

## Android Client Setup

1. Goto <https://console.firebase.google.com/>
  - 1.1 Add Project
  - 1.2. Setup Cloud Messaging
  - 1.3 Download [google-services.json](#)
  - 1.4 Find **FCM\_SERVER\_API\_KEY**  
from Project Settings > Cloud Messaging,  
and copy this for later use (for sending push from your server)
2. Copy [google-services.json](#) to `native/android/google-services.json`

## UWP Client Setup

1. Register app in Windows Store.  
<https://developer.microsoft.com>
2. Setup [WNS/MPNS notifications](#) for app in Win Store Dashboard
3. Find the **Package SID** (e.g `ms-app://xxxx`), and **Client Secret** for later use (for sending push from your server).

## iOS Client Setup

1. Run [iOS Certificate Wizard](#)
  - 1.1 Check "Include Push" check box.
2. Use certificate URL and passwords you receive via automated email in order to send push from your server

## Javascript Client Setup

Chrome: Add [gcm.sender\\_id](#) build hint

Firefox, MS Edge, Opera: No setup required

Safari: Not supported yet

Must serve app using <https://>

## Implementing Push in Client

```
public class MyApplication implements PushCallback {
    public void start() {
        // Register device to receive push
        Display.getInstance().registerPush();
        // The rest of your start() logic....
    }
    public void push(String value) {
        System.out.println("Received push message: "+value);
    }
    public void registeredForPush(String discard) {
        // Successfully registered device for push!!
        String DEVICE_ID = Push.getPushKey();
        // Store this DEVICE_ID for later use
    }
    public void pushRegistrationError(String err, int errCode) {
        System.out.println("An error occurred during push
registration.");
    }
}
```

## Sending a Push

```
https://push.codenameone.com/push/push?token=PUSH_TOKEN
&device=DEVICE_ID1&device=DEVICE_ID2&...&device=DEVICE_IDN
&type=PUSH_TYPE{1|2|3|4|5|99|100|101}
&auth=FCM_SERVER_API_KEY
&certPassword=ITUNES_CERT_PASSWORD
&cert=ITUNES_CERT_URL
&body=MESSAGE_BODY
&production=ITUNES_PRODUCTION_PUSH{true|false}
&sid=WNS_SID
&client_secret=WNS_CLIENT_SECRET
```

Obtain **PUSH\_TOKEN** from Codename one Dashboard

## Push Message Types

**0, 1** - The default push types, works everywhere. string as the push alert

**2** - hidden, This won't show any visual indicator on any OS!  
Android: triggers push(String) w message body. iOS: only if the app in fg o/w push is lost

**3** - 1 + 2 = 3 combines visual with non-visual portion. Message form:  
[This is what the user won't see](#); [This is something he will see](#).

**4** - Split a visual push request based on the format `title;body`

**5** - Regular push but no sound

**99** - XML message body; Root element contains at least type and body attrs. Supports image attachments and push actions. E.g.  
`<push type="1" body="Hello World"/>`

**100** - Set numeric badge on app icon. The body of the message must be a number e.g. unread count.

**101** - Same as 100 with added message separated with a space. E.g.  
`30 You have 30 unread messages`  
will set the badge to "30" and present the push notification text of "You have 30 unread messages".