Callback Administrator's Guide

Provisioning Push Notifications

4/4/2026

Contents

- 1 Callback Push Notifications
- 2 Firebase Cloud Messaging Push Notifications
  - 2.1 Before you start
- 3 Using the Push Notification page
- 4 Register your Firebase Credentials with Callback
  - 4.1 Project name
  - 4.2 Client email
  - 4.3 Private key
- 5 Test your credentials configuration
- 6 Displaying the Firebase credentials on the Push Notification page

- 
- Administrator

The **Push Notification** page in the Callback UI helps you to integrate Push Notification support into your apps. In the Callback UI, you implement Push Notifications on the **Developer** > **Credential Management** > **Push Notification** tab.

Related documentation:

-

Push Notifications are messages that are proactively sent from an app to a known, permitted client (either a mobile device or a web browser). Mobile device users are familiar with Push Notification behavior within their favorite apps.

If you have the Click-To-Call feature on your website or mobile app, the system can interact with your customers by pushing notifications about the callback to a customer's device. You can design your mobile app to offer follow-up actions such as the ability to confirm or cancel the callback. For more information about provisioning the Click-To-Call-In feature in Callback, see Provisioning the Click-to-Call-In scenario.

Only users who are members of the Callback Developer or Administrator Roles can access the **Push Notification** page in the Callback UI. For more information about controlling user access to Callback tabs and features, see Controlling User Access. As long as you have sufficient permissions to view the **Push Notification** page, then you have full access to use all of its features.

Callback Push Notifications

When Push parameters are included in a Callback Create API request on a standard callback virtual queue, the system can send Push Notifications for the following events:

- CALLBACK_UPCOMING—When the callback's Estimated Wait Time (EWT) is below the configured threshold for the queue (configured under the **Push Notification Threshold (minutes)** setting in the CALLBACK_SETTINGS data table), the CALLBACK_UPCOMING event sends a Push Notification to let the consumer know that they can expect a phone call soon. This event is sent only once. In other words, if the EWT increases above the threshold after the notification is sent and then drops below the threshold again, the CALLBACK_UPCOMING Push Notification is not sent again. The system might not send a Push Notification for the CALLBACK_UPCOMING event in the following situations:
  - When the EWT is already low (below the configured threshold) when the Callback Create API request arrives.
  - When the system determines that there is not enough time to complete the callback because the callback will be purged or the end of the business will occur before the callback is at the top of the queue.
- CALLBACK_READY—Used only with the Click-To-Call-In (Delayed) scenario. When a consumer's callback

request reaches the front of the queue, the CALLBACK_READY event sends a Push Notification to alert the consumer. If the consumer accepts the callback, the mobile app sends a Call-In Create API request. The system immediately sends another Push Notification with the information that the user needs to call in to the contact center. Note that the callback "acceptance" is a function of your mobile app's configuration. For example, your mobile app developer can configure your mobile app to make the call-in request automatically as soon as it receives the CALLBACK_READY notification. Alternatively, the app could require the consumer to make the call-in request manually by clicking a button to acknowledge receipt of the notification.

- CALLBACK_PURGED—This Push Notification is sent if the callback has been purged from the system and the callback was not rescheduled.
- CALLBACK_ATTEMPT_FAILED—This event occurs when the system makes the outbound call for a callback, but the call fails to connect to the consumer for some reason. The system makes another outbound call attempt later. In the CALLBACK_SETTINGS data table, you can configure the number of times that the system can make outbound calls for each Callback request before declaring failure. This Push Notification is not used with the Click-To-Call-In (Delayed) scenario because there are no outbound calls associated with that scenario.
- CALLBACK_FINAL_ATTEMPT_FAILED—This event occurs when the system makes the final outbound call associated with a Callback request and the call fails to connect to the consumer. In the CALLBACK_SETTINGS data table, you can configure the number of times that the system can make outbound calls for each Callback request before declaring failure. This Push Notification is not used with the Click-To-Call-In (Delayed) scenario because there are no outbound calls associated with that scenario.

Firebase Cloud Messaging Push Notifications

Genesys Callback currently supports Google Firebase Cloud Messaging (FCM) Push Notifications. Using the Callback UI, you can integrate Firebase Cloud Messaging into your web or mobile applications. For more information about FCM, see the Google Firebase Cloud Messaging documentation. You use the **Push Notification** page in the **Developer** tab to register your Firebase credentials with Callback and to test that the credentials are valid.

Before you start

Before you start working with the **Push Notification** page on the Callback **Developer** tab, ensure that you have a Google Firebase account. After you register your app with a Firebase project, you use the following Firebase data to register the app with Callback for Push Notification functionality:

- The Firebase project name that is associated with the app that you are registering with Callback.
- The client email address that is associated with the project.
- The private key.
- The application ID.
- The public API key.
- The sender ID.

> **Important**
>
> You might need to update your existing Firebase projects because of changes made to FCM in September 2019. For more information, see the entry for FCM in the Firebase JavaScript SDK Release Notes, under Version 7.0.0 - September 26, 2019.

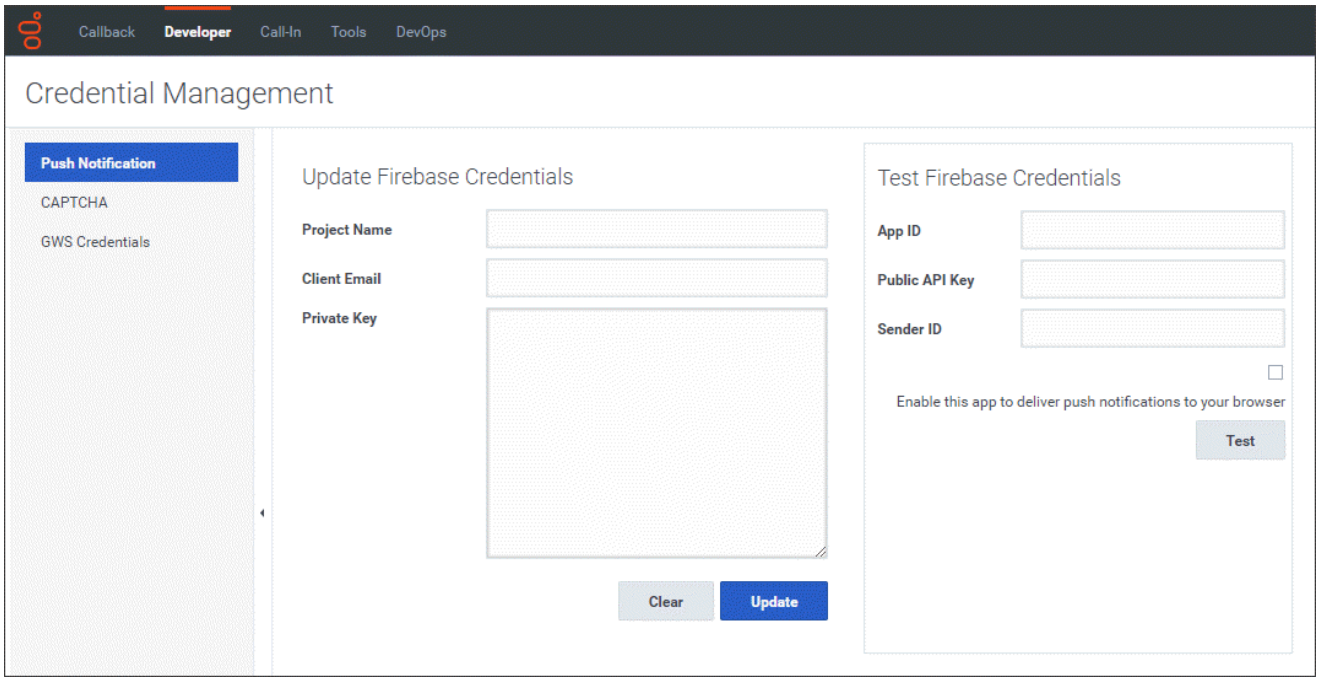
Using the Push Notification page

You can perform the following activities on the **Push Notification** page:

- Provision and update your Firebase credentials in Callback.
- Determine if your Firebase credentials for an app are already configured in Callback.
- Perform a test to verify that the Firebase credentials that you entered are correct and configured in Callback and, by extension, that the Push Notifications are working as intended.

To configure the Firebase credentials in Callback, you enter and update those credentials on the **Update Firebase Credentials** pane of the **Push Notification** page. You use the **Test Firebase Credentials** pane to verify that web Push Notifications are working in your environment. If the Firebase credentials that you provisioned on the **Update Firebase Credentials** pane are correct and valid, then – when you test the credentials – the **Push Notification** page returns a message that your Firebase credentials were successfully configured.

Register your Firebase Credentials with Callback



To implement the Firebase Cloud Messaging functionality, you must provide the following information in the **Update Firebase Credentials** pane on the **Push Notification** page:

- Project name
- Client email address
- Private key

After you click the **Update** button, a message appears in the bottom right-hand corner of the page to let you know that your credentials were successfully updated or if there was an issue.

Genesys recommends that you also test your Firebase credentials to make sure that the credentials have been successfully configured in Callback. To run the test, you require the Firebase Application ID, the Public API key, and the Sender ID associated with the project shown in the **Update Firebase Credentials** pane.

Project name

This name is a user-friendly name that you configure for the Firebase project. For example, `fcm-client-11959`. Once configured, you can find the Firebase project name (or Project ID) in the **Project settings** > **General** tab on the Firebase console. On the **Push Notification** page, you must enter the Firebase project name *exactly* as it is configured in Firebase.

Client email

The email address associated with the Firebase project. On the Firebase console, this is the service

account. In the file that contains the private key, the account is called the client email address. For example, a client email address could be `firebase-adminsdk-rkbys@fcm-client-11959.iam.gserviceaccount.com`. Once configured, you can find the service or client email account on the **Project settings** > **Service accounts** tab on the Firebase console or within the file that contains your private key.

Private key

An application uses the private key to access the Firebase API. You generate a file that contains the private key for your app on the **Project settings** > **Service accounts** tab of the Firebase console. Generating the file downloads it to your local machine. The private key is a very long character string and is only available in the downloaded file.

If, for any reason, you need to generate a new private key for an application that was previously configured in Callback, then you can do so on the Firebase console, but then you must update the credentials in Callback using the **Update Firebase Credentials** pane on the **Push Notification** page. Remember to test the new credentials as well.

Test your credentials configuration

If you are registering new Firebase credentials with Callback or updating credentials, then always test those credentials to verify that the Firebase and Callback configuration is valid. You require the Firebase App ID, Public API key, and Sender ID to test your credentials. To find that information, log into your Firebase project, open the list of apps, and click the Settings icon beside the app that you're registering with Callback. Scroll to the bottom of the page of settings until you locate the App ID, the Sender ID, and the API key.

You must check the **Enable this app to deliver push notifications to your browser** box to run the test.

The **Push Notification** page returns either a "successful configuration" message or a "failure" message.

If you receive a failure message, use the following process to recover and try again:

1. Check that the App ID, the Public API key, and the Sender ID that you entered on the **Update Firebase Credentials** pane match the values on the Firebase console for the app that you are registering with Callback. Make any necessary adjustments and then run the test again.
2. If the App ID, the Public API key, and the Sender ID match and you don't receive a Push Notification, then refresh the browser and run the test again.
3. If you don't receive a Push Notification, then navigate to the settings for your browser. Ensure that the browser allows notifications from your Genesys Engagement Service (GES)/Callback host. If you need help finding the setting, check the documentation for your browser. If your browser is blocking notifications, then alter the setting to enable notifications. Run the test again.
4. If you don't receive a Push Notification, then refresh your browser again, review your FCM credentials once more, and run the test again.

Displaying the Firebase credentials on the Push Notification page

If you refresh the **Push Notification** page or return to it some time after you successfully registered your Firebase credentials, you see only a few characters of the client email address and the private key. The obfuscation is for security purposes. Once configured, it is impossible to see the complete entries again on the **Push Notification** page. The Client email address and Private key sections on this page explain where you can find the full credentials again, if required.