Qualcomm® Toq™ Smartwatch SDK 1.4

API User Guide

Rev. D

March 11, 2014

Qualcomm Connected Experiences, Inc.
5775 Morehouse Drive
San Diego, CA 92121
U.S.A.

Revision history

Revision	Date	Description
A	November 27, 2013	Initial release
B	December 13, 2013	Corrected legal statements, reformatted to QCE standards.
C	February 7, 2014	Updated all sections for commercial SDK release.
D	March 11, 2014	Updates to notification pop up cards, added Toq Contextual menu feature, sample app updates and bug fixes.

Contents

1 Introduction

This document provides an overview of the Toq™ Smartwatch SDK. The following sections describe the API available to a third-party Android app which is resident on the same device as the Android Toq app, to allow interaction with the Qualcomm® Toq™ smartwatch.

1.1 What's New in this Version?

Version 1.4 of the Toq SDK contains the following new features:

- A notification pop-up is now defined as a card, analogous to a card in a deck of cards applet. This means that it is now possible to subscribe to notification pop-up card events, in the same way as regular card events in a deck of cards applet.
- A card in a deck of cards applet and a notification pop-up card now support a contextual menu. The contextual menu is defined as a property of the card and rendered on the Toq smartwatch as a set of menu options at the bottom of the card. Each time the user selects a menu option, it triggers an event that is propagated back to the third-party app.
- A notification pop-up's vibrating alert can now be enabled or disabled before it is sent to the Toq smartwatch.
- Updates to the sample app APK, Eclipse project and documentation to reflect the new features.
- Various bug fixes and minor enhancements.

1.2 Requirements

A third-party app using the Toq API Android library v1.4 requires a corresponding installation of the Toq app v1.4 or later. The Toq app v1.4 requires a device running Android v4.0.3 or later.

The Toq app v1.4 is backwards compatible with third-party apps that use the Toq API Android library v1.3.

2 Toq SDK Overview

The Qualcomm® Toq™ SDK allows developers to Toq enable their Android apps so they can interact and communicate with the Toq™ smartwatch.

2.1 Deck of Cards Concept

Toq smartwatch applications or applets use an approach similar to a deck of cards as a means of grouping together related screens of information. The user launches an applet by tapping the applet's launcher icon on the Toq smartwatch's home screen. The applet's initial view is a summary screen of all the cards in the applet's deck that comprises a list of individual summary cards, each containing truncated text from the corresponding full-sized card. The user can scroll through the list and upon selecting a summary card will be shown the full sized card. The user can then swipe back to the summary list to select a different card or back again to the home screen. The Notifications applet on the Toq smartwatch's home screen is an example of a deck of cards applet. Currently, two card levels below the launcher icon are supported for each applet.

2.2 Applet Creation

One of the main purposes of the SDK is to provide a means for a third-party app to create a deck of cards applet and install it on to the Toq smartwatch. The high-level process follows:

1. Include the Toq API library in the third-party Android app.
2. Use the API library to create a Java representation of a deck of cards.
3. Once the deck of cards is created, the library is used to trigger the installation of the deck of cards on to the Toq smartwatch.

Because the Java representation of the deck of cards is effectively a peer to the deck of cards applet on the Toq smartwatch, data can be updated in the Java representation and then used by the library to trigger an update of the deck of cards applet. In this way, the third-party app can regularly update the deck of cards applet on the Toq smartwatch.

The SDK also provides a means for third-party apps to send notifications or pop-ups to the Toq smartwatch. A pop-up notification on the Toq smartwatch is rendered as a vibrating pop-up notification. The Toq API library provides the means for creating these notifications and triggering them on the Toq smartwatch.

To use the Toq API library to communicate with the Toq smartwatch, the third-party app must be installed on the same device as the Toq smartphone app. The Toq smartwatch must also be paired and connected to the Toq app. This is required because direct communication between a third-party app and the Toq smartwatch is neither feasible nor desired, so the Toq app acts as an intermediary between the third-party app and the Toq smartwatch. Specifically, a third-party app communicates with the Toq app using inter-process communication via an API which is defined using the Android Interface Definition Language (AIDL). The Toq app maintains a dedicated

API service that processes incoming third-party application API calls. The Toq app in turn communicates with the Toq smartwatch using its own internal transport mechanisms.

This approach has an added advantage in that it enables the Toq app to perform a level of authentication and authorization to restrict access to the Toq smartwatch to only those third-party apps that have been granted access by the user.

3 Client API and Library

The SDK contains the following components:

- Toq API Android library JAR
- Javadocs for the Toq API
- Sample app which connects to the Toq app, installs/updates/uninstalls a deck of cards applet and sends notifications to the Toq smartwatch
- An Eclipse project containing the source for the sample Android app.
- *Qualcomm® Toq™ Smartwatch SDK API User Guide* (this document).

3.1 Toq Library and API

To communicate with the Toq app's API service, the third-party app must make use of the Toq API library included in the SDK. The library is an Android library JAR that that contains the following:

- AIDL definition files
- Classes that enable a developer to construct and maintain an applet or deck of cards
- Classes that deal with connecting to and interacting with the Toq app's API service.

3.2 Creating an Applet (Deck of Cards)

An applet or deck of cards is represented in the library by a DeckOfCards class. A DeckOfCards object contains a single ListCard object, which may hold one or more Card objects. Currently, there is only a single type of Card, a SimpleTextCard. A SimpleTextCard consists of a header, title, timestamp, info value and multi-line message.

The typical process a developer performs to create an applet follows.

1. Create a RemoteDeckOfCards object (a specialized implementation of a DeckOfCards class for use with the Toq app's API service).
2. Add newly created Cards which have been populated with useful information.
3. Install the deck of cards on to the Toq smartwatch via the DeckOfCardsManager.installDeckOfCards() method.
4. As information changes and corresponding Cards are updated, the deck of cards applet on the Toq smartwatch should be updated with further API calls to DeckOfCardsManager.updateDeckOfCards().

A deck of cards may also require access to resources such as icons, for example, to customize the deck of cards applet's launcher icon. The library provides a ResourceStore as a repository for

resources that can be shared among the deck of cards and its cards, and a specialized RemoteResourceStore implementation for use with the Toq app's API service.

3.3 Creating a Notification

A Toq notification is represented in the library by a ToqNotification class, which contains a notification Card object that defines the notification pop-up's content. Currently, there is only a single type of notification Card, a NotificationTextCard. A NotificationTextCard consists of a header, title, timestamp, info value and multi-line message.

The typical process a developer performs to create a notification follows.

1. Create a RemoteToqNotification object (a specialized implementation of a ToqNotification class for use with the Toq app's API service).
2. Send the notification to the Toq smartwatch via the DeckOfCardsManager.sendNotification() method.

NOTE: A third-party app must install a deck of cards applet before it is permitted to send a notification to the Toq smartwatch.

3.4 Remote Package

The Toq library includes a remote package that contains a number of classes required for a client to interact with the Toq app's API service. In addition to the remote specialization classes mentioned in the previous sections (RemoteDeckOfCards, RemoteResourceStore and RemoteToqNotification), this package contains a DeckOfCardsManager manager class that handles connectivity with the Toq app's API service and provides the API methods to interact with the Toq smartwatch via the Toq app.

The typical process a developer performs to connect to the Toq app's API service follows.

1. Get an instance of the DeckOfCardsManager singleton.
2. Register a DeckOfCardsManagerListener with the DeckOfCardsManager.
3. Connect to the Toq app's API service by calling DeckOfCardsManager.connect().
4. Receive a callback to DeckOfCardsManagerListener.onConnected() upon establishing the connection with the Toq app's API service.

Once connected, the developer can check state, utilize the DeckOfCardsManager methods to maintain a deck of cards applet on the Toq smartwatch and send notifications.

3.4.1 DeckOfCardsManager

The following subsections summarize the API methods in DeckOfCardsManager.

Refer to the Toq API Javadocs for more detailed information.

3.4.1.1 Connectivity with the Toq App's API Service

```
public boolean isConnected();
public synchronized void connect() throws RemoteDeckOfCardsException;
public synchronized void disconnect();
```

3.4.1.2 Toq Watch Status

```
public synchronized boolean isToqWatchConnected() throws
RemoteDeckOfCardsException;
```

3.4.1.3 Add and Remove Listeners

```
public void addDeckOfCardsManagerListener(DeckOfCardsManagerListener
listener);
public void removeDeckOfCardsManagerListener(DeckOfCardsManagerListener
listener);
public void addDeckOfCardsEventListener(DeckOfCardsEventListener listener);
public void removeDeckOfCardsEventListener(DeckOfCardsEventListener
listener);
```

3.4.1.4 Install, Update and Uninstall Deck of Cards

```
public synchronized boolean isInstalled() throws
RemoteDeckOfCardsException;
public synchronized void installDeckOfCards(RemoteDeckOfCards deckOfCards,
RemoteResourceStore resourceStore) throws RemoteDeckOfCardsException;
public synchronized void installDeckOfCards(RemoteDeckOfCards deckOfCards)
throws RemoteDeckOfCardsException;
public synchronized void updateDeckOfCards(RemoteDeckOfCards deckOfCards,
RemoteResourceStore resourceStore) throws RemoteDeckOfCardsException;
public synchronized void updateDeckOfCards(RemoteDeckOfCards deckOfCards)
throws RemoteDeckOfCardsException;
public synchronized void uninstallDeckOfCards() throws
RemoteDeckOfCardsException;
```

3.4.1.5 Send Notification

```
public synchronized void sendNotification(RemoteToqNotification
notification) throws RemoteDeckOfCardsException;
```

3.4.2 DeckOfCardsManagerListener

The DeckOfCardsManagerListener is a listener for connectivity status events between the third-party app and the Toq app's API service and applet installation events.

```
void onConnected();
void onDisconnected();
void onInstallationSuccessful();
void onInstallationDenied();
void onUninstalled();
```

3.5 Card Events

When creating a card that will be added to a DeckOfCards, or that will form the basis of a ToqNotification, there is an option to subscribe to events for that specific card. This means that, as the user interacts with the card on the Toq smartwatch, events will be sent back to a registered

DeckOfCardsEventListener in the third-party app. To enable event subscription for a card, the card must be set to receive events, by calling the method Card.setReceivingEvents(true).

Events are triggered during the card's lifecycle: when the user opens the card, when the card then becomes visible, when the card then becomes invisible and when it is closed. Events are also triggered when the user selects a menu option from the card's contextual menu, if one has been defined for the card.

A DeckOfCardsEventListener is registered with the DeckOfCardsManager, and contains the following event callbacks:

```
void onCardOpen(String cardId);
void onCardVisible(String cardId);
void onCardInvisible(String cardId);
void onCardClosed(String cardId);
void onMenuOptionSelected(String cardId, String menuOption);
```

3.6 Card Contextual Menu

A card, when used as a card in a deck of cards applet or as the basis of a notification pop-up, supports a contextual menu. The contextual menu is defined as a property of the card and rendered on the Toq smartwatch as a set of menu options at the bottom of the card.

Specifically, a card's contextual menu options are defined as an array of Strings and set as a property of the card by calling the method Card.setMenuOptions(String[] menuOptions).

Each time the user selects a menu option, it triggers an event, which is propagated back to the third-party app via the event listener's method DeckOfCardsEventListener.onMenuOptionSelected(String cardId, String menuOption).

3.7 Toq App Status Intents

The Toq app will send out a BroadcastIntent upon a change in the status of the Android device's Bluetooth driver or the pairing/connection status between the Android device and the Toq smartwatch. Table 1 lists the BroadcastIntents that a third-party app can register a BroadcastReceiver to receive.

Table 1. BroadcastIntent events

Event	BroadcastIntent Action
Android device's Bluetooth driver enabled	com.qualcomm.toq.smartwatch.bluetooth.enabled
Android device's Bluetooth driver disabled	com.qualcomm.toq.smartwatch.bluetooth.disabled
Android device and Toq smartwatch paired	com.qualcomm.toq.smartwatch.paired
Android device and Toq smartwatch unpaired	com.qualcomm.toq.smartwatch.unpaired
Android device and Toq smartwatch connected	com.qualcomm.toq.smartwatch.connected
Android device and Toq smartwatch disconnected	com.qualcomm.toq.smartwatch.disconnected

3.8 Logging

The library logging can be controlled via the Logger utility class, where logging can be disabled completely or customized to use a preferred tag.

3.9 Deck of Cards

The following subsections detail the processes supported by the applet.

3.9.1 Installation

The installation process requires the user to approve or cancel an installation request before the installation can be completed. The user approval process is managed by the Toq app and takes the form of a dialog box, which the user must interact with to approve or cancel the request.

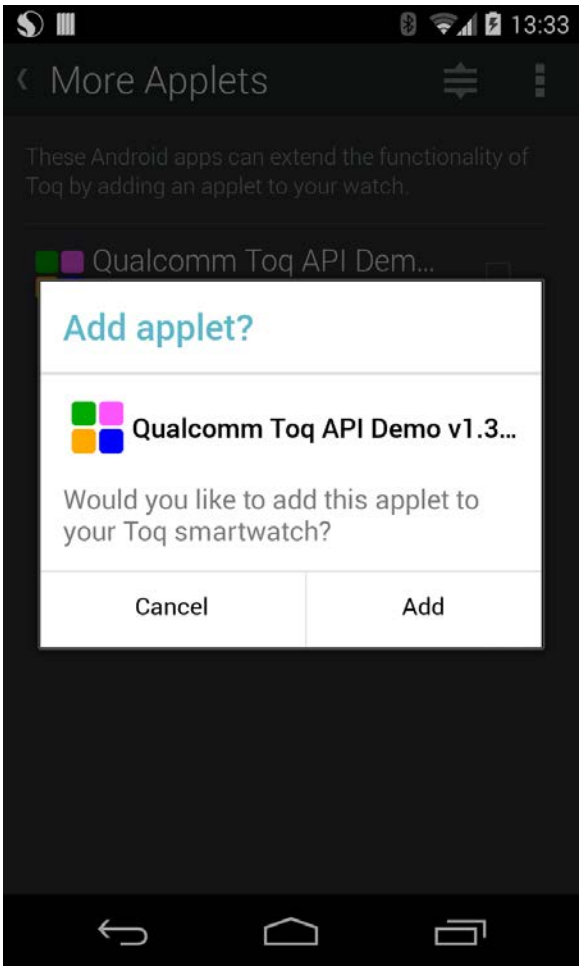


Figure 1. API sample app confirmation dialog

Once the user approves the request, and the deck of cards applet is installed, it will appear on the Toq smartwatch's home screen and also as a checked item (with the name of the third-party app from which it originated) in the list of applets in the Toq app's More Applets section under the Preferences screen.

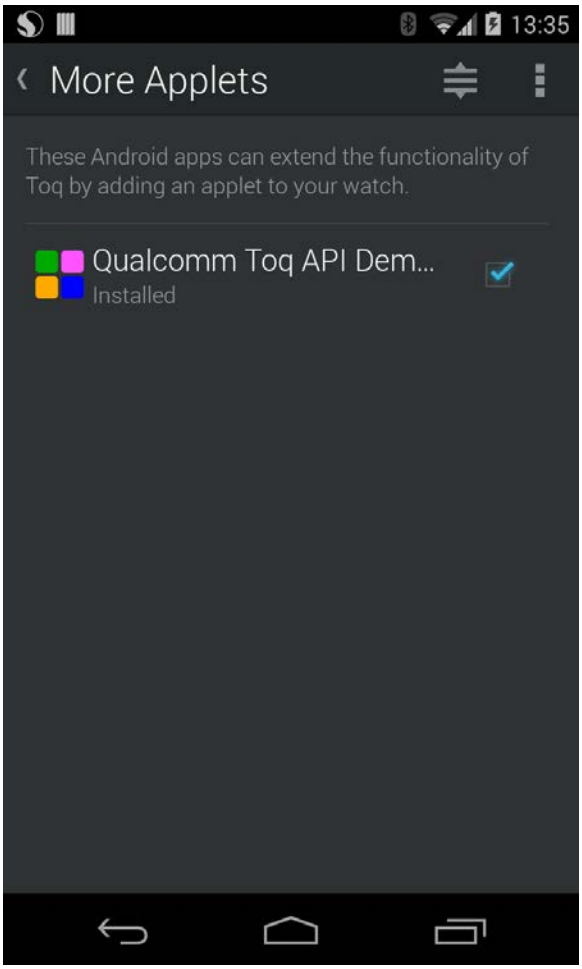


Figure 2. API sample app installed

The Toq app also performs a callback to the DeckOfCardsManagerListener.onInstallationSuccessful() method to notify the third-party app that the installation was successful.

If the user cancels the installation request, the Toq app performs a callback to the DeckOfCardsManagerListener.onInstallationDenied() method to notify the third-party app that the installation was canceled.

There are two ways to trigger the installation process, detailed in the following subsections.

3.9.1.1 Third-Party App-initiated Installation

A third-party app can test to see if the deck of cards applet is already installed on the Toq smartwatch by calling the DeckOfCardsManager.isInstalled() method. If it isn't installed, the third-party app can call the DeckOfCardsManager.installDeckOfCards() method to install the deck of cards applet.

This method of installation triggers the Toq app approval process detailed above. However, this is not the preferred way to trigger the installation (see section 3.9.1.2).

3.9.1.2 User-initiated Installation

The preferred installation method is to have the user initiate the deck of cards applet installation from the Toq app.

The user can view the list of currently uninstalled deck of cards applets in the More Applets section under the Preferences screen in the Toq app. The list of uninstalled applets is determined by discovering all of the third-party apps on the device that are Toq-enabled, meaning they each have a deck of cards applet which could be installed on to the Toq smartwatch.

For a third-party app to advertise that it has a deck of cards applet available to install, it must register a BroadcastReceiver to listen for the Toq app BroadcastIntent with the action com.qualcomm.toq.smartwatch.install.applet. The BroadcastReceiver must be registered in the third-party app's Android manifest rather than programmatically to ensure that it is registered correctly.

The installation process follows.

1. The user scrolls to the applet item in the applets list in the More Applets section under the Preferences screen in the Toq app.
2. The user taps the item's check box to select it.
3. The Toq app sends the BroadcastIntent specifically to that selected third-party app.
4. The Toq app acts upon the received intent and triggers the installation process by calling the API method DeckOfCardsManager.installDeckOfCards().

NOTE: Care should be taken when implementing the BroadcastReceiver in the third-party app to deal with edge-case scenarios where the intent may be received when the third-party app it isn't currently running, the app's UI activities aren't running or there isn't an available network connection and so on.

3.9.2 Uninstall

The following subsections detail the methods in which a deck of cards applet can be uninstalled from the Toq smartwatch.

3.9.2.1 Third-Party App-initiated Uninstall

The deck of cards applet can be manually uninstalled by the third-party app by calling the DeckOfCardsManager.uninstallDeckOfCards() method. This action silently uninstalls the deck of cards applet from the Toq smartwatch and removes the applet from the list of applets in the Toq app's More Applets section under the Preferences screen.

However, if the third-party app advertises that it has a deck of cards applet available to install (as explained in section 3.9.1.2), it remains listed but will be unchecked, indicating that it is now uninstalled.

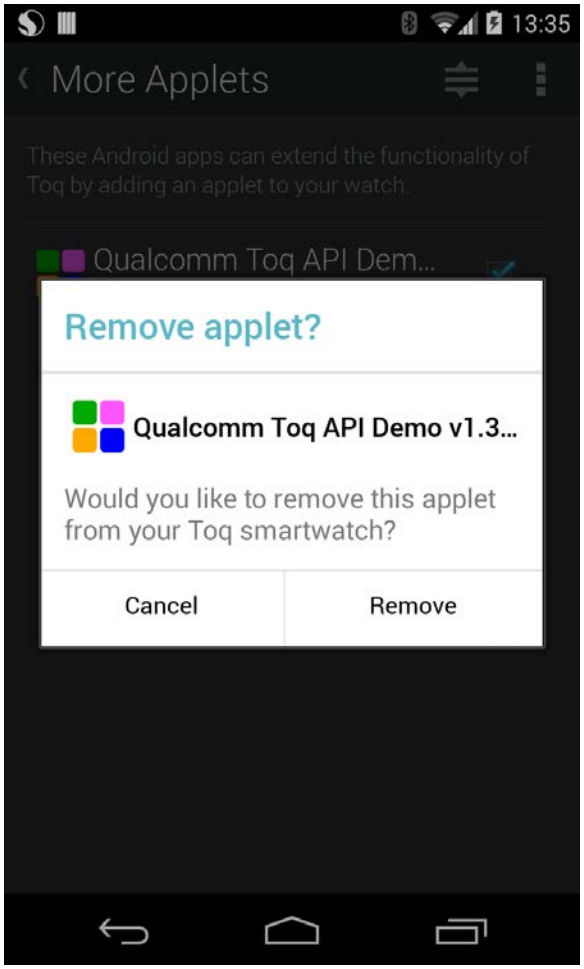
Upon completing the uninstall, the Toq app performs a callback to the DeckOfCardsManagerListener.onUninstalled() method to notify the third-party app that the deck of cards applet was uninstalled.

3.9.2.2 User-initiated Uninstall

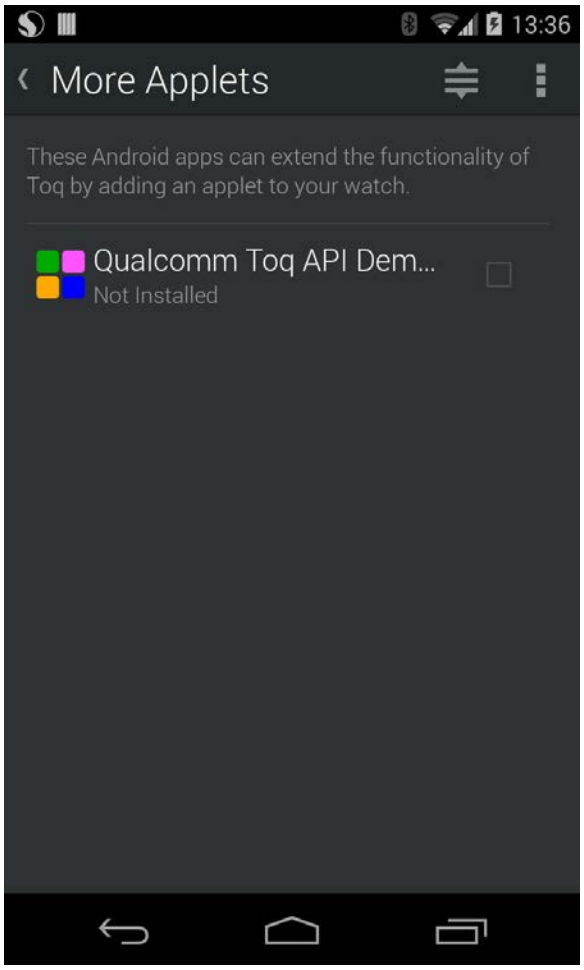
The preferred method to uninstall a deck of cards applet is to allow the user to uninstall the applet using the More Applets section under the Preferences screen in the Toq app.

The uninstallation process follows.

1. The user taps the checked item to clear the check box.
2. The user taps **Remove** in the uninstall confirmation dialog.



3. The deck of cards applet is uninstalled from the Toq smartwatch.



If the third-party app is advertising that it has a deck of cards applet available to install (as explained in section 3.9.1.2), it remains listed but will be unchecked, indicating that it is now uninstalled, otherwise it will be removed from the list.

Upon completing the uninstall, the Toq app performs a callback to the DeckOfCardsManagerListener.onUninstalled() method to notify the third-party app that the deck of cards was uninstalled.

4 Security

Security aspects of the SDK, from an architectural and API level perspective can be summarized as follows:

- At a high level, a third-party app cannot directly communicate with the Toq smartwatch. All communication from a third-party app is via the Toq app which manages all subsequent interaction with the Toq smartwatch.
- A third-party app cannot install, update, or uninstall a deck of cards or send a notification to the Toq smartwatch without first having been granted the user's explicit approval to do so, during the initial deck of cards installation process.
- At an API level, the third-party app is authenticated with the Toq app at the point that the initial deck of cards installation is approved by the user and completed. Subsequent API calls require the third-party app to present the same credentials for authentication, until the deck of cards applet is uninstalled in which case the process begins again. Most API calls require the third-party app to be authenticated in order to be used. Third-party app authentication is handled automatically within the Toq API library.

5 Sample Application

The SDK includes a sample app APK and the source for the sample app in the form of an Eclipse project that can be imported into Eclipse and run directly.

The app demonstrates installing, updating and uninstalling a deck of cards applet that consists of three cards on to the Toq smartwatch. The sample app can also send notifications to the Toq smartwatch. The app also registers a BroadcastReceiver to help illustrate how a user can trigger the install of the app's deck of cards applet from the Toq app.

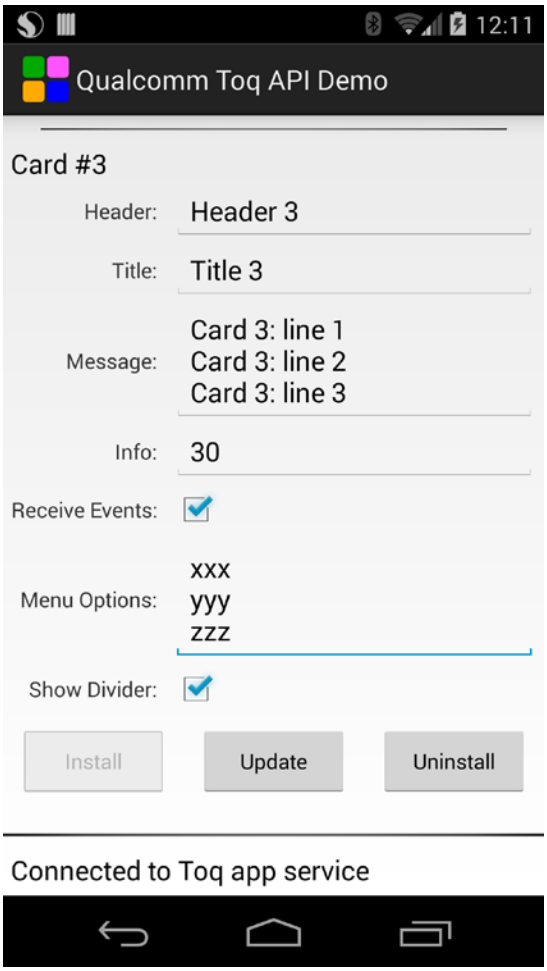


Figure 3. Sample app (Deck Of Cards section)

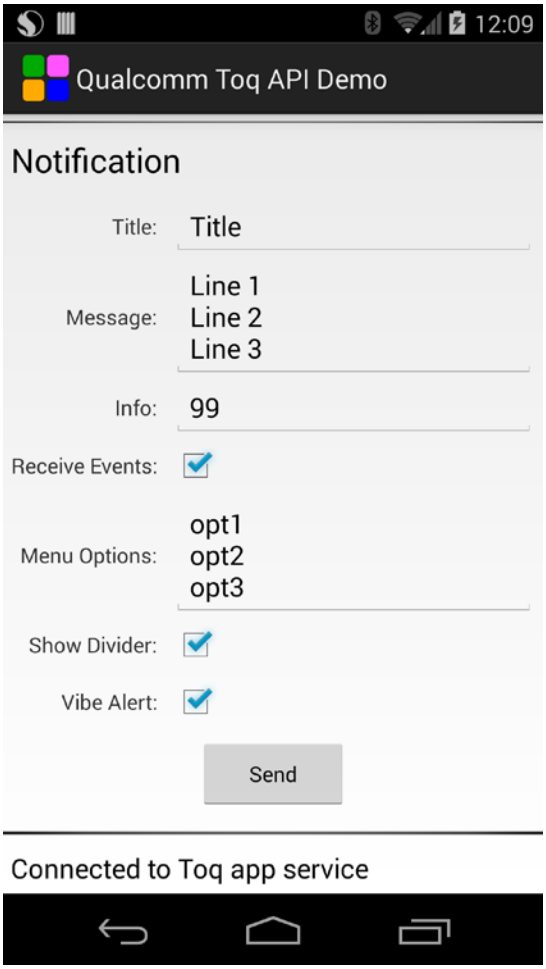


Figure 4. Sample app (Notification section)

6 Known Issues

The following known issues are currently being investigated for resolution in future releases.

- A third-party app must have been launched by the user at least once for any of its BroadcastReceivers to be registered by Android to receive a BroadcastIntent, including the Toq-specific com.qualcomm.toq.smartwatch.install.applet BroadcastIntent. This is an Android security restriction.
- Updating a third-party app with a new app name, after a deck of cards applet has been installed on the Toq smartwatch, requires a re-install of the applet in order to function correctly. Also, if the third-party app is persisting the deck of cards between instances, then a new deck of cards should be created rather than retrieving the existing version from storage to ensure that it picks up the new app name, before re-installing the deck of cards applet. This occurs when upgrading from the sample app v1.3 to v1.4 if a deck of cards applet has already been installed.
- A card's "info" field is not rendered when a card is displayed on the Toq smartwatch.
- It's not possible to update a deck of cards applet's launcher icons post installation.
- It's not possible to set the icon of a card used in a notification or deck of cards.