

Process Documents with Python Using the Document AI API

GSP925



- The Document AI API transforms unstructured documents and emails into structured data for easier analysis and consumption.
- This lab uses Python in a Vertex AI Workbench instance with JupyterLab to create processors like general form and Document OCR types.
- You'll enable the Document AI API and set up the required processors for document parsing.
- Install the Python client library within the Workbench to enable API interactions.
- Practice synchronous calls for single scanned forms and asynchronous calls for batch processing.

Free AI Paraphrasin... Transcript of Case St... GCP-LAB Hydra datawarehouse Products

Google Skills Partner

6425 3

Automate Data Capture at Scale with Document AI > Process Documents with Python Using the Document AI API

Contents

Process Documents with Python Using the Docur ▾

gcloud is the command-line tool for Google Cloud. It comes pre-installed on Cloud Shell and supports tab-completion.

3. (Optional) You can list the active account name with this command:

```
gcloud auth list
```

4. Click Authorize.

Output:

```
ACTIVE: *
ACCOUNT: "ACCOUNT"

To set the active account, run:
$ gcloud config set account `ACCOUNT`
```

5. (Optional) You can list the project ID with this command:

```
gcloud config list project
```

Task 3. Make a synchronous process document request 0/100

Task 4. Run the synchronous Document AI Python code

Task 5. Create a Document AI Document OCR processor

Task 6. Prepare your environment for asynchronous Document AI API calls

Task 7. Make an asynchronous process document request

Congratulations

< Previous

Next >



Dashboard – qwiklabs-gcp-03-d x +

https://console.cloud.google.com/home/dashboard?project=qwiklabs-gcp-03-d5494d0f455b&pli=1&cloudshell=true

Free AI Paraphrasing... Transcript of Case St... GCP-LAB Hydra datawarehouse Products

Google Cloud qwiklabs-gcp-03-d5494d0f455b Search (/) for resources, docs, products, and more Search

CLOUD SHELL Terminal (qwiklabs-gcp-03-d5494d0f455b) x + Open Editor

Gemini CLI is available in Cloud Shell terminal! Type gemini to try it. [Learn more](#) Don't show again Dismiss

Welcome to Cloud Shell! Type "help" to get started, or type "gemini" to try prompting with Gemini CLI.
Your Cloud Platform project in this session is set to **qwiklabs-gcp-03-d5494d0f455b**.
Use `gcloud config set project [PROJECT_ID]` to change to a different project.
student_01_75364ecd3f8c@cloudshell:~ (qwiklabs-gcp-03-d5494d0f455b)\$ gcloud auth list
Credentialed Accounts

ACTIVE: *
ACCOUNT: student-01-75364ecd3f8c@qwiklabs.net

To set the active account, run:
\$ gcloud config set account `ACCOUNT`

student_01_75364ecd3f8c@cloudshell:~ (qwiklabs-gcp-03-d5494d0f455b)\$ gcloud config list project
[core]
project = qwiklabs-gcp-03-d5494d0f455b

Your active configuration is: [cloudshell-8787]
student_01_75364ecd3f8c@cloudshell:~ (qwiklabs-gcp-03-d5494d0f455b)\$

15°C Clear

Search

06:06 PM 30-12-2025 ENG IN

Free AI Paraphrasin... Transcript of Case St... GCP-LAB Hydra datawarehouse Products

Google Skills Partner

Contents

Process Documents with Python Using the Docur ▾

Lab setup instructions and requirements

Protect your account and

End Lab 01:28:16 ? Time limit

Caution: When you are in the console, do not deviate from the lab instructions. Doing so may cause your account to be blocked. Learn more.

Open Google Cloud console

Username: student-01-75364ecd3f8c1

Password: YFyxc227VWPX

Project ID: qwiklabs-gcp-03-d5494d01

Task 1. Create and test a general form processor

In this task you will enable the Document AI API and create and test a general form processor. The general form processor will process any type of document and extract all the text content it can identify in the document. It is not limited to printed text, it can handle handwritten text and text in any orientation, supports a number of languages, and understands how form data elements are related to each other so that you can extract key:value pairs for form fields that have text labels.

Enable the Cloud Document AI API

Before you can begin using Document AI, you must enable the API.

1. In Cloud Console, from the **Navigation menu** (≡), click **APIs & services** > **Library**.
2. Search for **Cloud Document AI API**, then click the **Enable** button to use the API in your Google Cloud project.

If the Cloud Document AI API is already enabled you will see the **Manage** button and

Next >

API API/Service Details – APIs & Services

https://console.cloud.google.com/apis/api/documentai.googleapis.com/metrics?project=qwiklabs-gcp-03-d5494d0f455b

Free AI Paraphrasing... Transcript of Case Study GCP-LAB Hydra datawarehouse Products

Google Cloud qwiklabs-gcp-03-d5494d0f455b Search (/) for resources, docs, products, and more Search

API APIs & Services Enabled APIs & services

Cloud Document AI API

Service to parse structured information from unstructured or semi-structured documents using state-of-the-art Google AI such as natural language, computer vision, translation, and AutoML.

By Google Enterprise API

Service name: documentai.googleapis.com Type: Public API Status: Enabled Documentation: Learn more Explore: Try in API Explorer

Metrics Quotas & System Limits Credentials Cost

7 15°C Clear

Search

06:07 PM 30-12-2025 ENG IN

Dashboard Catalog Paths Collections

Contents

Process Documents with Python Using the Docur ▾

Create a general form processor

Create a Document AI processor using the Document AI form parser.

1. In the console, on the **Navigation menu** (≡), click **Document AI > Overview**.

2. Click **Explore processors** and click **Create Processor for Form Parser**, which is a type of general processor.

3. Specify the processor name as **form-parser** and select the region **US (United States)** from the list.

4. Click **Create** to create the general form-parser processor.

This will create the processor and return to the processor details page that will display the processor ID, status, and the prediction endpoint.

5. Make a note of the Processor ID as you will need to update variables in JupyterLab notebooks with the Processor ID in later tasks.

Task 3. Make a synchronous process document request 0/100

Task 4. Run the synchronous Document AI Python code

Task 5. Create a Document AI Document OCR processor

Task 6. Prepare your environment for asynchronous Document AI API calls

Task 7. Make an asynchronous process document request

Congratulations

Task 2. Configure your Vertex AI Workbench

< Previous

Next >



Processor gallery – Document AI

https://console.cloud.google.com/ai/document-ai/processor-library?project=qwiklabs-gcp-03-d5494d0f455b

Free AI Paraphrasing... Transcript of Case Study GCP-LAB Hydra datawarehouse Products

Google Cloud qwiklabs-gcp-03-d5494d0f455b document ai

Document AI / Processor gallery

Overview

Processors

My processors

Processor gallery

Custom Processors

Monitoring

Capacity Reservation

Processor gallery

Search processors

Ready to use out-of-the-box processors for general document goals.

Features

Trainable 9

Type

- General 3
- Specialized 12

Access status

- Public 17
- Private 2

Region

- eu 18

Document OCR

Identify and extract text in different types of documents

Form Parser

Extract form elements such as checkboxes

Specialized

Schematized processors for domain-specific documents.

Bank Statement Parser

Extract from Bank Statements

Expense Parser

Extracts from Receipts, invoices, bills, and expense reports

Create processor

Form Parser

Extract text and spatial structure from documents, including tabular content through OCR [Learn more](#)

Processor name * Must start with a letter. Can use letters, numbers, spaces, dashes, and underscores.

Region

Advanced options

Create Cancel

7 15°C Clear

Search

Eng IN 06:08 PM 30-12-2025

Contents

Process Documents with Python Using the Docur ▾

Lab setup instructions and requirements

Protect your account and

End Lab 01:25:59 ? Time limit

Caution: When you are in the console, do not deviate from the lab instructions. Doing so may cause your account to be blocked. [Learn more.](#)

[Open Google Cloud console](#)

Username: student-01-75364ecd3f8c4

Password: YFyxc227VWPX

Project ID: qwiklabs-gcp-03-d5494d01

Automate Data Capture at Scale with Document AI > Process Documents with Python Using the Document AI API

Task 2. Configure your Vertex AI Workbench instance to perform Document AI API calls

Next, connect to JupyterLab running on the Vertex AI Workbench instance that was created for you when the lab was started, then configure that environment for the remaining lab tasks.

1. In the Google Cloud console, on the **Navigation menu** (≡), click **Vertex AI > Workbench**.
2. Find the `jupyterlab` instance and click on the **Open JupyterLab** button.

The JupyterLab interface for your Workbench instance opens in a new browser tab.

3. Click **Terminal** to open a terminal shell inside the Vertex AI Workbench instance.
4. Enter the following command in the terminal shell to import the lab files into your Vertex AI Workbench instance:

```
gsutil cp gs://qwiklabs-gcp-03-d5494d0f455b-labconfig-bucket/notebooks/*.ipynb .
```

Task 3. Make a synchronous process document request 20/100

Task 4. Run the synchronous Document AI Python code

Task 5. Create a Document AI Document OCR processor

Task 6. Prepare your environment for asynchronous Document AI API calls

Task 7. Make an asynchronous process document request

Congratulations

[Previous](#)[Next](#)

form-parser – Document AI – qw X

https://console.cloud.google.com/ai/document-ai/locations/us/processors/566142678ebc6a83/details?project=qwiklabs-gcp-03-d5494d0f455b

Free AI Paraphrasing... Transcript of Case Study GCP-LAB Hydra datawarehouse Products

Google Cloud

- Cloud Hub >
- Cloud overview >
- Solutions >
- Recently visited >
- Favorite products
- Favorite products appear here
- Products
- Billing ☆
- IAM & Admin ☆ >
- Marketplace ☆
- APIs & Services ☆ >
- Vertex AI ☆ >
- Compute Engine ☆ >
- Kubernetes Engine ☆ >

[View all products](#)

document ai

Disable processor Activity Get Code

form-parser
566142678ebc6a83
Enabled
Form Parser
Dec 30, 2025, 6:08:44 PM
Google-managed
us

https://us-documentai.googleapis.com/v1/projects/780504336376/locations/us/processors/566142678ebc6a83:process

PDF, TIFF, TIF, GIF (15 pages, 20MB max)

Processor created successfully

7 14°C Clear

Search

ENG IN 06:09 PM 30-12-2025

Workbench – Vertex AI – qwiklab

https://console.cloud.google.com/vertex-ai/workbench?invt=AcFleA&project=qwiklabs-gcp-03-d5494d0f455b

Free AI Paraphrasin... Transcript of Case St... GCP-LAB Hydra datawarehouse Products

Google Cloud qwiklabs-gcp-03-d5494d0f455b Search (/) for resources, docs, products, and more Search

Vertex AI / Workbench

Dashboard Model Garden Vertex AI Studio New GenAI Evaluation New Tuning Agent Designer Preview Agent Garden Agent Engine Tools RAG Engine Vertex AI Search Vector Search Provisioned Throughput Pipelines Tutorials

Workbench + Create New ⌛ Refresh Instances Executions Schedules

View: Instances User-managed Notebooks Managed Notebooks

Dismiss

JupyterLab 4 is now generally available in Vertex AI Workbench.

Workbench Instances have JupyterLab 4 pre-installed and are configured with GPU-enabled machine learning frameworks. [Learn more](#)

Filter Enter property name or value

	Instance name ↑	Zone	Auto upgrade	Version	Machine Type	GPUs	Owner	Created
<input type="checkbox"/>	<input checked="" type="radio"/> jupyterlab	us-east1-c	—	M137	None	None	780504336376-compute@developer.gserviceaccount.com	Dec 3, 2025, 5:31:31 PM

Finance headline India reported 8... Search

06:10 PM 30-12-2025 ENG IN

Workbench – Vertex AI – qwiklab

https://console.cloud.google.com/vertex-ai/workbench?inv=AcFleA&project=qwiklabs-gcp-03-d5494d0f455b

Free AI Paraphrasin... Transcript of Case St... GCP-LAB Hydra datawarehouse Products

Google Cloud qwiklabs-gcp-03-d5494d0f455b Search (/) for resources, docs, products, and more Search

Vertex AI / Workbench

Dashboard Model Garden Vertex AI Studio New GenAI Evaluation New Tuning Agent Builder Agent Designer Preview Agent Garden Agent Engine Tools RAG Engine Vertex AI Search Vector Search Provisioned Throughput Pipelines Tutorials

1 30-12-2025 06:10 PM ENG IN

1 instance selected Start Reset Stop Delete

Instances Executions Schedules

View: Instances User-managed Notebooks Managed Notebooks

JupyterLab 4 is now generally available in Vertex AI Workbench. Dismiss

Workbench Instances have JupyterLab 4 pre-installed and are configured with GPU-enabled machine learning frameworks. [Learn more](#)

Filter Enter property name or value

Instance name	Zone	Auto upgrade	Version	Machine Type	GPUs	Owner	Created
jupyterlab	us-east1-c	—	M137	None	None	780504336376-compute@developer.gserviceaccount.com	Dec 3, 2025, 5:31:3

Finance headline India reported 8...

Search

06:10 PM 30-12-2025 ENG IN

Workbench – Vertex AI – qwiklab

https://console.cloud.google.com/vertex-ai/workbench?invt=AcFleA&project=qwiklabs-gcp-03-d5494d0f455b

Free AI Paraphrasin... Transcript of Case St... GCP-LAB Hydra datawarehouse Products

Google Cloud qwiklabs-gcp-03-d5494d0f455b Search (/) for resources, docs, products, and more Search

Vertex AI / Workbench

Dashboard Model Garden Vertex AI Studio New GenAI Evaluation New Tuning Agent Designer Preview Agent Garden Agent Engine Tools RAG Engine Vertex AI Search Vector Search Provisioned Throughput Pipelines Tutorials

Workbench + Create New Refresh Instances Executions Schedules

View: Instances User-managed Notebooks Managed Notebooks

JupyterLab 4 is now generally available in Vertex AI Workbench. Dismiss

Workbench Instances have JupyterLab 4 pre-installed and are configured with GPU-enabled machine learning frameworks. [Learn more](#)

Instance name	Zone	Auto upgrade	Version	Machine Type	GPUs	Owner	Created
jupyterlab	us-east1-c	—	M137	None	None	780504336376-compute@developer.gserviceaccount.com	Dec 3, 2025, 5:31:31

8 14°C Clear

Search

06:14 PM 30-12-2025 ENG IN

Workbench – Vertex AI – quiklabs-go JupyterLab https://2defb48634214276-dot-us-east1.notebooks.googleusercontent.com/lab

Free AI Paraphrasin... Transcript of Case St... GCP-LAB Hydra datawarehouse Products

File Edit View Run Kernel Git Tabs Settings Help

Launcher

Notebook

- Python 3 (ipykernel) (Local)
- PyTorch 1-13 (Local)
- TensorFlow 2-11 (Local)

Console

- Python 3 (ipykernel) (Local)
- PyTorch 1-13 (Local)
- TensorFlow 2-11 (Local)

Other

- Terminal
- Text File
- Markdown File
- Python File
- Show Contextual Help

There is a new version of Dataprof extension available. Would you like to update the extension?

Update Ignore

Would you like to get notified about official Jupyter news?

Open privacy policy Yes No

Simple 0 \$ 0 0 0 Launcher 3 0 14°C Clear

Search

06:22 PM 30-12-2025 ENG IN

Google Skills Partner

Contents

Process Documents with Python Using the Docur ▾

Workbench.

2. Find the **jupyterlab** instance and click on the **Open JupyterLab** button.

The JupyterLab interface for your Workbench instance opens in a new browser tab.

3. Click **Terminal** to open a terminal shell inside the Vertex AI Workbench instance.

4. Enter the following command in the terminal shell to import the lab files into your Vertex AI Workbench instance:

```
gsutil cp gs://qwiklabs-gcp-03-d5494d0f455b-labconfig-bucket/notebooks/*.ipynb .
```

5. Enter the following command in the terminal shell to install the Python client libraries required for Document AI and other required libraries:

```
python -m pip install --upgrade google-cloud-core google-cloud-doc
```

You should see output indicating that the libraries have been installed successfully.

Task 3. Make a synchronous process document request
Task 4. Run the synchronous Document AI Python code
Task 5. Create a Document AI Document OCR processor
Task 6. Prepare your environment for asynchronous Document AI API calls
Task 7. Make an asynchronous process document request
Congratulations

Contents

Process Documents with Python Using the Docur

Lab setup instructions and requirements

Protect your account and

End Lab 01:11:22 Time limit

Caution: When you are in the console, do not deviate from the lab instructions. Doing so may cause your account to be blocked. Learn more.

Open Google Cloud console

Username: student-01-75364ecd3f8c1

Password: YFyxc227VWPX

Project ID: qwiklabs-gcp-03-d5494d01

gsutil cp gs://qwiklabs-gcp-03-d5494d0f455b-labconfig-bucket/notebooks/*.ipynb .

5. Enter the following command in the terminal shell to install the Python client libraries required for Document AI and other required libraries:

```
python -m pip install --upgrade google-cloud-core google-cloud-doc
```

You should see output indicating that the libraries have been installed successfully.

Note: In case of permission related errors, re-run the command to ensure successful installation of the libraries. It can take a few minutes for the permissions to be applied.

6. Enter the following command in the terminal shell to import the sample health intake form:

```
gcloud ai-platform predict --model=health-intake-form --json-instances=sample-health-intake-form.json
```

20/100

Task 3. Make a synchronous process document request

Task 4. Run the synchronous Document AI Python code

Task 5. Create a Document AI Document OCR processor

Task 6. Prepare your environment for asynchronous Document AI API calls

Task 7. Make an asynchronous process document request

Congratulations

< Previous

Next >

Free AI Paraphrasin... Transcript of Case St... GCP-LAB Hydra datawarehouse Products

Google Skills Partner

6425 3

Automate Data Capture at Scale with Document AI > Process Documents with Python Using the Document AI API

Contents

Process Documents with Python Using the Docur ▾

Lab setup instructions and requirements

Protect your account and

End Lab 01:11:17 ? Time limit

You should see output indicating that the libraries have been installed successfully.

Note: In case of permission related errors, re-run the command to ensure successful installation of the libraries. It can take a few minutes for the permissions to be applied.

6. Enter the following command in the terminal shell to import the sample health intake form:

```
gsutil cp gs://qwiklabs-gcp-03-d5494d0f455b-labconfig-bucket/health-intake-form.pdf form.pdf
```

7. In the notebook interface open the JupyterLab notebook called documentai-sync-v1.0.0.ipynb.

8. In the **Select Kernel** dialog, choose **Python 3** from the list of available kernels.

Check that the Vertex AI instance has been prepared for synchronous

Task 3. Make a synchronous process document request
Task 4. Run the synchronous Document AI Python code
Task 5. Create a Document AI Document OCR processor
Task 6. Prepare your environment for asynchronous Document AI API calls
Task 7. Make an asynchronous process document request
Congratulations

Workbench – Vertex AI – qwiklabs-gcp lab - JupyterLab https://2defb48634214276-dot-us-east1.notebooks.googleusercontent.com/lab

Free AI Paraphrasing... Transcript of Case St... GCP-LAB Hydra datawarehouse Products

File Edit View Run Kernel Git Tabs Settings Help

Terminal 1

```
(base) jupyter@jupyterlab:~$ gsutil cp gs://qwiklabs-gcp-03-d5494d0f455b-labconfig-bucket/notebooks/*.ipynb .
Copying gs://qwiklabs-gcp-03-d5494d0f455b-labconfig-bucket/notebooks/documentai-async-v1.0.0.ipynb...
Copying gs://qwiklabs-gcp-03-d5494d0f455b-labconfig-bucket/notebooks/documentai-sync-v1.0.0.ipynb...
[2 files][ 14.4 KiB/ 14.4 KiB]
Operation completed over 2 objects/14.4 KiB.

(base) jupyter@jupyterlab:~$ python -m pip install --upgrade google-cloud-core google-cloud-documentai google-cloud-storage prettytable
Requirement already satisfied: google-cloud-core in /opt/conda/lib/python3.10/site-packages (2.5.0)
Collecting google-cloud-documentai
  Downloading google_cloud_documentai-3.7.0-py3-none-any.whl.metadata (9.8 kB)
Requirement already satisfied: google-cloud-storage in /opt/conda/lib/python3.10/site-packages (3.7.0)
Requirement already satisfied: prettytable in /opt/conda/lib/python3.10/site-packages (3.17.0)
Requirement already satisfied: google-api-core!=2.0.*,!2.1.*,!2.2.*,!2.3.0,<3.0.0,>=1.31.6 in /opt/conda/lib/python3.10/site-packages (from google-cloud-core) (2.28.1)
Requirement already satisfied: google-auth<3.0.0,>=1.25.0 in /opt/conda/lib/python3.10/site-packages (from google-cloud-core) (2.43.0)
Requirement already satisfied: googleapis-common-protos<2.0.0,>=1.56.2 in /opt/conda/lib/python3.10/site-packages (from google-api-core!=2.0.*,!2.1.*,!2.2.*,!2.3.0,<3.0.0,>=1.31.6->google-cloud-core) (1.72.0)
Requirement already satisfied: protobuf!=3.20.0,!3.20.1,!4.21.0,!4.21.1,!4.21.2,!4.21.3,!4.21.4,!4.21.5,<7.0.0,>=3.19.5 in /opt/conda/lib/python3.10/site-packages (from google-api-core!=2.0.*,!2.1.*,!2.2.*,!2.3.0,<3.0.0,>=1.31.6->google-cloud-core) (6.33.2)
Requirement already satisfied: proto-plus<2.0.0,>=1.22.3 in /opt/conda/lib/python3.10/site-packages (from google-api-core!=2.0.*,!2.1.*,!2.2.*,!2.3.0,<3.0.0,>=1.31.6->google-cloud-core) (1.26.1)
Requirement already satisfied: requests<3.0.0,>=2.18.0 in /opt/conda/lib/python3.10/site-packages (from google-api-core!=2.0.*,!2.1.*,!2.2.*,!2.3.0,<3.0.0,>=1.31.6->google-cloud-core) (2.32.5)
Requirement already satisfied: cachetools<7.0,>=2.0.0 in /opt/conda/lib/python3.10/site-packages (from google-auth<3.0.0,>=1.25.0->google-cloud-core) (6.2.2)
Requirement already satisfied: pyasn1-modules>=0.2.1 in /opt/conda/lib/python3.10/site-packages (from google-auth<3.0.0,>=1.25.0->google-cloud-core) (0.4.2)
Requirement already satisfied: rsa<5,>=3.1.4 in /opt/conda/lib/python3.10/site-packages (from google-auth<3.0.0,>=1.25.0->google-cloud-core) (4.9.1)
Requirement already satisfied: charset_normalizer<4,>=2 in /opt/conda/lib/python3.10/site-packages (from requests<3.0.0,>=2.18.0->google-api-core!=2.0.*,!2.1.*,!2.2.*,!2.3.0,<3.0.0,>=1.31.6->google-cloud-core) (3.4.4)
Requirement already satisfied: idna<4,>=2.5 in /opt/conda/lib/python3.10/site-packages (from requests<3.0.0,>=2.18.0->google-api-core!=2.0.*,!2.1.*,!2.2.*,!2.3.0,<3.0.0,>=1.31.6->google-cloud-core) (3.11)
Requirement already satisfied: urllib3<3,>=1.21.1 in /opt/conda/lib/python3.10/site-packages (from requests<3.0.0,>=2.18.0->google-api-core!=2.0.*,!2.1.*,!2.2.*,!2.3.0,<3.0.0,>=1.31.6->google-cloud-core) (2.6.1)
Requirement already satisfied: certifi>=2017.4.17 in /opt/conda/lib/python3.10/site-packages (from requests<3.0.0,>=2.18.0->google-api-core!=2.0.*,!2.1.*,!2.2.*,!2.3.0,<3.0.0,>=1.31.6->google-cloud-core) (2025.11.12)
Requirement already satisfied: pyasn1>=0.1.3 in /opt/conda/lib/python3.10/site-packages (from rsa<5,>=3.1.4->google-auth<3.0.0,>=1.25.0->google-cloud-core) (0.6.1)
Requirement already satisfied: grpcio<2.0.0,>=1.33.2 in /opt/conda/lib/python3.10/site-packages (from google-cloud-documentai) (1.76.0)
Requirement already satisfied: grpcio-status<2.0.0,>=1.33.2 in /opt/conda/lib/python3.10/site-packages (from google-api-core[grpc]!=2.0.*,!2.1.*,!2.10.*,!2.2.*,!2.3.*,!2.4.*,!2.5.*,!2.6.*,!2.7.*,!2.8.*,!2.9.*,<3.0.0,>=1.34.1->google-cloud-documentai) (1.76.0)
Requirement already satisfied: typing-extensions>=4.12 in /opt/conda/lib/python3.10/site-packages (from grpcio<2.0.0,>=1.33.2->google-cloud-documentai) (4.15.0)
```

Simple 1 \$ 0 🔍 Terminal 1 3 ENG IN 06:26 PM 30-12-2025

8 14°C Clear

Workbench – Vertex AI – qwiklabs-go lab - JupyterLab

https://2defb48634214276-dot-us-east1.notebooks.googleusercontent.com/lab

Free AI Paraphrasing... Transcript of Case St... GCP-LAB Hydra datawarehouse Products

File Edit View Run Kernel Git Tabs Settings Help

+

/

Name Modified

- documentai-async-v1.0.0.ipynb 6 sec. ago
- documentai-sync-v1.0.0.ipynb 6 sec. ago
- notebook_template.ipynb 51 min. ago
- requirements.txt 49 min. ago

Terminal 1

```
Collecting google-cloud-documentai
  Downloading google_cloud_documentai-3.7.0-py3-none-any.whl.metadata (9.8 kB)
Requirement already satisfied: google-cloud-storage in /opt/conda/lib/python3.10/site-packages (3.7.0)
Requirement already satisfied: prettytable in /opt/conda/lib/python3.10/site-packages (3.17.0)
Requirement already satisfied: google-api-core!=2.0.*,!=2.1.*,!=2.2.*,!=2.3.0,<3.0.0,>=1.31.6 in /opt/conda/lib/python3.10/site-packages (from google-cloud-core) (2.28.1)
Requirement already satisfied: google-auth<3.0.0,>=1.25.0 in /opt/conda/lib/python3.10/site-packages (from google-cloud-core) (2.43.0)
Requirement already satisfied: googleapis-common-protos<2.0.0,>=1.56.2 in /opt/conda/lib/python3.10/site-packages (from google-api-core!=2.0.*,!=2.1.*,!=2.2.*,!=2.3.0,<3.0.0,>=1.31.6->google-cloud-core) (1.72.0)
Requirement already satisfied: protobuf!=3.20.0,!=3.20.1,!=4.21.0,!=4.21.1,!=4.21.2,!=4.21.3,!=4.21.4,!=4.21.5,<7.0.0,>=3.19.5 in /opt/conda/lib/python3.10/site-packages (from google-api-core!=2.0.*,!=2.1.*,!=2.2.*,!=2.3.0,<3.0.0,>=1.31.6->google-cloud-core) (6.33.2)
Requirement already satisfied: proto-plus<2.0.0,>=1.22.3 in /opt/conda/lib/python3.10/site-packages (from google-api-core!=2.0.*,!=2.1.*,!=2.2.*,!=2.3.0,<3.0.0,>=1.31.6->google-cloud-core) (1.26.1)
Requirement already satisfied: requests<3.0.0,>=2.18.0 in /opt/conda/lib/python3.10/site-packages (from google-api-core!=2.0.*,!=2.1.*,!=2.2.*,!=2.3.0,<3.0.0,>=1.31.6->google-cloud-core) (2.32.5)
Requirement already satisfied: cachetools<7.0,>=2.0.0 in /opt/conda/lib/python3.10/site-packages (from google-auth<3.0.0,>=1.25.0->google-cloud-core) (6.2.2)
Requirement already satisfied: pyasn1-modules>=0.2.1 in /opt/conda/lib/python3.10/site-packages (from google-auth<3.0.0,>=1.25.0->google-cloud-core) (0.4.2)
Requirement already satisfied: rsa<5,>=3.1.4 in /opt/conda/lib/python3.10/site-packages (from google-auth<3.0.0,>=1.25.0->google-cloud-core) (4.9.1)
Requirement already satisfied: charset_normalizer<4,>=2 in /opt/conda/lib/python3.10/site-packages (from requests<3.0.0,>=2.18.0->google-api-core!=2.0.*,!=2.1.*,!=2.2.*,!=2.3.0,<3.0.0,>=1.31.6->google-cloud-core) (3.4.4)
Requirement already satisfied: idna<4,>=2.5 in /opt/conda/lib/python3.10/site-packages (from requests<3.0.0,>=2.18.0->google-api-core!=2.0.*,!=2.1.*,!=2.2.*,!=2.3.0,<3.0.0,>=1.31.6->google-cloud-core) (3.11)
Requirement already satisfied: urllib3<3,>=1.21.1 in /opt/conda/lib/python3.10/site-packages (from requests<3.0.0,>=2.18.0->google-api-core!=2.0.*,!=2.1.*,!=2.2.*,!=2.3.0,<3.0.0,>=1.31.6->google-cloud-core) (2.6.1)
Requirement already satisfied: certifi>=2017.4.17 in /opt/conda/lib/python3.10/site-packages (from requests<3.0.0,>=2.18.0->google-api-core!=2.0.*,!=2.1.*,!=2.2.*,!=2.3.0,<3.0.0,>=1.31.6->google-cloud-core) (2025.11.12)
Requirement already satisfied: pyasn1>=0.1.3 in /opt/conda/lib/python3.10/site-packages (from rsa<5,>=3.1.4->google-auth<3.0.0,>=1.25.0->google-cloud-core) (0.6.1)
Requirement already satisfied: grpcio<2.0.0,>=1.33.2 in /opt/conda/lib/python3.10/site-packages (from google-cloud-documentai) (1.76.0)
Requirement already satisfied: grpcio-status<2.0.0,>=1.33.2 in /opt/conda/lib/python3.10/site-packages (from google-api-core[grpc]!=2.0.*,!=2.1.*,!=2.10.*,!=2.2.*,!=2.3.*,!=2.4.*,!=2.5.*,!=2.6.*,!=2.7.*,!=2.8.*,!=2.9.*,<3.0.0,>=1.34.1->google-cloud-documentai) (1.76.0)
Requirement already satisfied: typing-extensions<=4.12 in /opt/conda/lib/python3.10/site-packages (from grpcio<2.0.0,>=1.33.2->google-cloud-documentai) (4.15.0)
Requirement already satisfied: google-resumable-media<3.0.0,>=2.7.2 in /opt/conda/lib/python3.10/site-packages (from google-cloud-storage) (2.8.0)
Requirement already satisfied: google-crc32c<2.0.0,>=1.1.3 in /opt/conda/lib/python3.10/site-packages (from google-cloud-storage) (1.7.1)
Requirement already satisfied: wctwidth in /opt/conda/lib/python3.10/site-packages (from prettytable) (0.2.14)
  Downloading google_cloud_documentai-3.7.0-py3-none-any.whl (303 kB)
  Installing collected packages: google-cloud-documentai
Successfully installed google-cloud-documentai-3.7.0
(base) jupyterlab:~$
```

Simple 1 \$ 0 🌐 🔍 Terminal 1 3 8 14°C Clear ENG IN 06:26 PM 30-12-2025

Workbench – Vertex AI – qwiklabs-gcp lab - JupyterLab

https://2defb48634214276-dot-us-east1.notebooks.googleusercontent.com/lab

Free AI Paraphrasing... Transcript of Case St... GCP-LAB Hydra datawarehouse Products

File Edit View Run Kernel Git Tabs Settings Help

Terminal 1

```
(base) jupyter@jupyterlab:~$ gsutil cp gs://qwiklabs-gcp-03-d5494d0f455b-labconfig-bucket/health-intake-form.pdf form.pdf
Copying gs://qwiklabs-gcp-03-d5494d0f455b-labconfig-bucket/health-intake-form.pdf...
/ [1 files][624.2 KiB/624.2 KiB]
Operation completed over 1 objects/624.2 KiB.
(base) jupyter@jupyterlab:~$
```

Simple 1 \$ 0 ⚡ Terminal 1 3 14°C ENG IN 06:27 PM 30-12-2025

Contents

Process Documents with Python Using the Docur ▾

Lab setup instructions and requirements

Protect your account and

End Lab 01:03:09 ? Time limit

Caution: When you are in the console, do not deviate from the lab instructions. Doing so may cause your account to be blocked. Learn more.

Open Google Cloud console

Username: student-01-75364ecd3f8c1

Password: YFyxc227VWPX

Project ID: qwiklabs-gcp-03-d5494d01

Task 4. Run the synchronous Document AI Python code

Execute the code to make synchronous calls to the Document AI API in the JupyterLab notebook.

- In the second **Set your Processor ID** code cell replace the `PROCESSOR_ID` placeholder text with the Processor ID for the **form-parser** processor you created in an earlier step.
- Select the first cell, click the **Run** menu and then click **Run Selected Cell and All Below** to run all the code in the notebook.

If you have used the sample health intake form, you will see data similar to the following for the output cell for the form data:

Form data detected:

Page Number:1 Phone #: (906) 917-3486 (Confidence Scores: (Name) 1.0, (Value) 0.9)

◀ Previous Next ▶

Workbench – Vertex AI – qwiklabs-gcp documentai-sync-v1.0.0.ipynb +

https://2defb48634214276-dot-us-east1.notebooks.googleusercontent.com/lab/tree/documentai-sync-v1.0.0.ipynb

Free AI Paraphrasing... Transcript of Case St... GCP-LAB Hydra datawarehouse Products

File Edit View Run Kernel Git Tabs Settings Help

Terminal 1 documentai-sync-v1.0.0.ipynb +

No Kernel Initializing

Document AI Synchronous API

This notebook shows you how use Python to make synchronous calls to the Document AI API

You must replace the `processor_id` variable value in the the second cell with the appropriate value for the Processor ID of the Document AI processor that you want to use. The processor may not support all of the Document AI output properties. Entity data is only returned by processors that use specialized parsers for ex

Select Kernel

Select kernel for: "documentai-sync-v1.0.0.ipynb"

Python 3 (ipykernel) (Local)

Always start the preferred kernel No Kernel Select

'us' location

Replace this with the name of the pdf file to process if necessary

```
[ ]: from google import documentai from google import api_core from prettytable import PrettyTable
```

```
[ ]: project_id='us' project_id='us' location = 'us' file_path = 'form.pdf' # Replace this with the name of the pdf file to process if necessary
```

```
[ ]: # Set your Processor ID processor_id = 'PROCESSOR_ID' # TODO: Replace with a valid Processor ID
```

```
[ ]: # Process Document Function
```

```
[ ]: def process_document( project_id=project_id, location=location, processor_id=processor_id, file_path=file_path ):
```

```
[ ]:     # Instantiates a client
```

```
[ ]:     client = documentai.DocumentProcessorServiceClient()
```

Simple 1 \$ 0 No Kernel | Initializing Mode: Command Ln 1, Col 1 documentai-sync-v1.0.0.ipynb 3 ENG IN 14°C Clear Search

8 30-12-2025 06:28 PM

Workbench – Vertex AI – qwiklabs-gcp documentai-sync-v1.0.0.ipynb

https://2defb48634214276-dot-us-east1.notebooks.googleusercontent.com/lab/tree/documentai-sync-v1.0.0.ipynb

File Edit View Run Kernel Git Tabs Settings Help

Terminal 1 documentai-sync-v1.0.0.ipynb

Python 3 (ipykernel) (Local)

Document AI Synchronous API

This notebook shows you how use Python to make synchronous calls to the Document AI API

You must replace the `processor_id` variable value in the the second cell with the appropriate value for the Processor ID of the Document AI processor that you want to use. The processor may not support all of the Document AI output properties. Entity data is only returned by processors that use specialized parsers for example.

```
[ ]: from google.cloud import documentai_v1beta3 as documentai
from google.cloud import storage
from prettytable import PrettyTable

project_id=%system gcloud config get-value core/project
project_id=project_id[0]
location = 'us' # Replace with 'eu' if processor does not use 'us' location
file_path = 'form.pdf' # Replace this with the name of the pdf file to process if necessary
```

```
[ ]: # Set your Processor ID
processor_id = 'PROCESSOR_ID' # TODO: Replace with a valid Processor ID
```

```
[ ]: # Process Document Function
def process_document(
    project_id=project_id, location=location,
    processor_id=processor_id, file_path=file_path
):
    # Instantiates a client
    client = documentai.DocumentProcessorServiceClient()
```

Simple 1 \$ 1 ⚡ Python 3 (ipykernel) (Local) | Connecting Mode: Command ✘ Ln 1, Col 1 documentai-sync-v1.0.0.ipynb 3 8 14°C ENG IN 06:28 PM 30-12-2025 Clear Search



```
print("Form data detected:\n")
# For each page fetch each form field and display fieldname, value and confidence scores
for page in document.pages:
    print("Page Number:{}\n".format(page.page_number))
    for form_field in page.form_fields:
        fieldName = get_text(form_field.field_name, document)
        nameConfidence = round(form_field.field_name.confidence, 4)
        fieldValue = get_text(form_field.field_value, document)
        valueConfidence = round(form_field.field_value.confidence, 4)
        print(fieldName + fieldValue + " (Confidence Scores: (Name) " + str(nameConfidence) + ", (Value) " + str(valueConfidence) + ")\n")

[ ]: # Display Entity Data
# Entity data is only detected by specialized parsers, such as the Procurement Expense parser.
# For each entity print the key/value pair and their corresponding confidence scores.
if 'entities' in dir(document):
    entities = document.entities
    table = PrettyTable(['Type', 'Value', 'Confidence'])
    entities_found = 0
    for entity in entities:
        entity_type = entity.type_
        value = entity.mention_text
        confidence = round(entity.confidence, 4)
        table.add_row([entity_type, value, confidence])
    print(table)
else:
    print("Document does not contain entity data.")

[ ]:
```

Workbench – Vertex AI – qwiklabs-gcp documentai-sync-v1.0.ipynb +

https://2defb48634214276-dot-us-east1.notebooks.googleusercontent.com/lab/tree/documentai-sync-v1.0.ipynb

Free AI Paraphrasing... Transcript of Case St... GCP-LAB Hydra datawarehouse Products

File Edit View Run Kernel Git Tabs Settings Help

Terminal 1 documentai-sync-v1.0.ipynb +

You must replace the `processor_id` variable value in the second cell with the appropriate value for the Processor ID of the Document AI processor that you want to use. The processor may not support all of the Document AI output properties. Entity data is only returned by processors that use specialized parsers for example.

```
[ ]: from google.cloud import documentai_v1beta3 as documentai
from google.cloud import storage
from prettytable import PrettyTable

project_id=%system gcloud config get-value core/project
project_id=project_id[0]
location = 'us'          # Replace with 'eu' if processor does not use 'us' location
file_path = 'form.pdf'    # Replace this with the name of the pdf file to process if necessary

# Set your Processor ID
processor_id = '566142678ebc6a83' # TODO: Replace with a valid Processor ID

# Process Document Function
def process_document(
    project_id=project_id, location=location,
    processor_id=processor_id, file_path=file_path
):
    # Instantiates a client
    client = documentai.DocumentProcessorServiceClient()
    # The full resource name of the processor, e.g.:
    # projects/project-id/locations/location/processor/processor-id
    # You must create new processors in the Cloud Console first
    name = f"projects/{project_id}/locations/{location}/processors/{processor_id}"
    with open(file_path, "rb") as image:
```

Python 3 (ipykernel) (Local)

Simple 1 \$ 1 ⚡ Python 3 (ipykernel) (Local) | Idle Mode: Command ✘ Ln 1, Col 1 documentai-sync-v1.0.ipynb 3 14°C Clear Search 📁 📺 🎙️ 📱 📺 🎙️ 📱 📺 🎙️ 📱 ENG IN 06:29 PM 30-12-2025

Workbench – Vertex AI – qwiklabs-gcp documentai-sync-v1.0.0.ipynb +

https://2defb48634214276-dot-us-east1.notebooks.googleusercontent.com/lab/tree/documentai-sync-v1.0.0.ipynb

Free AI Paraphrasing... Transcript of Case St... GCP-LAB Hydra datawarehouse Products

File Edit View Run Kernel Git Tabs Settings Help

Terminal 1 documentai-sync-v1.0.0.ipynb +

Form data detected:

Page Number: 1
Are you currently taking any medication? (If yes, please describe):
Vyvanse (25mg) daily for attention.
(Confidence Scores: (Name) 0.9433, (Value) 0.9433)

Phone #: walker@cmail.com
(906) (Confidence Scores: (Name) 0.9323, (Value) 0.9323)

Zip: 07082
(Confidence Scores: (Name) 0.9142, (Value) 0.9142)

City: Towaco
(Confidence Scores: (Name) 0.9005, (Value) 0.9005)

State: NJ (Confidence Scores: (Name) 0.8939, (Value) 0.8939)

DOB: 09/04/1986
(Confidence Scores: (Name) 0.8852, (Value) 0.8852)

Gender: F
(Confidence Scores: (Name) 0.8824, (Value) 0.8824)

Name:
Sally Walker
(Confidence Scores: (Name) 0.8728, (Value) 0.8728)

Marital Status:
Single (Confidence Scores: (Name) 0.8524, (Value) 0.8524)

Describe your medical concerns (symptoms, diagnoses, etc):
Ranny nose, mucas in thwat, weakness,
aches, chills, tired

Python 3 (ipykernel) (Local)

Simple 1 \$ 1 ⚙️ Python 3 (ipykernel) (Local) | Idle Mode: Command ✘ Ln 1, Col 1 documentai-sync-v1.0.0.ipynb 3 8 14°C Clear Search 📁 📱 📹 📺 📤 📥 📸 📹 📲 ENG IN 06:30 PM 30-12-2025

Workbench – Vertex AI – qwiklabs-gcp documentai-sync-v1.0.ipynb

https://2defb48634214276-dot-us-east1.notebooks.googleusercontent.com/lab/tree/documentai-sync-v1.0.ipynb

Free AI Paraphrasing... Transcript of Case St... GCP-LAB Hydra datawarehouse Products

File Edit View Run Kernel Git Tabs Settings Help

Terminal 1 documentai-sync-v1.0.ipynb

Single (Confidence Scores: (Name) 0.8524, (Value) 0.8524)

Describe your medical concerns (symptoms, diagnoses, etc):
Ranny nose, mucas in thwat, weakness,
aches, chills, tired
(Confidence Scores: (Name) 0.8439, (Value) 0.8439)

Date:
9/14/19
(Confidence Scores: (Name) 0.83, (Value) 0.83)

Address: 24 Barney Lane
(Confidence Scores: (Name) 0.8239, (Value) 0.8239)

Occupation: Software Engineer
(Confidence Scores: (Name) 0.815, (Value) 0.815)

Emergency Contact: Eva Walker (Confidence Scores: (Name) 0.8134, (Value) 0.8134)

Email: Sally, walker@cmail.com
(Confidence Scores: (Name) 0.7974, (Value) 0.7974)

Referred By: None
(Confidence Scores: (Name) 0.7694, (Value) 0.7694)

Emergency Contact Phone: (906)334-8926
(Confidence Scores: (Name) 0.7612, (Value) 0.7612)

[7]: # Display Entity Data
Entity data is only detected by specialized parsers, such as the Procurement Expense parser.
For each entity print the key/value pair and their corresponding confidence scores.
if 'entities' in dir(document):
 entities=document.entities

Simple 1 \$ 1 Python 3 (ipykernel) (Local) | Idle Mode: Command Ln 1, Col 1 documentai-sync-v1.0.ipynb 3 ENG IN 14°C Clear Search 06:31 PM 30-12-2025

Workbench – Vertex AI – qwiklabs-gcp documentai-sync-v1.0.0.ipynb

https://2defb48634214276-dot-us-east1.notebooks.googleusercontent.com/lab/tree/documentai-sync-v1.0.0.ipynb

File Edit View Run Kernel Git Tabs Settings Help

Terminal 1 documentai-sync-v1.0.0.ipynb

```
# Entity data is only detected by specialized parsers, such as the Procurement Expense parser.  
# For each entity print the key/value pair and their corresponding confidence scores.  
  
if 'entities' in dir(document):  
    entities=document.entities  
    table = PrettyTable(['Type', 'Value', 'Confidence'])  
    entities_found = 0  
    for entity in entities:  
        entity_type = entity.type_  
        value = entity.mention_text  
        confidence = round(entity.confidence,4)  
        table.add_row([entity_type, value, confidence])  
        print(table)  
    else:  
        print("Document does not contain entity data.")  
  
+-----+-----+-----+  
|      Type      | Value | Confidence |  
+-----+-----+-----+  
| generic_entities |      | 0.0 |  
+-----+-----+-----+
```

Python 3 (ipykernel) (Local)

Simple 1 \$ 1 Python 3 (ipykernel) (Local) | Idle Mode: Command Ln 1, Col 1 documentai-sync-v1.0.0.ipynb 3 14°C ENG IN 06:31 PM 30-12-2025 Clear

Workbench – Vertex AI – qwiklabs-gcp documentai-sync-v1.0.0.ipynb

https://2defb48634214276-dot-us-east1.notebooks.googleusercontent.com/lab/tree/documentai-sync-v1.0.0.ipynb

Free AI Paraphrasing... Transcript of Case Study GCP-LAB Hydra datawarehouse Products

File Edit View Run Kernel Git Tabs Settings Help

New New Text Notebook Jupyter New Launcher Ctrl+Shift+L

Open from Path... Open from URL... Open Recent

New View for Notebook New Console for Notebook

Close Tab Alt+W Close and Shut Down Notebook... Ctrl+Shift+Q Close All Tabs

Save Notebook Ctrl+S Save Notebook As... Ctrl+Shift+S Save All

Reload Notebook from Disk Revert Notebook to Checkpoint... Rename Notebook... Duplicate Notebook

Download

Save and Export Notebook As

Workspaces

Print... Ctrl+P

Log Out

Terminal 1 documentai-sync-v1.0.0.ipynb +

Entity data is only detected by specialized parsers, such as the Procurement Expense parser.
For each entity print the key/value pair and their corresponding confidence scores.

```
if 'entities' in dir(document):  
    entities=document.entities  
    table = PrettyTable(['Type', 'Value', 'Confidence'])  
    entities_found = 0  
    for entity in entities:  
        entity_type = entity.type_  
        value = entity.mention_text  
        confidence = round(entity.confidence,4)  
        table.add_row([entity_type, value, confidence])  
        print(table)  
    else:  
        print("Document does not contain entity data.")
```

Type	Value	Confidence
generic_entities		0.0

[]:

(Local) | Idle Mode: Command

8 14°C Clear Search

06:31 PM 30-12-2025 ENG IN

Free AI Paraphrasin... Transcript of Case St... GCP-LAB Hydra datawarehouse Products

Google Skills Partner

6425 3

Automate Data Capture at Scale with Document AI > Process Documents with Python Using the Document AI API

Contents

Process Documents with Python Using the Docur ▾

Lab setup instructions and requirements

Protect your account and

End Lab 01:02:38 ? Time limit

Caution: When you are in the console, do not deviate from the lab instructions. Doing so may cause your account to be blocked. Learn more.

Open Google Cloud console

Username: student-01-75364ecd3f8c1

Password: YFyxc227VWPX

Project ID: qwiklabs-gcp-03-d5494d01

Task 5. Create a Document AI Document OCR processor

In this task you will create a Document AI processor using the general Document OCR parser.

- From the **Navigation menu**, click **Document AI > Overview**.
- Click **Explore Processors** and then click **Create Processor for Document OCR**.
This is a type of general processor.
- Specify the processor name as **ocr-processor** and select the region **US (United States)** from the list.
- Click **Create** to create your processor.
- Make a note of the **processor ID**. You will need to specify this in a later task.

Task 6. Prepare your environment for Document AI API calls

60/100

Task 3. Make a synchronous process document request

Task 4. Run the synchronous Document AI Python code

Task 5. Create a Document AI Document OCR processor

Task 6. Prepare your environment for asynchronous Document AI API calls

Task 7. Make an asynchronous process document request

Congratulations

◀ Previous Next ▶

Processor gallery – Document AI x documentai-sync-v1.0.ipynb + https://console.cloud.google.com/ai/document-ai/processor-library?invt=AcFleA&project=qwiklabs-gcp-03-d5494d0f455b | 5 1 ? s

Free AI Paraphrasing... Transcript of Case St... GCP-LAB Hydra datawarehouse Products

Google Cloud qwiklabs-gcp-03-d5494d0f455b document ai X Search

Document AI / Processor gallery

Overview Overview

Processors My processors

Processor gallery

Custom Processors

Monitoring

Capacity Reservation

Processor gallery

Trainable 9

Type General 3 Specialized 12

Access status Public 17 Private 2

Region eu 18

Features

General Ready to use out-of-the-box processors for general document goals.

Document OCR Identify and extract text in different types of documents

Form Parser Extract form elements such as text and checkboxes

Layout Parser Identify and extract document layouts and chunks

Specialized Schematized processors for domain-specific documents.

8 14°C Clear Search

ENG IN 06:33 PM 30-12-2025

Processor gallery – Document AI x documentai-sync-v1.0.ipynb +

https://console.cloud.google.com/ai/document-ai/processor-library?inv=AcFleA&project=qwiklabs-gcp-03-d5494d0f455b | 5

Free AI Paraphrasing... Transcript of Case St... GCP-LAB Hydra datawarehouse Products

Google Cloud qwiklabs-gcp-03-d5494d0f455b document ai

Document AI / Processor gallery

Overview

Processors

My processors

Processor gallery

Custom Processors

Monitoring

Capacity Reservation

Processor gallery

Search processors

Features

Trainable 9

Type

General 3

Specialized 12

Access status

Public 17

Private 2

Region

eu 18

General

Ready to use out-of-the-box processors for general document goals.

Document OCR

Identify and extract text in different types of documents

Form Parser

Extract form elements such as checkboxes

Specialized

Schematized processors for domain-specific documents.

Create processor

Document OCR

Extract text from documents with world-class accuracy, supporting over 200 languages and 50 languages for handwriting recognition [Learn more](#)

Processor name * ocr-processor

Must start with a letter. Can use letters, numbers, spaces, dashes, and underscores.

Region US (United States) ▾ ?

Advanced options

Create Cancel

8 14°C Clear

Search

Eng IN

06:34 PM 30-12-2025

Google Skills Partner

Automate Data Capture at Scale with Document AI > Process Documents with Python Using the Document AI API

Contents

Process Documents with Python Using the Docur ▾

Lab setup instructions and requirements

Protect your account and

End Lab 01:00:01 ? Time limit

Caution: When you are in the console, do not deviate from the lab instructions. Doing so may cause your account to be blocked. Learn more.

Open Google Cloud console

Username: student-01-75364ecd3f8c1

Password: YFyxc227VWPX

Project ID: qwiklabs-gcp-03-d5494d01

Task 6. Prepare your environment for asynchronous Document AI API calls

In this task you upload the sample JupyterLab notebook to test asynchronous Document AI API calls and copy some sample forms for the lab to Cloud Storage for asynchronous processing.

1. Click the **Terminal** tab to re-open the terminal shell inside the Vertex AI Workbench instance.
2. Create a Cloud Storage bucket for the input documents and copy the sample W2 forms into the bucket:

```
export PROJECT_ID="$(gcloud config get-value core/project)"  
export BUCKET="${PROJECT_ID}"_doc_ai_async  
gsutil mb gs://${BUCKET}  
gsutil -m cp gs://qwiklabs-gcp-03-d5494d0f455b-1abconfig-  
bucket/async/*.* gs://${BUCKET}/input
```

3. In the notebook interface open the JupyterLab notebook called **documentai-async-v1.0.0.ipynb**.

Task 3. Make a synchronous process document request 60/100

Task 4. Run the synchronous Document AI Python code

Task 5. Create a Document AI Document OCR processor

Task 6. Prepare your environment for asynchronous Document AI API calls

Task 7. Make an asynchronous process document request

Congratulations

< Previous

Next >

ocr-processor – Document AI – qwikl ab - JupyterLab

https://2defb48634214276-dot-us-east1.notebooks.googleusercontent.com/lab

Free AI Paraphrasin... Transcript of Case St... GCP-LAB Hydra datawarehouse Products

File Edit View Run Kernel Git Tabs Settings Help

Terminal 1 documentai-sync-v1.0.0.ipynb

```
(base) jupyter@jupyterlab:~$ gsutil cp gs://qwiklabs-gcp-03-d5494d0f455b-labconfig-bucket/health-intake-form.pdf form.pdf
Copying gs://qwiklabs-gcp-03-d5494d0f455b-labconfig-bucket/health-intake-form.pdf...
/ [1 files][624.2 KiB/624.2 KiB]
Operation completed over 1 objects/624.2 KiB.
(base) jupyter@jupyterlab:~$ export PROJECT_ID="$(gcloud config get-value core/project)"
export BUCKET="${PROJECT_ID}"_doc_ai_async
gsutil mb gs://${BUCKET}
gsutil -m cp gs://qwiklabs-gcp-03-d5494d0f455b-labconfig-bucket/async/*.* gs://${BUCKET}/input
Creating gs://qwiklabs-gcp-03-d5494d0f455b_doc_ai_async/...
Copying gs://qwiklabs-gcp-03-d5494d0f455b-labconfig-bucket/async/google_invoice.pdf [Content-Type=application/pdf]...
Copying gs://qwiklabs-gcp-03-d5494d0f455b-labconfig-bucket/async/office-depot-receipt.pdf [Content-Type=application/pdf]...
/ [2/2 files][ 1.5 MiB/ 1.5 MiB] 100% Done
Operation completed over 2 objects/1.5 MiB.
(base) jupyter@jupyterlab:~$
```

Simple 1 \$ 1 ⌂ Terminal 1 3 14°C ENG IN 06:35 PM 30-12-2025

Dashboard Catalog Paths Collections

Free AI Paraphrasin... Transcript of Case St... GCP-LAB Hydra datawarehouse Products

Google Skills Partner

Automate Data Capture at Scale with Document AI > Process Documents with Python Using the Document AI API

Lab setup instructions and requirements

Protect your account and

End Lab 00:56:12 Time limit

Caution: When you are in the console, do not deviate from the lab instructions. Doing so may cause your account to be blocked. Learn more.

Open Google Cloud console

Username: student-01-75364ecd3f8c1

Password: YFyxc227VWPX

Project ID: qwiklabs-gcp-03-d5494d01

Contents

Process Documents with Python Using the Docu

Task 7. Make an asynchronous process document request

Review the Python code for asynchronous Document AI API calls

Take a minute to review the Python code in the `documentai-async-v1.0.0.ipynb` notebook.

The first code cell imports the required libraries.

```
from google.cloud import documentai_v1beta3 as documentai
from google.cloud import storage

import re
import os
import pandas as pd
import simplejson as json
```

← Previous Next →

Contents

Process Documents with Python Using the Docur ▾

Lab setup instructions and requirements

Protect your account and

End Lab 00:55:45 ? Time limit

Caution: When you are in the console, do not deviate from the lab instructions. Doing so may cause your account to be blocked. Learn more.

Open Google Cloud console

Username student-01-75364ecd3f8c1

Password YFyxc227VWPX

Project ID qwiklabs-gcp-03-d5494d01

Run the asynchronous Document AI Python code

Use the sample code provided for you in the Jupyterlab notebook to process documents asynchronously using a Document AI batch processing request.

1. In the second code cell replace the `PROCESSOR_ID` placeholder text with the Processor ID for the **form-parser** processor you created in an earlier step.
2. Select the first cell, click the **Run** menu and then click **Run Selected Cell and All Below** to run all the code in the notebook.
3. As the code cells execute, you can step through the notebook reviewing the code and the comments that explain how the asynchronous request object is created and used.

The notebook will take a minute or two to wait for the asynchronous batch process operation to complete at the **Start the batch (asynchronous) API operation** code cell. While the batch process API call itself is asynchronous the notebook uses the `result` method to force the notebook to wait until the asynchronous call has completed before enumerating and displaying the output data.

If the asynchronous job takes longer than expected and times out you may have to run

8 14°C Clear

Search

8 14°C Clear

ENG IN

06:39 PM 30-12-2025

Task 3. Make a synchronous process document request

Task 4. Run the synchronous Document AI Python code

Task 5. Create a Document AI Document OCR processor

Task 6. Prepare your environment for asynchronous Document AI API calls

Task 7. Make an asynchronous process document request

Congratulations

Google Skills Partner

Automate Data Capture at Scale with Document AI > Process Documents with Python Using the Document AI API

Contents

Process Documents with Python Using the Docu ▾

entity data printed out by the final cell.

4. In the JupyterLab menu click **File** and then click **Save Notebook** to save your progress.

Document processing complete.
Text: FakeDoc M.D.
HEALTH INTAKE FORM
Please **fill** out the questionnaire carefully. The information you provide w
your health profile **and** will be kept confidential.
Date:
Sally
Walker
Name:
9/14/19
...

Check that a document has been processed using the asynchronous Cloud Document API.

Check my progress

Task 3. Make a synchronous process document request 100/100

Task 4. Run the synchronous Document AI Python code

Task 5. Create a Document AI Document OCR processor

Task 6. Prepare your environment for asynchronous Document AI API calls

Task 7. Make an asynchronous process document request

Congratulations

Dashboard

Catalog

Paths

Collections

End Lab 00:54:22 ? Time limit

Open Google Cloud console

Username: student-01-75364ecd3f8c1

Password: YFyxc227VWPX

Project ID: qwiklabs-gcp-03-d5494d01

Previous Next

ocr-processor – Document AI – qwikl documentai-async-v1.0.ipynb + https://2defb48634214276-dot-us-east1.notebooks.googleusercontent.com/lab/tree/documentai-async-v1.0.ipynb

Free AI Paraphrasin... Transcript of Case St... GCP-LAB Hydra datawarehouse Products

File Edit View Run Kernel Git Tabs Settings Help

Terminal 1 documentai-async-v1.0.ipynb documentai-sync-v1.0.ipynb + No Kernel Initializing

Document AI Asynchronous API

This notebook shows you how use Python to make asynchronous calls to the Document AI API

You must replace the `processor_id` variable value in the the second cell with the appropriate value for the Processor ID of the Document AI processor that you want to use. The processor may not support all of the Document AI output properties. Entity data is only returned by processors that use specialized parsers for ex

Select Kernel

Select kernel for: "documentai-async-v1.0.ipynb"

Python 3 (ipykernel) (Local)

Always start the preferred kernel No Kernel Select

```
[ ]: # Import Libraries
from google.cloud import documentai_v1
from google.cloud import storage
from prettytable import PrettyTable
import re
import os
import pandas as pd
```

```
[ ]: # Set your Processor ID
processor_id = "PROCESSOR_ID" # TODO: Replace with a valid Processor ID
```

```
[ ]: # Set your variables
project_id = %system gcloud config get-value core/project
project_id = project_id[0]
location = 'us' # Replace with 'eu' if processor does not use 'us' location
gcs_input_bucket = project_id+"_doc_ai_async" # Bucket name only, no gs:// prefix
gcs_input_prefix = "input/" # Input bucket folder e.g. input/
gcs_output_bucket = project_id+"_doc_ai_async" # Bucket name only, no gs:// prefix
```

Simple 1 \$ 1 No Kernel | Initializing Mode: Command Ln 1, Col 1 documentai-async-v1.0.ipynb 3 14°C ENG IN 06:36 PM 30-12-2025

8 14°C Clear Search

ocr-processor – Document AI – qwikl documentai-async-v1.0.0.ipynb

https://2defb48634214276-dot-us-east1.notebooks.googleusercontent.com/lab/tree/documentai-async-v1.0.0.ipynb

Free AI Paraphrasin... Transcript of Case St... GCP-LAB Hydra datawarehouse Products

File Edit View Run Kernel Git Tabs Settings Help

Terminal 1 documentai-async-v1.0.0.ipynb documentai-sync-v1.0.0.ipynb

Name Modified

- documentai-async-v1.0.0.ipynb 6 sec. ago
- documentai-sync-v1.0.0.ipynb next yr.
- form.pdf now
- notebook_template.ipynb 54 min. ago
- requirements.txt 52 min. ago

Document AI Asynchronous API

This notebook shows you how use Python to make asynchronous calls to the Document AI API

You must replace the `processor_id` variable value in the the second cell with the appropriate value for the Processor ID of the Document AI processor that you want to use. The processor may not support all of the Document AI output properties. Entity data is only returned by processors that use specialized parsers for example.

```
[ ]: # Import Libraries
from google.cloud import documentai_v1beta3 as documentai
from google.cloud import storage
from prettytable import PrettyTable

import re
import os
import pandas as pd

[ ]: # Set your Processor ID
processor_id = "56ba104d66983697" # TODO: Replace with a valid Processor ID

[ ]: # Set your variables
project_id = %system gcloud config get-value core/project
project_id = project_id[0]
location = 'us' # Replace with 'eu' if processor does not use 'us' location
gcs_input_bucket = project_id+"_doc_ai_async" # Bucket name only, no gs:// prefix
gcs_input_prefix = "input/" # Input bucket folder e.g. input/
gcs_output_bucket = project_id+"_doc_ai_async" # Bucket name only, no gs:// prefix
```

Python 3 (ipykernel) (Local)

Simple 1 \$ 2 ⌂ Python 3 (ipykernel) (Local) | Idle Mode: Command 14°C 8 14°C Clear 14°C 30-12-2025 06:37 PM ENG IN

ocr-processor – Document AI – qwikl documentai-async-v1.0.0.ipynb +

https://2defb48634214276-dot-us-east1.notebooks.googleusercontent.com/lab/tree/documentai-async-v1.0.0.ipynb

Free AI Paraphrasin... Transcript of Case St... GCP-LAB Hydra datawarehouse Products

File Edit View Run Kernel Git Tabs Settings Help

+

Name Modified

- documentai-async-v1.0.0.ipynb 53 sec. ago
- documentai-sync-v1.0.0.ipynb next yr.
- form.pdf now
- notebook_template.ipynb 54 min. ago
- requirements.txt 52 min. ago

Terminal 1 documentai-async-v1.0.0.ipynb documentai-sync-v1.0.0.ipynb +

```
Fetched file 1:output/16038490229511477023/0/google_invoice-0.json
Text Data:
Google
John Smith
Bill To:
Jane Smith,
1600 Amphitheatre Pkwy
Mountain View, CA 94043
INVOICE
#23413561D
Date:
Sep 24, 2019
Due Date:
Sep 30, 2019
Balance Due:
$4,647.68
Item
Quantity
Rate
Amount
12 ft HDMI cable
12
27" Computer Monitor
12
Ergonomic Keyboard
12
Optical mouse
12
Laptop
12
22222
$9.99
$399.99
```

Python 3 (ipykernel) (Local)

Simple 1 \$ 2 ⚡ Python 3 (ipykernel) (Local) | Idle Mode: Edit 14°C ENG IN 30-12-2025 06:39 PM

Search

8 14°C Clear

ocr-processor – Document AI – qwikl documentai-async-v1.0.0.ipynb +

https://2defb48634214276-dot-us-east1.notebooks.googleusercontent.com/lab/tree/documentai-async-v1.0.0.ipynb

Free AI Paraphrasin... Transcript of Case St... GCP-LAB Hydra datawarehouse Products

File Edit View Run Kernel Git Tabs Settings Help

+

Name Modified

- documentai-async-v1.0.0.ipynb 53 sec. ago
- documentai-sync-v1.0.0.ipynb next yr.
- form.pdf now
- notebook_template.ipynb 54 min. ago
- requirements.txt 52 min. ago

Terminal 1 documentai-async-v1.0.0.ipynb documentai-sync-v1.0.0.ipynb +

SALE
950-1-2020-958724-20.6.2
328374 MAT, COOL, LAPTOP
39.99 SS
Subtotal:
Sales Tax:
951 39.99
3.60
Total:
Visa 5047:
ano 43.59
AUTH CODE 076243
TDS Chip Read
43.59
beileitea on enuoy !!
AID A00000000310101 CITI VISA xosqeniph
TVR 0800008000 silque soitto ol aysb. de midliw
not & ini beneqonu lis
CVS PIN Verified
bns siswiloa
veolonroe)
go
aldsm
Shop online at www.officedepot.com door

ଓগৱান
WE WANT TO HEAR FROM YOU!
Visit survey.officedepot.com and enter the survey code below:
15QQ JNFC W3MH
isteb

[11]: # Display entity data from asynchronous output JSON files

Simple 1 \$ 2 Python 3 (ipykernel) (Local) | Idle Mode: Edit 3 14°C ENG IN 06:39 PM 30-12-2025

Search

14°C Clear

Search

06:39 PM 30-12-2025

ocr-processor – Document AI – qwikl documentai-async-v1.0.0.ipynb +

https://2defb48634214276-dot-us-east1.notebooks.googleusercontent.com/lab/tree/documentai-async-v1.0.0.ipynb

Free AI Paraphrasin... Transcript of Case St... GCP-LAB Hydra datawarehouse Products

File Edit View Run Kernel Git Tabs Settings Help

+

Name Modified

- documentai-async-v1.0.0.ipynb 53 sec. ago
- documentai-sync-v1.0.0.ipynb next yr.
- form.pdf now
- notebook_template.ipynb 54 min. ago
- requirements.txt 52 min. ago

Terminal 1 documentai-async-v1.0.0.ipynb documentai-sync-v1.0.0.ipynb +

```
else:  
    print('No entity data returned by the Document AI processor for file'+blob.name)  
else:  
    print(f"Skipping non-supported file type {blob.name}")  
  
Fetched file 1:output/16038490229511477023/0/google_invoice-0.json  
+-----+  
| Type | Value | Confidence |  
+-----+  
+-----+  
+-----+  
Fetched file 2:output/16038490229511477023/1/office-depot-receipt-0.json  
+-----+  
| Type | Value | Confidence |  
+-----+  
+-----+  
+-----+  
[ ]:  
[ ]:
```

Python 3 (ipykernel) (Local)

Simple 1 \$ 2 ⚡ Python 3 (ipykernel) (Local) | Idle Mode: Edit 14°C LN 1, Col 1 documentai-async-v1.0.0.ipynb 3 ENG IN 06:39 PM 30-12-2025 Clear

Search

8 14°C Clear