



# Push Notifications

---

## Integration Guide

20 November 2011

### **Who should use this product?**

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

## Purpose of this guide

This guide is a technical reference for application developers who need to integrate their applications with the OpenMarket Push Notifications web service. The web service enables you to send a notification to an application on an end user's device, either to a single destination, multiple destinations, or to a predefined group of destinations.

Other documents you may find useful:

- ▶ *Push Notifications Overview*
- ▶ *Application Manager User's Guide*

### Change summary

Date	Description
1 August 2011	Added a new API for retrieving the status of a message receipt.
31 August 2011	Added a new API for retrieving summary status information about push notifications and message receipts.
20 November	Made minor editorial corrections and created a separate document for using the Application Manager. (See the <i>Application Manager User's Guide</i> .)

# Contents

---

## Chapter 1. Using Push Notifications 6

---

Prerequisites to using the service.....	6
Getting set up for APNs .....	7
Exporting Apple Push Certificate to a .p12 file .....	7
Getting set up for C2DM .....	8
Enabling your application to use C2DM .....	8
Special requirements for BPS .....	10
Enabling your application to use BPS .....	10
Deciding how to handle registration of application instances .....	11
Use cases you can implement with OpenMarket Push Notifications.....	12
Registering the application instance where your server manages the process .....	13
Data flow with application self-registration .....	14
Updating just the destinationId for an application instance .....	16
Deleting an application instance .....	17
Pushing a notification to an Apple device .....	19
Pushing a notification to an Android device .....	21
Pushing a notification to a BlackBerry device .....	23
Creating a message receipt.....	25
Connecting to the service.....	26
Testing a push notification .....	27
Reasons why a push notification can fail.....	28

## Chapter 2. API Reference 29

---

PUT New Application Instance .....	29
Request URL syntax, example, and parameters .....	29
Request header parameters .....	30
Request elements and attributes .....	31
XML schemas .....	33
Example request body .....	34
Response header parameters .....	35
Response elements and attributes .....	35
Example response body .....	38
Error conditions .....	38
PUT Updated Application Instance .....	39
Request URL syntax, example, and parameters .....	40
Request header parameters .....	40
Request elements and attributes .....	41
XML schemas .....	43
Example request body .....	44
Response header parameters .....	45
Response elements and attributes .....	45
Example response body .....	49

Error conditions .....	49
GET Application Instances information (server-side) .....	50
Request URL syntax, example, and parameters .....	51
Request header parameters .....	54
XML schemas for GET Application Instance .....	55
Response header parameters .....	55
Response elements and attributes .....	55
Example response body .....	59
Error conditions .....	60
GET Application Instance information (device-side) .....	60
Request URL syntax, example, and parameters .....	61
Request header parameters .....	62
Response header parameters .....	62
Response elements and attributes .....	63
Example response body .....	66
Error conditions for GET Application Instance .....	66
DELETE Application Instance .....	67
Request URL syntax, example, and parameters .....	67
Request header parameters .....	68
Response header parameters .....	69
Error conditions with DELETE Application Instance .....	69
POST Updated Destination ID .....	70
Request URL syntax, example, and parameters .....	70
Request header parameters .....	72
Response header parameters .....	73
Error conditions for POST Updated Destination .....	73
POST Add Group(s) to an application instance .....	73
Request URL syntax, example, and parameters .....	74
Request header parameters .....	75
Response header parameters .....	76
Error conditions for POST Add Group(s) and DELETE Group(s) .....	76
DELETE Group(s) from an application instance .....	76
Request URL syntax, example, and parameters .....	77
Request header parameters .....	78
Response header parameters .....	78
POST Notification .....	79
Request URL syntax, example, and parameters .....	79
Request header parameters .....	80
Request elements and attributes .....	80
XML schemas .....	85
Example request body .....	86
Conditions under which OpenMarket disables an application instance .....	87
C2DM data handling considerations .....	88
BlackBerry considerations .....	88
Response header parameters .....	89
Error conditions .....	89
PUT Message Receipt .....	90
Request URL syntax, example, and parameters .....	90
Request header parameters .....	91
XML schemas .....	92
Response header parameters .....	92
Error conditions .....	92
GET Notification Status for an Instance .....	93

- Request URL syntax, example, and parameters ..... 93
- Request header parameters ..... 94
- XML schemas ..... 94
- States of a notification..... 95
- Response elements and attributes ..... 95
- Example response body ..... 98
- Error conditions ..... 98
- GET Notification Status Summary ..... 99
  - Request URL syntax, example, and parameters ..... 99
  - Request header parameters ..... 101
  - Response elements and attributes ..... 101
  - XML schemas ..... 106
  - Example responses ..... 106
  - Error conditions ..... 108
- XML schemas ..... 109

## Chapter 2. Using Push Notifications

---

### Prerequisites to using the service

Before you use OpenMarket Push Notifications, you need to register your application with the networks you intend to use (APNs, C2DM, and/or BPS). The "Getting set up" topics in this section provide some information about how to register your application with the device networks.

Once you've registered your application, you can log in to OpenMarket's Customer Center and provide us with the network credentials for each network you plan to use.

In general here's what you need to do in order to start using OpenMarket Push Notifications:

- ▶ Make sure you have an OpenMarket account with valid authentication credentials.
- ▶ To register your application with OpenMarket, you will need to provide:
  - ▶ The application ID you intend to use with all push notifications activity for the particular application. Each application requires a separate ID.
  - ▶ The appropriate network credentials required by the device network (APNs, C2DM, BPS). Note that you can register for all networks at this time or register with just one and do the others at another time.

**APNs** — A .p12 certificate for the application and the password you used to create it. (Each application on each environment requires a certificate.)

**C2DM** — The Google account username and password that you used to register your application. Google recommends that you have a separate account for each application.

**BPS** — The application ID and password, and the Base URL provided by RIM when you register with their environments.

- ▶ Contact your OpenMarket account manager to start the process. Once your account is set up to use Push Notifications, you can log in to Customer Center and access the Application Manager tool to set up your application. For details see the *OpenMarket Application Manager User Guide*.

## Getting set up for APNs

Your APNs network credentials consist of a .p12 certificate and password. Below are the steps to register your application with APNs to obtain your Apple certificate.

- ▶ Register your application with Apple and request a push notification certificate. The certificate identifies you when communicating with APNs over SSL. You'll need one certificate per application and will need to provide each certificate to OpenMarket.

OpenMarket recommends that you create a separate private/public key pair for use with your push certificate rather than use the one you created to sign your application. If you do use that key pair then you will have to provide it to OpenMarket.

- ▶ Use the Customer Center Application Manager to set up your application in both the APN sandbox (test) and production environments. Each environment requires a separate push notification certificate.

---

**NOTE:** If your application is already in Apple's production environment, you do not need to register it for the sandbox environment.

---

## Exporting Apple Push Certificate to a .p12 file

Follow the steps below to export your Apple certificate to .p12 format and create a password for it. You'll need to provide the certificate and password to OpenMarket. Note that you will need a separate certificate for each application and each environment (sandbox and production).

1. On your local Mac, double-click the .cer file that you received from Apple to import your Push Notification certificate into your keychain.  
  
The Keychain Access tool will launch.
2. Filter by the Certificates category and you'll see your certificate with the prefix "Apple Development Push Services".
3. Double-click the certificate to get detailed information such as User ID; this should correspond to the Apple application ID.
4. Right-click the certificate title and select **Export Apple Development Push Services ID**.
5. Choose a filename and location, and keep the default file format of .p12.
6. Include a password that you will provide to OpenMarket.

## Getting set up for C2DM

Your C2DM network credentials consist of a username and password for the Google account you set up for your application. Below are the steps to register your application with C2DM.

- ▶ Create a Google account that you will use to send notifications through C2DM for your application. OpenMarket recommends that you create an account for each application.
- ▶ Sign up to use the C2DM service.
- ▶ If you have an existing application, you will need to enable it to use C2DM. To use C2DM, the end-user must be using Android 2.2 and above.
- ▶ Use the Customer Center Application Manager to set up your application in the production environment.

## Enabling your application to use C2DM

Follow these steps to integrate your application with C2DM:

1. Permit your application to register and receive data messages.
2. Declare use of the C2DM library.
3. Have your application use the REGISTER and RECEIVE filters to get the registration ID and to receive data messages.

These steps involve editing the manifest file.

### Editing the manifest file

Add the following XML to your manifest file:

```
<!-- Only this application can receive the messages and registration
result -->
<permission
android:name="com.openmarket.android.permission.C2D_MESSAGE"
android:protectionLevel="signature"/>
<uses-permission
android:name="com.openmarket.android.permission.C2D_MESSAGE"/>
<!-- This app has permission to register and receive data message -->
<uses-permission
android:name="com.google.android.c2dm.permission.RECEIVE"/>
<!-- Permissions for internet access -->
<uses-permission android:name="android.permission.INTERNET"/>
<!-- App must have this permission to use the library -->
<uses-permission android:name="android.permission.WAKE_LOCK"/>

<application...your application activities etc here

    <!-- In order to use the c2dm library, an
         application must declare a class with the name
C2DMReceiver, in its
         own package, extending
com.google.android.c2dm.C2DMBaseReceiver

        It must also include this section in the manifest,
replacing
        "com.google.android.apps.chrometophone" with its package
name. -->
    <service android:name=".C2DMReceiver"/>
    <!-- Only google service can send data messages for the
app. If permission is not set -
        any other app can generate it -->
    <receiver
android:name="com.openmarket.android.c2dm.C2DMBroadcastReceiver"
android:permission="com.google.android.c2dm.permission.SEND">
        <!-- Receive the actual message -->
        <intent-filter>
            <action
android:name="com.google.android.c2dm.intent.RECEIVE"/>
            <category android:name="com.openmarket.android"/>
        </intent-filter>
        <!-- Receive the registration id -->
        <intent-filter>
            <action
android:name="com.google.android.c2dm.intent.REGISTRATION"/>
            <category android:name="com.openmarket.android"/>
        </intent-filter>
    </receiver>
</application>
<uses-sdk android:minSdkVersion="8"/>
```

## Special requirements for BPS

In order to send notifications to BlackBerry devices, you must register with BlackBerry Push Service (BPS).

- ▶ Register for BPS evaluation credentials at:  
<https://www.blackberry.com/profile/?eventId=8121>
- ▶ RIM will send you an email with your BPS credentials.
- ▶ Log in to Customer Center Application Manager to set up your application in the sandbox environment. Once you've done this you can start testing your application against the OpenMarket sandbox.
- ▶ When you're ready to move to production, register for the BPS production environment to obtain production credentials and base URL:  
<https://www.blackberry.com/profile/?eventId=8207>
- ▶ Log in to Customer Center Application Manager to set up your application in the production environment.

---

**NOTE:** OpenMarket supports BlackBerry Push Essentials, but not BlackBerry Push Plus.

---

## Enabling your application to use BPS

The following steps explain how to integrate your application with BPS. Note that this information assumes that you've written your application for BlackBerry Device Software 5.0. Although push notifications are supported on earlier versions, the methodology is different and not documented here.

### Register with RIM for signature keys

In order for your application to receive push notifications, it must be signed using signature keys provided by RIM. You can register at:

<https://www.blackberry.com/SignedKeys/>

### Ensure that your application's class has this structure

A BlackBerry application that can receive push notifications will have a class with the following structure:

- ▶ The class implements the `net.rim.blackberry.api.push.PushApplication` interface

- ▶ The class constructs an instance of `net.rim.blackberry.api.push.PushApplicationDescriptor`, passing in:
  - ▶ Application ID
  - ▶ Port
  - ▶ Base URL (this will be either `http://pushapi.eval.blackberry.com` for the sandbox environment, or the URL provided by RIM for its production environment)
  - ▶ `PushApplicationDescriptor.SERVER_TYPE_BPAS`
  - ▶ `ApplicationDescriptor.currentApplicationDescriptor()`

The class uses this to call

```
net.rim.blackberry.api.push.PushApplicationRegistry.getStatus().
```

- ▶ If the result of this call is anything except `STATUS_ACTIVE` or `STATUS_PENDING`, then your class should call 

```
net.rim.blackberry.api.push.PushApplicationRegistry.registerApplication()
```
- ▶ This class overrides two methods:
  - ▶ `onStatusChange()` method is called when the subscription state changes
  - ▶ `onMessage()` method is called when a push notification arrives

This is where your application receives the push notification and does something with it. Generally, to prevent locking up the event thread, the `onMessage()` method should hand the push notification off to another thread for processing. Optionally, your application can call `UIApplication.getUIApplication().requestForeground()` to make the application visible.

## Deciding how to handle registration of application instances

To register an application instance and manage registration information, you supply OpenMarket with the application instance ID and the destination ID, and optionally group membership information. You can either design your application to supply 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.

### Example registration scenarios

Company A wants to manage the registration data themselves, so they designed their application to communicate all activity to their servers. Whenever an application instance needs to be registered or updated, Company A's servers send a request to OpenMarket.

In contrast, Company B prefers to not collect store registration information and has designed its application to interact directly with OpenMarket. For example, when an end user downloads the application, the application instance on the device calls OpenMarket to create a new application instance. This is referred to as *application self-registration*. Additionally, Company B's application sends in the end user's postal code, which enables OpenMarket to add the destination to a group. Now Company B can easily send a push notification to devices with that postal code.

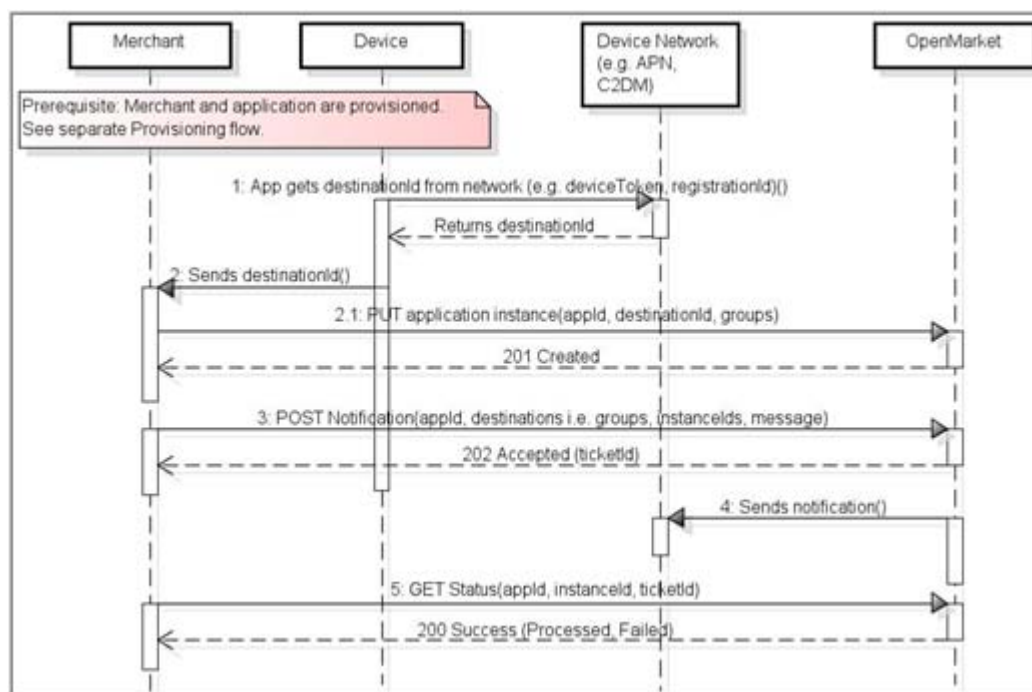
## Use cases you can implement with OpenMarket Push Notifications

The OpenMarket Push Notifications web service supports these use cases:

- ▶ Create, update, and delete an application instance
- ▶ Get an application instance
- ▶ Update the destination ID for an application instance
- ▶ Update the group membership for an application instance
- ▶ Create a push notification for an application instance
- ▶ Create a message receipt
- ▶ Get the status of a push notification

## Registering the application instance where your server manages the process

The following diagram depicts the interaction between your system, the device network, and OpenMarket when your server manages registration of the application instance. This flow assumes that your application has registered with the device network.



### 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 `destinationId` with OpenMarket Push Notifications.
3. OpenMarket responds with a success or fail, and if successful, returns the `instanceId`.
4. Upon successful device registration, your system submits a **POST Notification** to OpenMarket.
5. OpenMarket returns a `ticketId`.

6. Your service can, as an option, call **GET Notification Status for an Instance** 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.

**See Also**

[PUT New Application Instance](#) on page 29

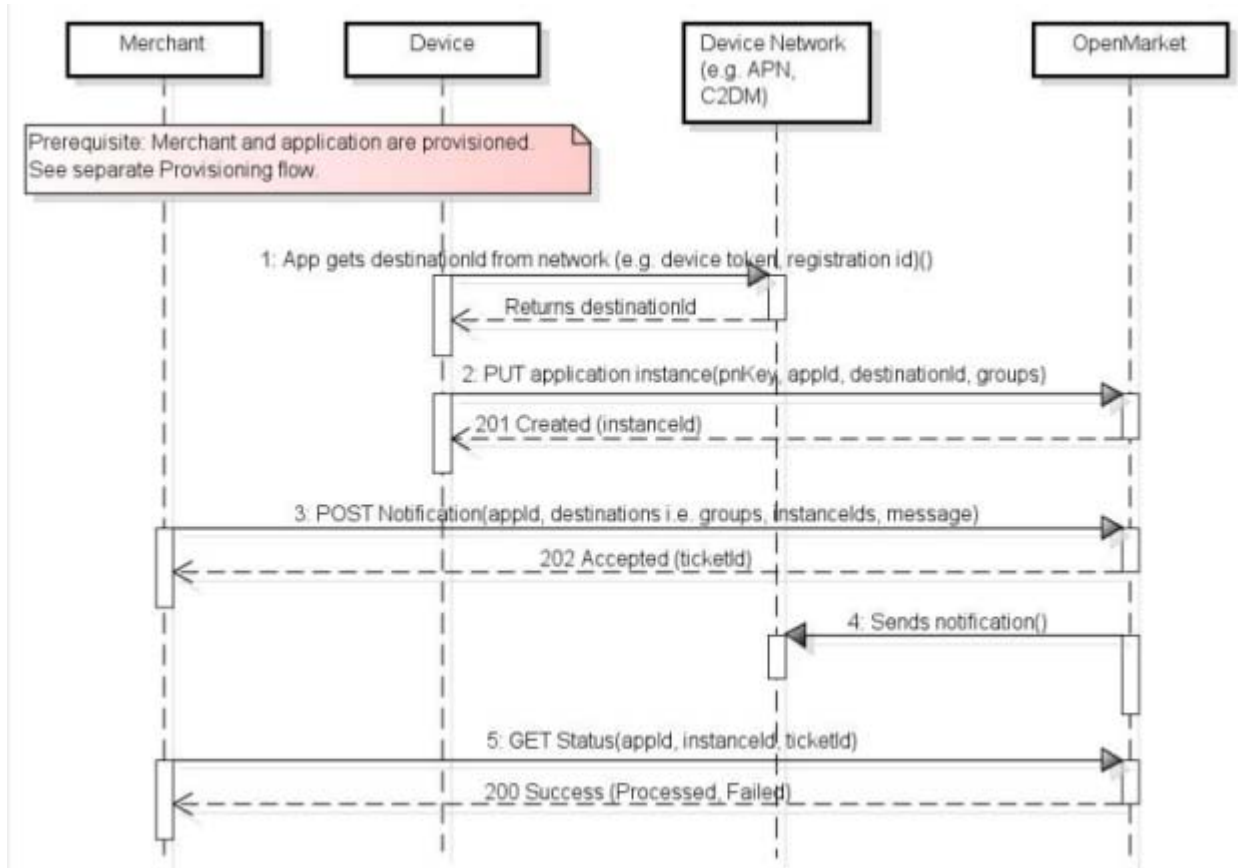
[POST Notification](#) on page 79

[GET Notification Status for an Instance](#) on page 93

## Data flow with application self-registration

The following diagram depicts the interaction between your system, the device network, and OpenMarket when OpenMarket manages registration of the application instance. This is also known as *self-registration*, in which your application provides 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.

This flow assumes that your application has registered with the device network.



**See Also**

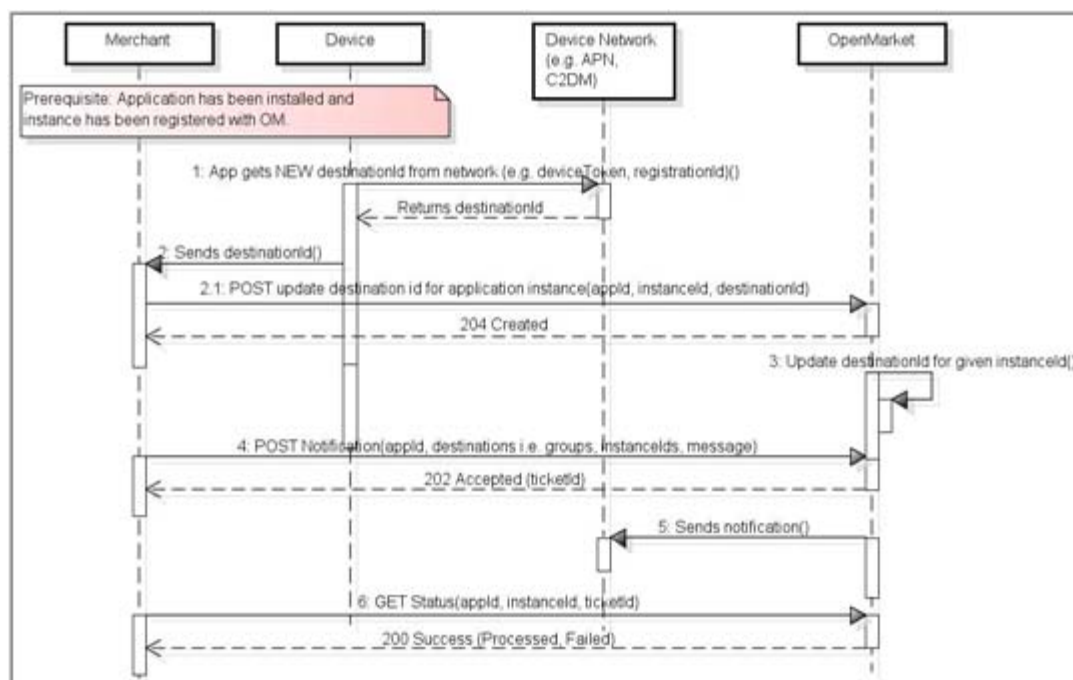
**PUT New Application Instance** on page 29

**POST Notification** on page 79

**GET Notification Status for an Instance** on page 93

## Updating just the destinationId for an application instance

The following diagram depicts the process for updating just the destinationId (either the APN device token, C2DM registration ID, or BlackBerry PIN) for an application instance. This is a convenience API.



### 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 `destinationId` 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 `ticketId`.

- 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.

**See Also**

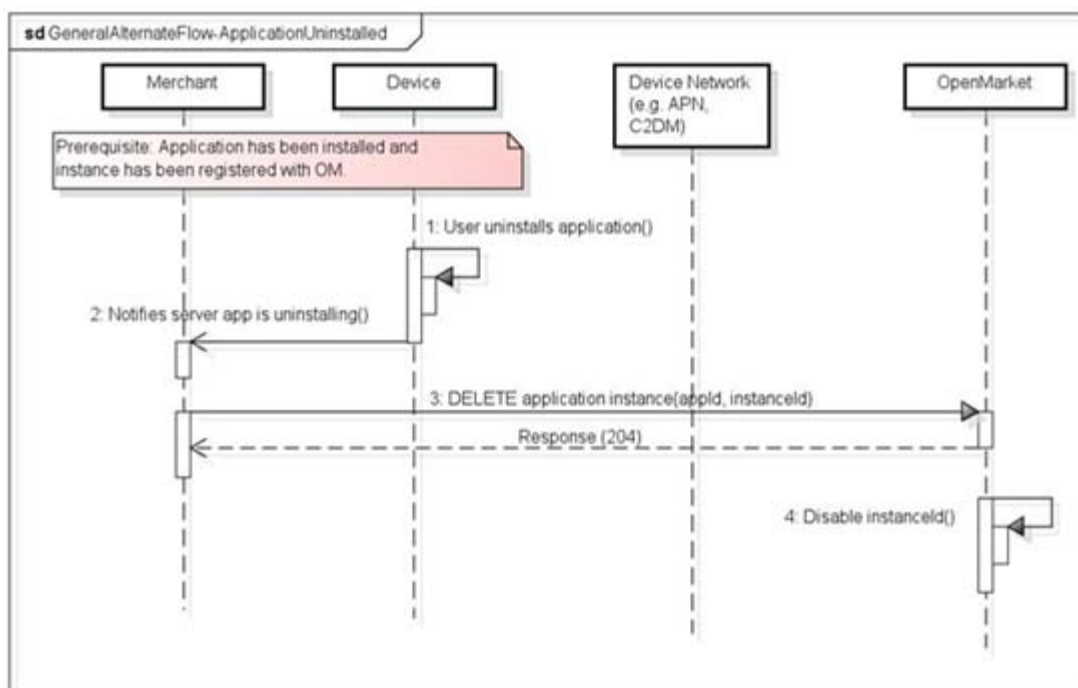
[POST Updated Destination ID on page 70](#)

[POST Notification on page 79](#)

[GET Notification Status for an Instance on page 93](#)

## Deleting an application instance

The following diagram depicts the process for deleting an application instance when the end user uninstalls the application and you want to disable the application instance within OpenMarket.



### Steps in the process

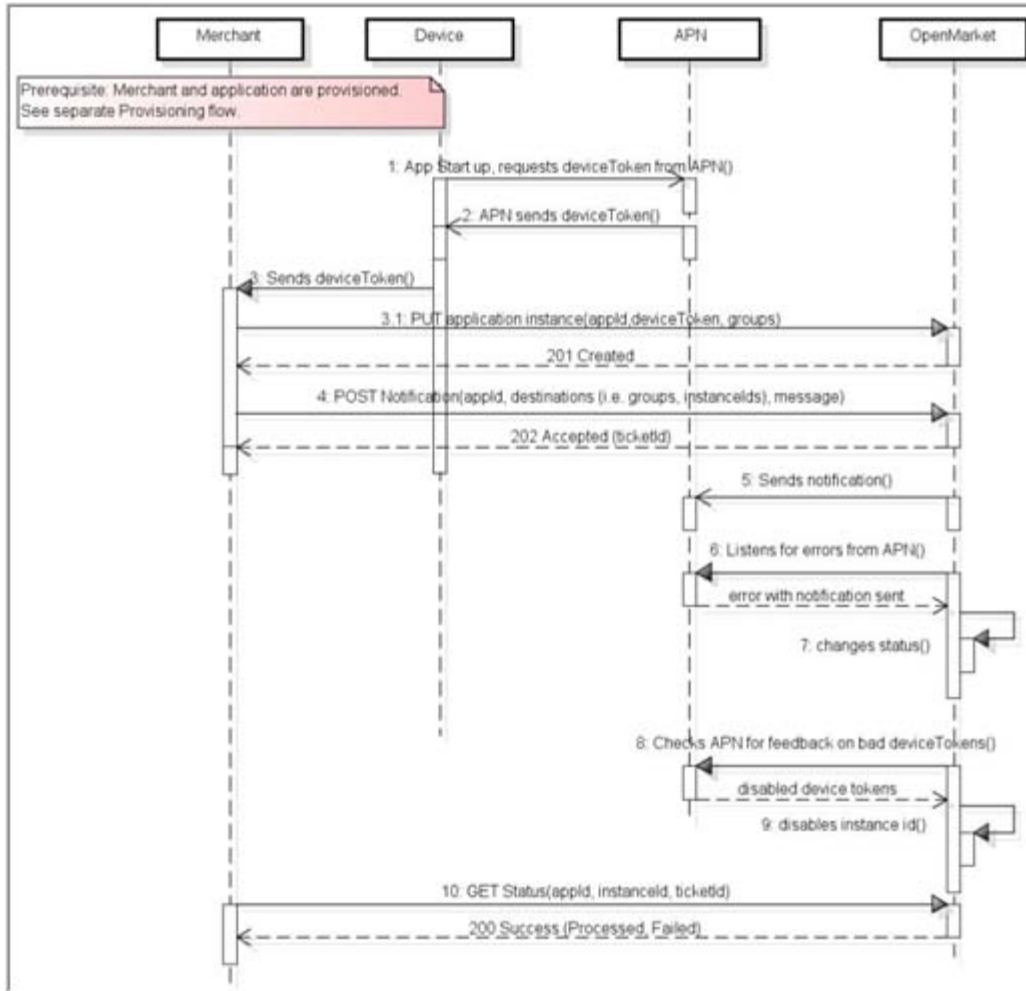
1. The end user uninstalls the application from her device, and the application notifies your servers.
2. Your system calls **DELETE Application Instance** to delete the instance from OpenMarket Push Notifications.
3. OpenMarket disables the application instance.
4. OpenMarket responds with a success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

[See Also](#)

[DELETE Application Instance](#) on page [67](#)

## Pushing a notification to an Apple device

The following diagram depicts the process of pushing a notification to an application instance on an Apple iPhone, iPad, or iPod. This flow assumes that the application is registered in the APNs and that the end user has configured the application to allow push notifications.



### Steps in the process

1. The end user launches an application that she previously downloaded to her device. 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 for an Instance** to get the state of the notification.

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

#### See Also

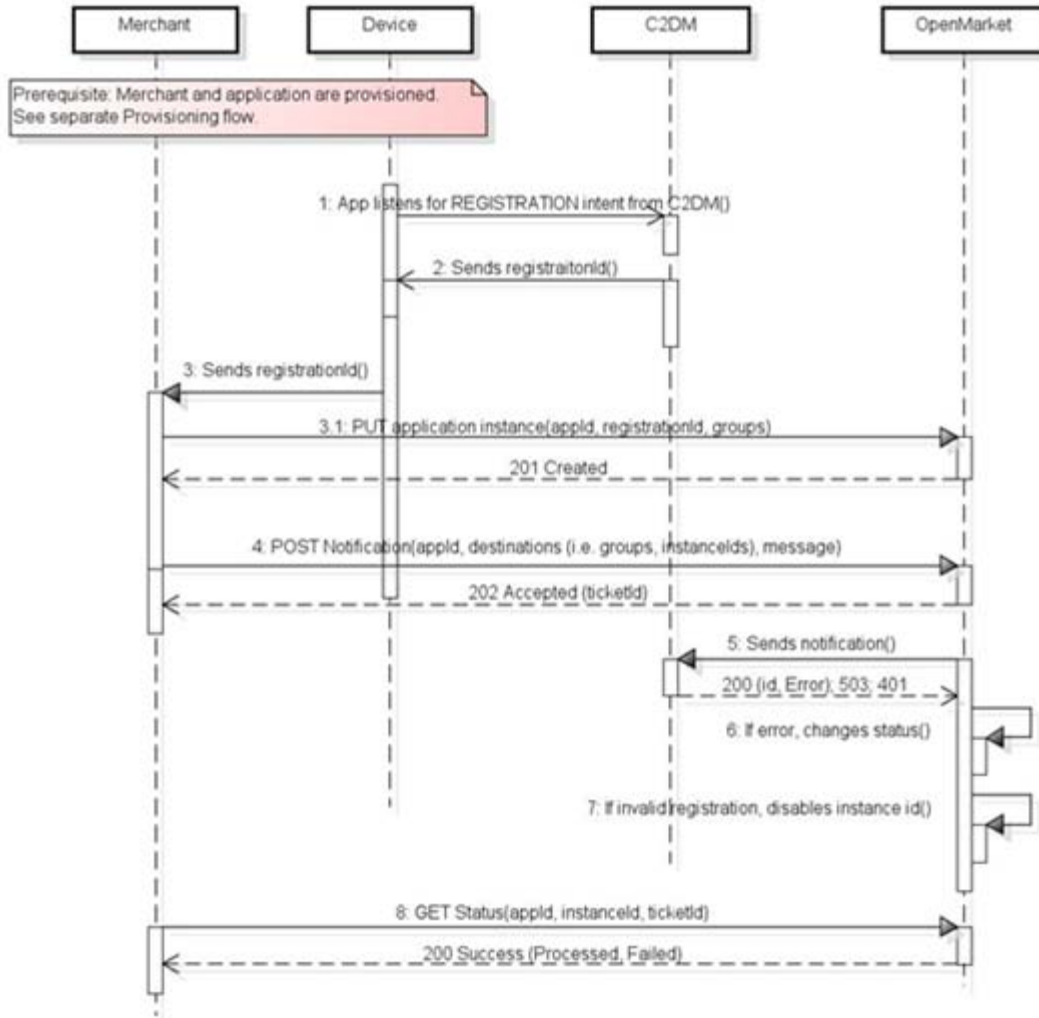
**PUT New Application Instance** on page [29](#)

**POST Notification** on page [79](#)

**GET Notification Status for an Instance** on page [93](#)

## Pushing a notification to an Android device

The following diagram depicts the process of pushing a notification to an application instance on an Android device. This flow assumes that the application is registered in the C2DM and that the end user has configured the application to allow push notifications.



### Steps in the process

1. The end user launches an application that she previously downloaded to her Android device. The application obtains a `registrationId` from C2DM and returns it to your servers.
2. Your system calls **PUT New Application Instance** to register the `registrationId`, 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 `ticketId` 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 `registrationId` that the C2DM indicates is invalid.
7. Your service can, as an option, call **GET Notification Status for an Instance** to get the state of the notification.

#### See Also

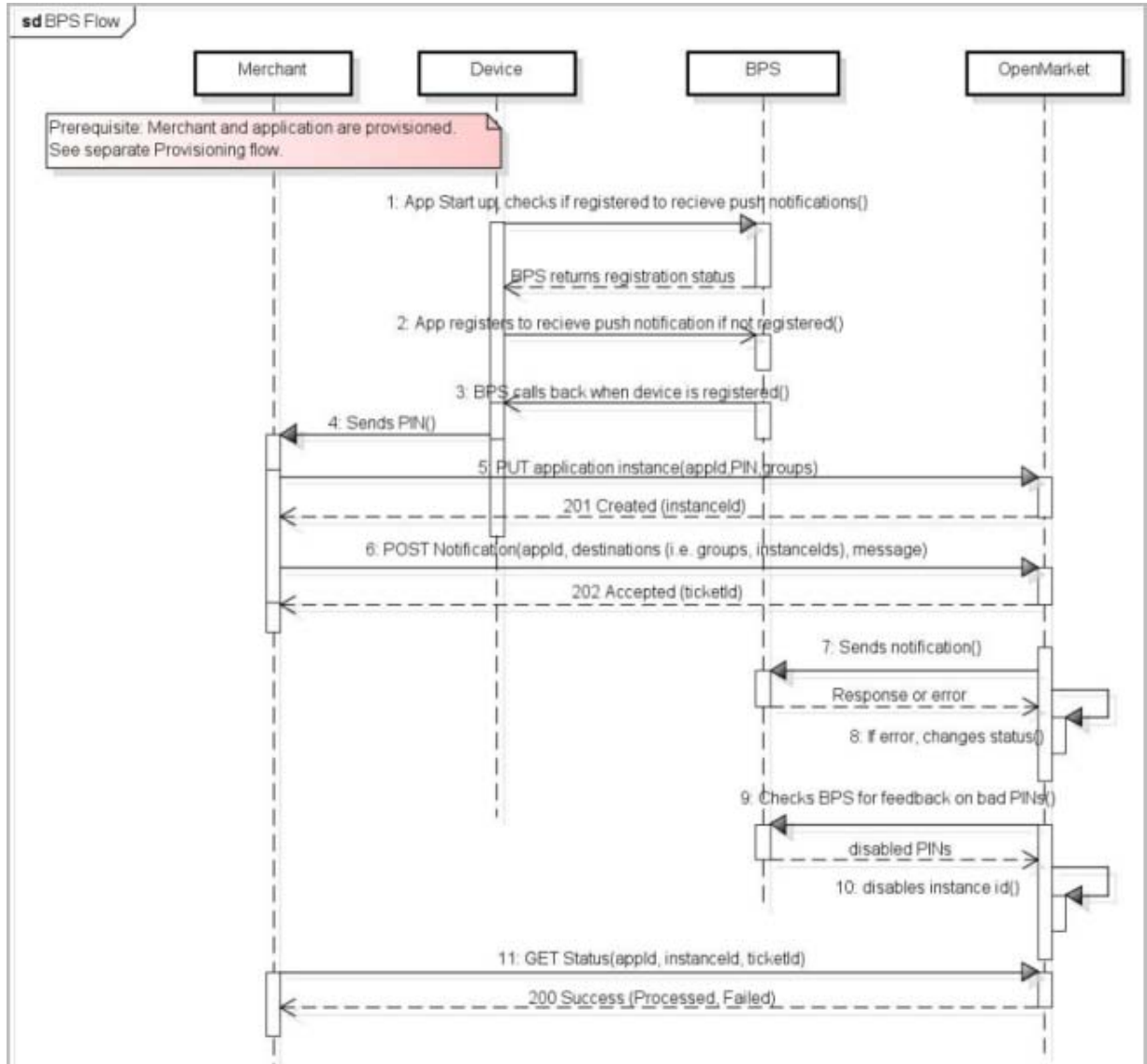
**PUT New Application Instance** on page [29](#)

**POST Notification** on page [79](#)

**GET Notification Status for an Instance** on page [93](#)

## Pushing a notification to a BlackBerry device

The following diagram depicts the process of pushing a notification to a BlackBerry device. This flow assumes that the application is registered in the BPS and that the end user has configured the application to allow push notifications.



### Steps in the process

1. The end user launches an application that she previously downloaded to her device. The application obtains a PIN from the BPS and returns it to your servers.
2. Your system calls **PUT New Application Instance** to register the PIN and, optionally, the groups it belongs to with OpenMarket.

OpenMarket responds with a success or fail, and if successful, OpenMarket returns the `instanceId`.

3. Upon successful registration, your system submits a **POST Notification** to OpenMarket Push Notifications.
4. OpenMarket returns a `ticketId` and sends the notification to the BPS.
5. If OpenMarket receives an error, we update the status of the notification with the error information.

OpenMarket also checks the BPS feedback service on a daily basis and disables any PINs that the BPS indicates are no longer valid.

6. Your service can, as an option, call **GET Notification Status for an Instance** to get the state of the notification.
7. OpenMarket returns the state, `appId`, `instanceId`, `ticketId`, `createdDate`, and `processedDate`. If the notification's state is "failed", the service returns a reason for failure.

### See Also

**PUT New Application Instance** on page [29](#)

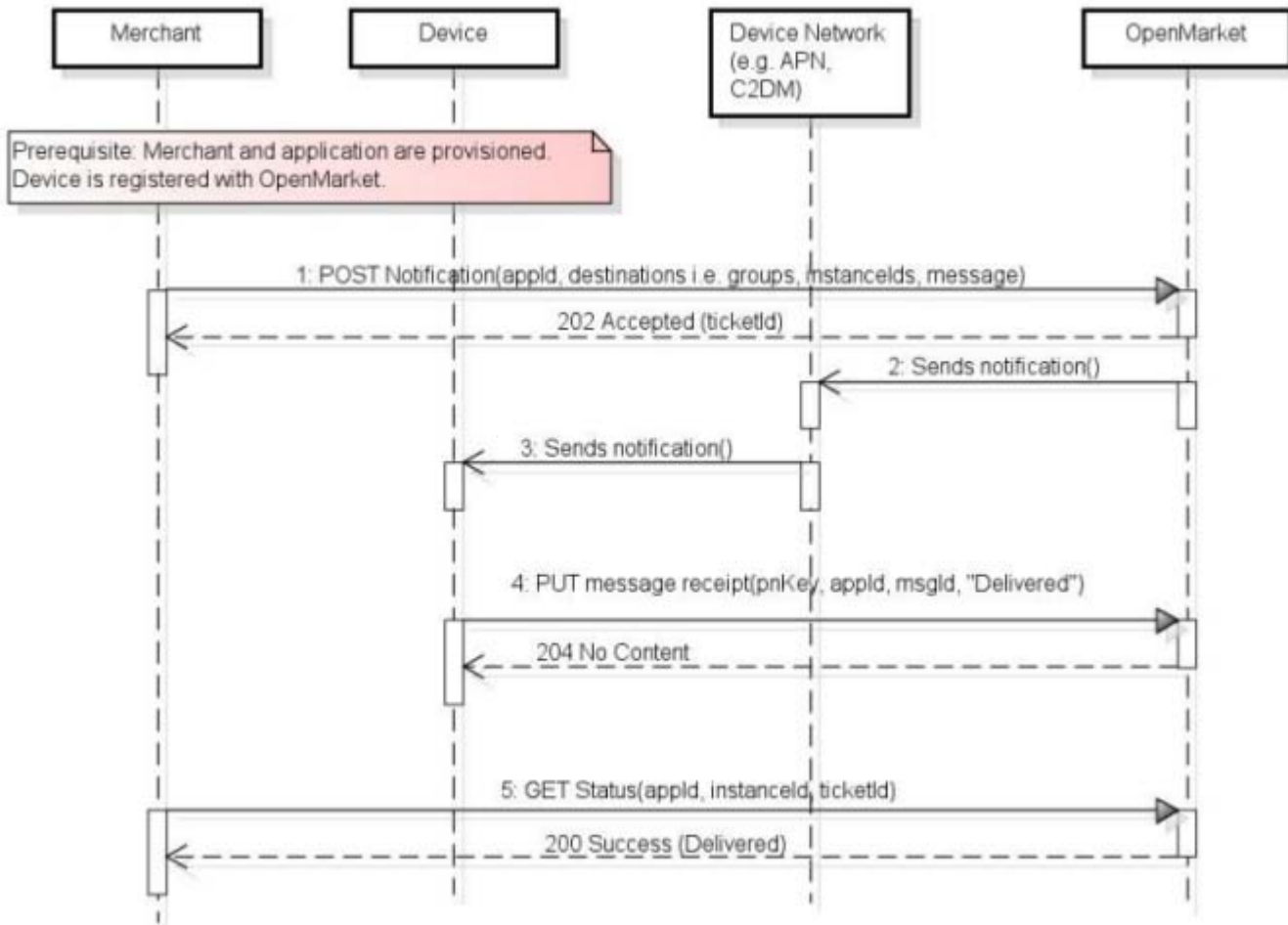
**POST Notification** on page [79](#)

**GET Notification Status for an Instance** on page [93](#)

## Creating a message receipt

The following diagram depicts the process of creating a message receipt for a push notification. The message receipt indicates that the message was delivered and it can additionally indicate whether the user interacted with the application or deleted the message.

**NOTE:** This feature is supported for C2DM and BPS.



### Steps in the process

1. Your system submits a **POST Notification** to OpenMarket Push Notifications.
2. OpenMarket returns a ticket ID and routes the notification to the device network.
3. The application on the end-user's device retrieves the message ID (MID) from the notification post and submits a **PUT Message Receipt** request to OpenMarket to set the notification status.
4. If successful, OpenMarket system responds with a ticketId.
5. Optionally, your service can call **GET Notification Status for an Instance** to get the state of the notification for an application instance.
6. OpenMarket returns the state (if reported in the PUT Message Receipt call), appId, instanceId, ticketId, createdAt, and processedDate. If the notification's state is "failed", delivered, user\_engaged, or user\_deleted, the service returns a reason for failure or any optionally entered details submitted when the PUT Message Receipt call was made.

## Connecting to the service

You access OpenMarket Push Notifications with your account ID and password credentials using an HTTPS call. Requests are verified using HTTP Basic Authentication.

You can also use your pnKey to authenticate with the service. This is the recommended method when using application self-registration.

You can send notifications to just one network (APNs, C2DM, or BPS) using their native syntax or the OpenMarket generic syntax. It is also possible to send a notification to devices in a single API call. With Apple devices it's important to remember that when using the combined call to send notifications, you use the correct URL for the appropriate APNs environment.

## Testing a push notification

APNs and BPS provide two environments for application development: production and sandbox. C2DM does not provide a separate sandbox or test environment.

OpenMarket's sandbox environment is the C2DM production environment, and the APNs and BPS sandboxes, so to test a push notification simply add "-sandbox" to the URL, as in this example:

```
https://tdc-push-  
sandbox.openmarket.com/pn/appId/DailyLuckyNumberApp/instances/ipad  
23492387234/apn
```

When testing your push notification, keep in mind that most notification networks, including the APNs, BPS (BlackBerry Push Essentials), and C2DM, do not commit to any level of availability; delivery is not guaranteed. In addition, the network environments do not provide any status feedback about whether a notification is successfully delivered to the end user's device.

### Testing tips

- ▶ If your push notification does not reach the application...
  - ▶ Check that the device is connected to the network via 3G or WiFi. All networks require that the device be connected to the data network. If you are roaming, make sure that Data Roaming is turned on.
  - ▶ Check that the target application is actually on the device.
- ▶ Make sure you have a .p12 certificate for the APN sandbox. The sandbox has its own assigned IP address and requires its own certificate.
- ▶ Verify that your C2DM credentials are correct.
- ▶ Verify that the BlackBerry application ID, password, and Base URL are correct for the evaluation (sandbox) environment.

## Reasons why a push notification can fail

Several conditions can cause a push notification to fail, including:

- ▶ OpenMarket could not authenticate your call due to a problem with your OpenMarket Push Notifications credentials.
- ▶ OpenMarket does not have the APNs security certificate data for your application.
- ▶ OpenMarket does not have the correct credentials for your account as registered with C2DM.
- ▶ OpenMarket does not have the correction information for your account and application as registered with BPS.
- ▶ Your call contains invalid parameters or uses an invalid URL.
- ▶ The destination ID (the device token, registration ID, or PIN) may have changed. This may occur when the end user migrates to a new device or reinstalls the OS on the device.

---

**NOTE:** APNs, C2DM, and BPS 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.

---

## Chapter 3. API Reference

---

### PUT New Application Instance

Use PUT New Application Instance to register a new application instance in OpenMarket. The operation returns an instanceId, which is the destination to which push notifications will be sent.

- ▶ If the destinationId in your request already exists in OpenMarket, the API returns an error.
- ▶ If the destinationId already exists but is disabled for the application, a new instance is created with a new instanceId.

---

**NOTE:** Avoid registering the same application instance multiple times, as this could result in duplicate messages being sent to the device.

---

### Request URL syntax, example, and parameters

#### Syntax

```
PUT https://{VIP}/pn/v1/applications/{appId}/instances
```

#### Example

```
PUT https://tdc-  
push.openmarket.com/pn/v1/applications/DailyLuckyNumberApp/instanc  
es
```

#### Parameters

Element	Description
appId	<p>The application name as defined by the merchant. This name must be unique per merchant and must be specified during provisioning.</p> <p><b>Required?</b> yes</p> <p><b>Type:</b> string</p> <p><b>Length:</b> 1- 25 characters</p>

	<b>Multiple allowed:</b> no
--	-----------------------------

## Request header parameters

Element	Description
Basic Authentication using: Authorization: Basic <encoded userName:password>	Used to pass the caller's userName/password credentials over an HTTPS connection.  <b>Required:</b> must use either pnAuth or Basic Authentication  <b>Type:</b> userName and password (both are case-sensitive)
pnAuth using: Authorization: pnAuth <pnKey>	A unique key provided to the merchant during provisioning for Push Notifications, and intended for use when the merchant's application self-registers.  <b>Required:</b> must use either pnAuth or Basic Authentication. Use pnAuth if your application is self-registering.  <b>Special validation:</b> the pnKey provided in the call must match the one provisioned for your OpenMarket account.

### Basic Authentication example

Authorization: Basic QWxhZGRpbjpvYVUHNlc2FtZQ==

### pnKey authentication example

Authorization: pnAuth 9FB2D4725A9B4AC01EE1E2DB579558EE725FAD4D

## Request elements and attributes

Element	Description	
<b>instance</b>	A complex element containing subelements that describe the application instance. <b>Required:</b> no	
	status	The status of the application instance. Default value is ENABLED. <b>Required:</b> no <b>Type:</b> string <b>Valid values:</b> ENABLED or DISABLED <b>Multiple allowed:</b> no
	statusdetails	Detailed information about the status of the application instance. For example, an application instance may be disabled via the APN feedback service. <b>Required:</b> no <b>Type:</b> string <b>Length:</b> 1-255 characters <b>Multiple allowed:</b> no
<b>apn</b>	Indicates that the instanceId being created or updated is for the Apple network. Child element of <b>instance</b> . <b>Required:</b> use either: <ul style="list-style-type: none"> <li>▶ c2dm for Android</li> <li>▶ apn for Apple</li> <li>▶ bps for BlackBerry</li> </ul> <b>Multiple allowed:</b> no	
	deviceToken	An identifier returned by APN to the application instance when the end user registers the app for push notifications. This identifier is used as a type of destination ID. <b>Required?</b> yes <b>Type:</b> string <b>Length:</b> 1-64 characters <b>Multiple allowed?</b> No

Element	Description	
<b>c2dm</b>	Indicates that the instanceId being created or updated is on the Android network. <b>Required:</b> use one of: <ul style="list-style-type: none"> <li>▶ c2dm (for Android)</li> <li>▶ apn (for Apple)</li> <li>▶ bps (for BlackBerry)</li> </ul> <b>Multiple allowed:</b> no	
	registrationId	An identifier returned by C2DM to the application instance when the end user registers the app for push notifications. This identifier is used as a type of destination ID.  <b>Type:</b> string <b>Required:</b> yes <b>Length:</b> 1-512 characters <b>Multiple allowed:</b> no
<b>bps</b>	Indicates that the instanceId that's created or updated is for the BlackBerry network. <b>Type:</b> element <b>Required:</b> use one of: <ul style="list-style-type: none"> <li>▶ c2dm (for Android)</li> <li>▶ apn (for Apple)</li> <li>▶ bps (for BlackBerry)</li> </ul> <b>Multiple allowed:</b> no	
	pin	A unique ID for a specific BlackBerry device. This ID is used as a type of destination ID.  <b>Type:</b> string <b>Required:</b> yes <b>Length:</b> 8 characters <b>Multiple allowed:</b> no
<b>groups</b>	Complex element containing the groups to which the application instance belongs. <b>Required:</b> no <b>Multiple allowed:</b> no	
	group	The group to which the application instance belongs. An instance

Element	Description	
		can be in more than one group. <b>Type:</b> string <b>Required:</b> yes <b>Multiple allowed:</b> yes
	id	The ID of the group to which the application instance belongs (not case-sensitive). <b>Type:</b> string <b>Required:</b> yes

## XML schemas

[http://tdc-push.openmarket.com/pn/xsd/com/openmarket/pn/v1/instance\\_V1.0.xsd](http://tdc-push.openmarket.com/pn/xsd/com/openmarket/pn/v1/instance_V1.0.xsd)

[http://tdc-push.openmarket.com/pn/xsd/com/openmarket/pn/v1/instanceType\\_V1.0.xsd](http://tdc-push.openmarket.com/pn/xsd/com/openmarket/pn/v1/instanceType_V1.0.xsd)

[http://tdc-push.openmarket.com/pn/xsd/com/openmarket/pn/v1/error\\_V1.0.xsd](http://tdc-push.openmarket.com/pn/xsd/com/openmarket/pn/v1/error_V1.0.xsd)

## Example request body

```
<?xml version="1.0" encoding="UTF-8"?>
<instance xmlns="http://pn.openmarket.com/v1"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
    status="ENABLED">
    <apn
deviceToken="FE66489F304DC75B8D6E8200DFF8A456E8DAEACEC428B427E9518741C9
2C6660"/>
</instance>
```

```
<?xml version="1.0" encoding="UTF-8"?>
<instance xmlns="http://pn.openmarket.com/v1"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
    status="ENABLED">
    <c2dm
registrationId="APA91bG0c6eAYjoE7xyqlbf8sbDaH1Kw_bQced9J0tJyYIE5tP3oSSq
s3_3I4R8oggLrsxXkAMR3dkOmYS6dxXPVpl_dKNLBOKq15J3GfolDaRXHDCmZ"/>
    <groups>
        <group id="Soccer"/>
        <group id="Tennis"/>
    </groups>
</instance>
```

```
<?xml version="1.0" encoding="UTF-8"?>
<instance xmlns="http://pn.openmarket.com/v1"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" status="ENABLED">
    <bps pin="22dcd556"/>
    <groups>
        <group id="Seattle"/>
    </groups>
</instance>
```

## Response header parameters

Element	Description
X-Ticket-Id	The ticket ID. <b>Required?</b> yes <b>Type:</b> string
X-Instance-Id	Openmarket-generated ID of the application instance that was created. This ID is the reference used by OpenMarket to represent the instance of the application on the end user's device. <b>Required?</b> yes <b>Type:</b> string
Location	The URI of the newly created resource. <b>Required?</b> yes <b>Type:</b> string

### Example

X-Ticket-Id: 7EYYG-EF3XM-5LSQ5-ILR73

X-Instance-Id: IILR735LSQ5EF3XM7EYYG

Location: https://tdc-  
push.openmarket.com/pn/v1/applications/DailyLuckyNumberApp/instances/IILR735LSQ5EF3XM7EYYG

## Response elements and attributes

Element / Attribute	Description
instance (element)	A complex element containing subelements that describe the application instance. <b>Required?</b> yes
id (attribute of instance)	The ID of the application instance. <b>Required:</b> no <b>Type:</b> string

Element / Attribute	Description
status (attribute of instance)	<p>The status of the application instance. Default value is ENABLED.</p> <p><b>Required:</b> yes</p> <p><b>Type:</b> string</p> <p><b>Valid values:</b> ENABLED or DISABLED</p>
statusDetails (attribute of instance)	<p>Detailed information about the status of the application instance. For example, an application instance may be disabled via the APN feedback service.</p> <p><b>Required:</b> no</p> <p><b>Type:</b> string</p> <p><b>Length:</b> 1-255 characters</p>
apn (element of instance)	<p>Indicates that the instanceId is for the Apple network.</p> <p><b>Required:</b> use either apn (for Apple) or c2dm (for Android)</p> <p><b>Multiple allowed:</b> no</p>
deviceToken (attribute of apn)	<p>An identifier returned by APN to the application instance when the end user registers the app for push notifications. This identifier is used as a type of destination ID.</p> <p><b>Required?</b> yes</p> <p><b>Type:</b> string</p> <p><b>Length:</b> 1-64 characters</p> <p><b>Multiple allowed?</b> No</p>
c2dm (element of instance)	<p>Indicates that the instanceId is for the C2DM network.</p> <p><b>Required:</b> use either c2dm (for Android) or apn (for Apple)</p> <p><b>Multiple allowed:</b> no</p>
registrationId (attribute of c2dm)	<p>An identifier returned by C2DM to the application instance when the end user registers the app for push notifications. This identifier is used as a type of destination ID.</p> <p><b>Type:</b> string</p> <p><b>Required:</b> yes</p> <p><b>Length:</b> 1-512 characters</p>

Element / Attribute	Description
groups (element of instance)	<p>Complex element containing the groups to which the application instance belongs.</p> <p><b>Required:</b> no</p> <p><b>Multiple allowed:</b> no</p>
bps	<p>Indicates that the instanceId that's created or updated is for the BlackBerry network.</p> <p><b>Type:</b> element</p> <p><b>Required:</b> use one of:</p> <ul style="list-style-type: none"> <li>▶ c2dm (for Android)</li> <li>▶ apn (for Apple)</li> <li>▶ bps (for BlackBerry)</li> </ul> <p><b>Multiple allowed:</b> no</p>
pin	<p>A unique ID for a specific BlackBerry device. This ID is used as a type of destination ID.</p> <p><b>Type:</b> string</p> <p><b>Required:</b> yes</p> <p><b>Length:</b> 8 characters</p> <p><b>Multiple allowed:</b> no</p>
group (element of groups)	<p>The group to which the application instance belongs. An instance can be in more than one group.</p> <p><b>Required:</b> yes</p> <p><b>Multiple allowed:</b> yes</p>
id (attribute of group)	<p>The ID of the group to which the application instance belongs (not case-sensitive).</p> <p><b>Type:</b> string</p> <p><b>Required:</b> yes</p> <p><b>Length:</b> 1-50</p> <p><b>Multiple allowed:</b> no</p>

## Example response body

```
<?xml version="1.0" encoding="UTF-8"?>
<instance xmlns="http://pn.openmarket.com/v1"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  id="IILR735LSQ5EF3XM7EYYG" status="ENABLED">
  <apn
deviceToken="FE66489F304DC75B8D6E8200DFF8A456E8DAEACEC428B427E9518741C9
2C6660"/>
</instance>
```

```
<?xml version="1.0" encoding="UTF-8"?>
<instance xmlns="http://pn.openmarket.com/v1"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  id="IILR735LSQ5EF3XM7EYYG" status="ENABLED">
  <c2dm
registrationId="APA91bG0c6eAYjoE7xyqlbf8sbDaH1Kw_bQced9J0tJyYIE5tP3oSSq
s3_3I4R8oggLrsxXkAMR3dkOmYS6dxXPVpl_dKNLBOKq15J3GfolDaRXHDCmZ"/>
  <groups>
    <group id="Soccer"/>
    <group id="Tennis"/>
  </groups>
</instance>
```

```
<?xml version="1.0" encoding="UTF-8"?>
<instance xmlns="http://pn.openmarket.com/v1"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  id="IILR735LSQ5EF3XM7EYYG" status="ENABLED">
  <bps pin="22dcd556"/>
  <groups>
    <group id="Seattle"/>
  </groups>
</instance>
```

## Error conditions

The errors you may encounter are usually caused by these conditions:

- ▶ There's a problem with your OpenMarket account that's impeding authentication, such as:
  - ▶ The account ID or password you are using is invalid
  - ▶ Your account isn't enabled for Push Notifications
  - ▶ Your pnKey is invalid

- ▶ This application ID does not exist in OpenMarket, perhaps because the application is not provisioned.
- ▶ The application ID is disabled.
- ▶ Your application is sending information that does not validate. Make sure the XML request body conforms to the schema.
- ▶ Your application is using an invalid URL, has specified the incorrect version of the API, or the URL contains an invalid parameter.
- ▶ The destination ID (APN device token, C2DM registration ID, or BPS PIN) is invalid; e.g., it exceeds the maximum length.
- ▶ The destination ID already exists for the application.

For details about all error and response codes, log on to Customer Center, select the Resources menu, and view the Response Codes.

## PUT Updated Application Instance

Use PUT Updated Application Instance to replace the existing instance information with updated application instance information in OpenMarket Push Notifications. Calling this operation replaces the device token (APN) or registration ID (C2DM) for the instance ID, the status information, and the group information. As an alternative to updating the whole application instance, you can use one of the convenience APIs to:

- ▶ Update the APN device token, the C2DM registration ID, or the BPS PIN for the application instance
- ▶ Add one or more groups to an application instance
- ▶ Delete one or more groups from an application instance

Use [GET Application Instance](#) on page 60 to get the current information about the application instance. Once you have that information, use PUT Updated Application Instance to submit the update.

## Request URL syntax, example, and parameters

### Syntax

```
PUT https://{VIP}/pn/v1/applications/{appId}/instances/{instanceId}
```

### Example

```
PUT https://tdc-
push.openmarket.com/pn/v1/applications/DailyLuckyNumberApp/instanc
es/IILR735LSQ5EF3XM7EYYG
```

### Parameters

Element	Description
appId	<p>The application name as defined by the merchant. This name must be unique per merchant and must be specified during provisioning.</p> <p><b>Required?</b> yes</p> <p><b>Type:</b> string</p> <p><b>Length:</b> 1- 25 characters</p> <p><b>Multiple allowed:</b> no</p>
instanceId (attribute of status)	<p>An identifier for an instance of an application on a mobile device. OpenMarket returns this ID when the instance is created or replaced.</p> <p><b>Required?</b> yes</p> <p><b>Type:</b> string</p> <p><b>Length:</b> 1-24</p> <p><b>Multiple allowed:</b> no</p>

## Request header parameters

Element	Description
Basic Authentication using: Authorization: Basic <encoded userName:password>	<p>Used to pass the caller's userName/password credentials over an HTTPS connection.</p> <p><b>Required:</b> must use either pnAuth or Basic Authentication</p> <p><b>Type:</b> userName and password (both are case-sensitive)</p>

<p>pnAuth using: Authorization: pnAuth &lt;pnKey&gt;</p>	<p>A unique key provided to the merchant during provisioning for Push Notifications, and intended for use when the merchant's application self-registers.</p> <p><b>Required:</b> must use either pnAuth or Basic Authentication. Use pnAuth if your application is self-registering.</p> <p><b>Special validation:</b> the pnKey provided in the call must match the one provisioned for your OpenMarket account.</p>
--	--

### Example Basic Authentication

Authorization: Basic QWxhZGRpbjpvYVUHNlc2FtZQ==

### Example pnKey authentication

Authorization: pnAuth 9FB2D4725A9B4AC01EE1E2DB579558EE725FAD4D

## Request elements and attributes

Element / Attribute	Description
instance (element)	<p>A complex element containing subelements that describe the application instance.</p> <p><b>Required?</b> yes</p>
status (attribute of instance)	<p>The status of the application instance. Default value is ENABLED.</p> <p><b>Required?</b> yes</p> <p><b>Type:</b> element</p> <p><b>Values:</b> ENABLED or DISABLED</p> <p><b>Multiple allowed:</b> no</p>
statusDetails (attribute of instance)	<p>Detailed information about the status of the application instance. For example, an application instance may be disabled via the APN feedback service.</p> <p><b>Required?</b> no</p> <p><b>Length:</b> 1-255 characters</p> <p><b>Multiple allowed:</b> no</p>

Element / Attribute	Description
apn (element of instance)	<p>Indicates that the instanceId is for the Apple network.</p> <p><b>Required:</b> use one of:</p> <ul style="list-style-type: none"> <li>▶ c2dm (for Android)</li> <li>▶ apn (for Apple)</li> <li>▶ bps (for BlackBerry)</li> </ul> <p><b>Multiple allowed:</b> no</p>
deviceToken (attribute of apn)	<p>An identifier returned by APN to the application instance when the end user registers the app for push notifications. This identifier is used as a type of destination ID.</p> <p>Required? yes</p> <p><b>Length:</b> 1-64 characters</p> <p><b>Multiple allowed:</b> no</p>
c2dm (element of instance)	<p>Indicates that the instanceId being created or updated is on the Android network.</p> <p><b>Required?</b> use either c2dm (for Android) or apn (for Apple)</p> <p><b>Multiple allowed:</b> no</p>
registrationId (attribute of c2dm)	<p>An identifier returned by C2DM to the application instance when the end user registers the app for push notifications. This identifier is used as a type of destination ID.</p> <p><b>Required?</b> yes</p> <p><b>Length:</b> 1-512 characters</p> <p><b>Multiple allowed:</b> no</p>
bps	<p>Indicates that the instanceId that's created or updated is for the BlackBerry network.</p> <p><b>Type:</b> element</p> <p><b>Required:</b> use one of:</p> <ul style="list-style-type: none"> <li>▶ c2dm (for Android)</li> <li>▶ apn (for Apple)</li> <li>▶ bps (for BlackBerry)</li> </ul> <p><b>Multiple allowed:</b> no</p>

Element / Attribute	Description
pin	<p>A unique ID for a specific BlackBerry device. This ID is used as a type of destination ID.</p> <p><b>Type:</b> string</p> <p><b>Required:</b> yes</p> <p><b>Length:</b> 8 characters</p> <p><b>Multiple allowed:</b> no</p>
groups (element of instance)	<p>Complex element containing the groups to which the application instance belongs.</p> <p><b>Required?</b> yes</p> <p><b>Type:</b> element</p> <p><b>Multiple allowed:</b> no</p>
group (element of groups)	<p>The group to which the application instance belongs. An instance can be in more than one group.</p> <p><b>Required?</b> yes</p> <p><b>Multiple allowed:</b> yes</p>
id (attribute of group)	<p>The ID of the group to which the application instance belongs (not case-sensitive).</p> <p><b>Required?</b> yes</p> <p><b>Length:</b> 1-50</p> <p><b>Multiple allowed:</b> no</p>

## XML schemas

[http://tdc-push.openmarket.com/pn/xsd/com/openmarket/pn/v1/instance\\_V1.0.xsd](http://tdc-push.openmarket.com/pn/xsd/com/openmarket/pn/v1/instance_V1.0.xsd)

[http://tdc-push.openmarket.com/pn/xsd/com/openmarket/pn/v1/instanceType\\_V1.0.xsd](http://tdc-push.openmarket.com/pn/xsd/com/openmarket/pn/v1/instanceType_V1.0.xsd)

[http://tdc-push.openmarket.com/pn/xsd/com/openmarket/pn/v1/error\\_V1.0.xsd](http://tdc-push.openmarket.com/pn/xsd/com/openmarket/pn/v1/error_V1.0.xsd)

## Example request body

```
<?xml version="1.0" encoding="UTF-8"?>
<instance xmlns="http://pn.openmarket.com/v1"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  id="IILR735LSQ5EF3XM7EYYG" status="ENABLED">
  <apn
deviceToken="FE66489F304DC75B8D6E8200DFF8A456E8DAEACEC428B427E9518741C9
2C6660"/>
</instance>
```

```
<?xml version="1.0" encoding="UTF-8"?>
<instance xmlns="http://pn.openmarket.com/v1"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  id="IILR735LSQ5EF3XM7EYYG" status="ENABLED">
  <c2dm
registrationId="APA91bG0c6eAYjoE7xyqlbf8sbDaH1Kw_bQced9J0tJyYIE5tP3oSSq
s3_3I4R8oggLrsxXkAMR3dkOmYS6dxXPVpl_dKNLBOKq15J3GfolDaRXHDCmZ"/>
  <groups>
    <group id="Soccer"/>
    <group id="Tennis"/>
  </groups>
</instance>
```

```
<?xml version="1.0" encoding="UTF-8"?>
<instance xmlns="http://pn.openmarket.com/v1"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  id="IILR735LSQ5EF3XM7EYYG" status="ENABLED">
  <bps
pin="22dcd556"/>
  <groups>
    <group id="Seattle"/>
  </groups>
</instance>
```

## Response header parameters

Element	Description
X-Ticket-Id	The ticket ID. <b>Required?</b> yes <b>Type:</b> string
X-Instance-Id	The ID of the instance that was updated. <b>Required?</b> yes <b>Type:</b> string
Location	The URI of the newly updated resource. <b>Required?</b> yes <b>Type:</b> string

### Example

X-Ticket-Id: 7EYYG-EF3XM-5LSQ5-ILR73

X-Instance-Id: IILR735LSQ5EF3XM7EYYG

Location: https://tdc-  
push.openmarket.com/pn/v1/applications/DailyLuckyNumberApp/instances/IILR735LSQ5EF3XM7EYYG

## Response elements and attributes

Element / Attribute	Description
instance (complex element)	A complex element containing subelements that describe the application instance. <b>Required?</b> yes <b>Type:</b> element
id (attribute of instance)	The ID of the application instance. <b>Required?</b> no <b>Type:</b> string

Element / Attribute	Description
status (attribute of instance)	<p>The status of the application instance. Default value is ENABLED.</p> <p><b>Required:</b> yes</p> <p><b>Type:</b> string</p> <p><b>Valid values:</b> ENABLED or DISABLED</p> <p><b>Multiple allowed:</b> no</p>
statusDetails (attribute of instance)	<p>Detailed information about the status of the application instance. For example, an application instance may be disabled via the APN feedback service.</p> <p><b>Required:</b> no</p> <p><b>Type:</b> string</p> <p><b>Length:</b> 1-255 characters</p>
apn (element of instance)	<p>Indicates that the instanceId is for the Apple network.</p> <p><b>Required:</b> use either apn (for Apple) or c2dm (for Android)</p> <p><b>Multiple allowed:</b> no</p>
deviceToken (attribute of apn)	<p>An identifier returned by APN to the application instance when the end user registers the app for push notifications. This identifier is used as a type of destination ID.</p> <p><b>Required?</b> yes</p> <p><b>Type:</b> string</p> <p><b>Length:</b> 1-64 characters</p> <p><b>Multiple allowed?</b> No</p>
c2dm (element of instance)	<p>Indicates that the instanceId being created or updated is on the Android network.</p> <p><b>Required:</b> use either c2dm (for Android) or apn (for Apple)</p> <p><b>Multiple allowed:</b> no</p>

Element / Attribute	Description
registrationId (attribute of c2dm)	<p>An identifier returned by C2DM to the application instance when the end user registers the app for push notifications. This identifier is used as a type of destination ID.</p> <p><b>Type:</b> string</p> <p><b>Required:</b> yes</p> <p><b>Length:</b> 1-512 characters</p>
bps	<p>Indicates that the instanceId that's created or updated is for the BlackBerry network.</p> <p><b>Type:</b> element</p> <p><b>Required:</b> use one of:</p> <ul style="list-style-type: none"> <li>▶ c2dm (for Android)</li> <li>▶ apn (for Apple)</li> <li>▶ bps (for BlackBerry)</li> </ul> <p><b>Multiple allowed:</b> no</p>
pin	<p>A unique ID for a specific BlackBerry device. This ID is used as a type of destination ID.</p> <p><b>Type:</b> string</p> <p><b>Required:</b> yes</p> <p><b>Length:</b> 8 characters</p> <p><b>Multiple allowed:</b> no</p>
groups (element of instance)	<p>Complex element containing the groups to which the application instance belongs.</p> <p><b>Required:</b> no</p> <p><b>Multiple allowed:</b> no</p>
group (element of groups)	<p>The group to which the application instance belongs. An instance can be in more than one group.</p> <p><b>Required:</b> yes</p> <p><b>Multiple allowed:</b> yes</p>

Element / Attribute	Description
id (attribute of group)	The ID of the group to which the application instance belongs (not case-sensitive). <b>Type:</b> string <b>Required:</b> yes <b>Length:</b> 1-50 <b>Multiple allowed:</b> no

## Example response body

```
<?xml version="1.0" encoding="UTF-8"?>
<instance xmlns="http://pn.openmarket.com/v1"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  id="IILR735LSQ5EF3XM7EYYG" status="ENABLED">
  <apn
deviceToken="FE66489F304DC75B8D6E8200DFF8A456E8DAEACEC428B427E9518741C9
2C6660"/>
</instance>
```

```
<?xml version="1.0" encoding="UTF-8"?>
<instance xmlns="http://pn.openmarket.com/v1"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  id="IILR735LSQ5EF3XM7EYYG" status="ENABLED">
  <c2dm
registrationId="APA91bG0c6eAYjoE7xyqlbf8sbDaH1Kw_bQced9J0tJyYIE5tP3oSSq
s3_3I4R8oggLrsxXkAMR3dkOmYS6dxXPVpl_dKNLBOKq15J3GfolDaRXHDCmZ"/>
  <groups>
    <group id="Soccer"/>
    <group id="Tennis"/>
  </groups>
</instance>
```

```
<?xml version="1.0" encoding="UTF-8"?>
<instance xmlns="http://pn.openmarket.com/v1"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  id="IILR735LSQ5EF3XM7EYYG" status="ENABLED">
  <bps
pin="22dcd556"/>
  <groups>
    <group id="Seattle"/>
  </groups>
</instance>
```

## Error conditions

The following conditions will result in HTTP errors:

- ▶ There's a problem with your OpenMarket account that's impeding authentication, such as:
  - ▶ The account ID or password you are using is invalid
  - ▶ Your account isn't enabled for Push Notifications
  - ▶ Your pnKey is invalid
- ▶ This applicationId does not exist in OpenMarket, perhaps because the application is not provisioned.

- ▶ The applicationId is disabled.
- ▶ Your application is sending information that does not validate. Make sure the XML request body conforms to the schema.
- ▶ Your application is using an invalid URL, has specified an invalid API version, or the URL contains an invalid parameter.
- ▶ The destination ID (APN device token, C2DM registration ID, or BPS PIN) is invalid; e.g., it exceeds the maximum length.
- ▶ The instanceId you are trying to update does not exist in the system.

For details about all error and response codes, log on to Customer Center, select the Resources menu, and view the Response Codes.

## GET Application Instances information (server-side)

Use the server-side GET Application Instances operation to retrieve all the instance IDs for your application. Using this interface you can:

- ▶ Retrieve just a list of instance IDs or a list plus detailed information such as group, status, and destination ID
- ▶ Query for a particular platform (e.g., device token or registration ID)
- ▶ Query for a specific group
- ▶ Query for a status (e.g., enabled)
- ▶ Paginate your result set

The service responds with an HTTP 200 message for all successful calls. OpenMarket returns an error if the instance ID does not exist.

## Request URL syntax, example, and parameters

### Syntax

```
GET
https://{host:port}/pn/v1/applications/{appId}/instances/?{param}=
{value}
```

### Examples

This call returns instances for two groups called "hockey" and "football".

```
GET https://tdc-
pn.openmarket.com/pn/v1/applications/weatherApp/instances?group=hockey&group=football
```

This call returns just the one instance for a weather application.

```
GET https://tdc-
pn.openmarket.com/pn/v1/applications/weatherAPP/instances/?instanceId=IILR735LSQ5EF3XM7EYYG
```

This call returns the instance with the specified PIN for a weather application.

```
GET https://tdc-
pn.openmarket.com/pn/v1/applications/weatherAPP/instances?pin=22dcd556
```

This call returns a list of IDs for the first 2000 active instances. Group, status, and destinationId information is not included in this call.

```
GET https://tdc-
pn.openmarket.com/pn/v1/applications/weatherAPP/instances?limit=2000&offset=0&status=enabled&out=ids
```

This call returns a list of IDs of the next 2000 active instances for a weather application. Group, status, and destinationId information is not included in this call.

```
GET https://tdc-
pn.openmarket.com/pn/v1/applications/weatherAPP/instances?limit=2000&offset=2000&status=enabled&out=ids
```

This call returns the default maximum limit of instances and all related data, including groups, for a weather application. The default limit is 1000 instances.

```
GET https://tdc-
pn.openmarket.com/pn/v1/applications/weatherAPP/instances/
```

### Parameters

Element	Description
appId	<p>The application name as defined by the merchant. This name must be unique per merchant and must be specified during provisioning.</p> <p><b>Required?</b> yes</p> <p><b>Type:</b> string</p> <p><b>Length:</b> 1- 25 characters</p> <p><b>Multiple allowed:</b> no</p>
groupId	<p>The name of the group, for example "baseball" or "teenagers".</p> <p>Include the group ID(s) to include in the query.</p> <p><b>Required?</b> no</p> <p><b>Type:</b> string</p> <p><b>Length:</b> 1- 50 characters</p> <p><b>Multiple allowed:</b> yes</p>
instanceId	<p>An identifier for an instance of an application on a mobile device.</p> <p>Include the instance IDs to include in the query.</p> <p><b>Required?</b> no</p> <p><b>Type:</b> string</p> <p><b>Length:</b> 1-24</p> <p><b>Multiple allowed:</b> yes</p>
deviceToken	<p>An identifier returned by APN to the application instance when the end user registers the app for push notifications. This identifier is used as a type of destination ID.</p> <p>Include the device tokens to include in the query.</p> <p><b>Required?</b> no</p> <p><b>Type:</b> string</p> <p><b>Length:</b> 1-64 characters</p> <p><b>Multiple allowed?</b> yes</p>

Element	Description
pin	<p>A unique ID for a specific BlackBerry device. This ID is used as a type of destination ID.</p> <p>Include the pin(s) to include in the query.</p> <p><b>Required?</b> no</p> <p><b>Type:</b> string</p> <p><b>Length:</b> 1-8</p> <p><b>Multiple allowed:</b> yes</p>
registrationId	<p>An identifier returned by C2DM to the application instance when the end user registers the app for push notifications. This identifier is used as a type of destination ID.</p> <p>Include the registration ID(s) to include in the query.</p> <p><b>Required?</b> no</p> <p><b>Type:</b> string</p> <p><b>Length:</b> 1-512</p> <p><b>Multiple allowed:</b> yes</p>
statusFilter	<p>The status (either ENABLED or DISABLED) on which to filter the results.</p> <p><b>Required?</b> no</p> <p><b>Type:</b> string</p> <p><b>Multiple allowed:</b> no</p>
limit	<p>The numeric limit of the number of instances to return. Supports pagination.</p> <p>When retrieving just IDs (by using the out=id parameter), the limit default and maximum are 5000. When retrieving IDs and the additional information, the limit default and maximum are 1000.</p> <p><b>Required?</b> no</p> <p><b>Type:</b> integer</p> <p><b>Multiple allowed:</b> no</p>

Element	Description
offset	<p>The numeric offset from the previous result set from which to start. Supports pagination.</p> <p>For example, if you make two consecutive calls, the first call could specify offset=0 and limit=2000 to query for the first 2000 active instanceIds. The subsequent call would specify offset=2000 and limit=2000 to query for the next 2000 instances.</p> <p><b>Required?</b> no  <b>Type:</b> integer  <b>Multiple allowed:</b> no</p>
out	<p><b>out=id</b> specifies that only a list of instance IDs will be returned.</p> <p><b>Required?</b> no  <b>Type:</b> string  <b>Multiple allowed:</b> no</p>

## Request header parameters

Element	Description
Basic Authentication using: Authorization: Basic <encoded userName:password>	<p>Used to pass the caller's userName/password credentials over an HTTPS connection.</p> <p><b>Required:</b> must use either pnAuth or Basic Authentication  <b>Type:</b> userName and password (both are case-sensitive)</p>

### Example

```
Authorization: Basic QWxhZGRpbjpvceGVuIHNLc2FtZQ==
```

## XML schemas for GET Application Instance

`http://tdc-push.openmarket.com/pn/xsd/com/openmarket/pn/v1/instances_V1.0.xsd`

`http://tdc-push.openmarket.com/pn/xsd/com/openmarket/pn/v1/instanceType_V1.0.xsd`

`http://tdc-push.openmarket.com/pn/xsd/com/openmarket/pn/v1/error_V1.0.xsd`

## Response header parameters

Element	Description
X-Ticket-Id	<p>The ticket ID.</p> <p><b>Required?</b> yes</p> <p><b>Type:</b> string</p>

### Example

`X-Ticket-Id:7EYYG-EF3XM-5LSQ5-ILR73`

## Response elements and attributes

Element / Attribute	Description
instances	<p>Element containing all the retrieved instances.</p> <p><b>Required?</b> yes</p> <p><b>Type:</b> element</p> <p><b>Multiple allowed:</b> no</p>
instance	<p>The instance element.</p> <p><b>Required?</b> yes</p> <p><b>Type:</b> element</p> <p><b>Multiple allowed:</b></p>

Element / Attribute	Description
id (attribute of instance)	The ID of the application instance to which the notification is pushed. <b>Required?</b> yes <b>Type:</b> string <b>Multiple allowed:</b> no
status (attribute of instance)	Status of the application instance (ENABLED or DISABLED). <b>Required?</b> yes <b>Type:</b> string <b>Multiple allowed:</b> no
statusDetails (attribute of instance)	Details related to the status of the instance. <b>Required?</b> no <b>Type:</b> string <b>Length:</b> 1-255 characters <b>Multiple allowed:</b> no
apn (element of instance)	Indicates that the instanceId is for the Apple network. <b>Required?</b> must be one of the following: <ul style="list-style-type: none"> <li>▶ apn</li> <li>▶ c2dm</li> <li>▶ bps</li> </ul> <b>Type:</b> element <b>Multiple allowed:</b> no
deviceToken (attribute of apn)	A hexadecimal string representing the ID that is returned to the application instance when it registers for push notifications with APN. <b>Required?</b> no <b>Type:</b> string <b>Length:</b> 1-64 characters <b>Multiple allowed:</b> no

Element / Attribute	Description
c2dm (element of instance)	<p>Indicates that the instance ID is for the Android network.</p> <p><b>Required?</b> must be one of the following:</p> <ul style="list-style-type: none"> <li>▶ apn</li> <li>▶ c2dm</li> <li>▶ bps</li> </ul> <p><b>Type:</b> element</p> <p><b>Multiple allowed:</b> no</p>
registrationId (attribute of c2dm)	<p>An identifier returned by C2DM to the application instance when the end user registers the app for push notifications. This identifier is used as a type of destination ID.</p> <p><b>Required?</b> no</p> <p><b>Type:</b> string</p> <p><b>Length:</b> 1-512 characters</p> <p><b>Multiple allowed:</b> no</p>
bps (element of instance)	<p>Indicates that the instance ID is for the BlackBerry network.</p> <p><b>Required?</b> must be one of the following:</p> <ul style="list-style-type: none"> <li>▶ apn</li> <li>▶ c2dm</li> <li>▶ bps</li> </ul> <p><b>Type:</b> element</p> <p><b>Multiple allowed:</b> no</p>
pin (attribute of bps)	<p>A unique ID for a specific BlackBerry device. This ID is used as a type of destination ID.</p> <p><b>Required?</b> no</p> <p><b>Type:</b> string</p> <p><b>Length:</b> 8 characters</p> <p><b>Multiple allowed:</b> no</p>

Element / Attribute	Description
groups (element of instance)	The groups to which the instance belongs. <b>Required?</b> no <b>Type:</b> element <b>Multiple allowed:</b> no
group (element of groups)	The group to which the instance belongs. <b>Required?</b> no <b>Type:</b> element <b>Multiple allowed:</b> yes
id (attribute of group)	The ID of the group to which the instance belongs. <b>Required?</b> no <b>Type:</b> string <b>Maximum length:</b> 1-50 characters <b>Multiple allowed:</b> no

## Example response body

### Example containing only a list of instance IDs

```
<?xml version="1.0" encoding="UTF-8"?>
<instances xmlns="http://pn.openmarket.com/v1"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  <instance id="IILR735LSQ5EF3XM7EYYG"/>
  <instance id="BKLJ735LSQ5EF3XM7EYYG"/>
  <instance id="ZKHY735LSQ5EF3XM7EXXG"/>
  <instance id="LKYJ735LSQ5EF3XM7EYYG"/>
</instances>
```

### Example containing detailed information about the instance IDs

```
<?xml version="1.0" encoding="UTF-8"?>
<instances xmlns="http://pn.openmarket.com/v1"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  <instance id="IILR735LSQ5EF3XM7EYYG" status="DISABLED"
statusDetails="APN Feedback">
  <apn
deviceToken="FE66489F304DC75B8D6E8200DFF8A456E8DAEACEC428B427E9518741C9
2C6660"/>
  </instance>
  <instance id="BKLJ735LSQ5EF3XM7EYYG" status="ENABLED">
  <c2dm
registrationId="APA91bG0c6eAYjoE7xyqlbf8sbDaH1Kw_bQced9J0tJyYIE5tP3oSSq
s3_3I4R8oggLrsxXkAMR3dkOmYS6dxXPVpl_dKNLBOKq15J3GfolDaRXHDCmZ"/>
  <groups>
    <group id="Soccer"/>
    <group id="Tennis"/>
  </groups>
  </instance>
  <instance id="ZKHY735LSQ5EF3XM7EXXG" status="ENABLED">
  <bps pin="22dcd556"/>
  <groups>
    <group id="Seattle"/>
  </groups>
  </instance>
</instances>
```

## Error conditions

The following conditions will result in HTTP 40x errors:

- ▶ There's a problem with your OpenMarket account that's impeding authentication, such as:
  - ▶ The account ID or password you are using is invalid
  - ▶ Your account isn't enabled for Push Notifications
- ▶ This applicationId does not exist in OpenMarket, perhaps because the application is not provisioned.
- ▶ The application is disabled on the target device.
- ▶ Your application is using an invalid URL, has specified the wrong version of the API, or the URL contains an invalid parameter.

For details about all error and response codes, log on to Customer Center, select the Resources menu, and view the Response Codes.

## GET Application Instance information (device-side)

Use the GET Application Instance operation to retrieve the current application instance. OpenMarket returns XML that you can update and then use in the request body for **PUT Updated Application Instance** on page 39.

The service responds with a HTTP 200 message for all successful calls. OpenMarket returns an error if the instance ID does not exist.

### XML schema

[http://tdc-push.openmarket.com/pn/xsd/com/openmarket/pn/v1/instance\\_V1.0.xsd](http://tdc-push.openmarket.com/pn/xsd/com/openmarket/pn/v1/instance_V1.0.xsd)

[http://tdc-push.openmarket.com/pn/xsd/com/openmarket/pn/v1/instanceType\\_V1.0.xsd](http://tdc-push.openmarket.com/pn/xsd/com/openmarket/pn/v1/instanceType_V1.0.xsd)

[http://tdc-push.openmarket.com/pn/xsd/com/openmarket/pn/v1/error\\_V1.0.xsd](http://tdc-push.openmarket.com/pn/xsd/com/openmarket/pn/v1/error_V1.0.xsd)

## Request URL syntax, example, and parameters

### Syntax

```
GET https://{VIP}/pn/v1/applications/{appId}/instances/{instanceId}
```

### Example

```
GET https://tdc-
push.openmarket.com/pn/v1/applications/DailyLuckyNumberApp/instanc
es/IILR735LSQ5EF3XM7EYYG
```

### Resource parameters

Element	Description
appId	<p>The application name as defined by the merchant. This name must be unique per merchant and must be specified during provisioning.</p> <p><b>Required?</b> yes</p> <p><b>Type:</b> string</p> <p><b>Length:</b> 1- 25 characters</p> <p><b>Multiple allowed:</b> no</p>
instanceId (attribute of status)	<p>An identifier for an instance of an application on a mobile device. OpenMarket returns this ID when the instance is created or replaced.</p> <p><b>Required?</b> yes</p> <p><b>Type:</b> string</p> <p><b>Length:</b> 1-24</p> <p><b>Multiple allowed:</b> no</p>

## Request header parameters

Element	Description
Basic Authentication using: Authorization: Basic <encoded userName:password>	Used to pass the caller's userName/password credentials over an HTTPS connection.  <b>Required:</b> must use either pnAuth or Basic Authentication  <b>Type:</b> userName and password (both are case-sensitive)
pnAuth using: Authorization: pnAuth <pnKey>	A unique key provided to the merchant during provisioning for Push Notifications, and intended for use when the merchant's application self-registers.  <b>Required:</b> must use either pnAuth or Basic Authentication. Use pnAuth if your application is self-registering.  <b>Special validation:</b> the pnKey provided in the call must match the one provisioned for your OpenMarket account.

### Example Basic Authentication

Authorization: Basic QWxhZGRpbjpvYGVuIHNLc2FtZQ==

### Example pnKey authentication

Authorization: pnAuth 9FB2D4725A9B4AC01EE1E2DB579558EE725FAD4D

## Response header parameters

Element	Description
X-Ticket-Id	The ticket ID.  <b>Required?</b> yes  <b>Type:</b> string

### Example

X-Ticket-Id: 7EYYG-EF3XM-5LSQ5-ILR73

## Response elements and attributes

Element / Attribute	Description
instance (element)	A complex element containing subelements that describe the application instance. <b>Required?</b> yes
id (attribute of instance)	The ID of the application instance. <b>Required:</b> no <b>Type:</b> string
status (attribute of instance)	The status of the application instance. Default value is ENABLED. <b>Required:</b> yes <b>Type:</b> string <b>Valid values:</b> ENABLED or DISABLED
statusDetails (attribute of instance)	Detailed information about the status of the application instance. For example, an application instance may be disabled via the APN feedback service. <b>Required:</b> no <b>Type:</b> string <b>Length:</b> 1-255 characters
apn (element of instance)	Indicates that the instanceId is for the Apple network. <b>Required:</b> use either apn (for Apple) or c2dm (for Android) <b>Multiple allowed:</b> no
deviceToken (attribute of apn)	An identifier returned by APN to the application instance when the end user registers the app for push notifications. This identifier is used as a type of destination ID. <b>Required?</b> yes <b>Type:</b> string <b>Length:</b> 1-64 characters <b>Multiple allowed?</b> No

Element / Attribute	Description
c2dm (element of instance)	<p>Indicates that the instanceId is for the C2DM network.</p> <p><b>Required:</b> use either c2dm (for Android) or apn (for Apple)</p> <p><b>Multiple allowed:</b> no</p>
registrationId (attribute of c2dm)	<p>An identifier returned by C2DM to the application instance when the end user registers the app for push notifications. This identifier is used as a type of destination ID.</p> <p><b>Type:</b> string</p> <p><b>Required:</b> yes</p> <p><b>Length:</b> 1-512 characters</p>
groups (element of instance)	<p>Complex element containing the groups to which the application instance belongs.</p> <p><b>Required:</b> no</p> <p><b>Multiple allowed:</b> no</p>
bps	<p>Indicates that the instanceId that's created or updated is for the BlackBerry network.</p> <p><b>Type:</b> element</p> <p><b>Required:</b> use one of:</p> <ul style="list-style-type: none"> <li>▶ c2dm (for Android)</li> <li>▶ apn (for Apple)</li> <li>▶ bps (for BlackBerry)</li> </ul> <p><b>Multiple allowed:</b> no</p>
pin	<p>A unique ID for a specific BlackBerry device. This ID is used as a type of destination ID.</p> <p><b>Type:</b> string</p> <p><b>Required:</b> yes</p> <p><b>Length:</b> 8 characters</p> <p><b>Multiple allowed:</b> no</p>
group (element of groups)	<p>The group to which the application instance belongs. An instance can be in more than one group.</p> <p><b>Required:</b> yes</p> <p><b>Multiple allowed:</b> yes</p>

Element / Attribute	Description
id (attribute of group)	The ID of the group to which the application instance belongs (not case-sensitive). <b>Type:</b> string <b>Required:</b> yes <b>Length:</b> 1-50 <b>Multiple allowed:</b> no

## Example response body

```
<?xml version="1.0" encoding="UTF-8"?>
<instance xmlns="http://pn.openmarket.com/v1"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
id="IILR735LSQ5EF3XM7EYYG" status="DISABLED" statusDetails="APN
Feedback">
  <apn
deviceToken="FE66489F304DC75B8D6E8200DFF8A456E8DAEACEC428B427E9518741C9
2C6660"/>
</instance>
```

```
<?xml version="1.0" encoding="UTF-8"?>
<instance xmlns="http://pn.openmarket.com/v1"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
id="IILR735LSQ5EF3XM7EYYG" status="ENABLED">
  <c2dm
registrationId="APA91bG0c6eAYjoE7xyqlbf8sbDaH1Kw_bQced9J0tJyYIE5tP3oSSq
s3_3I4R8oggLrsxXkAMR3dkOmYS6dxXPVpl_dKNLBOKq15J3GfolDaRXHDCmZ"/>
  <groups>
    <group id="Soccer"/>
    <group id="Tennis"/>
  </groups>
</instance>
```

```
<?xml version="1.0" encoding="UTF-8"?>
<instance xmlns="http://pn.openmarket.com/v1"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
id="IILR735LSQ5EF3XM7EYYG" status="ENABLED">
  <bps pin="22dcd556"/>
  <groups>
    <group id="Seattle"/>
  </groups>
</instance>
```

## Error conditions for GET Application Instance

The following conditions will result in errors:

- ▶ There's a problem with your OpenMarket account that's impeding authentication, such as:
  - ▶ The account ID or password you are using is invalid
  - ▶ Your account isn't enabled for Push Notifications
  - ▶ Your pnKey is invalid
- ▶ This applicationId does not exist in OpenMarket, perhaps because the application is not provisioned.

- ▶ The application is disabled on the target device.
- ▶ Your application is sending information that does not validate. Make sure the XML request body conforms to the schema.
- ▶ Your application is using an invalid URL, has specified the wrong version of the API, or the URL contains an invalid parameter.
- ▶ The instanceId you are trying to update does not exist in the system.

For details about all error and response codes, log on to Customer Center, select the Resources menu, and view the Response Codes.

## DELETE Application Instance

Use DELETE Application Instance when the application has been uninstalled and you want to disable the application instance in OpenMarket.

The service responds with a HTTP 204 message for all successful calls. OpenMarket returns an error if the instance ID does not exist.

## Request URL syntax, example, and parameters

### Syntax

```
DELETE
https://{VIP}/pn/v1/applications/{appId}/instances/{instanceId}
```

### Example

```
DELETE https://tdc-
push.openmarket.com/pn/v1/applications/DailyLuckyNumberApp/instanc
es/IILR735LSQ5EF3XM7EYYG
```

### Parameters

Element	Description
appId	<p>The application name as defined by the merchant. This name must be unique per merchant and must be specified during provisioning.</p> <p><b>Required:</b> yes  <b>Type:</b> string  <b>Length:</b> 1-100</p>
instanceId	<p>An identifier for an instance of an application on a mobile device.</p> <p><b>Required:</b> yes  <b>Type:</b> string  <b>Length:</b> 1-24</p>

### Request header parameters

Element	Description
Basic Authentication using: Authorization: Basic <encoded userName:password>	<p>Used to pass the caller's userName/password credentials over an HTTPS connection.</p> <p><b>Required:</b> must use either pnAuth or Basic Authentication  <b>Type:</b> userName and password (both are case-sensitive)</p>
pnAuth using: Authorization: pnAuth <pnKey>	<p>A unique key provided to the merchant during provisioning for Push Notifications, and intended for use when the merchant's application self-registers.</p> <p><b>Required:</b> must use either pnAuth or Basic Authentication. Use pnAuth if your application is self-registering.</p> <p><b>Special validation:</b> the pnKey provided in the call must match the one provisioned for your OpenMarket account.</p>

### Example Basic Authentication

Authorization: Basic QWxhZGRpbjpwVGVuIHNlc2FtZQ==

### Example pnKey authentication

Authorization: pnAuth 9FB2D4725A9B4AC01EE1E2DB579558EE725FAD4D

## Response header parameters

Element	Description
X-Ticket-Id	<p>The ticket ID.</p> <p><b>Required?</b> no</p> <p><b>Type:</b> string</p>

### Example

X-Ticket-Id: 7EYYG-EF3XM-5LSQ5-ILR73

## Error conditions with DELETE Application Instance

The following conditions will result in errors:

- ▶ There's a problem with your OpenMarket account that's impeding authentication, such as:
  - ▶ The account ID or password you are using is invalid
  - ▶ Your account isn't enabled for Push Notifications
  - ▶ Your pnKey is invalid
- ▶ The applicationId does not exist in OpenMarket, perhaps because the application is not provisioned.
- ▶ Your application is using an invalid URL, has specified the wrong version of the API, or the URL contains an invalid parameter.
- ▶ The instanceId you are trying to delete does not exist in the system.

For details about all error and response codes, log on to Customer Center, select the Resources menu, and view the Response Codes.

## POST Updated Destination ID

POST Updated Destination ID is a convenience API that lets you update the destination ID for the application instance. If, instead, you want to update all the information for an application instance, use PUT Updated Application Instance.

The service responds with a HTTP 204 message for all successful calls. OpenMarket returns an error if the instance ID you supplied does not exist or the destination ID is not valid for the network you specified.

### Request URL syntax, example, and parameters

#### Syntax

APNs	POST https://{VIP}/pn/v1/applications/{appId}/instances/{instanceId}?deviceToken={deviceToken}
C2DM	POST https://{VIP}/pn/v1/applications/{appId}/instances/{instanceId}?registrationId={registrationId}
BPS	POST https://{VIP}/pn/v1/applications/{appId}/instances/{instanceId}?pin={pin}

#### Examples

APNs	POST https://tdc-push.openmarket.com/pn/v1/applications/DailyLuckyNumberApp/instances/IILR735LSQ5EF3XM7EYYG?deviceToken=KDJ39KD03JDJ39KSKSLKJ2LK43ISF832JTFJSDL2058KFO39KFD93FI93KOO2NVNC
C2DM	POST https://tdc-push.openmarket.com/pn/v1/applications/DailyLuckyNumberApp/instances/IILR735LSQ5EF3XM7EYYG?registrationId=APA91bG0c6eAYjoE7xyqlbf8sbDaH1Kw_bQced9J0tJyYIE5tP3oSSqs3_3I4R8oggLrsxXkAMR3dkOmYS6dxXPVpl_dKNLBOKq15J3GfolDaRXHDCmZ-oI
BPS	POST https://tdc-push.openmarket.com/pn/v1/applications/DailyLuckyNumberApp/instances/IILR735LSQ5EF3XM7EYYG?pin=22dcd556

### Parameters

Element	Description
appId	<p>The application name as defined by the merchant. This name must be unique per merchant and must be specified during provisioning.</p> <p><b>Required:</b> yes</p> <p><b>Type:</b> string</p> <p><b>Length:</b> 1-100</p>
instanceId	<p>An identifier for an instance of an application on a mobile device.</p> <p>Used here to specify the instance ID to update.</p> <p><b>Required:</b> yes</p> <p><b>Type:</b> string</p> <p><b>Length:</b> 1-24</p>
deviceToken	<p>An identifier returned by APN to the application instance when the end user registers the app for push notifications. This identifier is used as a type of destination ID. Used here to specify a new deviceToken.</p> <p><b>Required:</b> conditionally required depending on the network. You must submit one of the following:</p> <ul style="list-style-type: none"> <li>▶ registrationId for c2dm (Android)</li> <li>▶ deviceToken for apn (Apple)</li> <li>▶ pin for bps (BlackBerry)</li> </ul> <p><b>Type:</b> string</p> <p><b>Length:</b> 1-64</p>
registrationId	<p>An identifier returned by C2DM to the application instance when the end user registers the app for push notifications. This identifier is used as a type of destination ID.</p> <p>Used here to specify a new registrationId.</p> <p><b>Required:</b> conditionally required depending on the network. You must submit one of the following:</p> <ul style="list-style-type: none"> <li>▶ registrationId for c2dm (Android)</li> <li>▶ deviceToken for apn (Apple)</li> <li>▶ pin for bps (BlackBerry)</li> </ul> <p><b>Type:</b> string</p> <p><b>Length:</b> 1-512</p>

Element	Description
pin	<p>A unique ID for a specific BlackBerry device. This ID is used as a type of destination ID.</p> <p><b>Required:</b> conditionally required depending on the network. You must submit one of the following:</p> <ul style="list-style-type: none"> <li>▶ registrationId for c2dm (Android)</li> <li>▶ deviceToken for apn (Apple)</li> <li>▶ pin for bps (BlackBerry)</li> </ul> <p><b>Type:</b> string</p> <p><b>Length:</b> 1-8</p>

## Request header parameters

Element	Description
Basic Authentication using: Authorization: Basic <encoded userName:password>	<p>Used to pass the caller's userName/password credentials over an HTTPS connection.</p> <p><b>Required:</b> must use either pnAuth or Basic Authentication</p> <p><b>Type:</b> userName and password (both are case-sensitive)</p>
pnAuth using: Authorization: pnAuth <pnKey>	<p>A unique key provided to the merchant during provisioning for Push Notifications, and intended for use when the merchant's application self-registers.</p> <p><b>Required:</b> must use either pnAuth or Basic Authentication. Use pnAuth if your application is self-registering.</p> <p><b>Special validation:</b> the pnKey provided in the call must match the one provisioned for your OpenMarket account.</p>

### Example Basic Authentication

```
Authorization: Basic QWxhZGRpbjpvYVUyIHNlc2FtZQ==
```

### Example pnKey authentication

```
Authorization: pnAuth 9FB2D4725A9B4AC01EE1E2DB579558EE725FAD4D
```

## Response header parameters

Element	Description
X-Ticket-Id	The ticket ID. <b>Required?</b> no <b>Type:</b> string

### Example

```
X-Ticket-Id: 7EYYG-EF3XM-5LSQ5-ILR73
```

## Error conditions for POST Updated Destination

The following conditions will result in HTTP errors:

- ▶ There's a problem with your OpenMarket account that's impeding authentication, such as:
  - ▶ The account ID or password you are using is invalid
  - ▶ Your account isn't enabled for Push Notifications
  - ▶ Your pnKey is invalid
- ▶ This applicationId does not exist in OpenMarket, perhaps because the application is not provisioned.
- ▶ The application is disabled.
- ▶ The instanceId does not exist in the OpenMarket system.
- ▶ Your application is using an invalid URL, has specified the wrong version of the API, or the URL contains an invalid parameter.
- ▶ The destination ID (APN device token, C2DM registration ID, or BPS PIN) is invalid; e.g., it exceeds the maximum length.

## POST Add Group(s) to an application instance

OpenMarket provides two convenience operations for managing an application instance's group membership:

- ▶ Use POST Add Group(s) to add one or more groups to an application instance
- ▶ Use DELETE Group(s) to remove one or more groups from an application instance

If, instead, you want to update all of the information for an application instance, use PUT Updated Application Instance.

The service responds with a HTTP 204 message for all successful calls. For both operations, OpenMarket returns an error if the instance ID does not exist.

## Request URL syntax, example, and parameters

### Syntax

```
POST
https://{VIP}/pn/v1/applications/{appId}/instances/{instanceId}?group={groupName}&group={groupName}
```

### Example

```
POST https://tdc-
pn.openmarket.com/pn/v1/applications/SportScoresApp/instances/IILR
735LSQ5EF3XM7EYYG?group=baseball
```

```
POST https://tdc-
pn.openmarket.com/pn/v1/applications/SportsScoresApp/instances/IILR
735LSQ5EF3XM7EYYG?group=soccer&group=baseball
```

### Parameters

Element	Description
appId	<p>The application name as defined by the merchant. This name must be unique per merchant and must be specified during provisioning.</p> <p>The application name as defined by the merchant. This name must be unique per merchant and must be specified during provisioning.</p> <p><b>Required:</b> no</p> <p><b>Type:</b> string</p> <p><b>Length:</b> 1-100 characters</p>
instanceId	<p>An identifier for an instance of an application on a mobile device.</p> <p>An identifier for an instance of an application on a mobile device.</p> <p><b>Required:</b> no</p> <p><b>Type:</b> string</p> <p><b>Length:</b> 1-24 characters</p>

group	<p>The name of the group, or groups, to which you want to add the application instance.</p> <p><b>Required:</b> yes</p> <p><b>Type:</b> string</p> <p><b>Length:</b> 1-50 characters</p>
-------	--

## Request header parameters

Element	Description
Basic Authentication using: Authorization: Basic <encoded userName:password>	<p>Used to pass the caller's userName/password credentials over an HTTPS connection.</p> <p><b>Required:</b> must use either pnAuth or Basic Authentication</p> <p><b>Type:</b> userName and password (both are case-sensitive)</p>
pnAuth using: Authorization: pnAuth <pnKey>	<p>A unique key provided to the merchant during provisioning for Push Notifications, and intended for use when the merchant's application self-registers.</p> <p><b>Required:</b> must use either pnAuth or Basic Authentication. Use pnAuth if your application is self-registering.</p> <p><b>Special validation:</b> the pnKey provided in the call must match the one provisioned for your OpenMarket account.</p>

### Example Basic Authentication

```
Authorization: Basic QWxhZGRpbjpvceGVuIHNLc2FtZQ==
```

### Example pnKey authentication

```
Authorization: pnAuth 9FB2D4725A9B4AC01EE1E2DB579558EE725FAD4D
```

## Response header parameters

Element	Description
X-Ticket-Id	The ticket ID. <b>Required?</b> no <b>Type:</b> string

### Example

```
X-Ticket-Id:7EYYG-EF3XM-5LSQ5-ILR73
```

## Error conditions for POST Add Group(s) and DELETE Group(s)

The following conditions will result in HTTP errors:

- ▶ There's a problem with your OpenMarket account that's impeding authentication, such as:
  - ▶ The account ID or password you are using is invalid
  - ▶ Your account isn't enabled for Push Notifications
  - ▶ Your pnKey is invalid
- ▶ You specified an application that OpenMarket cannot find, perhaps because it has not been registered with the device network or there is no instance ID for the application.
- ▶ Your application is sending information that does not validate. Make sure the XML request body conforms to the schema.
- ▶ Your application is using an invalid URL, or the URL contains an invalid parameter.

## DELETE Group(s) from an application instance

OpenMarket provides two convenience operations for managing an application instance's group membership:

- ▶ Use POST Add Group(s) to add one or more groups to an application instance
- ▶ Use DELETE Group(s) to remove one or more groups from an application instance

If, instead, you want to update all of the information for an application instance, use PUT Updated Application Instance.

The service responds with a HTTP 204 message for all successful calls. For both operations, OpenMarket returns an error if the instance ID does not exist.

## Request URL syntax, example, and parameters

### Syntax

```
DELETE
https://{VIP}/pn/v1/applications/{appId}/instances/{instanceId}?group={groupName}&group={groupName}
```

### Example

```
DELETE https://tdc-
pn.openmarket.com/pn/v1/applications/SportScoresApp/instances/IILR
735LSQ5EF3XM7EYYG?group=baseball
```

```
DELETE https://tdc-
pn.openmarket.com/pn/v1/applications/SportScoresApp/instances/IILR
735LSQ5EF3XM7EYYG?group=soccer&group=baseball
```

### Parameters

Element	Description
appId	The application name as defined by the merchant. This name must be unique per merchant and must be specified during provisioning. <b>Required:</b> yes <b>Type:</b> string <b>Length:</b> 1-100
instanceId	An identifier for an instance of an application on a mobile device. <b>Required:</b> yes <b>Type:</b> string <b>Length:</b> 1-24
group	The group, or groups, from which you want to remove the application instance. <b>Required:</b> yes <b>Type:</b> string <b>Length:</b> 1-50

## Request header parameters

Element	Description
Basic Authentication using: Authorization: Basic <encoded userName:password>	Used to pass the caller's userName/password credentials over an HTTPS connection.  <b>Required:</b> must use either pnAuth or Basic Authentication  <b>Type:</b> userName and password (both are case-sensitive)
pnAuth using: Authorization: pnAuth <pnKey>	A unique key provided to the merchant during provisioning for Push Notifications, and intended for use when the merchant's application self-registers.  <b>Required:</b> must use either pnAuth or Basic Authentication. Use pnAuth if your application is self-registering.  <b>Special validation:</b> the pnKey provided in the call must match the one provisioned for your OpenMarket account.

### Example Basic Authentication

```
Authorization: Basic QWxhZGRpbjpwVGVuIHNlc2FtZQ==
```

### Example pnKey authentication

```
Authorization: pnAuth {pnKey}
```

## Response header parameters

Element	Description
X-Ticket-Id	The ticket ID.  <b>Required?</b> no  <b>Type:</b> string

### Example

```
X-Ticket-Id:7EYYG-EF3XM-5LSQ5-ILR73
```

## POST Notification

Use POST Notification to create one or more push notifications to send to an application. When you send a push notification, you can choose to submit one POST request per notification, or you can use the groups capability to send a notification to multiple destinations. You can use a group defined as "all" to send the notification to all registered instances that are enabled.

The payload size of a notification varies depending on device type:

- ▶ 194 bytes for Apple devices
- ▶ 1000 bytes for Android devices
- ▶ 8192 bytes for BlackBerry devices

OpenMarket will not send duplicate notifications if you include a destination more than once in your request. OpenMarket attempts to process all destinations regardless of failures. If your request has malformed XML, none of the notifications may process successfully.

The service responds with a HTTP 202 Accepted message for all successful calls.

---

**NOTE:** The XML request cannot exceed 60K or an error will be returned. We suggest that you use groups to manage destination IDs.

---

## Request URL syntax, example, and parameters

### Syntax

```
https://{VIP}/pn/v1/applications/{appId}/notifications
```

---

**IMPORTANT:** The appId must be the same one as the one provided when the application was provisioned.

---

### Example

```
https://tdc-  
push.openmarket.com/pn/v1/applications/DailyLuckyNumberApp/notific  
ations
```

### Parameters

Element	Description
appId	<p>The application name as defined by the merchant. This name must be unique per merchant and must be specified during provisioning.</p> <p><b>Required?</b> yes</p> <p><b>Type:</b> string</p> <p><b>Length:</b> 1- 25 characters</p> <p><b>Multiple allowed:</b> no</p>

### Request header parameters

Element	Description
Basic Authentication using: Authorization: Basic <encoded userName:password>	<p>Used to pass the caller's userName/password credentials over an HTTPS connection.</p> <p><b>Required:</b> yes</p> <p><b>Type:</b> userName and password (both are case-sensitive)</p>

**IMPORTANT:** Do not use pnAuth for this call. If you do you will get a 401-Unauthorized error.

### Example

```
Authorization: Basic QWxhZGRpbjpvcmVudVlHbnlc2FtZQ==
```

### Request elements and attributes

Element	Description
notificationRequest (element)	<p>Contains the message to be sent.</p> <p><b>Required?</b> yes</p> <p><b>Multiple allowed?</b> No</p>

Element	Description
message (element of notificationRequest)	<p>Contains the message information in a push notification. Can be an <b>alert</b> or <b>advanced</b> type.</p> <p><b>Required?</b> yes</p> <p><b>Multiple allowed?</b> no</p>
expiration (attribute of message)	<p>Expiration date in seconds for when the message will expire. OpenMarket will not send an expired message to the network (e.g. APN, C2DM). If you do not specify this attribute, no expiration will be set. For nonexpired messages sent to the APN, this value will be included.</p> <p><b>Required?</b> no</p> <p><b>Type:</b> integer</p> <p><b>Multiple allowed?</b> no</p>
c2dmCollapseKey (attribute of message)	<p>C2DM-specific attribute to specify the collapse key. If not specified, all messages will be sent with the value "key".</p> <p><b>Type:</b> string</p> <p><b>Required?</b> no</p> <p><b>Length:</b> 1-50</p> <p><b>Multiple allowed?</b> no</p>
c2dmDelayWhileIdle (attribute of message)	<p>C2DM-specific attribute to direct C2DM to queue the message if the device is idle. If not specified, this value is set to true.</p> <p><b>Type:</b> Boolean</p> <p><b>Required?</b> no</p> <p><b>Multiple allowed?</b> no</p>
alert (element of message)	<p>Indicates that an alert-type message should be sent.</p> <p><b>Required?</b> Specify either an <b>alert</b> element or a <b>native</b> element.</p> <p><b>Multiple allowed?</b> no</p>
text (attribute of alert)	<p>Message to show in the alert.</p> <p><b>Required?</b> yes</p> <p><b>Type:</b> string</p> <p><b>Maximum size:</b> 189 bytes for APN</p>

Element	Description
native (element of message)	<p>This message type allows network-specific information to be included. A message must be either <b>alert</b> or <b>native</b> type. The native types are:</p> <ul style="list-style-type: none"> <li>▶ apn — Apple</li> <li>▶ c2dm — Android</li> <li>▶ bps — BlackBerry</li> </ul> <p>For each destination the applicable network is applied. For example, for iPhone instances, the apn type message is used to build the notification, but for MyTouch instances, the c2dm type is used.</p> <p><b>Required?</b> specify either <b>native</b> or <b>alert</b></p> <p><b>Multiple allowed?</b> no</p>
apn (element of native)	<p>Indicates that the payload message is the APN payload. This payload is sent unaltered.</p> <p><b>Required?</b> no</p> <p><b>Multiple allowed:</b> no</p>
payload (attribute of apn)	<p>The exact payload to include in the APN notification.</p> <p><b>Required?</b> yes</p> <p><b>Type:</b> string</p> <p><b>Maximum size:</b> 194 bytes</p>
c2dm (element of native)	<p>Indicates the payload type message is C2DM. The payload is specified as name-value pairs.</p> <p><b>Required?</b> no</p> <p><b>Multiple allowed?</b> no</p>
payload (element of c2dm)	<p>The name-value pairs to include in the notification.</p> <p><b>Required?</b> yes</p> <p><b>Maximum size:</b> 1000 bytes for C2DM</p>
data (element of c2dm.payload)	<p>The name-value pair to include in the notification.</p> <p><b>Required?</b> yes</p> <p><b>Multiple allowed?</b> yes</p>

Element	Description
name (attribute of data element)	The name of the name-value pair. <b>Required?</b> yes <b>Type:</b> string <b>Multiple allowed?</b> no
bps (subelement of native)	BPS payload type message. <b>Required?</b> no <b>Multiple allowed?</b> no
payload (subelement of bps)	Elements to consume to construct the bps message. <b>Required?</b> yes <b>Length:</b> 8192 bytes <b>Multiple allowed?</b> no
header (subelement of bps.payload)	Specifies an individual header key/value pair. <b>Required?</b> no <b>Multiple allowed?</b> yes
name (attribute of header)	Specifies the name of the header. <b>Required?</b> yes <b>Type:</b> string <b>Multiple allowed?</b> no
value (attribute of header)	Specifies the value of the header. <b>Required?</b> yes <b>Type:</b> string <b>Multiple allowed?</b> no
content (element of bps.payload)	Specifies BPS native content of the push notification. <b>Required?</b> yes <b>Type:</b> element <b>Multiple allowed?</b> no

Element	Description
type (attribute of bps.payload.content)	Specifies the contentType of the push notification. <b>Required?</b> no <b>Type:</b> string <b>Multiple allowed?</b> no
value (attribute of data element)	The value of the name-value pair. <b>Required?</b> yes <b>Type:</b> string <b>Multiple allowed?</b> no
destinations (element of message)	The destinations to which the message will be sent. Can be one of two types: group or instance. Notifications are sent to the combination of all destinations; i.e., the instance IDs included in groups as well as individual instance IDs. If duplicate IDs are identified, the duplicates are removed. <b>Required?</b> yes <b>Multiple allowed:</b> no
instance (element of destination)	Destination is specified as an application instance ID. <b>Required?</b> no <b>Minimum/maximum:</b> 0-5000 <b>Multiple allowed:</b> yes
id (attribute of instance)	The ID of the application instance to which the notification is pushed. <b>Required?</b> yes <b>Type:</b> string <b>Multiple allowed:</b> no
group (element of destinations)	A group of instance IDs representing a destination. <b>Required?</b> no <b>Minimum/maximum:</b> 0-500 <b>Multiple allowed:</b> yes

Element	Description
id (attribute of group)	The ID of the group of application instance IDs. Notifications are sent to all registered instances that are enabled.  <b>Required?</b> yes  <b>Type:</b> string  <b>Maximum:</b> 1-50  <b>Multiple allowed:</b> no

## XML schemas

`http://tdc-push.openmarket.com/pn/xsd/com/openmarket/pn/v1/notificationRequest_V1.0.xsd`

`http://tdc-push.openmarket.com/pn/xsd/com/openmarket/pn/v1/error_V1.0.xsd`

## Example request body

```

<?xml version="1.0" encoding="UTF-8"?>
<notificationRequest xmlns="http://pn.openmarket.com/v1"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
  <message expiration="1295394966">
    <native>
      <apn>
        <payload>{"aps": {"alert": "You have new mail.",
"badge" : 1, "sound" : "bingbong.aiff"}}</payload>
      </apn>
      <c2dm>
        <payload>
          <data name="alert" value="You have new mail."/>
          <data name="sound" value="beep"/>
          <data name="badge" value="1"/>
        </payload>
      </c2dm>
      <bps>
        <payload>
          <header name="username"
value="Bob"/>
          <header name="wantsAlerts"
value"true"/>
          <content
type="image/gif">cm5hbCBwbGVhc3VyZS4=</content>
        </payload>
      </bps>
    </native>
    <destinations>
      <group id="ALL"/>
      <instance id="myiPhoneA"/>
    </destinations>
  </message>
</notificationRequest>

<?xml version="1.0" encoding="UTF-8"?>
<notificationRequest xmlns="http://pn.openmarket.com/v1"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
  <message expiration="1295394966">
    <native>
      <apn>
        <payload>{"aps": {"alert": "You have new mail.",
"badge" : 1, "sound" : "bingbong.aiff"}}</payload>
      </apn>
      <c2dm>
        <payload>
          <data name="alert" value="You have new mail."/>
          <data name="sound" value="beep"/>
        </payload>
      </c2dm>
    </native>
  </message>
</notificationRequest>

```

```

        <data name="badge" value="1"/>
      </payload>
    </c2dm>
    <bps>
      <payload>
        <header name="username"
value="Bob"/>
        <header name="wantsAlerts"
value"true"/>
        <content
type="text/json">{menu:{id:4,value: true}}</content>
      </payload>
    </bps>
  </native>
  <destinations>
    <group id="ALL"/>
    <instance id="myiPhoneA"/>
  </destinations>
</message>
</notificationRequest>

<?xml version="1.0" encoding="UTF-8"?>
<notificationRequest xmlns="http://pn.openmarket.com/v1"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
  <message c2dmCollapseKey="Reminder" c2dmDelayWhileIdle="false">
    <alert text="Time to do a backup!"/>
    <destinations>
      <instance id="myiPhoneA"/>
      <instance id="myTouchA"/>
    </destinations>
  </message>
</notificationRequest>

```

## Conditions under which OpenMarket disables an application instance

The following conditions will cause OpenMarket to disable an application instance. To re-enable the instance, call either PUT Updated Application Instance or POST Updated Destination ID.

- ▶ If APNs rejects an application's certificate. In this case you should provision a valid certificate.
- ▶ If APNs rejects a device token. In this case, update the device token for the instance using PUT Update Application Instance.
- ▶ If C2DM responds with InvalidRegistration or NotRegistered error to the push request. In this case, provide valid C2DM credentials.

- ▶ If BPS responds with `UnauthorizedException` to the push request. In this case, provide a valid application ID and password.

---

**NOTE:** OpenMarket checks the APN and BPS feedback lists (which identify invalid instance IDs) and will not send notifications to any destinations on the list.

---

## C2DM data handling considerations

When creating your C2DM request, it is important to consider the interface between the Android application and the request payload. Following are C2DM data handling considerations to take into account.

- ▶ All C2DM payload name value pairs will be sent as data. `<name>=<value>`
- ▶ For the Alert type, the alert text will be sent as `data.alert=<alert text>`
- ▶ Several additional parameters are sent in the payload. These values can be optionally specified in the message body.
  - ▶ `collapse_key=<c2dmCollapseKey>` (default=key)
  - ▶ `delay_while_idle=<c2dmDelayWhileIdle>` (default=true)
  - ▶ `data.mid=<mid>...` This is an internal message id that is set by OM and included in the payload. (Not an attribute of request).

## BlackBerry considerations

When creating your request, note the following about BlackBerry Push Service notifications:

- ▶ For the Alert type, the alert text will be sent as content type "text/plain".
- ▶ The HTTP header, X-ALERT, will be set with a value of "true".
- ▶ Notifications will include HTTP header X-MID. The value of this header is the message ID set by OpenMarket.

## Response header parameters

Element	Description
X-Ticket-Id	The ticket ID. <b>Required?</b> yes <b>Type:</b> string
X-Estimated-Msg-Count	An estimate of the number of message that will be sent for this request. This count is only an estimate and does not take into account any instances that might belong to more than one group specified in the request. It also may change due to instances that have registered or unregistered since the count was made. If an estimate is not retrievable, a value of -1 is returned to represent unknown. <b>Required?</b> yes <b>Type:</b> integer

### Examples

X-Ticket-Id: 7EYYG-EF3XM-5LSQ5-ILR73

X-Estimated-Msg-Count: 303

## Error conditions

The following conditions will result in HTTP errors:

- ▶ There's a problem with your OpenMarket account that's impeding authentication, such as:
  - ▶ The account ID or password you are using is invalid
  - ▶ Your account isn't enabled for Push Notifications
- ▶ You specified an application that OpenMarket cannot find, perhaps because it has not been registered with the device network or there is no instance ID for the application.
- ▶ Your application is sending information that does not validate. Make sure the XML request body conforms to the schema.
- ▶ Your application is using an invalid URL, has specified the wrong version of the API, or the URL contains an invalid parameter.
- ▶ Your application is using an invalid HTTP method.
- ▶ The application on the target device is disabled.

- ▶ The expiration set for the notification has passed.

For details about all error and response codes, log on to Customer Center, select the Resources menu, and view the Response Codes.

## PUT Message Receipt

Use the callback operation, PUT Message Receipt, to create a message receipt for a push notification. The message receipt indicates that the message was delivered, and it can additionally indicate whether the user interacted with the application or deleted the message.

The service responds with a HTTP 202 Accepted message for all successful calls.

---

**NOTE:** This feature is supported for C2DM and BPS.

---

## Request URL syntax, example, and parameters

### Syntax

```
PUT https://{VIP}/pn/v1/applications/{appId}/messageReceipts/{msgId}?status={status}&statusDetails={statusDetails}
```

### Examples

```
PUT https://tdc-pn.openmarket.com/pn/v1/applications/DailyLuckyNumberApp/messageReceipts/AAAAff78
```

```
PUT https://tdc-pn.openmarket.com/pn/v1/applications/DailyLuckyNumberApp/messageReceipts/8Fx9-4t6?status=User_Engaged
```

```
PUT https://tdc-pn.openmarket.com/pn/v1/applications/DailyLuckyNumberApp/messageReceipts/He9_-3F6?status=User_Deleted&statusDetails=User has removed the notification from the inbox
```

### Parameters

Element	Description
appId	<p>The application name as defined by the merchant. This name must be unique per merchant and must be specified during provisioning.</p> <p><b>Required?</b> yes</p> <p><b>Type:</b> string</p> <p><b>Length:</b> 1- 25 characters</p> <p><b>Multiple allowed:</b> no</p>
messageId	<p>The identifier for the message that is generated by OpenMarket and included in the notification payload to the device. Must be a valid ID that exists in the system.</p> <p>For BPS, this is included in the header as X-MID.</p> <p>For C2DM, this is found in the payload as data.MID.</p> <p>For APNs, this is in the payload as "MID": "&lt;msg id&gt;".</p> <p><b>IMPORTANT:</b> messageId is case-sensitive. It must be entered on the URL exactly as it was received.</p> <p><b>Required?</b> yes</p> <p><b>Type:</b> string</p> <p><b>Length:</b> 1-16 characters</p> <p><b>Multiple allowed:</b> no</p>

### Request header parameters

Element	Description
Basic Authentication using: Authorization: Basic <encoded userName:password>	<p>Used to pass the caller's userName/password credentials over an HTTPS connection.</p> <p><b>Required:</b> must use either pnAuth or Basic Authentication</p> <p><b>Type:</b> userName and password (both are case-sensitive)</p>
pnAuth using: Authorization: pnAuth	<p>A unique key provided to the merchant during provisioning for Push Notifications, and intended for use when the merchant's application self-</p>

<pnKey>	<p>registers.</p> <p><b>Required:</b> must use either pnAuth or Basic Authentication. Use pnAuth if your application is self-registering.</p> <p><b>Special validation:</b> the pnKey provided in the call must match the one provisioned for your OpenMarket account.</p>
---------	--

### Example using Basic Authentication

Authorization: Basic QWxhZGRpbjpvvcGVuIHNlc2FtZQ==

### Example using pnKey

Authorization: pnAuth 9FB2D4725A9B4AC01EE1E2DB579558EE725FAD4D

## XML schemas

[http://tdc-push.openmarket.com/pn/xsd/com/openmarket/pn/v1/status\\_V1.0.xsd](http://tdc-push.openmarket.com/pn/xsd/com/openmarket/pn/v1/status_V1.0.xsd)

[http://tdc-push.openmarket.com/pn/xsd/com/openmarket/pn/v1/error\\_V1.0.xsd](http://tdc-push.openmarket.com/pn/xsd/com/openmarket/pn/v1/error_V1.0.xsd)

## Response header parameters

Element	Description
X-Ticket-Id	<p>The ticket ID.</p> <p><b>Required?</b> no</p> <p><b>Type:</b> string</p>

### Example

X-Ticket-Id:7EYYG-EF3XM-5LSQ5-ILR73

## Error conditions

- ▶ There's a problem with your OpenMarket account that's impeding authentication, such as:
  - ▶ The account ID or password you are using is invalid
  - ▶ Your account isn't enabled for Push Notifications
- ▶ You specified an application that OpenMarket cannot find, perhaps because it has not been registered with the device network or there is no instance ID for the application.

- ▶ The notification status is not valid.
- ▶ A notification with the specified message ID cannot be found.
- ▶ Your application is using an invalid URL, has specified the wrong version of the API, or the URL contains an invalid parameter.

## GET Notification Status for an Instance

Use GET Notification Status to retrieve the state of a notification for an application instance.

The service responds with a HTTP 200 message for all successful calls.

**NOTE:** Statuses for notifications sent within the last 30 days are guaranteed to be returned.

## Request URL syntax, example, and parameters

### Syntax

```
https://{VIP}/pn/v1/applications/{appId}/instances/{instanceId}/notifications/{ticketId}/status
```

### Example

```
https://tdc-
push.openmarket.com/pn/v1/applications/DailyLuckyNumberApp/instances/IILR735LSQ5EF3XM7EYYG/notifications/7EYYG-EF3XM-5LSQ5-IILR73/status
```

### Parameters

Element	Description
appId	<p>The application name as defined by the merchant. This name must be unique per merchant and must be specified during provisioning.</p> <p><b>Required:</b> yes</p> <p><b>Type:</b> string</p> <p><b>Length:</b> 1-100</p>

instanceId	<p>An identifier for an instance of an application on a mobile device.</p> <p><b>Required:</b> yes</p> <p><b>Type:</b> string</p> <p><b>Length:</b> 1-24</p>
ticketId	<p>The identifier OpenMarket assigns to a transaction following a merchant's call to the POST Notification API.</p> <p>In this case it is the identifier returned from the POST Notification operation.</p> <p><b>Required:</b> yes</p> <p><b>Type:</b> string</p> <p><b>Length:</b> 1-32</p>

## Request header parameters

Element	Description
Basic Authentication using: Authorization: Basic <encoded userName:password>	<p>Used to pass the caller's userName/password credentials over an HTTPS connection.</p> <p><b>Required:</b> yes</p> <p><b>Type:</b> userName and password (both are case-sensitive)</p>

**IMPORTANT:** Do not use pnAuth for this call. If you do you will get a 401-Unauthorized error.

### Example

```
Authorization: Basic QWxhZGRpbjpvcmVudHJlc2FtZQ==
```

## XML schemas

[http://tdc-push.openmarket.com/pn/xsd/com/openmarket/pn/v1/status\\_v1.0.xsd](http://tdc-push.openmarket.com/pn/xsd/com/openmarket/pn/v1/status_v1.0.xsd)

[http://tdc-push.openmarket.com/pn/xsd/com/openmarket/pn/v1/error\\_v1.0.xsd](http://tdc-push.openmarket.com/pn/xsd/com/openmarket/pn/v1/error_v1.0.xsd)

## 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.
- ▶ **Delivered** — OpenMarket sent the notification and the application on the device responded back that it received the message. Depending on the needs of your application, you can set this state using [PUT Message Receipt](#) on page 90.
- ▶ **User\_Engaged** — The application on the end user's device responded back that the notification led to the end user engaging with the application. Depending on the needs of your application, you can set this state using [PUT Message Receipt](#) on page 90.
- ▶ **User\_Deleted** — The application responded back that the end user ignored or deleted the message. Depending on the needs of your application, you can set this state using [PUT Message Receipt](#) on page 90.

## Response elements and attributes

Element	Description
status (element)	The status of the application instance. Default value is ENABLED. <b>Required?</b> yes <b>Type:</b> string
appId	The application name as defined by the merchant. This name must be unique per merchant and must be specified during provisioning. <b>Required?</b> yes <b>Type:</b> string <b>Length:</b> 1- 25 characters <b>Multiple allowed:</b> no

<p>instanceId (attribute of status)</p>	<p>An identifier for an instance of an application on a mobile device. OpenMarket returns this ID when the instance is created or replaced.</p> <p><b>Required?</b> yes</p> <p><b>Type:</b> string</p> <p><b>Length:</b> 1-24</p> <p><b>Multiple allowed:</b> no</p>
<p>ticketId (attribute of status)</p>	<p>The identifier OpenMarket assigns to a transaction following a merchant's call to the POST Notification API.</p> <p><b>Required?</b> yes</p> <p><b>Type:</b> string</p> <p><b>Maximum length:</b> 1-32 characters</p>
<p>submittedDate (attribute of status)</p>	<p>The date the push notification was submitted. It is an attribute of the <b>status</b> element.</p> <p><b>Required?</b> yes</p> <p><b>Type:</b> string</p>
<p>value (attribute of status)</p>	<p>The value of the push notification, which can be:</p> <ul style="list-style-type: none"> <li>▶ Processed</li> <li>▶ Failed</li> </ul> <p><b>Required?</b> yes</p> <p><b>Type:</b> string</p> <p><b>Maximum character length:</b> 25</p>
<p>mid (attribute of status)</p>	<p>The identifier for the message that is generated by OpenMarket and included in the notification payload to the device.</p> <p><b>Required?</b> no</p> <p><b>Type:</b> string</p> <p><b>Length:</b> 1-16 characters</p>
<p>processedDate (attribute of status)</p>	<p>An optional attribute of the status element, this is the date when the notification was processed.</p> <p><b>Required?</b> no</p> <p><b>Type:</b> date</p>

deliveryDate (attribute of status)	The date when the application instance on the end user's device confirmed receipt of the notification. <b>Required?</b> no <b>Type:</b> date
userEngagedDate (attribute of status)	The date when the application instance on the end user's device reported that end user interacted with the application after receiving the notification. <b>Required?</b> no <b>Type:</b> date
userDeletedDate (attribute of status)	The date when the application instance on the end user's device reported that the end user deleted the notification. <b>Required?</b> no <b>Type:</b> date
details (element of status)	Provides additional details about the status, such as why a notification failed. <b>Required?</b> no <b>Type:</b> element of status <b>Length:</b> up to 500 characters

## Example response body

### Example for a notification with a "processed" state

```
<?xml version="1.0" encoding="UTF-8"?>
<status xmlns="http://pn.openmarket.com/v1"
  appId="DailyLuckyNumberApp"
  instanceId="IILR735LSQ5EF3XM7EYYG"
  ticketId="7EYYG-EF3XM-5LSQ5-ILR73"
  MID="AAAAff78"
  value="Processed"
  submittedDate="2010-10-01T15:59:00.001-08:00"
  processedDate="2010-10-01T15:59:00.003-08:00"/>
```

### Example for a notification with a "failed" state

```
<?xml version="1.0" encoding="UTF-8"?>
<status xmlns="http://pn.openmarket.com/v1"
  appId="DailyLuckyNumberApp"
  MID="8Fx9-4t6"
  instanceId="IILR735LSQ5EF3XM7EYYG"
  ticketId="7EYYG-EF3XM-5LSQ5-ILR73"
  value="Failed"
  submittedDate="2010-10-01T15:59:00.001-08:00"
  processedDate="2010-10-01T15:59:00.003-08:00">
  <details>Invalid token</details>
</status>
```

### Example for a notification with a "delivered" state

```
<?xml version="1.0" encoding="UTF-8"?>
<status xmlns="http://pn.openmarket.com/v1"
  value="DELIVERED" ticketId="01A94-470HJ-1J1FB-4JOJ7"
  MID="AAAAff78"
  processedDate="2011-05-31T17:51:55.000-04:00"
  instanceId="I02A96470H0130R8BH91R"
  deliveredDate="2011-05-31T17:51:59.000-04:00"
  submittedDate="2011-05-31T17:51:55.000-04:00"
  appId="DailyLuckyNumberApp">
  <details>Message received, sent to notification
bar.</details>
</status>
```

## Error conditions

The following conditions will result in HTTP errors:

- ▶ There's a problem with your OpenMarket account that's impeding authentication, such as:
  - ▶ The account ID or password you are using is invalid

- ▶ Your account isn't enabled for Push Notifications
- ▶ The applicationId or instanceId does not exist.
- ▶ OpenMarket cannot find the specified ticket ID.
- ▶ Your application is using an invalid HTTP method.
- ▶ Your application is using an invalid URL, has specified the wrong version of the API, or the URL contains an invalid parameter.

For details about all error and response codes, log on to Customer Center, select the Resources menu, and view the Response Codes.

## GET Notification Status Summary

Use GET Notification Status Summary to retrieve information, by network, about the push notifications and message receipts that you submit to OpenMarket. For example, you can retrieve the HTTP and OpenMarket response codes returned for a push notification, as well as the state of the push notification, and the content that was included in the notification.

The GET operation returns summary data that's available when you make the call. If you make subsequent calls you may get back different results since the state of notifications changes over time.

OpenMarket will return data for all requests sent within 30 days of the date when you called the API.

The service responds with a HTTP 200 message for all successful calls.

## Request URL syntax, example, and parameters

### Syntax

```
https://{host:port}/pn/v1/applications/{appId}/notifications/{ticketId}
```

### Examples

In this example, neither the **content** nor the **summary** parameter are included, which invokes the default behavior; both the content of the push notification and the summary information will be returned.

```
https://tdc-  
pn.openmarket.com/pn/v1/applications/DailyLuckyNumberApp/notificat  
ions/7EYYG-EF3XM-5LSQ5-ILR73
```

In this example, the **content** parameter is set to false, meaning that the content of the notification will not be returned.

```
https://tdc-
pn.openmarket.com/pn/v1/applications/DailyLuckyNumberApp/notificat
ions/7EYYG-EF3XM-5LSQ5-ILR73?content=false
```

In this example, the **summary** parameter is set to false, meaning that summary data of the notification request will not be returned.

```
https://tdc-
pn.openmarket.com/pn/v1/applications/DailyLuckyNumberApp/notificat
ions/7EYYG-EF3XM-5LSQ5-ILR73?summary=false
```

### Parameters

Element	Description
appId	<p>The application name as defined by the merchant. This name must be unique per merchant and must be specified during provisioning.</p> <p><b>Required:</b> yes</p> <p><b>Type:</b> string</p> <p><b>Length:</b> 1-100</p>
ticketId	<p>The identifier OpenMarket assigns to a transaction following a merchant's call to the POST Notification API.</p> <p><b>Required:</b> yes</p> <p><b>Type:</b> string</p> <p><b>Length:</b> 1-32</p>
content	<p>Indicates whether the content of the notification request should be returned. Default value is true.</p> <p><b>Required:</b> yes</p> <p><b>Type:</b> Boolean</p> <p><b>Length:</b> 1-32</p> <p><b>Multiple allowed:</b> no</p>
summary	<p>Indicates whether the message summary data of the notification request should be returned. Default is true.</p> <p><b>Type:</b> Boolean</p> <p><b>Multiple allowed:</b> no</p>

## Request header parameters

Element	Description
Basic Authentication using: Authorization: Basic <encoded userName:password>	Used to pass the caller's userName/password credentials over an HTTPS connection.  <b>Required:</b> yes  <b>Type:</b> userName and password (both are case-sensitive)

### Example

Authorization: Basic QWxhZGRpbjpwcvGVuIHNlc2FtZQ==

## Response elements and attributes

Element	Description and Attributes		
<b>notificationSummary</b>	A complex element containing subelements that provide information—by network—about the push notifications and message receipts that you submit to OpenMarket.  <b>Required?</b> yes  <b>Type:</b> element  <b>Multiple allowed:</b> no		
	<table border="1"> <tr> <td>appId</td> <td>The application name as defined by the merchant. This name must be unique per merchant and must be specified during provisioning.  <b>Required?</b> yes  <b>Type:</b> string  <b>Length:</b> 1-100  <b>Multiple allowed:</b> no</td> </tr> </table>	appId	The application name as defined by the merchant. This name must be unique per merchant and must be specified during provisioning.  <b>Required?</b> yes  <b>Type:</b> string  <b>Length:</b> 1-100  <b>Multiple allowed:</b> no
appId	The application name as defined by the merchant. This name must be unique per merchant and must be specified during provisioning.  <b>Required?</b> yes  <b>Type:</b> string  <b>Length:</b> 1-100  <b>Multiple allowed:</b> no		

Element	Description and Attributes	
	ticketId	<p>The identifier OpenMarket assigns to a transaction following a merchant's call to the POST Notification API.</p> <p><b>Required?</b> yes</p> <p><b>Type:</b> string</p> <p><b>Length:</b> 1-32</p> <p><b>Multiple allowed:</b> no</p>
	submittedDate	<p>The date when you submitted the notification request.</p> <p><b>Required?</b> yes</p> <p><b>Type:</b> date</p> <p><b>Multiple allowed:</b> no</p>
	reportingDate	<p>The date when the summary data was generated and returned.</p> <p><b>Required?</b> yes</p> <p><b>Type:</b> date</p> <p><b>Multiple allowed:</b> no</p>
<b>content</b>	<p>Child element of <b>notificationSummary</b> containing the content included in the notification request you submitted (e.g., the alert text or message text).</p> <p><b>Required?</b> no</p> <p><b>Type:</b> string</p> <p><b>Multiple allowed:</b> no</p>	
<b>response</b>	<p>Child element of <b>notificationSummary</b> containing the response to the notification request you submitted.</p> <p><b>Required?</b> yes</p> <p><b>Type:</b> element</p> <p><b>Multiple allowed:</b> no</p>	
	httpResponseCode	<p>The HTTP response code sent back in response to the notification request you submitted.</p> <p><b>Required?</b> yes</p> <p><b>Type:</b> int</p>

Element	Description and Attributes
	<p><b>Multiple allowed:</b> no</p>
<b>omResponseCode</b>	<p>Child element of <b>response</b> containing the OpenMarket response code that was sent back in the response body to the notification request you submitted.</p> <p><b>Required?</b> no</p> <p><b>Type:</b> string</p> <p><b>Multiple allowed:</b> no</p>
<b>omResponseDesc</b>	<p>Child element of <b>response</b> containing the description of the OpenMarket response code that was sent back in the response body to the notification request you submitted.</p> <p><b>Required?</b> no</p> <p><b>Type:</b> string</p> <p><b>Multiple allowed:</b> no</p>
<b>messageSummary</b>	<p>Child element of <b>notificationSummary</b> containing the summary count and related states of the messages (notifications) generated for the submitted notification request (i.e., POST Notification).</p> <p><b>Required?</b> no</p> <p><b>Type:</b> element</p> <p><b>Multiple allowed:</b> no</p>
<b>apn</b>	<p>Child element of <b>messageSummary</b> containing the APN (Apple) network messages.</p> <p><b>Required?</b> yes</p> <p><b>Type:</b> element</p> <p><b>Multiple allowed:</b> no</p>

Element	Description and Attributes	
	Total	The total count of messages for the network. <b>Required?</b> Child element of <b>messageSummary</b> containing the APN (Apple) network messages. <b>Required?</b> yes <b>Type:</b> element <b>Multiple allowed:</b> no <b>Type:</b> integer <b>Multiple allowed:</b> no
<b>c2dm</b>		Child element of <b>messageSummary</b> containing the C2DM (Android) network messages. <b>Required?</b> yes <b>Type:</b> element <b>Multiple allowed:</b> no
	total	The total count of messages for this network. <b>Required?</b> yes <b>Type:</b> integer <b>Multiple allowed:</b> no
<b>bps</b>		Child element of <b>messageSummary</b> containing the BPS (BlackBerry) network messages. <b>Required?</b> yes <b>Type:</b> element <b>Multiple allowed:</b> no
	total	The total count of messages for this network. <b>Required?</b> yes <b>Type:</b> integer <b>Multiple allowed:</b> no

Element	Description and Attributes	
<b>status</b>	Child element of <b>apn</b> , <b>c2dm</b> , and <b>bps</b> , which contains the status of the network.  <b>Required?</b> yes <b>Type:</b> element <b>Multiple allowed:</b> yes	
	count	The count of notifications for each status.  <b>Required?</b> yes if <b>messageSummary</b> is requested <b>Type:</b> integer <b>Multiple allowed:</b> no
	name	The status name.  <b>Required?</b> yes if <b>messageSummary</b> is requested <b>Type:</b> string <b>Multiple allowed:</b> no
<b>subStatus</b>	Child element of <b>status</b> containing detailed information that may exist for a notification in a FAILED state.  <b>Required?</b> no <b>Type:</b> element of status <b>Multiple allowed:</b> yes	
	count	The count of notifications for each subStatus.  <b>Required?</b> yes <b>Type:</b> integer <b>Multiple allowed:</b> no
	name	The subStatus name.  <b>Required?</b> yes <b>Type:</b> string <b>Multiple allowed:</b> no

## XML schemas

`http://tdc-push.openmarket.com/pn/xsd/com/openmarket/pn/v1/notificationSummary_V1.0.xsd`

`http://tdc-push.openmarket.com/pn/xsd/com/openmarket/pn/v1/error_V1.0.xsd`

## Example responses

**Example response with content and message summary**

```

<?xml version="1.0" encoding="UTF-8"?>
<notificationSummary xmlns="http://pn.openmarket.com/v1"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
appId="DailyLuckyNumberApp" ticketId="7EYYG-EF3XM-5LSQ5-ILR73"
submittedDate="2010-10-01T15:59:00.001-08:00" reportingDate="2010-10-
01T19:00:00.000-08:00">
  <response httpResponseCode="202"/>
  <content>&lt;?xml version=&quot;1.0&quot; encoding=&quot;UTF-
8&quot;?&gt;&lt;notificationRequest
xmlns=&quot;http://pn.openmarket.com/v1&quot;
xmlns:xsi=&quot;http://www.w3.org/2001/XMLSchema-instance&quot;&gt;
&lt;message&gt;&lt;native&gt;&lt;apn&gt;&lt;payload&gt;{&quot;aps&quot;
:&quot;alert&quot;; &quot;This is Chris' push to the
APN.&quot;;&quot;badge&quot;;1,&quot;sound&quot;;:&quot;default&quot;}}&
lt;/payload&gt;
&lt;/apn&gt;&lt;/native&gt;&lt;/destinations&gt;&lt;/instance
id=&quot;I01ASW300JC1E005Q4VP9&quot;/&gt;&lt;/instance
id=&quot;I01AEY5E0HW0J0UC7TU64&quot;/&gt;&lt;/destinations&gt;&lt;/mess
age&gt;&lt;/notificationRequest&gt;
  </content>
  <messageSummary>
    <apn total="418">
      <status name="PROCESSED" count="289"/>
      <status name="DELIVERED" count="38"/>
      <status name="USER_ENGAGED" count="6"/>
      <status name="USER_DELETED" count="6"/>
      <status name="FAILED" count="79">
        <subStatus name="APN Send failed"
count="8"/>
        <subStatus name="INVALID_TOKEN_SIZE"
count="1"/>
        <subStatus name="PROCESSING_ERROR"
count="1"/>
        <subStatus name="INVALID_TOKEN" count="16"/>
        <subStatus name="Instance Not Enabled"
count="53"/>
      </status>
    </apn>
    <c2dm total="182">
      <status name="FAILED" count="182">
        <subStatus name="C2DM payload exceeds max
size" count="182"/>
      </status>
    </c2dm>
    <bps total="201">
      <status name="PROCESSED" count="2"/>
      <status name="DELIVERED" count="125"/>
      <status name="USER_ENGAGED" count="53"/>
      <status name="USER_DELETED" count="10"/>
      <status name="FAILED" count="11">

```

```
count="10"/>
    <subStatus name="Missing BPS credentials"
  </status>
</bps>
</messageSummary>
</notificationSummary>
```

### Example with a POST notification request that had an error

```
<?xml version="1.0" encoding="UTF-8"?>
<notificationSummary
  appId="DailyLuckyNumberApp"
  ticketId="3AYYG-EF3XM-5LSQ5-98LKE"
  submittedDate="2010-10-01T15:59:00.002-08:00"
  reportingDate="2010-10-01T19:00:00.001-08:00">
  <response httpResponseCode="400">
<omResponseCode>9017</omResponseCode>mResponseDesc>Application is
disabled</omResponseDesc> </response> <response httpResponseCode="400">
  <omResponseCode>9017</omResponseCode>
  <omResponseDesc>Application is disabled</omResponseDesc>
  </response>
</notificationSummary>
```

## Error conditions

The following conditions will result in HTTP errors:

- ▶ There's a problem with your OpenMarket account that's impeding authentication, such as:
  - ▶ The account ID or password you are using is invalid
  - ▶ Your account isn't enabled for Push Notifications
- ▶ The applicationId or instanceId does not exist.
- ▶ OpenMarket cannot find the specified ticket ID.
- ▶ Your application is using an invalid HTTP method.
- ▶ Your application is using an invalid URL, has specified the wrong version of the API, or the URL contains an invalid parameter.

For details about all error and response codes, log on to Customer Center, select the Resources menu, and view the Response Codes.

## XML schemas

Access the Push Notifications schemas at these URLs:

[http://tdc-push.openmarket.com/pn/xsd/com/openmarket/pn/v1/instance\\_V1.0.xsd](http://tdc-push.openmarket.com/pn/xsd/com/openmarket/pn/v1/instance_V1.0.xsd)

[http://tdc-push.openmarket.com/pn/xsd/com/openmarket/pn/v1/instanceType\\_V1.0.xsd](http://tdc-push.openmarket.com/pn/xsd/com/openmarket/pn/v1/instanceType_V1.0.xsd)

[http://tdc-push.openmarket.com/pn/xsd/com/openmarket/pn/v1/notificationRequest\\_V1.0.xsd](http://tdc-push.openmarket.com/pn/xsd/com/openmarket/pn/v1/notificationRequest_V1.0.xsd)

[http://tdc-push.openmarket.com/pn/xsd/com/openmarket/pn/v1/notificationSummary\\_V1.0.xsd](http://tdc-push.openmarket.com/pn/xsd/com/openmarket/pn/v1/notificationSummary_V1.0.xsd)

[http://tdc-push.openmarket.com/pn/xsd/com/openmarket/pn/v1/status\\_V1.0.xsd](http://tdc-push.openmarket.com/pn/xsd/com/openmarket/pn/v1/status_V1.0.xsd)

[http://tdc-push.openmarket.com/pn/xsd/com/openmarket/pn/v1/error\\_V1.0.xsd](http://tdc-push.openmarket.com/pn/xsd/com/openmarket/pn/v1/error_V1.0.xsd)

# Index

---

## A

API Reference • 29

## B

BlackBerry considerations • 88

## C

C2DM data handling considerations • 88  
 Conditions under which OpenMarket disables an application instance • 87  
 Connecting to the service • 26  
 Creating a message receipt • 25

## D

Data flow with application self-registration • 14  
 Deciding how to handle registration of application instances • 11  
 DELETE Application Instance • 67  
 DELETE Group(s) from an application instance • 76  
 Deleting an application instance • 17

## E

Enabling your application to use BPS • 10  
 Enabling your application to use C2DM • 8  
 Error conditions • 38, 49, 60, 89, 92, 98, 108  
 Error conditions for GET Application Instance • 66  
 Error conditions for POST Add Group(s) and DELETE Group(s) • 76  
 Error conditions for POST Updated Destination • 73  
 Error conditions with DELETE Application Instance • 69  
 Example request body • 34, 44, 86  
 Example response body • 38, 49, 59, 66, 98  
 Example responses • 106  
 Exporting Apple Push Certificate to a .p12 file • 7

## G

GET Application Instance information (device-side) • 60  
 GET Application Instances information (server-side) • 50  
 GET Notification Status for an Instance • 93

GET Notification Status Summary • 99  
 Getting set up for APNs • 7  
 Getting set up for C2DM • 8

## P

POST Add Group(s) to an application instance • 73  
 POST Notification • 79  
 POST Updated Destination ID • 70  
 Prerequisites to using the service • 6  
 Pushing a notification to a BlackBerry device • 23  
 Pushing a notification to an Android device • 21  
 Pushing a notification to an Apple device • 19  
 PUT Message Receipt • 90  
 PUT New Application Instance • 29  
 PUT Updated Application Instance • 39

## R

Reasons why a push notification can fail • 28  
 Registering the application instance where your server manages the process • 13  
 Request elements and attributes • 31, 41, 80  
 Request header parameters • 30, 40, 54, 62, 68, 72, 75, 78, 80, 91, 94, 101  
 Request URL syntax, example, and parameters • 29, 40, 51, 61, 67, 70, 74, 77, 79, 90, 93, 99  
 Response elements and attributes • 35, 45, 55, 63, 95, 101  
 Response header parameters • 35, 45, 55, 62, 69, 73, 76, 78, 89, 92

## S

Special requirements for BPS • 10  
 States of a notification • 95

## T

Testing a push notification • 27

## U

Updating just the destinationId for an application instance • 16  
 Use cases you can implement with OpenMarket Push Notifications • 12  
 Using Push Notifications • 6

## X

XML schemas • 33, 43, 85, 92, 94, 106, 109  
 XML schemas for GET Application Instance • 55

