<https://cloud.google.com/vision/docs/quickstart-client-libraries>

<https://cloud.google.com/vision/docs/ocr>

# 텍스트 감지 샘플(Python)

## (1st uri는 label detection이라서 환경 설정 부분만 참고, code는 2nd uri를 참조함)

## Before you begin

1. Select or create a GCP project.

[GO TO THE MANAGE RESOURCES PAGE](https://console.cloud.google.com/cloud-resource-manager)

1. Make sure that billing is enabled for your project.

[LEARN HOW TO ENABLE BILLING](https://cloud.google.com/billing/docs/how-to/modify-project)

1. Enable the Cloud Vision API.

[ENABLE THE API](https://console.cloud.google.com/flows/enableapi?apiid=vision-json.googleapis.com)

1. Set up authentication:
   1. In the GCP Console, go to the **Create service account key** page.

[GO TO THE CREATE SERVICE ACCOUNT KEY PAGE](https://console.cloud.google.com/apis/credentials/serviceaccountkey)

* 1. From the **Service account** drop-down list, select **New service account**.
  2. In the **Service account name** field, enter a name .
  3. From the **Role** drop-down list, select **Project** > **Owner**.

**Note**: The **Role** field authorizes your service account to access resources. You can view and change this field later by using [GCP Console](https://console.cloud.google.com/). If you are developing a production app, specify more granular permissions than **Project > Owner**. For more information, see [granting roles to service accounts](https://cloud.google.com/iam/docs/granting-roles-to-service-accounts).

* 1. Click **Create**. A JSON file that contains your key downloads to your computer.

1. Set the environment variable **GOOGLE\_APPLICATION\_CREDENTIALS** to the file path of the JSON file that contains your service account key. This variable only applies to your current shell session, so if you open a new session, set the variable again.

▸

**Example:** Linux or macOS

▾

**Example:** Windows

Replace **[PATH]** with the file path of the JSON file that contains your service account key, and **[FILE\_NAME]** with the filename.

With PowerShell:

$env:GOOGLE\_APPLICATION\_CREDENTIALS="[PATH]"

For example:

$env:GOOGLE\_APPLICATION\_CREDENTIALS="C:\Users\username\Downloads\[FILE\_NAME].json"

With command prompt:

set GOOGLE\_APPLICATION\_CREDENTIALS=[PATH]

## Install the client library

### PYTHON

### RUBY

For more on setting up your Python development environment, refer to the [Python Development Environment Setup Guide](https://cloud.google.com/python/setup).

pip install --upgrade google-cloud-vision

## Label detection

Now you can use the Vision API to request information from an image, such as label detection. Run the following code to perform your first image label detection request:

### PYTHON

### RUBY

Before trying this sample, follow the Python setup instructions in the [Vision API Quickstart Using Client Libraries](https://cloud.google.com/vision/docs/quickstart-client-libraries). For more information, see the [Vision API Python API reference documentation](https://googleapis.github.io/google-cloud-python/latest/vision/index.html).

[VIEW ON GITHUB](https://github.com/GoogleCloudPlatform/python-docs-samples/blob/master/vision/cloud-client/quickstart/quickstart.py) [FEEDBACK](https://cloud.google.com/vision/docs/quickstart-client-libraries)

import io  
import os  
  
# Imports the Google Cloud client library  
from google.cloud import vision  
from google.cloud.vision import types  
  
# Instantiates a client  
client = vision.ImageAnnotatorClient()  
  
# The name of the image file to annotate  
file\_name = os.path.join(  
    os.path.dirname(\_\_file\_\_),  
    'resources/wakeupcat.jpg')  
  
# Loads the image into memory  
with io.open(file\_name, 'rb') as image\_file:  
    content = image\_file.read()  
  
image = types.Image(content=content)  
  
# Performs label detection on the image file  
response = client.label\_detection(image=image)  
labels = response.label\_annotations  
  
print('Labels:')  
for label in labels:  
    print(label.description)

Congratulations! You've sent your first request to the Vision API.

IT WORKED!I GOT AN ERROR

## Clean up

To avoid incurring charges to your Google Cloud Platform account for the resources used in this quickstart:

* Use the [GCP Console](https://console.cloud.google.com/) to delete your project if you do not need it.

<https://cloud.google.com/vision/docs/detecting-text>

# 텍스트 감지 샘플(Java)

**텍스트 감지**는 광학 문자 인식을 수행하는 기능입니다. 이미지 내에서 텍스트를 감지하고 추출하며, 광범위한 언어를 지원합니다. 또한 자동 언어 식별이 제공됩니다.

## 로컬 이미지의 텍스트 감지

### 자바

Vision API 클라이언트 설치 및 생성에 대한 자세한 내용은 [Vision API 클라이언트 라이브러리](https://cloud.google.com/vision/docs/reference/libraries)를 참조하세요.

[GITHUB에서 보기](https://github.com/GoogleCloudPlatform/java-docs-samples/blob/master/vision/cloud-client/src/main/java/com/example/vision/Detect.java) [의견 보내기](https://cloud.google.com/vision/docs/detecting-text)

public static void detectText(String filePath, PrintStream out) throws Exception, IOException {  
  List<AnnotateImageRequest> requests = new ArrayList<>();  
  
  ByteString imgBytes = ByteString.readFrom(new FileInputStream(filePath));  
  
  Image img = Image.newBuilder().setContent(imgBytes).build();  
  Feature feat = Feature.newBuilder().setType(Type.TEXT\_DETECTION).build();  
  AnnotateImageRequest request =  
      AnnotateImageRequest.newBuilder().addFeatures(feat).setImage(img).build();  
  requests.add(request);  
  
  try (ImageAnnotatorClient client = ImageAnnotatorClient.create()) {  
    BatchAnnotateImagesResponse response = client.batchAnnotateImages(requests);  
    List<AnnotateImageResponse> responses = response.getResponsesList();  
  
    for (AnnotateImageResponse res : responses) {  
      if (res.hasError()) {  
        out.printf("Error: %s\n", res.getError().getMessage());  
        return;  
      }  
  
      // For full list of available annotations, see http://g.co/cloud/vision/docs  
      for (EntityAnnotation annotation : res.getTextAnnotationsList()) {  
        out.printf("Text: %s\n", annotation.getDescription());  
        out.printf("Position : %s\n", annotation.getBoundingPoly());  
      }  
    }  
  }  
}

## 원격 이미지의 텍스트 감지

Vision API는 사용자의 편의를 위해 Google Cloud Storage 또는 웹에 위치한 이미지 파일을 대상으로 직접 텍스트 감지를 수행할 수 있으며, 이러한 경우 요청 본문에 이미지 파일의 내용을 포함하여 보낼 필요가 없습니다.

### 자바

Vision API 클라이언트 설치 및 생성에 대한 자세한 내용은 [Vision API 클라이언트 라이브러리](https://cloud.google.com/vision/docs/reference/libraries)를 참조하세요.

[GITHUB에서 보기](https://github.com/GoogleCloudPlatform/java-docs-samples/blob/master/vision/cloud-client/src/main/java/com/example/vision/Detect.java) [의견 보내기](https://cloud.google.com/vision/docs/detecting-text)

public static void detectTextGcs(String gcsPath, PrintStream out) throws Exception, IOException {  
  List<AnnotateImageRequest> requests = new ArrayList<>();  
  
  ImageSource imgSource = ImageSource.newBuilder().setGcsImageUri(gcsPath).build();  
  Image img = Image.newBuilder().setSource(imgSource).build();  
  Feature feat = Feature.newBuilder().setType(Type.TEXT\_DETECTION).build();  
  AnnotateImageRequest request =  
      AnnotateImageRequest.newBuilder().addFeatures(feat).setImage(img).build();  
  requests.add(request);  
  
  try (ImageAnnotatorClient client = ImageAnnotatorClient.create()) {  
    BatchAnnotateImagesResponse response = client.batchAnnotateImages(requests);  
    List<AnnotateImageResponse> responses = response.getResponsesList();  
  
    for (AnnotateImageResponse res : responses) {  
      if (res.hasError()) {  
        out.printf("Error: %s\n", res.getError().getMessage());  
        return;  
      }  
  
      // For full list of available annotations, see http://g.co/cloud/vision/docs  
      for (EntityAnnotation annotation : res.getTextAnnotationsList()) {  
        out.printf("Text: %s\n", annotation.getDescription());  
        out.printf("Position : %s\n", annotation.getBoundingPoly());  
      }  
    }  
  }  
}

*이 페이지가 도움이 되었나요? 평가를 부탁드립니다.*

<https://cloud.google.com/vision/docs/libraries>

# Vision API 클라이언트 라이브러리

이 페이지에서는 새로운 Cloud Vision API용 Cloud 클라이언트 라이브러리를 시작하는 방법을 보여줍니다. Cloud API용 클라이언트 라이브러리 및 이전 Google API 클라이언트 라이브러리에 대한 자세한 내용은 [클라이언트 라이브러리 설명](https://cloud.google.com/apis/docs/client-libraries-explained)을 참조하세요.

**Beta**

This is a beta release of the Cloud Client Libraries for the Cloud Vision API. These libraries might be changed in backward-incompatible ways and are not subject to any SLA or deprecation policy.

### 클라이언트 라이브러리 설치

### 자바

If you are using [Maven](https://maven.apache.org/), add this to your pom.xml file:

<dependency>  
  <groupId>com.google.cloud</groupId>  
  <artifactId>google-cloud-vision</artifactId>  
  <version>1.51.0</version>  
</dependency>

If you are using [Gradle](https://gradle.org/" \t "_blank), add this to your dependencies:

compile 'com.google.cloud:google-cloud-vision:1.51.0'

If you are using [SBT](https://www.scala-sbt.org/), add this to your dependencies:

libraryDependencies += "com.google.cloud" % "google-cloud-vision" % "1.51.0"

If you're using IntelliJ or Eclipse, you can add client libraries to your project using these IDE plugins:

* [Cloud Tools for IntelliJ](https://cloud.google.com/tools/intellij/docs/client-libraries)
* [Cloud Tools for Eclipse](https://cloud.google.com/eclipse/docs/libraries)

The plugins provide additional functionality, such as key management for service accounts. Refer to each plugin's documentation for details.

**Note:** Cloud Java client libraries do not currently support Android.

### 인증 설정

클라이언트 라이브러리를 실행하려면 우선 서비스 계정을 만들고 환경 변수를 설정하여 [인증](https://cloud.google.com/docs/authentication/production)을 설정해야 합니다.

### GCP CONSOLE

1. In the GCP Console, go to the **Create service account key** page.

[GO TO THE CREATE SERVICE ACCOUNT KEY PAGE](https://console.cloud.google.com/apis/credentials/serviceaccountkey)

1. From the **Service account** drop-down list, select **New service account**.
2. In the **Service account name** field, enter a name.
3. Don't select a value from the **Role** drop-down list. No role is required to access this service.
4. Click **Create**. A note appears, warning that this service account has no role.
5. Click **Create without role**. A JSON file that contains your key downloads to your computer.

### COMMAND Line

You can run the following commands using the [Cloud SDK](https://cloud.google.com/sdk/) on your local machine, or within [Cloud Shell](https://cloud.google.com/shell/).

1. Create the service account. Replace **[NAME]** with your desired service account name.

gcloud iam service-accounts create [NAME]

1. Generate the key file. Replace **[FILE\_NAME]** with a name for the key file.

gcloud iam service-accounts keys create [FILE\_NAME].json --iam-account [NAME]@[PROJECT\_ID].iam.gserviceaccount.com

Provide authentication credentials to your application code by setting the environment variable **GOOGLE\_APPLICATION\_CREDENTIALS**. Replace **[PATH]** with the file path of the JSON file that contains your service account key, and **[FILE\_NAME]** with the filename. This variable only applies to your current shell session, so if you open a new session, set the variable again.

### LINUX OR MACOS

### WINDOWS

export GOOGLE\_APPLICATION\_CREDENTIALS="[PATH]"

For example:

export GOOGLE\_APPLICATION\_CREDENTIALS="/home/user/Downloads/[FILE\_NAME].json"

### WINDOWS

With PowerShell:

$env:GOOGLE\_APPLICATION\_CREDENTIALS="[PATH]"

For example:

$env:GOOGLE\_APPLICATION\_CREDENTIALS="C:\Users\username\Downloads\[FILE\_NAME].json"

With command prompt:

set GOOGLE\_APPLICATION\_CREDENTIALS=[PATH]

### 클라이언트 라이브러리 사용

다음 예에서는 클라이언트 라이브러리를 사용하는 방법을 보여 줍니다.

### 자바

Cloud Vision API 클라이언트 라이브러리에 대한 자세한 내용은 [자바 API 참조 문서](https://googlecloudplatform.github.io/google-cloud-java)를 확인하세요.

[GITHUB에서 보기](https://github.com/GoogleCloudPlatform/java-docs-samples/blob/master/vision/cloud-client/src/main/java/com/example/vision/QuickstartSample.java) [의견 보내기](https://cloud.google.com/vision/docs/libraries)

// Imports the Google Cloud client library  
  
import com.google.cloud.vision.v1.AnnotateImageRequest;  
import com.google.cloud.vision.v1.AnnotateImageResponse;  
import com.google.cloud.vision.v1.BatchAnnotateImagesResponse;  
import com.google.cloud.vision.v1.EntityAnnotation;  
import com.google.cloud.vision.v1.Feature;  
import com.google.cloud.vision.v1.Feature.Type;  
import com.google.cloud.vision.v1.Image;  
import com.google.cloud.vision.v1.ImageAnnotatorClient;  
import com.google.protobuf.ByteString;  
import java.nio.file.Files;  
import java.nio.file.Path;  
import java.nio.file.Paths;  
import java.util.ArrayList;  
import java.util.List;  
  
public class QuickstartSample {  
  public static void main(String... args) throws Exception {  
    // Instantiates a client  
    try (ImageAnnotatorClient vision = ImageAnnotatorClient.create()) {  
  
      // The path to the image file to annotate  
      String fileName = "./resources/wakeupcat.jpg";  
  
      // Reads the image file into memory  
      Path path = Paths.get(fileName);  
      byte[] data = Files.readAllBytes(path);  
      ByteString imgBytes = ByteString.copyFrom(data);  
  
      // Builds the image annotation request  
      List<AnnotateImageRequest> requests = new ArrayList<>();  
      Image img = Image.newBuilder().setContent(imgBytes).build();  
      Feature feat = Feature.newBuilder().setType(Type.LABEL\_DETECTION).build();  
      AnnotateImageRequest request = AnnotateImageRequest.newBuilder()  
          .addFeatures(feat)  
          .setImage(img)  
          .build();  
      requests.add(request);  
  
      // Performs label detection on the image file  
      BatchAnnotateImagesResponse response = vision.batchAnnotateImages(requests);  
      List<AnnotateImageResponse> responses = response.getResponsesList();  
  
      for (AnnotateImageResponse res : responses) {  
        if (res.hasError()) {  
          System.out.printf("Error: %s\n", res.getError().getMessage());  
          return;  
        }  
  
        for (EntityAnnotation annotation : res.getLabelAnnotationsList()) {  
          annotation.getAllFields().forEach((k, v) ->  
              System.out.printf("%s : %s\n", k, v.toString()));  
        }  
      }  
    }  
  }  
}

### 추가 자료

### 자바

* [API 참조 문서](https://googlecloudplatform.github.io/google-cloud-java)
* [소스 코드](https://github.com/GoogleCloudPlatform/google-cloud-java)
* [GitHub 문제 추적기](https://github.com/GoogleCloudPlatform/google-cloud-java/issues)
* [Stack Overflow](http://stackoverflow.com/search?q=%5Bgoogle-cloud-vision%5D+%5Bjava%5D)

*이 페이지가 도움이 되었나요? 평가를 부탁드립니다.*

<https://cloud.google.com/apis/docs/client-libraries-explained>

* [Cloud APIs](https://cloud.google.com/apis/)

# Client Libraries Explained

While you can use [Google Cloud APIs](https://cloud.google.com/apis/docs/overview) by making direct HTTP requests to the server (or RPC calls where available), we provide client library code for all our Cloud APIs that makes it easier to access them from your favorite languages. This document explains the different types of client libraries we provide for Cloud APIs. You can find out more about the available libraries for your product or language of choice in the product or language’s documentation.

## Google Cloud Client Libraries

Google Cloud Client Libraries use our latest client library model and are our recommended option for accessing Cloud APIs programmatically, where available. Cloud Client Libraries:

* Provide idiomatic, generated or hand-written code in each language, making the Cloud API simple and intuitive to use.
* Handle all the low-level details of communication with the server, including [authenticating with Google](https://cloud.google.com/docs/authentication).
* Can be installed using familiar package management tools such as npm and pip.
* In some cases, give you performance benefits by using gRPC. You can find out more in the [gRPC APIs](https://cloud.google.com/apis/docs/client-libraries-explained#grpc_apis) section below.

You can find installation instructions and reference material for the appropriate Cloud Client Library on your chosen [Cloud API](https://cloud.google.com/apis/docs/overview)’s Client Libraries page.

For almost all our supported languages, you can also download a single client library that provides an interface to all supported Cloud APIs. You can find links to get started with these and their reference documentation on our [Cloud Client Libraries page](https://cloud.google.com/apis/docs/cloud-client-libraries).

## Google API Client Libraries

A number of Google Cloud APIs do not yet have Google Cloud Client Libraries available in all languages. If you want to use one of these APIs and there is no Cloud Client Library for your preferred language, you can still use an older version of our client libraries called [Google API Client Libraries](https://developers.google.com/api-client-library/). These libraries:

* Provide access to the API’s REST interface only; gRPC is not supported.
* Have autogenerated interface code that may not be as idiomatic as our newer libraries.
* Handle all the low-level details of communication with the server, including [authenticating with Google](https://cloud.google.com/docs/authentication).
* Can be installed using familiar package management tools such as npm and pip.

You'll find links to these libraries on the relevant Cloud API's Client Libraries page, where necessary.

## Using the Firebase mobile platform

Firebase is the Google-wide solution for building applications on mobile devices. It offers an SDK with client code that lets you access mobile-relevant Cloud APIs directly from iOS, Android, and Web apps. Visit the [Firebase documentation](https://firebase.google.com/docs) for more information on which Cloud APIs are supported and how to get started with Firebase.

## Using your own client code

Generally we recommend using our client libraries to access Cloud APIs. However, if you are an experienced developer and our client libraries don't meet your specific needs, you can write your own custom code to access the service's lower-level service APIs directly.

### REST/HTTP APIs

All Cloud APIs expose a simple traditional JSON/REST interface. If you need to write your own custom code to directly access the REST API using a third-party HTTP client library of your choice, you can find out more about how Cloud APIs work with different HTTP versions and implementations in our [HTTP Guidelines](https://cloud.google.com/apis/docs/http).

### gRPC APIs

gRPC is a language-neutral, platform-neutral, open source, remote procedure call (RPC) system initially developed at Google: you can find out much more about it at [grpc.io](http://www.grpc.io/). gRPC-enabled Cloud APIs generally have both REST and RPC interfaces, so rather than just using JSON over HTTP to talk to the REST interface, gRPC-enabled API clients can also use [protocol buffers](https://developers.google.com/protocol-buffers/) and gRPC over HTTP2 to talk to the RPC interface. You can find out if an API is gRPC-enabled by checking its APIs and Reference section.

If a Cloud API is gRPC-enabled, you can generate your own gRPC client libraries for it in any gRPC-supported language. To do this, you’ll need the API’s [protocol buffers](https://developers.google.com/protocol-buffers/) service definition (typically available from [our repository on GitHub](https://github.com/googleapis/googleapis)). You can then follow the instructions for your preferred language on [grpc.io](http://www.grpc.io/) to generate and use your client.

If you don't want to generate your own gRPC code, you can still benefit from gRPC: a growing number of Cloud Client Libraries for gRPC-enabled APIs use gRPC “under the hood” to communicate with Google’s servers. This is significantly more efficient in terms of throughput and CPU usage — accessing an API using gRPC can increase throughput per CPU by as much as a factor of 10 compared to the JSON REST API. We plan to upgrade as many Cloud Client Libraries as possible to gRPC; in the meantime, you’ll still get all the other advantages of our client library code.

## Other ways to access Cloud APIs

If you don’t want to access an API programmatically, you can access the same functionality using the tools in [Google Cloud SDK](https://cloud.google.com/sdk/), or via the [Google Cloud Platform Console](https://console.cloud.google.com/).

*이 페이지가 도움이 되었나요? 평가를 부탁드립니다.*