



This PDF is generated from authoritative online content, and is provided for convenience only. This PDF cannot be used for legal purposes. For authoritative understanding of what is and is not supported, always use the online content. To copy code samples, always use the online content.

Client Samples

Android Sample

Android Sample

Updated in 8.5.112, 8.5.114

You can download the source code via a zip file: [Genesys Mobile Services Android Sample Source Code](#)

Overview

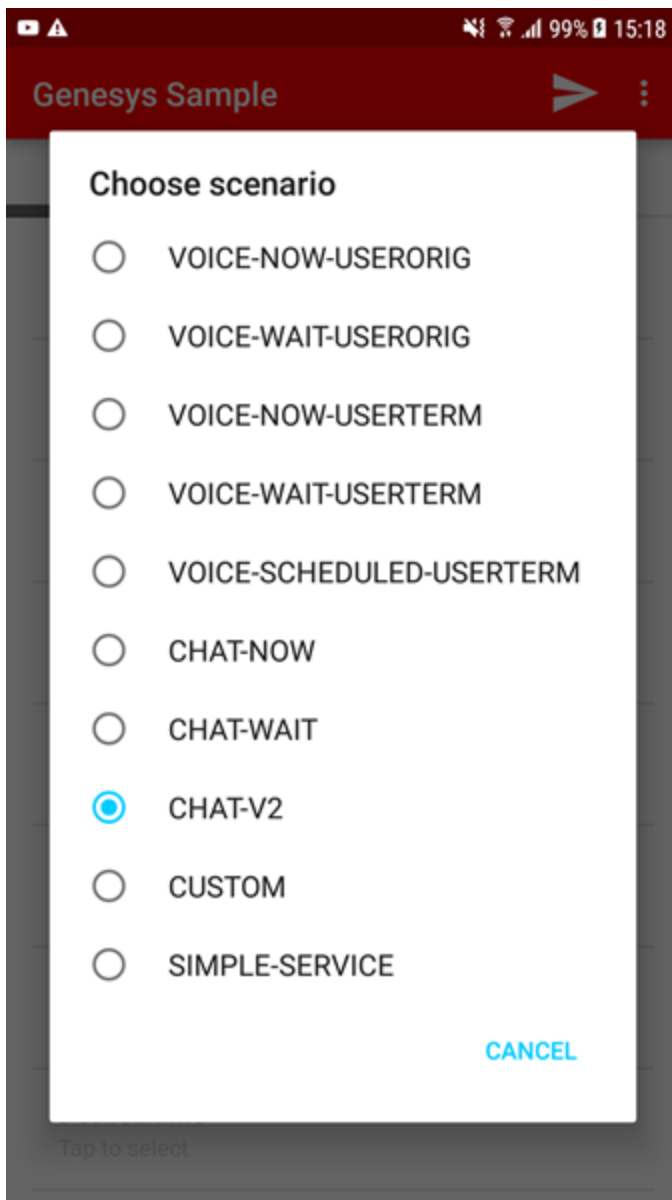
The Genesys Mobile Services (GMS) Android sample illustrates how to implement an Android app that communicates with GMS and performs supported contact scenarios. It is primarily meant to be used by developers as a reference to build an Android app with GMS. This sample app can also be used to test an existing GMS deployment because the scenario parameters are configurable on the GMS Service Management UI.

In 8.5.114, the app was extended to support:

- Chat V2 features, that are now available in the CHAT-V2 scenario:
 - File Transfer
 - Typing notifications
 - Typing preview
- Mobile Push Notifications to Android devices using Firebase Cloud Messaging (FCM)
- ApiGee integration

Implemented Scenarios

The Android sample supports the **Click to Call** and **Click to Chat** scenarios, in addition to [Callback Scenarios](#).



- Voice Callback scenarios:
 - VOICE-NOW-USERORIG
 - VOICE-WAIT-USERORIG
 - VOICE-NOW-USERTERM
 - VOICE-WAIT-USERTERM
- Chat V1 scenarios:
 - CHAT-NOW
 - CHAT-WAIT

- Chat V2 implementing CometD:
CHAT-V2
- SIMPLE-SERVICE—Enables you to test any of your callback services.
- CUSTOM—Enables you to test a simple service (not only the Callback service). For instance, you can test a stat service. The sample displays the result in a dialog box.

These scenarios are server-driven, which means that the server instructs the client with the actions needed to carry out the scenario. The client just needs to perform those actions and follow up the dialog with the server. Therefore, the client is flexible enough to support any scenario that is built using the same kind of actions. The following actions are supported:

- DialNumber - The app makes a phone call.
- ConfirmationDialog - The app displays a message.
- DisplayMenu - The app displays a menu for the user to select an item that may affect how the scenario proceeds.
- StartChat - The app starts a chat conversation. Asynchronous HTTP notifications (comet messages) are used for receiving chat server events.

Push notifications through Firebase Cloud Messaging are supported. Delayed scenarios are supported by using push notifications only; the app will not poll the server to be notified about agent availability.

Prerequisites

In order to use the Android sample app, you need to have **GMS version 8.5.114** and higher installed and running. Additionally:

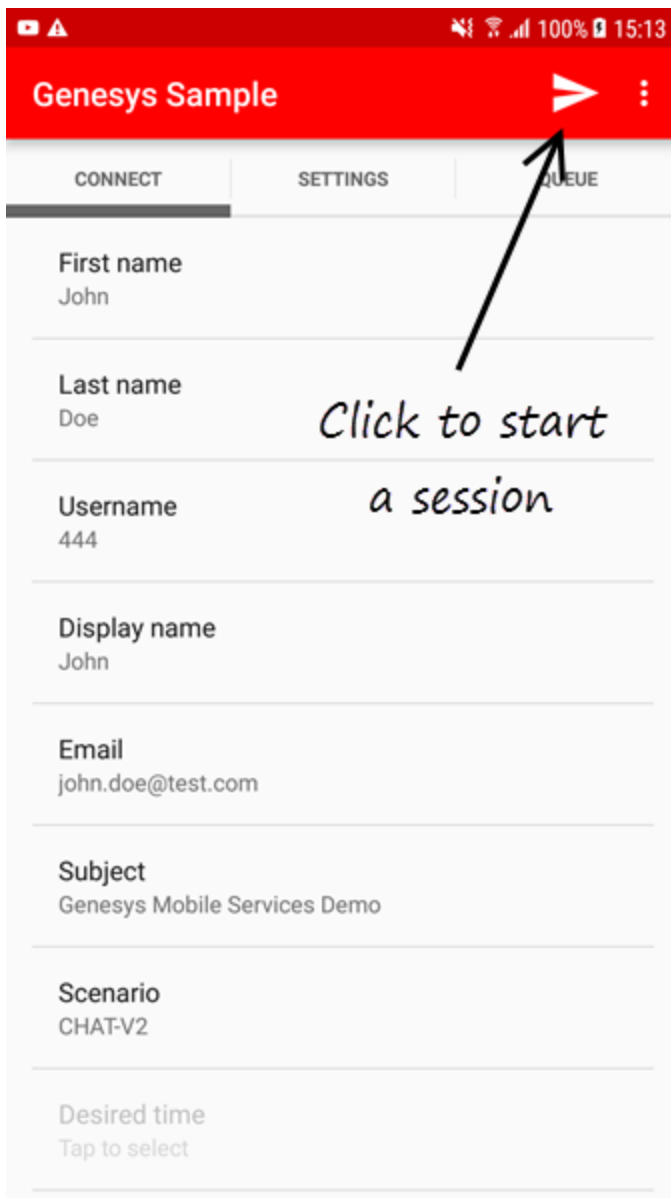
- Create your **gmstester-callback** service in the **Configured Services** panel of the Service Management UI.
 - Click create and select callback in the **Service Template** list.
 - Enter gmstester-callback for the **Service Name** and select sample in **Common Default Configuration**.
- Deploy and configure the services that you want to use in the **Configured Services** panel of the Service Management UI or in the **GMS Configuration** for the Digital Channel APIs. For example, create the chat.customer-support service for **CHAT-V2** in your **GMS configuration**.
 - See **Adding a Callback Service** for further details about callback services.
 - See **ORS-Chat Services** for Chat V1 services.
 - See **Configuring the Digital Channels API** for details about Chat V2 configuration.
- Configure **Firebase Cloud Messaging**. Edit the **fcm** section of the **GMS configuration** to provide your FCM API key.
- Enable push notifications for **Firebase Cloud Messaging** in your **GMS configuration**:
pushEnabled=fcm

Important

Restart GMS after configuring Firebase Cloud Messaging in the fcm and push sections.

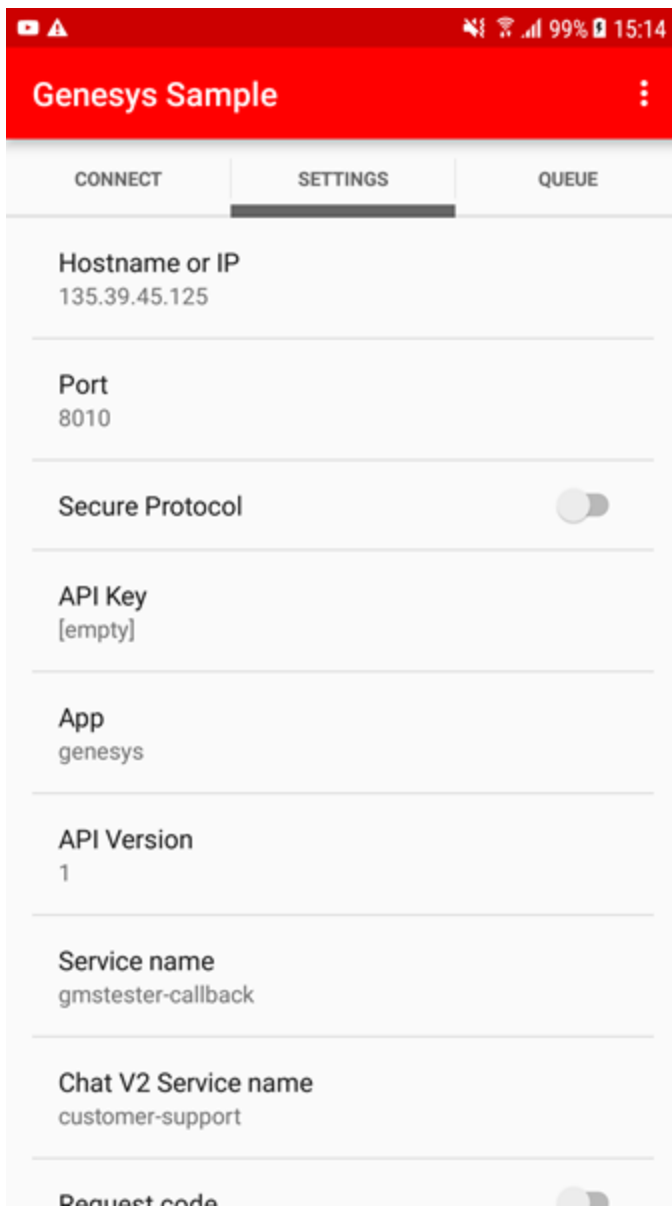
Running the Sample

In order to run the Android sample, you can just download the APK file and deploy it on your Android device. Alternatively, you can also compile the source code to make your own APK file by using the Android SDK. Using the sample is easy, as long as GMS is already deployed.

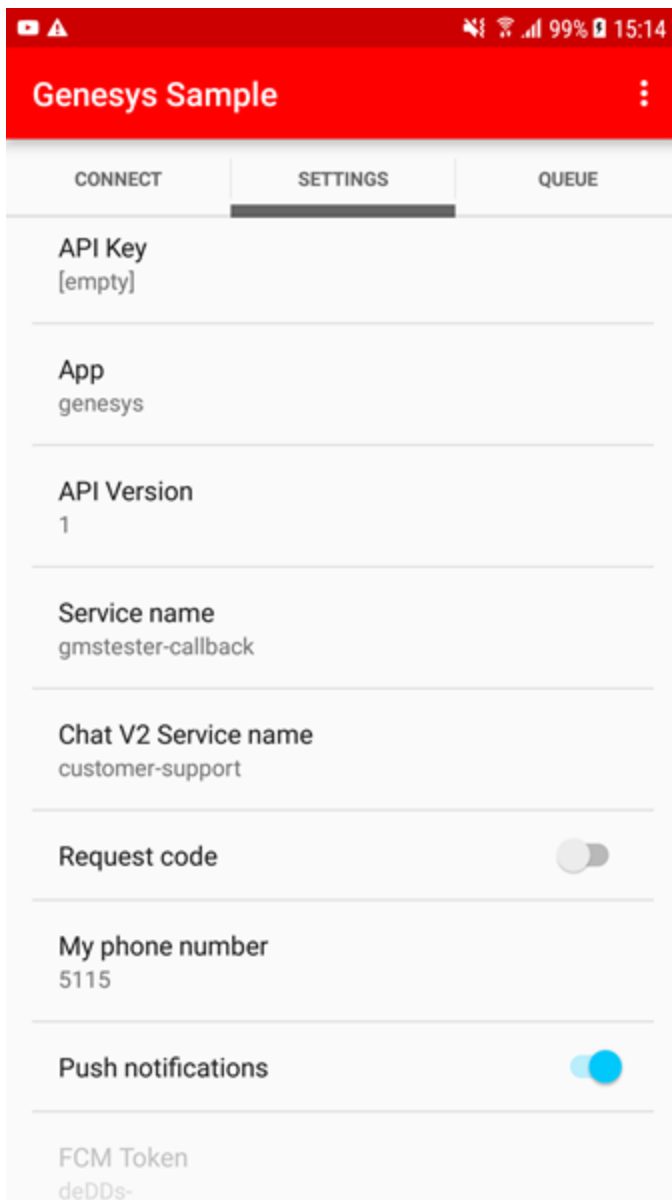


The startup screen is a configuration screen. Remember that in order for push notifications to work, you need to have registered a project in Google Cloud, and have enabled the *Firebase Cloud Messaging (FCM) for Android* API. In the app, you have to fill the FCM sender id with your Google Cloud project number. For more details, please refer to [Firebase Cloud Messaging - Getting Started](#).

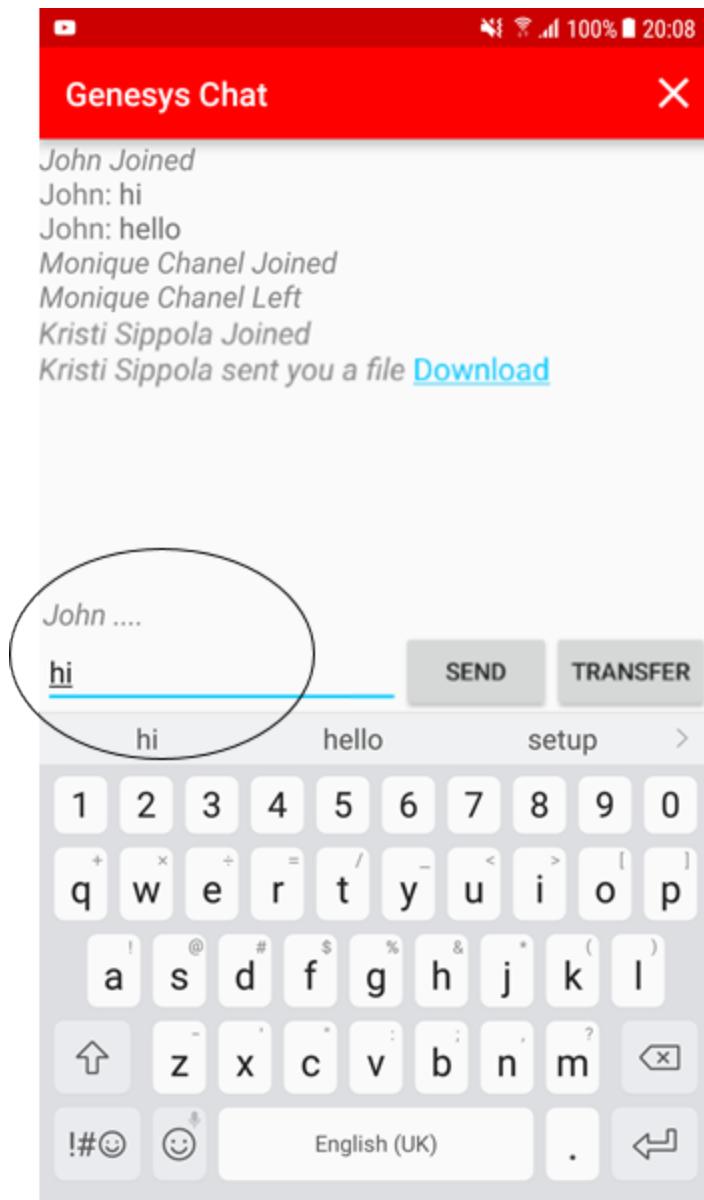
Click or tap **Scenario** to display the list of available scenarios, then select one of them, for example, CHAT-V2.



Fill the configuration options with the appropriate values for your GMS environment. You will need your sample service name, for instance, gmstester-callback, and the service name used to implement your CHAT-V2 scenario, for example, customer-service.



Scroll down and make sure that push notifications are enabled. Then, connect from the first screen.



The Chat session will start, enabling you to send and receive messages and files. You will be able to see when the other participant is typing and to leave the chat session by clicking the 'X' icon.



In order to review in detail the past communication between the client and the server, you can use the *Log* screen, which you can open from the *show log* option in the startup screen.

You can also inspect the standard Android logging, by using the Android logcat tool, for obtaining more details or displaying logs more comfortably on your computer screen. Filter by the *GenesysService* tag in order to see the client-server communication.

About the Code

The app is designed to be minimal so that developers can focus on the implementation of Mobile Services scenarios.

The app consists of three screens (Android activities):

- GenesysSampleActivity - Displays the app configuration options, allows launching the Mobile Services scenarios, and interprets the server-driven actions.
- GenesysChatActivity - Chat screen.
- LogActivity - Displays logs.

The GenesysSession class is in charge of the client-server communication via HTTP for a Mobile Services user session, including comet notifications. It relies on the [Jetty](#) HTTP Client, and on the [CometD](#) Java Client. GenesysSession can be reused "as is" in your code. (Note: this is provided as sample code, so always test your app thoroughly. See the disclaimer below.)

The GenesysService class implements an Android service responsible for creating GenesysSession objects and keeping them running in the background, even when the user switches to other apps. GenesysService distributes server responses and comet messages to the preceding activities.

The FcmMessagingService class is in charge of receiving push notifications. It is implemented following Android recommendations, using the Google Play Services library. See [Firebase Cloud Messaging - Implementing FCM Client](#).

The ChatSession class and some companion classes implement chat operations. These classes are meant to be reusable in your code. (See the disclaimer below).

Disclaimer

THIS CODE IS PROVIDED BY GENESYS TELECOMMUNICATIONS LABORATORIES, INC. ("GENESYS") "AS IS" WITHOUT ANY WARRANTY OF ANY KIND. GENESYS HEREBY DISCLAIMS ALL EXPRESS, IMPLIED, OR STATUTORY CONDITIONS, REPRESENTATIONS AND WARRANTIES WITH RESPECT TO THIS CODE (OR ANY PART THEREOF), INCLUDING, BUT NOT LIMITED TO, IMPLIED WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE OR NON-INFRINGEMENT. GENESYS AND ITS SUPPLIERS SHALL NOT BE LIABLE FOR ANY DAMAGE SUFFERED AS A RESULT OF USING THIS CODE. IN NO EVENT SHALL GENESYS AND ITS SUPPLIERS BE LIABLE FOR ANY DIRECT, INDIRECT, CONSEQUENTIAL, ECONOMIC, INCIDENTAL, OR SPECIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, ANY LOST REVENUES OR PROFITS).