



Push Notifications

Overview

20 November 2011

Contents

Overview	3
FAQs	3
Key terms	5
General information about push notifications	8
Summary of API operations	10
Supported platforms	11
Helpful resources for application developers	11
How OpenMarket Push Notifications works	12
Index	13

Overview

FAQs

What is OpenMarket Push Notifications

OpenMarket Push Notifications is a web service that enables you to send a notification to an application on an end user's device, even when the application isn't running. You can send a notification to a single destination, multiple destinations, or you can broadcast it to a predefined group of destinations. Depending on the device platform, a notification can include a badge, text alert, or sound, or a combination of these.

What are push notifications

A push notification is a short message pushed to a specific application on an end user's smart device. The message informs the end user about an update that's available for the application, or an event relating to the application. For example, a push notification can be used to notify the end user about sports scores or stock movements. The Apple network supports such messages as text alerts, a badges or audible alerts, and combinations of these. Android supports *toast notifications*, a brief message that appears momentarily on the screen and messages to the phone's status bar.

Why use push notifications

Push notifications represent an additional messaging channel for your mobile strategy. Consider using push notifications to:

- ▶ Send alerts for new application features or updates
- ▶ Inform end users about breaking news relevant to the application
- ▶ Provide coupons or information about special offers
- ▶ Communicate timely information about events such as sports scores or stock movements
- ▶ Inform users of their turn to play in a multiplayer game
- ▶ Implement peer-to-peer messaging between users of an application

Any business that develops or is looking to develop smart phone applications should consider using Push Notifications as a messaging channel to compliment SMS or MMS messaging.

What notification platforms are supported

The current release of Push Notifications supports:

- ▶ Apple Push Notification Service (APNs) for sending notifications to iPhone, iPad, and iPod touch devices
- ▶ Android Cloud To Device Messaging (C2DM) for Android phones and tablets using Android 2.2 and above
- ▶ BlackBerry Push Service (BPS) for BlackBerry devices (specifically BPS Essentials)

OpenMarket intends to release support for additional platforms in 2011.

How do push notifications work

Smart devices provide controls that allow the end user to enable or disable push notifications for an application. When the end user installs an application, the device retrieves a destination ID from the network (e.g., APNs) and either the application itself or your system (depending on how you design the application) will register the destination ID with OpenMarket. Upon successful registration of the application instance, you can submit POST requests to send notifications to the end user. As an option you can define multiple destinations as a group, which allows you to send a notification to multiple destinations.

As of November 2011, OpenMarket Push Notifications supports message receipts, which enable the mobile application to report back whether a notification was delivered successfully, and whether and how the end user interacted with the notification.

How do you get provisioned to use the service

Contact your OpenMarket account manager. Once your account is provisioned you can log in to Customer Center to submit network credentials for the networks you intend to use. For more information about application setup, see the *OpenMarket Application Manager User Guide*.

Is reporting information available

No, initially there are no transactional reports. You can, however, use GET Notification Status for Instance to determine whether a notification's state is "processed" indicating the message was successfully passed to the network for delivery or "failed" meaning the message was rejected by the platform.

Can you test push notifications

Yes, OpenMarket provides a sandbox environment that connects to both the APNs and BPS sandbox (test) environments. Google does not provide a specific sandbox environment for Android testing, so OpenMarket's sandbox points to their production service.

Where can I get more information about APNs, C2DM, and BPS

This document provides information about OpenMarket Push Notifications and assumes that you are familiar with APNs, C2DM, and BPS. For a list of external resources see "Helpful resources" on page 10.

Key terms

APNs

The Apple Push Notification service (APNs) transports and routes push notification messages to Apple devices, including the iPhone, iPad, and iPod touch.

Application credentials

The credentials that an application receives when it registers with either APNs or C2DM.

Application ID

A name for the application that is provided by you during provisioning. The ID must be unique to the merchant. The OpenMarket Push Notification APIs use the AppId data element.

Badge

An Apple-specific type of push notification that displays an image or number on the application icon to indicate to the end user that an update is available for the application.

Base URL

The unique URL that RIM (for BlackBerry) provides when registering to use the BPS production environment.

BPS

BlackBerry Push Service is the service RIM provides to deliver push notifications to BlackBerry devices. OpenMarket specifically supports BPS Essentials.

C2DM

Cloud to Device Messaging is the Google service that provides push notification capabilities to Android devices (v2.2 and higher).

CollapseKey

If a notification is not sent to the device because the device is offline, C2DM will keep one notification per collapseKey. Others notifications will be overwritten. C2DM allows the submission of up to four different CollapseKeys. There is no guarantee which notification is kept and which is overwritten. APN performs a similar overwriting but does not allow additional CollapseKeys. For APN, subsequent notifications will be automatically overwritten.

Content Provider ID

An ID assigned by RIM to identify a content provider or merchant. The content provider ID is unique to each environment.

DelayWhileIdle

A server-side setting that controls how a notification is handled when the destination device is offline. When DelayWhileIdle is "true", C2DM will hold the notification. When DelayWhileIdle is "false", C2DM will discard the message.

Destination

The application instance on a particular device that will receive the push notification.

Destination ID

A location to which a push notification is sent, as specified by the device network. Apple uses the "device token" to specify the precise device on which the application is installed, C2DM uses Registration ID, and BlackBerry uses PIN.

Device network

Refers to the network serving the device—either APN, C2DM, or BPS.

Device token

An identifier that APN returns to the application instance when it registers for push notifications. APN uses the device token to locate the device on which the client application is installed. The device token is also used to authenticate the routing of a push notification.

Expiration

The date a notification will expire and should not be sent to the device. For APN, if the notification has not expired, the expiration will be sent to the APN for further use. For C2DM and BPS, OpenMarket checks the expiration date before attempting to send to the network.

Instance ID

The ID of a specific installed instance of an application on a device. OpenMarket defines this ID and returns it when an application instance is registered. The OpenMarket Push Notification operations use the InstanceId data element.

Message ID

Message identifier set by OpenMarket and included in the notification payload to the device.

Message receipt

The response from the end user's application indicating that a notification was received. Other reportable statuses are that the end user engaged with the application or deleted the notification.

Notification payload

With a push notification, the payload contains the message content and other optional information, such as how the device will alert the user to the update that is available for the target application.

PIN

An ID used by RIM (manufacturer of BlackBerry devices) to identify a specific device. A PIN is one type of destination ID.

pnKey

A unique key provided to the merchant during provisioning for Push Notifications, and intended for use when the merchant's application self-registers.

Push notification

A short message that an application server sends to a device, and in turn the device informs the end user about an update to a specific application.

Registration ID

The ID that is returned to the application instance when it registers for push notifications with C2DM. This is one type of DestinationId.

Status bar notification

An Android-specific feature in which a message is sent to the end user and added to the status bar. An intent can be initiated when the end user clicks the notification. An intent can be a ticker-text message, a sound or a flashing light.

Toast notification

An Android-specific feature in which a small, unobtrusive message appears briefly on the end-user's screen. It does not take focus or allow user input.

General information about push notifications

Push notifications enable you to send a short message to a registered application instance on an end user's smart device. Push notifications help extend the battery life of the device by allowing the application to be off while still enabling the device to receive information from the server side of the application. From the end user's perspective, push notifications are configurable; the user can choose to turn off all notifications, or notifications for certain applications.

How your application interfaces with OpenMarket and external networks

The server side of your application is the sender of a push notification. To route a notification to an application instance through OpenMarket, you first get provisioned to use OpenMarket Push Notifications.

- ▶ For Apple devices, you must obtain security certificates that OpenMarket uses to connect to APNs environments.
- ▶ For Android devices, C2DM requires a Google account, which OpenMarket uses to connect to the C2DM interface.
- ▶ For BlackBerry devices, BPS requires an application ID, password, and the Base URL that OpenMarket uses to connect to the BPS environments. RIM provides these credentials when you register for their environments.

Once you're provisioned, your application can submit a notification to OpenMarket's service. As the following diagram shows, OpenMarket pushes your notification to the appropriate platform for routing to the end user's device. You can submit notifications individually, to a list of instances or to groups, and to either to an individual network or to both.



Summary of API operations

Registration operations

Operation	Purpose
PUT Application Instance	Creates a new application instance.
PUT Updated Application Instance	Overwrites all existing information for the application instance, including the destination ID, the status information, and the group information.
GET Application Instance	Retrieves the application instance.
DELETE Application Instance	Disables the application instance.
POST Update Destination ID	Updates the destination ID of the application instance. Other data remains unchanged.
POST Add Group(s)	Adds one or more groups to an application instance.
DELETE Group(s)	Removes one or more groups from an application instance.

Create notification operations

Operation	Purpose
POST Notification	Creates a push notification to be sent to the device.

Callback operations

Operation	Purpose
GET Notification Status for an Instance	Retrieves the status of a push notification for an application instance.
PUT Message Receipt	Creates a message receipt for a push notification.

Supported platforms

OpenMarket Push Notifications currently supports:

- ▶ Apple Push Notification Service (APNs)
- ▶ Android C2DM (requires the end user to use Android 2.2 or higher)
- ▶ BlackBerry Push Service (specifically BlackBerry Push Essentials)

Helpful resources for application developers

This OpenMarket documentation assumes that you are familiar with application development for Apple, BlackBerry, and Android devices. If you are new to these technologies, here are some resources to help you get up and running.

Apple

- ▶ The Apple iOS Reference Library, in particular:
 - ▶ **Local and Push Notification Programming Guide**
(<http://developer.apple.com/library/ios/#documentation/NetworkingInternet/Conceptual/RemoteNotificationsPG/ApplePushService/ApplePushService.html>)
 - ▶ **The App Store Review Guidelines**
(<https://developer.apple.com/appstore/resources/approval/guidelines.html>)

Note that you'll need a username and password to access the guidelines.

C2DM

- ▶ **Android C2DM Service** (<http://code.google.com/android/c2dm/index.html>)
- ▶ **Android application publishing guide**
<http://developer.android.com/guide/publishing/publishing.html>

BlackBerry Push Service

- ▶ **Blackberry Push Service**
(<http://us.blackberry.com/developers/platform/pushapi.jsp>)
- ▶ **Blackberry Developer Zone** <http://us.blackberry.com/developers/>
- ▶ **Blackberry API Reference (Javadocs)**
<http://www.blackberry.com/developers/docs/5.0.0api/index.html>

How OpenMarket Push Notifications works

By integrating your application with OpenMarket Push Notifications, you can send a notification to a target application on an end user's device. The process of sending a notification involves these functions:

1. Registration of the application instance with OpenMarket Push Notifications.
2. A POST call to OpenMarket sends the push notification to one or more destinations, or to a group of destinations.
3. OpenMarket returns a ticket ID and sends the notification to the specified destination(s).

Registering the application instance

Registration of an application instance, and management of that information, involves supplying OpenMarket with the destination ID, and optionally group membership information. You can design your application to *self-register* by supplying the registration information directly to OpenMarket, or your application can provide the information to your servers which then pass it on to OpenMarket. Regardless of how you design your application, you use the same APIs.

Specifying the destinations

You have a couple options for how to specify the target destination of a push notification. You can submit a request to send a notification to one destination—one application instance. You can also define one or more groups of destinations, which lets you send a notification to multiple destinations. This capability is useful if you want to segment your customer base. You can also send a notification to a group defined as "all", which results in the notification being sent to all registered application instances that are enabled.

For additional information about application instance registration and groups, see [Deciding how to handle registration of application instances](#).

Receiving message status

You can design your application to respond to a push notification by sending back a status. Reportable statuses include:

- ▶ The notification was received
- ▶ The end user engaged with the application
- ▶ The end user deleted the message

Index

F

FAQs • 3

G

General information about push notifications • 8

H

Helpful resources for application developers • 11

How OpenMarket Push Notifications works • 12

K

Key terms • 5

O

Overview • 3

S

Summary of API operations • 10

Supported platforms • 11