

# Lens Android Long Receipt

## Adding Lens SDK to your project

1. Add the Maven repository details to the `dependencyResolutionManagement.repositories` section of your `settings.gradle` as shown in this minimalistic example:

```
dependencyResolutionManagement {
    repositories {
        google()
        mavenCentral()

        // Maven repository for VeryfiLens
        maven {
            url "https://nexus.veryfi.com/repository/maven-releases/"
            credentials {
                username = 'USERNAME'
                password = 'PASSWORD'
            }
            authentication {
                basic(BasicAuthentication)
            }
        }
    }
}
```

1. Add Veryfi Lens to the dependencies in your `build.gradle (:app)` file (replace "X.X.X" with the Lens Long Receipts SDK version you're currently using):

```
implementation 'com.veryfi.lens:veryfi-lens-long-receipts-sdk:X.X.X'
```

## Configuring your project to use Lens Long Receipts SDK

1. Ensure your `build.gradle (:app)` file includes at a minimum the following configuration, plus any other required settings

```

plugins {
    id 'kotlin-kapt'
}

android {
    defaultConfig {
        minSdkVersion 21

        ndk {
            // Specifies the ABI configurations of your native app
            abiFilters "armeabi-v7a", "arm64-v8a", "x86", "x86_64"
        }

        javaCompileOptions {
            annotationProcessorOptions {
                arguments += [
                    "room.schemaLocation":"$projectDir/schemas".
toString(),
                    "room.incremental":"true",
                    "room.expandProjection":"true"]
            }
        }

        androidResources {
            noCompress "tflite"
        }

        buildFeatures {
            dataBinding true
        }

        // Required if minSdkVersion is set to lower than 26
        compileOptions {
            sourceCompatibility JavaVersion.VERSION_1_8
            targetCompatibility JavaVersion.VERSION_1_8
        }
    }
}

```

1. Ensure your `build.gradle (:app)` file includes at a minimum the following dependencies (replace "X.X.X" with the Lens SDK version you're currently using) as well as the latest **AndroidX** libraries:

```
dependencies {
    implementation 'androidx.appcompat:appcompat:X.X.X'
    implementation 'com.verifyfi.lens:verifyfi-lens-long-receipts-sdk:X.X.X.X'
    implementation "org.jetbrains.kotlin:kotlin-stdlib:$kotlin_version"
}
```

#### 1. Ensure your main

build.gradle file includes the required repositories and set Android Gradle Plugin to version 7:

```
allprojects {
    repositories {
        google()
        mavenCentral()
    }
}

buildscript {
    dependencies {
        classpath 'com.android.tools.build:gradle:7.0.2'
    }
}
```

#### 1. Remove/disable

repositoriesMode in your settings.gradle file :

```
dependencyResolutionManagement {
    // Disable the following line:
    // repositoriesMode.set(RepositoriesMode.FAIL_ON_PROJECT_REPOS)

    repositories {
        google()
        mavenCentral()
    }
}
```

#### 1. If your app uses the android:allowBackup tag, include the following settings in the Activity section of your AndroidManifest.xml file :

```
<application
    tools:ignore="AllowBackup,GoogleAppIndexingWarning"
    tools:replace="android:allowBackup">
```

## Initializing Lens SDK to your project

1. Import required classes from Lens SDK:

```
import com.verifyfi.lens.VerifyfiLens
import com.verifyfi.lens.VerifyfiLensCredentials
import com.verifyfi.lens.VerifyfiLensSettings
import com.verifyfi.lens.VerifyfiLensDelegate
```

1. Configure your

authentication credentials:

```
val verifyfiLensCredentials = VerifyfiLensCredentials()
verifyfiLensCredentials.clientId = "XXX" // replace XXX with your
assigned Client Id
verifyfiLensCredentials.username = "XXX" // replace XXX with your
assigned Username
verifyfiLensCredentials.apiKey = "XXX" // replace XXX with your
assigned API Key
verifyfiLensCredentials.url = "XXX" // replace XXX with your
assigned Endpoint URL
```

1. Mandatory settings to use Lens Credit Cards. More available settings later in this section.

```
val verifyfiLensSettings = VerifyfiLensSettings()
verifyfiLensSettings.documentTypes = arrayListOf("long_receipt",
"receipt")
verifyfiLensSettings.showDocumentTypes = true
```

1. Initialize Lens:

```
VerifyfiLens.configure(this, verifyfiLensCredentials, verifyfiLensSettings)
```

1. Launch the Lens camera

```
VerifyfiLens.showCamera()
```

Available settings:

docDetectFillUIColor: card detection rectangle fill color (default: "#9653BF8A")

docDetectStrokeUIColor: card detection rectangle stroke color (default: null)

....

## Communicating with Lens

1. Use the `VeryfiLens.setDelegate` function to define your delegate functions that will be responsible for handling events triggered by Veryfi Lens

```
VeryfiLens.setDelegate(object : VeryfiLensDelegate {  
    override fun veryfiLensClose(json: JSONObject) {  
        Log.d(TAG, json.toString(2)) // do something with the JSON  
        response here  
    }  
  
    override fun veryfiLensError(json: JSONObject) {  
        Log.d(TAG, json.toString(2)) // do something with the JSON  
        response here  
    }  
  
    override fun veryfiLensSuccess(json: JSONObject) {  
        Log.d(TAG, json.toString(2)) // do something with the JSON  
        response here  
    }  
  
    override fun veryfiLensUpdate(json: JSONObject) {  
        Log.d(TAG, json.toString(2)) // do something with the JSON  
        response here  
    }  
})
```

`veryfiLensClose` - the Veryfi Lens camera has been closed, either as a result of submitting a receipt for processing, or the user closed the camera without submitting a receipt.

```
{  
    "status": "close",  
    "queue_count": 1,  
    "session_scan_count": 1,  
    "framework-version": "1.4.0",  
    "framework-build": "1"  
}
```

`veryfiLensUpdate` - during the processing of a receipt, this delegate will be fired multiple times.

```
{  
    "status": "start",  
    "package_id": "edc8653e4c2b4ef1"  
}
```

`veryfiLensSuccess` - this delegate fires once a receipt has finished processing. This delegate provides the response from the data extraction.

```
{
  "status": "done",
  "package_id": "edc8653e4c2b4ef1",
  "data": {"abn_number": "", "account_number": "", "bill_to_address":
  "", "bill_to_name": "", "card_number": "", "category": "Meals &
  Entertainment", "created": "2021-01-14 05: 19: 51", "currency_code":
  "USD", "date": "2021-01-14 05: 19: 51", "delivery_date": "",
  "discount": 0, "due_date": "", "external_id": "", "id": 31428417,
  "img_file_name": "xxxxxxxxx.png", "img_thumbnail_url": "https: \\/\\cdn.
  veryfi.com\\/partner-receipts\\/xxxxxxxxx.t.png", "img_url": "https: \\/\\
  /cdn.veryfi.com\\/partner-receipts\\/xxxxxxxxx.png", "incoterms": "",
  "insurance": "", "invoice_number": "", "line_items": [], "notes": "",
  "ocr_text": "starbucks 23.4", "order_date": "", "payment_display_name":
  "", "payment_terms": "", "payment_type": "", "phone_number": "",
  "purchase_order_number": "", "reference_number": "VBDEC-0000",
  "service_end_date": "", "service_start_date": "", "ship_date": "",
  "shipping": 0, "subtotal": 23.4, "tags": [], "tax": 0, "tip": 0,
  "total": 23.4, "total_weight": "", "vat_number": "", "vendor":
  {"address": "", "fax_number": "", "name": "Starbucks", "phone_number":
  "", "raw_name": "Starbucks", "vendor_logo": "https: \\/\\cdn.veryfi.com\\
  /logos\\/us\\/910419611.png", "vendor_type": ""}}
}
```

`veryfiLensError` - if an error occurs during processing a submitted card, an error object is sent. If a general exception or crash is caught in `Verify Lens`, an exception object is sent instead

```
{
  "status": "error",
  "package_id": "f7d3d73ba7264742",
  "error": "Error message"
}
```

**Please note:** adding the Android Lens Receipts SDK to your app will increase your final app size by up to 8.9 MB. This is due to machine learning models, support libraries, etc included in the SDK.