

Push Notifications API

Adobe Digital Publishing Suite

18 December 2014

Copyright 2014–2015 Adobe Systems Incorporated. All rights reserved. This documentation contains material protected under U.S. Copyright law and international treaties and is provided under the terms of the Digital Publishing Suite Terms of Use. Except as expressly permitted herein, any unauthorized reproduction, transmission, or use of this material in any form or by any means is prohibited unless approved in writing from Adobe.

©2014–2015 Adobe Systems, Incorporated

Table of Contents

Version Notes	3
Setup	3
Getting Started.....	3
Request of IMS Client.....	3
Authentication Setup.....	3
Creation of the IMS User Token	5
Notification API	6
notifications.....	6
notificationStatus	9
notificationHistory	10
stopNotification.....	12
isNotificationDisabled.....	13
issueList	14
articleList.....	15
Error Codes for the Notification API.....	17
Certificates API	18
getCertificate	18
uploadIOSCertificate.....	19
getBundleInformation	20
Certificate Error Codes.....	21

Version Notes

This is the first version of the Push Notifications API. Sample API code is still under development.

Setup

Getting Started

The following is a list of necessary credentials/documentations utilized by the PUSH Notification API:

- Digital Publishing Suite's Login Credentials
- IMS Client (See Setup Below)
- Apple iOS Developer Certificate

Request of IMS Client

Every software-client that wants to consume the Adobe Push Notification API has to request an IMS ClientId (with corresponding secret key). This clientID can be requested through Adobe World Wide Developer Support (wwds@adobe.com).

Authentication Setup

The following headers are required for all web service calls.

Header	Value
Authorization	Bearer IMSToken
X-Api-Key	IMS Client Id

*Value of IMSToken to be replaced with token obtained during the IMS User Token creation phase.

Calling the /healthCheck API Example:

URL: <https://dps-notification.adobe.io/notification/healthCheck>

Method: GET

Headers:

Header:Authorization Value: Bearer
eyJhbGciOiJSUzI1NiIsIng1dSI6Imltc19uYTEtc3RnMS1rZXktMS5jZXIifQ.
eyJpZCI6IjE0MTQxNzk1OTY0NDktYTg1ZmU0NDgtM2NkMy00M2ViLWJi
ZTgtMmQ1YjVjODE0MWMzliwic2NvcGUiOiJvcGVuaWQsc2FvLmRpZ210
YWxfchVibGlzaGluZyIsImMiOiJwSGNEMTVua2gvODB6Skw2S0pabkNnP
T0iLCJhcyI6Imltcy1uYTEtc3RnMSIsImNyZWZ0ZWRFYXQiOiIxNDE0MTc
5NTk2NDQxliwiZXhwaXJlc19pbiI6Ijg2NDAwMDAwliwidXNlcl9pZCI6Ik
FEN0ExRkM1NTMxMEM1OTYwQTQ5MEQzNUBBZG9iZUIEliwiY2xpZW5

0X2lkIjoiZHBzX3Jwc19pbXMiLCJ0eXBliIjoiYWNjZXNzX3Rva2VuIn0.dKM
qoHh5z8IgrCFitf4iejKFH9EnFUIGWFCz99gCCU2ooxvEs0mWBIBuKtwv
audWSw1Eaj7057MPtkMl8W-
oHKiXvYUu2jRpDQKjese2Ilmwh6KOFgoCysFBaxfMhBxXec-
3WAkMX90fTtlIfYxqNBwEOUOA-
oUlgCR6JGpJ24nDQmcYsrd4mM_m00xK0R6AKTihRkbJICGDwCW678X
Qh1N50aF7jgFTNjBeJGYy08VpTho9dabrsYwJ1jGiwHX3a4Wofa8_XfqCS
3j92IeKyGI9BAGVY8aUy70_aSSbkE_eQem9KXdHgCJGn9E_6pSuXGuBwg
VW2BXgcA1D7oiqQ

Header: X-Api-Key Value: dps_rps_ims

Creation of the IMS User Token

The IMS User token can be created by making call to the following IMS API.

Method: GET

End Point: <https://ims-na1-cc1.adobelogin.com>

Path: /ims/login/v1/token

Query Parameter	Value	Description
client_id		The IMS Client ID, see Setup
client_secret		The IMS Client Secret, see Setup
scope	openid,sao.digital_publishing`	Please use the given value for the scope
username		The DPS username
password		The DPS password

W

Example:

Call:

https://ims-na1-stg1.adobelogin.com/ims/login/v1/token?client_id=dps_rps_ims&client_secret=eed7218f-0baf-400e-83b3-4ecabe4c33f7&scope=openid%2Csao.digital_publishing&username=winglis%2Bstage%2Bpush2%40adobetest.com&password=tester

Response:

```
{"countryCode":"US","mrktPermEmail":null,"expires_in":"86399989","serviceAccounts":[{"ident":"AD7A1FC55310C5960A490D35","serviceCode":"digital_publishing","serviceStatus":"ACTIVE","serviceLevel":"PAID_LVL_2","ownerGuid":"AD7A1FC55310C5960A490D35","ownerAuthSrc":"WCD","ownerAcctLabel":null,"delegateGuid":"AD7A1FC55310C5960A490D35","delegateAuthSrc":"WCD","delegateAcctLabel":null,"serviceUrl":null,"createDts":1393608087000,"modDts":1393608087000,"params":[{"pn":"storage_quota","pv":"9999"}]},{"emailVerified":"true","first_name":"Stage Push2","phoneNumber":null,"email":"winglis+stage+push2@adobetest.com","preferred_languages":null,"userId":"AD7A1FC55310C5960A490D35@AdobeID","name":"Stage Push2 Push Notification Test","last_name":"Push Notification Test","utcOffset":"null","token_type":"bearer","mrktPerm":"","displayName":"Stage Push2 Push Notification Test","access_token":"eyJhbGciOiJIUzI1NiIsInR5cGU6IjY0dGE0MmMzIiwiaWF0IjoiMTYyOTY0NDU0IiwiaXNjaWkiOiJvcGVuaWQsc2FvLmRwZ2I0YWxhcGVibGlzaGluZyIsImMiOiJwSGNEMTVua2gvODB6Skw2S0pabkNnPT0i"}]}
```

LCJhcyI6Imltcy1uYTEtc3RnMSIsImNyZWFOZWRFYXQiOiIxNDE0MTc5NTk2NDQxliwiZXhwaXJlc19pbil6Ijg2NDAwMDAwliwidXNlcl9pZCI6IkFEN0ExRkM1NTMxMEM1OTYwQTQ5MEQzNUBBZG9iZUIEliwiY2xpZW50X2lkIjoizHBzX3Jwc19pbXMiLCJ0eXBlljoiYWNjZXNzX3Rva2VuIn0.dKMqoHh5z8IgrCfitf4iejKFH9EnFUiGWFuCu99gCCU2ooxvEs0mWBIBuKtwvaudWSw1Eaj7O57MPtkMl8W-oHKiXvYUu2jRpDQKjese2IlmwH6KOFgoCysFBaxfMhBxXec-3WAkMX90fTtlIfYxqNBwEOUOA-oUlgCR6JGpJ24nDQmcYsrd4mM_m00xK0R6AKTihRkBJICGDwCW678XQh1N5OaF7jgFTNjBeJGYyO8VpTho9dabrsYwJ1jGiwHX3a4WOfa8_XfqCS3j92IeKyGI9BAGVY8aUy70_aSSbkE_eQem9KXdHgCJGn9E_6pSuXGuBwgVW2BXgcA1D7oiqQ"} }

Error Response:

```
{ "error": "invalid_client" }           The client ID does not exist.
{ "error": "invalid_scope" }         The scope does not match
"openid,sao.digital_publishing", please make sure the URL is encoded.
{ "error": "invalid_credentials" }   The DPS credentials do not match.
```

Notification API

notifications

Send a notification.

Method: POST

End Point: <https://dps-notification.adobe.io/notification>

Path: /notifications

Consumes: multipart-form/data

Parameter	Parameter Type	Required	Description
deviceFile	Form Data	No	If the request includes a csv file of devices to send the notification to, this should be set
push	Form Data	Yes	A JSON parameter that includes information about the notification. See Below
Return	Type		
StatusResponse	application/xml, application/json		Returns the status of the passed in notification

JSON for push

```
{
  "push": {
    "schedule": "",
    "devices": "",
    "message": "",
    "issueIdForBackgroundDownload": "",
    "soundId": "-1",
    "badge": false,
    "isProduction": false,
    "segmentDef": "",
    "issueName": "",
    "articleName": "",
    "applicationId": ""
    "backgroundMessage": "",
    "remindInHours": "",
  }
}
```

push.schedule - the scheduled sending time for the notification in milliseconds since the epoch. If less than "now" the notification will be sent immediately.

push.devices - If the request includes a csv list of devices to send the notification too, this will include that list. If the list is not set or empty, all devices associated with the applicationId are sent.

push.message - The text to send with the notification. If this value is set, the notification will be a text-only notification.

push.issueIdForBackgroundDownload - The issueId to send the background download notification to.

push.soundId - The name of the sound to play when the notification is received (it must be in application bundle and in the proper format). "default" will play the default notification sound

push.badge - A boolean whether to show the badge on the application icon.

push.isProduction - A boolean. True means to send the notification to production devices, false to sandbox (development) devices.

push.segmentDef - The segment of devices to send the text notification to (only valid for text notifications, not background notifications).

Segment JSON Examples:

Users can't define their own segments. These are the only available segments.

```
{ "name": "DevicesBasedOnIpadIphone", "ipadOrIphone": "ipad" }  
{ "name": "DevicesBasedOnIpadIphone", "ipadOrIphone": "iphone" }  
{ "name": "DevicesNotLaunchedFor", "daysSinceLastLaunch": 30 }
```

push.issueName – String of the name of the folio to which the deep link is directed. This is the FolioName in FolioProducer.

push.articleName – String of the name of the article to which the deep link is directed.

IssueName and ArticleName reserved keywords:

- issueName = "customSlot", then the articleName param will be the name of the customSlot where we take the client after they tap on the notification.
- issueName = "library" and articleName = "library", then we will take the client to the library after they tap on the notification.
- issueName = "arbitrary", then the articleName param will be the arbitrary url where we take the client after they tap on the notification. Only http:// and https:// are allowed.

Any other values for issueName and articleName will be the issueName and articleName in the specified folio where we take the client after they tap on the notification. Rules regarding entitlement and previewing will be enforced.

push.applicationId - A String specifying which applicationID to send the notification to (also referred to as bundleId or applicationId).

push.backgroundMessage - This is the message that is sent along with background notifications. When background download fails for non-entitled users, a local text notification will be shown with this message.

push.remindInHours - On iOS 8 devices, users can ask to be reminded after a few hours/days. This value is configured using this property. This value has to be either hours or days. So anything more than 24hours is converted to days. Examples: "remindInHours":30 is rounded down to the closest day - 1 day.

Sample Response

Xml status response


```

<statusResponse>
  <errorCode>7</errorCode>
  <notificationStatus>cancelled</notificationStatus>
  <statusCode>5</statusCode>
  <schedule>1390863115184</schedule>
  <completeTime>1392790382866</completeTime>
  <accountId>78a7abae-f314-55ab-a499-a82b4d498205</accountId>
  <issueIdForBackgroundDownloads></issueIdForBackgroundDownloads>
  <message>The service is working. Yay!</message>
  <backgroundMessage></ backgroundMessage>
  <sound>0</sound>
  <isProduction>>true</isProduction>
  <badge>>false</badge>
  <totalChunks>262</totalChunks>
  <chunksCompleted>9</chunksCompleted>
  <totalDevices>521500</totalDevices>
  <totalSent>0</totalSent>
  <totalFailed>0</totalFailed>
  <notificationId>1ccc3867225845b19c3c34d69d61b374</notificationId>
  <productId></productId>
  <deeplinkingURL></ deeplinkingURL>
</statusResponse>

```

notificationStatus

Get the status of a given notification.

Method: GET

End Point: <https://dps-notification.adobe.io/notification>

Path: /notifications/{notificationId}

Parameter	Parameter Type	Required	Description
notificationId	Path Parameter	Yes	UUID identifying the previously requested notification
version	Query Parameter	No	APIVersion - unused - defaults to "1.0"
Return	Type		
StatusResponse	application/xml, application/json		Returns the status of the passed in notification

Sample Response

Xml statusResponse

```

<statusResponse>
  <errorCode>0</errorCode>
  <notificationStatus>all_sent</notificationStatus>
  <statusCode>3</statusCode>
  <schedule>1396539135173</schedule>
  <completeTime>1396539143266</completeTime>
  <accountId>63fb6ce6-c26f-49ad-bf4e-8611d3449f60</accountId>
  <issueIdForBackgroundDownloads></issueIdForBackgroundDownloads>
  <message>It is working - Yeah</message>
  <backgroundMessage></backgroundMessage>
  <sound>-1</sound>
  <isProduction>true</isProduction>
  <badge>false</badge>
  <totalChunks>1</totalChunks>
  <chunksCompleted>1</chunksCompleted>
  <totalDevices>2</totalDevices>
  <totalSent>2</totalSent>
  <totalFailed>0</totalFailed>
  <notificationId>3abebd9dd82a4571a672dd30850da617</notificationId>
  <productId></productId>
  <deeplinkingURL></deeplinkingURL>
</statusResponse>

```

notificationHistory

Get the history of sent notifications between two dates

Method: GET

End Point: <https://dps-notification.adobe.io/notification>

Path: /notifications

Parameter	Parameter Type	Required	Description
historyFromDate	Query Parameter	No	Begin date to gather notification history from - seconds from the epoch format. Null or missing means start at the very first notification sent

historyToDate	Query Parameter	No	End date to gather notification history until - seconds from the epoch format. 0 or missing means until now.
version	Query Parameter	No	APIVersion - unused - defaults to "1.0"
Return	Type		
HistoryResponse	application/xml, application/json		Returns the notification history of this account.

Sample Response

Xml history response

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<historyResponse>
  <isValidAccount>true</isValidAccount>
  <pushHistory>
    <SuccessfulHistoryRetrieval>false</SuccessfulHistoryRetrieval>
    <pushItems>
      <TimeNotificationInitiated>0</TimeNotificationInitiated>
      <NumberOfDevicesNotificationwasSentto>94568</NumberOfDevicesNotificati
onwasSentto>
      <NumberOfDevicesWhereNotificationFailed>65</NumberOfDevicesWhereNoti
ficationFailed>
      <NotificationText>Get 50% off on the next issue!</NotificationText>
      <backgroundMessage></backgroundMessage>
      <isBackground>false</isBackground>
      <notificationStatus>4</notificationStatus>
      <errorCode>0</errorCode>
    </pushItems>
  </pushHistory>
</historyResponse>
```

stopNotification

Cancel a previously requested notification. May or may not be possible.

Method: DELETE

End Point: <https://dps-notification.adobe.io/notification>

Path: /notifications/{notificationId}

Parameter	Parameter Type	Required	Description
notificationId	Path Parameter	Yes	UUID identifying the notification user wishes to cancel
Return	Type		
StatusResponse	application/xml, application/json		Contains status of this notification after attempting to cancel

Sample Response

Xml statusResponse

```
<statusResponse>
  <errorCode>7</errorCode>
  <notificationStatus>cancelled</notificationStatus>
  <statusCode>5</statusCode>
  <schedule>1390863115184</schedule>
  <completeTime>1392790382866</completeTime>
  <accountId>78a7abae-f314-55ab-a499-a82b4d498205</accountId>
  <issueIdForBackgroundDownloads></issueIdForBackgroundDownloads>
  <message>The service is working. Yeah!</message>
  <backgroundMessage></ backgroundMessage>
  <sound>0</sound>
  <isProduction>>true</isProduction>
  <badge>>false</badge>
  <totalChunks>262</totalChunks>
  <chunksCompleted>9</chunksCompleted>
  <totalDevices>521500</totalDevices>
  <totalSent>0</totalSent>
  <totalFailed>0</totalFailed>
  <notificationId>1ccc3867225845b19c3c34d69d61b374</notificationId>
  <productId></productId>
  <deeplinkingURL></ deeplinkingURL>
</statusResponse>
```

isNotificationDisabled

Get the status of our back-end notification system. This is to check if the full notification service system is working.

Method: GET

End Point: <https://dps-notification.adobe.io/notification>

Path: /healthCheck

Return	Type	
IsDisabledResponse	application/xml, application/json	Returns true if the notification back-end is working.

Sample Response

Xml isDisabled response

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<isDisabledResponse>
```

```
<isNotificationDisabled>>false</isNotificationDisabled>
</isDisabledResponse>
```

issueList

Get the issue list for a given account id.

Method: GET

End Point: <https://dps-notification.adobe.io/notification>

Path: /issues

Parameter	Parameter Type	Required	Description
version	Query Parameter	No	APIVersion - unused - defaults to "1.0"
Return	Type		
IssueListResponse	application/xml, application/json		Contains the list of issues for this account

Sample Response

If unauthorized, following response is sent with HTTP Status code of 401.

```
<?xml version="1.0" encoding="UTF-8" standalone="yes" ?>
<response>
<status>
  UNAUTHORIZEDERROR
</status>
<message>
  AMS Denied Access
</message>
</response>
```

If authorized, following response is sent.

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<issueListResponse>
  <isValidAccount>true</isValidAccount>
  <issueList productId="yobaby" id="123456789a">
    <magazineTitle>Push Notification 3</magazineTitle>
    <issueNumber>1024 x 768 r27</issueNumber>
    <publicationDate>2013-11-26T08:00:00Z</publicationDate>
    <description>This is a sample description. </description>
```

```

<manifestXRef>Push iPad2</manifestXRef>
<state>production</state>
<libraryPreviewUrl>http://xxx.adobe.com/etc/{issueId}/libraryPreview</librar
yPreviewUrl>
<targetDimensions>
  <targetDimension>1024x768</targetDimension>
</targetDimensions>
</issueList>
<issueList productId="new" id="123456789a">
  <magazineTitle>Push Notification 1</magazineTitle>
  <issueNumber>1024 x 768 r26</issueNumber>
  <publicationDate>2013-06-07T07:00:00Z</publicationDate>
  <description>General stuff x4Jim2 Shared from e5e </description>
  <manifestXRef>Push iPad</manifestXRef>
  <state>production</state>
  <libraryPreviewUrl>http://xxx.adobe.com/etc/{issueId}/libraryPreview</librar
yPreviewUrl>
  <targetDimensions>
    <targetDimension>1024x768</targetDimension>
  </targetDimensions>
</issueList>
</issueListResponse>

```

articleList

Get article list for a given issueId.

Method: GET

End Point: <https://dps-notification.adobe.io/notification>

Path: /articles/{issueId}/{issueVersion}

Parameter	Parameter Type	Required	Description
issueId	Path Parameter	Yes	Id of the issue you want the article list for
issueVersion	Path Parameter	No	Version of the articles (currently unused)

version	Query Parameter	No	APIVersion - unused – defaults to “1.0”
Return	Type		
ArticleListResponse	application/xml, application/json		Contains the list of articles for this issue

Sample Response

If unauthorized, following response is sent with HTTP Status code of 401.

```
<?xml version="1.0" encoding="UTF-8" standalone="yes" ?>
<response>
<status>
  UNAUTHORIZEDERROR
</status>
<message>
  AMS Denied Access
</message>
</response>
```

If authorized, following response is sent.

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<articleListResponse>
  <isValidAccount>true</isValidAccount>
  <articleIssue id="123456789a " downloadSize="1428537" version="4"
manifestVersion="2" state="production" broker="noChargeStore"
uberbundleMD5="123456789a " uberbundleSize="1440181">
  <articles>
    <article id="123456789a " downloadSize="358461" downloadPriority="low"
access="closed" manifestXref="autodown2" uncompressedSize="392752"
md5="123456789a " version="1" targetViewer="27.0.0"/>
    <article id="123456789a " downloadSize="356723" downloadPriority="low"
access="closed" manifestXref="cosmo1" uncompressedSize="390664"
md5="123456789a " version="1" targetViewer="27.0.0"/>
    <article id="123456789a " downloadSize="355573" downloadPriority="low"
access="closed" manifestXref="arts2" uncompressedSize="389466"
md5="123456789a " version="1" targetViewer="27.0.0"/>
    <article id="123456789a " downloadSize="357780" downloadPriority="low"
access="closed" manifestXref="autodown1" uncompressedSize="392146"
```



```

md5="123456789a " version="1" targetViewer="27.0.0"/>
  </articles>
  </articleIssue>
</articleListResponse>

```

Error Codes for the Notification API

NO_ERROR		0
UNKNOWN_ERROR		1
INVALID_BACKGROUND	Can't have a background notification with a text message. Remove the text message	2
INVALID_BADGE	If message has a badge, it must have a background notification	3
INVALID_MESSAGE	No text message or background download	4
INVALID_BACK_SOUND	Background download with sound isn't allowed	5
CANNOT_GET_DEVICES	Can't get list of device based on account	6
CERTIFICATE_NOT_FOUND	Unable to retrieve the proper push notification	7
CERTIFICATE_EXPIRED	Push certificate has expired	8
INVALID_NOTIFICATION_ID	NotificationID passed in to status or cancel API's invalid	9
UNAUTHORIZED		10
MISSING_BACKGROUND	Error when trying to find background issue	11
NO_DEVICES_IN_FILE	Error when client uploaded file contains no valid tokens	12
NO_TARGET_DIMENSIONS	Background download issue has not dimensions. No devices will be considered	13
NO_VALID_DEVICES	There were no valid devices found for this request. No action was needed	14
INVALID_SEGMENT_DEFINITION	The supplied segment definition JSON was invalid	15
DEVICE_FILE_TOO_LARGE	The supplied file of devices was too large	16
CERTIFICATE_REVOKED	Push certificate was revoked	17
PAYLOAD_SIZE_EXCEEDED	Payload exceeds the permitted max size	18
PAYLOAD_EMPTY	Payload is empty	19
DOWNSTREAM_SERVER_ERROR	Unable to contact downstream server	20

Certificates API

This allows clients to retrieve existing Push Certificates and to upload new ones into the Certificate database.

getCertificate

Get information about a pair of Apple iOS Certificate.

Method: GET

End Point: <https://dps-notification.adobe.io/notification>

Path: /certificate/{certificateType}/{certificateId}

Parameter	Parameter Type	Required	Description
certificateType	Path Parameter	Yes	Specifies which certificate to retrieve: <ul style="list-style-type: none">"iOS" for Apple certificates. (Android certificates not supported)
certificateId	Path Parameter	Yes	Specifies which application certificate to retrieve: <ul style="list-style-type: none">"bundleId" for iOS
Return	Type		
CertificateResponse	application/xml, application/json		Details about a found certificate or an error code.

Sample Response

xml response - Apple certificate

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<certificateResponse>
  <appleCertificate>
    <accountId>78a7abae-f314-55ab-a499-a82b4d498205</accountId>
    <bundleId>com.thirstybear.app.c</bundleId>
    <productionExpirationDate>2015-01-31T09:55:31-
08:00</productionExpirationDate>
```

```

        <sandboxExpirationDate>2015-01-31T09:53:40-
08:00</sandboxExpirationDate>
    </appleCertificate>
    <error>0</error>
</certificateResponse>

```

Error response

```

<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<certificateResponse>
    <error>17</error>
</certificateResponse>

```

uploadIOSCertificate

Upload a pair of iOS certificates. Both Production and Sandbox push certificates with the same bundleId/applicationId must be uploaded at the same time.

Method: POST

End Point: <https://dps-notification.adobe.io/notification>

Path: /certificate/uploadIOSCertificate

Consumes: multipart-form/data

Parameter	Parameter Type	Required	Description
certificate	Form Data Parameter	Yes	A JSON parameter that includes information about the Apple certificates to be uploaded. See below.
iOSProductionCertificateFile	Form Data Parameter	Yes	The multipart file data for the .p12 production (distribution) certificate.
iOSSandboxCertificateFile	Form Data Parameter	Yes	The multipart file data for the .p12 development (sandbox) certificate.
Result	Type		
UploadResponse	application/xml, application/json		returns the status of the uploaded certificates. Can contain multiple errors.

JSON for certificate

```
{
  "certificate": {
    "iOSBundleId": "",
    "iOSSandboxCertificatePassword": "",
    "iOSProductionCertificatePassword": ""
  }
}
```

certificate.iOSBundleId - the bundleId, or applicationId to be associated with this certificate.

certificate.iOSProductionCertificatePassword - The password of the production certificate.

certificate.iOSSandboxCertificatePassword - The password of the sandbox certificate.

Sample Response

xml response - certificate upload success

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<uploadResponse>
  <uploadSucceeded>true</uploadSucceeded>
</uploadResponse>
```

xml response - certificate upload failure

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<uploadResponse>
  <uploadSucceeded>>false</uploadSucceeded>
  <certificateErrorList>9</certificateErrorList>
  <certificateErrorList>8</certificateErrorList>
</uploadResponse>
```

getBundleInformation

Get information about a bundle.

Method: GET

End Point: <https://dps-notification.adobe.io/notification>

Path: /certificate/bundles

Result	Type	Description
--------	------	-------------

BundleInfoList	application/json	Returns bundle information
----------------	------------------	----------------------------

Sample Response

If unauthorized, following response is sent with HTTP Status code of 401.

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
  <response>
    <status>UNAUTHORIZEDERROR</status>
    <message> AMS Denied Access</message>
  </response>
```

If authorized, following response is sent.

```
{
  "bundles": [
    {
      "appType": "APPLE",
      "appId": "com.thirstybear.app.c",
      "apiKey": "",
      "certExpiration": {
        "dist": 0,
        "dev": 0
      },
      "appName": [
        "John's Million App"
      ]
    }
  ]
}
```

Certificate Error Codes

NO_ERROR	0
UNKNOWN_ERROR	1
INVALID_SANDBOX_CERT_OR_PASSWORD	2
INVALID_PRODUCTION_CERT_OR_PASSWORD	3
EXPIRED_SANDBOX_CERT	4
EXPIRED_PRODUCTION_CERT	5
INVALID_SANDBOX_PASSWORD	6
INVALID_PRODUCTION_PASSWORD	7
BUNDLE_ID_INVALID	8
BAD_SANDBOX_BUNDLE_ID	9
BAD_PRODUCTION_BUNDLE_ID	10
ANDROID_PROJECT_NUMBER_INVALID	11
INVALID_API_KEY	12
INVALID_CERTIFICATE_TYPE	13
IO_ERROR	14

CERTIFICATE_ERROR	15
SERVER_ERROR	16
CERTIFICATE_NOT_FOUND	17
PRODUCTION_CERT_IS_SANDBOX	18
SANDBOX_CERT_IS_PRODUCTION	19