

Android SDK Guide

This guide provides detailed instructions for integrating the Scanovate Colombia SDK into your Android application, enabling robust identity validation processes through facial biometric verification.

Requirements and Compatibility

Before starting the integration process, ensure your development environment meets the following requirements:

- **Android Studio:** The latest version is recommended for optimal compatibility.
- **Minimum SDK Version:** Android SDK version 21 (Lollipop) or higher.
- **Target SDK Version:** Android SDK version 35 (Android 15) to ensure your app is compatible with the latest Android OS.
- **Compile SDK Version:** Android SDK version 35.

Installation

1. Add the library

Download the "hybridComponent_3_0_0_15.aar" library and add it to your project's `libs` folder. Ensure you configure your project's `build.gradle` file to include the library as a dependency:

```
// dependencies {  
    implementation(name: 'hybridComponent_3_0_0_15', ext: 'aar')  
}
```

2. Import Required Libraries

Add the following imports in your activity or fragment where you intend to use the Scanovate SDK:

```
// Java  
  
import mabel_tech.com.scanovate_demo.ScanovateHandler;  
import mabel_tech.com.scanovate_demo.ScanovateSdk;  
import mabel_tech.com.scanovate_demo.model.CloseResponse;  
import mabel_tech.com.scanovate_demo.network.ApiHelper;
```

```
import mabel_tech.com.scanovate_demo.network.RetrofitClient;
```

The `CloseResponse` object will contain the results of the transaction, providing detailed feedback on the validation process.

Example Implementation

For a practical example of how to implement the Scanovate SDK in your Android application, refer to the following steps:

- **Setup UI Elements:** Initialize buttons, text views, and other UI elements in your activity's `onCreate` method. This setup includes buttons for starting the enrollment and verification processes, a text view for displaying results, and an edit text for user input.
- **Invoke the SDK:** Use the `HybridComponent.start` method to launch the Scanovate SDK. This method requires several parameters, including language, project name, API key, product ID, and the SDK URL. It also allows you to specify the type of capture (e.g., liveness detection, document capture) and whether to capture the front or back side of a document.
- **Handle Callbacks:** Implement `ScanovateHandler` to manage success and failure callbacks. On success, process the `CloseResponse` object to display the transaction result. On failure, handle errors accordingly.

Example

// Example capture method implementation

```
public void capture() {
    HybridComponent.start(this,
        "documentType"           //Tipo de Documento "VerificarID en
Documentación de ADO"
        "es",                     //language: "en") // en (para ingles) es (para español)
        "lulobankqa"             //ProjectName
        "db92efc69991",          //ApiKey
        "1",                     //ProductId
        "https://adocolumbia.ado-tech.com/lulobankqa/api/", //Url_Sdk
        "https://api-dev.ado-tech.com/api/EventTracer/",
        //Url_TracerBackendServices (Servicio Proporcionado Por nosotros para la Flujo de los
LOGS o TAGS del proceso.) ** Opcional**
```

```

        "true" //ImmersiveMode
        "f47ac10b-58cc-4372-a567-0e02b2c3d479", //ProcessID (ID creado
con el servicio de CreateProccees para trasa de eventos) **Opcional**
        functionCapture, 1 Livennes , 2 CardCapture
        isFrontSide, // Captura de Documento (True captura
Frontal)(False Captura Trasera)
        null, //Token
        "null",
        new ScanovateHandler() {
            @Override
            public void onSuccess(CloseResponse response, int code, String uuidDevice) {
                progress.show();
                String calificacion = response.getExtras().getStateName();
                evaluateTransaction(response.getTransactionId());
            }

            @Override
            public void onFailure(CloseResponse closeResponse) {
                String calificacion = closeResponse.getExtras().getStateName() + " "+
closeResponse.getExtras().getAdditionalProp1() ;
            }

        });
    }
}

```

Parameters Explained

- **language:** Sets the language for the SDK's UI.
- **projectName:** Unique identifier for your project.
- **apiKey:** Authentication key provided by Scanovate.
- **productId:** Identifies the specific Scanovate product/service being used.
- **sdkUrl:** The base URL for making API calls to the Scanovate services.
- **Url_TracerBackendServices:** Url for the event reporting service is not required and is only an extra service. **(Optional)**
- **ImmersiveMode:** Mode to make the component consume all available space while hiding the system UI.
- **Process_ID:** Process identifier to perform the events mapped at the SDK level. **(Optional)**

- **functionCapture**: Specifies the operation mode of the SDK.
- **documentSide**: Determines which side of the document to capture.
- **additionalParameters**: Allows for passing any additional required parameters.
- **completionHandler**: Closure that handles the response or error from the SDK.

Process Transaction Results

After capturing the necessary data, use the `RetrofitClient` to send the data for validation and display the final state of the transaction to the user.

State Codes Reference

Be aware of the following state codes when processing responses:

- `200`: "SUCCESS"
- `201`:
"THE_NUMBER_OF_CONFIGURED_ATTEMPTS_WAS_EXCEEDED_AND_NO_LIFE_WAS_FOUND_IN_THESE"
- `203`: "TIMEOUT"
- `302`: "INTERNAL_ERROR"
- `204`: "CANCELED_PROCEED"
- `205`: "PERMISSIONS_DENIED"
- `401`: "TOKEN_ERROR"
- `404`: "INVALID_CREDENTIALS"
- `500`: "CONNECTION_ERROR"

This guide aims to streamline the integration process of the Scanovate Colombia SDK into your Android application, ensuring you can efficiently implement a robust identity validation system.

Demo Application

For a comprehensive example, including full source code demonstrating the integration and usage of the Scanovate Colombia SDK, visit our [GitHub repository](#):



ADO Technologies
Colombia SAS | Powered by Scanovate

RESULTADOS



Android Version: 14

Android ID: b16ee3f258175989

Connection Type: WiFi

Time: 2024-04-11 13:24:26

Status Code: 200

Message: SUCCESS

Key Process Liveness:
20244099d75b43bdSZBIfhSMfZks7HPY

[Scanovate Colombia SDK Demo App For Android](#)

This demo app provides a hands-on example to help you understand how to integrate and utilize the SDK in your own applications.

Revision #14

Created 22 March 2024 21:50:02 by Admin

Updated 28 April 2025 22:43:20 by roger de avila