

POSTIDENT SDK Documentation for Business Customers

POSTIDENT SDK

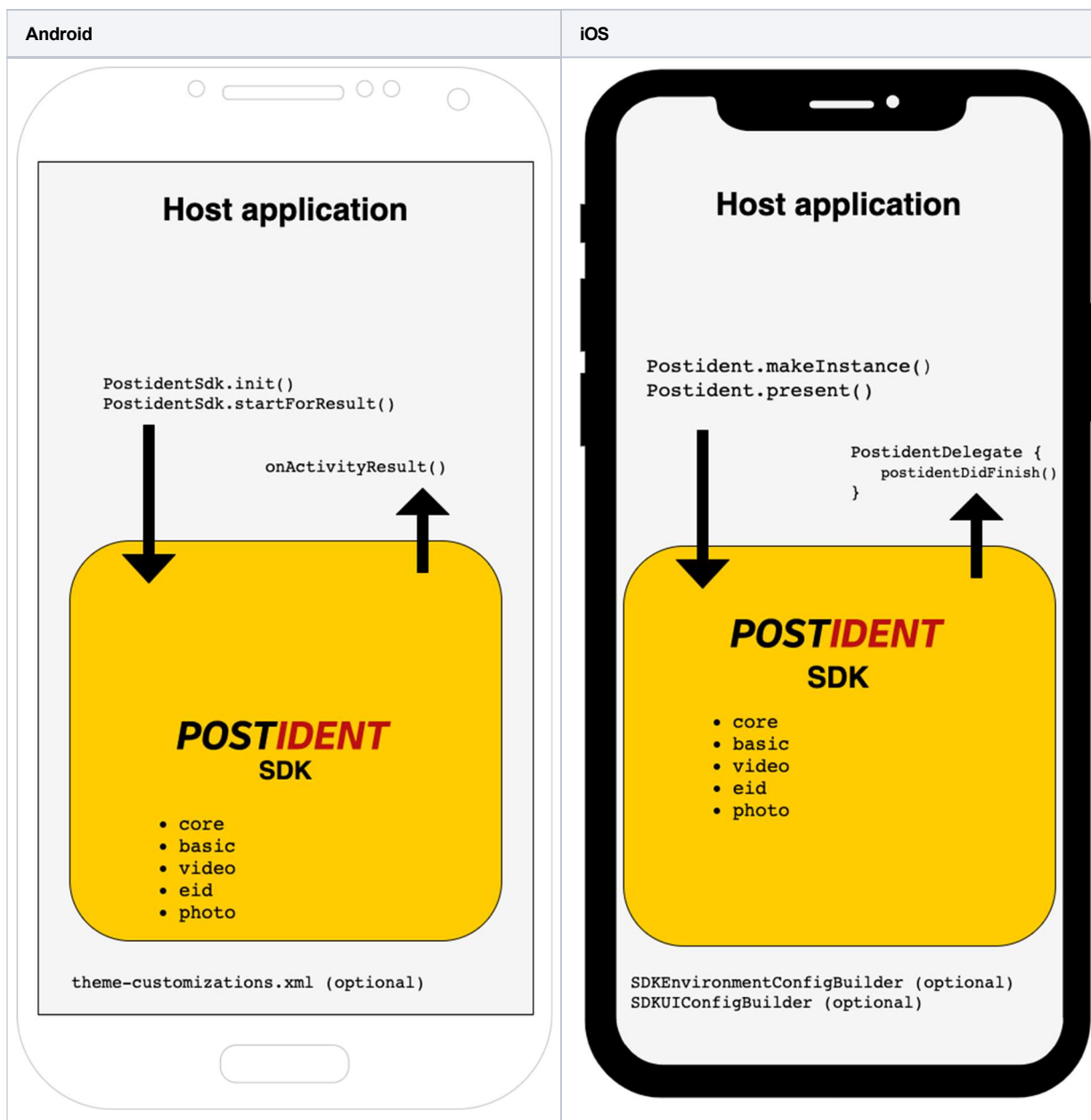
Integration of the SDK

The POSTIDENT SDK offers you the ability to run through the POSTIDENT identification process in your own Android or iOS application (called 'host application' in this document).

The whole identification process is managed by the SDK. In case you want the look and feel of your own app, you can customize some UI elements (see below).

The SDK is added as a dependency and can be started by supplying a CaseID. When finished you will get a result code.

The following pictures show a small abstract of launching the SDK and getting the result codes depending on the mobile platform. Additionally they show the customization possibilities.



Requirements

There are several preconditions needed to use the POSTIDENT SDK.

- There must be a CaseID (unique process ID) created before starting the SDK. The CaseID has to be requested via API or URL from the POSTIDENT service portal through e.g. the backend of the host application. It is 12 characters long and consists of uppercase letters and digits. If you are already using the POSTIDENT services, CaseID creation should be set up. If not, get in touch with your contact person.
- In addition, a Mobile SDK API Key is mandatory. The Mobile SDK API Key is a unique string that identifies your application against our backend servers unambiguously. You should have received a mandatory Mobile SDK API Key. If not, please ask your contact person for details.
- There must be an internet connection at all times during the identification process
- The POSTIDENT SDK only works in portrait mode, even on iPad devices
- The following permissions will be needed for the different identification methods:
 - **Basic:** Location permission for the branch finder
 - **Video:** Camera, microphone and photo library permission to collect the data
 - **EID:** NFC reader permission to get access to the NFC reader
 - **Photo:** Camera and photo library permission to collect the data
 - **AutoID:** Camera and photo library permission to collect the data
- **Android requirements:**
 - Minimum supported Android SDK version: 21 (Android 5.0)
 - Supported processor architectures: armeabi-v7a & arm64-v8a
 - Minimum supported Java version: 1.8 (see sample app `build.gradle`)
 - Device must not be rooted (will be checked upon initialization of the SDK)
 - To make the branch finder available for the user it is mandatory to add a *Google Maps API key* to your `AndroidManifest.xml`-file
- **iOS requirements:**
 - Minimum supported iOS version: 13.0, but can be built with iOS 12
 - EID module is only supported on iOS 13.2 and above
 - CocoaPods installed
 - If you include the Postident SDK you may have to provide the following datatypes and uses in your App Privacy Details for your app:
 - **Contact Info:** App Functionality: We collect name, address, phone and email of the user to identify him/her. (Photo, EID, Videoident, AutoID)
 - **User Content:** App Functionality: We collect texts, pictures, sounds and videos of the user to identify him/her and communicate via videochat and textchat (Photo, Videoident, AutoID)
 - **Identifiers:** App Functionality: We collect the ID of the user to identify him/her (Photo, EID, Videoident, AutoID)
 - **Usage Data:** Analytics: We collect usage data to improve our service
- Currently the SDK is able to display the process in the following **languages:**
 - German (DE)
 - English (EN)

The *POSTIDENT SDK* will set the language according to the currently selected device language and default to *German (DE)* in case the language is not supported.

Integration of modules

Android

```

repositories {
  maven {
    url "$rootDir/postident_maven_repository"
  }
}

dependencies {
  def sdkVersion = "CURRENT_SDK_VERSION" // insert current version
  def postident = "de.deutschepost.postidentsdk"
  implementation "$postident:core:$sdkVersion"
  implementation "$postident:basic:$sdkVersion"
  implementation "$postident:video:$sdkVersion"
  implementation "$postident:eid:$sdkVersion"
  implementation "$postident:photo:$sdkVersion"
}

```

iOS

```

target 'PostidentHost' do
  use_frameworks!
  pod 'PostidentSDK/Basic', :path => '../PostidentSDK'
  pod 'PostidentSDK/Video', :path => '../PostidentSDK'
  pod 'PostidentSDK/EID', :path => '../PostidentSDK'
  pod 'PostidentSDK/Photo', :path => '../PostidentSDK'
end

```

Start the SDK

Android

Kotlin

```
class MainKotlinActivity : AppCompatActivity() {
    override fun onCreate(savedInstanceState: Bundle?) {
        super.onCreate(savedInstanceState)
        //...
        PostidentSdk.init(
            context = this,
            mobileSdkApiKey = "SUPPLIED_MOBILE_SDK_API_KEY"
        )
        PostidentSdk.startForResult(this@MainKotlinActivity,
        POSTIDENT_SDK_REQUEST_CODE, "CUSTOMCASEID")
    }
}
```

Java

```
public class MainJavaActivity extends AppCompatActivity {
    @Override
    protected void onCreate(@Nullable Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        //...
        new PostidentSdk.Builder(MainJavaActivity.this)
            .mobileSdkApiKey("SUPPLIED_MOBILE_SDK_API_KEY")
            .init();

        PostidentSdk.INSTANCE.startForResult(MainJavaActivity.this,
        POSTIDENT_SDK_REQUEST_CODE, "CUSTOMCASEID");
    }
}
```

iOS

```
import PostidentSDK

class ViewController: UIViewController {
    //..
    @IBAction func didTapPresentPostident(_ sender: Any) {
        let postident = Postident.makeInstance(caseId: "CUSTOMCASEID",
        mobileSdkApiKey: "MOBILE_SDK_API_KEY")
        postident.present(on: self, delegate: self)
    }
}
```

Module overview

The POSTIDENT SDK includes different identification methods. Which methods can be used is determined by our backend as well as the module configuration for each platform.

Basic Ident

The basic ident method generates the data matrix code to be scanned by an employee at a Deutsche Post branch office. This method contains a branch finder with a map to find branch offices.

If the user wants to find the nearest branch office, the location permission and a Google Maps API key on Android are required.

Video Ident

The video ident method lets the user initiate a video chat with a call center agent. To chat with an agent the permission to access the camera / microphone is needed (see section Permissions below).

In the first step the user has to take photos of his ID card. The images will be sent to a backend server to process the data through image processing and patch the data from the ID card to the user record. In specific cases (if permitted by regulations) these images will be shown to the agent later to speed up the process.

If there is mandatory data missing after the recording of the images, the User Self Assessment will be shown and prompt the user to input the data.

The user will then be redirected to the Waiting Room. The waiting room lets the user choose the language of an available agent and start the process.

During the video chat with the agent, the user has to show his id card and verify his personal data.

After a successful identification the user can rate the process and will be redirected to the host application.

Signing Ident

Videoidents which require documents to be digitally signed are currently only partially supported within the SDK. It is possible to complete the video chat with an agent, but before signing the generated documents, the SDK will close with a success. It is expected of the host app to at this point inform the user about the continuation of the process and refer the user to the Postident app or website. If the host app initializes the SDK with a signing case in which the videoident portion is already complete, the SDK will also close with a success and it is also necessary to inform the user about the continuation of the process. While it is partially functional it is currently not advised to handle signing cases via the SDK, because the user will be sent to the Postident app or website at a later point anyway. Full support for signing may be added to the SDK in the future.

eID

The eID ident method lets the user identify himself by scanning a supported eID document via NFC. No human interaction is necessary and the process can be completed 24/7. There is also the ability to replace the initially set *Transport PIN* and set the *Personal PIN*. The user data is transferred between the ID card and the eID server and cannot be read / stored by the device itself.

Photo Ident

The photo ident method offers the possibility to perform an identification via photos of the identification document/s. Additionally, a short video clip of the users face is necessary as proof of life. After taking the photos and video the documents are sent to a call center agent who will review them asynchronously. Upon successful review the user is notified of the result.

AutoID

The Auto Ident method is our newest method designed to allow the user to identify himself by sending images of himself and his ID card to a machine learning service. Right now these images are still being reviewed by human agents, but in the future the service is going to identify the user autonomously and with little human validation.

Obtaining SDK version

In case you need to obtain the **SDK version name / version code** (for logging purposes) you can access those via:

- Kotlin: `PostidentSdk.getVersionName() / PostidentSdk.getVersionCode()`
- Java: `PostidentSdk.INSTANCE.getVersionName() / PostidentSdk.INSTANCE.getVersionCode()`
- Swift: `Postident.getVersion()`

Result Codes

The SDK finishes with the following result codes which should be handled by the host application:

The user was already notified by the SDK and in some special cases (see comments in table below) the user was given the choice to install the POSTIDENT app.

Note: Regardless of the result code provided by the POSTIDENT SDK it is possible that the provided result code does not match the actual case status. Best practice would be to check the case status after receiving the result code.

Result code	Meaning
RESULT_OK	Success.
RESULT_CANCELLED	User cancelled the process.
RESULT_METHOD_NOT_AVAILABLE	The selected method is not available. User was given the choice to install POSTIDENT app.
RESULT_TECHNICAL_ERROR	Technical error (not specified) occurred.

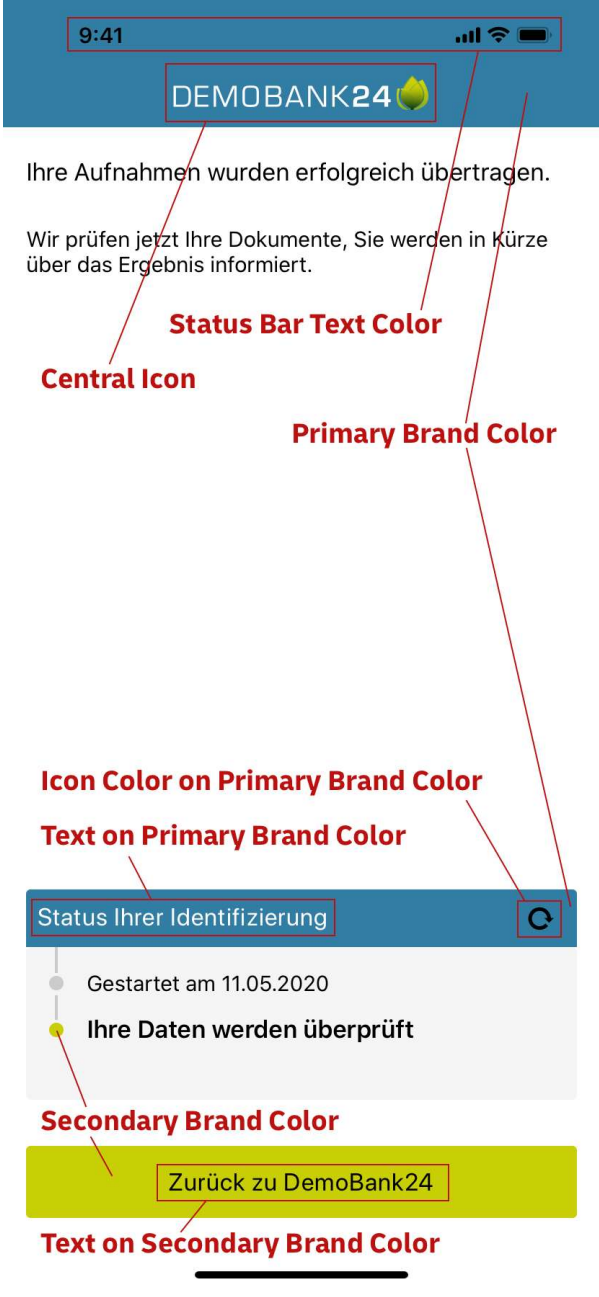
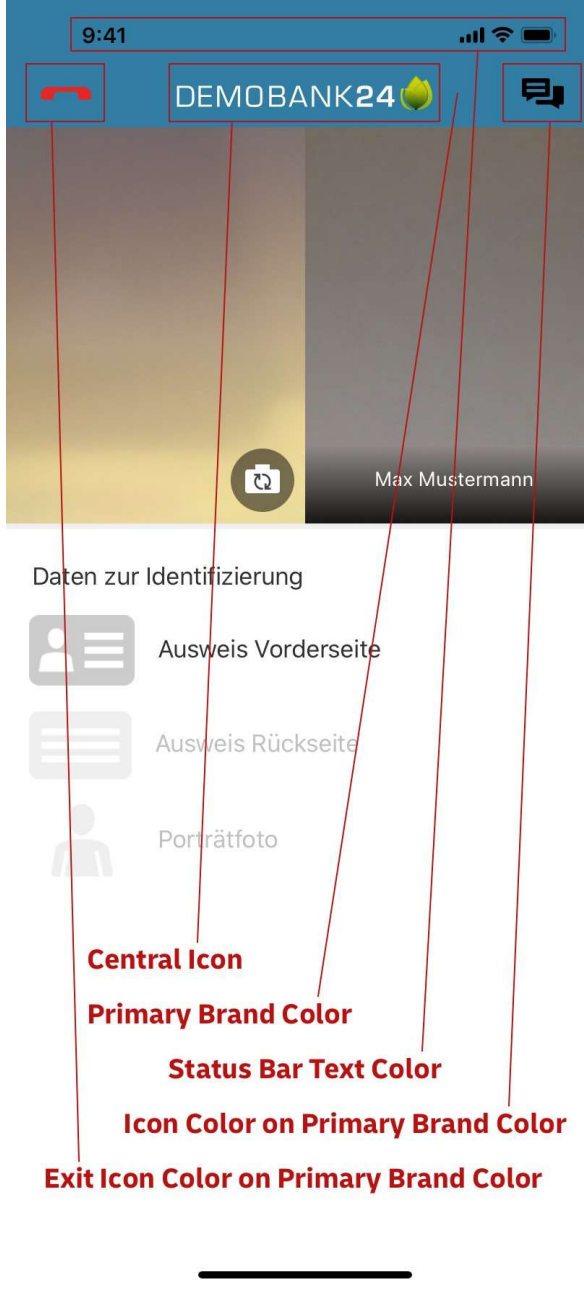
ERROR_SERVER_CONNECTION	The server returned an error (timeout or bad request).
ERROR_SSL_PINNING	A secure connection to the server could not be established (check certificate hashes). User was given the choice to install POSTIDENT app.
ERROR_WRONG_MOBILE_SDK_API_KEY	The supplied 'mobile SDK API key' is invalid. User was given the choice to install POSTIDENT app.
ERROR_SDK_UPDATE	SDK version is no longer supported. User was given the choice to install POSTIDENT app.
ERROR_OS_VERSION	Android version of the device is not supported.
ERROR_OFFLINE	The device has no active internet connection.
ERROR_CASE_DONE	The case is already closed.
ERROR_CASE_NOT_FOUND	The caseID wasn't found in the system.
ERROR_CASE_INVALID	The caseID is invalid (correct format: [A-Z0-9]{12}).

Color customization (whitelabeling)

You can specify the colors of the UI element of the POSTIDENT SDK through the color configuration.

In Android you have to specify a custom theme xml and in iOS the SDKUIConfigBuilder allows you to set the colors.

These are the customization possibilities:

Example for photoident status screen	Example for videochat screen
 <p>Example for photoident status screen</p> <p>9:41</p> <p>DEMOBANK24</p> <p>Ihre Aufnahmen wurden erfolgreich übertragen.</p> <p>Wir prüfen jetzt Ihre Dokumente, Sie werden in Kürze über das Ergebnis informiert.</p> <p>Status Bar Text Color</p> <p>Central Icon</p> <p>Primary Brand Color</p> <p>Icon Color on Primary Brand Color</p> <p>Text on Primary Brand Color</p> <p>Status Ihrer Identifizierung</p> <p>Gestartet am 11.05.2020</p> <p>Ihre Daten werden überprüft</p> <p>Secondary Brand Color</p> <p>Zurück zu DemoBank24</p> <p>Text on Secondary Brand Color</p>	 <p>Example for videochat screen</p> <p>9:41</p> <p>DEMOBANK24</p> <p>Max Mustermann</p> <p>Daten zur Identifizierung</p> <p>Ausweis Vorderseite</p> <p>Ausweis Rückseite</p> <p>Porträtfoto</p> <p>Central Icon</p> <p>Primary Brand Color</p> <p>Status Bar Text Color</p> <p>Icon Color on Primary Brand Color</p> <p>Exit Icon Color on Primary Brand Color</p>