	인식된 텍스트 줄. 텍스트 목록의 일부로 반환됩니다.
감지한 텍스트의 페이지 번호	integer	인식된 텍스트 줄이 있는 페이지. 텍스트 목록의 일부로 반환됩니다.
감지한 텍스트의 높이	부동	텍스트 줄의 높이. 텍스트 목록의 일부로 반환됩니다.
감지한 텍스트의 왼쪽 위치	부동	텍스트 줄의 왼쪽 위치. 텍스트 목록의 일부로 반환됩니다.
감지한 텍스트의 상단 위치	부동	텍스트 줄의 위쪽 위치. 텍스트 목록의 일부로 반환됩니다.
감지한 텍스트의 너비	부동	텍스트 줄의 너비. 텍스트 목록의 일부로 반환됩니다.

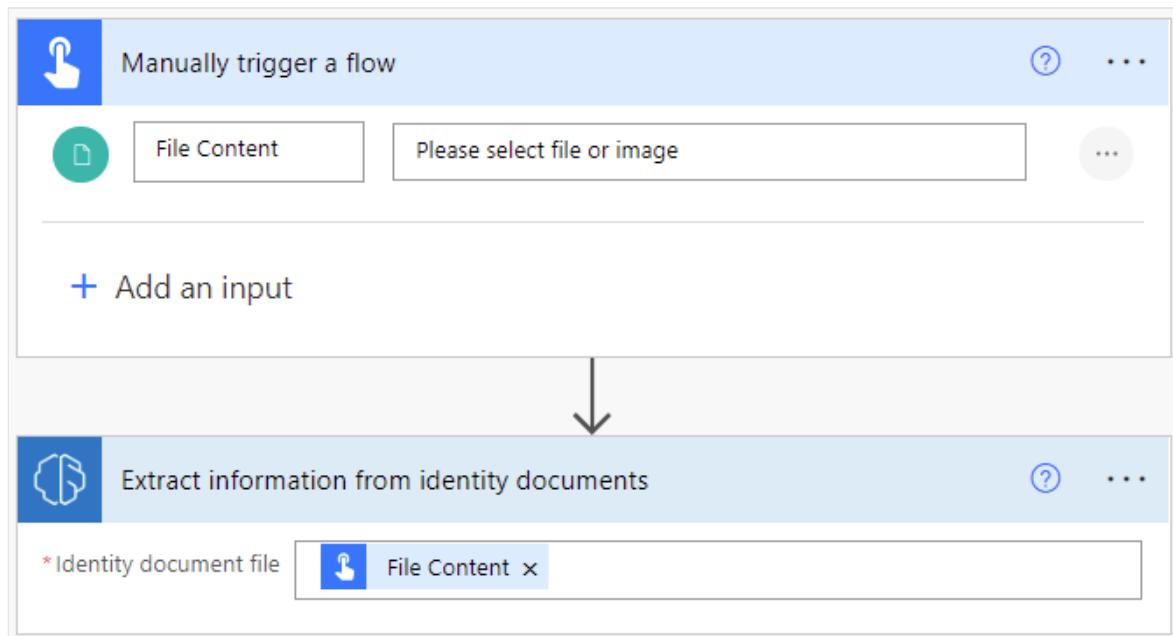
참조 항목

- 영수증 처리 개요
- 학습: AI Builder로 영수증 처리(모듈)

Power Automate의 ID 판독기 미리 빌드 된 모델

아티클 • 2024. 01. 10.

1. Power Automate [에](#) 로그인합니다.
2. 상단 메뉴에서 **내 흐름**을 선택한 다음 **새 흐름**>**인스턴트 클라우드 흐름**을 선택합니다.
3. 흐름에 이름을 지정한 다음 **이 흐름을 트리거하는 방법** 아래에서 **수동으로 흐름 트리거**를 선택하고 **만들기**를 선택합니다.
4. **수동으로 흐름 트리거**를 확장한 다음 입력 유형으로 **+입력>파일**을 추가를 선택합니다.
5. **+새 단계>AI Builder>ID 문서에서 정보 추출**을 선택합니다.
6. 흐름에서 처리할 ID 문서 파일로 **파일 내용**을 지정합니다.



축하합니다! ID 판독기 모델을 사용하는 흐름을 만들었습니다. 저장을 선택하고 오른쪽 상단의 테스트를 선택하여 흐름을 사용해 봅니다.

추출된 정보를 Excel 워크시트에 추가하는 예제 흐름

다음 예에서는 Excel 워크시트에 추출된 정보를 입력하는 단계를 흐름에 추가합니다. 먼저 흐름에서 사용할 테이블을 준비합니다. 테이블은 추출하려는 정보와 일치해야 합니다.

그런 다음 흐름에 Excel 커넥터를 추가합니다.

Excel 테이블 만들기

1. Microsoft OneDrive 또는 SharePoint 폴더에 Excel 통합 문서를 만듭니다.
2. 워크시트의 첫 번째 행에 **이름**, **성**, **ID 문서 번호** 및 **국가** 값을 열에 하나씩 입력합니다. 이 값은 테이블의 열 헤더입니다.
3. 셀을 선택하고 첫 번째 행을 머리글로 사용하여 표로 서식을 지정합니다.

1	First name ▾	Last name ▾	Identity document number ▾	Country ▾
2				
3				

4. 통합 문서를 저장하고 닫습니다.

추출된 데이터를 테이블에 입력

1. 생성한 ID 판독기 흐름을 사용하거나 이 예에 대해 다른 흐름을 만듭니다.
2. +새 단계>Excel Online(Business)>**테이블에 행 추가**를 선택합니다.
3. 위치, 문서 라이브러리 및 파일을 선택하여 Excel 통합 문서를 찾을 위치를 지정합니다.
4. 이전 단계에서 생성한 **테이블**을 선택합니다.
5. **이름**, **성** 및 **ID 문서 번호**에서 동적 콘텐츠 목록에서 일치하는 값을 선택합니다.
6. **국가**의 동적 콘텐츠 목록에서 **국가/지역**을 선택합니다.

Add a row into a table

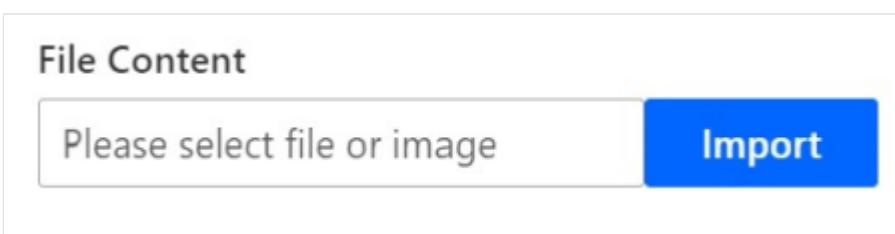
* Location	OneDrive for Business	
* Document Library	OneDrive	
* File	/ID Documents/ID Documents.xlsx	
* Table	Table1	
First name	First name	
Last name	Last name	
Identity document number	Identity docum...	
Country	Country/Region	

Show advanced options

7. 저장을 선택합니다.

흐름 테스트

- 테스트를 선택하고 수동을 선택한 다음 테스트를 선택하여 작업을 트리거합니다.
- 파일 콘텐츠에서 신분증 파일 또는 이미지를 선택한 후 가져오기를 선택합니다.



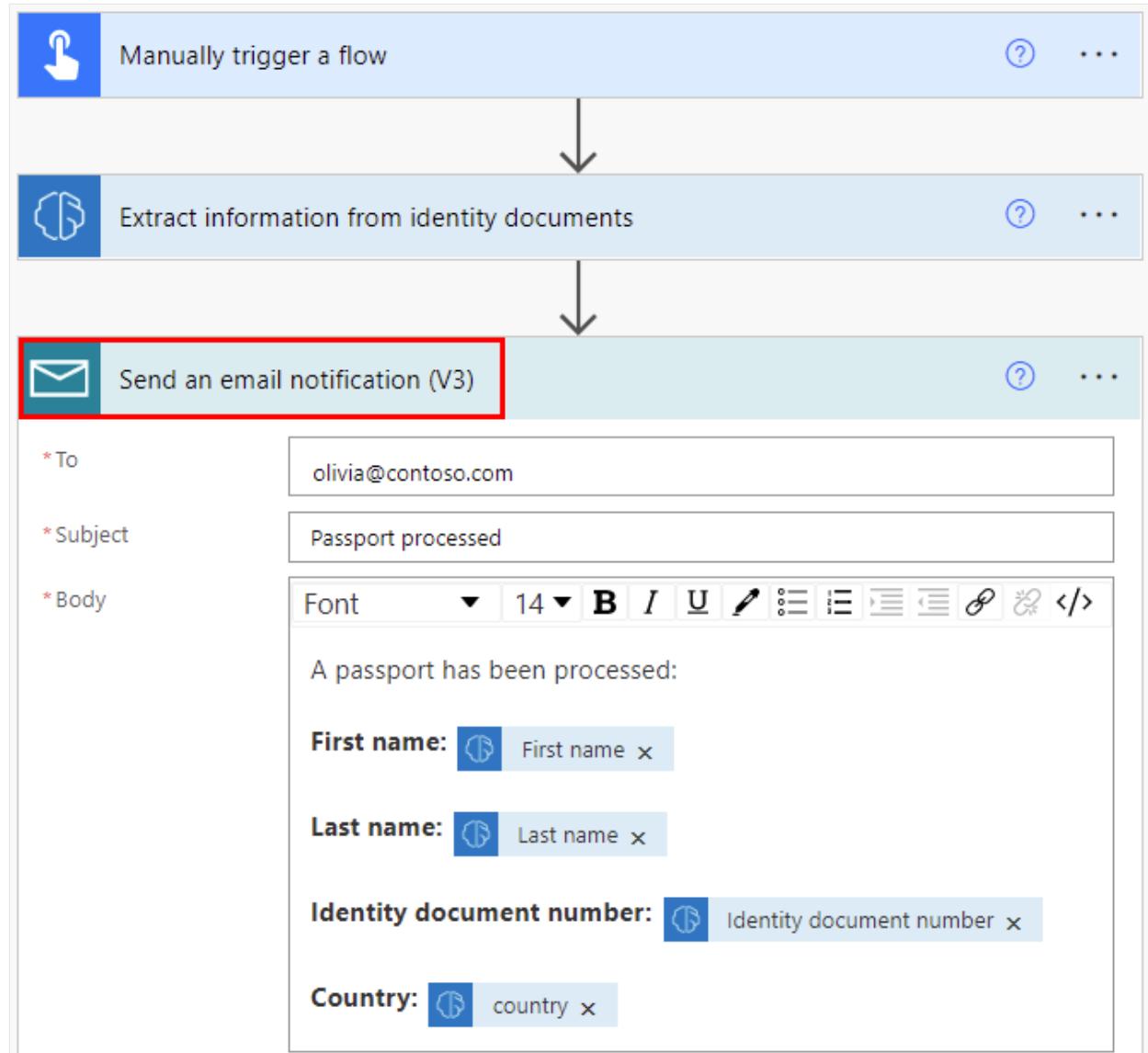
- 흐름 실행을 선택합니다.

AI Builder가 데이터를 추출하고 Excel의 테이블에 새 항목을 추가하는 동안 흐름을 실행하는 데 몇 초가 걸릴 수 있습니다. Excel 통합 문서를 열어 추출된 정보가 입력되었는지 확인합니다.

	A	B	C	D
1	First name	Last name	Identity document number	Country
2	JENNIFER	BROOKS		340020013 USA
3				

추출된 정보를 이메일로 보내는 흐름 예시

다음 예는 추출된 정보를 이메일로 보내도록 흐름을 설정하는 방법을 보여줍니다. **이메일 알림 보내기** 커넥터를 이전에 생성한 흐름에 추가하거나 이 예에 대한 ID 판독기 흐름을 생성할 수 있습니다.



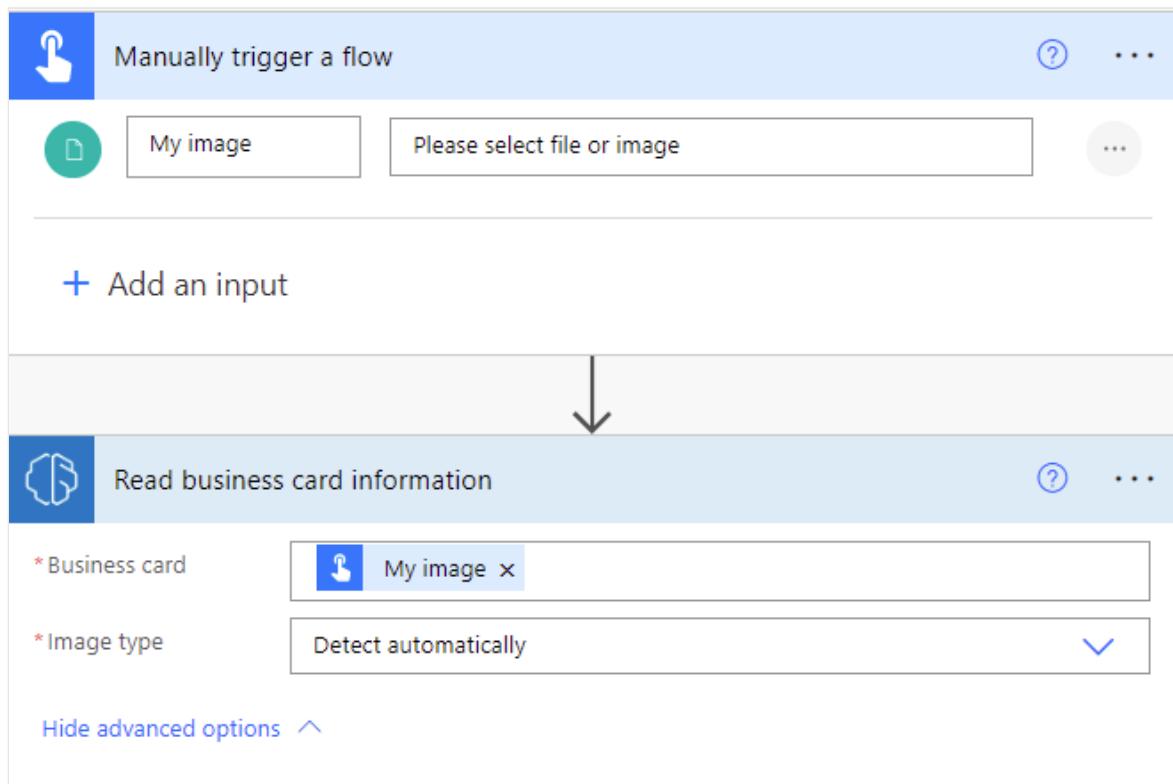
참조 항목

[ID 판독기 미리 빌드된 모델 개요](#)

Power Automate에서 비즈니스 카드 판독기 미리 빌드된 모델 사용

아티클 • 2024. 01. 10.

1. Power Automate [에](#) 로그인합니다.
2. 왼쪽 창에서 **내 흐름**을 선택한 다음 **새 흐름**>**인스턴트 클라우드 흐름**을 선택합니다.
3. 흐름에 이름을 지정한 다음 **이 흐름을 트리거하는 방법** 아래에서 **수동으로 흐름 트리거**를 선택하고 **만들기**를 선택합니다.
4. **수동으로 흐름 트리거**를 확장한 다음 입력 유형으로 **+입력**>**파일**을 추가를 선택합니다.
5. **파일 내용을 내 이미지(제목이라고도 함)**로 바꿉니다.
6. **+새 단계**>**AI Builder**를 선택한 다음 작업 목록에서 **비즈니스 카드 정보 판독**을 선택합니다.
7. 흐름에 대한 **명함** 입력의 트리거에서 **내 이미지를** 지정합니다.
8. **고급 옵션 표시**를 선택하고 **자동 검색이 이미지 유형** 입력에 있는지 확인합니다.

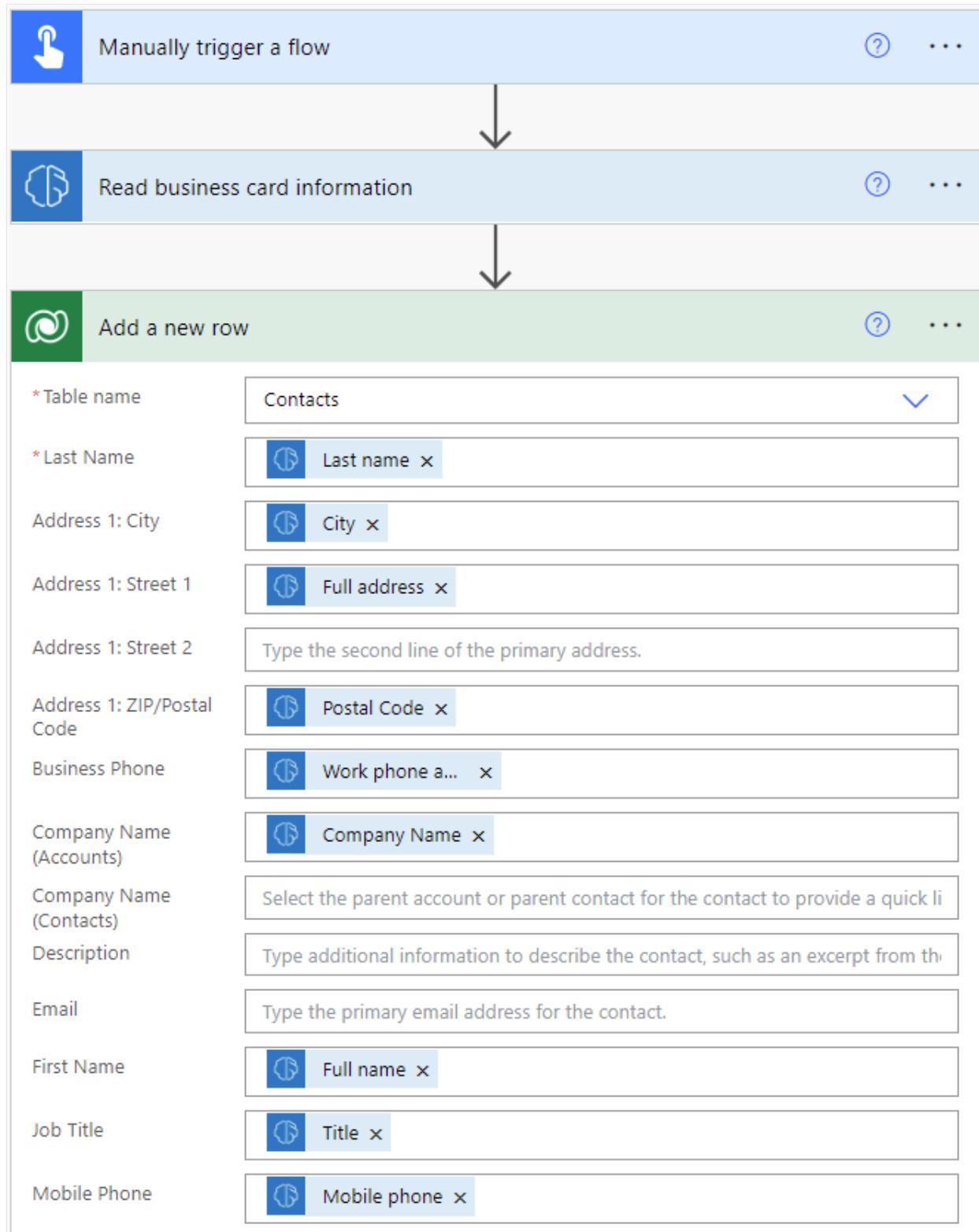


축하합니다! 비즈니스 카드 판독기 AI 모델을 사용하는 흐름을 만들었습니다. 저장을 선택하고 오른쪽 상단의 테스트를 선택하여 흐름을 사용해 봅니다.

명함 판독기 흐름의 예

다음 예제에서는 비즈니스 카드 데이터를 사용하여 Microsoft Dataverse에서 생성되는 새 연락처를 보여 줍니다.

새 행 추가 단계를 추가하려면 + 새 단계>Microsoft Dataverse>새 행 추가를 선택합니다.



매개 변수

입력

[+] 테이블 확장

입력	필수	종류	설명	값
이미지 유형	있음	문자열	시간의 MEME 유형	"자동"은 기본값입니다. 이 열은 사용되지 않으며 모든 값이 허용됩니다.
Image	있음	파일	분석할 이미지 파일	

출력

[+] 테이블 확장

이름	형식	설명
구/군/시	string	도시 주소
국가	string	국가 주소
우편 번호	string	우편 번호 주소
사서함	string	주소의 사서함
State(상태)	string	주소의 시/도
나머지 주소	string	구/군/시 주소
직장 전화 또는 기타 전화	string	첫 번째 전화번호 또는 팩스 번호
회사 이름	string	회사 이름
부서	string	비즈니스 카드에서 찾은 조직 부서
Email	string	비즈니스 카드에서 찾은 연락처 전자 메일(있는 경우)
팩스	string	세 번째 전화번호 또는 팩스 번호
이름	string	연락처 이름
전체 주소	string	연락처 전체 주소
전체 이름	string	연락처 전체 이름

이름	형식	설명
제목	string	연락처 직위
성	string	연락처 성
휴대폰	string	두 번째 전화번호 또는 팩스 번호
Website	string	웹 사이트

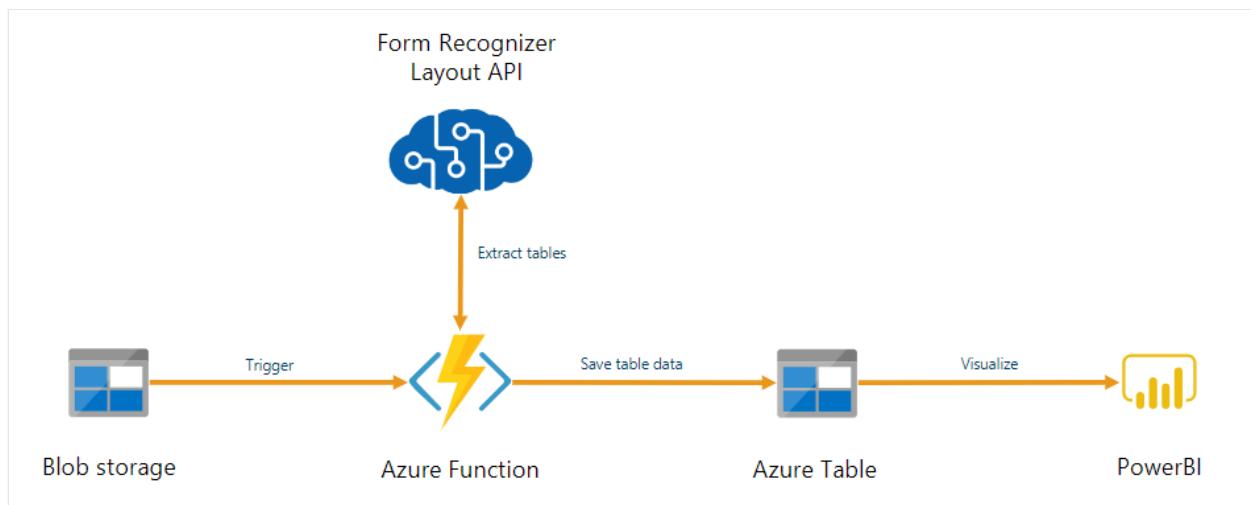
참조 항목

[명함 판독기 개요](#)

자습서: Azure Functions 및 Python을 사용하여 저장된 문서 처리

아티클 • 2023. 10. 11.

문서 인텔리전스는 Azure Functions를 사용하여 빌드된 자동화된 데이터 처리 파이프라인의 일부로 사용할 수 있습니다. 이 가이드에서는 Azure Functions를 사용하여 Azure Blob 스토리지 컨테이너에 업로드된 문서를 처리하는 방법을 보여 줍니다. 이 워크플로는 문서 인텔리전스 레이아웃 모델을 사용하여 저장된 문서에서 테이블 데이터를 추출하고 테이블 데이터를 Azure의 .csv 파일에 저장합니다. 그런 다음, Microsoft Power BI를 사용하여 데이터를 표시할 수 있습니다(여기서는 설명하지 않음).



이 자습서에서는 다음 작업 방법을 알아봅니다.

- ✓ Azure Storage 계정 만들기
- ✓ Azure Functions 프로젝트를 만듭니다.
- ✓ 업로드된 양식에서 레이아웃 데이터를 추출합니다.
- ✓ 추출된 레이아웃 데이터를 Azure Storage에 업로드합니다.

필수 조건

- **Azure 구독** - [체험 구독 만들기](#)
- **문서 인텔리전스 리소스**. Azure 구독이 있으면 Azure Portal에서 [문서 인텔리전스 리소스](#)를 만들어 키와 앤드포인트를 가져옵니다. 평가판 가격 책정 계층(F0)을 통해 서비스를 사용해보고, 나중에 프로덕션용 유료 계층으로 업그레이드할 수 있습니다.
 - 리소스를 배포한 후 **리소스로 이동**을 선택합니다. 애플리케이션을 문서 인텔리전스 API에 연결하려면 만든 리소스의 키와 앤드포인트가 필요합니다. 이 자습서의 뒷부분에서 키와 앤드포인트를 아래 코드에 붙여넣습니다.

- Python 3.6.x, 3.7.x, 3.8.x 또는 3.9.x [또는 3.9.x](#) (Python 3.10.x는 이 프로젝트에서 지원되지 않음)
- 다음 확장이 설치된 최신 버전의 [VS Code](#) (Visual Studio Code)
 - [Azure Functions 확장](#). 설치되면 왼쪽 탐색 창에 Azure 로고가 표시됩니다.
 - [Azure Functions Core Tools](#) 버전 3.x(버전 4.x는 이 프로젝트에서 지원되지 않음)
 - [Python 확장](#) (Visual Studio Code용). 자세한 내용은 VS Code에서 Python 시작을 참조[하세요.](#)
- [Azure Storage Explorer](#) 설치
- 분석할 로컬 PDF 문서. [샘플 PDF 문서](#)를 이 프로젝트에 사용할 수 있습니다.

Azure Storage 계정 만들기

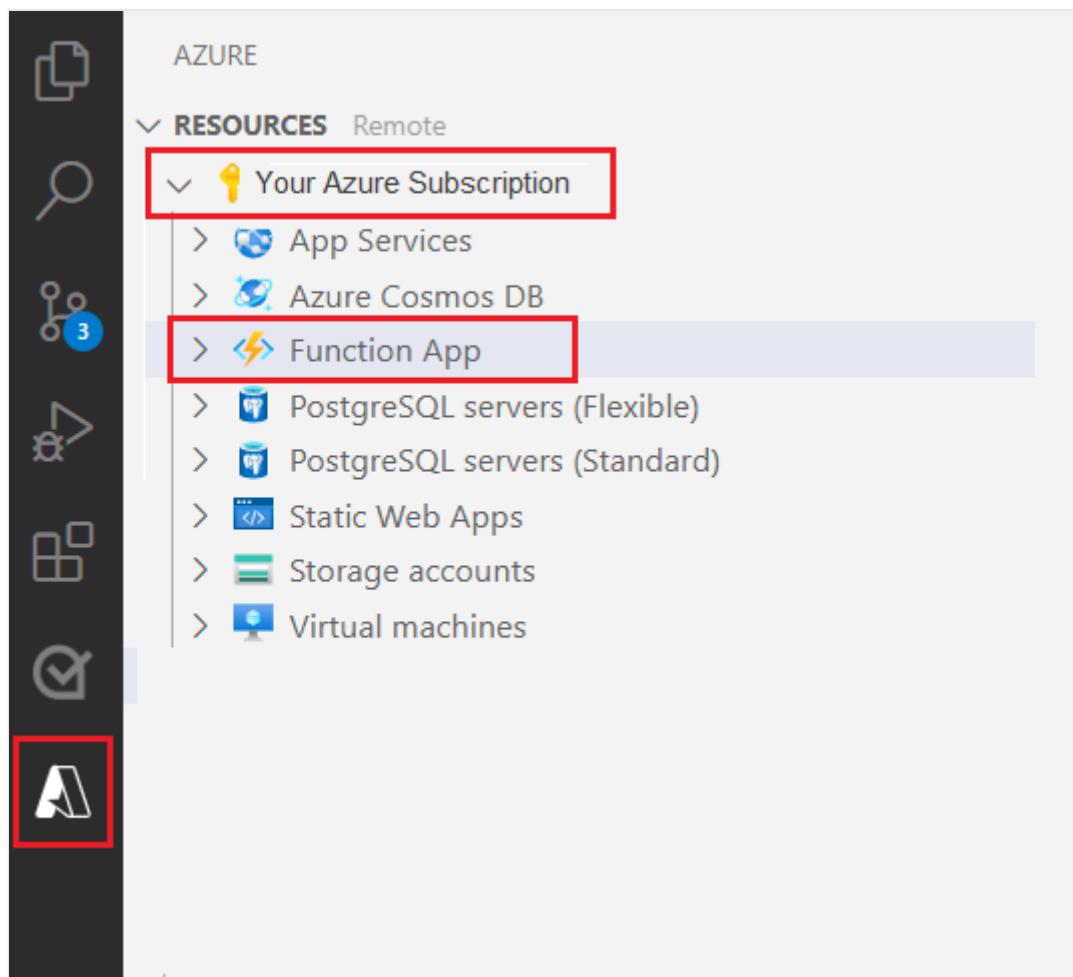
1. Azure Portal에서 [범용 v2 Azure Storage 계정을 만듭니다](#). 스토리지 컨테이너를 사용하여 Azure 스토리지 계정을 만드는 방법을 모르는 경우 다음 빠른 시작을 따릅니다.

- [스토리지 계정 만들기](#) 스토리지 계정을 만들 때 인스턴스 세부 사항>성능 필드에서 표준 성능을 선택합니다.
- [컨테이너를 만듭니다](#). 컨테이너를 만드는 경우 새 컨테이너 창에서 퍼블릭 액세스 수준을 컨테이너(컨테이너 및 파일에 대한 익명 읽기 액세스)로 설정합니다.

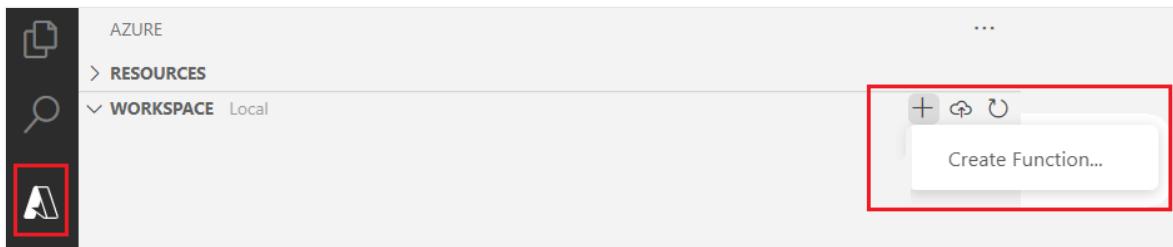
2. 왼쪽 창에서 **리소스 공유(CORS)** 탭을 선택하고, 기존 CORS 정책을 제거합니다(있는 경우).
3. 배포되면 **test** 및 **output**이라는 두 개의 빈 Blob 스토리지 컨테이너를 만듭니다.

Azure Functions 프로젝트 만들기

1. 프로젝트를 포함할 **functions-app**이라는 새 폴더를 만들고, **선택**을 선택합니다.
2. Visual Studio Code를 열고, 명령 팔레트를 엽니다(Ctrl+Shift+P). **Python: 인터프리터 선택**을 검색하여 선택하고, 버전 3.6.x, 3.7.x, 3.8.x 또는 3.9.x인 설치된 Python 인터프리터를 선택합니다. 이 선택 항목은 선택한 Python 인터프리터 경로를 프로젝트에 추가합니다.
3. 왼쪽 탐색 창에서 Azure 로고를 선택합니다.
 - [리소스] 보기에서 기존 Azure 리소스가 표시됩니다.
 - 이 프로젝트에 사용하는 Azure 구독을 선택합니다. 그러면 그 아래에 [Azure Function 앱]이 표시됩니다.



4. 나열된 리소스 아래에 있는 [작업 영역(로컬)] 섹션을 선택합니다. 더하기 기호, 함수 만들기 단추를 차례로 선택합니다.



5. 메시지가 표시되면 새 프로젝트 만들기를 선택하고, function-app 디렉터리로 이동 합니다. 선택을 선택합니다.

6. 다음과 같은 몇 가지 여러 설정을 구성하라는 메시지가 표시됩니다.

- 언어 선택 → Python을 선택합니다.
- 가상 환경을 만들 Python 인터프리터 선택 → 이전에 기본값으로 설정한 인터프리터를 선택합니다.
- 템플릿 선택 → Azure Blob Storage 트리거를 선택하고, 트리거 이름을 지정하거나 기본 이름을 적용합니다. Enter 키를 눌러 확인합니다.
- 설정 선택 → 드롭다운 메뉴에서 + 새 로컬 앱 설정 만들기를 선택합니다.
- 구독 선택 → 만든 스토리지 계정이 있는 Azure 구독을 선택하고, 스토리지 계정을 선택하고, 스토리지 입력 컨테이너의 이름(이 경우 `input/{name}`)을 선택합니다. Enter 키를 눌러 확인합니다.
- 프로젝트를 여는 방법 선택 → 드롭다운 창에서 현재 창에서 프로젝트 열기를 선택합니다.

7. 이러한 단계가 완료되면 VS Code에서 `_init_.py` Python 스크립트를 사용하여 새 Azure Function 프로젝트를 추가합니다. 이 스크립트는 파일을 `input` 스토리지 컨테이너에 업로드할 때 트리거됩니다.

```
Python

import logging

import azure.functions as func


def main(myblob: func.InputStream):
    logging.info(f"Python blob trigger function processed blob \n"
                f"Name: {myblob.name}\n"
                f"Blob Size: {myblob.length} bytes")
```

함수 테스트

1. F5 키를 눌러 기본 함수를 실행합니다. VS Code에서 인터페이스할 스토리지 계정을 선택하라는 메시지를 표시합니다.
 2. 만든 스토리지 계정을 선택하고 계속합니다.
 3. Azure Storage Explorer를 열고, 샘플 PDF 문서를 `input` 컨테이너에 업로드합니다.
그런 다음, VS Code 터미널을 확인합니다. 스크립트에서 PDF 업로드를 통해 트리거 되었다고 기록해야 합니다.

4. 계속하기 전에 스크립트를 중지합니다.

문서 처리 코드 추가

다음으로, 문서 인텔리전스 서비스를 호출하고 문서 인텔리전스 [레이아웃 모델](#)을 사용하여 업로드된 문서를 구문 분석하는 자체 코드를 Python 스크립트에 추가합니다.

1. VS Code에서 함수의 *requirements.txt* 파일로 이동합니다. 이 파일은 스크립트에 대한 종속성을 정의합니다. 다음 Python 패키지를 파일에 추가합니다.

txt
cryptography
azure-functions
azure-storage-blob
azure-identity
requests
pandas
numpy

2. 그런 다음, `_init_.py` 스크립트를 엽니다. 다음 `import` 문을 추가합니다.

```
Python

import logging
from azure.storage.blob import BlobServiceClient
import azure.functions as func
import json
import time
from requests import get, post
import os
import requests
from collections import OrderedDict
import numpy as np
import pandas as pd
```

3. 생성된 `main` 함수를 있는 그대로 둘 수 있습니다. 사용자 지정 코드를 이 함수 내에 추가합니다.

```
Python

# This part is automatically generated
def main(myblob: func.InputStream):
    logging.info(f"Python blob trigger function processed blob \n"
    f"Name: {myblob.name}\n"
    f"Blob Size: {myblob.length} bytes")
```

4. 다음 코드 블록에서는 업로드된 문서에서 문서 인텔리전스 [레이아웃 분석](#) API를 호출합니다. 엔드포인트 및 키 값을 입력합니다.

```
Python

# This is the call to the Document Intelligence endpoint
endpoint = "Your Document Intelligence Endpoint"
apim_key = "Your Document Intelligence Key"
post_url = endpoint + "/formrecognizer/v2.1/layout/analyze"
source = myblob.read()

headers = {
    # Request headers
    'Content-Type': 'application/pdf',
    'Ocp-Apim-Subscription-Key': apim_key,
}

text1=os.path.basename(myblob.name)
```

 중요

완료되면 코드에서 키를 제거하고 공개적으로 게시하지 마세요. 프로덕션의 경우 [Azure Key Vault](#)와 같은 자격 증명을 안전하게 저장하고 액세스하는 방법을 사용합니다. 자세한 내용은 Azure AI 서비스 [보안](#)을 참조하세요.

5. 다음으로, 서비스를 쿼리하고 반환된 데이터를 가져오는 코드를 추가합니다.

Python

```
resp = requests.post(url=post_url, data=source, headers=headers)

if resp.status_code != 202:
    print("POST analyze failed:\n%s" % resp.text)
    quit()
print("POST analyze succeeded:\n%s" % resp.headers)
get_url = resp.headers["operation-location"]

wait_sec = 25

time.sleep(wait_sec)
# The layout API is async therefore the wait statement

resp = requests.get(url=get_url, headers={"Ocp-Apim-Subscription-Key": apim_key})

resp_json = json.loads(resp.text)

status = resp_json["status"]

if status == "succeeded":
    print("POST Layout Analysis succeeded:\n%s")  
    results = resp_json
else:
    print("GET Layout results failed:\n%s")
    quit()

results = resp_json
```

6. Azure Storage output 컨테이너에 연결하는 다음 코드를 추가합니다. 스토리지 계정 이름 및 키에 대한 사용자 고유의 값을 입력합니다. Azure Portal에 있는 스토리지 리소스의 [액세스 키](#) 탭에서 키를 가져올 수 있습니다.

Python

```
# This is the connection to the blob storage, with the Azure Python SDK
blob_service_client =
BlobServiceClient.from_connection_string("DefaultEndpointsProtocol=http  
s;AccountName="Storage Account Name";AccountKey="storage account
```

```
key";EndpointSuffix=core.windows.net")
container_client=blob_service_client.get_container_client("output")
```

다음 코드에서는 반환된 문서 인텔리전스 응답을 구문 분석하고, .csv 파일을 생성한 후 **output** 컨테이너에 업로드합니다.

ⓘ 중요

자체 문서 구조와 일치하도록 이 코드를 편집해야 할 수 있습니다.

Python

```
# The code below extracts the json format into tabular data.
# Please note that you need to adjust the code below to your form
structure.
# It probably won't work out-of-the-box for your specific form.
pages = results["analyzeResult"]["pageResults"]

def make_page(p):
    res=[]
    res_table=[]
    y=0
    page = pages[p]
    for tab in page["tables"]:
        for cell in tab["cells"]:
            res.append(cell)
            res_table.append(y)
        y=y+1

    res_table=pd.DataFrame(res_table)
    res=pd.DataFrame(res)
    res["table_num"]=res_table[0]
    h=res.drop(columns=["boundingBox","elements"])
    h.loc[:, "rownum"]=range(0,len(h))
    num_table=max(h["table_num"])
    return h, num_table, p

h, num_table, p= make_page(0)

for k in range(num_table+1):
    new_table=h[h.table_num==k]
    new_table.loc[:, "rownum"]=range(0,len(new_table))
    row_table=pages[p]["tables"][k]["rows"]
    col_table=pages[p]["tables"][k]["columns"]
    b=np.zeros((row_table,col_table))
    b=pd.DataFrame(b)
    s=0
    for i,j in zip(new_table["rowIndex"],new_table["columnIndex"]):
        b.loc[i,j]=new_table.loc[new_table.loc[s,"rownum"],"text"]
    new_table=new_table.drop(["rowIndex","columnIndex"],axis=1)
    new_table=new_table.reset_index(drop=True)
    h=h.append(new_table)
```

s=s+1

7. 마지막으로, 코드의 마지막 블록에서는 추출된 테이블과 텍스트 데이터를 Blob 스토리지 요소에 업로드합니다.

Python

```
# Here is the upload to the blob storage
tab1_csv=b.to_csv(header=False,index=False,mode='w')
name1=(os.path.splitext(text1)[0]) +'.csv'
container_client.upload_blob(name=name1,data=tab1_csv)
```

함수 실행

1. F5 키를 눌러 함수를 다시 실행합니다.
2. Azure Storage Explorer를 사용하여 샘플 PDF 양식을 **input** 스토리지 컨테이너에 업로드합니다. 이 작업은 실행할 스크립트를 트리거한 다음, **output** 컨테이너에 결과 .csv 파일(테이블로 표시됨)을 표시합니다.

이 컨테이너를 Power BI에 연결하여 포함된 데이터에 대한 풍부한 시각화를 만들 수 있습니다.

다음 단계

이 자습서에서는 Python으로 작성된 Azure 함수를 사용하여 업로드된 PDF 문서를 자동으로 처리하고 해당 콘텐츠를 보다 더 데이터 친화적인 형식으로 출력하는 방법을 알아보았습니다. 다음으로 Power BI를 사용하여 데이터를 표시하는 방법을 알아봅니다.

Microsoft Power BI

- 문서 인텔리전스란?
- 레이아웃 모델에 대해 자세히 알아보기

Document Models - Build Model

참조

Service: Azure AI Services

API Version: 2024-02-29-preview

사용자 지정 문서 분석 모델을 빌드합니다.

HTTP
<pre>POST {endpoint}/documentintelligence/documentModels:build?api-version=2024-02-29-preview</pre>

URI 매개 변수

[+] 테이블 확장

Name	In(다음 안에)	필수	형식	Description
endpoint	path	True	string	Document Intelligence 서비스 엔드포인트입니다.
api-version	query	True	string	이 작업에 사용할 API 버전입니다.

요청 본문

[+] 테이블 확장

Name	필수	형식	Description
buildMode	True	DocumentBuild Mode	사용자 지정 문서 모델 빌드 모드입니다.
modelId	True	string	고유한 문서 모델 이름입니다.
azureBlobFileListSource		AzureBlobFile ListContent Source	학습 데이터를 지정하는 파일 목록을 Azure Blob Storage. azureBlobSource 또는 azureBlobFileListSource를 지정해야 합니다.
azureBlobSource		AzureBlob ContentSource	학습 데이터를 포함하는 Azure Blob Storage 위치입니다. azureBlobSource 또는

		azureBlobFileListSource를 지정해야 합니다.
description	string	문서 모델 설명입니다.
tags	object	문서 모델과 연결된 키-값 태그 특성 목록입니다.

응답

[+] 테이블 확장

Name	형식	Description
202 Accepted		처리를 위해 요청이 수락되었지만 처리가 아직 완료되지 않았습니다. Headers Operation-Location: string Retry-After: integer
Other Status Codes	Error Response	예기치 않은 오류 응답입니다.

보안

Ocp-Apim-Subscription-Key

Type: apiKey

In: header

OAuth2Auth

Type: oauth2

Flow: accessCode

Authorization URL: <https://login.microsoftonline.com/common/oauth2/authorize>

Token URL: <https://login.microsoftonline.com/common/oauth2/token>

Scopes

[+] 테이블 확장

Name	Description

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

예제

Build Document Model

Sample Request

HTTP

HTTP

POST

<https://myendpoint.cognitiveservices.azure.com/documentintelligence/documentModels:build?api-version=2024-02-29-preview>

```
{  
    "modelId": "myCustomModel",  
    "description": "Custom model description",  
    "buildMode": "template",  
    "azureBlobSource": {  
        "containerUrl":  
        "https://myStorageAccount.blob.core.windows.net/myContainer?mySasToken",  
        "prefix": "trainingDocs/"  
    },  
    "tags": {  
        "createdBy": "myUserId"  
    }  
}
```

Sample Response

Status code: 202

HTTP

Operation-Location:

<https://myendpoint.cognitiveservices.azure.com/documentintelligence/operations/b704bb00-d130-4f3f-a1d8-ca96de3eabb4?api-version=2024-02-29-preview>

정의

Name	Description
AzureBlobContentSource	콘텐츠를 Azure Blob Storage.
AzureBlobFileListContentSource	Azure Blob Storage 파일 목록입니다.
BuildDocumentModelRequest	새 사용자 지정 문서 모델을 빌드하도록 본문을 요청합니다.
DocumentBuildMode	사용자 지정 문서 모델 빌드 모드입니다.
Error	Error 개체.
ErrorResponse	오류 응답 개체입니다.
InnerError	오류에 대한 보다 구체적인 정보를 포함하는 개체입니다.

AzureBlobContentSource

콘텐츠를 Azure Blob Storage.

Name	형식	Description
containerUrl	string	컨테이너 URL을 Azure Blob Storage.
prefix	string	Blob 이름 접두사입니다.

AzureBlobFileListContentSource

Azure Blob Storage 파일 목록입니다.

Name	형식	Description
containerUrl	string	컨테이너 URL을 Azure Blob Storage.
fileList	string	문서의 하위 집합을 지정하는 컨테이너 내의 JSONL 파일 경로입니다.

BuildDocumentModelRequest

새 사용자 지정 문서 모델을 빌드하도록 본문을 요청합니다.

Name	형식	Description
azureBlobFileListSource	AzureBlob FileList Content Source	학습 데이터를 지정하는 파일 목록을 Azure Blob Storage. azureBlobSource 또는 azureBlobFileListSource를 지정해야 합니다.
azureBlobSource	AzureBlob Content Source	학습 데이터를 포함하는 Azure Blob Storage 위치입니다. azureBlobSource 또는 azureBlobFileListSource를 지정해야 합니다.
buildMode	Document BuildMode	사용자 지정 문서 모델 빌드 모드입니다.
description	string	문서 모델 설명입니다.
modelId	string	고유한 문서 모델 이름입니다.
tags	object	문서 모델과 연결된 키-값 태그 특성 목록입니다.

DocumentBuildMode

사용자 지정 문서 모델 빌드 모드입니다.

Name	형식	Description
neural	string	다양한 시각적 템플릿이 있는 문서를 지원합니다.
template	string	유사한 시각적 템플릿을 사용하여 문서를 대상으로 지정합니다.

Error

Error 개체.

Name	형식	Description
code	string	서버 정의 오류 코드 집합 중 하나입니다.
details	Error[]	이 보고된 오류로 이어진 특정 오류에 대한 세부 정보 배열입니다.
innererror	InnerError	오류에 대한 현재 개체보다 더 구체적인 정보를 포함하는 개체입니다.

다.

message	string	사람이 읽을 수 있는 오류 표현입니다.
target	string	오류의 대상입니다.

ErrorResponse

오류 응답 개체입니다.

[\[\] 테이블 확장](#)

Name	형식	Description
error	Error	오류 정보입니다.

InnerError

오류에 대한 보다 구체적인 정보를 포함하는 개체입니다.

[\[\] 테이블 확장](#)

Name	형식	Description
code	string	서버 정의 오류 코드 집합 중 하나입니다.
innererror	InnerError	내부 오류입니다.
message	string	사람이 읽을 수 있는 오류 표현입니다.

문서 인텔리전스

아티클 • 2024. 01. 30.

REST 작업 그룹

... 테이블 확장

작업 그룹
문서 분류자
문서 모델
기타

Document Models - Analyze Document

참조

Service: Azure AI Services

API Version: 2023-07-31

문서 모델을 사용하여 문서를 분석합니다.

HTTP

```
POST {endpoint}/formrecognizer/documentModels/{modelId}:analyze?api-version=2023-07-31
```

With optional parameters:

HTTP

```
POST {endpoint}/formrecognizer/documentModels/{modelId}:analyze?pages={pages}&locale={locale}&stringIndexType={stringIndexType}&api-version=2023-07-31&features={features}
```

URI 매개 변수

[+] 테이블 확장

Name	In(다음에)	필수	형식	Description
endpoint	path	True	string url	지원되는 Cognitive Services 엔드포인트(프로토콜 및 호스트 이름(예: https://westus2.api.cognitive.microsoft.com)).
modelId	path	True	string	고유한 문서 모델 이름입니다. Regex pattern: ^[a-zA-Z0-9][a-zA-Z0-9._~-]{1,63}\$
api-version	query	True	string	요청된 API 버전입니다.
features	query		Document Analysis Feature[]	선택적 분석 기능 목록입니다.
locale	query		string	텍스트 인식 및 문서 분석을 위한 로캘 힌트입니다. 값에는 언어 코드(예: "en", "fr") 또는 BCP 47 언어 태그

(예: "en-US")만 포함될 수 있습니다.

pages	query	string	분석할 1 기반 페이지 번호 목록입니다. 예: "1-3,5,7-9" Regex pattern: <code>^(\d+(-\d+)?)(,\s*(\d+(-\d+)?))*\$</code>
StringIndex Type	query	StringIndex Type	문자열 오프셋 및 길이를 계산하는 데 사용되는 메서드입니다.

요청 헤더

Media Types: "application/json", "application/octet-stream", "application/pdf", "image/jpeg", "image/png", "image/tiff", "image/bmp", "image/heif", "text/html", "application/vnd.openxmlformats-officedocument.wordprocessingml.document", "application/vnd.openxmlformats-officedocument.spreadsheetml.sheet", "application/vnd.openxmlformats-officedocument.presentationml.presentation"

[+] 테이블 확장

Name	필수	형식	Description
Ocp-Apim-Subscription-Key	True	string	Azure Cognitive Services 구독의 비밀 키입니다.

요청 본문

Media Types: "application/json", "application/octet-stream", "application/pdf", "image/jpeg", "image/png", "image/tiff", "image/bmp", "image/heif", "text/html", "application/vnd.openxmlformats-officedocument.wordprocessingml.document", "application/vnd.openxmlformats-officedocument.spreadsheetml.sheet", "application/vnd.openxmlformats-officedocument.presentationml.presentation"

[+] 테이블 확장

Name	형식	Description
base64Source	string	분석할 문서의 Base64 인코딩입니다. urlSource 또는 base64Source를 지정해야 합니다.
urlSource	string	분석할 문서 URL입니다. urlSource 또는 base64Source를 지정해야 합니다.

응답

 테이블 확장

Name	형식	Description
202 Accepted	Headers	요청이 성공적으로 큐에 대기됩니다.
Other Status Codes	Error Response	오류가 발생했습니다.

보안

Ocp-Apim-Subscription-Key

Azure Cognitive Services 구독의 비밀 키입니다.

Type: apiKey

In: header

예제

 테이블 확장

Analyze Document from Base64
Analyze Document from Url

Analyze Document from Base64

Sample Request

HTTP
<p>HTTP POST https://myendpoint.cognitiveservices.azure.com/formrecognizer/documentMo</p>

```
deIs/prebuilt-layout:analyze?pages=1-2,4&locale=en-US&stringIndexType=textElements&api-version=2023-07-31

{
    "base64Source": "e2Jhc2U2NEVuY29kZWRQZGZ9"
}
```

Sample Response

Status code: 202

HTTP

Operation-Location:
<https://myendpoint.cognitiveservices.azure.com/formrecognizer/documentModels/prebuilt-layout/analyzeResults/3b31320d-8bab-4f88-b19c-2322a7f11034?api-version=2023-07-31>

Analyze Document from Url

Sample Request

HTTP

HTTP

POST
<https://myendpoint.cognitiveservices.azure.com/formrecognizer/documentModels/customModel:analyze?pages=1-2,4&locale=en-US&stringIndexType=textElements&api-version=2023-07-31>

```
{
    "urlSource": "http://host.com/doc.pdf"
}
```

Sample Response

Status code: 202

HTTP

Operation-Location:
<https://myendpoint.cognitiveservices.azure.com/formrecognizer/documentModels>

/customModel/analyzeResults/3b31320d-8bab-4f88-b19c-2322a7f11034?api-version=2023-07-31

정의

[+] 테이블 확장

Name	Description
AnalyzeDocumentRequest	문서 분석 매개 변수.
DocumentAnalysisFeature	선택적 분석 기능 목록입니다.
Error	오류 정보입니다.
ErrorResponse	오류 응답 개체입니다.
InnerError	자세한 오류입니다.
StringIndexType	문자열 오프셋 및 길이를 계산하는 데 사용되는 메서드입니다.

AnalyzeDocumentRequest

문서 분석 매개 변수.

[+] 테이블 확장

Name	형식	Description
base64Source	string	분석할 문서의 Base64 인코딩입니다. urlSource 또는 base64Source를 지정해야 합니다.
urlSource	string	분석할 문서 URL입니다. urlSource 또는 base64Source를 지정해야 합니다.

DocumentAnalysisFeature

선택적 분석 기능 목록입니다.

[+] 테이블 확장

Name	형식	Description
barcodes	string	문서에서 바코드 검색을 사용하도록 설정합니다.

formulas	string	문서에서 수학 식 검색을 사용하도록 설정합니다.
keyValuePairs	string	문서에서 일반 키 값 쌍(양식 필드)의 검색을 사용하도록 설정합니다.
languages	string	텍스트 콘텐츠 언어의 검색을 사용하도록 설정합니다.
ocrHighResolution	string	더 높은 해상도로 OCR을 수행하여 세밀한 인쇄로 문서를 처리합니다.
styleFont	string	다양한 글꼴 스타일을 인식할 수 있습니다.

Error

오류 정보입니다.

[\[\]](#) 테이블 확장

Name	형식	Description
code	string	오류 코드
details	Error[]	자세한 오류 목록입니다.
innererror	InnerError	자세한 오류입니다.
message	string	오류 메시지입니다.
target	string	오류의 대상입니다.

ErrorResponse

오류 응답 개체입니다.

[\[\]](#) 테이블 확장

Name	형식	Description
error	Error	오류 정보입니다.

InnerError

자세한 오류입니다.

[\[\]](#) 테이블 확장

Name	형식	Description
code	string	오류 코드
innererror	InnerError	자세한 오류입니다.
message	string	오류 메시지입니다.

StringIndexType

문자열 오프셋 및 길이를 계산하는 데 사용되는 메서드입니다.

 테이블 확장

Name	형식	Description
textElements	string	유니코드 8.0.0에 정의된 사용자 인식 표시 문자 또는 그래프 클러스터입니다.
unicodeCodePoint	string	단일 유니코드 코드 포인트로 표현되는 문자 단위입니다. Python 3에서 사용됩니다.
utf16CodeUnit	string	16비트 유니코드 코드 단위로 표현되는 문자 단위입니다. JavaScript, Java 및 .NET에서 사용됩니다.

오류 가이드 v4.0, v3.1 및 v3.0

아티클 • 2024. 03. 12.

문서 인텔리전스는 통합 디자인을 사용하여 REST API에서 발생한 모든 오류를 나타냅니다. API 작업이 4xx 또는 5xx 상태 코드를 반환할 때마다 오류에 대한 추가 정보가 다음과 같이 응답 JSON 본문에 반환됩니다.

JSON

```
{  
  "error": {  
    "code": "InvalidRequest",  
    "message": "Invalid request.",  
    "innererror": {  
      "code": "InvalidContent",  
      "message": "The file format is unsupported or corrupted. Refer to  
      documentation for the list of supported formats."  
    }  
  }  
}
```

여러 오류가 발생할 수 있는 장기 실행 작업의 경우 최상위 오류 코드는 가장 심각한 오류로 설정되며 개별 오류는 `error.details` 속성 아래에 나열됩니다. 이러한 시나리오에서 각 개별 오류의 대상 속성은 오류의 트리거를 지정합니다.

JSON

```
{  
  "status": "failed",  
  "createdDateTime": "2021-07-14T10:17:51Z",  
  "lastUpdatedDateTime": "2021-07-14T10:17:51Z",  
  "error": {  
    "code": "InternalServerError",  
    "message": "An unexpected error occurred.",  
    "details": [  
      {  
        "code": "InternalServerError",  
        "message": "An unexpected error occurred."  
      },  
      {  
        "code": "InvalidContentDimensions",  
        "message": "The input image dimensions are out of range.  
        Refer to documentation for supported image dimensions.",  
        "target": "2"  
      }  
    ]  
  }  
}
```

최상위 *error.code* 속성은 다음 오류 코드 메시지 중 하나일 수 있습니다.

테이블 확장

오류 코드	메시지	HTTP 상태
InvalidRequest	잘못된 요청입니다.	400
InvalidArgument	인수가 잘못되었습니다.	400
금지	정책 또는 기타 구성으로 인해 액세스가 금지되었습니다.	403
NotFound	리소스를 찾을 수 없습니다.	404
MethodNotAllowed	요청된 HTTP 메서드는 허용되지 않습니다.	405
충돌	충돌로 인해 요청을 완료할 수 없습니다.	409
UnsupportedMediaType	요청 콘텐츠 형식은 지원되지 않습니다.	415
InternalServerError	예기치 않은 오류가 발생했습니다.	500
ServiceUnavailable	일시적인 오류가 발생했습니다. 다시 시도하세요.	503

가능하면 내부 오류 속성에 자세한 정보가 지정됩니다.

테이블 확장

최상위 오류 코드	내부 오류 코드	메시지
충돌	ModelExists	제공된 이름을 가진 모델이 이미 있습니다.
금지	AuthorizationFailed	권한 부여 실패: {details}
금지	InvalidDataProtectionKey	데이터 보호 키가 잘못됨: {details}
금지	OutboundAccessForbidden	요청에 허용되지 않는 도메인 이름이 포함되어 있거나 현재 액세스 제어 정책을 위반합니다.
InternalServerError	Unknown	알 수 없는 오류입니다.
InvalidArgument	InvalidContentSourceFormat	잘못된 콘텐츠 원본: {details}
InvalidArgument	InvalidParameter	매개 변수 {parameterName}이 잘못됨: {details}
InvalidArgument	InvalidParameterLength	매개 변수 {parameterName} 길이는 {maxChars} 문자를 초과할 수 없습니다.

최상위 오류 코드	내부 오류 코드	메시지
InvalidArgument	InvalidSasToken	SAS(공유 액세스 서명)가 잘못됨: {details}
InvalidArgument	ParameterMissing	매개 변수 {parameterName}이 필요합니다.
InvalidRequest	ContentSourceNotAccessible	콘텐츠에 액세스할 수 없음: {details}
InvalidRequest	ContentSourceTimeout	클라이언트에서 파일을 수신하는 동안 시간이 초과되었습니다.
InvalidRequest	DocumentModelLimit	계정은 {maximumModels} 모델을 초과하여 만들 수 없습니다.
InvalidRequest	DocumentModelLimitNeural	계정은 매월 10개를 초과하여 사용자 지정 신경망 모델을 만들 수 없습니다. 더 많은 용량을 요청하려면 지원에 문의하세요.
InvalidRequest	DocumentModelLimitComposed	계정은 {details} 구성 요소 모델을 초과하여 모델을 만들 수 없습니다.
InvalidRequest	InvalidContent	파일이 손상되었거나 지원되지 않은 형식입니다. 지원되는 형식 목록은 설명서를 참조하세요.
InvalidRequest	InvalidContentDimensions	입력 이미지 차원이 범위를 벗어났습니다. 지원되는 이미지 차원에 대한 설명서를 참조하세요.
InvalidRequest	InvalidContentSize	입력 이미지가 너무 큽니다. 최대 파일 크기에 대한 설명서를 참조하세요.
InvalidRequest	InvalidFieldsDefinition	잘못된 필드: {details}
InvalidRequest	InvalidTrainingContentSize	학습 콘텐츠에는 {bytes} 바이트가 포함됩니다. 학습은 {maxBytes} 바이트로 제한됩니다.
InvalidRequest	InvalidTrainingContentPageCount	학습 콘텐츠에는 {pages} 페이지가 포함되어 있습니다. 학습은 {pages} 페이지로 제한됩니다.
InvalidRequest	ModelError	사용자 지정 모델을 사용하여 분석할 수 없음: {details}
InvalidRequest	ModelError	모델을 빌드할 수 없음: {details}
InvalidRequest	ModelError	모델을 구성할 수 없음: {details}

최상위 오류 코드	내부 오류 코드	메시지
InvalidRequest	ModelNotReady	모델이 요청한 작업을 수행할 준비가 되지 않았습니다. 학습이 완료되기를 기다리거나 작업 오류를 확인합니다.
InvalidRequest	ModelReadOnly	요청된 모델은 읽기 전용입니다.
InvalidRequest	NotSupportedApiVersion	요청된 작업에는 {minimumApiVersion} 이상이 필요합니다.
InvalidRequest	OperationNotCancellable	작업을 더 이상 취소할 수 없습니다.
InvalidRequest	TrainingContentMissing	학습 데이터가 누락됨: {details}
InvalidRequest	UnsupportedContent	콘텐츠가 지원되지 않음: {details}
NotFound	ModelNotFound	요청한 모델을 찾을 수 없습니다. 삭제하거나 여전히 빌드 중입니다.
NotFound	OperationNotFound	요청된 작업을 찾을 수 없습니다. 식별자가 잘못되었거나 작업이 만료되었습니다.

Azure Document Intelligence client library for .NET - version 1.0.0-beta.2

Article • 03/06/2024

Note: on July 2023, the Azure Cognitive Services Form Recognizer service was renamed to Azure AI Document Intelligence. Any mentions of Form Recognizer or Document Intelligence in documentation refer to the same Azure service.

Azure AI Document Intelligence is a cloud service that uses machine learning to analyze text and structured data from your documents. It includes the following main features:

- Layout - Extract text, selection marks, table structures, styles, and paragraphs, along with their bounding region coordinates from documents.
- Read - Read information about textual elements, such as page words and lines in addition to text language information.
- Prebuilt - Analyze data from certain types of common documents using prebuilt models. Supported documents include receipts, invoices, business cards, identity documents, US W2 tax forms, and more.
- Custom analysis - Build custom document models to analyze text, field values, selection marks, table structures, styles, and paragraphs from documents. Custom models are built with your own data, so they're tailored to your documents.
- Custom classification - Build custom classifier models that combine layout and language features to accurately detect and identify documents you process within your application.

[Source code ↗](#) | [Package \(NuGet\) ↗](#) | [API reference documentation ↗](#) | [Product documentation](#) | [Samples ↗](#)

Getting started

This section should include everything a developer needs to do to install and create their first client connection *very quickly*.

Install the package

Install the Azure Document Intelligence client library for .NET with [NuGet ↗](#):

.NET CLI

```
dotnet add package Azure.AI.DocumentIntelligence --prerelease
```

Note: This version of the client library defaults to the 2024-02-29-preview version of the service.

Prerequisites

- An [Azure subscription](#).
- A [Cognitive Services or Document Intelligence resource](#) to use this package.

Create a Cognitive Services or Document Intelligence resource

Document Intelligence supports both [multi-service and single-service access](#). Create a Cognitive Services resource if you plan to access multiple cognitive services under a single endpoint and key. For Document Intelligence access only, create a Document Intelligence resource. Please note that you will need a single-service resource if you intend to use [Azure Active Directory authentication](#).

You can create either resource using:

- Option 1: [Azure Portal](#).
- Option 2: [Azure CLI](#).

Below is an example of how you can create a Document Intelligence resource using the CLI:

PowerShell

```
# Create a new resource group to hold the Document Intelligence resource
# If using an existing resource group, skip this step
az group create --name <your-resource-name> --location <location>
```

PowerShell

```
# Create the Form Recognizer resource
az cognitiveservices account create \
    --name <resource-name> \
    --resource-group <resource-group-name> \
    --kind FormRecognizer \
    --sku <sku> \
    --location <location> \
    --yes
```

For more information about creating the resource or how to get the location and sku information see [here](#).

Authenticate the client

In order to interact with the Document Intelligence service, you'll need to create an instance of the [DocumentIntelligenceClient](#) class. An **endpoint** and a **credential** are necessary to instantiate the client object.

Get the endpoint

You can find the endpoint for your Document Intelligence resource using the [Azure Portal](#) or the [Azure CLI](#):

PowerShell

```
# Get the endpoint for the Document Intelligence resource
az cognitiveservices account show --name "<resource-name>" --resource-group
"<resource-group-name>" --query "properties.endpoint"
```

Either a regional endpoint or a custom subdomain can be used for authentication. They are formatted as follows:

```
Regional endpoint: https://<region>.api.cognitive.microsoft.com/
Custom subdomain: https://<resource-name>.cognitiveservices.azure.com/
```

A regional endpoint is the same for every resource in a region. A complete list of supported regional endpoints can be consulted [here](#). Please note that regional endpoints do not support AAD authentication.

A custom subdomain, on the other hand, is a name that is unique to the Document Intelligence resource. They can only be used by [single-service resources](#).

Get the API Key

The API key can be found in the [Azure Portal](#) or by running the following Azure CLI command:

PowerShell

```
az cognitiveservices account keys list --name "<resource-name>" --resource-group "<resource-group-name>"
```

Create DocumentIntelligenceClient with AzureKeyCredential

Once you have the value for the API key, create an `AzureKeyCredential`. With the endpoint and key credential, you can create the [DocumentIntelligenceClient](#):

C#

```
string endpoint = "<endpoint>";
string apiKey = "<apiKey>";
var client = new DocumentIntelligenceClient(new Uri(endpoint), new
AzureKeyCredential(apiKey));
```

Create DocumentIntelligenceClient with Azure Active Directory Credential

`AzureKeyCredential` authentication is used in the examples in this getting started guide, but you can also authenticate with Azure Active Directory using the [Azure Identity library](#). Note that regional endpoints do not support AAD authentication. Create a custom subdomain for your resource in order to use this type of authentication.

To use the [DefaultAzureCredential](#) provider shown below, or other credential providers provided with the Azure SDK, please install the `Azure.Identity` package:

.NET CLI

```
dotnet add package Azure.Identity
```

You will also need to [register a new AAD application](#) and [grant access](#) to Document Intelligence by assigning the "Cognitive Services User" role to your service principal.

Set the values of the client ID, tenant ID, and client secret of the AAD application as environment variables: AZURE_CLIENT_ID, AZURE_TENANT_ID, AZURE_CLIENT_SECRET.

C#

```
string endpoint = "<endpoint>";
var client = new DocumentIntelligenceClient(new Uri(endpoint), new
DefaultAzureCredential());
```

Key concepts

DocumentIntelligenceClient

`DocumentIntelligenceClient` provides operations for:

- Analyzing input documents using prebuilt and custom models through the `AnalyzeDocument` API.
- Detecting and identifying custom input documents with the `ClassifyDocument` API.

Sample code snippets are provided to illustrate using a `DocumentIntelligenceClient` [here](#).

More information about analyzing documents, including supported features, locales, and document types can be found in the [service documentation](#).

DocumentIntelligenceAdministrationClient

`DocumentIntelligenceAdministrationClient` provides operations for:

- Building custom models to analyze specific fields you specify by labeling your custom documents.
- Compose a model from a collection of existing models.
- Managing models created in your account.
- Copying a custom model from one Document Intelligence resource to another.
- Getting or listing operations created within the last 24 hours.
- Building and managing document classification models to accurately detect and identify documents you process within your application.

See examples for [Build a Custom Model](#), [Manage Models](#), and [Build a Document Classifier](#).

Please note that models and classifiers can also be built using a graphical user interface such as the [Document Intelligence Studio](#).

Thread safety

We guarantee that all client instance methods are thread-safe and independent of each other ([guideline](#)). This ensures that the recommendation of reusing client instances is always safe, even across threads.

Additional concepts

[Client options ↗](#) | [Accessing the response ↗](#) | [Long-running operations ↗](#) | [Handling failures ↗](#) | [Diagnostics ↗](#) | [Mocking ↗](#) | [Client lifetime ↗](#)

Examples

The following section provides several code snippets illustrating common patterns used in the Document Intelligence .NET API. Most of the snippets below make use of asynchronous service calls, but keep in mind that the Azure.AI.DocumentIntelligence package supports both synchronous and asynchronous APIs.

- [Extract Layout](#)
- [Use Prebuilt Models](#)
- [Build a Custom Model](#)
- [Manage Models](#)
- [Build a Document Classifier](#)
- [Classify a Document](#)

Extract Layout

Extract text, selection marks, table structures, styles, and paragraphs, along with their bounding region coordinates from documents.

C#

```
Uri uriSource = new Uri("<uriSource>");

var content = new AnalyzeDocumentContent()
{
    UrlSource = uriSource
};

Operation<AnalyzeResult> operation = await
client.AnalyzeDocumentAsync(WaitUntil.Completed, "prebuilt-layout",
content);
AnalyzeResult result = operation.Value;

foreach (DocumentPage page in result.Pages)
{
    Console.WriteLine($"Document Page {page.PageNumber} has
{page.Lines.Count} line(s), {page.Words.Count} word(s)," +
    $" and {page.SelectionMarks.Count} selection mark(s.)");

    for (int i = 0; i < page.Lines.Count; i++)
    {
```

```

        DocumentLine line = page.Lines[i];

        Console.WriteLine($"  Line {i}:");
        Console.WriteLine($"    Content: '{line.Content}'");

        Console.Write("      Bounding polygon, with points ordered
clockwise:");
        for (int j = 0; j < line.Polygon.Count; j += 2)
        {
            Console.Write($" ({line.Polygon[j]}, {line.Polygon[j + 1]})");
        }

        Console.WriteLine();
    }

    for (int i = 0; i < page.SelectionMarks.Count; i++)
    {
        DocumentSelectionMark selectionMark = page.SelectionMarks[i];

        Console.WriteLine($"  Selection Mark {i} is
{selectionMark.State}");
        Console.WriteLine($"    State: {selectionMark.State}");

        Console.Write("      Bounding polygon, with points ordered
clockwise:");
        for (int j = 0; j < selectionMark.Polygon.Count; j++)
        {
            Console.Write($" ({selectionMark.Polygon[j]},
{selectionMark.Polygon[j + 1]})");
        }

        Console.WriteLine();
    }
}

for (int i = 0; i < result.Paragraphs.Count; i++)
{
    DocumentParagraph paragraph = result.Paragraphs[i];

    Console.WriteLine($"Paragraph {i}:");
    Console.WriteLine($"  Content: {paragraph.Content}");

    if (paragraph.Role != null)
    {
        Console.WriteLine($"    Role: {paragraph.Role}");
    }
}

foreach (DocumentStyle style in result.Styles)
{
    // Check the style and style confidence to see if text is handwritten.
    // Note that value '0.8' is used as an example.

    bool isHandwritten = style.IsHandwritten.HasValue && style.IsHandwritten
== true;
}

```

```

if (isHandwritten && style.Confidence > 0.8)
{
    Console.WriteLine($"Handwritten content found:");

    foreach (DocumentSpan span in style.Spans)
    {
        var handwrittenContent = result.Content.Substring(span.Offset,
span.Length);
        Console.WriteLine($" {handwrittenContent}");
    }
}

for (int i = 0; i < result.Tables.Count; i++)
{
    DocumentTable table = result.Tables[i];

    Console.WriteLine($"Table {i} has {table.RowCount} rows and
{table.ColumnCount} columns.");

    foreach (DocumentTableCell cell in table.Cells)
    {
        Console.WriteLine($" Cell ({cell.RowIndex}, {cell.ColumnIndex}) is
a '{cell.Kind}' with content: {cell.Content}");
    }
}

```

For more information, see [here](#).

Use Prebuilt Models

Analyze data from certain types of common documents using prebuilt models provided by the Document Intelligence service.

For example, to analyze fields from an invoice, use the prebuilt Invoice model provided by passing the `prebuilt-invoice` model ID to the `AnalyzeDocumentAsync` method:

C#

```

Uri uriSource = new Uri("<uriSource>");

var content = new AnalyzeDocumentContent()
{
    UrlSource = uriSource
};

Operation<AnalyzeResult> operation = await
client.AnalyzeDocumentAsync(WaitUntil.Completed, "prebuilt-invoice",
content);
AnalyzeResult result = operation.Value;

```

```
// To see the list of all the supported fields returned by service and its
// corresponding types for the
// prebuilt-invoice model, see:
// https://aka.ms/azsdk/formrecognizer/invoicefieldschema

for (int i = 0; i < result.Documents.Count; i++)
{
    Console.WriteLine($"Document {i}:");

    AnalyzedDocument document = result.Documents[i];

    if (document.Fields.TryGetValue("VendorName", out DocumentField
vendorNameField)
        && vendorNameField.Type == DocumentFieldType.String)
    {
        string vendorName = vendorNameField.ValueString;
        Console.WriteLine($"Vendor Name: '{vendorName}', with confidence
{vendorNameField.Confidence}");
    }

    if (document.Fields.TryGetValue("CustomerName", out DocumentField
customerNameField)
        && customerNameField.Type == DocumentFieldType.String)
    {
        string customerName = customerNameField.ValueString;
        Console.WriteLine($"Customer Name: '{customerName}', with confidence
{customerNameField.Confidence}");
    }

    if (document.Fields.TryGetValue("Items", out DocumentField itemsField)
        && itemsField.Type == DocumentFieldType.List)
    {
        foreach (DocumentField itemField in itemsField.ValueList)
        {
            Console.WriteLine("Item:");

            if (itemField.Type == DocumentFieldType.Dictionary)
            {
                IReadOnlyDictionary<string, DocumentField> itemFields =
itemField.ValueDictionary;

                if (itemFields.TryGetValue("Description", out DocumentField
itemDescriptionField)
                    && itemDescriptionField.Type ==
DocumentFieldType.String)
                {
                    string itemDescription =
itemDescriptionField.ValueString;
                    Console.WriteLine($" Description: '{itemDescription}',
with confidence {itemDescriptionField.Confidence}");
                }
            }

            if (itemFields.TryGetValue("Amount", out DocumentField
itemAmountField)
```

```

        && itemAmountField.Type == DocumentFieldType.Currency)
    {
        CurrencyValue itemAmount =
itemAmountField.ValueCurrency;
        Console.WriteLine($" Amount:
'{itemAmount.CurrencySymbol}{itemAmount.Amount}', with confidence
{itemAmountField.Confidence}");
    }
}
}

if (document.Fields.TryGetValue("SubTotal", out DocumentField
subTotalField)
    && subTotalField.Type == DocumentFieldType.Currency)
{
    CurrencyValue subTotal = subTotalField.ValueCurrency;
    Console.WriteLine($"Sub Total: '{subTotal.CurrencySymbol}
{subTotal.Amount}', with confidence {subTotalField.Confidence}");
}

if (document.Fields.TryGetValue("TotalTax", out DocumentField
totalTaxField)
    && totalTaxField.Type == DocumentFieldType.Currency)
{
    CurrencyValue totalTax = totalTaxField.ValueCurrency;
    Console.WriteLine($"Total Tax: '{totalTax.CurrencySymbol}
{totalTax.Amount}', with confidence {totalTaxField.Confidence}");
}

if (document.Fields.TryGetValue("InvoiceTotal", out DocumentField
invoiceTotalField)
    && invoiceTotalField.Type == DocumentFieldType.Currency)
{
    CurrencyValue invoiceTotal = invoiceTotalField.ValueCurrency;
    Console.WriteLine($"Invoice Total: '{invoiceTotal.CurrencySymbol}
{invoiceTotal.Amount}', with confidence {invoiceTotalField.Confidence}");
}
}

```

You are not limited to invoices! There are a couple of prebuilt models to choose from, each of which has its own set of supported fields. More information about the supported document types can be found in the [service documentation ↗](#).

For more information, see [here ↗](#).

Build a Custom Model

Build a custom model on your own document type. The resulting model can be used to analyze values from the types of documents it was built on.

C#

```
// For this sample, you can use the training documents found in the
`trainingFiles` folder.
// Upload the documents to your storage container and then generate a
container SAS URL. Note
// that a container URI without SAS is accepted only when the container is
public or has a
// managed identity configured.

// For instructions to set up documents for training in an Azure Blob
Storage Container, please see:
// https://aka.ms/azsdk/formrecognizer/buildcustommodel

string modelId = "<modelId>";
Uri blobContainerUri = new Uri("<blobContainerUri>");

// We are selecting the Template build mode in this sample. For more
information about the available
// build modes and their differences, see:
// https://aka.ms/azsdk/formrecognizer/buildmode

var content = new BuildDocumentModelContent(modelId,
DocumentBuildMode.Template)
{
    AzureBlobSource = new AzureBlobContentSource(blobContainerUri)
};

Operation<DocumentModelDetails> operation = await
client.BuildDocumentModelAsync(WaitUntil.Completed, content);
DocumentModelDetails model = operation.Value;

Console.WriteLine($"Model ID: {model.ModelId}");
Console.WriteLine($"Created on: {model.CreatedOn}");

Console.WriteLine("Document types the model can recognize:");
foreach (KeyValuePair<string, DocumentTypeDetails> docType in
model.DocTypes)
{
    Console.WriteLine($"  Document type: '{docType.Key}', which has the
following fields:");
    foreach (KeyValuePair<string, DocumentFieldSchema> schema in
docType.Value.FieldSchema)
    {
        Console.WriteLine($"    Field: '{schema.Key}', with confidence
{docType.Value.FieldConfidence[schema.Key]}");
    }
}
```

For more information, see [here ↗](#).

Manage Models

Manage the models stored in your account.

C#

```
// Check number of custom models in the Document Intelligence resource, and  
// the maximum number  
// of custom models that can be stored.  
  
ResourceDetails resourceDetails = await client.GetResourceInfoAsync();  
  
Console.WriteLine($"Resource has  
{resourceDetails.CustomDocumentModels.Count} custom models.");  
Console.WriteLine($"It can have at most  
{resourceDetails.CustomDocumentModels.Limit} custom models.");  
  
// Get a model by ID.  
string modelId = "<modelId>";  
DocumentModelDetails model = await client.GetModelAsync(modelId);  
  
Console.WriteLine($"Details about model with ID '{model.ModelId}':");  
Console.WriteLine($" Created on: {model.CreatedOn}");  
Console.WriteLine($" Expires on: {model.ExpiresOn}");  
  
// List up to 10 models currently stored in the resource.  
int count = 0;  
  
await foreach (DocumentModelDetails modelItem in client.GetModelsAsync())  
{  
    Console.WriteLine($"Model details:");  
    Console.WriteLine($" Model ID: {modelItem.ModelId}");  
    Console.WriteLine($" Description: {modelItem.Description}");  
    Console.WriteLine($" Created on: {modelItem.CreatedOn}");  
    Console.WriteLine($" Expires on: {model.ExpiresOn}");  
  
    if (++count == 10)  
    {  
        break;  
    }  
}
```

For more information, see [here](#).

Build a Document Classifier

Build a document classifier by uploading custom training documents.

C#

```
// For this sample, you can use the training documents found in the  
`classifierTrainingFiles` folder.  
// Upload the documents to your storage container and then generate a
```

```

container SAS URL. Note
// that a container URI without SAS is accepted only when the container is
public or has a
// managed identity configured.

// For instructions to set up documents for training in an Azure Blob
Storage Container, please see:
// https://aka.ms/azsdk/formrecognizer/buildclassifiermodel

string classifierId = "<classifierId>";
Uri blobContainerUri = new Uri("<blobContainerUri>");
var sourceA = new AzureBlobContentSource(blobContainerUri) { Prefix = "IRS-
1040-A/train" };
var sourceB = new AzureBlobContentSource(blobContainerUri) { Prefix = "IRS-
1040-B/train" };
var docTypeA = new ClassifierDocumentTypeDetails() { AzureBlobSource =
sourceA };
var docTypeB = new ClassifierDocumentTypeDetails() { AzureBlobSource =
sourceB };
var docTypes = new Dictionary<string, ClassifierDocumentTypeDetails>()
{
    { "IRS-1040-A", docTypeA },
    { "IRS-1040-B", docTypeB }
};

var content = new BuildDocumentClassifierContent(classifierId, docTypes);

Operation<DocumentClassifierDetails> operation = await
client.BuildClassifierAsync(WaitUntil.Completed, content);
DocumentClassifierDetails classifier = operation.Value;

Console.WriteLine($"Classifier ID: {classifier.ClassifierId}");
Console.WriteLine($"Created on: {classifier.CreatedOn}");

Console.WriteLine("Document types the classifier can recognize:");
foreach (KeyValuePair<string, ClassifierDocumentTypeDetails> docType in
classifier.DocTypes)
{
    Console.WriteLine($"  {docType.Key}");
}

```

For more information, see [here ↗](#).

Classify a Document

Use document classifiers to accurately detect and identify documents you process within your application.

C#

```

string classifierId = "<classifierId>";
Uri uriSource = new Uri("<uriSource>");

var content = new ClassifyDocumentContent()
{
    UrlSource = uriSource
};

Operation<AnalyzeResult> operation = await
client.ClassifyDocumentAsync(WaitUntil.Completed, classifierId, content);
AnalyzeResult result = operation.Value;

Console.WriteLine($"Input was classified by the classifier with ID
'{result.ModelId}'.");

foreach (AnalyzedDocument document in result.Documents)
{
    Console.WriteLine($"Found a document of type: {document.DocType}");
}

```

For more information, see [here](#).

Troubleshooting

General

When you interact with the Document Intelligence client library using the .NET SDK, errors returned by the service will result in a `RequestFailedException` with the same HTTP status code returned by the [REST API](#) request.

For example, if you submit a receipt image with an invalid `Uri`, a `400` error is returned, indicating "Bad Request".

C#

```

var content = new AnalyzeDocumentContent()
{
    UrlSource = new Uri("http://invalid.uri")
};

try
{
    Operation<AnalyzeResult> operation = await
client.AnalyzeDocumentAsync(WaitUntil.Completed, "prebuilt-receipt",
content);
}
catch (RequestFailedException e)
{

```

```
        Console.WriteLine(e.ToString());
    }
```

You will notice that additional information is logged, like the client request ID of the operation.

Message:

```
Azure.RequestFailedException: Service request failed.  
Status: 400 (Bad Request)  
ErrorCode: InvalidRequest
```

Content:

```
{"error": {"code": "InvalidRequest", "message": "Invalid  
request.", "innererror": {"code": "InvalidContent", "message": "The file is  
corrupted or format is unsupported. Refer to documentation for the list of  
supported formats."}}}
```

Headers:

```
Transfer-Encoding: chunked  
x-envoy-upstream-service-time: REDACTED  
apim-request-id: REDACTED  
Strict-Transport-Security: REDACTED  
X-Content-Type-Options: REDACTED  
Date: Fri, 01 Oct 2021 02:55:44 GMT  
Content-Type: application/json; charset=utf-8
```

Error codes and messages raised by the Document Intelligence service can be found in the [service documentation](#).

Setting up console logging

The simplest way to see the logs is to enable the console logging.

To create an Azure SDK log listener that outputs messages to console use the `AzureEventSourceListener.CreateConsoleLogger` method.

C#

```
// Setup a listener to monitor logged events.  
using AzureEventSourceListener listener =  
AzureEventSourceListener.CreateConsoleLogger();
```

To learn more about other logging mechanisms see [Diagnostics Samples](#).

Next steps

Samples showing how to use the Document Intelligence library are available in this GitHub repository. Samples are provided for each main functional area:

- [Extract the layout of a document ↗](#)
- [Analyze a document with a prebuilt model ↗](#)
- [Build a custom model ↗](#)
- [Manage models ↗](#)
- [Classify a document ↗](#)
- [Build a document classifier ↗](#)
- [Get and List document model operations ↗](#)
- [Compose a model ↗](#)
- [Copy a custom model between Document Intelligence resources ↗](#)
- [Analyze a document with add-on capabilities ↗](#)
- [Extract the layout of a document as Markdown ↗](#)

Contributing

This project welcomes contributions and suggestions. Most contributions require you to agree to a Contributor License Agreement (CLA) declaring that you have the right to, and actually do, grant us the rights to use your contribution. For details, visit cla.microsoft.com ↗.

When you submit a pull request, a CLA-bot will automatically determine whether you need to provide a CLA and decorate the PR appropriately (e.g., label, comment). Simply follow the instructions provided by the bot. You will only need to do this once across all repos using our CLA.

This project has adopted the [Microsoft Open Source Code of Conduct](#) ↗. For more information see the [Code of Conduct FAQ](#) ↗ or contact opencode@microsoft.com with any additional questions or comments.



Collaborate with us on
GitHub



Azure SDK for .NET
feedback

The source for this content can be found on GitHub, where you can also create and review issues and pull requests. For more information, see [our contributor guide](#).

Azure SDK for .NET is an open source project. Select a link to provide feedback:

 [Open a documentation issue](#)

 [Provide product feedback](#)

Azure DocumentIntelligence client library for Java - version 1.0.0-beta.2

Article • 03/22/2024

Azure Document Intelligence ([previously known as Form Recognizer](#)) is a cloud service that uses machine learning to analyze text and structured data from your documents. It includes the following main features:

- Layout - Analyze text, table structures, and selection marks, along with their bounding region coordinates, from documents.
- Prebuilt - Analyze data from certain types of common documents (such as receipts, invoices, identity documents or US W2 tax forms) using prebuilt models.
- Custom - Build custom models to extract text, field values, selection marks, and table data from documents. Custom models are built with your own data, so they're tailored to your documents.
- Read - Read information about textual elements, such as page words and lines in addition to text language information.
- Classifiers - Build custom classifiers to categorize documents into predefined classes.

[Source code](#) | [Package \(Maven\)](#) | [API reference documentation](#) | [Product Documentation](#) | [Samples](#)

Getting started

Prerequisites

- [Java Development Kit \(JDK\)](#) with version 8 or above
- [Azure Subscription](#)
- [AI Services or Document Intelligence account](#) to use this package.

Adding the package to your product

XML

```
<dependency>
    <groupId>com.azure</groupId>
    <artifactId>azure-ai-documentintelligence</artifactId>
```

```
<version>1.0.0-beta.1</version>
</dependency>
```

Note: This version of the client library defaults to the "2024-02-29-preview" version of the service.

This table shows the relationship between SDK versions and supported API versions of the service:

[+] Expand table

SDK version	Supported API version of service
1.0.0-beta.1	2023-10-31-preview
1.0.0-beta.2	2024-02-29-preview

Note: Please rely on the older `azure-ai-formrecognizer` library through the older service API versions for retired models, such as "prebuilt-businessCard" and "prebuilt-document". For more information, see [Changelog](#). The below table describes the relationship of each client and its supported API version(s):

[+] Expand table

API version	Supported clients
2023-10-31-preview, 2024-02-29-preview	DocumentIntelligenceClient and DocumentIntelligenceAsyncClient
2023-07-31	DocumentAnalysisClient and DocumentModelAdministrationClient in <code>azure-ai-formrecognizer</code> SDK

Please see the [Migration Guide](#) for more information about migrating from `azure-ai-formrecognizer` to `azure-ai-documentintelligence`.

Authentication

In order to interact with the Azure Document Intelligence Service you'll need to create an instance of client class, [DocumentIntelligenceAsyncClient](#) or [DocumentIntelligenceClient](#) by using [DocumentIntelligenceClientBuilder](#). To configure a client for use with Azure DocumentIntelligence, provide a valid endpoint URI to an Azure DocumentIntelligence resource along with a corresponding key credential,

token credential, or [Azure Identity](#) credential that's authorized to use the Azure DocumentIntelligence resource.

Create an Azure DocumentIntelligence client with key credential

Get Azure DocumentIntelligence `key` credential from the Azure Portal.

Java

```
DocumentIntelligenceClient documentIntelligenceClient = new  
DocumentIntelligenceClientBuilder()  
.credential(new AzureKeyCredential("{key}"))  
.endpoint("{endpoint}")  
.buildClient();
```

or

Java

```
DocumentIntelligenceAdministrationClient client =  
new DocumentIntelligenceAdministrationClientBuilder()  
.credential(new AzureKeyCredential("{key}"))  
.endpoint("{endpoint}")  
.buildClient();
```

Create an Azure DocumentIntelligence client with Azure Active Directory credential

Azure SDK for Java supports an Azure Identity package, making it easy to get credentials from Microsoft identity platform.

Authentication with AAD requires some initial setup:

- Add the Azure Identity package

XML

```
<dependency>  
  <groupId>com.azure</groupId>  
  <artifactId>azure-identity</artifactId>  
  <version>1.11.3</version>  
</dependency>
```

After setup, you can choose which type of [credential](#) from azure-identity to use. As an example, [DefaultAzureCredential](#) can be used to authenticate the client: Set the values

of the client ID, tenant ID, and client secret of the AAD application as environment variables: `AZURE_CLIENT_ID`, `AZURE_TENANT_ID`, `AZURE_CLIENT_SECRET`.

Authorization is easiest using [DefaultAzureCredential](#). It finds the best credential to use in its running environment. For more information about using Azure Active Directory authorization with DocumentIntelligence service, please refer to [the associated documentation](#).

Java

```
DocumentIntelligenceAsyncClient documentIntelligenceAsyncClient = new  
DocumentIntelligenceClientBuilder()  
    .credential(new DefaultAzureCredentialBuilder().build())  
    .endpoint("{endpoint}")  
    .buildAsyncClient();
```

Key concepts

DocumentAnalysisClient

The [DocumentAnalysisClient](#) and [DocumentAnalysisAsyncClient](#) provide both synchronous and asynchronous operations for analyzing input documents using custom and prebuilt models through the `beginAnalyzeDocument` API. See a full list of supported models [here](#).

Sample code snippets to illustrate using a DocumentAnalysisClient [here](#). More information about analyzing documents, including supported features, locales, and document types can be found [here](#).

DocumentModelAdministrationClient

The [DocumentModelAdministrationClient](#) and [DocumentModelAdministrationAsyncClient](#) provide both synchronous and asynchronous operations

- Build custom document analysis models to analyze text content, fields, and values found in your custom documents. See example [Build a document model](#). A `DocumentModelDetails` is returned indicating the document types that the model can analyze, along with the fields and schemas it will extract.
- Managing models created in your account by building, listing, deleting, and see the limit of custom models your account. See example [Manage models](#).
- Copying a custom model from one Document Intelligence resource to another.

- Creating a composed model from a collection of existing built models.
- Listing document model operations associated with the Document Intelligence resource.

Sample code snippets are provided to illustrate using a `DocumentModelAdministrationClient` [here](#).

Long-running operations

Long-running operations are operations that consist of an initial request sent to the service to start an operation, followed by polling the service at intervals to determine whether the operation has completed or failed, and if it has succeeded, to get the result.

Methods that build models, analyze values from documents, or copy and compose models are modeled as long-running operations. The client exposes a

`begin<MethodName>` method that returns a `SyncPoller` or `PollerFlux` instance. Callers should wait for the operation to be completed by calling `getFinalResult()` on the returned operation from the `begin<MethodName>` method. Sample code snippets are provided to illustrate using long-running operations [below](#).

Examples

The following section provides several code snippets covering some of the most common Document Intelligence tasks, including:

- [Analyze Layout](#)
- [Use Prebuilt Models](#)
- [Build a Document Model](#)
- [Analyze Documents using a Custom Model](#)
- [Manage Your Models](#)

Analyze Layout

Analyze text, table structures, and selection marks like radio buttons and check boxes, along with their bounding box coordinates from documents without the need to build a model.

Java

```
File layoutDocument = new File("local/file_path/filename.png");
Path filePath = layoutDocument.toPath();
BinaryData layoutDocumentData = BinaryData.fromFile(filePath, (int)
```

```
layoutDocument.length());  
  
SyncPoller<AnalyzeResultOperation, AnalyzeResultOperation>  
analyzeLayoutResultPoller =  
    documentIntelligenceClient.beginAnalyzeDocument("prebuilt-layout",  
        null,  
        null,  
        null,  
        null,  
        null,  
        null,  
        new  
    AnalyzeDocumentRequest().setBase64Source(Files.readAllBytes(layoutDocument.toPath())));  
  
AnalyzeResult analyzeLayoutResult =  
    analyzeLayoutResultPoller.getFinalResult().getAnalyzeResult();  
  
// pages  
analyzeLayoutResult.getPages().forEach(documentPage -> {  
    System.out.printf("Page has width: %.2f and height: %.2f, measured with  
unit: %s%n",  
        documentPage.getWidth(),  
        documentPage.getHeight(),  
        documentPage.getUnit());  
  
// lines  
documentPage.getLines().forEach(documentLine ->  
    System.out.printf("Line '%s' is within a bounding box %s.%n",  
        documentLine.getContent(),  
        documentLine.getPolygon().toString()));  
  
// selection marks  
documentPage.getSelectionMarks().forEach(documentSelectionMark ->  
    System.out.printf("Selection mark is '%s' and is within a bounding  
box %s with confidence %.2f.%n",  
        documentSelectionMark.getState().toString(),  
        documentSelectionMark.getPolygon().toString(),  
        documentSelectionMark.getConfidence()));  
});  
  
// tables  
List<DocumentTable> tables = analyzeLayoutResult.getTables();  
for (int i = 0; i < tables.size(); i++) {  
    DocumentTable documentTable = tables.get(i);  
    System.out.printf("Table %d has %d rows and %d columns.%n", i,  
        documentTable.getRowCount(),  
        documentTable.getColumnCount());  
    documentTable.getCells().forEach(documentTableCell -> {  
        System.out.printf("Cell '%s', has row index %d and column index  
%d.%n", documentTableCell.getContent(),  
            documentTableCell.getRowIndex(),  
            documentTableCell.getColumnIndex());  
    });
```

```
        System.out.println();
    }
```

Use Prebuilt Models

Extract fields from select document types such as receipts, invoices, and identity documents using prebuilt models provided by the Document Intelligence service. Supported prebuilt models are:

- Analyze receipts using the `prebuilt-receipt` model (fields recognized by the service can be found [here](#)).
- Analyze invoices using the `prebuilt-invoice` model (fields recognized by the service can be found [here](#)).
- Analyze identity documents using the `prebuilt-idDocuments` model (fields recognized by the service can be found [here](#)).
- Analyze US W2 tax forms using the `prebuilt-tax.us.w2` model. [Supported fields](#).

For example, to analyze fields from a sales receipt, into the `beginAnalyzeDocumentFromUrl` method:

Java

```
File sourceFile = new File("../documentintelligence/azure-ai-
documentintelligence/src/samples/resources/"
    + "sample-forms/receipts/contoso-allinone.jpg");

SyncPoller<AnalyzeResultOperation, AnalyzeResultOperation>
analyzeReceiptPoller =
    documentIntelligenceClient.beginAnalyzeDocument("prebuilt-receipt",
        null,
        null,
        null,
        null,
        null,
        null,
        new
    AnalyzeDocumentRequest().setBase64Source(Files.readAllBytes(sourceFile.toPath())));
    }

AnalyzeResult receiptResults =
    analyzeReceiptPoller.getFinalResult().getAnalyzeResult();

for (int i = 0; i < receiptResults.getDocuments().size(); i++) {
    Document analyzedReceipt = receiptResults.getDocuments().get(i);
    Map<String, DocumentField> receiptFields = analyzedReceipt.getFields();
    System.out.printf("----- Analyzing receipt info %d -----%n",
        i);
    DocumentField merchantNameField = receiptFields.get("MerchantName");
```

```

        if (merchantNameField != null) {
            if (DocumentFieldType.STRING == merchantNameField.getType()) {
                String merchantName = merchantNameField.getValueString();
                System.out.printf("Merchant Name: %s, confidence: %.2f%n",
                    merchantName, merchantNameField.getConfidence());
            }
        }

        DocumentField merchantPhoneNumberField =
receiptFields.get("MerchantPhoneNumber");
        if (merchantPhoneNumberField != null) {
            if (DocumentFieldType.PHONE_NUMBER ==
merchantPhoneNumberField.getType()) {
                String merchantAddress =
merchantPhoneNumberField.getValuePhoneNumber();
                System.out.printf("Merchant Phone number: %s, confidence:
%.2f%n",
                    merchantAddress, merchantPhoneNumberField.getConfidence());
            }
        }

        DocumentField merchantAddressField =
receiptFields.get("MerchantAddress");
        if (merchantAddressField != null) {
            if (DocumentFieldType.STRING == merchantAddressField.getType()) {
                String merchantAddress = merchantAddressField.getValueString();
                System.out.printf("Merchant Address: %s, confidence: %.2f%n",
                    merchantAddress, merchantAddressField.getConfidence());
            }
        }

        DocumentField transactionDateField =
receiptFields.get("TransactionDate");
        if (transactionDateField != null) {
            if (DocumentFieldType.DATE == transactionDateField.getType()) {
                LocalDate transactionDate = transactionDateField.getValueDate();
                System.out.printf("Transaction Date: %s, confidence: %.2f%n",
                    transactionDate, transactionDateField.getConfidence());
            }
        }
    }
}

```

For more information and samples using prebuilt models, see:

- [Identity Documents ↗](#)
- [Invoices ↗](#)
- [Receipts sample ↗](#)

Build a document model

Build a machine-learned model on your own document type. The resulting model will be able to analyze values from the types of documents it was built on. Provide a container SAS url to your Azure Storage Blob container where you're storing the training documents. See details on setting this up in the [service quickstart documentation](#).

Note

You can use the [Document Intelligence Studio preview](#) for creating a labeled file for your training forms. More details on setting up a container and required file structure can be found in [here](#).

Java

```
// Build custom document analysis model
String blobContainerUrl = "{SAS_URL_of_your_container_in_blob_storage}";
// The shared access signature (SAS) Url of your Azure Blob Storage
// container with your forms.
SyncPoller<DocumentModelBuildOperationDetails, DocumentModelDetails>
buildOperationPoller =
    administrationClient.beginBuildDocumentModel(new
BuildDocumentModelRequest("modelID", DocumentBuildMode.TEMPLATE)
    .setAzureBlobSource(new AzureBlobContentSource(blobContainerUrl)));

DocumentModelDetails documentModelDetails =
buildOperationPoller.getFinalResult();

// Model Info
System.out.printf("Model ID: %s%n", documentModelDetails.getModelId());
System.out.printf("Model Description: %s%n",
documentModelDetails.getDescription());
System.out.printf("Model created on: %s%n%n",
documentModelDetails.getCreatedDateTime());

System.out.println("Document Fields:");
documentModelDetails.getDocTypes().forEach((key, documentTypeDetails) -> {
    documentTypeDetails.getFieldSchema().forEach((field,
documentFieldSchema) -> {
        System.out.printf("Field: %s", field);
        System.out.printf("Field type: %s", documentFieldSchema.getType());
        System.out.printf("Field confidence: %.2f",
documentTypeDetails.getFieldConfidence().get(field));
    });
});
```

Analyze Documents using a Custom Model

Analyze the key/value pairs and table data from documents. These models are built with your own data, so they're tailored to your documents. You should only analyze documents of the same doc type that the custom model was built on.

Java

```
String documentUrl = "{document-url}";
String modelId = "{custom-built-model-ID}";
SyncPoller<AnalyzeResultOperation, AnalyzeResultOperation>
analyzeDocumentPoller =
documentIntelligenceClient.beginAnalyzeDocument(modelId,
    "1",
    "en-US",
    StringIndexType.TEXT_ELEMENTS,
    Arrays.asList(DocumentAnalysisFeature.LANGUAGES),
    null,
    ContentFormat.TEXT,
    new AnalyzeDocumentRequest().setUrlSource(documentUrl));

AnalyzeResult analyzeResult =
analyzeDocumentPoller.getFinalResult().getAnalyzeResult();

for (int i = 0; i < analyzeResult.getDocuments().size(); i++) {
    final Document analyzedDocument = analyzeResult.getDocuments().get(i);
    System.out.printf("----- Analyzing custom document %d -----%n", i);
    System.out.printf("Analyzed document has doc type %s with confidence : %.2f%n",
        analyzedDocument.getDocType(), analyzedDocument.getConfidence());
}

analyzeResult.getPages().forEach(documentPage -> {
    System.out.printf("Page has width: %.2f and height: %.2f, measured with unit: %s%n",
        documentPage.getWidth(),
        documentPage.getHeight(),
        documentPage.getUnit());

    // lines
    documentPage.getLines().forEach(documentLine ->
        System.out.printf("Line '%s' is within a bounding polygon %s.%n",
            documentLine.getContent(),
            documentLine.getPolygon()));

    // words
    documentPage.getWords().forEach(documentWord ->
        System.out.printf("Word '%s' has a confidence score of %.2f.%n",
            documentWord.getContent(),
            documentWord.getConfidence()));
});

// tables
List<DocumentTable> tables = analyzeResult.getTables();
for (int i = 0; i < tables.size(); i++) {
    DocumentTable documentTable = tables.get(i);
    System.out.printf("Table %d has %d rows and %d columns.%n", i,
        documentTable.getRowCount(),
        documentTable.getColumnCount());
```

```

documentTable.getCells().forEach(documentTableCell -> {
    System.out.printf("Cell '%s', has row index %d and column index
%d.%n",
        documentTableCell.getContent(),
        documentTableCell.getRowIndex(),
        documentTableCell.getColumnIndex());
});
System.out.println();
}

```

Manage your models

Manage the models in your Document Intelligence account.

Java

```

ResourceDetails resourceDetails = administrationClient.getResourceInfo();
System.out.printf("The resource has %s models, and we can have at most %s
models.%n",
    resourceDetails.getCustomDocumentModels().getCount(),
    resourceDetails.getCustomDocumentModels().getLimit());

// Next, we get a paged list of all of our models
PagedIterable<DocumentModelDetails> customDocumentModels =
administrationClient.listModels();
System.out.println("We have following models in the account:");
customDocumentModels.forEach(documentModelInfo -> {
    System.out.println();
    // get custom document analysis model info
    DocumentModelDetails documentModel =
administrationClient.getModel(documentModelInfo.getModelId());
    System.out.printf("Model ID: %s%n", documentModel.getModelId());
    System.out.printf("Model Description: %s%n",
documentModel.getDescription());
    System.out.printf("Model created on: %s%n",
documentModel.getCreatedDateTime());
    if (documentModel.getDocTypes() != null) {
        documentModel.getDocTypes().forEach((key, documentTypeDetails) -> {
            documentTypeDetails.getFieldSchema().forEach((field,
documentFieldSchema) -> {
                System.out.printf("Field: %s, ", field);
                System.out.printf("Field type: %s, ",
documentFieldSchema.getType());
                if (documentTypeDetails.getFieldConfidence() != null) {
                    System.out.printf("Field confidence: %.2f%n",
documentTypeDetails.getFieldConfidence().get(field));
                }
            });
        });
    };
}

```

```
});
```

For more detailed examples, refer to [samples ↗](#).

Troubleshooting

Enable client logging

You can set the `AZURE_LOG_LEVEL` environment variable to view logging statements made in the client library. For example, setting `AZURE_LOG_LEVEL=2` would show all informational, warning, and error log messages. The log levels can be found here: [log levels ↗](#).

Default HTTP Client

All client libraries by default use the Netty HTTP client. Adding the above dependency will automatically configure the client library to use the Netty HTTP client. Configuring or changing the HTTP client is detailed in the [HTTP clients wiki ↗](#).

Default SSL library

All client libraries, by default, use the Tomcat-native Boring SSL library to enable native-level performance for SSL operations. The Boring SSL library is an uber jar containing native libraries for Linux / macOS / Windows, and provides better performance compared to the default SSL implementation within the JDK. For more information, including how to reduce the dependency size, refer to the [performance tuning ↗](#) section of the wiki.

Next steps

- Samples are explained in detail [here ↗](#).

Contributing

For details on contributing to this repository, see the [contributing guide ↗](#).

1. Fork it
2. Create your feature branch (`git checkout -b my-new-feature`)

3. Commit your changes (`git commit -am 'Add some feature'`)
4. Push to the branch (`git push origin my-new-feature`)
5. Create new Pull Request

 Collaborate with us on GitHub

The source for this content can be found on GitHub, where you can also create and review issues and pull requests. For more information, see [our contributor guide](#).



Azure SDK for Java feedback

Azure SDK for Java is an open source project. Select a link to provide feedback:

 [Open a documentation issue](#)

 [Provide product feedback](#)

JavaScript용 Azure Document Intelligence(이전의 Form Recognizer) REST 클라이언트 라이브러리 - 버전 1.0.0-beta.2

아티클 • 2024. 03. 22.

문서에서 콘텐츠, 레이아웃 및 구조화된 데이터를 추출합니다.

이 라이브러리를 사용하려면 [REST 클라이언트 문서에 크게 의존하세요.](#)

참고: Form Recognizer 문서 인텔리전스로 리브랜딩되었습니다. [에서 로 마이그레이션 가이드](#) [@azure/ai-form-recognizer@azure-rest/ai-document-intelligence](#) 를 검사.

주요 링크:

- [소스 코드](#)
- [패키지\(NPM\)](#)
- [API 참조 설명서](#)
- [샘플](#)
- [Changelog](#)
- [Form Recognizer 마이그레이션 가이드](#)

이 버전의 클라이언트 라이브러리는 기본적으로 서비스 버전으로 설정됩니다 "2024-02-29-preview".

다음 표에서는 SDK 버전 및 지원되는 API 버전의 서비스 간 관계를 보여 줍니다.

[\[+\] 테이블 확장](#)

SDK 버전	지원되는 API 버전의 서비스
1.0.0-beta.2	2024-02-29-preview
1.0.0-beta.1	2023-10-31-preview

및 와 같은 "prebuilt-businessCard" "prebuilt-document" 사용 중지된 모델에 대한 이전 서비스 API 버전을 통해 이전 [@azure/ai-form-recognizer](#) 라이브러리를 사용하세요. 자세한 내용은 [Changelog](#) 를 참조하세요.

아래 표에서는 각 클라이언트와 지원되는 API 버전의 관계를 설명합니다.

[] 테이블 확장

서비스 API 버전	지원되는 클라이언트	패키지
2024-02-29- preview	DocumentIntelligenceClient	@azure-rest/ai-document-intelligence 버전 1.0.0-beta.2
2023-10-31- preview	DocumentIntelligenceClient	@azure-rest/ai-document-intelligence 버전 1.0.0-beta.1
2023-07-31	DocumentAnalysisClient 및 DocumentModelAdministrationClient	@azure/ai-form-recognizer 버전 ^5.0.0
2022-08-01	DocumentAnalysisClient 및 DocumentModelAdministrationClient	@azure/ai-form-recognizer 버전 ^4.0.0

시작

현재 지원되는 환경

- Node.js의 LTS 버전

사전 요구 사항

- 이 패키지를 사용하려면 [Azure 구독](#)이 있어야 합니다.

@azure-rest/ai-document-intelligence 패키지를 설치합니다.

을 사용하여 JavaScript용 Azure DocumentIntelligence(이전의 FormRecognizer) REST 클라이언트 REST 클라이언트 라이브러리를 npm 설치합니다.

Bash

```
npm install @azure-rest/ai-document-intelligence
```

DocumentIntelligenceClient 만들기 및 인증

AAD(Azure Active Directory) 토큰 자격 증명 [을](#) 사용하려면 `@azure/ID` 라이브러리에서 가져온 원하는 자격 증명 형식의 instance 제공합니다.

AAD를 사용하여 인증하려면 먼저 `npm` 설치해야 합니다. [@azure/identity](#)

설치 후 사용할 [자격 증명](#) `@azure/identity` 유형을 선택할 수 있습니다. 예를 들어 `DefaultAzureCredential`을 [사용](#)하여 클라이언트를 인증할 수 있습니다.

AAD 애플리케이션의 클라이언트 ID, 테넌트 ID 및 클라이언트 암호 값을 환경 변수 (`AZURE_CLIENT_ID`, `AZURE_TENANT_ID`, `AZURE_CLIENT_SECRET`)

토큰 자격 증명 사용

ts

```
import DocumentIntelligence from "@azure-rest/ai-document-intelligence";

const client = DocumentIntelligence(
  process.env["DOCUMENT_INTELLIGENCE_ENDPOINT"],
  new DefaultAzureCredential()
);
```

API 키 사용

ts

```
import DocumentIntelligence from "@azure-rest/ai-document-intelligence";

const client =
  DocumentIntelligence(process.env["DOCUMENT_INTELLIGENCE_ENDPOINT"], {
    key: process.env["DOCUMENT_INTELLIGENCE_API_KEY"],
 });
```

문서 모델

미리 빌드된 레이아웃 분석(urlSource)

ts

```
const initialResponse = await client
  .path("/documentModels/{modelId}:analyze", "prebuilt-layout")
  .post({
    contentType: "application/json",
    body: {
```

```

    urlSource:
      "https://raw.githubusercontent.com/Azure/azure-sdk-for-
js/6704eff082aaaf2d97c1371a28461f512f8d748a/sdk/formrecognizer/ai-form-
recognizer/assets/forms/Invoice_1.pdf",
    },
    queryParameters: { locale: "en-IN" },
  });

```

미리 빌드된 레이아웃 분석(base64Source)

ts

```

import fs from "fs";
import path from "path";

const filePath = path.join(ASSET_PATH, "forms", "Invoice_1.pdf");
const base64Source = fs.readFileSync(filePath, { encoding: "base64" });
const initialResponse = await client
  .path("/documentModels/{modelId}:analyze", "prebuilt-layout")
  .post({
    contentType: "application/json",
    body: {
      base64Source,
    },
    queryParameters: { locale: "en-IN" },
  });

```

초기 응답에서 폴러 만들기 계속

ts

```

import {
  getLongRunningPoller,
  AnalyzeResultOperationOutput,
  isUnexpected,
} from "@azure-rest/ai-document-intelligence";

if (isUnexpected(initialResponse)) {
  throw initialResponse.body.error;
}
const poller = await getLongRunningPoller(client, initialResponse);
const result = (await poller.pollUntilDone()).body as
  AnalyzeResultOperationOutput;
console.log(result);
// {
//   status: 'succeeded',
//   createdDateTime: '2023-11-10T13:31:31Z',
//   lastUpdatedDateTime: '2023-11-10T13:31:34Z',
//   analyzeResult: {
//     apiVersion: '2023-10-31-preview',
//     .

```

```
//      .
//      .
//      contentFormat: 'text'
//  }
// }
```

Markdown 콘텐츠 형식

기본 일반 텍스트와 함께 Markdown 콘텐츠 형식의 출력을 지원합니다. 지금은 "미리 빌드된 레이아웃"에 대해서만 지원됩니다. Markdown 콘텐츠 형식은 채팅 또는 자동화 사용 시나리오에서 LLM 사용에 더 친숙한 형식으로 간주됩니다.

서비스는 Markdown 형식에 대한 [GFM](#) 사양(GitHub Flavored Markdown)을 따릅니다. 또한 결과 콘텐츠 형식을 나타내기 위해 값이 "text" 또는 "markdown"인 새 *contentFormat* 속성이 도입되었습니다.

ts

```
import DocumentIntelligence from "@azure-rest/ai-document-intelligence";
const client =
  DocumentIntelligence(process.env[ "DOCUMENT_INTELLIGENCE_ENDPOINT" ], {
    key: process.env[ "DOCUMENT_INTELLIGENCE_API_KEY" ],
  });

const initialResponse = await client
  .path("/documentModels/{modelId}:analyze", "prebuilt-layout")
  .post({
    contentType: "application/json",
    body: {
      urlSource:
        "https://raw.githubusercontent.com/Azure/azure-sdk-for-
js/6704eff082aaaf2d97c1371a28461f512f8d748a/sdk/formrecognizer/ai-form-
recognizer/assets/forms/Invoice_1.pdf",
    },
    queryParameters: { outputContentFormat: "markdown" }, // <-- new query
    parameter
  });

```

쿼리 필드

이 기능 플래그를 지정하면 서비스는 *queryFields* 쿼리 매개 변수를 통해 지정된 필드의 값을 추가로 추출하여 모델에 정의된 기존 필드를 대체로 보완합니다.

ts

```
await client.path("/documentModels/{modelId}:analyze", "prebuilt-
layout").post({
```

```
contentType: "application/json",
body: { urlSource: "..." },
queryParameters: {
  features: ["queryFields"],
  queryFields: ["NumberOfGuests", "StoreNumber"],
}, // <-- new query parameter
});
```

분할 옵션

이전 `@azure/ai-form-recognizer` 라이브러리에서 지원되는 이전 API 버전에서 문서 분할 및 분류 작업(`/documentClassifiers/{classifierId}:analyze`)은 항상 입력 파일을 여러 문서로 분할하려고 했습니다.

더 넓은 시나리오 집합을 사용하도록 설정하기 위해 서비스는 새로운 "2023-10-31-preview" 서비스 버전을 사용하는 "분할" 쿼리 매개 변수를 도입합니다. 지원되는 값은 다음과 같습니다.

- `split: "auto"`

서비스에서 분할할 위치를 결정하도록 합니다.

- `split: "none"`

전체 파일은 단일 문서로 처리됩니다. 분할이 수행되지 않습니다.

- `split: "perPage"`

각 페이지는 별도의 문서로 처리됩니다. 각 빈 페이지는 자체 문서로 유지됩니다.

문서 분류자 #Build

ts

```
import {
  DocumentClassifierBuildOperationDetailsOutput,
  getLongRunningPoller,
  isUnexpected,
} from "@azure-rest/ai-document-intelligence";

const containerSasUrl = (): string =>
  process.env["DOCUMENT_INTELLIGENCE_TRAINING_CONTAINER_SAS_URL"];
const initialResponse = await
  client.path("/documentClassifiers:build").post({
    body: {
      classifierId: `customClassifier${getRandomNumber()}`,
      description: "Custom classifier description",
    }
  })
  .withHeaders("Content-Type", "application/json")
  .withQuery("maxDegreeOfParallelism", "1");
const poller = getLongRunningPoller(
  initialResponse,
  (response) => response.value.state === "Succeeded"
);
```

```

docTypes: {
  foo: {
    azureBlobSource: {
      containerUrl: containerSasUrl(),
    },
  },
  bar: {
    azureBlobSource: {
      containerUrl: containerSasUrl(),
    },
  },
},
});

if (isUnexpected(initialResponse)) {
  throw initialResponse.body.error;
}
const poller = await getLongRunningPoller(client, initialResponse);
const response = (await poller.pollUntilDone())
  .body as DocumentClassifierBuildOperationDetailsOutput;
console.log(response);
// {
//   operationId: '31466834048_f3ee629e-73fb-48ab-993b-1d55d73ca460',
//   kind: 'documentClassifierBuild',
//   status: 'succeeded',
//   .
//   .
//   result: {
//     classifierId: 'customClassifier10978',
//     createdDateTime: '2023-11-09T12:45:56Z',
//     .
//     .
//     description: 'Custom classifier description'
//   },
//   apiVersion: '2023-10-31-preview'
// }

```

정보 가져오기

ts

```

const response = await client.path("/info").get();
if (isUnexpected(response)) {
  throw response.body.error;
}
console.log(response.body.customDocumentModels.limit);
// 20000

```

문서 모델 나열

ts

```
import { paginate } from "@azure-rest/ai-document-intelligence";
const response = await client.path("/documentModels").get();
if (isUnexpected(response)) {
    throw response.body.error;
}

const modelsInAccount: string[] = [];
for await (const model of paginate(client, response)) {
    console.log(model.modelId);
}
```

문제 해결

로깅

로깅을 사용하도록 설정하면 실패에 대한 유용한 정보를 파악하는 데 도움이 될 수 있습니다. HTTP 요청 및 응답 로그를 보려면 `AZURE_LOG_LEVEL` 환경 변수를 `info`로 설정합니다. 또는 `@azure/logger`에서 `setLogLevel`을 호출하여 런타임에 로깅을 사용하도록 설정할 수 있습니다.

JavaScript

```
const { setLogLevel } = require("@azure/logger");

setLogLevel("info");
```

로그를 사용하는 방법에 대한 자세한 내용은 [@azure/logger package docs](#)를 참조하세요.

GitHub에서 Microsoft와 공동 작업

이 콘텐츠의 원본은 GitHub에서 찾을 수 있으며, 여기서 문제와 끌어오기 요청을 만들고 검토할 수도 있습니다. 자세한 내용은 [참여자 가이드](#)를 참조하세요.



Azure SDK for JavaScript 피드백

Azure SDK for JavaScript은(는) 오픈 소스 프로젝트입니다. 다음 링크를 선택하여 피드백을 제공해 주세요.

설명서 문제 열기

☞ 제품 사용자 의견 제공

Python용 Azure AI Document Intelligence 클라이언트 라이브러리 - 버전 1.0.0b1

아티클 • 2023. 11. 22.

Azure AI 문서 인텔리전스([이전에는 Form Recognizer라고 했](#))는 기계 학습을 사용하여 문서에서 텍스트 및 구조화된 데이터를 분석하는 클라우드 서비스입니다. 여기에는 다음과 같은 기본 기능이 포함됩니다.

- 레이아웃 - 문서에서 콘텐츠 및 구조(예: 단어, 선택 표시, 테이블)를 추출합니다.
- 문서 - 문서의 일반 레이아웃 외에도 키-값 쌍을 분석합니다.
- 읽기 - 문서에서 페이지 정보를 읽습니다.
- 미리 빌드 - 미리 빌드된 모델을 사용하여 선택한 문서 유형(예: 영수증, 청구서, 명함, ID 문서, 미국 W-2 세금 문서 등)에서 공통 필드 값을 추출합니다.
- 사용자 지정 - 사용자 고유의 데이터에서 사용자 지정 모델을 빌드하여 문서에서 일반 레이아웃 외에도 맞춤형 필드 값을 추출합니다.
- 분류자 - 레이아웃 및 언어 기능을 결합하여 애플리케이션 내에서 처리하는 문서를 정확하게 검색하고 식별하는 사용자 지정 분류 모델을 빌드합니다.
- 추가 기능 - 바코드/QR 코드, 수식, 글꼴/스타일 등을 추출하거나 선택적 매개 변수가 있는 큰 문서에 대해 고해상도 모드를 사용하도록 설정합니다.

시작

패키지 설치

Bash

```
python -m pip install azure-ai-documentintelligence
```

필수 조건

- 이 패키지를 사용하려면 Python 3.7 이상이 필요합니다.
- 이 패키지를 사용하려면 [Azure 구독](#)이 필요합니다.
- 기존 Azure AI 문서 인텔리전스 instance.

Cognitive Services 또는 문서 인텔리전스 리소스 만들기

Document Intelligence는 [다중 서비스 및 단일 서비스 액세스를](#) 모두 지원합니다. 단일 엔드포인트/키에서 여러 Cognitive Services에 액세스하려는 경우 Cognitive Services 리소스를 만듭니다. 문서 인텔리전스 액세스에 대해서만 문서 인텔리전스 리소스를 만듭니다. [Azure Active Directory 인증](#)을 사용하려는 경우 단일 서비스 리소스가 필요합니다.

다음을 사용하여 리소스를 만들 수 있습니다.

- 옵션 1: [Azure Portal](#).
- 옵션 2: [Azure CLI](#).

다음은 CLI를 사용하여 문서 인텔리전스 리소스를 만드는 방법의 예입니다.

PowerShell

```
# Create a new resource group to hold the Document Intelligence resource
# if using an existing resource group, skip this step
az group create --name <your-resource-name> --location <location>
```

PowerShell

```
# Create the Document Intelligence resource
az cognitiveservices account create \
    --name <your-resource-name> \
    --resource-group <your-resource-group-name> \
    --kind FormRecognizer \
    --sku <sku> \
    --location <location> \
    --yes
```

리소스를 만드는 방법 또는 위치 및 sku 정보를 가져오는 방법에 대한 자세한 내용은 여기를 참조 [하세요](#).

클라이언트 인증

Document Intelligence 서비스와 상호 작용하려면 클라이언트의 instance 만들어야 합니다. 클라이언트 개체를 인스턴스화하려면 [엔드포인트](#) 및 [자격 증명](#)이 필요합니다.

엔드포인트 가져오기

[Azure Portal](#) 또는 [Azure CLI](#)를 사용하여 Document Intelligence 리소스에 대한 엔드포인트를 찾을 수 있습니다.

Bash

```
# Get the endpoint for the Document Intelligence resource
az cognitiveservices account show --name "resource-name" --resource-group
"resource-group-name" --query "properties.endpoint"
```

지역 엔드포인트 또는 사용자 지정 하위 도메인을 인증에 사용할 수 있습니다. 형식은 다음과 같이 지정됩니다.

```
Regional endpoint: https://<region>.api.cognitive.microsoft.com/
Custom subdomain: https://<resource-name>.cognitiveservices.azure.com/
```

지역 엔드포인트는 지역의 모든 리소스에 대해 동일합니다. 지원되는 지역 엔드포인트의 전체 목록은 [여기에서](#) 참조할 수 있습니다. 지역 엔드포인트는 AAD 인증을 지원하지 않습니다.

반면에 사용자 지정 하위 도메인은 Document Intelligence 리소스에 고유한 이름입니다. 단일 서비스 리소스 [에서만](#) 사용할 수 있습니다.

API 키 가져오기

API 키는 [Azure Portal](#) 에서 또는 다음 Azure CLI 명령을 실행하여 찾을 수 있습니다.

Bash

```
az cognitiveservices account keys list --name "<resource-name>" --resource-
group "<resource-group-name>"
```

AzureKeyCredential을 사용하여 클라이언트 만들기

API 키를 매개 변수로 `credential` 사용하려면 [AzureKeyCredential](#) 의 instance 키를 문자열로 전달합니다.

Python

```
from azure.core.credentials import AzureKeyCredential
from azure.ai.documentintelligence import DocumentIntelligenceClient

endpoint = "https://<my-custom-subdomain>.cognitiveservices.azure.com/"
credential = AzureKeyCredential("<api_key>")
document_analysis_client = DocumentIntelligenceClient(endpoint, credential)
```

Azure Active Directory 자격 증명을 사용하여 클라이언트 만들기

`AzureKeyCredential` 인증은 이 시작 가이드의 예제에서 사용되지만 [azure-identity](#) 라이브러리를 사용하여 Azure Active Directory로 인증할 수도 있습니다. 지역 엔드포인트는 AAD 인증을 지원하지 않습니다. 이러한 유형의 인증을 사용하기 위해 리소스에 대한 [사용자 지정 하위 도메인](#) 이름을 만듭니다.

아래에 표시된 `DefaultAzureCredential` 형식 또는 Azure SDK와 함께 제공되는 기타 자격 증명 형식을 사용하려면 패키지를 설치 `azure-identity` 하세요.

```
pip install azure-identity
```

또한 [새 AAD 애플리케이션을 등록하고](#) 서비스 주체에게 역할을 할당하여 `"Cognitive Services User"` 문서 인텔리전스에 대한 액세스 권한을 부여해야 합니다.

완료되면 AAD 애플리케이션의 클라이언트 ID, 테넌트 ID 및 클라이언트 암호 값을 환경 변수 `AZURE_CLIENT_ID`, `AZURE_TENANT_ID`로 `AZURE_CLIENT_SECRET` 설정합니다.

Python

```
"""DefaultAzureCredential will use the values from these environment
variables: AZURE_CLIENT_ID, AZURE_TENANT_ID, AZURE_CLIENT_SECRET
"""
from azure.ai.documentintelligence import DocumentIntelligenceClient
from azure.identity import DefaultAzureCredential

endpoint = os.environ["DOCUMENTINTELLIGENCE_ENDPOINT"]
credential = DefaultAzureCredential()

document_analysis_client = DocumentIntelligenceClient(endpoint, credential)
```

주요 개념

DocumentIntelligenceClient

`DocumentIntelligenceClient` 는 API를 통해 미리 빌드된 사용자 지정 모델을 사용하여 입력 문서를 분석하기 `begin_analyze_document` 위한 작업을 제공합니다. 매개 변수를 `model_id` 사용하여 분석할 모델 유형을 선택합니다. 지원되는 모델의 전체 목록은 여기를 참조 [하세요](#). 또한는 `DocumentIntelligenceClient` API를 통해 `begin_classify_document` 문서를 분류하기 위한 작업을 제공합니다. 사용자 지정 분류 모델은 입력 파일의 각 페이지를 분류하여 문서를 식별할 수 있으며, 입력 파일 내에서 여러 문서 또는 단일 문서의 여러 인스턴스를 식별할 수도 있습니다.

여기에서 DocumentIntelligenceClient 예제를 사용하는 방법을 설명하기 위해 샘플 코드 조각 제공됩니다. 지원되는 기능, 로캘 및 문서 형식을 포함한 문서 분석에 대한 자세한 내용은 [서비스 설명서](#)에서 확인할 수 있습니다.

DocumentIntelligenceAdministrationClient

DocumentIntelligenceAdministrationClient는 다음에 대한 작업을 제공합니다.

- 사용자 지정 문서에 레이블을 지정하여 지정한 특정 필드를 분석하는 사용자 지정 모델을 빌드합니다. 모델이 분석할 수 있는 문서 유형과 각 필드에 대한 예상 신뢰도를 나타내는가 `DocumentModelDetails` 반환됩니다. 자세한 설명은 [서비스 설명서](#)를 참조하세요.
- 기존 모델 컬렉션에서 구성된 모델을 만듭니다.
- 계정에서 생성된 모델을 관리합니다.
- 작업을 나열하거나 지난 24시간 이내에 만든 특정 모델 작업을 가져옵니다.
- 한 문서 인텔리전스 리소스에서 다른 리소스로 사용자 지정 모델을 복사합니다.
- 사용자 지정 분류 모델을 빌드하고 관리하여 애플리케이션 내에서 처리하는 문서를 분류합니다.

[Document Intelligence Studio](#)와 같은 그래픽 사용자 인터페이스를 사용하여 모델을 빌드할 수도 있습니다.

여기에서 DocumentIntelligenceAdministrationClient 예제를 사용하는 방법을 설명하기 위해 샘플 코드 조각 제공됩니다.

장기 실행 작업

장기 실행 작업은 작업을 시작하기 위해 서비스로 전송된 초기 요청으로 구성된 작업이며, 그 다음에는 간격으로 서비스를 폴링하여 작업이 완료되었는지 또는 실패했는지, 성공했는지 여부를 확인하여 결과를 가져옵니다.

문서를 분석하거나, 모델을 빌드하거나, 모델을 복사/작성하는 메서드는 장기 실행 작업으로 모델링됩니다. 클라이언트는 또는 `AsyncLROPoller`를 `begin_<method-name>` 반환하는 메서드를 `LROPoller` 노출합니다. 호출자는 메서드에서 `begin_<method-name>` 반환된 폴러 개체를 호출 `result()` 하여 작업이 완료되기를 기다려야 합니다. 샘플 코드 조각은 아래의 장기 실행 작업 를 사용하여 설명하기 위해 제공됩니다.

예제

다음 섹션에서는 다음을 포함하여 가장 일반적인 문서 인텔리전스 작업을 다루는 몇 가지 코드 조각을 제공합니다.

- 레이아웃 추출
- 일반 문서 모델 사용
- 미리 빌드된 모델 사용
- 사용자 지정 모델 빌드
- 사용자 지정 모델을 사용하여 문서 분석
- 모델 관리
- 추가 기능

레이아웃 추출

문서에서 경계 영역 좌표와 함께 텍스트, 선택 표시, 텍스트 스타일 및 테이블 구조를 추출합니다.

Python

```
from azure.core.credentials import AzureKeyCredential
from azure.ai.documentintelligence import DocumentIntelligenceClient

endpoint = os.environ["DOCUMENTINTELLIGENCE_ENDPOINT"]
key = os.environ["DOCUMENTINTELLIGENCE_API_KEY"]

document_intelligence_client = DocumentIntelligenceClient(
    endpoint=endpoint, credential=AzureKeyCredential(key)
)
with open(path_to_sample_documents, "rb") as f:
    poller = document_intelligence_client.begin_analyze_document(
        "prebuilt-layout", analyze_request=f,
        content_type="application/octet-stream"
    )
result = poller.result()

for idx, style in enumerate(result.styles):
    print(
        "Document contains {} content".format(
            "handwritten" if style.is_handwritten else "no handwritten"
        )
    )

for page in result.pages:
    print("----Analyzing layout from page #{}----".format(page.page_number))
    print(
        "Page has width: {} and height: {}, measured with unit: {}".format(
            page.width, page.height, page.unit
        )
    )

    for line_idx, line in enumerate(page.lines):
        words = line.get_words()
        print(
            "...Line # {} has word count {} and text '{}' within bounding"
        )
```

```
        polygon '{}'.format(
            line_idx,
            len(words),
            line.content,
            line.polygon,
        )
    )

    for word in words:
        print(
            ".....Word '{}' has a confidence of {}".format(
                word.content, word.confidence
            )
        )

for selection_mark in page.selection_marks:
    print(
        "...Selection mark is '{}' within bounding polygon '{}' and has
a confidence of {}".format(
            selection_mark.state,
            selection_mark.polygon,
            selection_mark.confidence,
        )
    )

for table_idx, table in enumerate(result.tables):
    print(
        "Table # {} has {} rows and {} columns".format(
            table_idx, table.row_count, table.column_count
        )
    )
    for region in table.bounding_regions:
        print(
            "Table # {} location on page: {} is {}".format(
                table_idx,
                region.page_number,
                region.polygon,
            )
        )
    for cell in table.cells:
        print(
            "...Cell[{}][{}] has content '{}'".format(
                cell.row_index,
                cell.column_index,
                cell.content,
            )
        )
        for region in cell.bounding_regions:
            print(
                "...content on page {} is within bounding polygon
'{}'.format(
                    region.page_number,
                    region.polygon,
                )
            )
    
```

```
print("-----")
```

미리 빌드된 모델 사용

문서 인텔리전스 서비스에서 제공하는 미리 빌드된 모델을 사용하여 영수증, 청구서, 명함, ID 문서 및 미국 W-2 세금 문서와 같은 일부 문서 유형에서 필드를 추출합니다.

예를 들어 판매 영수증의 필드를 분석하려면 메서드에 전달 `model_id="prebuilt-receipt"` `begin_analyze_document` 하여 제공된 미리 빌드된 영수증 모델을 사용합니다.

Python

```
from azure.core.credentials import AzureKeyCredential
from azure.ai.documentintelligence import DocumentIntelligenceClient

endpoint = os.environ["DOCUMENTINTELLIGENCE_ENDPOINT"]
key = os.environ["DOCUMENTINTELLIGENCE_API_KEY"]

document_analysis_client = DocumentIntelligenceClient(endpoint=endpoint,
credential=AzureKeyCredential(key))
with open(path_to_sample_documents, "rb") as f:
    poller = document_analysis_client.begin_analyze_document(
        "prebuilt-receipt", analyze_request=f, locale="en-US",
        content_type="application/octet-stream")
    )
receipts = poller.result()

for idx, receipt in enumerate(receipts.documents):
    print(f"-----Analysis of receipt #{idx + 1}-----")
    print(f"Receipt type: {receipt.doc_type if receipt.doc_type else 'N/A'}")
    merchant_name = receipt.fields.get("MerchantName")
    if merchant_name:
        print(f"Merchant Name: {merchant_name.get('valueString')} has
confidence: " f"{merchant_name.confidence}")
        transaction_date = receipt.fields.get("TransactionDate")
        if transaction_date:
            print(
                f"Transaction Date: {transaction_date.get('valueDate')} has
confidence: "
                f"{transaction_date.confidence}"
            )
    if receipt.fields.get("Items"):
        print("Receipt items:")
        for idx, item in
enumerate(receipt.fields.get("Items").get("valueArray")):
            print(f"...Item #{idx + 1}")
            item_description = item.get("valueObject").get("Description")
            if item_description:
                print(
```

```

        f".....Item Description:
{item_description.get('valueString')} has confidence: "
            f"{item_description.confidence}"
        )
        item_quantity = item.get("valueObject").get("Quantity")
        if item_quantity:
            print(
                f".....Item Quantity:
{item_quantity.get('valueString')} has confidence: "
                    f"{item_quantity.confidence}"
            )
        item_total_price = item.get("valueObject").get("TotalPrice")
        if item_total_price:
            print(
                f".....Total Item Price:
{format_price(item_total_price.get('valueCurrency'))} has confidence: "
                    f"{item_total_price.confidence}"
            )
        subtotal = receipt.fields.get("Subtotal")
        if subtotal:
            print(f"Subtotal: {format_price(subtotal.get('valueCurrency'))} has
confidence: {subtotal.confidence}")
        tax = receipt.fields.get("TotalTax")
        if tax:
            print(f"Total tax: {format_price(tax.get('valueCurrency'))} has
confidence: {tax.confidence}")
        tip = receipt.fields.get("Tip")
        if tip:
            print(f"Tip: {format_price(tip.get('valueCurrency'))} has
confidence: {tip.confidence}")
        total = receipt.fields.get("Total")
        if total:
            print(f"Total: {format_price(total.get('valueCurrency'))} has
confidence: {total.confidence}")
        print("-----")
    
```

당신은 영수증에 국한되지 않습니다! 미리 빌드된 몇 가지 모델 중에서 선택할 수 있으며, 각 모델에는 지원되는 필드 집합이 있습니다. 여기에서 지원되는 다른 미리 빌드된 모델을 참조 [하세요](#).

사용자 지정 모델 빌드

사용자 고유의 문서 형식으로 사용자 지정 모델을 빌드합니다. 결과 모델을 사용하여 학습된 문서 형식의 값을 분석할 수 있습니다. 학습 문서를 저장하는 Azure Storage Blob 컨테이너에 컨테이너 SAS URL을 제공합니다.

컨테이너 및 필수 파일 구조 설정에 대한 자세한 내용은 [서비스 설명서](#)에서 확인할 수 있습니다.

```

from azure.ai.formrecognizer import (
    DocumentIntelligenceAdministrationClient,
    ModelBuildMode,
)
from azure.core.credentials import AzureKeyCredential

endpoint = os.environ["DOCUMENTINTELLIGENCE_ENDPOINT"]
key = os.environ["DOCUMENTINTELLIGENCE_API_KEY"]
container_sas_url = os.environ["CONTAINER_SAS_URL"]

document_model_admin_client = DocumentIntelligenceAdministrationClient(
    endpoint, AzureKeyCredential(key)
)
poller = document_model_admin_client.begin_build_document_model(
    ModelBuildMode.TEMPLATE,
    blob_container_url=container_sas_url,
    description="my model description",
)
model = poller.result()

print(f"Model ID: {model.model_id}")
print(f"Description: {model.description}")
print(f"Model created on: {model.created_on}")
print(f"Model expires on: {model.expires_on}")
print("Doc types the model can recognize:")
for name, doc_type in model.doc_types.items():
    print(
        f"Doc Type: '{name}' built with '{doc_type.build_mode}' mode which
has the following fields:"
    )
    for field_name, field in doc_type.field_schema.items():
        print(
            f"Field: '{field_name}' has type '{field['type']}' and
confidence score "
            f"{doc_type.field_confidence[field_name]}"
        )
)

```

사용자 지정 모델을 사용하여 문서 분석

문서 필드, 테이블, 선택 표시 등을 분석합니다. 이러한 모델은 사용자 고유의 데이터로 학습되므로 문서에 맞게 조정됩니다. 최상의 결과를 위해 사용자 지정 모델이 빌드된 것과 동일한 문서 형식의 문서만 분석해야 합니다.

Python

```

from azure.core.credentials import AzureKeyCredential
from azure.ai.documentintelligence import DocumentIntelligenceClient

endpoint = os.environ["DOCUMENTINTELLIGENCE_ENDPOINT"]
key = os.environ["DOCUMENTINTELLIGENCE_API_KEY"]

```

```
model_id = os.getenv("CUSTOM_BUILT_MODEL_ID", custom_model_id)

document_analysis_client = DocumentIntelligenceClient(endpoint=endpoint,
credential=AzureKeyCredential(key))

# Make sure your document's type is included in the list of document types
# the custom model can analyze
with open(path_to_sample_documents, "rb") as f:
    poller = document_analysis_client.begin_analyze_document(
        model_id=model_id, analyze_request=f,
        content_type="application/octet-stream"
    )
result = poller.result()

for idx, document in enumerate(result.documents):
    print(f"-----Analyzing document #{idx + 1}-----")
    print(f"Document has type {document.doc_type}")
    print(f"Document has document type confidence {document.confidence}")
    print(f"Document was analyzed with model with ID {result.model_id}")
    for name, field in document.fields.items():
        field_value = field.get("valueString") if field.get("valueString")
else field.content
        print(
            f".....found field of type '{field.type}' with value
'{field_value}' and with confidence {field.confidence}"
        )

# iterate over tables, lines, and selection marks on each page
for page in result.pages:
    print(f"\nLines found on page {page.page_number}")
    for line in page.lines:
        print(f"...Line '{line.content}'")
    for word in page.words:
        print(f"...Word '{word.content}' has a confidence of
{word.confidence}")
    if page.selection_marks:
        print(f"\nSelection marks found on page {page.page_number}")
        for selection_mark in page.selection_marks:
            print(
                f"...Selection mark is '{selection_mark.state}' and has a
confidence of {selection_mark.confidence}"
            )

for i, table in enumerate(result.tables):
    print(f"\nTable {i + 1} can be found on page:")
    for region in table.bounding_regions:
        print(f"...{region.page_number}")
    for cell in table.cells:
        print(f"...Cell[{cell.row_index}][{cell.column_index}] has text
'{cell.content}'")
    print("-----")
```

또한 문서 URL을 사용하여 메서드를 사용하여 `begin_analyze_document` 문서를 분석할 수도 있습니다.

Python

```
from azure.core.credentials import AzureKeyCredential
from azure.ai.documentintelligence import DocumentIntelligenceClient
from azure.ai.documentintelligence.models import AnalyzeDocumentRequest

endpoint = os.environ["DOCUMENTINTELLIGENCE_ENDPOINT"]
key = os.environ["DOCUMENTINTELLIGENCE_API_KEY"]

document_analysis_client = DocumentIntelligenceClient(endpoint=endpoint,
credential=AzureKeyCredential(key))
url = "https://raw.githubusercontent.com/Azure/azure-sdk-for-
python/main/sdk/documentintelligence/azure-ai-
documentintelligence/tests/sample_forms/receipt/contoso-receipt.png"
poller = document_analysis_client.begin_analyze_document("prebuilt-receipt",
AnalyzeDocumentRequest(url_source=url))
receipts = poller.result()
```

모델 관리

계정에 연결된 사용자 지정 모델을 관리합니다.

Python

```
from azure.ai.documentintelligence import
DocumentIntelligenceAdministrationClient
from azure.core.credentials import AzureKeyCredential
from azure.core.exceptions import ResourceNotFoundError

endpoint = "https://<my-custom-subdomain>.cognitiveservices.azure.com/"
credential = AzureKeyCredential("<api_key>")

document_model_admin_client =
DocumentIntelligenceAdministrationClient(endpoint, credential)

account_details = document_model_admin_client.get_resource_info()
print("Our account has {} custom models, and we can have at most {} custom
models".format(
    account_details.custom_document_models.count,
    account_details.custom_document_models.limit
))

# Here we get a paged list of all of our models
models = document_model_admin_client.list_models()
print("We have models with the following ids: {}".format(
    ", ".join([m.model_id for m in models])
))
```

```

# Replace with the custom model ID from the "Build a model" sample
model_id = "<model_id from the Build a Model sample>"

custom_model = document_model_admin_client.get_model(model_id=model_id)
print("Model ID: {}".format(custom_model.model_id))
print("Description: {}".format(custom_model.description))
print("Model created on: {}\\n".format(custom_model.created_on))

# Finally, we will delete this model by ID
document_model_admin_client.delete_model(model_id=custom_model.model_id)

try:
    document_model_admin_client.get_model(model_id=custom_model.model_id)
except ResourceNotFoundError:
    print("Successfully deleted model with id
{}".format(custom_model.model_id))

```

추가 기능

Document Intelligence는 보다 정교한 분석 기능을 지원합니다. 이러한 선택적 기능은 문서 추출 시나리오에 따라 사용하도록 설정하고 사용하지 않도록 설정할 수 있습니다.

다음 추가 기능 기능은 2023-07-31(GA) 이상 릴리스에 사용할 수 있습니다.

- [barcode/QR 코드 ↗](#)
- [수식 ↗](#)
- [글꼴/스타일 ↗](#)
- [고해상도 모드 ↗](#)
- [language ↗](#)

일부 추가 기능에서는 추가 요금이 발생합니다. 가격 책정을

<https://azure.microsoft.com/pricing/details/ai-document-intelligence/> 참조하세요.

문제 해결

일반

문서 인텔리전스 클라이언트 라이브러리는 [Azure Core ↗](#)에 정의된 예외를 발생합니다. 문서 인텔리전스 서비스에서 발생한 오류 코드 및 메시지는 [서비스 설명서 ↗](#)에서 찾을 수 있습니다.

로깅

이 라이브러리는 로깅에 표준 [로깅 ↗](#) 라이브러리를 사용합니다.

HTTP 세션(URL, 헤더 등)에 대한 기본 정보는 수준에서 기록됩니다 [INFO](#).

DEBUG 요청/응답 본문 및 **수정되지 않은** 헤더를 포함한 자세한 수준 로깅은 클라이언트에서 또는 키워드(keyword) 인수를 `logging_enable` 사용하여 작업별로 사용하도록 설정할 수 있습니다.

여기에 예제가 포함된 전체 SDK 로깅 설명서를 참조 [하세요](#).

선택적 구성

선택적 키워드(keyword) 인수는 클라이언트 및 작업별 수준에서 전달할 수 있습니다. [azure-core 참조 설명서](#) 에서는 재시도, 로깅, 전송 프로토콜 등에 사용할 수 있는 구성에 대해 설명합니다.

다음 단계

추가 샘플 코드

Document Intelligence Python API에서 사용되는 일반적인 패턴을 보여 주는 몇 가지 코드 조각은 [샘플 추가 정보를](#) 참조하세요.

추가 설명서

Azure AI 문서 인텔리전스에 대한 자세한 설명서는 [docs.microsoft.com 대한 문서 인텔리전스 설명서를](#) 참조하세요.

참여

이 프로젝트에 대한 기여와 제안을 환영합니다. 대부분의 경우 기여하려면 권한을 부여하며 실제로 기여를 사용할 권한을 당사에 부여한다고 선언하는 CLA(기여자 라이선스 계약)에 동의해야 합니다. 자세한 내용은 <https://cla.microsoft.com> 을 참조하세요.

끌어오기 요청을 제출하면 CLA-bot은 CLA를 제공하고 PR을 적절하게 데코레이팅해야 하는지 여부를 자동으로 결정합니다(예: 레이블, 설명). 봇에서 제공하는 지침을 따르기만 하면 됩니다. 이 작업은 CLA를 사용하여 모든 리포지토리에서 한 번만 수행하면 됩니다.

이 프로젝트에는 [Microsoft Open Source Code of Conduct](#) (Microsoft 오픈 소스 준수 사항)가 적용됩니다. 자세한 내용은 행동 강령 FAQ를 참조하거나 추가 질문이나 의견을 문의 opencode@microsoft.com 하세요.

GitHub에서 Microsoft와 공동 작업

이 콘텐츠의 원본은 GitHub에서 찾을 수 있으며, 여기서 문제와 끌어오기 요청을 만들고 검토할 수도 있습니다. 자세한 내용은 [참여자 가이드](#)를 참조하세요.



Azure SDK for Python 피드백

Azure SDK for Python은(는) 오픈 소스 프로젝트입니다. 다음 링크를 선택하여 피드백을 제공해 주세요.

 설명서 문제 열기

 제품 사용자 의견 제공

Azure AI 서비스 지원 및 도움말 옵션

아티클 • 2024. 02. 22.

이제 막 Azure AI 서비스의 기능을 살펴보기 시작하셨나요? 애플리케이션에 새로운 기능을 구현하는 경우를 가정해 볼 수 있습니다. 또는 서비스를 사용한 후 개선 방법에 대한 제안이 있나요? Azure AI 서비스에 대한 지원을 받고, 최신 상태를 유지하고, 피드백을 제공하고, 버그를 보고할 수 있는 옵션은 다음과 같습니다.

Azure 지원 요청 만들기

A

지금 클라우드를 시작하려는 개발자든 비즈니스에 중요한 전략적 애플리케이션을 배포하려는 대규모 조직이든 관계없이 가장 적합한 [Azure 지원 옵션 및 플랜 선택](#)의 범위를 탐색합니다. Azure 고객은 Azure Portal에서 지원 요청을 만들고 관리할 수 있습니다.

- [Azure Portal](#)
- [미국 정부의 Azure Portal](#)

Microsoft Q&A에 질문 게시

Microsoft 엔지니어, Azure MVP(가장 귀중한 전문가) 또는 전문가 커뮤니티의 기술 제품 관련 질문에 대한 빠르고 안정적인 답변을 얻으려면 Azure가 커뮤니티 지원을 위해 선호하는 대상인 [Microsoft Q&A](#)에 참여하세요.

검색을 사용하여 문제에 대한 답변을 찾을 수 없으면 Microsoft Q&A에 새 질문을 제출합니다. 질문을 할 때 다음 태그 중 하나를 사용합니다.

- [Azure AI 서비스](#)

비전

- [Azure AI Vision](#)
- [Custom Vision](#)
- [Face](#)
- [문서 인텔리전스](#)
- [Video Indexer](#)

언어

- [Immersive Reader](#)
- [언어 이해\(LUIS\)](#)

- QnA Maker
- 언어 서비스
- Translator

음성

- Speech Service

의사 결정

- Anomaly Detector
- Content Moderator
- Metrics Advisor
- Personalizer

Azure OpenAI

- Azure OpenAI

Stack Overflow에 질문을 게시합니다.



가장 큰 커뮤니티 개발자 에코시스템의 개발자 질문에 대한 답변을 보려면 Stack Overflow에서 질문하세요.

Stack Overflow에 새 질문을 제출하는 경우 질문을 만들 때 다음 태그 중 하나 이상을 사용하세요.

- Azure AI 서비스 ↗

비전

- Azure AI Vision ↗
- Custom Vision ↗
- Face ↗
- 문서 인텔리전스 ↗
- Video Indexer ↗

언어

- Immersive Reader ↗
- 언어 이해(LUIS) ↗
- QnA Maker ↗
- 언어 서비스 ↗

- Translator ↗

음성

- Speech Service ↗

의사 결정

- Anomaly Detector ↗
- Content Moderator ↗
- Metrics Advisor ↗
- Personalizer ↗

Azure OpenAI

- Azure OpenAI ↗

피드백 제출

새로운 기능을 요청하려면 <https://feedback.azure.com> 에 게시합니다. Azure AI 서비스 와 해당 API가 개발하는 애플리케이션에 더 잘 작동하도록 만들기 위한 아이디어를 공유하세요.

- Azure AI 서비스 ↗

비전

- Azure AI Vision ↗
- Custom Vision ↗
- Face ↗
- 문서 인텔리전스 ↗
- Video Indexer ↗

언어

- Immersive Reader ↗
- 언어 이해(LUIS) ↗
- QnA Maker ↗
- 언어 서비스 ↗
- Translator ↗

음성

- Speech Service ↗

의사 결정

- [Anomaly Detector ↗](#)
- [Content Moderator ↗](#)
- [Metrics Advisor ↗](#)
- [Personalizer ↗](#)

최신 소식 수신

새 릴리스의 기능이나 Azure 블로그의 뉴스에 대한 최신 정보를 얻으면 프로그래밍 오류, 서비스 버그 또는 아직 Azure AI 서비스에서 사용할 수 없는 기능 간의 차이점을 찾는데 도움이 될 수 있습니다.

- [Azure 업데이트 ↗](#)에서 제품 업데이트, 로드맵 및 공지 사항에 대해 자세히 알아봅니다.
- Azure AI 서비스에 대한 소식은 [Azure 블로그 ↗](#)에서 공유됩니다.
- Azure AI 서비스에 대한 [Reddit 대화에 참여하세요 ↗](#).

다음 단계

[Azure AI 서비스란?](#)