



Push Notifications

Overview

22 April 2011

Who should use this interface?

This interface is intended for any customer who wants to deliver push notifications to an application on a device. Initial support is for applications on the Apple iPhone, iPad, and iPod touch devices, and on Android devices running 2.2 or above.

Contents

Overview	3
FAQs	3
Key terms	5
General information about push notifications	7
How OpenMarket Push Notifications works	9
Basic data flow	10
Data flow with application self-registration	11
Update a destination ID for an application instance	12
Delete an application instance	13
Detailed APNs flow	14
Detailed C2DM flow	15
Summary of operations	17
Supported notification platforms	18
States of a notification	18
Reasons why a notification can fail	18
Helpful resources	19
Index	20

Overview

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.

FAQs

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 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 the Apple Push Notification Service (APNs) for sending notifications to iPhone, iPad, and iPod touch devices, and Android Cloud To Device Messaging (C2DM) for Android phones and tablets using Android 2.2 and above.

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—APNs or C2DM—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.

How do you get provisioned to use the service

Contact your OpenMarket account manager. You'll need to have an active OpenMarket account and provide information about your application and network credentials.

Is reporting information available

No, initially there are no transactional reports. You can, however, use GET Notification Status 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 the APNs sandbox (test) environment. Android does not provide a specific sandbox environment, so OpenMarket's sandbox points to their production service.

Where can I get more information about APNs and C2DM

This document provides information about OpenMarket Push Notifications and assumes that you are familiar with APNs and C2DM. For a list of external resources see [Helpful resources](#) on page 19.

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 provided by the merchant during provisioning. 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.

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. Other 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.

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, while C2DM uses Registration ID.

Device network

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

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, OpenMarket will check the expiration 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.

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.

pnKey

A pnKey is 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 push notification is 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 a user experience perspective, push notifications on the device are configurable, as are push notifications per application. The end user can choose to turn off all notifications, or just notifications on 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, this involves obtaining security certificates that OpenMarket uses to connect to APNs environments. C2DM requires a Google account, which OpenMarket uses to connect to the C2DM interface.

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.



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 is submitted to OpenMarket to send 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.

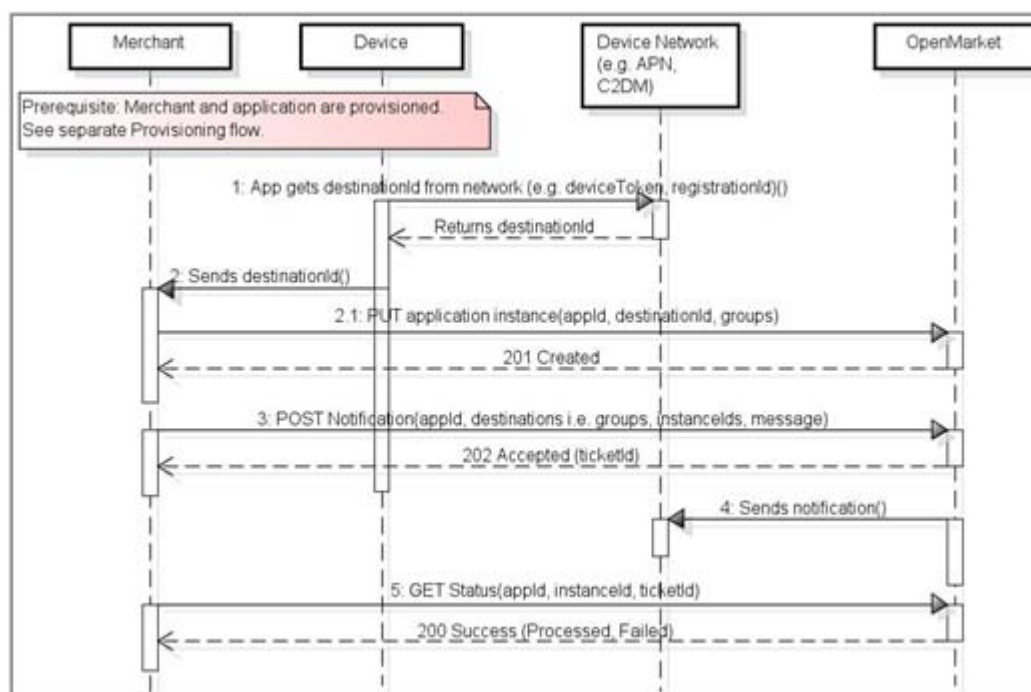
Specifying the destinations

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 gives you the ability to 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](#).

Basic data flow

This diagram depicts the interaction between your system, the device network, and OpenMarket when registration of the application instance is managed through your servers.



Steps in the process

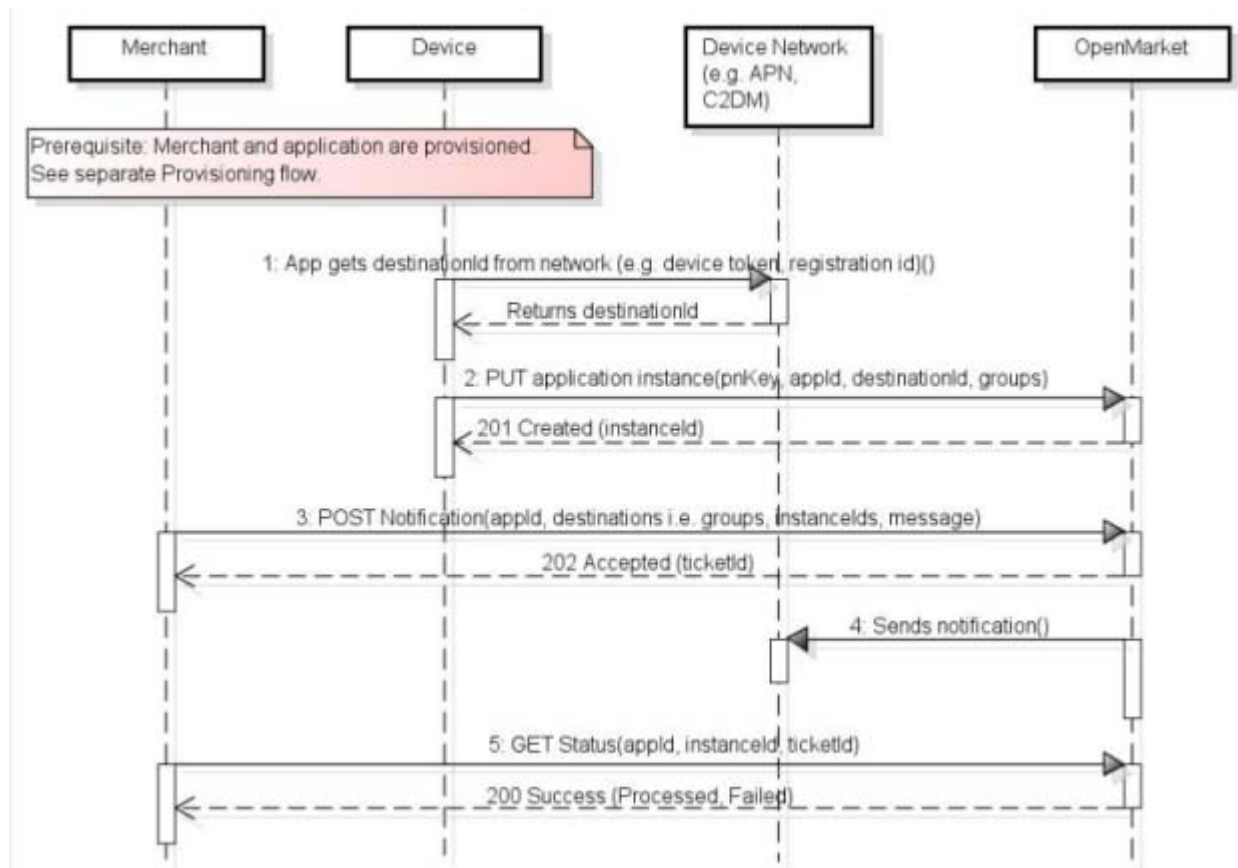
1. The end user launches an application that she previously downloaded to her device. (When she installed the application, she configured it to allow push notifications.) The application obtains a destinationId from the network and returns it to your servers.
2. Your system calls **PUT New Application Instance** to register the destination ID with OpenMarket Push Notifications.
3. OpenMarket responds with a success or fail, and if successful, returns the instance ID.
4. Upon successful device registration, your system submits a **POST Notification** to OpenMarket.
5. OpenMarket returns a ticket ID.
6. Your service can, as an option, call **GET Notification Status** to retrieve the state of the notification.

OpenMarket will return the status, appId, instanceId, ticketId, createdDate, and processedDate.

If the notification's state is "failed", OpenMarket returns a failure reason.

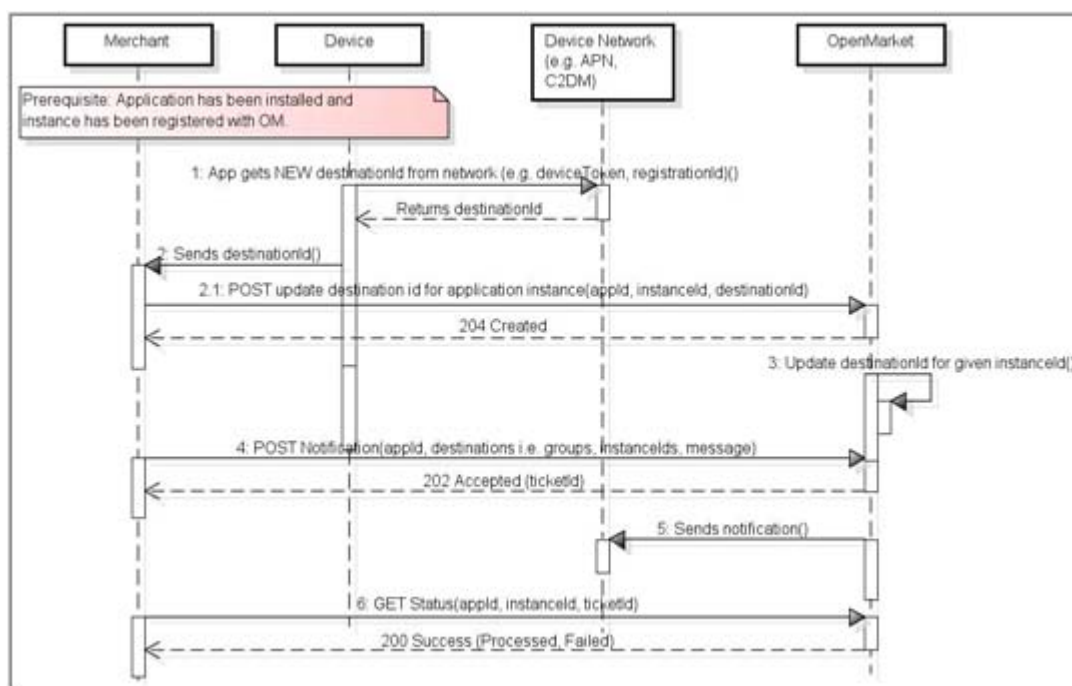
Data flow with application self-registration

Application self-registration involves providing the application instance ID and the destination ID, and optionally group membership information to OpenMarket. With self-registration, the application instance obtains the information from the network and reports it directly to OpenMarket.



Update a destination ID for an application instance

OpenMarket provides a convenience API for updating just the destination ID (either the APN device token or C2DM registration ID) for an application instance.



Steps in the process

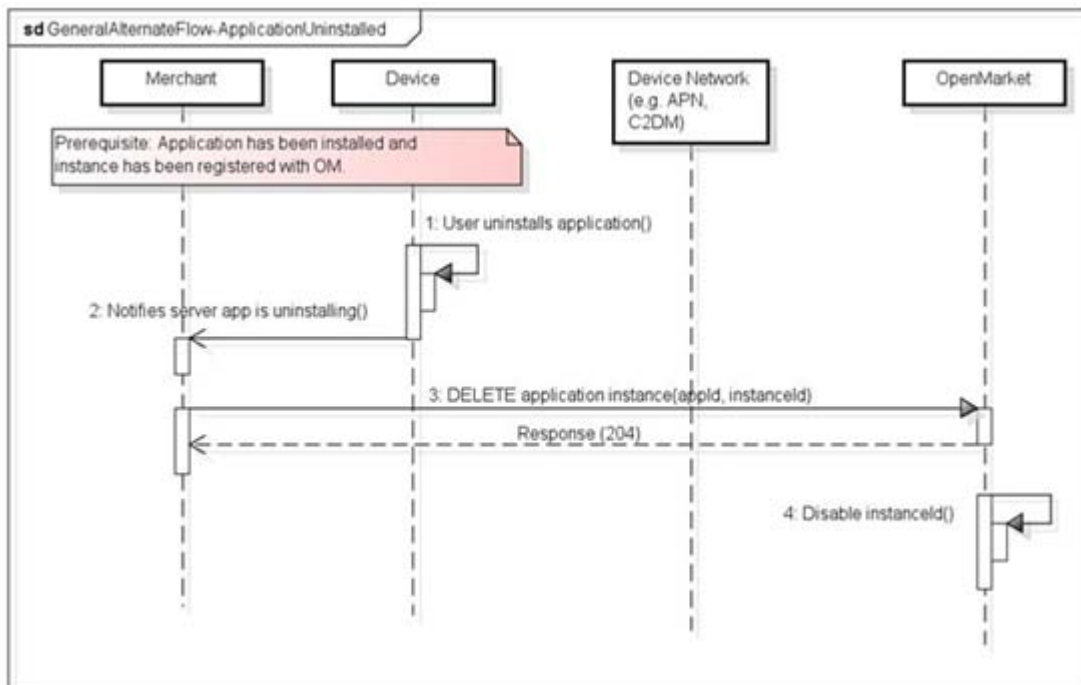
1. The end user launches an application that he previously downloaded to his device and configured to allow push notifications. Your system has already registered this instance with OpenMarket. The application obtains a new destination ID from the network and returns it to your servers.
2. Your system requests to update only the destination ID by calling **POST Updated Destination ID**.
All the other existing data remains unchanged.
3. OpenMarket responds with a success or failure.
4. Upon successful registration, your system submits a **POST Notification** to OpenMarket.
5. OpenMarket returns a ticket ID.
6. Your service can, as an option, call **GET Notification Status** to retrieve the state of the notification.

OpenMarket will return the state, appId, instanceId, ticketId, createdAt, and processedDate.

If the notification's state is "failed", OpenMarket returns a reason for failure.

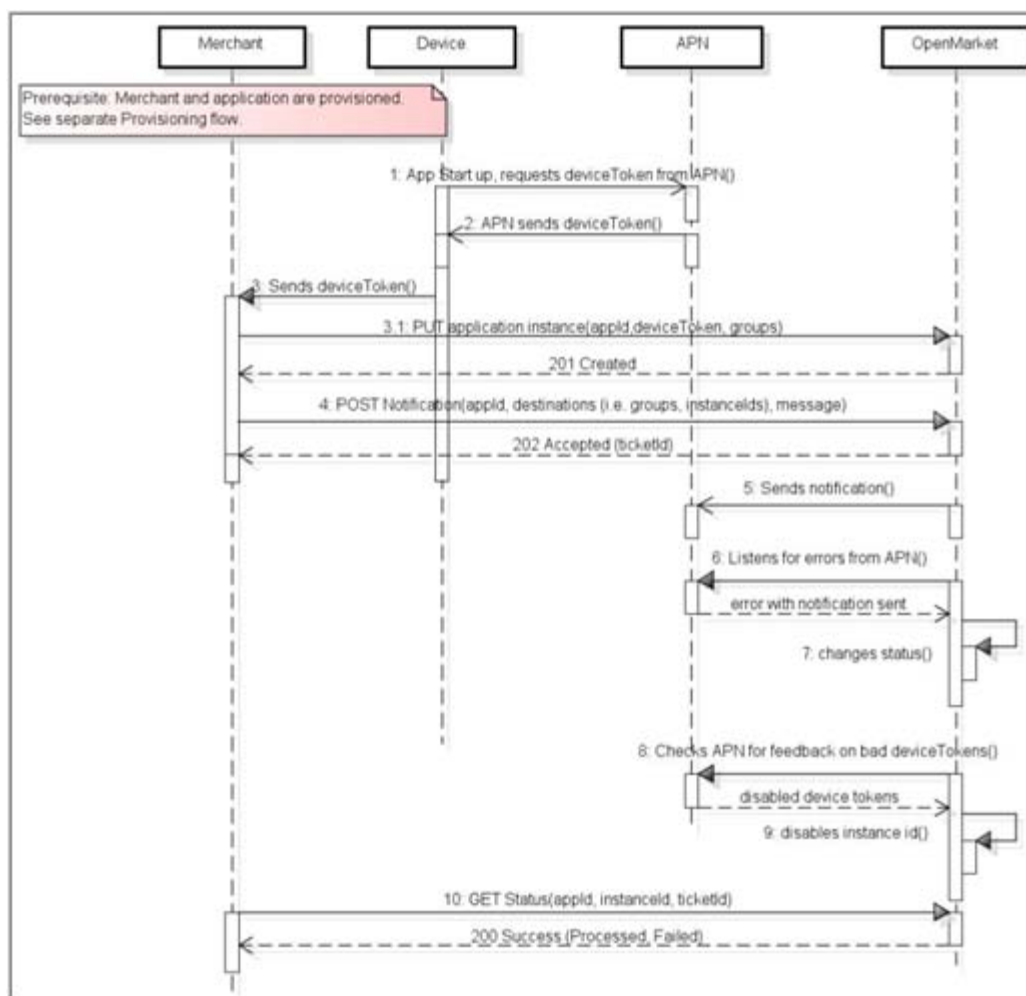
Delete an application instance

OpenMarket provides an API for deleting an application instance when the application has been uninstalled and you want to disable it within OpenMarket.



1. The end user uninstalls the application from her device, and the application notifies your servers.
2. Your system calls **DELETE Application Instance** to delete the instance from OpenMarket Push Notifications.
3. OpenMarket disables the application instance.
4. OpenMarket responds with a success or failure.

Detailed APNs flow

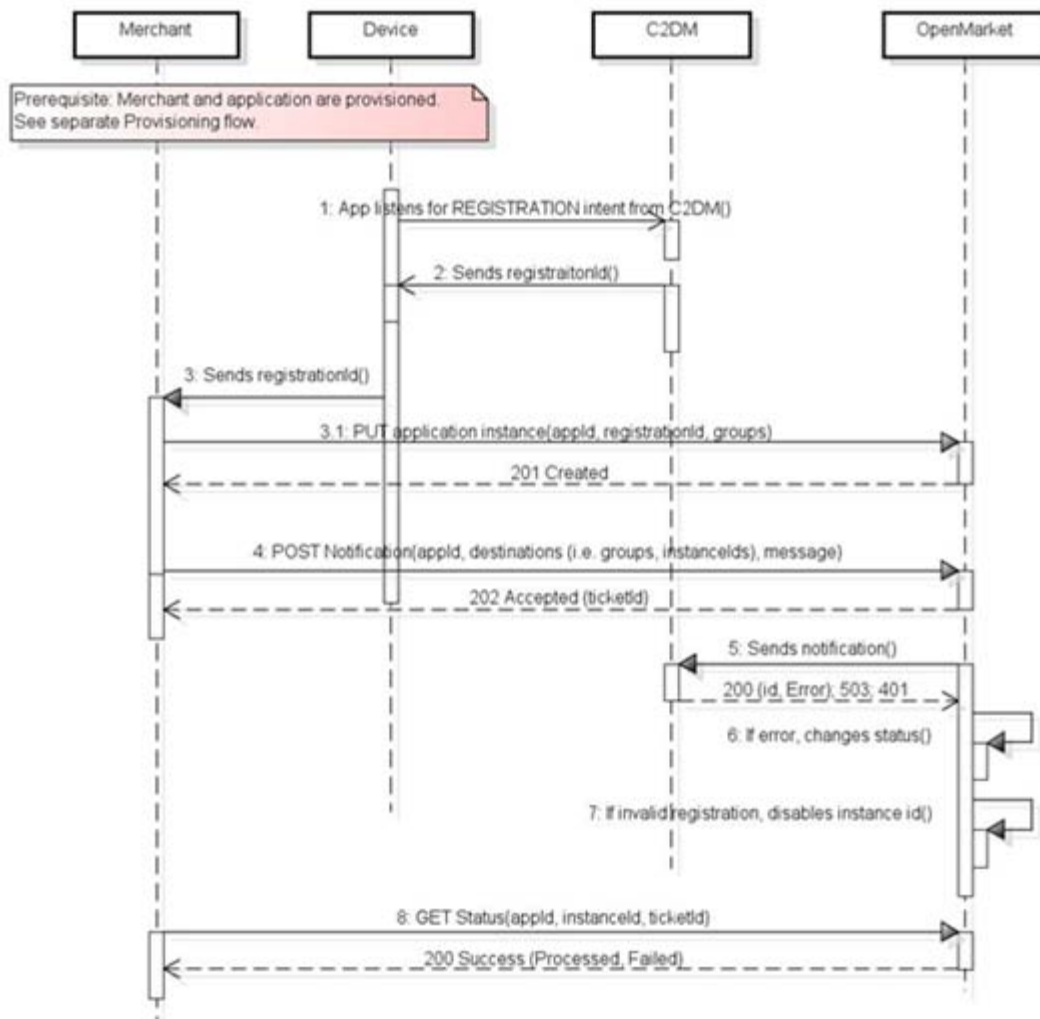


1. The end user launches an application that she previously downloaded to her device. (When she installed the application, she configured it to allow push notifications.) The application obtains a device token from the APNs and returns it to your servers.
2. Your system calls **PUT New Application Instance** to register the device token and optionally, the groups it belongs to.
OpenMarket responds with a success or failure, and if successful returns the instance ID.
3. Upon successful registration, your system submits a **POST Notification** to OpenMarket.
4. OpenMarket returns a ticket ID and sends the notification to the APNs.

5. OpenMarket listens for a response from the APNs. If an error is received, the status of the notification is updated with the error information.
6. OpenMarket also checks the APNs feedback service on a daily basis and disables any device tokens that the APNs indicates are now bad.
7. Your application can, as an option, call **GET Notification Status** to get the state of the notification.

OpenMarket returns the state, appId, instanceId, ticketId, createdAt, and processedDate. If the notification's state is "failed", the service returns a reason for failure.

Detailed C2DM flow



1. The end user launches an application that she previously downloaded to her device. (When she installed the application, she configured it to allow push notifications.) The application obtains a registration ID from C2DM and returns it to your servers.
2. Your system calls **PUT New Application Instance** to register the registration ID and the groups it belongs to with OpenMarket.

OpenMarket responds with a success or fail, and if successful, OpenMarket returns the instance ID.
3. Upon successful registration, your system submits a POST Notification to OpenMarket Push Notifications.
4. OpenMarket returns a ticket ID and sends the notification to the C2DM.
5. If OpenMarket receives an error, the status of the notification is updated with the error information.
6. OpenMarket disables any registration ID that the C2DM indicates is invalid.
7. Your service can, as an option, call GET Notification Status to get the state of the notification.
8. OpenMarket returns the state, appId, instanceId, ticketId, createdDate, and processedDate. If the notification's state is "failed", the service returns a reason for failure.

Summary of operations

The following tables list the operations supported by OpenMarket Push Notifications and provide links to the interface reference information.

Registration operations

Operation	Purpose
PUT Application Instance	Creates a new application instance in OpenMarket.
PUT Updated Application Instance	Replaces the existing instance information in OpenMarket, including the device token (APN) or registration ID (C2DM) for the instance ID, the status information, and the group information.
GET Application Instance	Retrieves the application instance in OpenMarket.
POST Update Destination ID	Updates the instance with a new Apple device token in OpenMarket, and may also include adding a group(s). 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.
DELETE Application Instance	Disables the application instance in OpenMarket.

Notification operations

Operation	Purpose
POST Notification	Creates a push notification to be sent to the device.
GET Notification Status	Retrieves the status of a notification.
DELETE Application Instance	Disables the application instance in OpenMarket.

Supported notification platforms

As of April 2011, OpenMarket Push Notifications supports:

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

OpenMarket intends to support additional notification networks in the future.

States of a notification

A push notification can be in one of these states:

- ▶ **Processed** — OpenMarket sent the notification to the device network or device.
- ▶ **Failed** — OpenMarket was unable to successfully send the notification to the device network or device due to an error such as an invalid deviceToken, invalid payload size, etc.

Reasons why a notification can fail

Several conditions can cause a notification to fail. These include:

- ▶ OpenMarket could not authenticate your call due to a problem with your OpenMarket credentials.
- ▶ OpenMarket does not have the APNs security certificate data for your application.
- ▶ OpenMarket does not have the correct credentials for your account registered with C2DM.
- ▶ Your call contains invalid parameters or uses an invalid URL.
- ▶ The destination ID (the device token or registration ID) may have changed. This may occur when the end user migrates to a new device or reinstalls the OS on the device.
- ▶ either because the end user's phone number changed or the end user reinstalled the OS on the device.

NOTE: APNs and C2DM do not guarantee delivery of a push notification, and there could be cases where the notification status cannot be determined. In such cases you should resend the notification.

Helpful resources

This OpenMarket documentation assumes that you are familiar with Apple and Android development environments. If you are new to the technology, here are some links to online resources you may find helpful.

Apple resources

- ▶ 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 resources

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

Index

B

Basic data flow • 10

D

Data flow with application self-registration • 11
Delete an application instance • 13
Detailed APNs flow • 14
Detailed C2DM flow • 15

F

FAQs • 3

G

General information about push notifications • 7

H

Helpful resources • 4, 19
How OpenMarket Push Notifications works • 9

K

Key terms • 5

O

Overview • 3

R

Reasons why a notification can fail • 18

S

States of a notification • 18
Summary of operations • 17
Supported notification platforms • 18

U

Update a destination ID for an application instance • 12

Copyright © 2011 OpenMarket Inc. All rights reserved. No part of this publication may be reproduced, distributed, electronically stored, or translated into any language, in any form by any means, without the prior written permission of OpenMarket. The trademarks and service marks of OpenMarket Inc., MX Telecom Inc., and MX Telecom LTD are the exclusive property of Amdocs, and may not be used without permission. All other marks are the property of their respective owners. OpenMarket is a business of Amdocs.
