



# Aadhaar Hackathon Developer Guide

[Overview](#)

[Contents](#)

[SDK](#)

[Android SDK](#)

[Non-Android SDK](#)

[Client API Documentation](#)

[Gateway API Documentation](#)

[Gateway URL](#)

[Samples](#)

[Android Application \(AadhaarAuthenticatorAndroidSample\)](#)

[Standalone Application \(AadhaarPacketCreatorStandaloneSample\)](#)

[FaqS](#)

## Overview

The documentation details out the content of the kit provided for hackathon participants along with faqs to help and accelerate the app development during the hackathon event.

## Contents

The hackathon kit contains the following artifacts

- Aadhaar Bridge Client API Documentation
- Aadhaar Bridge Gateway Documentation
- Android SDK App
- Standalone SDK for Auth/KYC Packet Creation (for non-android platforms)
- Gateway URL
- Samples

## SDK

The kit provides two SDK for the purpose of integrating with an Android application as well as other non-android platform. The SDK library is primarily available only on Java platform.

### Android SDK

Android App SDK is a android application, which needs to be installed on the Android Phone with OTG support. The app supports the **Morpho MSO 1300 Fingerprint Device**, which is used as part of this hackathon event. The integration with the application is through Android INTENTS. The documentation is available in the [Client API Documentation Section](#).

Look at the sample android application for usage.

### Non-Android SDK

For non-android application a Java library has been provided, which can be included as a maven or as a lib dependency. To install as a maven artifact please use the script provided as part of the folder(for sample application it is included as a local repo).

Please note that the library include the following components

- Client API POJOs
- Aadhaar protobuf schema (google protobuf)
- UIDAI Public certificates

Look at the sample for usage and code documentation.

### Client API Documentation

The client SDK documentation can found on the following link with code snippets.

<http://bridge.aadhaarconnect.com/clientsdk.html>

### Gateway API Documentation

The gateway SDK documentation can found on the following link with code snippets.

<http://bridge.aadhaarconnect.com/gatewaysdk.html>

## Gateway URL

All requests are required to be submitted to the following URLs with Content Type as "application/json"

API	Endpoint URL
Auth	<a href="https://ac.khosla labs.com/hackgate/hackathon/auth">https://ac.khosla labs.com/hackgate/hackathon/auth</a>
OTP	<a href="https://ac.khosla labs.com/hackgate/hackathon/otp">https://ac.khosla labs.com/hackgate/hackathon/otp</a>
Kyc	<a href="https://ac.khosla labs.com/hackgate/hackathon/kyc">https://ac.khosla labs.com/hackgate/hackathon/kyc</a>

## Samples

The hackathon kit provides following sample applications

### Android Application (AadhaarAuthenticatorAndroidSample)

This application accepts the aadhaar number from the user and invokes the Aadhaar Bridge SDK app to capture the biometrics, which in-turn returns the Aadhaar compliant packet as a JSON structure, which is then submitted to the hackathon gateway Auth API for further processing.

### Standalone Application (AadhaarPacketCreatorStandaloneSample)

For non-android applications (desktops, webapp), the kit provides a Java library that creates a Aadhaar compliant packet, which then can be submitted to the hackathon gateway for Auth/KYC. The code includes a sample methods for building Authentication and KYC Packets. The sample is a maven project, hence use Eclipse and import it (existing maven project).

## Faqs

***Q. What are the steps to capture the fingerprint once the SDK is integrated with the Android app?***

A. After invoking the INTENT on the Aadhaar Bridge Android Application for capturing the fingerprint, the app will show the screen with 6 fingers.

- Select the finger position, which the user wants to scan by tapping on the circle.
- Completion of the previous step(depending on the number of fingers parameter), user places the indicated finger on the scanner for biometric capture.
- Successful capture turns the circle to green and all captures will make the PROCEED button enabled.
- Tap on the PROCEED button for complete the process (this step will return the AuthCaptureData JSON data).

NOTE: Make sure that the fingerprint sensor is connected to the phone.

***Q. Is there any special consideration while processing JSON Request/Response?***

A. Yes. Any request/response containing byte[] data requires a special care. For this reason a utility class ***com.khoslaLabs.util.GsonSerializerUtil*** is available as part of the sample applications. ***GsonSerializerUtil*** for android and standalone applications are different as they use different library for base64 encoding/decoding. Hence pick the right class based on application platform.

***Q. How to check whether my Android phone supports OTG or not.***

A. Connect the fingerprint sensor to the phone using the OTG cable. The light on the sensor should glow. If it does not glow then sensor will not work. Additionally you can check the phone specification for OTG support.