

Push IO ®

API Guide

Version 2.0

Last Updated August 1, 2012

THIS DOCUMENT IS PRELIMINARY - INCOMPLETE AND SUBJECT TO CHANGE.

Copyright © 2009 - 2012 All Rights Reserved Push IO Inc.

Push IO® is a registered trademark in the United States.

All other trademarks are the property of their respective owners.

Software License Information

The use of any and all Push IO software regardless of release state is governed by the terms of a Software License Agreement and Terms of Use that are not included here but required by reference.

Privacy Information

The use of Push IO services is subject to various privacy policies and restrictions on the use of end user information. For complete information please refer to your Master Agreement, our website privacy policies, and/or other related documents.

Billing

For information on your account and billing, please contact sales@push.io

Contact Info

Push IO

1035 Pearl Street, Suite 400

Boulder, CO 80302 USA

303-335-0903

support@push.io

<http://push.io>

Table of Contents

Preface	4
Definitions	5
API Integration	6
Overview	6
API Prerequisites	6
API Setup	7

Preface

Push IO is a leading provider of real-time push notification alerts and mobile data delivery. This document provides the necessary information to send push notifications via a REST API.

This document corresponds to v1 of the Push IO Public API.

Definitions

App ID

Each individual application has a unique ID used by the application to communicate with the Push IO service.

Sender Secret

This is a secret shared between Push IO and the developer using direct device messaging which authorizes them to send to an account.

Category

A category defines a specific content type that your app has which users might be interested in. For example, if your app is a sports app, a category may be a specific team.

Audience

An audience defines a group of users based on one or more categories. For example, you might make an audience out of all of the users who have registered to a category for a team in the Western Conference.

Test Device

A test device is a device which is contained in the Test Audience shown under the Set Up > Audiences section of your account at <https://manage.push.io>. This audience is also automatically populated with devices that have the Push IO Mobile Dashboard that you have used to sign in.

API Integration

Overview

Push IO provides a simple RESTful API with JSON responses.

The API offers methods for sending broadcast, targeted (category-based), and test device push notifications. Sending through the API allows you to target users on all of your platforms with a single call.

Push IO is the only push notification provider which provides data that allows you to understand how push notification engagement leads your users to high-value actions like in-app purchases, premium content viewing, and more. The PushIOManager library provides a simple mechanism to capture this push conversion information, which is available to you via the Push IO dashboard at <https://manage.push.io>.

API Prerequisites

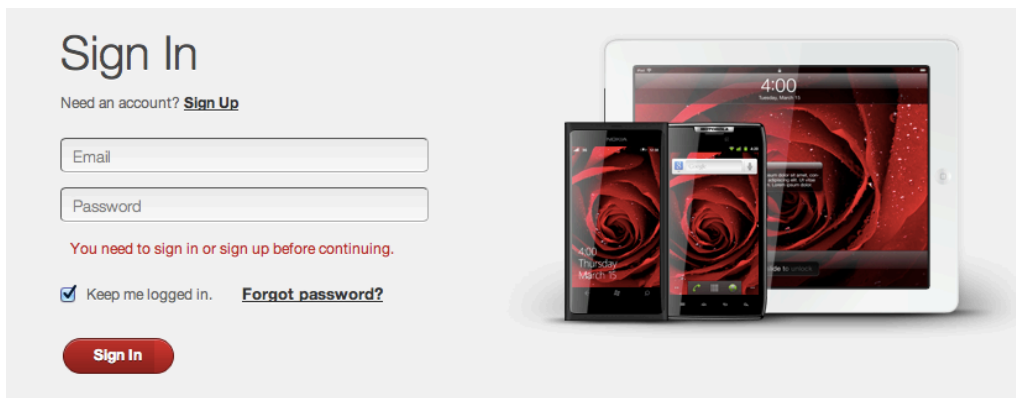
Before continuing, be sure you have what you need to integrate Push IO into your app.

1. Sign up for an account at <https://manage.push.io>
2. Complete PushIOManager integration for your app (see Push IO Integration Guide)

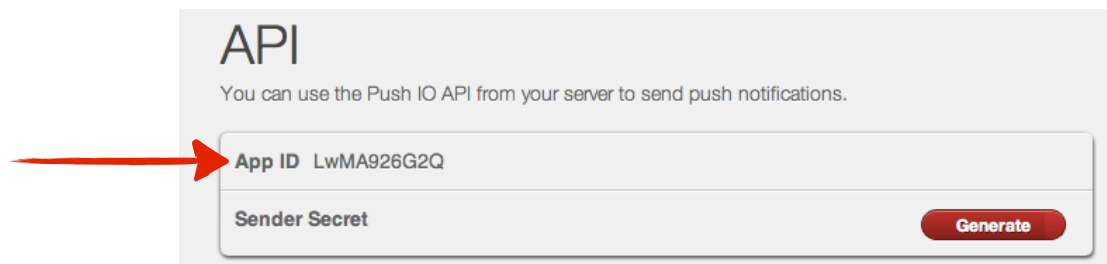
All set? Now you're ready to integrate with the Push IO API!

API Setup

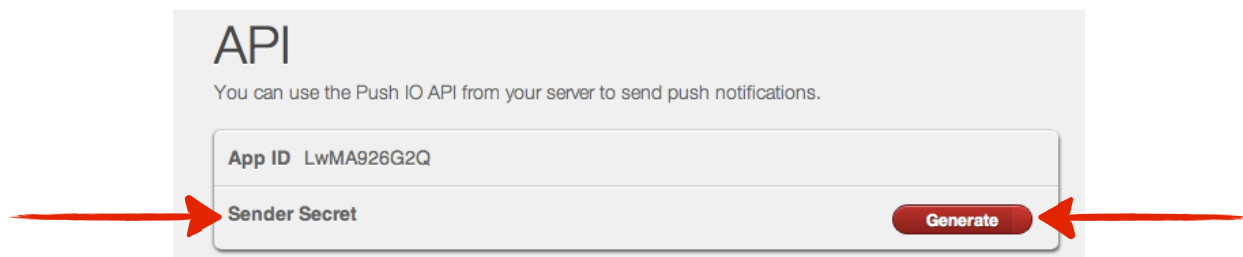
Step 1: Sign in to your Push IO dashboard (<https://manage.push.io>)



Step 2: Locate your App ID from Set Up > API on the Push IO dashboard.



Step 3: Navigate to Set Up > API from your Push IO dashboard to generate a *Service Secret* required to send to the Push IO REST API.



You may always re-generate your Service Secret from Set Up > API in order to maintain a high-level of security for sending to your app users.

Sample API Calls

All calls to production Push IO API commands are to : <https://manage.push.io>

Test Device Push

Send a message to your test devices.

```
POST api/v1/test_app/$YOUR APP ID/$YOUR SERVICE SECRET
```

POST Data

```
payload={"message": "Test Device Push"}
```

Broadcast Push

Send a message to all your users.

```
POST api/v1/notify_app/$YOUR APP ID/$YOUR SERVICE SECRET
```

POST Data

```
payload={"message": "Broadcast Push"}&tag_query=broadcast
```

Targeted Push

Send a message to all your users.

```
POST api/v1/notify_app/$YOUR APP ID/$YOUR SERVICE SECRET
```

POST Data

```
payload={"message": "Targeted Push"}&tag_query= $YOUR CATEGORY NAME
```


Advanced API Calls

Broadcast Push with Extra Data

Send a message to all your users, with the text “Hello All Users” with all other parameters set as platform defaults. In addition, the app will receive the JSON key value pair “key” : ”value”.

POST api/v1/notify_app/**\$YOUR APP ID/\$YOUR SERVICE SECRET**

POST Data

payload={"message": "Hello All Users", "extra" : {"key" : "value"}}
&tag_query=broadcast

Broadcast Push with Platform Specific Data

Send a message to all your users, with alternative data sent to your iOS app. Users on all platforms EXCEPT iOS will see the text “Hello All Users” and receive the JSON pair “key” : ”value”. Users on iOS will see the text “Hello iOS Users”, receive a badge number of 2 and the JSON pair “ioskey” : ”iosvalue”.

POST api/v1/notify_app/**\$YOUR APP ID/\$YOUR SERVICE SECRET**

POST Data

payload={"message": "Hello All Users", "extra" : {"key" : "value"}, "payload_apns" :
{ "alert" : "Hello iOS Users", "badge" : 2, "extra" : {"ioskey" : "iosvalue"}}}
&tag_query=broadcast

JSON Objects Descriptions

Push IO allows you to send data to your app using JSON key pairs. These pairs are contained within the payload parameter in your POST command. The following is a list of all the allowable pairs and a description.

Payload Object	
Parameter Name	Description
“message”	The default message for each platform. This message will be displayed as the notification text unless overridden at the platform payload level
“extra”	General JSON key/value pairs added as appropriate to each platform, unless overridden by platform-specific extra
“payload_apns”	A shipment_payload_apns JSON object (described below) which contains key/value pair which will apply only to iOS devices
“payload_c2dm”	A shipment_payload_c2dm JSON object (described below) which contains key/value pair which will apply only to notifications sent through C2DM
“payload_gcm”	A shipment_payload_gcm JSON object (described below) which contains key/value pair which will apply only to notifications sent through GCM
“payload_mpns”	A shipment_payload_mpns JSON object (described below) which contains key/value pair which will apply only to Windows Phones

payload_apns object	
Parameter	Description
“alert”	The text that will be displayed in the notification
“badge”	The number that will appear over the top left corner of an app's icon
“sound”	The name of the sound file to be played when the notification arrives. This file must be contained in your app's bundle
“extra”	iOS specific key/value pairs (provided in JSON) that are passed along with the 'aps' payload. If any value is provided for this it will override the entire payload-level extra values; i.e. it is not merged with the extra data in the parent platform object

payload_gcm object	
Parameter	Description
“alert”	The value of the data.alert field that is shown as the notification text on Android by the Push IO client library
“badge”	The value of the data.badge field that is sent to Android devices
“sound”	the value of the data.sound field that is sent to Android devices
“extra”	GCM-specific key/value pairs (provided in JSON) that are passed as children of the 'data' element of the payload. If any value is provided for this it will override the entire payload-level extra values; i.e. it is not merged with the extra data in the parent platform object
“collapse_key”	The GCM collapse key, used for send-to-sync messages, as described here: http://developer.android.com/guide/google/gcm/adv.html#collapsible
“delay_while_idle”	Boolean value, default false, which GCM servers use to determine if the message will be delivered immediately or when the device becomes active
“time_to_live”	Integer value determining how long (in seconds) the message should be kept for delay_while_idle situations, default 4 weeks

payload_mpnns object	
Parameter	Description
"toast_text1"	Text that displays in a bold typeface in one row along with text2, approx 40 characters fit if just text1 is shown
“toast_text2”	Text that displays in a roman typeface after toast_text1. This is where the global payload's 'message' parameter will be copied unless overridden. Approx 47 characters fit if only text2 is used, if text1 and text2 are evenly split, approx 41 characters fit
"extra"	Windows Phone specific key/value pairs (provided in JSON) that are passed along with the 'param' querystring-type value in a toast payload. If any value is provided for this it will override the entire payload-level extra values; i.e. it is not merged with the extra data in the parent platform object
"tile_id"	The id of the secondary tile to update, which is an exact navigation Uri of the secondary tile, ex: SecondaryTile.xaml?DefaultTitle=FromTile
"tile_title"	Title of the application tile, 15 characters max

payload_mpons object	
Parameter	Description
"tile_count"	An integer value from 1-99 shown on the front of the tile. This value is copied down from the global payload's 'badge' parameter unless overridden
tile_background_image	The image shown as the background on the front of the tile, full url, host must be authorized on device by app, must not be https, .jpg or .png image, 173x173px, 80k or less, must download within 30 seconds
"tile_back_title"	Like the tile_title but on the back of the tile, 15 characters max
"tile_back_background_image"	Full url of the image shown on the back of the tile, same rules as tile_background_image
"tile_back_content"	Text displayed at the top of the back tile, approx 40 characters fit
"props_to_clear"	Comma separated string of tile elements to be reset to defaults

payload_c2dm object	
Parameter	Description
"alert"	The value of the data.alert field that is shown as the notification text on Android by the Push IO client library
"badge"	The value of the data.badge field that is sent to Android devices
"sound"	the value of the data.sound field that is sent to Android devices
"extra"	C2DM-specific key/value pairs (provided in JSON) that are passed as children of the 'data' element of the payload. If any value is provided for this it will override the entire payload-level extra values; i.e. it is not merged with the extra data in the parent platform object
"collapse_key"	the C2DM collapse key, used for send-to-sync messages, as described here: http://developer.android.com/guide/google/gcm/adv.html#collapsible
"delay_while_idle"	Boolean value, default false, which C2DM servers use to determine if the message will be delivered immediately or when the device becomes active