

Java



Manual



HOME > SDK > JAVA > GETTING STARTED > MANUAL

Manually integrate Nutrient Java SDK for PDF processing

This guide helps you gain precise control over Nutrient Java SDK integration by manually adding files and managing all dependencies for powerful PDF functionalities. By the end, you'll be able to load a PDF document into your application using the Nutrient Java API.

Use the manual integration method only when absolutely necessary; otherwise, we recommend referring to the [automatic method of integrating Nutrient Java library](#) for detailed instructions.

Select the build system you're using:

GRADLE

MAVEN

If you're using the [Gradle](#) build system, a good example of how to integrate the Nutrient Java library can be seen in the Catalog example distributed in the downloaded ZIP.

Follow the instructions below for manual Gradle integration:

Create a folder within your project named `libs`.

[Download the latest release.](#)

ASK AI

In the downloaded ZIP file, there will be a JAR file, `pspdfkit-{$LATEST_VERSIONS.JAVA %}.jar`. Copy this JAR file into the `libs` folder you just created.

In your project's `build.gradle` folder, add a dependency to the copied JAR file (replace the wildcards with the version number):

```
dependencies {  
    ...  
    implementation files('libs/pspdfkit-1.8.0.jar')  
  
    // Required by PSPDFKit.  
    implementation 'org.json:json:20180813'  
    implementation 'org.apache.httpcomponents:httpclient:4.5.9'  
    ...  
}
```

Now your Gradle project will depend upon the JAR file. More information about declaring dependencies can be found in the [Gradle documentation](#).

Add the OCR dependency (optional). This is only necessary if you plan to use the OCR feature. To do that, a separate package, `libraries-java-ocr`, is required. [Download the latest release](#), which contains the OCR package.

Using the Nutrient Java library

Now that the Nutrient library is added, you can use the SDK in your application. You need to initialize the SDK prior to calling any other method. Use the

`initializeTrial` method if you haven't yet bought a license key, or the

`initialize` method if you have a key:

JAVA

KOTLIN

```
public void initializePSPDFKit() throws PSPDFKitInitializeException {  
    PSPDFKit.initializeTrial();  
}
```

Next, load a document:

JAVA

KOTLIN

```
File file = new File("path/to/document.pdf");  
PdfDocument document = PdfDocument.open(new FileDataProvider(file));
```

Once you have a document loaded, you can use the rest of the library. See the [API docs](#) and [guides](#) for more information, or try out the Catalog examples (see below).

Building and running the Catalog app

The Nutrient library comes with an example application called the Catalog app, which contains many useful examples for exploring the different features of Nutrient. You can find the app in the `Example` folder of the [SDK ZIP file](#).

To build and run the Catalog app, follow the instructions below.

LINUX

MAC

WINDOWS

Download the latest release.

Extract `PSPDFKit-Libraries-java_binary-{version}.zip` to your desired location and `cd` into the root of its directory.

`cd` into the Catalog (`cd catalog`).

Build and run the Catalog application:

```
./gradlew runJava
```



Copyright 2025 Nutrient. All rights reserved.