

ADVANCED FEATURES



Table of Contents

Viewing content in a browser.....	3
Browser Client.....	4
Push notifications.....	9
Sending Push Notifications.....	10
iOS: Setup Push Notifications in Firebase	15
Android: Setup Push Notifications in Firebase	20
About Segmented Push Notifications.....	23
iOS: Setup key-based authentication	26
Automated Content Upload.....	29
Content Sources	30
Integration API: push content from 3rd party solutions	33

Viewing content in a browser

Browser Client

The Browser Client is a third Twixl publishing platform. Next to publishing your app on iOS and Android, you can also offer the app experience in Safari, Chrome, Firefox, Edge etc.



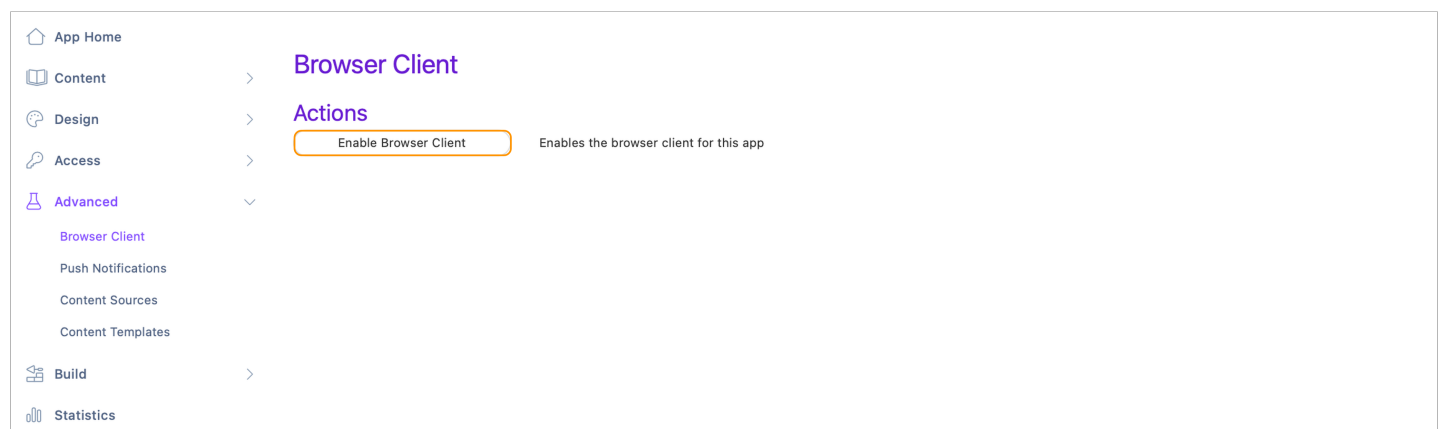
ABOUT THE PURPOSE OF THE BROWSER CLIENT:

Please note that the Browser Client is not intended as a replacement for a mobile app, nor it is intended as a replacement for a normal web site. Its primary focus is to display all contents of an app in a desktop browser. This may be especially handy for in-house applications (sales tools, marketing tools): if not all employees in the company have access to a mobile device, you can use the browser client to make this content available to these users.

How to activate the Browser Client?

To activate the Browser Client option for your app:

- Go to the sidebar menu for your app > and select 'Browser Client' in the 'Advanced' section.
- Now click on '*Enable Browser Client*' to enable the Browser Client for this app.



Available options after enabling the Browser Client

Browser Client

[Try Browser Client](#)
[Settings](#)
[Disable Browser Client](#)
[Pre-render Content](#)
[Delete Pre-rendered Content](#)

Try the browser client with this app

Edit the settings of the browser client for this app

Disables the browser client for this app

This prepares all your content for fast viewing in the browser client

This deletes the pre-rendered content (Twixl only)

Pre-render Status

6 of 7 content item(s) are pre-rendered

1. **Try Browser Client:** This will launch a new window with the Browser Client version of your app.
2. **Settings:** allows you to change the settings for the browser client.
3. **Disable Browser Client:** lets you deactivate the Browser Client for the app.
4. **Pre-render Content:** This prepares all your content for fast viewing in the Browser Client. By default, the Twixl platform will pre-render content on a regular basis in the background.

Custom Domain Settings

Custom Domain Name

Custom Domain ⓘ

If you want to work with a **Custom Domain Name**, that's possible. Enter the subdomain name in the settings field. Don't forget to first add a [CNAME record](#) to the DNS server for your domain (like `myapp.mydomain.com`) that points to `browserclient.twixlmedia.com`.

IMPORTANT

Once you have entered the details for your custom domain in the Browser Client Settings, it will not be activated automatically. Please create a support ticket to

ask our support team to activate it. A HTTPS (Let's Encrypt) certificate will then be created for you. The renewal of this certificate will be done automatically as well. Good for your peace of mind!

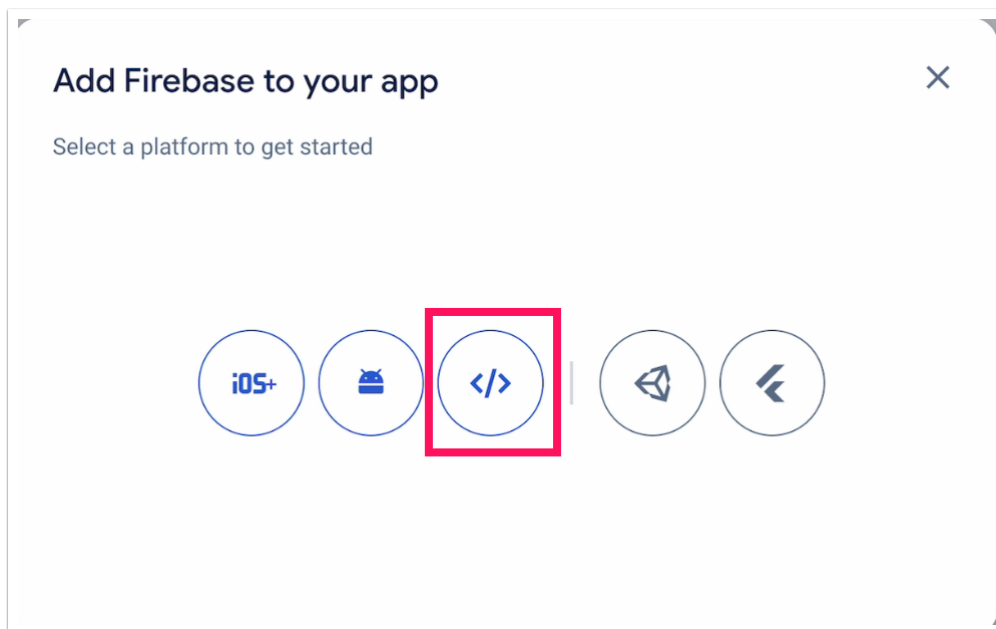
! CUSTOM DOMAINS ARE A PAID OPTION

See:

- [Custom domain for Browser Client with Let's Encrypt SSL Certificate - One App - one-time fee](#)
- [Custom domain for Browser Client with Let's Encrypt SSL Certificate - Multiple Apps - one-time fee](#)

Google Analytics 4 Settings

1. In Firebase, Click on the **Gear icon** next to 'Project Overview', and select 'Project Settings' to add a 'web app' to your configuration.



2. Fill in the *App nickname*.
3. Click *Register* to continue.
4. Copy the Config information

iOS+ janire.push.notification

Web apps

test app
Web App

SDK setup and configuration

☐ npm
☐ CDN
☒ Config

Get the snippet for your app's Firebase config object. [Learn more](#).

Firebase configuration object containing keys and identifiers for your app:

```
// For Firebase JS SDK v7.20.0 and later, measurementId is optional
const firebaseConfig = {
  apiKey: "AIzaSyCDoNoAY75Lek2AFGgR4K0fMBkF7sQs8T4",
  authDomain: "analytics-test---ga4.firebaseio.com",
  projectId: "analytics-test---ga4",
  storageBucket: "analytics-test---ga4.appspot.com",
  messagingSenderId: "975535085422",
  appId: "1:975535085422:web:22c06219edf4e636da937f",
  measurementId: "G-K71D38Z9Z1"
};
```

Are you using npm and a bundler like webpack or Rollup? Check out the [modular SDK](#).

Learn more about Firebase for web: [Get Started](#), [Web SDK API Reference](#), [Samples](#)

Remove this app

5. In left menu > Select Advanced > Browser Client > Settings

Paste the Config Information in the Browser Client Firebase Config field (Under Google Analytics 4).

Google Analytics 4

Browser Client Firebase Config ⓘ

Other settings

Other settings

Apple App ID ⓘ

1. **Apple App ID:** Safari provides a standardized method of bannerizing apps on the App Store from a web site. The Browser Client has built-in support for that.
 1. **You need the Apple ID for your app from AppStore Connect.** You'll find it under General – App Information.
 2. Copy-paste that number in your Browser Client Settings *Apple App ID* field.

Designing the Browser Client interface

You can create specific Browse Grid and Item Styles for the Browser Client of your app.

- [More info about Browse Grid Styles](#)
- [More info about Item Styles](#)

Push notifications

Sending Push Notifications

A notification is a great way to let users/readers know when new content is available or to push other information.

1. To start

Before you can start sending push notifications, make sure the required push notification configuration information has been added for your app on the platform.

Check out the following articles for help on how to setup your app for Push Notifications:

- For iOS:
 - Step 1: [Setup Key based authentication](#)
 - Step 2: [Setting up Firebase Push Notifications](#)
- For Android:
 - [Setting up Firebase Push Notifications](#)

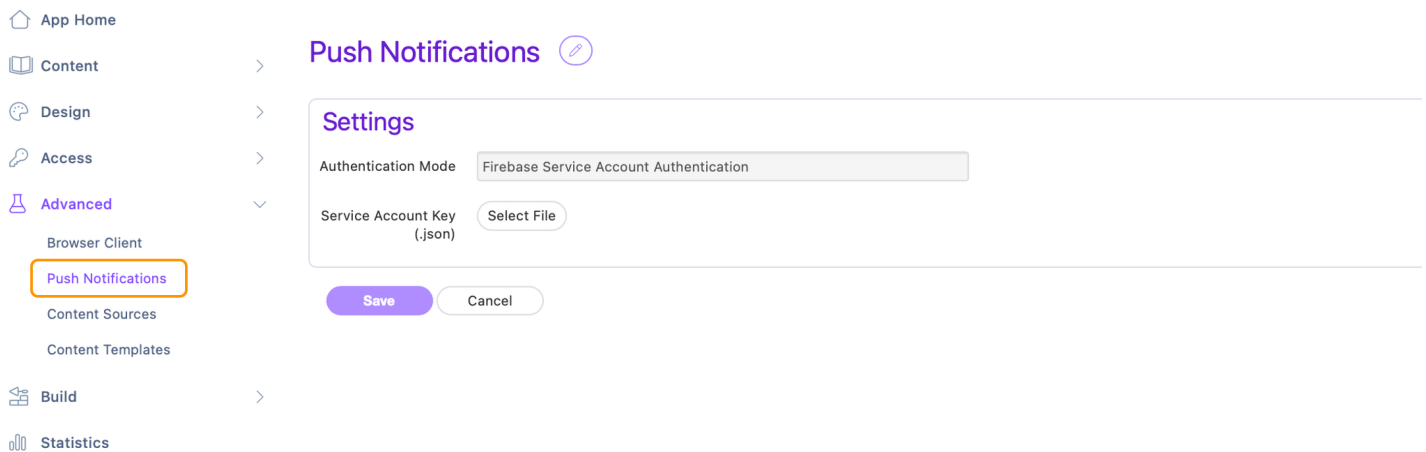


IMPORTANT NOTE:

If you activate push notifications for an existing app, you'll need to create a new build of the app.

Once you have followed all the previous steps, go to the Twixl platform to upload your Service Account Key.

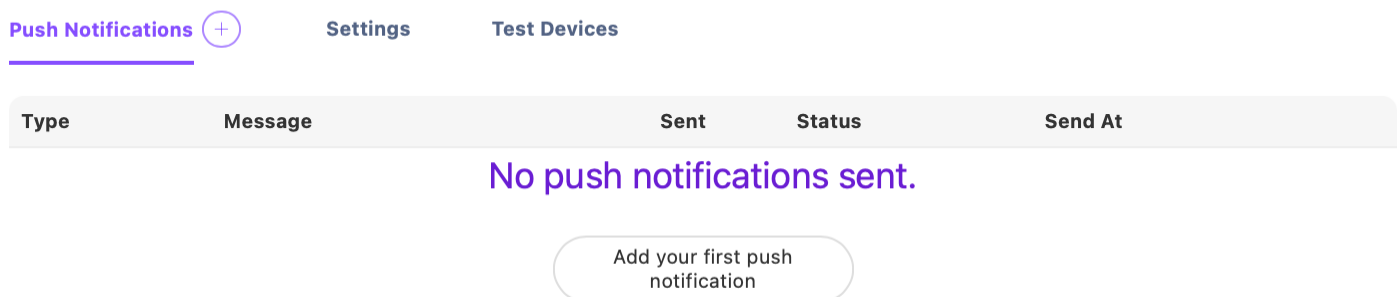
Via Platform > Menu > Advanced > Push Notifications



Once you have uploaded the Service Account Key and saved your Settings, 3 tabs will appear:

- **Push Notifications:** Where you can create your Push Notifications to send to the app users.
- **Settings:** If you ever need to upload a different Service Account Key.
- **Test Devices:** Shows a list with all known Test Devices.

Push Notifications



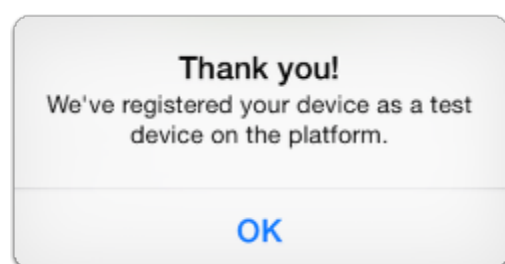
2. Testing

If you want to use push notifications in your app, we recommend to test this before going live. You can register a number of test devices (phones, tablets) and send a test message to these devices only. Read below for more details on how to do this.

We provide a special custom URL scheme that allows you to test push notifications before distributing the app to the public. If your app is already live, you can enter this

`tp-register-test-device://` to a discrete location in your app so your users don't register their devices by accident.

1. Make sure push notifications for your app have been configured properly, as described [above](#).
2. After you have completed the setup (for iOS and/or Android), create a new build of your app.
3. On the platform, add a web link to your app with the URL `tp-register-test-device://`.
4. Install the app on your device and make sure the app has requested to allow push notifications. Usually this requires you to open the app, close it and open it again before a pop-up will appear requesting your approval.
5. Selecting the link in the app will then add the current device to the list of test devices for push notifications.



! NOTE:

The test functionality on iOS devices is only available in [TestFlight](#) builds of your app. Android builds of your app can just be installed on your Android device to register as a test device and test your Push Notifications.

3. Sending Push Notifications

Sending Push Notifications is done from the first tab 'Push Notifications' (see above) by clicking the '+' icon.

You can send **Plain** or **Rich Push Notifications**. **Rich Push Notifications** can include an optional **Title**, **Description** and even an **Image**.

💡 Do you want to link to a published content item? Make sure to send your push notification at least 5 minutes after you marked the content item as published live, because of a caching mechanism that Twixl apps use! This way, users clicking on the link in the push notification will be able to access the content.

3.1. Plain Push Notifications

1. **Type: Plain Push Notifications**
2. **Message:** Of course you want to include a **Message** in your *Plain Push Notification*. This **Message** can include 140 characters.
3. **Link:** You can attach a `tp-collection` url scheme. This way, when readers tap on the *Push Notification*, they will be redirected to the right *Collection* or *Content Item*. For more info, see [Using custom URL Schemes in your app](#).
4. **Send at:** This way, a *Push Notification* can be sent out at a specified time in the future.
5. **Which Devices:** You can target specific segments of devices, i.e.:
 - **All devices:** Will send the *Push Notification* to **all registered devices**.
 - **All iOS devices:** Will send the *Push Notification* to all registered **iOS devices**.
 - **All Android devices:** Will send the *Push Notification* to all registered **Android devices**.
 - **Specific test devices:** Here you can select specific **Test Devices** as a target for your *Push Notification*. For more info, see [Testing](#)

Add Push Notification

1 Plain Rich

2 Message ⓘ

Characters Remaining: 140

3 Link ⓘ

4 Send At

28/06/2024 13:55

5 Send to *

All device(s)

Cancel Send

3.2. Rich Push Notifications

1. **Type: Rich Push Notifications**
2. **Title:** Your *Rich Push Notification* can include a **Title**
3. **Description:** The **Description** has no limitations in terms of the number of characters.
4. **Image:** Your *Rich Push Notification* can include an optional **Image**.

5. **Link:** You can attach a `tp-collection` url scheme. This way, when readers tap on the *Push Notification*, they will be redirected to the right *Collection* or *Content Item*. For more info, see [Using custom URL Schemes in your app](#).
6. **Send at:** This way, a *Push Notification* can be sent out at a specified time in the future.
7. **Which Devices:** You can target specific segments of devices, i.e.:
 - **All devices:** Will send the *Push Notification* to **all registered devices**.
 - **All iOS devices:** Will send the *Push Notification* to all registered **iOS devices**.
 - **All Android devices:** Will send the *Push Notification* to all registered **Android devices**.
 - **Specific test devices:** Here you can select specific **Test Devices** as a target for your *Push Notification*. For more info, see [Testing](#)

Add Push Notification

Plain
Rich ①

②

Title

③

Message

④

Image

Drag and drop image here ③

OR

Select Files

⑤

Link ①

⑥

Send At

28/06/2024 13:55

⑦

Send to *

All device(s) ▼

Cancel
Send

NOTE:

You will only be able to start sending notifications when your app has been installed and opened on at least 1 device.

iOS: Setup Push Notifications in Firebase

As of release 19.0, **new Twixl apps** can only use Google Firebase for sending out push notifications.

We chose to adopt Firebase on both iOS and Android because of the flexibility it offers for more advanced users that want to build marketing campaigns across platforms in a unified way.

Watch a video with step by step instructions

1. Get Started

1. Go to <https://console.firebase.google.com/>
2. **Signin** with a Google Account, preferably but not necessarily the same one you use for the *Google Play Developer account*.

2. Add a Project

1. Click on **Add Project**. (You can also choose an existing project if you already created one before.)
2. Choose a **Project Name**.
3. Choose a **Project ID** or accept the proposed one.

3. Add an iOS app to the project

1. In Project Settings, select the General tab.
2. Add an iOS app.
3. Fill in the **Apple Bundle ID** (that should correspond to the **Bundle ID in the build settings**, a reverse DNS name like `com.mycompany.myapp`), and optionally specify an app nickname.
4. Click Register to continue.
5. Download the `GoogleService-Info.plist` file and save it. You'll need it later in your build setting on the platform.

4. Configure your app to use Apple Push Notifications service (APNs)

There are two ways to configure Firebase to connect to APNs: through the use of a private key that never expires, or through the use of certificates issued from Apple's Developer Portal. The approach of using a **private key** is the recommended way to configure Firebase for Twixl apps, because this key never expires.

1. Follow the steps to create a p8 private key [here](#). Write down the Key ID, the Team ID and download the key as a p8 file.
2. Upload this p8 key to the Firebase console.
 1. Inside your project in the Firebase console, select the gear icon, select Project Settings, and then select the Cloud Messaging tab.
 2. At the bottom part of the page, select the iOS app.

Apple app configuration

Apple apps

iOS+

com.twinklmedia.documentation

Firebase Cloud Messaging can use either an APNs authentication key or APNs certificate to connect with APNs

APNs Authentication Key

★ Configuration with auth keys is recommended as they are the more current method for sending notifications to Apple devices

File	Key ID	Team ID
No APNs auth key		

Upload

APNs Certificates

▼

3. In APNs authentication key under Apple app configuration, click the Upload button.
4. Select the saved key, and click Open.
5. Add the Key ID and Team ID click the Upload button.

5. Download the Service Account Private Key

1. Click on the **Gear icon** next to 'Project Overview' (you can skip this step if you already configured Firebase for your Android app and uploaded the key to the Twixl platform).
2. Select **Project Settings**.
3. Select the tab **Cloud Messaging API**.

4. Enable 'Firebase Cloud Messaging API' if it's not enabled yet (for new projects it will be enabled by default).
5. Select 'Manage service accounts'.
6. Click on the service account `firebaseadminsdk-...`
7. Go to 'Keys', click 'Generate a new private key' and save the private key (a .json file). This key is the one that needs to be uploaded in the *Push Notifications* section on the platform.

6. Upload the JSON Private Key to the Twixl platform

1. (You can also skip this if you already configured push for Android): Go to the detail view of your app on the Twixl platform, then select *Push Notifications* from the side menu (under 'Advanced').
2. Upload the Service Account Private Key (JSON) generated in the previous step.

Push Notifications

Settings

Authentication Mode

Firebase Service Account Authentication

Service Account Key
(.json)

Select File

Save

Cancel

7. Configuring iOS push settings on the Twixl platform

1. On the Twixl platform, go to your app, then navigate to the build settings section.
2. In the build setting for your app, upload the Service Account File that you downloaded in step 2 (`GoogleService-Info.plist`).

Push Notifications and Google Analytics 4

google-services.json

The instructions for the google-services.json can be found [here](#).

☐ Remove google-services.json

GoogleService-Info.plist

The instructions for the GoogleService-Info.plist can be found [here](#).

☐ Remove GoogleService-Info.plist

Android Push Notification **FIREBASE**

iOS Push Notification **KEY-BASED**

To enable push notification you need to do extra configuration [here](#).

Make sure you have done the correct [configuration for Google Analytics 4](#).

8. Create a new build using the Twixl Publisher macOS app

1. Create a build using the build setting you just updated. Install the build, via TestFlight (iOS), or directly on the device (Android).
2. Start up the app, then exit it. Start up the app a second time. It should prompt for push notification permissions.
3. You can register as a test device and send push notifications as documented [here](#).

9. What about push notifications for an existing app ?

When you already have a Twixl app configured with Push Notifications, the existing configuration will continue to work through November 30, 2023.

If you need to change the settings of the app, the only way to do so is to convert the push notifications setup to use Firebase, based on the documentation above. Note that to continue to use push notifications after November 30th, it is mandatory to move to Firebase.

10. Testing Push Notifications on iOS

To test push notifications, first create a build and [upload it to App Store Connect](#).

In the **TestFlight** tab there will be a list of all the builds uploaded. In the "Internal Testing" section it is possible to add a user that is a member of your App Store Connect team to test. Invite that member (possibly yourself) and they will receive an invitation e-mail from TestFlight to install the app. Use TestFlight on an iOS device to install the application.

On the first run, the push notifications are not activated yet, so force quit the app and run it a second time. This time permission will be asked and the device will be registered to receive push notifications.

It is possible to [register as a test device](#). After that, on the Twixl platform, you can send the first push notification to this specific test device.

If this works properly, then it is safe to assume the app is ready to be distributed and all users will be able to receive push notifications.

Send To

Bundle Identifier

com.twixlmedia.documentation

Send to *

Specific test device(s) 

Devices

☐ iPhone

Send

Cancel

Android: Setup Push Notifications in Firebase

As of release 19.0, you can only use Google Firebase for sending out push notifications.

We chose to adopt Firebase on both iOS and Android because of the flexibility it offers for more advanced users that want to build marketing campaigns across platforms in a unified way.

Watch a video with step by step instructions

1. Get Started

1. Go to <https://console.firebase.google.com/>
2. **Sign in** with a Google Account, preferably but not necessarily the same one you use for the *Google Play Developer account*.

2. Add a Project

1. Click on **Add Project**. (You can also choose an existing project if you already created one before.)
2. Choose a **Project Name**.
3. Choose a **Project ID** or accept the proposed one.

3. Add an Android app to your project

1. Click on the **Gear icon** next to 'Project Overview'
2. In **Project Settings**, select the **General** tab.
3. Add an **Android** app.
4. Fill in the *Android package name* (a reverse DNS name like *com.mycompany.myapp*), and optionally specify an *App nickname*.
5. Click *Register* to continue.
6. Download the `google-services.json` file and save it. You can skip the subsequent steps.

4. Setup your Firebase Project

1. Click on the **Gear icon** next to 'Project Overview'
2. Select **Project Settings**.
3. Select the tab **Cloud Messaging API**.
4. Enable 'Firebase Cloud Messaging API' if it's not enabled yet (for new projects it will be enabled by default).
5. Select 'Manage service accounts'.
6. Click on the service account `firebaseadminsdk-...`
7. Go to 'Keys', click 'Generate a new private key' and save the private key (a .json file). This key is the one that needs to be uploaded in the *Push Notifications* section on the platform.



USING FIREBASE FOR BOTH ANDROID AND IOS APPS:

In Firebase, you can setup an Android and/or iOS app. For more details about configuring for iOS, check [here](#).

5. Upload the JSON Key to the platform

1. Go to the detail view of your app on the Twixl platform, then select *Push Notifications* from the side menu (under 'Advanced').
2. Upload the Service Account Private Key (JSON) generated in [step 4](#).

Push Notifications

Settings

Authentication Mode

Firebase Service Account Authentication

Service Account Key
(.json)

Select File

Save

Cancel

6. Add google-services.json to your build setting

In the build setting for your app on the platform, select the `google-services.json` file from [step 3](#) and upload it there.

Push Notifications and Google Analytics 4

google-services.json
The instructions for the google-services.json can be found [here](#).

GoogleService-Info.plist
The instructions for the GoogleService-Info.plist can be found [here](#).

Android Push Notification **NOT CONFIGURED**

iOS Push Notification **NOT CONFIGURED**

To enable push notification you need to do extra configuration [here](#).

Make sure you have done the correct [configuration for Google Analytics 4](#).

7. Build your app

You have now configured push notifications for Android and are ready to create the build of your app, or you can first configure [push for iOS](#) as well.

8. Test your push notification

To test push notifications, you first have to [register your device as a test device](#). Once you have registered your device, you can test push notifications for your app. Make sure your device settings and the app settings allow push notifications.

About Segmented Push Notifications

With Twixl Publisher 19, we moved the sending of push notifications to Google Firebase. By doing this, we automatically provide a number of capabilities for sending messages to a subset of your users. We call this 'segmented push'.

1. To start

When an application links itself to Firebase, Firebase will automatically collect a number of characteristics about the user, which can be used to target push notifications.

Within a Firebase push notification one can readily limit the target with these parameters:

- The app: iOS and Android can be targeted individually
- App version The versionName (Android) or the Bundle version (iOS).
- Language: The language setting of the device OS (such as en-us or pt-br).
- Country: The country from which user activity originated.
- Region: The geographic region from which user activity originated.
- First Opened
- Last App Engagement

The UI would then look like this:

User segment

Topic

Target user if...

App		com.twixlmedia.documentation	▼		
Version	▼	contains	▼	19	▼
Languages	▼	is in	▼	Spanish	▼
First open	▼	More than	▼	10 ↕ days ago	
Last app engagement	▼	More than	▼	5 ↕ days ago	and

The push notification can be scheduled now, later, or recurring (daily or custom).

For instance, for daily, you can further customise like this:

3

Scheduling

Send to eligible users

Daily

at

12:00

(GMT+01:00) France Time

Subject to frequency limit

Once per user for this message

Start

End

Today

No end date

Next

2. Other Firebase properties

Next to the options mentioned above, Firebase in itself tracks other properties of the users.

These include:

- App store: The store from which the app was downloaded and installed.
- Browser: The browser from which user activity originated.
- City: The city from which user activity originated.
- Continent: The continent from which user activity originated.
- Device brand: The brand name of the mobile device (such as Motorola, LG, or Samsung).
- Device category: The category of the mobile device (such as mobile or tablet).
- Device model: The mobile device model name (such as iPhone 5s or SM-J500M).
- New: First opened the app within the last 7 days.
- Established: First opened the app more than 7 days ago.
- Operating system: The operating system used by visitors to your website or mobile app.
- OS version: The operating system version used by visitors to your website or mobile app (such as 9.3.2 or 5.1.1).
- Platform: The platform on which your website or mobile app ran (such as web, iOS, or Android).
- Subcontinent: The subcontinent from which user activity originated.

On the basis of these properties, it is possible to create audiences. Users become members of an audience based on predefined behavior, and remain member of this audience for a predefined time.

For instance, one could define an audience of Spanish speaking people living in New York. With an audience you can target a push notification specifically to these people.

3. Integrating Google Analytics 4

If you also gather analytics in your app, using Google Analytics 4, you even have more filtering options for selecting your audience.

iOS: Setup key-based authentication

1. Setup your App ID in the Apple Portal

To enable Push Notifications in your iOS app, you need to configure the **App ID** properly.

1. Login to your Apple Developer Account on <https://developer.apple.com>, then in the section **Certificates, Identifiers & Profiles**, select **Identifiers** > **App ID**
2. If you want to edit an existing App ID, select *Configure*, otherwise select '+' to add a new *App ID*.
3. After you entered a description for this *App ID* and the *Bundle Identifier* (**make sure to use the same reverse-DNS name as in the Twixl platform**), you can configure it.

The screenshot shows the 'Edit your App ID Configuration' page in the Apple Developer Portal. The page title is 'Certificates, Identifiers & Profiles'. Below the title is a link '< All Identifiers'. The main heading is 'Edit your App ID Configuration' with 'Remove' and 'Save' buttons. The configuration details are as follows:

Platform	App ID Prefix
iOS, macOS, tvOS, watchOS	ZM7SDYSP39 (Team ID)

Description: Avantgard Magazine (with a text input field and a note: 'You cannot use special characters such as @, &, * , ' , " , ~ , .')

Bundle ID: com.twiximedia.avantgard (explicit)

Capabilities

ENABLED	NAME
☐	Access WiFi Information ⓘ
☐	App Attest ⓘ
☐	App Groups ⓘ Configure
☐	Apple Pay Payment Processing ⓘ Configure
☐	Associated Domains ⓘ
☐	AutoFill Credential Provider ⓘ
☐	ClassKit ⓘ
☐	Custom Network Protocol ⓘ

2. Activate Push Notifications for your App ID

1. When in the App ID detail window, you'll notice a list of all *App Capabilities*.
2. Enable *Push Notifications*.
3. Click 'Done'.

3. How to create the authentication key (p8)

1. In the Apple dev portal, go to **Certificates, Identifiers & Profiles**, then select **Keys**.
2. Click + to add a new key.

3. Give the key a name.
4. Enable the service called Apple Push Notifications (APNs).
5. Click *Continue*. You will then be asked to confirm your key configuration.
6. Click on *Confirm* to create the key.
7. Select *Download* in the next screen to download the **.p8 file** to your local machine.
8. Write down the value for *Key ID*.
9. Then, navigate to <https://developer.apple.com/account> and click on *Membership* in the left column, and write down the *Team ID*.
10. You are now ready to go ahead and configure push notifications on Firebase, as described [here](#).

Certificates, Identifiers & Profiles

[← All Keys](#)

Register a New Key

5

[Continue](#)

Key Name

AvantGand Magazine

You cannot use special characters such as @, &, *, ' ", -, .

ENABLE

NAME

SERVICE



Apple Push Notifications service (APNs)

Establish connectivity between your notification server and the Apple Push Notification service. One key is used for all of your apps. [Learn more](#)



DeviceCheck

Access the DeviceCheck and AppAttest APIs to get data that your associated server can use in its business logic to protect your business while maintaining user privacy. [Learn more](#)

Certificates, Identifiers & Profiles

[← All Keys](#)

Register a New Key

[Back](#)

[Register](#)

6

Key Name

AvantGand Magazine

ENABLE

NAME

SERVICE



Apple Push Notifications service (APNs)

Establish connectivity between your notification server and the Apple Push Notification service. One key is used for all of your apps.

Certificates, Identifiers & Profiles

[← All Keys](#)

Download Your Key

7

Download

Done



After downloading your key, it cannot be re-downloaded as the server copy is removed. If you are not prepared to download your key at this time, click Done and download it at a later time. Be sure to save a backup of your key in a secure place.

Name: AvantGand Magazine

Key ID:

8

Services: Apple Push Notifications service (APNs)

Apple Developer

Account

Program Resources

Overview

Membership

People

Certificates, IDs & Profiles

App Store Connect

CloudKit Dashboard

Servers



Membership Details

Your team's membership information and legal agreements.

Membership Information

Program Type	Apple Developer Program
Team Name	Twixl media bvba
Team ID	9

Automated Content Upload

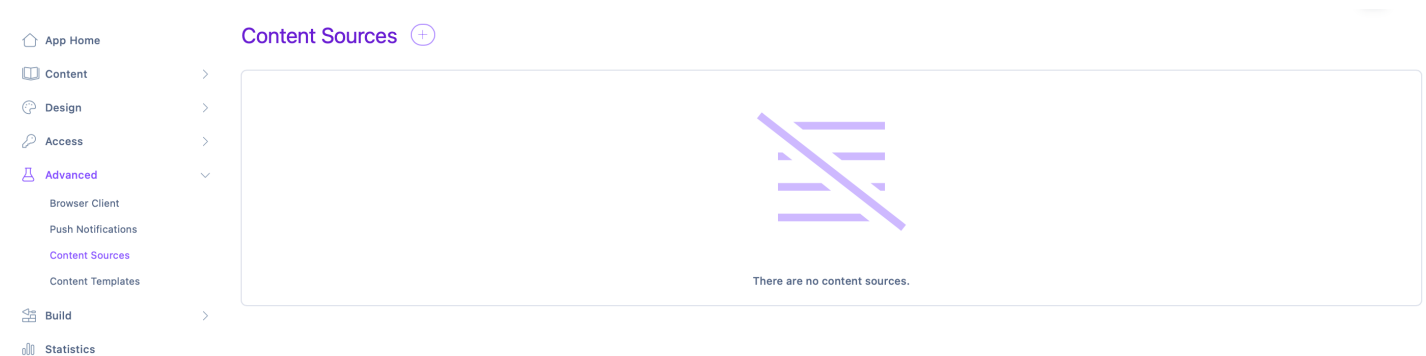
Content Sources

You can also import content from different sources automatically via content sources. Depending on your settings, new external content will be added to your app automatically.

Types of Content Sources

Content sources are designed to make automated publishing in your apps easier, allowing to easily integrate different types of feeds in your apps.

- In the left menu > Advanced > Content Sources > Click "+" icon to add a content source



Configuration

Select one of these source types from the dropdown list:

- **RSS/Atom feed:** content from just about any web site, Drupal, Wordpress, etc.
- **YouTube Channel:** a list of videos from a YouTube channel
- **YouTube User:** a list of videos from a YouTube user
- **YouTube Playlist:** a list of videos from a YouTube playlist
- **Vimeo Channel:** a list of videos from a Vimeo channel
- **Vimeo User:** a list of videos from a Vimeo user
- **Vimeo Group:** a list of videos from a Vimeo group

Enter the required information (feed URL, YouTube channel ID, etc.) and a refresh interval that determines the update frequency for the content.

Furthermore you can define the collection where you want to add the feed, the item style that will determine how the feed will be displayed in the cell, and you can select from a number of predefined templates for styling the content of an imported RSS entry.

i Some content sources have a maximum amount of items that can be imported.
E.g. :

- YouTube will let you import the 15 most recent movies only.
- Vimeo has this number limited to 10.

If your channel, user or playlist contains more than 10 or 15 videos, you will need to upload the remaining ones manually.

Add Content Source

i The types of content sources are.

- RSS/Atom feed: content from just about any web site, Drupal, Wordpress, etc.
- YouTube Channel: a list of videos from a YouTube channel
- YouTube User: a list of videos from a YouTube user
- YouTube Playlist: a list of videos from a YouTube playlist
- Vimeo Channel: a list of videos from a Vimeo channel
- Vimeo User: a list of videos from a Vimeo user
- Vimeo Group: a list of videos from a Vimeo group

[Read more](#)

Content Source Details

Type * **i**
RSS/Atom Feed

Name *

Feed URL * **i**

Refresh Interval **i**
Every day

Max nr. of Items **i**
0

Import Preferences

Target Collection **i**
-- unassigned --

Item Style * **i**
Default

Content Template **i**
-- select content template --

Content Source Options

Enabled **i** ☒ Publish Automatically **i** ☒

After saving the Content Source, this entry will appear in the Content Source page. By selecting the Content Source item, one can enable, disable, manually refresh or delete the Content Source.

Content Sources **(+)**

	Name	Type	Target Collection	Items	Status
	Select action:				
☒	Demo	YouTube Channel		0	Disabled

! IMPORTANT NOTE:

"Automated Content Sources" is an extra paid option.

Integration API: push content from 3rd party solutions

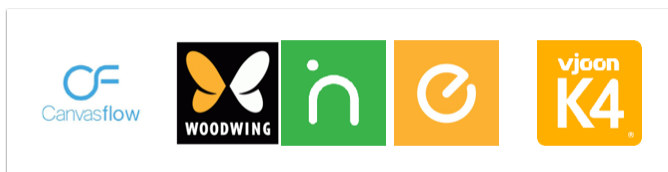
The "Integration API" is an extra paid option that allows you to use the Twixl Platform Admin API to push content from external CMS systems.

Twixl offers the ability to integrate with different solutions for automated production flows and so to push content to the Twixl Platform and into your app.

The "Integration API" option allows you to integrate with a number of solutions that we support out of the box, but using that same API, basically any type of CMS could be setup to push content to the Twixl Platform. Some custom development or help from your local integrator may be required to achieve this.

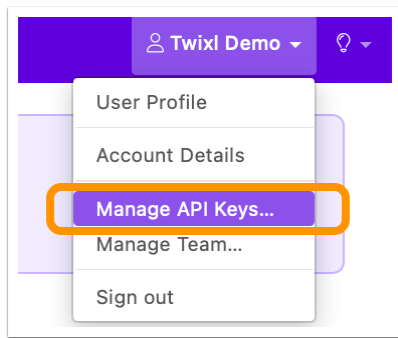
The following solutions have built-in support for integration with Twixl Platform:

- [Canvasflow](#)
- [vjoon K4](#)
- [WoodWing Enterprise](#)
- [WoodWing Inception](#)

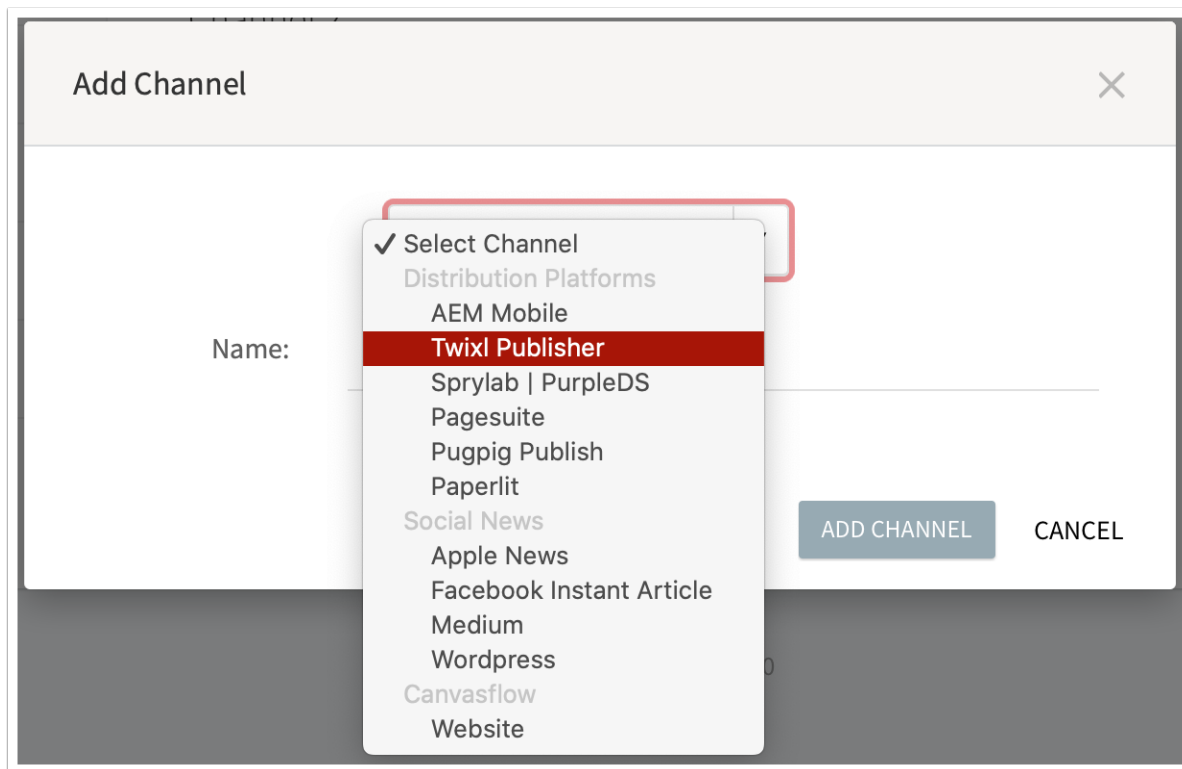


Setup

To integrate with an external solution, navigate to "Manage API Keys" in the User menu in the top right corner. There, you will be able to create an API Key for your app that will have to be used in the external solution.



1. Canvasflow



Login to your CanvasFlow account on <https://canvasflow.io>. Navigate to Settings, then select 'Publish Channels'.

As soon as you select Twixl Publisher from the list of channels, you'll get a panel where you need to enter the required information:

- Name of the app
- App Key
- Admin API Key
- Publish option: decide whether you are just going to upload content that can be published later, or whether you want to upload the content and make it available in the app immediately.

Add Channel [X]

Twixl Publisher ▼

Name:

Your Twixl App Key & Admin API Key can be found by navigating to the [API & Integration](#) section from the app menu within the [Twixl Distribution platform](#).
[Click here](#) for more help with integrating with Twixl.

App Key:

Admin API Key:

Publish:

1.1. Video with instructions

2. WoodWing Inception

1. [Login](#) to your Woodwing Inception account
2. Go to the [Publish Channels](#)
3. Add a new custom publish channel by clicking the **Add Channel** button under custom channels
4. As the endpoint url, enter: `https://platform.twixlmedia.com/api/2/[app_key]/inception-channel`
5. Enter a name and target name
6. Next time you publish a story from Inception, you now have the option to send it to the Twixl Platform
7. After publishing the story, the story will show up in the [Unassigned Content Items](#)

2.1. Automatic Assignment

To automatically add a content item to a specific collection, you can add that information to the publish channel URL:

1. Find the numeric id of the collection in which you want the story to appear
2. Change the publish channel URL to: `https://platform.twixlmedia.com/api/2/[app_key]/inception-channel?collection_id=[ID]`

Free Of Charge	Yes
Sort Mode	Not set
Created On	2016-11-02 17:47
Published On	-
Twixl ID	33385
External ID	-

2.2. WoodWing Enterprise

In order to be able to send content from WoodWing Enterprise, you'll need to have a partner that provides a connector from Content Station to Twixl Publisher.

3. vjoon K4

You can use vjoon K4[®] to connect to Twixl Publisher to produce digital editions of your publications for tablet and other mobile devices. This process makes use of standard vjoon K4 functions and can be highly automated, allowing you to manage layouts, articles and meta data within the system, have multiple users work at content parallel, and make use of Adobe InDesign Server to render files in the background.

Follow the link below to download a PDF with step-by-step instructions on how to push content from vjoon K4 to Twixl Publisher:

[Download the vjoon K4 integration manual](#)

4. Other solutions

While the solutions above are already pre-configured to be used with Twixl's Integration API, and can be used more or less 'out of the box', you can basically push content from any other external CMS solution, such as WordPress or Drupal-based web sites, using the Integration API documentation on the Twixl platform.

