What’s New in Apple Push Notification Service

Session 724

Mayur Mahajan Apple Push Notification Service Engineer
Review

- HTTP/2 provider API
- Instant feedback
- Larger payload
- Simplified certificate handling
HTTP/2 Provider API

Sending notifications
HTTP/2 Provider API

Sending notifications

- APNs
- Provider
- Client App
HTTP/2 Provider API

Sending notifications

APNs

Provider

Client App
HTTP/2 Provider API

Sending notifications

APNs

Provider

Client App
HTTP/2 Provider API

Sending notifications

APNs

Provider

Client App
HTTP/2 Provider API

Sending notifications

APNs

Provider

Client App

HTTP/2 request
HTTP/2 Provider API

Sending notifications

- APNs
- Provider
- Client App

HTTP/2 request

Diagram showing the flow of data between APNs, Provider, and Client App.
Simplified Certificate Handling

Application push
VoIP push
Complication push
Development and production environment
Token Authentication
Token Authentication
Token Authentication

Simplified provider authentication when connecting to APNs
Secure
Easy to generate
No more expired certificates!
Certificate Authentication

Connecting to APNs with client certificate

APNs

Provider

Account
Certificate Authentication

Connecting to APNs with client certificate

- APNs
- Provider
- Account
Certificate Authentication

Connecting to APNs with client certificate
Certificate Authentication

Connecting to APNs with client certificate

- APNs
- Provider
- Account

TLS
Token Authentication

Connecting to APNS with Token Authentication

APNs

Provider

Account
Token Authentication

Connecting to APNS with Token Authentication

- APNs
- Provider
- Account
Token Authentication

Connecting to APNS with Token Authentication
Token Authentication
Connecting to APNS with Token Authentication

APNs -> Provider
TLS
Provider -> Account
Token Authentication
Connecting to APNS with Token Authentication
Token Authentication

APNs  TLS  Provider
Token Authentication

APNs

Request with token

TLS

Provider
Token Authentication

APNs

Provider

TLS
Token Authentication

APNs

Response

TLS

Provider
Token Authentication

Details

Provision signing key via Developer Account

- **App Store and Ad Hoc**
  Sign your iOS app for submission to the App Store or for Ad Hoc distribution.

- **Apple Push Notification Service Key (Sandbox & Production)**
  Generate an authentication key that can be used to sign your push notification requests.

- **Apple Push Notification service SSL (Sandbox & Production)**
  Establish connectivity between your notification server, the Apple Push Notification service sandbox, and production environments to deliver remote notifications to your app. When utilizing HTTP/2, the same certificate can be used to deliver app notifications, update ClockKit complication data, and alert background VoIP apps of incoming activity. A separate certificate is required for each app you distribute.
Token Authentication

Details

Provision signing key via Developer Account

- **App Store and Ad Hoc**
  Sign your iOS app for submission to the App Store or for Ad Hoc distribution.

- **Apple Push Notification Service Key (Sandbox & Production)**
  Generate an authentication key that can be used to sign your push notification requests.

- **Apple Push Notification service SSL (Sandbox & Production)**
  Establish connectivity between your notification server, the Apple Push Notification service sandbox, and production environments to deliver remote notifications to your app. When utilizing HTTP/2, the same certificate can be used to deliver app notifications, update ClockKit complication data, and alert background VoIP apps of incoming activity. A separate certificate is required for each app you distribute.
Token Authentication

Details

Provision signing key via Developer Account

Construct the token

Sign token using ES256 (ECDSA using P256 curve, SHA256 hash)
// Token Authentication
// JSON Web Tokens

eyJhbGciOiJFUzI1NiIsImtpZCI6IkFCQ0RFRkdISUoiLCJpc3MiOiIxMjM0NTY3ODkwIiwiaWF0IjoiMTQ2Mzk3NTIzMyJ9.aBup1REyXE9Po8ekWJ3ifUly67NlPyTlcgfSwy...

Header
{
  "alg": "ES256",
  "kid": "ABCDEFGHIJ"
}

Claims
{
  "iss": "QRSTUVWXYZ",
  "iat": "1463975233"
}

Signature
// Token Authentication
// JSON Web Tokens

eyJhbGciOiJFUzI1NiIsImtpZCI6IkFCQ0RFRkdISUoiLCJpc3MiOiIxMjM0NTY3ODkwIiwiaWF0Ijoi
MTQ2Mzk3NTIzMyJ9.aBup1REyXE9Po8ekWJ3ifUly67NlPyTlcgfSwy...

Header
{
  "alg": "ES256",
  "kid": "ABCDEFGHIJ"
}

Claims
{
  "iss": "QRSTUWXYZ",
  "iat": "1463975233"
}

Signature
// Token Authentication
// JSON Web Tokens

eyJhbGciOiJFUzI1NiIsImtpZCI6IkFCQ0RFRkdISUoifQ.eyJpc3MiOiIxMjM0NTY3ODkwIiwiaWF0IjoiMTQ2Mzk3NTIzMyJ9.aBup1REyXE9Po8ekWJ3ifUly67NlPyTlcgfSwy...

Header
{
  "alg": "ES256",
  "kid": "ABCDEFGHIJ"
}

Claims
{
  "iss": "QRSTUWXYZ",
  "iat": "1463975233"
}

Signature
// Token Authentication
// JSON Web Tokens

eyJhbGciOiJFUzI1NiIsImtpZCI6IkFCQ0RFRkdISUoiLCJpc3MiOiIxMjM0NTY3ODkwIiwiaWF0IjoiMTQ2Mzk3NTIzMyJ9.aBup1REyXE9Po8ekWJ3ifUly67NlPyTlgfSwy...

Header
{
  "alg": "ES256",
  "kid": "ABCDEFGHIJKLMNOPQRSTUVWXYZ"
}

Claims
{
  "iss": "QRSTUWVXYZ",
  "iat": "1463975233"
}

Signature
// Token Authentication
// JSON Web Tokens

eyJhbGciOiJFUzI1NiIsImtpZCI6IkFCQ0RFRkdISUoiLCeyJpc3MiOiIxMjM0NTY3ODkwIiwiaWF0IjoiMTQ2Mzk3NTIzMyJ9.aBup1REyXE9Po8ekWJ3ifUly67NlPyTlcgfSwy...

Header
{
    "alg": "ES256",
    "kid": "ABCDEFGHIJ"
}

Claims
{
    "iss": "QRSTUVWXYZ",
    "iat": "1463975233"
}

Signature
// Token Authentication
// JSON Web Tokens

eyJhbGciOiJFUzI1NiIsImtpZCI6IkJFCQ0RFRkdISUoiLCJpYiI6IkJFCQ0RFRkdISUoiLCJpc3MiOiIxMjM0NTY3ODkwIiwiaWF0IjoiMTQ2Mzk3NTIzMyJ9.aBup1REyXE9Po8ekWJ3ifUly67NlPyTlcgfSwy...

Header
{
    "alg": "ES256",
    "kid": "ABCDEFGHIJ"
}

Claims
{
    "iss": "QRSTUvwxyz",
    "iat": "1463975233"
}

Signature
// Token Authentication
// JSON Web Tokens

eyJhbGciOiJFUzI1NiIsImtpZCI6IkJCQ0RFRkdISUoiLCeyJpc3MiOiIxMjM0NTY3ODkwIiwiaWF0Ijoi
MTQ2Mzk3NTIzMyJ9.aBup1REyXE9Po8ekWJ3ifUly67NlPyTlgfSwy...

Header
{
    "alg": "ES256",
    "kid": "ABCDEFGHIJ"
}

Claims
{
    "iss": "QRSTU VWXYZ",
    "iat": "1463975233"
}

Signature
// Token Authentication

// Request example

HEADERS
- END_STREAM
+ END_HEADERS
:authority = api.push.apple.com
:method = POST
:path = /3/device/ad2bcd38f6773cdad050411...
authorization = bearer eyJhbGciOiJFUzI1N...
apns-topic = com.foo.tokenauth

DATA
+ END_STREAM

{ "aps" : { "alert" : "Hello Token Authentication" } }
// Token Authentication
// Request example

HEADERS
- END_STREAM
+ END_HEADERS
:authority = api.push.apple.com
:method = POST
:path = /3/device/ad2bcd38f6773cdad050411...
authorization = bearer eyJhbGciOiJFUzI1N...
apns-topic = com.foo.tokenauth

DATA
+ END_STREAM
{
  "aps": {
    "alert": "Hello Token Authentication"
  }
}
// Token Authentication
// Response example

HEADERS
+ END_STREAM
+ END_HEADERS
:status = 200
// Token Authentication
// Response example

HEADERS
- END_STREAM
+ END_HEADERS
:status = 403
content-type = application/json
DATA
+ END_STREAM
{
  "reason": "InvalidProviderToken"
}
// Token Authentication
// Response example

HEADERS
- END_STREAM
+ END_HEADERS
:status = 403
content-type = application/json

DATA
+ END_STREAM
{
  "reason": "ExpiredProviderToken"
}
Token Authentication
Details

Signed tokens need to be generated periodically
Signing key does not expire
Signing key can be revoked through your Account
Coming later this year...
More Information

https://developer.apple.com/wwdc16/724
## Related Sessions

<table>
<thead>
<tr>
<th>Session</th>
<th>Location</th>
<th>Date</th>
</tr>
</thead>
<tbody>
<tr>
<td>Introduction to Notifications</td>
<td>Pacific Heights</td>
<td>Wednesday 9:00AM</td>
</tr>
<tr>
<td>Advanced Notifications</td>
<td>Pacific Heights</td>
<td>Wednesday 10:00AM</td>
</tr>
<tr>
<td>Lab Name</td>
<td>Other Lab Name</td>
<td>Time</td>
</tr>
<tr>
<td>--------------------------------</td>
<td>-------------------------------------</td>
<td>---------------</td>
</tr>
<tr>
<td>Notifications Lab</td>
<td>Frameworks Lab C</td>
<td>Wednesday 11:00AM</td>
</tr>
<tr>
<td>Notifications Lab</td>
<td>Graphics, Games &amp; Media Lab B</td>
<td>Friday 9:00AM</td>
</tr>
</tbody>
</table>