

[AI & Machine Learning Products](https://cloud.google.com/products/machine-learning/) (<https://cloud.google.com/products/machine-learning/>)

[Cloud Vision API](https://cloud.google.com/vision/) (<https://cloud.google.com/vision/>)

[Documentation](https://cloud.google.com/vision/docs/) (<https://cloud.google.com/vision/docs/>) [Guides](#)

Using Cloud Vision with Spring framework

[Spring Cloud GCP](https://spring.io/projects/spring-cloud-gcp) (<https://spring.io/projects/spring-cloud-gcp>) offers convenient libraries to interface with Cloud Vision from a Spring application. These libraries include [Auto-Configuration and helper classes](#)

(<https://docs.spring.io/spring-boot/docs/current/reference/html/using-boot-auto-configuration.html>) and Spring Boot Template classes to allow developers to get started with Cloud Vision quickly.

If you're already familiar with the [Spring Framework](https://spring.io/projects/spring-framework) (<https://spring.io/projects/spring-framework>), then Spring Cloud Vision can make it easier to work with the Vision API in your application and reduce the amount of code that you need to write.

This page explains how to add Spring Cloud Vision to a Java application. For detailed information about the module, see the [Spring Cloud Vision reference](#)

(<https://cloud.spring.io/spring-cloud-static/spring-cloud-gcp/1.2.0.RELEASE/reference/html/#google-cloud-vision>)

Dependency setup

To begin using this library, add the `spring-cloud-gcp-starter-vision` artifact to your project.

Maven coordinates, using Spring Cloud GCP BOM:

[vision/spring-framework/pom.xml](https://github.com/GoogleCloudPlatform/java-docs-samples/blob/master/vision/spring-framework/pom.xml)

(<https://github.com/GoogleCloudPlatform/java-docs-samples/blob/master/vision/spring-framework/pom.xml>)

`GOOGLECLOUDPLATFORM/JAVA-DOCS-SAMPLES/BLOB/MASTER/VISION/SPRING-FRAMEWORK/POM.XML`

```
<dependencyManagement>
  <dependencies>
    <dependency>
      <groupId>org.springframework.cloud</groupId>
      <artifactId>spring-cloud-gcp-dependencies</artifactId>
      <version>1.2.1.RELEASE</version>
```

```
<type>pom</type>
<scope>import</scope>
</dependency>
</dependencies>
</dependencyManagement>
```

[vision/spring-framework/pom.xml](https://github.com/GoogleCloudPlatform/java-docs-samples/blob/master/vision/spring-framework/pom.xml)

(<https://github.com/GoogleCloudPlatform/java-docs-samples/blob/master/vision/spring-framework/pom.xml>)

OGLECLOUDPLATFORM/JAVA-DOCS-SAMPLES/BLOB/MASTER/VISION/SPRING-FRAMEWORK/POM.XML)

```
<dependency>
  <groupId>org.springframework.cloud</groupId>
  <artifactId>spring-cloud-gcp-starter-vision</artifactId>
</dependency>
```

You must also [create a service account](#)

(<https://cloud.google.com/docs/authentication/getting-started>) and use the service account key to authenticate with Google Cloud.

For more information, see the instructions for [setting up a Java development environment](#) (<https://cloud.google.com/java/docs/setup>). You do not need to install the Google Cloud Client Library for Java; the Spring Boot starter installs the client library automatically.

Image analysis

After configuring the Spring Cloud GCP Vision dependencies on your classpath, you can immediately begin processing your images by getting an instance of `CloudVisionTemplate` using [Spring dependency injection](#)

(<https://docs.spring.io/spring-boot/docs/current/reference/html/using-boot-spring-beans-and-dependency-injection.html>)

[vision/spring-framework/src/main/java/com/example/vision/VisionController.java](https://github.com/GoogleCloudPlatform/java-docs-samples/blob/master/vision/spring-framework/src/main/java/com/example/vision/VisionController.java)

(<https://github.com/GoogleCloudPlatform/java-docs-samples/blob/master/vision/spring-framework/src/main/java/com/example/vision/VisionController.java>)

ER/VISION/SPRING-FRAMEWORK/SRC/MAIN/JAVA/COM/EXAMPLE/VISION/VISIONCONTROLLER.JAVA)

```
@Autowired
private CloudVisionTemplate cloudVisionTemplate;
```

The `CloudVisionTemplate` is a wrapper around the Cloud Vision Client Libraries and lets you process images easily through the Cloud Vision API. For more information about the `CloudVisionTemplate` features, see the [Cloud Vision template reference page](https://cloud.spring.io/spring-cloud-static/spring-cloud-gcp/1.2.0.RELEASE/reference/html/#google-cloud-vision) (<https://cloud.spring.io/spring-cloud-static/spring-cloud-gcp/1.2.0.RELEASE/reference/html/#google-cloud-vision>)

The following sections contain code samples for common use cases of the `CloudVisionTemplate`. All code snippets come from the [Spring and Cloud Vision sample application](https://github.com/GoogleCloudPlatform/java-docs-samples/tree/master/vision/spring-framework) (<https://github.com/GoogleCloudPlatform/java-docs-samples/tree/master/vision/spring-framework>).

Getting the classification labels for an image

The code below extracts the classification labels for an image, providing you with general descriptions of image content.

```
vision/spring-framework/src/main/java/com/example/vision/VisionController.java
(https://github.com/GoogleCloudPlatform/java-docs-samples/blob/master/vision/spring-framework/src/main/java/com/example/vision/VisionController.java)
```

```
ER/VISION/SPRING-FRAMEWORK/SRC/MAIN/JAVA/COM/EXAMPLE/VISION/VISIONCONTROLLER.JAVA)
```

```
AnnotateImageResponse response = this.cloudVisionTemplate.analyzeImage(
    this.resourceLoader.getResource(imageUrl), Type.LABEL_DETECTION);

Map<String, Float> imageLabels =
    response.getLabelAnnotationsList()
        .stream()
        .collect(
            Collectors.toMap(
                EntityAnnotation::getDescription,
                EntityAnnotation::getScore,
                (u, v) -> {
                    throw new IllegalStateException(String.format("Duplicate key %s",
                        u));
                },
                LinkedHashMap::new));
```

Extracting the Text In an Image

The code sample below describes another common operation of extracting the text from an image.

```
vision/spring-framework/src/main/java/com/example/vision/VisionController.java  
(https://github.com/GoogleCloudPlatform/java-docs-samples/blob/master/vision/spring-  
framework/src/main/java/com/example/vision/VisionController.java)
```

```
ER/VISION/SPRING-FRAMEWORK/SRC/MAIN/JAVA/COM/EXAMPLE/VISION/VISIONCONTROLLER.JAVA)
```

```
String textFromImage = this.cloudVisionTemplate.extractTextFromImage(  
    this.resourceLoader.getResource(imageUrl));  
return "Text from image: " + textFromImage;
```

What's next

- [Get started with Spring Cloud Google Cloud](https://cloud.spring.io/spring-cloud-static/spring-cloud-gcp/1.2.0.RELEASE/reference/html/#getting-started)
(https://cloud.spring.io/spring-cloud-static/spring-cloud-gcp/1.2.0.RELEASE/reference/html/#getting-started)
- [Learn more about using Spring Cloud Vision in your applications](https://cloud.spring.io/spring-cloud-static/spring-cloud-gcp/1.2.0.RELEASE/reference/html/#google-cloud-vision)
(https://cloud.spring.io/spring-cloud-static/spring-cloud-gcp/1.2.0.RELEASE/reference/html/#google-cloud-vision)
- [File a GitHub issue](https://github.com/spring-cloud/spring-cloud-gcp/issues) (https://github.com/spring-cloud/spring-cloud-gcp/issues) to report a bug or ask a question about the module.
- Get more information about [Spring Framework support on Google Cloud](https://cloud.google.com/java/docs/reference/spring)
(https://cloud.google.com/java/docs/reference/spring).
- Try a codelab to [deploy and run an application that uses Spring Cloud Google Cloud](https://codelabs.developers.google.com/spring/)
(https://codelabs.developers.google.com/spring/).

Except as otherwise noted, the content of this page is licensed under the [Creative Commons Attribution 4.0 License](https://creativecommons.org/licenses/by/4.0/) (https://creativecommons.org/licenses/by/4.0/), and code samples are licensed under the [Apache 2.0 License](https://www.apache.org/licenses/LICENSE-2.0) (https://www.apache.org/licenses/LICENSE-2.0). For details, see our [Site Policies](https://developers.google.com/terms/site-policies) (https://developers.google.com/terms/site-policies). Java is a registered trademark of Oracle and/or its affiliates.

Last updated January 16, 2020.