

MANAGE YOUR APP



Table of Contents

Apps on the platform	4
Managing your app on the platform	5
Creating your app	6
Working with Collections	12
Working with Content Items	15
Automated Content Sources	19
Hamburger Menu	22
Managing content downloading	26
Manage Users	34
Searching in your app	37
Browser Client	41
Integration API	46
Sharing on social media	48
Optimizing PDF files for viewing on mobile devices	51
About file size and Twixl Publisher	52
Deleting an app on the platform	54
Working with custom "DIY" storage	56
Metered Access	58
Analytics	63
Analytics Reports Overview	64
Analytics Events Overview	82
Google Analytics 4	87
Google Analytics (legacy)	88
Adding analytics information to InDesign interactive elements	90
Push Notifications	92
iOS: Setup key-based authentication in the Apple Developer Portal	93
iOS: Setup Push Notifications in Firebase (TP19 and up)	96
Android: Setup Push Notifications in Firebase (TP19 and up)	100
Sending Push Notifications	103
Entitlements	109

Entitlement Diagram: how it works	110
Using the Entitlements option	111
Using "reset password" for the "Users & Groups" scenario (beta)	116
Integrating a Custom Entitlements Server	121
Custom Entitlements Servers Inquiry Form.....	128
In-app purchases & subscriptions.....	129
In-app purchases for Android apps.....	130
In-app purchases for iOS apps.....	135
Subscriptions for iOS & Android	138
Metered Access.....	143

Apps on the platform

Managing your app on the platform

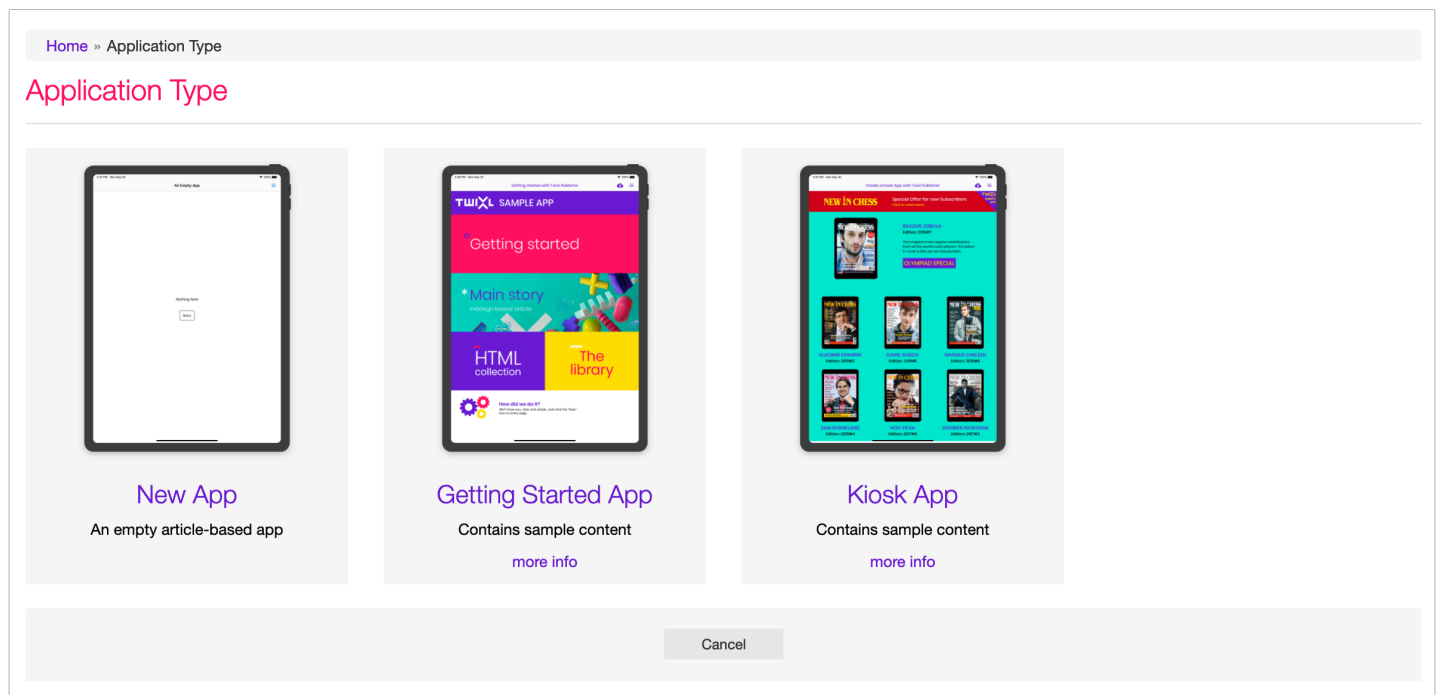
Creating your app

Everything starts with the creation of an app on the Twixl platform. This article explains the ins and outs.

Add Application

Application Type

When you add a Twixl app on the Twixl test platform for the first time, you can choose between several "templates" to get you started. For more info about each of these templates, see: [Getting Started & Sample Apps](#)



Application Mode

The next thing to select, is the **Application Mode**. The idea is that you start with a **Development** App and once your Development App has been tested and approved, you can then copy this Development App to a **Production** App. Of course you can start with a Production App right away as well.

Home » Add App » Application Mode

Application Mode

Development
An app for testing

Production
An app for distribution

Cancel



ABOUT APPLICATION MODES AND APPLICATION BUILDS:

Development and Production Apps are linked to several Application Build Types. For a complete overview, see:

- [iOS app build types](#)
- [Android app build types](#)

Essential app options

Add Article-Based Application

ARTICLE-BASED | DEVELOPMENT MODE

Application Details

Application Name *
La Grande Motte

Storage Type *
Twixl Storage

Bundle Identifier *
com.twixlmedia.la-grande-motte

This should be a reverse DNS name like: com.twixlmedia.appname. Only lowercase characters, numbers and dots should be used.

Storage type cannot be changed after the app has been created

Metadata

Application Title
La Grande Motte

Application Icon
Select File

Support Email *
demo+app@twixlmedia.com


The email address where a copy of the support emails should be sent to.

The following information should be entered:

- **Application Name**
- **Bundle Identifier:** This should be a reverse DNS name like com.twixlmedia.appname - only lowercase characters, numbers and dots should be used.

- **Application Title:** Use this to customize the title shown in the home screen of your app. Leave empty to use the application name
- **Application Icon:** Upload an icon that will be displayed in your app list – it will be resized automatically
- **Support Email:** The email address where the support emails should be sent to.

App options overview


Update App: La Grande Motte - Production
ARTICLE-BASED | PRODUCTION MODE

1

Application Details

Application Name *

Storage Type

Twixl Storage

Bundle Identifier *

Storage Region

Europe

This should be a reverse DNS name like: com.twixlmedia.appname. Only lowercase characters, numbers and dots should be used.

2

Metadata

Application Title

Application Icon

Select File

Support Email *

The email address where a copy of the support emails should be sent to.

3

Privacy Policy

Button Title

The title of the button which triggers the privacy policy

URL

The URL of your privacy policy

When you select 'Edit application' after your app has been created, here's an overview of all the options that can be set.

- 1. Application Details:** App name and bundle identifier – both are required fields when creating an app
- 2. Metadata:** Here you can enter a title for your app, select an icon and enter a support e-mail address – this is also part of the new app creation process
- 3. Privacy Policy:** Insert a link to an external url containing your privacy policy or terms and conditions to have a button – with the title of your choice – in 'settings' and the paywall (if applicable). If you publish your app in the Apple App Store or Google Play Store, a Privacy Policy URL for your app is a requirement.

4

Search / Downloads

Grid Style

Default

The grid style to use for the search results and downloaded collections

Item Style

Default

The item style to use for the search results and downloaded collections

Search Mode

Full-Text Search

Search in the metadata of the content items or search in the metadata and the actual contents of the content items.

5

Supported Orientations (Android only)

Tablets

Portrait and Landscape

Phones

Portrait Only

6

Navigation Bar / Action Bar

manage custom fonts

Title Font

YanoneKaffeesatz-Bold

The font for the title in the toolbar

Title Font Scale

100

%

The size of the font for the title in the toolbar

Action Font

YanoneKaffeesatz-Bold

The font for the action buttons in the toolbar

Action Font Scale

100

%

The size of the font for the action buttons in the toolbar

Background Color

FFFFFF

#

Background color for the toolbar

Foreground Color

000000

#

Color of the title in the toolbar

Tint Color

007AFF

#

Color of the action buttons in the toolbar

App Tint Color

007AFF

#

Color of the action buttons in the app

4. Search / Downloads: See [this article](#) for more info.

5. Supported orientations (Android): Set the orientations you intend to support for Android tablets and phones. On iOS, orientation changes are handled automatically, and also support using your app in [Split View on iPad](#).

6. Navigation Toolbar: Set toolbar background color, toolbar foreground colors (i.e. for displaying the title), toolbar tint color (for navigation actions), app tint color (for messages in the app). For the **Toolbar**, you can also select custom fonts – see more on this [here](#).

7

Google Analytics Keys

IOS

The Google Analytics key to use for iOS

Android

The Google Analytics key to use for Android

Browser Client

The Google Analytics key to use for the Browser Client

8

Application Behavior

☒ Prompt for App Store Rating

Prompts the user to rate your app in the App Store.

☐ Blur snapshot in iOS & Android app switcher

Enable this option to hide potentially sensitive information in the app switcher screen.

☐ Keep All Data Offline

Requires users to download all app contents on first startup. Ideal for certain in-house apps, but not recommended for App Store apps, as such an app may be rejected by Apple.

☒ Sharing on Social Media

When allowing sharing on social media, users can share articles on social media.

☐ Encrypt PDF Files

Enables on-the-fly encryption to PDF content items.

☐ Externally Managed Content

Enable this when you have an external system that manages the content. This puts the content items and collections of your app in read-only mode on the Twixl Distribution Platform.

7. Google Analytics Keys: here you can add a Google Analytics property for your iOS and Android App and for the Browser Client. For info on where to create such a property, you can check the [Google Analytics article](#).

8. Application behavior:

- **Prompt for App Store Rating:** Prompts the user to rate your app in the App Store.
- **Blur snapshot in iOS and Android app switcher:** If you enable this option, potentially sensitive information in the app switcher screen will automatically be blurred.
- **Keep All Data Offline:** With this option enabled, an app will be forced to download all available content when starting up for the first time. See [this article](#) for more info.
- **Sharing on Social Media:** When allowing sharing on social media, users can share articles. For more info, see [this article](#).
- **Encrypt PDF Files:** Enables on-the-fly encryption for PDF content items. That way, PDF files are protected on mobile devices (most relevant on 'open' systems like Android).
- **Externally Managed Content:** Enable this when you have an external system that manages the content. This puts the content items and collections of your app in read-only mode on the Twixl Distribution Platform. For more info, see [this article](#).

9

Allow Users to access this Application

select all

deselect all

<input type="checkbox"/>	
<input type="checkbox"/>	
<input type="checkbox"/>	
<input checked="" type="checkbox"/>	
<input type="checkbox"/>	
<input type="checkbox"/>	
<input type="checkbox"/>	
<input type="checkbox"/>	
<input type="checkbox"/>	
<input type="checkbox"/>	
<input type="checkbox"/>	
<input type="checkbox"/>	
<input checked="" type="checkbox"/>	
<input type="checkbox"/>	

9. Allow Users to access this Application:

Here you can define which other users can access this app. More info about **Managing Users** can be found [here](#).

Working with Collections

Collections are a very flexible way of organizing your content in an app.

Here are some use cases for collections:

- create a collection with your articles for the day / the week (more or less like a traditional issue)
- divide your content into different sections, each their own collection, so browse pages can then allow you to go a level deeper to another browse page for a particular section
- publish a catalog with each chapter (or sub-chapter) represented by one or more collections

A Twixl app will always start with one initial browse page when the app starts, that will be based on the 'root collection'. The root collection will always have to be present and cannot be deleted.

Note that collection links can not only contain different articles (['content items'](#)), but can also have ['Collection links'](#), i.e. links to other "subcollections".

Home - La Grande Motte - Add Collection

Add Collection
La Grande Motte | Development Mode

Collection Details

Name *

Title

Published On
This is used to determine if a collection is part of a subscription or entitlement.

Styling

Open in *

Grid Style *
Tablet: 4 columns | Vertical scrolling
Phone: 4 columns | Vertical scrolling
Web: 4 columns | Vertical scrolling

Default Item Style

Sort Mode

Purchase Info

Product Identifier
If the collection is a purchase, this should match the product identifier for the In-App Purchase in App Store Connect and Google Play.
Leave empty to use the suggested identifier.
Only characters, numbers and dots should be used.

☒ **Free Of Charge**
If enabled, all users can access the contents in this collection. If disabled, the users will be presented with a payment before they can access the contents.

Thumbnail

Thumbnail
Select any JPEG/PNG file to be used as the thumbnail for the collection. If left empty, the cover of the first content item will be used as the thumbnail.

Collection Options

☐ **Root Collection?**
If checked, this is the default collection that will be displayed when the app starts

☐ **Requires entitlements**
If enabled, the entitlements sign-in form will be shown for unauthenticated users

Advanced Scripting

```

1 // Executed once for every collection before the filtering of the items
2 = function setupFilter(collection, environment) {
3   }
4
5 // Executed once for every collection
6 = function shouldShowCollection(collection, environment) {
7   return true;
8 }

```

A collection can be set to open in **browse mode**, in which case it will display the browse grid for all the items in the collection. If a collection is set to open in **detail mode**, it will automatically open the first article (content item) in that collection. A reader can then

browse from article to article in the collection by swiping. When opening in detail mode, the selected **Sort mode** will determine which article will be considered the first.

Styling

Open In *
Browse Mode

Grid Style *

Browse Mode
Detail Mode

Default Item Style
Default

Sort Mode
Publish Date, most recent first

Styling

Open In *
Browse Mode

Grid Style *
Default

Default Item Style
Default

Sort Mode

Publish Date, most recent first
Publish Date, oldest one first
Creation Date, most recent first
Creation Date, oldest one first
Title, A to Z
Title, Z to A
Manual

Purchase Info

Product Identifier

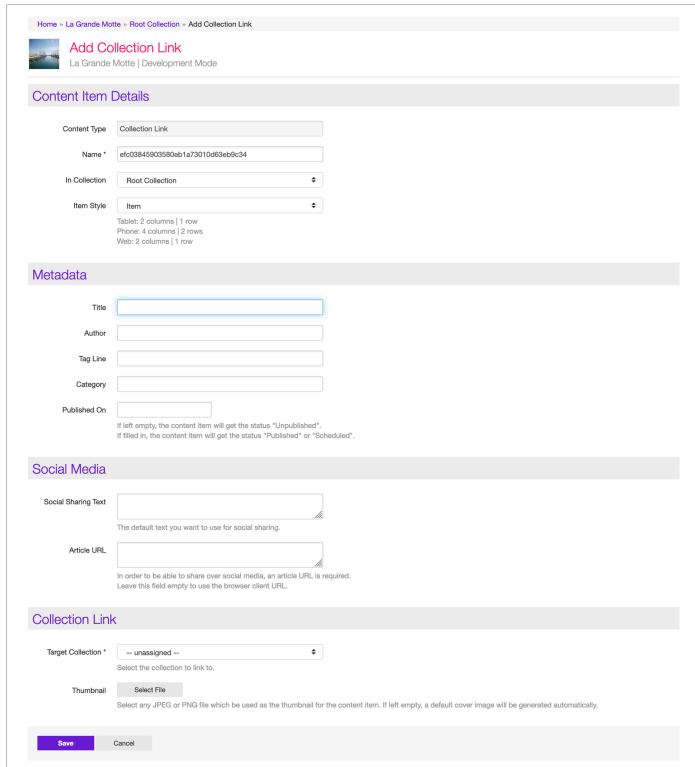
Except for the root collection itself, all collections can be offered as In-App Purchases. In that case, make sure your product identifier matches the one in App Store Connect Connect and/or Google Play.

Related articles & chapters

- [Designing your app](#)

- [Sharing on social media](#)
- [Advanced Scripting](#)
- [Entitlements](#)

Collection links



The screenshot shows the 'Add Collection Link' form in the TwiXL app. The form is divided into several sections:

- Content Item Details:** Includes fields for 'Content Type' (set to 'Collection Link'), 'Name' (with a unique ID), 'In Collection' (set to 'Root Collection'), and 'Item Style' (set to 'Item').
- Metadata:** Includes fields for 'Title', 'Author', 'Tag Line', 'Category', and 'Published On'.
- Social Media:** Includes fields for 'Social Sharing Text' and 'Article URL'.
- Collection Link:** Includes a 'Target Collection' dropdown (set to 'unassigned') and a 'Thumbnail' field with a 'Select File' button.

At the bottom, there are 'Save' and 'Cancel' buttons.

A Collection Link is a very flexible type of content item: it allows you to link to other collections in your app. When you create a collection link, an unique ID is generated for you automatically. You can change that ID but the name needs to be unique.

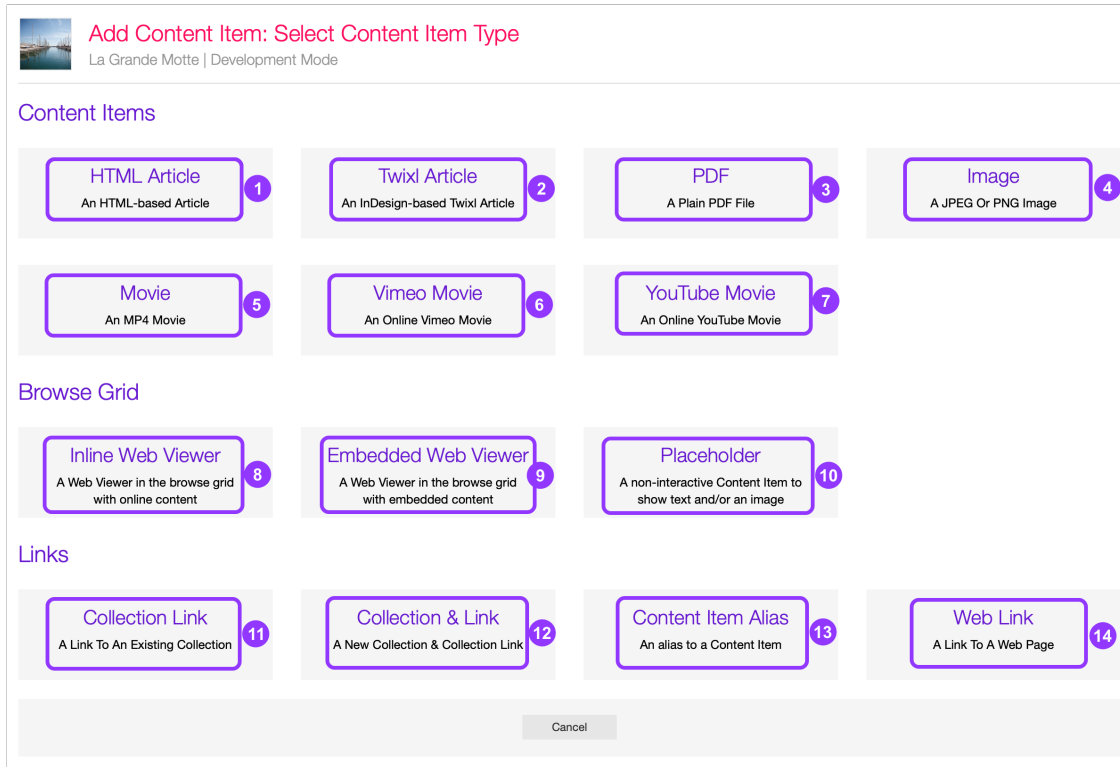
The title will be displayed in the cell for the content item. You can then select the collection to link to and determine the item style.

Custom URL schemes for navigation in your app

For a complete overview of the different custom URL schemes that can be used in an app, please refer to [this article](#).

Working with Content Items

Different types of **content items** can be added to a collection.



Overview

1. **HTML article:** uploads a .html file or a zipped folder.
2. **Twixl InDesign .article or .publication:** exported from InDesign, to be uploaded in zipped format.
3. **PDF:** uploads a plain PDF file.
4. **Image:** can be either a `.jpg` or `.png` file.
5. **Movie:** uploads a H.264 encoded `.mp4` file.
6. **YouTube movie:** links to an online movie on YouTube.
7. **Vimeo movie:** links to an online movie on Vimeo.
8. **Inline Web Viewer:** will display web content in line.
9. **Embedded Web Viewer:** will display uploaded web content in line.
10. **Placeholder:** a non-interactive Content Item to show text and/or an image, mainly used for a visual.
11. **Collection Link:** creates a link to an existing collection.
12. **Collection & Link:** creates a new collection and a link to it at the same time.
13. **Content Item Alias:** lets you link to the same content item from different locations, even with different Cell Styles.

14. **Web link:** creates a link to a web page – limited to 255 characters. If you have web links that are longer, you can create a short url with e.g. [Tiny URL](#) or [Bitly](#).

How to prepare your content for upload

1. **HTML article:** a plain .html file or a zipped folder containing all assets (images, CSS, etc.) with an index.html file at the root.
2. **Twixl InDesign articles:** you can either add individual .article InDesign exports, or you can [import a complete .publication](#) (for multiple articles). Before uploading to the platform, make sure to zip your .article or .publication.
3. **PDF file:** for best practices when uploading `.pdf` files, please check [this article](#).
4. **Image:** Images are accepted in `.jpg` or `.png` format
5. **Movie:** You can upload H.264 encoded `.mp4` movies – make sure your files are encoded at a quality optimized for mobile devices, so that these are not unnecessarily too big and preferably not larger than 1080 pixels wide. Also make sure that the bitrate of your video is not too high, as these might result in error when rendering these on some lower-end devices.
6. **YouTube movie:** Make sure the movie on YouTube is public and copy the YouTube video ID or the URL.
7. **Vimeo movie:** Make sure the movie on Vimeo is public and copy the Vimeo video ID or the URL.
8. **Inline Web Viewer:** the URL of the content that you want displayed inline
9. **Embedded Web Viewer:** will display uploaded web content in line. The web content should be uploaded as a zipped file in the same way as an HTML article (see above), but the .oam file format (Adobe Animate, Tumult Hype) are also supported.
10. **Placeholder:** a non-interactive Content Item to show text and/or an image, mainly used for a visual – images are accepted in `.jpg` or `.png` format
11. **Collection Link:** create a link to an existing collection
12. **Collection & Link:** create a new collection and a link to it at the same time
13. **Content Item Alias:** links to the same content item from different locations, even with different Cell Styles
14. **Web link:** links to a web page – limited to 255 characters. If you have web links that are longer, you can create a short url with e.g. [Tiny URL](#) or [Bitly](#).

Four ways to add content items to your app

1. Upload manually


You can upload individual content items manually, one by one.

2. Upload multiple items

You can also upload multiple items at once. The following file formats are supported:

- Zipped InDesign .article files
- HTML files
- Zipped HTML articles
- PDF files
- PNG and JPEG images
- MP4 movies

3. Import a complete InDesign .publication



Collection: Collection with .publication
LA GRANDE MOTTE - PRODUCTION | PRODUCTION MODE

Name	collection with .publication	Monolithic Download	No
Title	Collection with .publication	Free Of Charge	Yes
Product Identifier	com.twixlmedia.v5.la-grande-motte.collection_with_publication	Sort Mode	Publish Date, most recent first
Grid Style	Default	Created On	2022-01-05 23:13
Open In	Browse Mode	Published On	2022-01-05 23:12
Requires entitlements	No	Twixl ID	184925
Browser Client URL	https://browserclient.twixlmedia.com/b7638f848670358ef1171d...	Twixl UUID	9f0f636ae495374ae2a75c050d78bd4d
Twixl Link	tp-collection://collection%20with%20.publication	External ID	-

Content Items
Items are sorted on publish date, the most recent item is shown first.

Add Content Item
Add Multiple
Import Publication

You can import complete Twixl .publication files exported from InDesign: in your app, select a collection, and at the top you'll notice an option "Import publication". When you have uploaded the zipped .publication, it will be processed and automatically split into separate .article content items.


4. Push content from an external solution

[Push content](#) from solutions like WoodWing Inception, Canvasflow, vjoon K4 etc.

TWIXL DISTRIBUTION PLATFORM
La Grande Motte
Reports

Search content item
Search
Duplicate
Edit





Home > La Grande Motte > Root Collection


Collection: Root Collection
La Grande Motte | Development Mode

Title La Grande Motte	Sort Mode Manual
Product Identifier f9f9e0a3817d4f50e906b561efa7d784	Created On 2020-06-03 15:47
Grid Style Default	Twixl ID
Open In Browse Mode	Twixl UUID
Browser Client URL	External ID -
Twixl Link tp-collection://home	

Content Items
Items are sorted manually.
You can use drag-and-drop to change the order of the items.

1 Add Content Item
2 Add Multiple
3 Import Publication

Name	Content Type	Item Style	Status
<input type="checkbox"/>  La Grande Motte Name: 00_cover Not pre-rendered for browser client	HTML Article	Header	Published on 21 february 2019
<input type="checkbox"/>  Contents Name: 01_contents Not pre-rendered for browser client	HTML Article	Item	Published on 21 september 2017
<input type="checkbox"/>  About La Grande Motte Name: 02_about Not pre-rendered for browser client	HTML Article	Item	Published on 21 september 2017
<input type="checkbox"/>  The salt at Aigues-Mortes Name: 03_aigues-mortes Not pre-rendered for browser client	HTML Article	Item	Published on 21 september 2017

Sorting content items in a collection

Open In *
Grid Style *
Default Cell Style
Sort Mode

Browse Mode
Publish Date, most recent first
Publish Date, oldest one first
Creation Date, most recent first
Creation Date, oldest one first
Title, A to Z
Title, Z to A
Manual

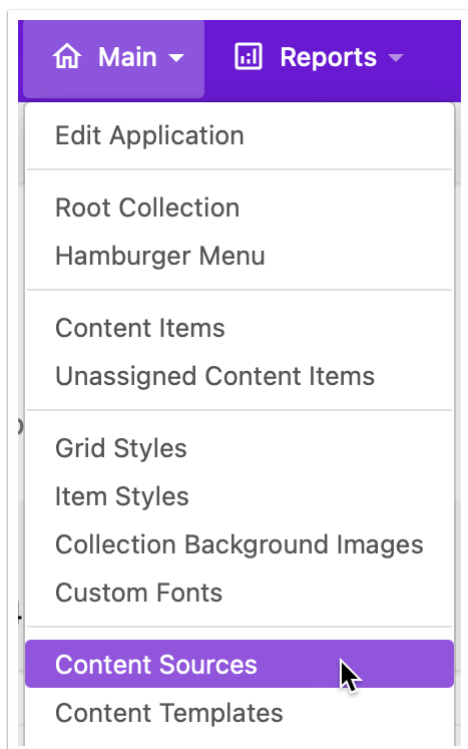
There are different ways to determine the order of the content items in a collection:

- **By Publish date, most recent first**
- **By Publish date, oldest first**
- **By Creation Date, most recent first**
- **By Creation date, oldest first**
- **By Title, A to Z**
- **By Title, Z to A**
- **Manual:** You can use drag and drop to arrange the order of the content items.

Automated Content Sources

Automated content publishing made easy.

Types of Content Sources




Content sources are designed to make automated publishing in your apps easier.

They allow to easily integrate different types of feeds in your apps.

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



Add Content Source: Select Content Source Type

La Grande Motte | Development Mode

RSS/Atom Feed

Content from another website

YouTube Channel

Videos from a YouTube Channel

YouTube User

Videos from a YouTube User

YouTube Playlist

Videos from a YouTube Playlist

Vimeo Channel

Videos from a Vimeo Channel

Vimeo User

Videos from a Vimeo User

Vimeo Group

Videos from a Vimeo Group


Cancel



IMPORTANT NOTE:

"Automated Content Sources" is an extra paid option.

Configuration



Add Content Source

La Grande Motte | Development Mode

Content Source Options

Type:

Name:

Feed URL:

The source url for the feed. RSS and Atom feeds are supported.

Refresh Interval:

How often the content source needs to be checked for new content

Max nr. Of Items:

Set to 0 to import all current and future items

Import Preferences

Target Collection:

The collection in which the new entries should be added.

Item Style:

Tablet: 2 columns | 1 row
Phone: 4 columns | 2 rows
Web: 2 columns | 1 row

Content Template:

The template to use for styling the content of the imported entry.

Options

☒ **Enabled**
If enabled, the content source will be checked for new content.

☒ **Publish Automatically**
If enabled, the content source will automatically be published.
If unchecked, the new content will remain unpublished.

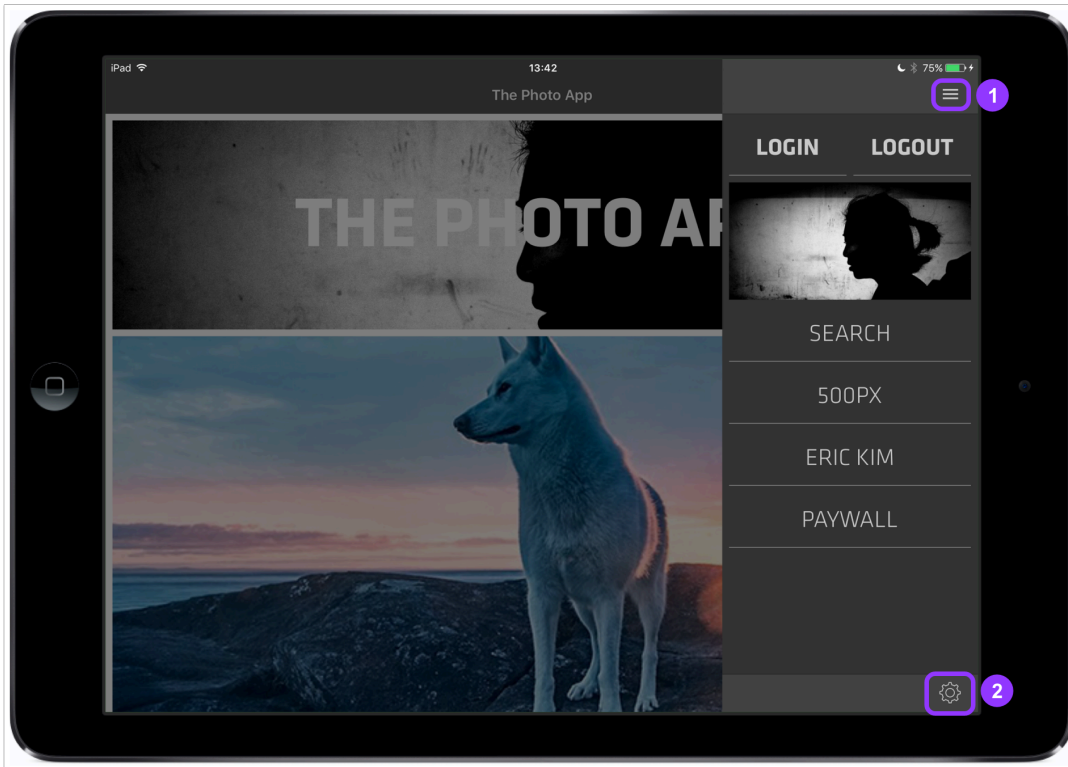
Save Cancel

Select one of these sources from the list, and enter the required information (fee 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.

Hamburger Menu

You can add a fully customizable “Hamburger Menu” to your app. This is a **quick access menu** and resides in the **top right corner** of your app (1). It is managed using a **special collection** on the Twixl Platform called *Hamburger Menu*.





1. How to activate the Hamburger Menu?

The Hamburger Menu isn't enabled by default.

You can enable it on Twixl Platform > Home > Special Collections > Hamburger Menu.

Special Collections



Title	# Items	Grid Style
 Root Collection The root collection of your app	11	Default
 Hamburger Menu The hamburger menu is not enabled. Click here to enable it		

When it is enabled, you can populate this special collection with content.

2. How to deactivate the Hamburger Menu

To deactivate the Hamburger Menu, you can just click **disable**.

Special Collections

	Title	# Items	Grid Style	
	Root Collection The root collection of your app	11	Default	
	Hamburger Menu The collection holding the items for the hamburger menu in the app.	0	Default	edit add content item disable

3. General remarks

GEAR MENU:

When activating the Hamburger Menu, the 'Gear Menu' (to access certain config settings) will be moved to the bottom of the Hamburger Menu (see (2) in the screenshot on top).

ACTIVATION AND DEACTIVATION:

When you want to temporarily deactivate the Hamburger Menu, no content will be deleted. The options will just be disabled and the special collection will become invisible.

4. What type of content can be added to your Hamburger Menu?

Since the Hamburger Menu is actually a special collection, it has to contain content items. The following types of content Items are supported in the Hamburger Menu:

- **Placeholder:** Ideal to add an image or a logo
- **Collection Link:** Ideal to create navigation shortcuts
- **Web link:** Ideal to link to social media, a privacy policy, etc.

For more info about the different types of content items, you can check [this article](#).

To upload content items to the Hamburger Menu:

- Home > Special Collections > Hamburger Menu > select 'add content item' at the right (3)

Special Collections			
Title	# Items	Grid Style	
<div>/</div> Root Collection The root collection of your app	1	Default	
<div>☰</div> Hamburger Menu The collection holding the items for the hamburger menu	0	Hamburger	<div>3</div> <div>disable edit</div> <div>add content item</div>

- Menu > Content Items > Add Content Item > In collection > Hamburger Menu (4)

Content Item Details

Content Type

Placeholder

Name *

In Collection

-- unassigned --

Item Style

-- unassigned --
Root Collection
collection the new collection
Hamburger Menu

Metadata

4

5. Look & Feel of the Hamburger Menu

⚠ WARNING:

The Hamburger Menu has a **fixed width of 300px** (tablets & phones).

The layout of your Hamburger Menu is fully customizable. You can change the layout by using the usual Twixl Platform options:

- The overall layout is defined by a [Grid Style](#) of your choice.
- The layout per content item is defined by one or more [Item Styles](#) of your choice.

**TIP:**

We strongly advise you to use **separate Grid and Cell Styles** for your Hamburger Menu. Also, give them a **good and logical name** (e.g. Hamburger Grid Style 1). This way it will be much easier to organise and select different Styles in different Collections / Content Items of your app.

6. Custom URL Scheme

Via 'Weblink' you can also use a custom URL scheme to refer to specific content such as collections or articles and much more.

For more info, [see this article](#).

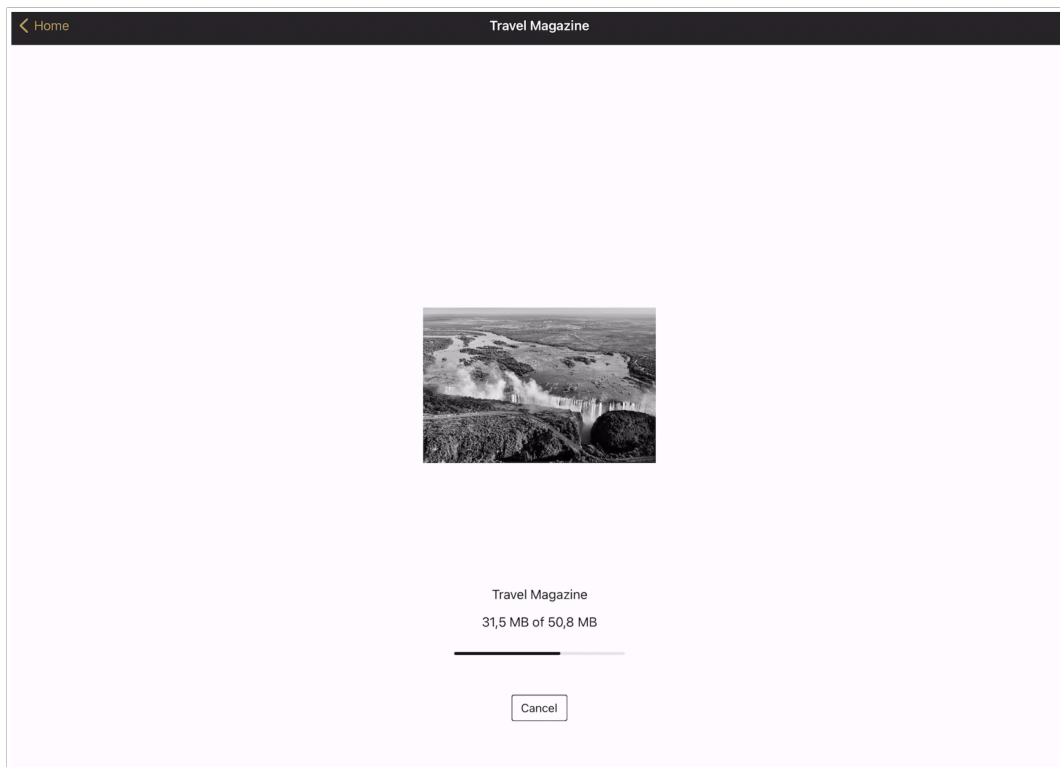
Managing content downloading

With Twixl, a user has several ways to read an article:

- **On the fly:** the article is downloaded and opened as the user swipes to the article.
- **Monolithic collection:** when the publisher has marked a collection as monolithic, the user can download this collection for easier offline reading of the articles in that collection.
- **Collection on demand (Long-press):** the user can choose to download a complete collection (both on-the-fly and monolithic) for easy offline access by long-pressing the thumbnail of that collection. This requires no configuration from the publisher's side.
- **Full offline mode (Keep all data offline):** if activated by the publisher, the user must first download all content before he can use the app.

1. Download a 'monolithic' collection for offline access

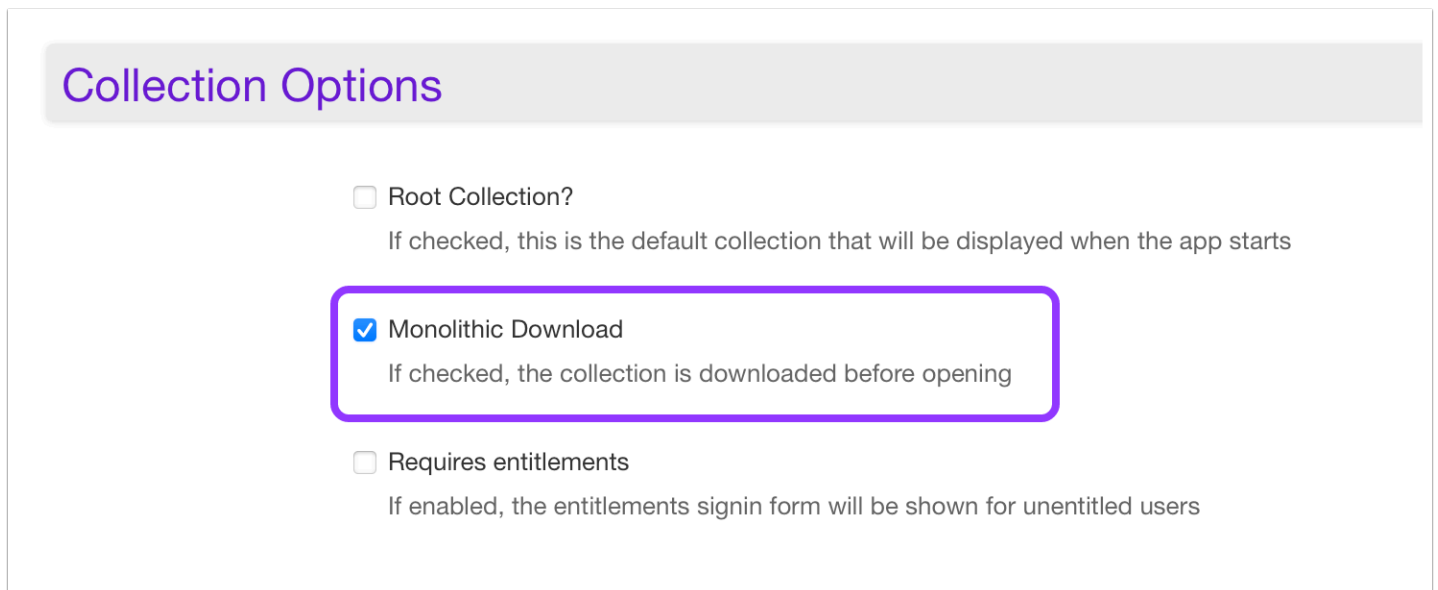
Setting a collection to '**monolithic**' gives the reader the option to **download** one or more collections so every page of the collection is immediately ready to read or the collection can be read without an internet connection. The monolithic setting can be activated by the publisher for each individual collection in the '**collection options**'.



1.1. Watch a short 'How to' movie

1.2. How to activate this feature?

If you edit the settings for a particular collection, under 'Collection Options' you'll see an option called 'Monolithic download' that can be checked. If this option is checked, when a user/reader navigates to this collection in the app, the complete collection will be downloaded first, before the first content item or the browse grid is displayed.



Collection Options

☐ Root Collection?
If checked, this is the default collection that will be displayed when the app starts

☒ Monolithic Download
If checked, the collection is downloaded before opening

☐ Requires entitlements
If enabled, the entitlements signin form will be shown for unentitled users

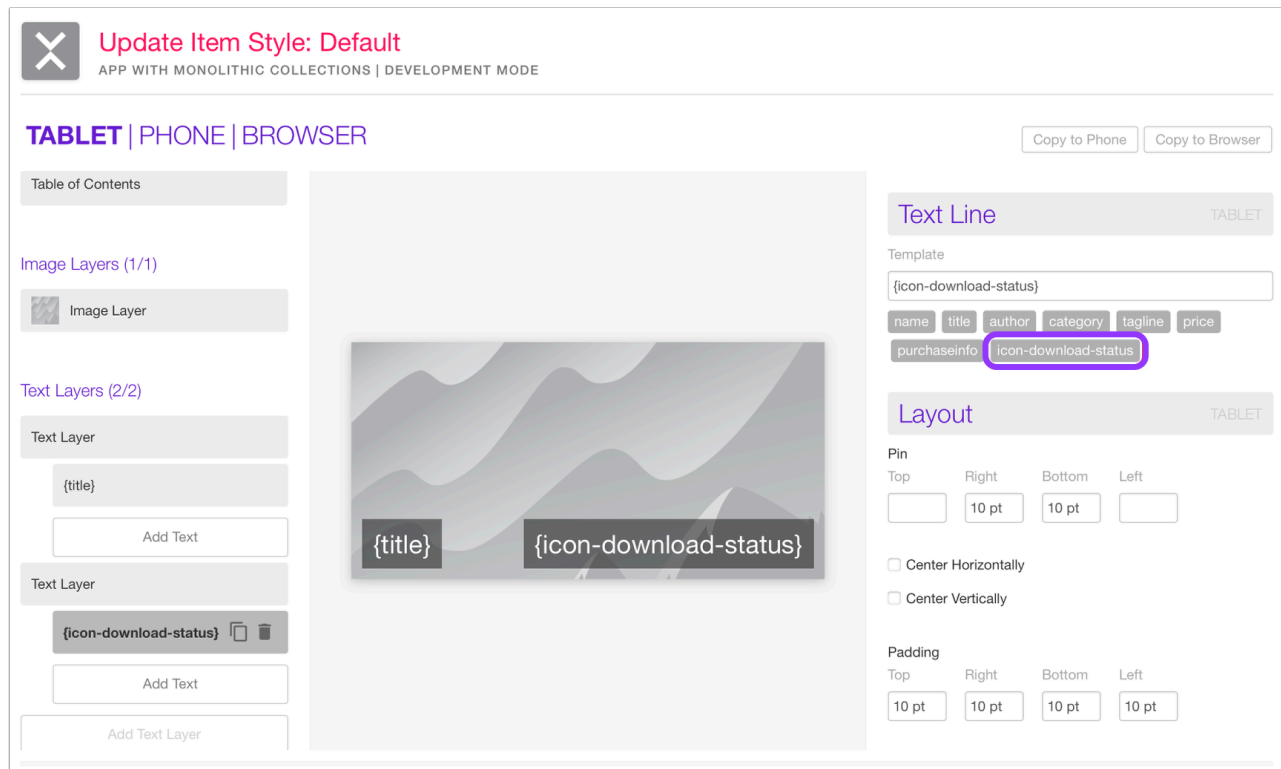
Note that if the option **Keep All Data Offline** in your *Application Settings* is active (see below), this option will not appear in your collection settings, as it is irrelevant in that case.

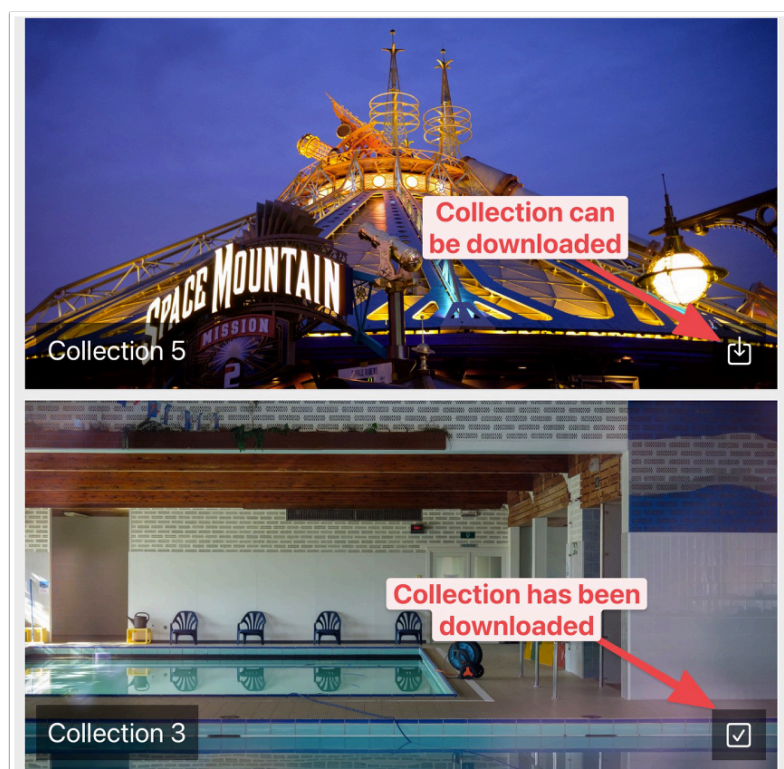
1.3. How can the reader use this?

1. For collections that have the option 'Monolithic download' checked, the process will be fully transparent, and a progress bar will be displayed during the download.
2. Monolithic downloads are never automatically deleted by the system and stay on the device. To remove the downloaded collection, a user can long-press the item in the browse grid, then delete the content for that collection. It can always be re-downloaded later.
3. You can use the custom URL scheme [tp-downloads://](#) to allow users to get an overview of all downloaded collections.

1.4. How to display a download status icon in your browse grid

If you want to make it clearer for a user whether a collection is ready to download or has already been downloaded, you can include the placeholder `{icon-download-status}` in your item style.



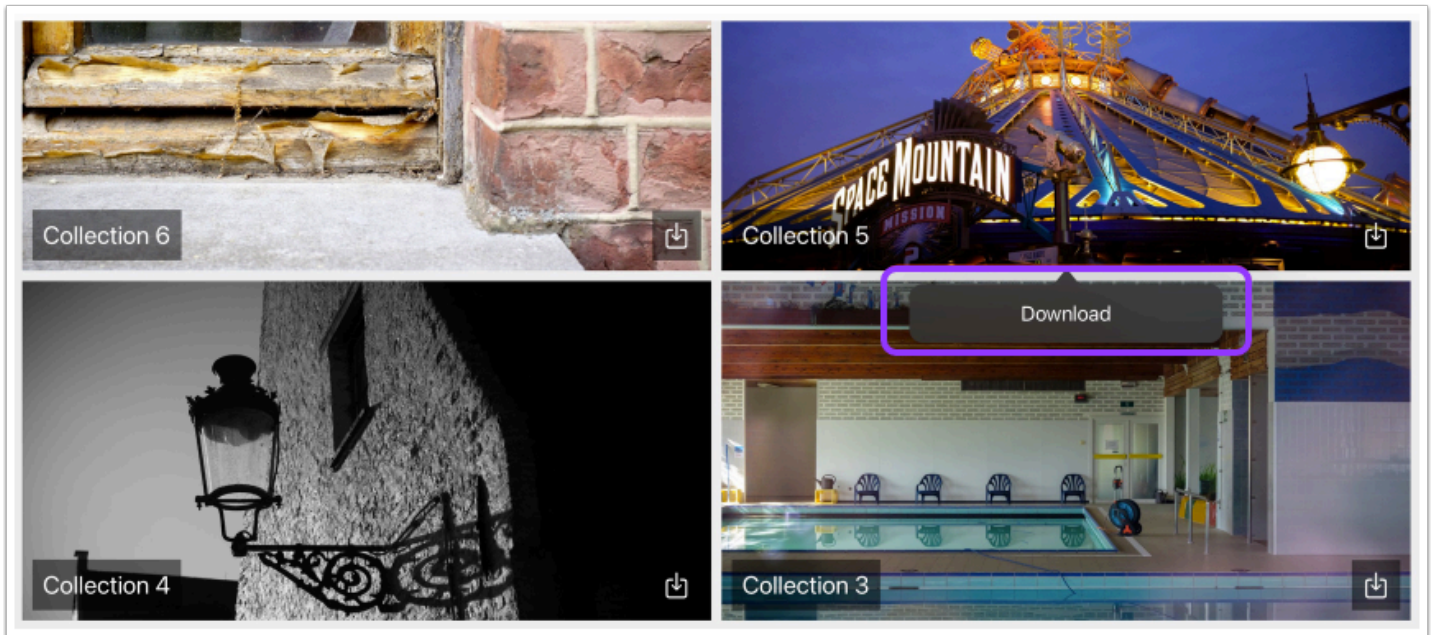


IMPORTANT NOTE:

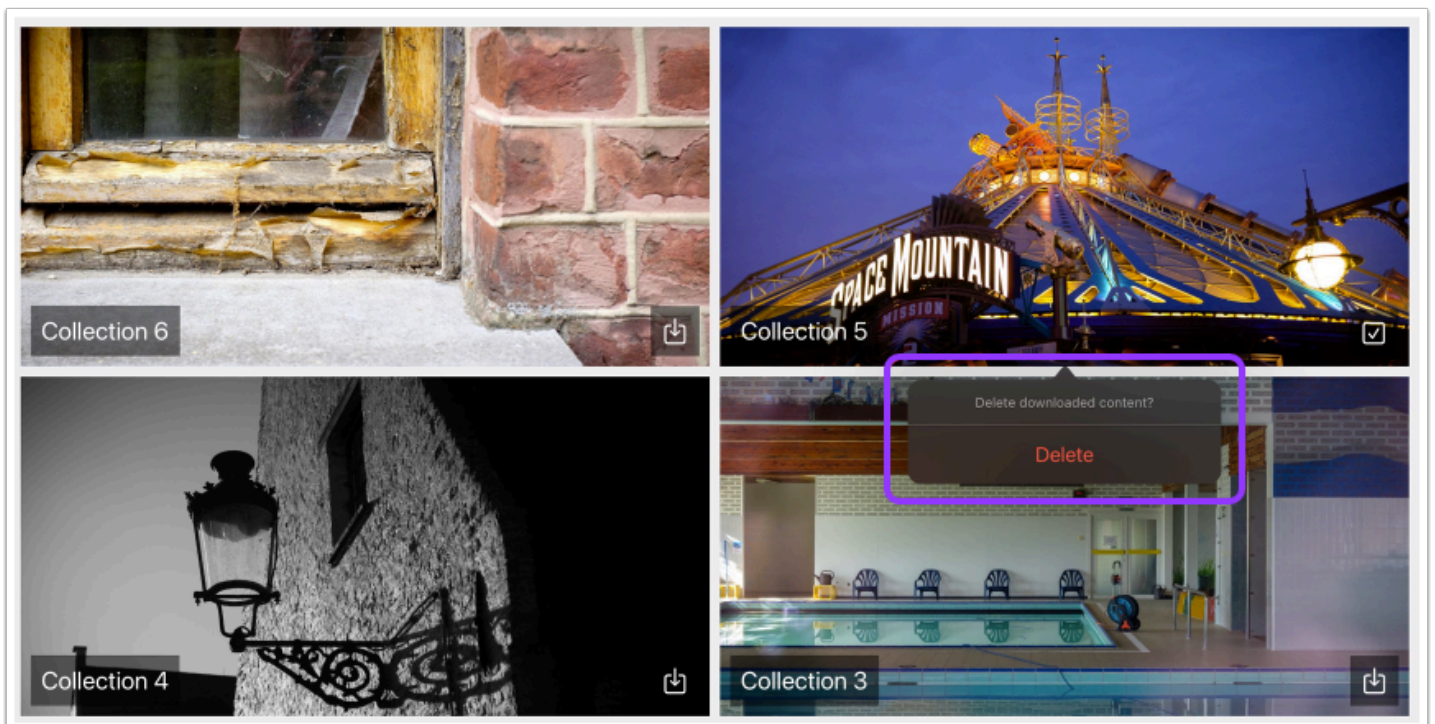
When downloading a monolithic collection, the content items that are downloaded will only go one level deep. This means that, if the collection contains e.g. collection links (to other collections), these will not be downloaded.

2. Download a collection on demand

Using long press on any collection, whether monolithic or not, will always allow download of the full collection for offline access.



Long-pressing a collection that has already been downloaded will show the option to delete the downloaded content from the device. It can always be downloaded again.



⚠ IMPORTANT NOTE:

Similar to how a monolithic collection is treated, the content items that are downloaded will only go one level deep. This means that, if the collection contains e.g. collection links (to other collections), these will not be downloaded.

3. Full Offline Mode

We provide an app setting (i.e. set by the publisher), where all content of all collections will be downloaded completely when first starting up the app. This is ideal for sales people who sometimes don't have an internet connection when they are on the road. Every time a user goes back to the app and he's online, a check will be performed if there is new and/or updated content that needs to be downloaded.

WARNING:

Requires users to download all app contents on first startup. Ideal for certain in-house apps, but not recommended for App Store apps, as such an app will most likely be rejected by Apple.



3.1. How to activate this (as a publisher)?

1. Create a build of your app. Test the new build thoroughly.
2. Activate **Keep All Data Offline** in the *Application Settings* of your app.

Application Behavior

☒ Prompt for App Store Rating
Prompts the user to rate your app in the App Store.

☒ Keep All Data Offline
Requires users to download all app contents on first startup. Ideal for certain in-house apps, but not recommended for App Store apps, as such an app may be rejected by Apple.

☒ Sharing on Social Media
When allowing sharing on social media, users can share articles on social media.

☐ Encrypt PDF Files
Enables on-the-fly encryption to PDF content items.
This requires a reader application built with version 5.6 or newer to be able to read the PDF files.

☐ Externally Managed Content
Enable this when you have an external system that manages the content.
This puts the content items and collections of your app in read-only mode on the TwixL Distribution Platform.

3.2. How can the reader use this?

First of all, the reader needs to download the latest version of your app. Every time a user goes back to the app and he is online, a check will be performed whether there is new

and/or updated content that needs to be downloaded. When new content is available, a dialog will force the user to download the latest content. When the content has been downloaded and is available offline, the app can be used.

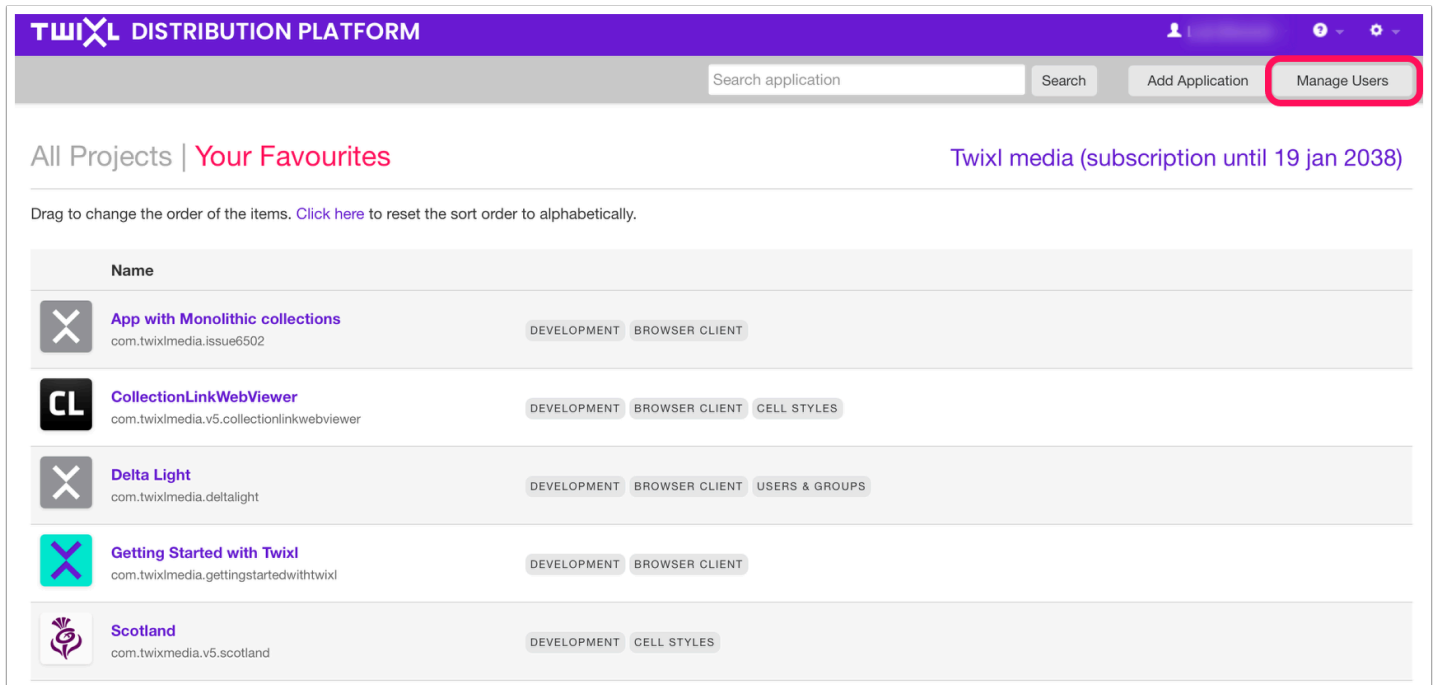
Manage Users

You can add different types of users to the Twixl Distribution Platform. Here, we explain how to do that and which types can be created.






Managing Users

To manage users, you can do one of the following things:

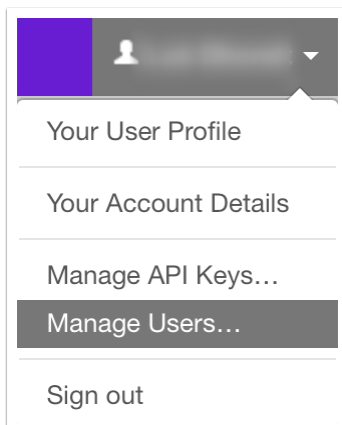
- Go the Homepage of our Twixl Distribution Platform and click on *Manage Users*.



The screenshot shows the Twixl Distribution Platform interface. At the top, there is a purple header with the Twixl logo and the text 'DISTRIBUTION PLATFORM'. Below the header, there is a search bar and buttons for 'Search', 'Add Application', and 'Manage Users'. The 'Manage Users' button is highlighted with a red box. Below the header, there is a section for 'All Projects | Your Favourites' and a link to 'Twixl media (subscription until 19 jan 2038)'. Below this, there is a table of applications with columns for Name, Development, Browser Client, and Users & Groups. The table lists five applications: 'App with Monolithic collections', 'CollectionLinkWebViewer', 'Delta Light', 'Getting Started with Twixl', and 'Scotland'.

Name	Development	Browser Client	Users & Groups
 App with Monolithic collections com.twixlmedia.issue6502	DEVELOPMENT	BROWSER CLIENT	
 CollectionLinkWebViewer com.twixlmedia.v5.collectionlinkwebviewer	DEVELOPMENT	BROWSER CLIENT	CELL STYLES
 Delta Light com.twixlmedia.deltalight	DEVELOPMENT	BROWSER CLIENT	USERS & GROUPS
 Getting Started with Twixl com.twixlmedia.gettingstartedwithtwixl	DEVELOPMENT	BROWSER CLIENT	
 Scotland com.twixlmedia.v5.scotland	DEVELOPMENT	CELL STYLES	

- Or go to your Account Name and select Manage Users... from the dropdown menu.



The screenshot shows a user account dropdown menu. The menu is open, showing options: 'Your User Profile', 'Your Account Details', 'Manage API Keys...', 'Manage Users...', and 'Sign out'. The 'Manage Users...' option is highlighted with a dark background.

Define different user roles

Home » Users » Select User Type

Select User Type

1 Administrator All Access	2 App Administrator Platform & Twixl App Reviewer	3 App Reviewer Twixl App Reviewer	4 App Report Viewer Allows access to app reports
--	---	---	--

Cancel

Different types of users can be created on the Twixl Distribution Platform:

1. **Administrator** – has full access to everything and all apps under that account, both in the back end and in the App Reviewer
2. **App Administrator** – has access to selected apps only, both on the Twixl Distribution Platform and in the Twixl App Reviewer
3. **App Reviewer** – can access selected apps in the Twixl app only. Has access to the Twixl platform to scan QR-codes. See [this](#) KB-article.
4. **App Report Viewer** – can access the Reports (analytics) of selected apps on the Twixl Distribution Platform. Has no access to the Twixl App Reviewer app and can't see any settings for the selected apps.

Two-Factor Authentication – 2FA for users.

2FA adds an extra layer of security to your user account on the Twixl platform.

What is 2FA?

Two-Factor Authentication (also known as *2FA*) is all about **security**. It's an extra layer of security that is known as multi factor authentication that requires not only a password and a username but also something else. In other words: it is the authentication process where two of the three possible factors of authentication are combined.

The possible factors of authentication are:

1. Something the user knows (e.g.: a password, PIN code, an answer to a secret question, ...).
2. Something the user has. (e.g.: a token, a mobile phone, a USB, ...).
3. Something the user is (e.g.: face or voice recognition, fingerprint, ...).

For internet security, the most common used combination is: `1 + 2`. And that's what we use on the Twixl Distribution Platform.

Why should I activate 2FA?

2FA provides a secondary layer of security that makes it far more difficult for hackers to access a person's devices and online accounts. It reduces the risk of exposure when your password is stolen (or compromised) or your email account has been compromised.

How to activate 2FA?

1. Login on the Twixl platform
2. Go to your User-account and click on your Username in the menu.
3. Select the first item in that Dropdown-menu: *Logged in as:* `username`.
4. Select Two-Factor Authentication.
5. Follow the instructions!

Possible 2FA-apps

We certainly have some favourites:

- [Google Authenticator](#): basic but decent.
- [IPassword](#): a wonderful password manager with support for 2FA on all platforms.
- [Authy](#): very good 2FA app, available on all platforms.

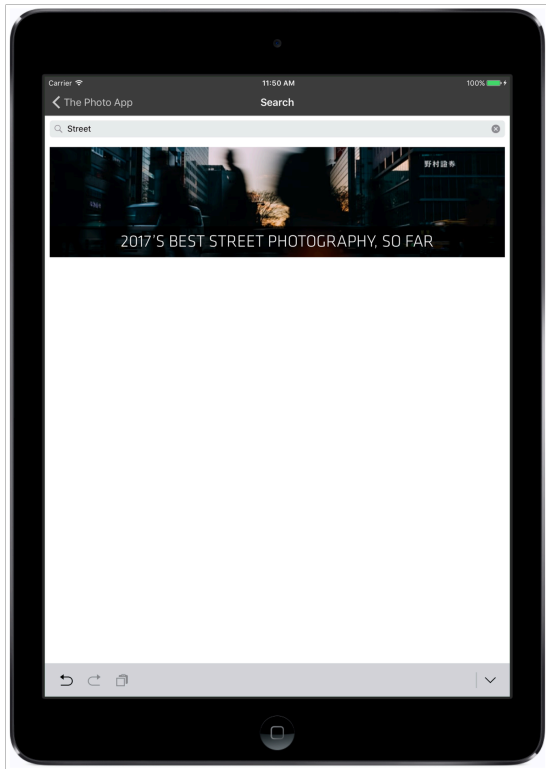
GOOD TO KNOW:

- A good password remains important. It's your first and most important layer of security. Don't neglect it!
- Activating 2FA on the Twixl platform is not a requirement, but we strongly advise you to do so.
- At this moment, 2FA can't be activated for all users at once (for the same *Account*). Activating 2FA is the choice of a user.
- You should think about activating 2FA on other (non-Twixl) accounts as well. The more accounts have a better security, the less chance you have in creating a chain-reaction when one important account is compromised.

Security is the responsibility of both service providers and users!

Searching in your app

You can provide a search capability through all metadata and indexed text in your app (content item name, title, tag line, etc.). From the search results, a user can quickly navigate to a relevant article.



1. Provide a search option in your app

If you want to allow users to search in your app, you will need to provide them a way to activate the search screen. You can do this by using a special url scheme `tp-search://`, e.g. in a ['web link' content item](#). Tapping the web link will then display a search bar. More info about this URL scheme can be found [here](#).

Web Link

Web Link URL *

tp-search://

The URL to open in the embedded web browser.

Web Links are limited to 255 characters. If you need Web Links with more charact

Thumbnail

Select File

Select any JPEG or PNG file which be used as the thumbnail for the content item.

2. Determine the look & feel of the search results

You can change the layout of the search results by using familiar Twixl platform options:

- The overall layout is defined by a [Grid Style](#) of your choice.
- The layout per Content Item is defined by one or more [Item Styles](#) of your choice.

If you go to your app and select 'Edit app', you can select a grid and item style that will be used to display the search results.

Search / Library / Downloads

Grid Style

Default

-- select grid style --
Default
Hamburger

Search Mode

Search in the metadata of the content items or search in the metadata and the actual contents of the content items.

Item Style

Default

Tablet: 8 columns | 2 rows
Phone: 1 column | 1 row

Search by Content Item Type

All Content Types
PDF
HTML Articles
Twixl Articles

Multiple types can be selected.

The grid and item styles used for search are the same as what you can use for the downloads overview, see [Custom URL schemes for Twixl apps](#).

TIP:

We strongly advise you to use **separate Grid and Item Styles** for your Search and Downloads results screen. Also, give them a **good and logical name** (e.g. Search

& Downloads Grid Style). That way it will be much easier to organise and select different styles in different Collections / Content Items of your app.

3. Search Mode

Search / Library / Downloads

Grid Style

Default

Tablet: 8 columns | Vertical scrolling
Phone: 2 columns | Vertical scrolling

Item Style

Default

Tablet: 8 columns | 2 rows
Phone: 1 column | 1 row

Search Mode

Full-Text Search

Search Metadata Only

Full-Text Search

Search by Content Item Type

All Content Types

PDF

HTML Articles

Twixl Articles

Multiple types can be selected.

You can choose between two options:

- Search in the metadata of the content items only
- Search in the metadata as well as in the actual content of the content items.

IMPORTANT:

Since this functionality also depends heavily on **metadata**, it is highly recommended that **a publisher enters the relevant metadata** in all *Collections* and *Content Items* of the app. Without that, a search request can show incomplete results. So, don't forget to fill in Title, Author, Tagline, etc.

4. Search by Content Item Type

Too many search results or a specific app setup that requires searching in just PDF's? Then you can narrow down the search feature by selecting the content item type(s) in which the app has to look for hits.

By default, all content item types will be searched. But by clicking the dropdown menu, one can choose one or more of these content item types:

- HTML Articles
- Twixl Articles
- PDF's

Search by Content
Item Type

All Content Types

PDF

HTML Articles

Twixl Articles

Multiple types can be selected.

- i** "All Content Types" searches in all possible content items (movies, images...). Whereas selecting "PDF, HTML Articles and Twixl Articles" will not search in movies or images but will only limit the search to PDF, HTML Articles and Twixl Articles.

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.

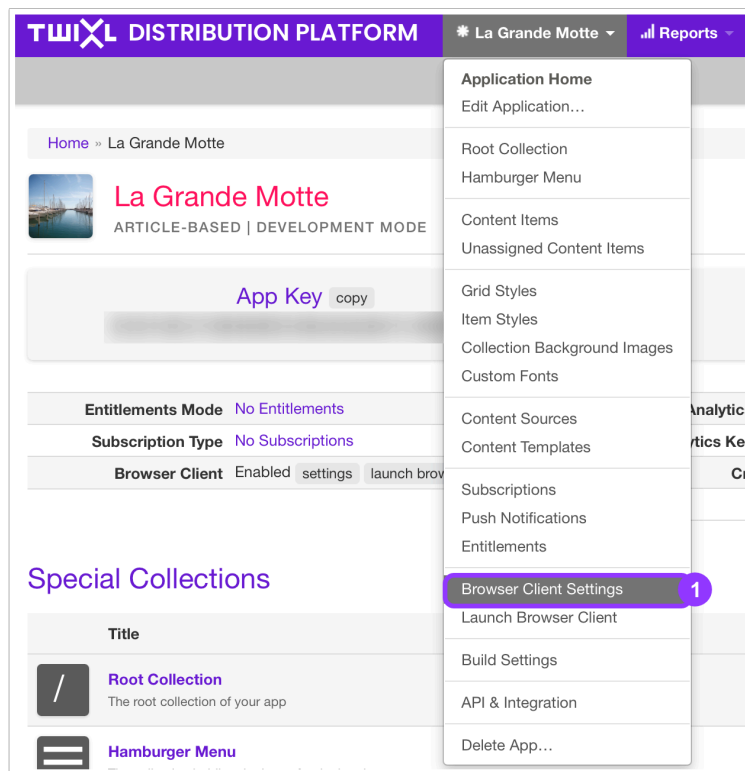
ABOUT THE PURPOSE OF THE BROWSER CLIENT:

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.

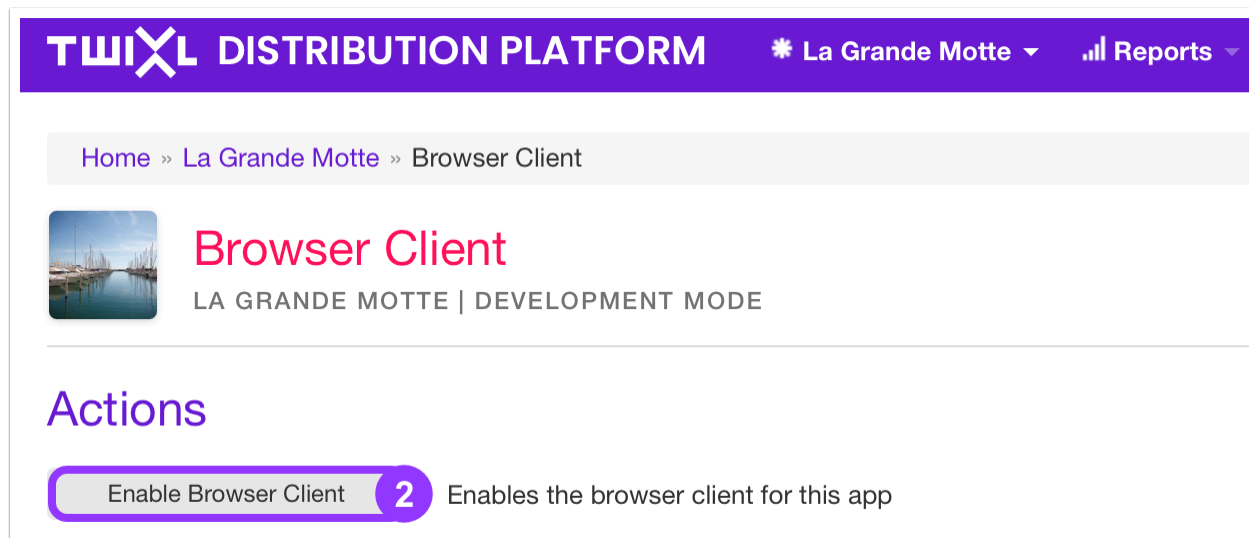
How to activate the Browser Client?

To activate the Browser Client option for your app:

- Go to the menu of your app > select 'Browser Client Settings' (1)



- Click on 'Enable Browser Client' (2) to enable the Browser Client for this app.



Available options after enabling the Browser Client

The screenshot shows the Twixl interface for managing the Browser Client for an app named 'La Grande Motte'. At the top, there is a breadcrumb trail: Home » La Grande Motte » Browser Client. Below this, there is a header section with a small image of a boat and the title 'Browser Client' in red, followed by 'La Grande Motte | Development Mode'. A URL field displays 'https://app-' followed by a blurred domain and a 'copy' button. Under the 'Actions' section, there are four numbered items: 1. 'Try Browser Client' (Try the browser client with this app), 2. 'Settings' (Edit the settings of the browser client for this app), 3. 'Disable Browser Client' (Disables the browser client for this app), and 4. 'Pre-render Content' (This prepares all your content for fast viewing in the browser client). Below this, the 'Pre-render Status' section indicates 'All content items 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 this particular 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

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`.

Custom Domain Name

Custom Domain

Needs to be a CNAME of **browserclient.twixlmedia.com**.

An HTTPS certificate will be created automatically.



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 ease 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](#)

Other Browser Client Settings

Other settings

- 1** Google Analytics Key
The Google Analytics Key to use for the browser client
- 2** Apple App ID
The unique ID of your app on App Store Connect

1. **Google Analytics Key:** You can add a Google Analytics property in this field. For details on how to setup such a property, you can check the article [Google Analytics](#).
2. **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. Next, copy-paste that number in your Browser Client *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.

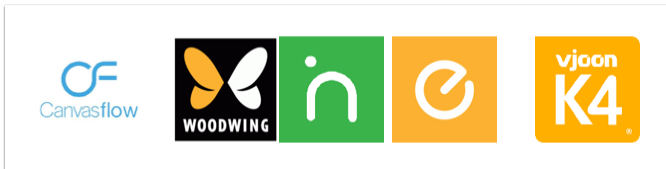
- [More info about Browse Grid Styles](#)
- [More info about Item Styles](#)

Integration API

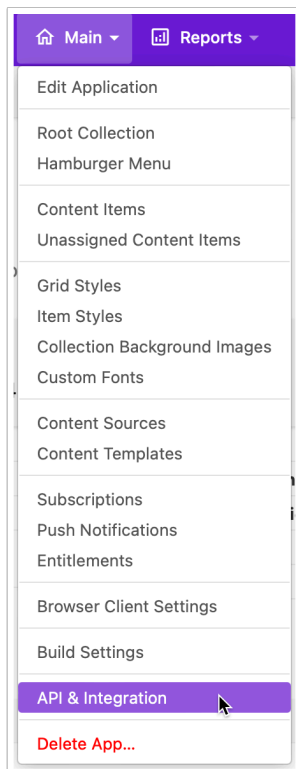
The *Integration API* is an extra paid option that allows you to use the Twixl Distribution Platform Admin API to push content from external CMS systems.

Twixl Publisher offers the ability to integrate with different solutions for automated production flows and so to push content to the Twixl Distribution 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 Distribution Platform. Some custom development or help from your local integrator may be required to achieve this.



Setup officially supported external services



To integrate with an external solution, navigate to the *API & Integration* section from the main app menu.

There, you will find:

1. The unique App Key
2. Admin API Key management for your app
3. API documentation
4. Links to the tutorials of the officially supported external services, being:
 - Woodwing Inception
 - Woodwing Aurora
 - Canvasflow
 - Vjoon K4

Other solutions

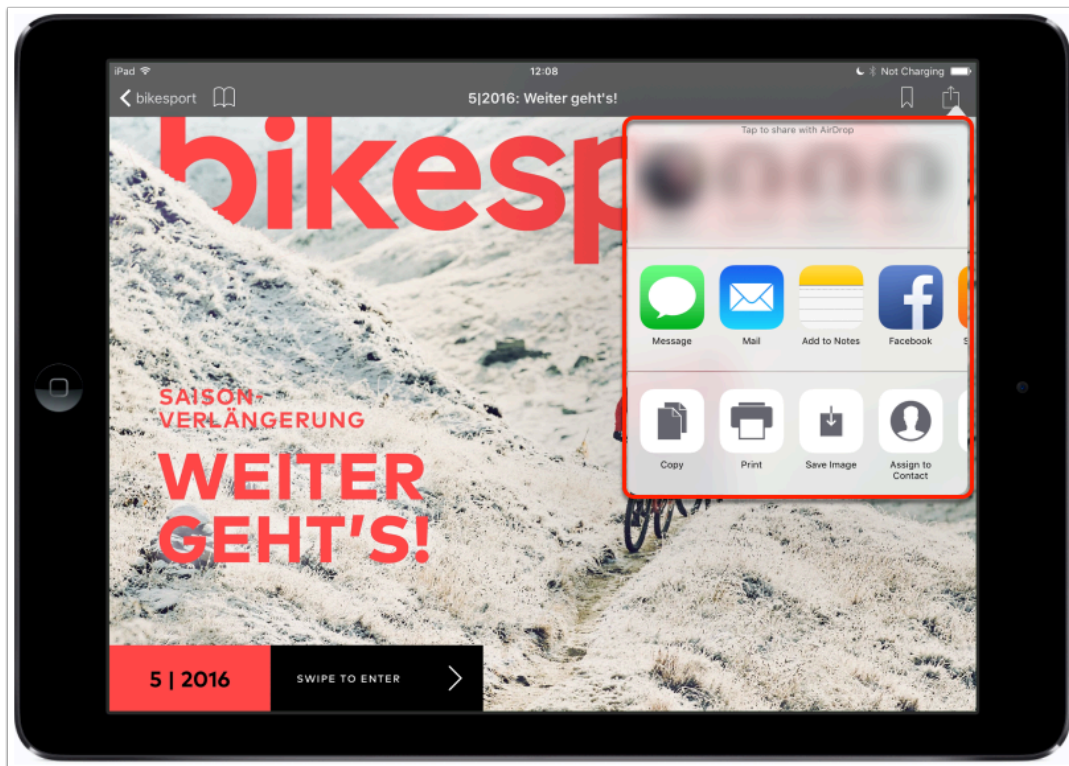
While the solutions above are already pre-configured to be used with Twixl's Integration API, and can be used more or less 'out of the box', you can basically push content from any other external CMS solution, such as WordPress or Drupal-based web sites, using the Integration API documentation on the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



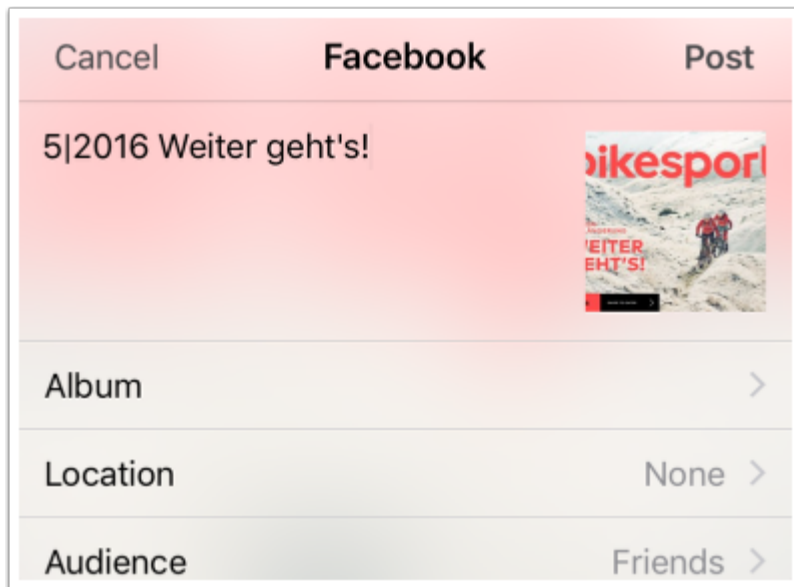
Sharing on social media

Social sharing allows the reader to share an article via social media or default mobile device sharing options. This article explains how to prepare both your app and content for that feature.

What does the reader see?



Readers will be able to share via e-mail, Twitter, Facebook, ... (depending on the OS and the installed applications). The result will be similar to the screenshot below.



How to configure for your app

1. Activating the option on the Twixl platform

There is only 1 requirement to enable Social Sharing in your app: you need to activate the option in the Application Settings of your app. For more info, see [this article](#).

2. Configuring Social Media metadata

Social Sharing works for Content Items. Each Content Item can have specific metadata for your Social Sharing.

Home » La Grande Motte » Root Collection » Oysters » Update

Update Content Item: Oysters
La Grande Motte | Development Mode

Content Item Details

Content Type:

Name:

In Collection:

Item Style:

Tablet: 2 columns | 1 row
Phone: 4 columns | 2 rows
Web: 2 columns | 1 row

Metadata

Title:

Author:

Tag Line:

Category:

Published On:

If left empty, the content item will get the status "Unpublished".
If filled in, the content item will get the status "Published" or "Scheduled".

Social Media

Social Sharing Text:

The default text you want to use for social sharing.

Article URL:

In order to be able to share over social media, an article URL is required.
Leave this field empty to use the browser client URL.

2.1. Available options

- **Text for Social Sharing:** The default text you want to use for social sharing. This text will be inserted for your readers when he/she wants to share an article.
- **Article URL:** In order to be able to share over social media, an article URL is required. Leave this field empty to use the Browser Client URL (if you have enabled the browser client for your app). For more info about the Browser Client, see [this article](#).

2.2. Possible scenarios

- **No Article URL defined + Browser Client enabled:** the reader will be able to share the article, with the *Browser Client URL*.
- **Custom Article URL defined + Browser Client enabled or disabled:** the reader will be able to share the article, with the *custom Article URL*.
- **No Article URL defined + Browser Client disabled:** your reader will not be able to share the article.



REQUIRED FIELDS:

As a publisher, it is not required that the text for Social Sharing is filled in, in order to activate Social Sharing. Only the Article URL is mandatory.

Optimizing PDF files for viewing on mobile devices

PDF files can be uploaded directly to your app on the Twixl Platform. You can even mix InDesign .article files, HTML articles and PDF files in the same app.

TIPS:

How your PDFs have been generated is not important by itself, but there are a few things to take into account when creating these PDF files:

- If you export from Adobe InDesign, use the "Twixl Publisher PDF" export preset that has been installed automatically when you installed the plug-in.
- Make sure the PDF is an RGB PDF as CMYK PDFs will display very strange colors on mobile devices.
- Make sure the PDF doesn't use any ICC profiles as these are not supported on mobile devices.
- It's preferred not to flatten the PDF but to retain full transparency in order to avoid output artifacts and large file sizes.
- Please note that the Twixl Platform will not fix any of these problems automatically in PDF files that are uploaded.
- By using tools like [Enfocus Switch](#) in combination with [Enfocus Pitstop Server](#), it is possible to automate the whole process of optimizing the PDFs and uploading the files to the Twixl platform.

About file size and Twixl Publisher

A few thoughts on the importance of file size...

Reasons to keep content small

- **Download time:** Nobody wants to wait a long time. If readers need to wait too long, they might get annoyed and cancel the download. They might even delete your app, because of the huge download time... Keeping content file size small is all about keeping readers satisfied. See the comparison table below for average download times based on the download speed.
- **Device space:** Not every reader has that state of the art device with 250GB of storage. In fact, a lot of readers still have devices with 16GB or 32GB of storage. So, if you want them to read your content, understand that readers need to be selective in terms of saving content on their devices and reader don't want to clean up content all the time...

Average Download Times

Size (MB)	4Mbps	8Mbps	16Mbps	32Mbps	50Mbps	100Mbps
5	10s	5s	2,5s	1,25s	0,8s	0,4s
10	20s	10s	5s	2,5s	1,6s	0,8s
50	1m 40s	50s	25s	12,5s	8s	4s
100	3m 20s	1m 40s	50s	25s	16s	8s
450	15m	7m 30s	3m 45s	1m 52s	1m 12s	36s
700	23m 20s	11m 40s	5m 50s	2m 55s	1m 52s	56s
1500	50m	25m 30s	12m 30s	6m 15s	4m	2m



IMPORTANT NOTE:

Download times are average download times. Actual speed is often influenced by a lot of additional factors. Realistically, the times above should be multiplied by 4 or 5.

Tips to keep your content small

- **Movies** are a lot of fun. Make sure your movies are optimized and – if possible – avoid embedding movies in your app. Use streaming instead. See the following articles for more info about using video:
 - [Video](#)
 - [Working with Content Items](#)
 - [Content Sources](#)
- Optimize your **PDF-files**. That's not only important for colors (cmyk vs. rgb) but also very important for the file size: downsampling pictures is a must! See [Optimising PDF's for viewing on mobile devices](#).
- **Custom Fonts** are a nice way to give your article-based application a nice design. But make sure you only upload fonts you really plan to use. All uploaded fonts will have to be downloaded in the app when the reader starts it up for the first time. [More info](#).
- Check all your other resources and make sure your `WebResources` folder doesn't contain any files that aren't used:
 - [Animations](#)
 - [Web Viewers](#)
 - [Web Overlay Buttons](#)
- The most important thing remains: **Testing**. Test your creations, test your applications, test the content etc. And then test again and once more to be sure.

ABOUT THE SIZE OF YOUR TWIXL APP:

Did you know we have a handy tool to investigate how big your app is (in terms of file size)?

You can read all about it [here](#)!

ABOUT WEBRESOURCES:

For more info about the `WebResources` folder, [see this KB-article](#).

Deleting an app on the platform

An app on the Twixl platform can be deleted. Here's how to do it, but make sure to first check the consequences of this procedure!

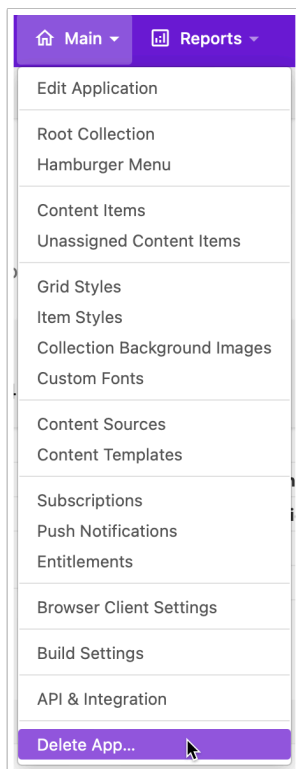
How to delete an app on the Twixl platform

! WARNING:

This process is irreversible! Proceed with extreme caution!

To delete an app on the [Twixl platform](#), you need to:

1. Login on the [Twixl platform](#)
2. Navigate and select the app you want to delete
3. Select the *App Menu* of the app
4. Select **Delete...** in the available list of options
5. Read and confirm the warnings



Consequences

! CONSEQUENCES OF DELETING AN APP:

If a user decides to delete an app, the Twixl platform will delete all info related to the deleted app:

- Content
- Analytics
- Settings
- ...

Due to GDPR-regulations, **this process is irreversible!** Depending on the size of the app, this process will take about 1 minute maximum.

As a result, readers will no longer have access to the content of your app.

Working with custom "DIY" storage

Publishers that use the Twixl platform, and that have very high download volumes, have the option to use their own storage account on **Amazon S3** or **Microsoft Azure**.

⚠ IMPORTANT NOTE:

DIY storage is only available as an option for subscribers of the Twixl Publisher Unlimited or 10+ plans. Also, if you use DIY storage, you do not have the benefits of using the Akamai Content Delivery Network (CDN).

1. How to configure for Amazon S3

If the DIY option has been activated for your account, you will be able to select the storage type when adding new apps. Select Amazon S3, then enter the Acces Key and Secret Access Key for your account, along with the region where you want the content to be hosted.

Application Details

Application Name *	<input type="text"/>	Storage Type *	<div>Amazon S3</div>
Application Identifier *	<input type="text"/>	Storage type cannot be changed after the app has been created	
This should be a reverse DNS name like: com.twixlmedia.appname. Only lowercase characters, numbers and dots should be used.		Access Key	<input type="text"/>
		Secret Access Key	<input type="text"/>
		Region	<div>-- select a region --</div>

2. How to configure for Microsoft Azure

If the DIY option has been activated for your account, you will be able to select the storage type when adding new apps. Select Windows Azure, then enter the Account Name and Account Key for your account.

Application Details

Application Name *

Application Identifier *

This should be a reverse DNS name like: com.twixlmedia.appname.
Only lowercase characters, numbers and dots should be used.

Storage Type *

Storage type cannot be changed after the app has been created

Account Name

Account Key

Metered Access

If your app has purchasable content and you want to offer a free preview of that content to the app user, you can use the metered access feature to make your life easier. Note that this feature requires TP19 or higher.

Principle

The metered access feature makes it easier to create links between preview and paid collections. The app will check whether the user has access to either the preview or the purchased collection. As a publisher you have complete control over the content that will be shown to the user because the linking is performed between collections. These collections can contain most types of Twixl content items*.

Workflow

1. [Enabling the metered access feature \(app setting\)](#)
2. [Defining collections as purchase or preview](#)
3. [Linking collections to a preview or purchase collection](#)
4. [Adding content to the collections](#)
5. [Adding purchase button](#)

1. Enabling the metered access feature


To be able to use the metered access feature, you have to enable this function in the app settings.

How:

Go to app settings via:

Main > Edit Application > Application Behavior > check the 'Metered Access' box > Save

Application Behavior

- ☐ **Prompt for App Store Rating**
Prompts the user to rate your app in the App Store.
- ☐ **Blur snapshot in iOS & Android app switcher**
Enable this option to hide potentially sensitive information in the app switcher screen.
- ☐ **Keep All Data Offline**
Requires users to download all app contents on first startup. Ideal for certain in-house apps, but not recommended for App Store apps, as such an app may be rejected by Apple.
- ☐ **Sharing on Social Media**
When allowing sharing on social media, users can share articles on social media.
- ☐ **Encrypt PDF Files**
Enables on-the-fly encryption to PDF content items.
- ☐ **Save read position in PDF files**
Save the last page read in PDF content items.
- ☐ **Externally Managed Content**
Enable this when you have an external system that manages the content.
This puts the content items and collections of your app in read-only mode on the Twixl Distribution Platform.
-  ☐ **Metered Access**
Enables metered access for collections.

2. Defining collections as purchase or preview

Once the metered access feature has been enabled, an extra 'preview' option appears when creating or editing a collection. When creating or editing a collection you have 3 options:

- Free of charge
- Purchase
- Preview

Collection Details

Collection Type	<div>Free of charge</div> <div> <div>Free of charge</div> <div>Purchase</div> <div>Preview</div> </div>
Name *	
Title	
Product Identifier	<div>7799715d1f2dd42e3f0af5299a5bd5da</div> <div>Leave empty to use the suggested identifier. Only characters, numbers and dots should be used.</div>
Published On	<div>10/01/2023 10:23</div> <div>This is used to determine if a collection is part of a subscription or entitlement.</div>

When a collection is marked as Preview, you will be able to link it to an existing Purchase Collection in Collection Details while creating the collection. A collection marked as Purchase will offer the possibility to link to an existing Preview collection via the dropdown. Linking the collection is not mandatory. You can link to a collection afterwards too. You can complete all collection fields as usual and save it.

Collection Details

Collection Type

Purchase

Free of charge: a free collection
Purchase: a purchasable collection
Preview: a preview collection for a purchase collection

Name *

Title

Product Identifier

Leave empty to use the suggested identifier.
Only characters, numbers and dots should be used.

Published On

This is used to determine if a collection is part of a subscription or entitlement.

Preview Collection

-- unassigned --

Select the collection to link to. A preview collection can only be linked to one purchase collection.

Whether a collection is free, preview or purchase is shown in the Type column in collection overview in Home or via Main > Collections. If there is already a collection linked to this collection, will also be shown in the 'Linked To' column in this overview.

Collections

Add Collection

	Title	Type	Options	Items	Grid Style	Linked To
<input type="checkbox"/>	<div>FREE</div> Free collection 1 didier.test.app.free_collection_1 Published 21 hours ago	Free of charge		1	Default	
<input type="checkbox"/>	<div>CONTENT</div> Other Content didier.test.app.othercontent Published 21 hours ago	Free of charge		3	Default	
<input type="checkbox"/>	<div>REVIEW</div> Preview collection 1 didier.test.app.preview_collection_1 Published 6 days ago	Preview		2	Default	
<input type="checkbox"/>	<div>PURCHASE</div> Purchase collection 1 didier.test.app.purchase_collection_1 Published 6 days ago	Purchase		1	Default	Preview collection...


3. Linking collections to a preview or purchase collection

Some possibilities to create links between collections:

- Via edit in a collection and select the matching collection
- By creating a new collection and selecting the matching collection

- By clicking on 'Create' in the Collection Details of a collection and creating a new collection

Collection details

Name	purchase collection 2	Created On	2023-01-10 10:32
Title	Purchase Collection 2	Published On	2023-01-10 10:31
Product Id	didier.test.app.purchase_collection_2	Monolithic Download	No
Twixl ID	27876	Requires entitlements	No
Twixl UUID	854aa44c2e367924abda165c116b89aa	Grid Style	Default
External ID	-	Open In	Browse Mode
Collection Type	Purchase	Preview Collection	Create 
Browser Client URL	https://browserclient-staging.twixlmedia.com/5b05a146eed6...		
Twixl Link	tp-collection://purchase%20collection%202		

4. Adding content to the collections

In Preview collections you can add your content as usual. For preview collections, the type of content items that are allowed are:

- HTML Article
- Twixl Article
- PDF
- Image
- Movie
- Vimeo Movie
- YouTube Movie
- Inline Web Viewer
- Embedded Web Viewer
- Placeholder
- Weblink

This means you are free to provide preview content that maximises purchase attraction.

5. Adding purchase button

How you integrate and design the purchase button is also completely customisable. If you create a Twixl article (InDesign), you can integrate the "tp-paywall://" scheme via the Twixl Plugin in your design. This will call the paywall function and once the content has been purchased, the app will only show the purchase collection. After the purchase of a pcollection has been completed, its preview collection will no longer appear.

Other content types will require to add a web link containing a "tp-paywall://" scheme.

Click [here](#) for more information on these custom url schemes.

**Final check:**

Before the go live of your publication, make sure both your content items both in the purchase and the preview collection are in the published status.

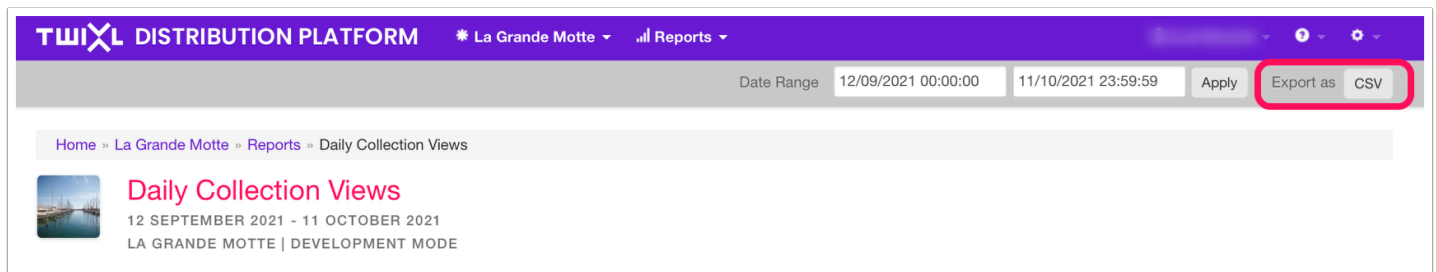
Analytics

Analytics Reports Overview

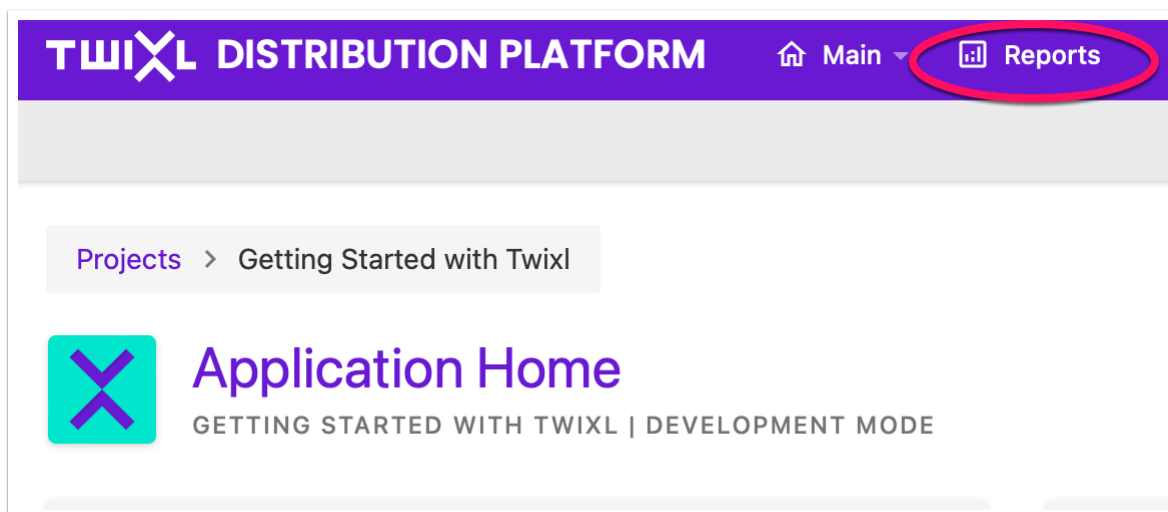
The Twixl platform offers built-in reporting of events that are tracked in your apps. You can export data from these reports and use the numbers into your own analytics presentations, tables, tools...

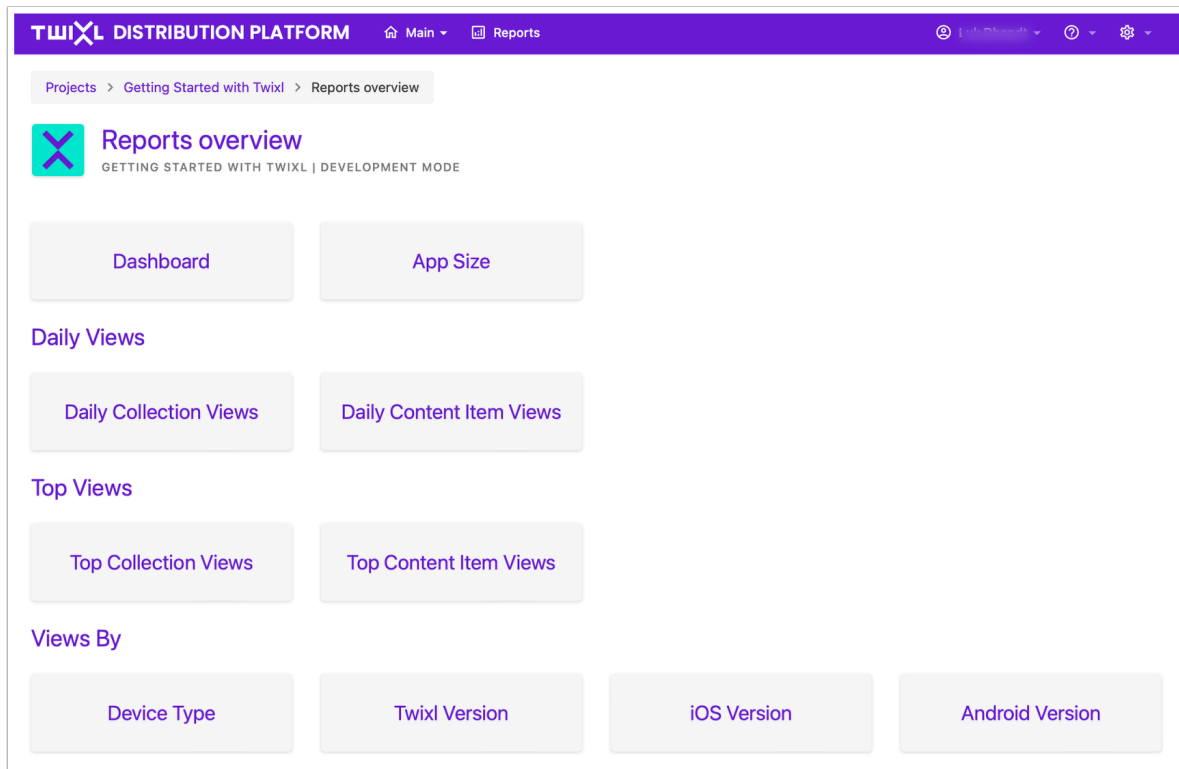
1. Exporting reports data

The data for all reports can be exported as comma-separated values (CSV) based on a selected date range. It is also possible to extract all event information via the Twixl Integration API (note, however, that the latter is a paid option).



2. Reports overview

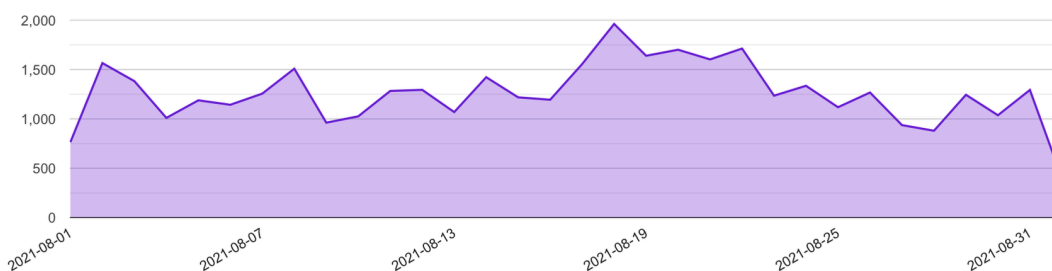




2.1. Daily views

2.1.1. Daily Collection views

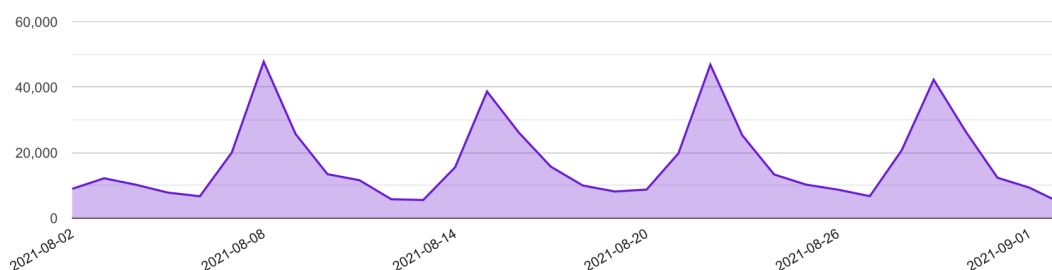
This report displays a chart with the number of views for collections per day. The report defaults to the last 30 days, but a date picker lets you change that. The list view below the chart displays the number of views per day, and the percentage it represents over the given period.



Date	# Views	% Views
01 september 2021	400	1 %
31 august 2021	1,293	3 %
30 august 2021	1,037	3 %
29 august 2021	1,244	3 %
28 august 2021	880	2 %
27 august 2021	937	2 %
26 august 2021	1,267	3 %

2.1.2. Daily Content Item views

This report displays a chart with the number of views for content items per day. The report defaults to the last 30 days, but a date picker lets you change that. The list view below the chart displays the number of views per day, and the percentage it represents over the given period.

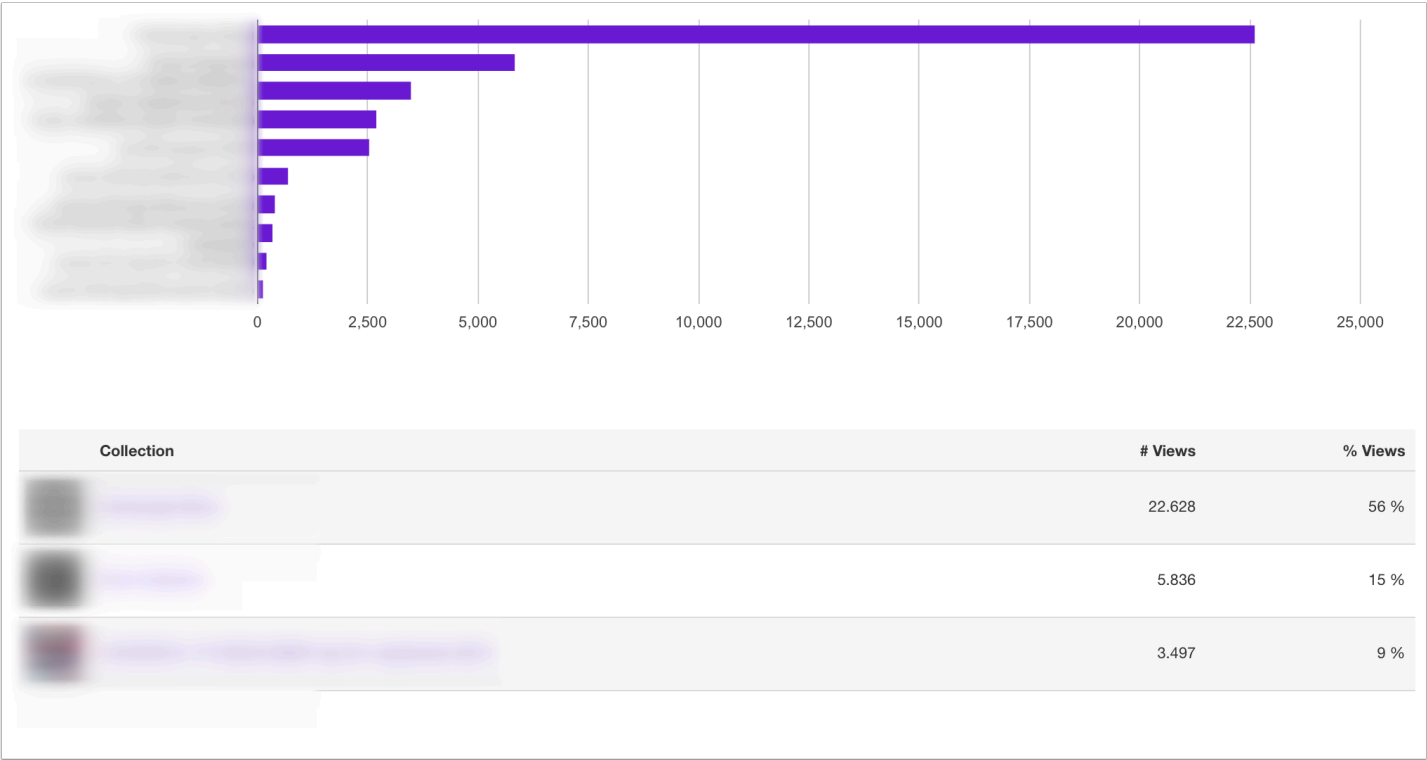


Date	# Views	% Views
02 September 2021	4,623	1 %
01 September 2021	9,339	2 %
31 August 2021	12,366	2 %
30 August 2021	26,510	5 %

2.2. Top views

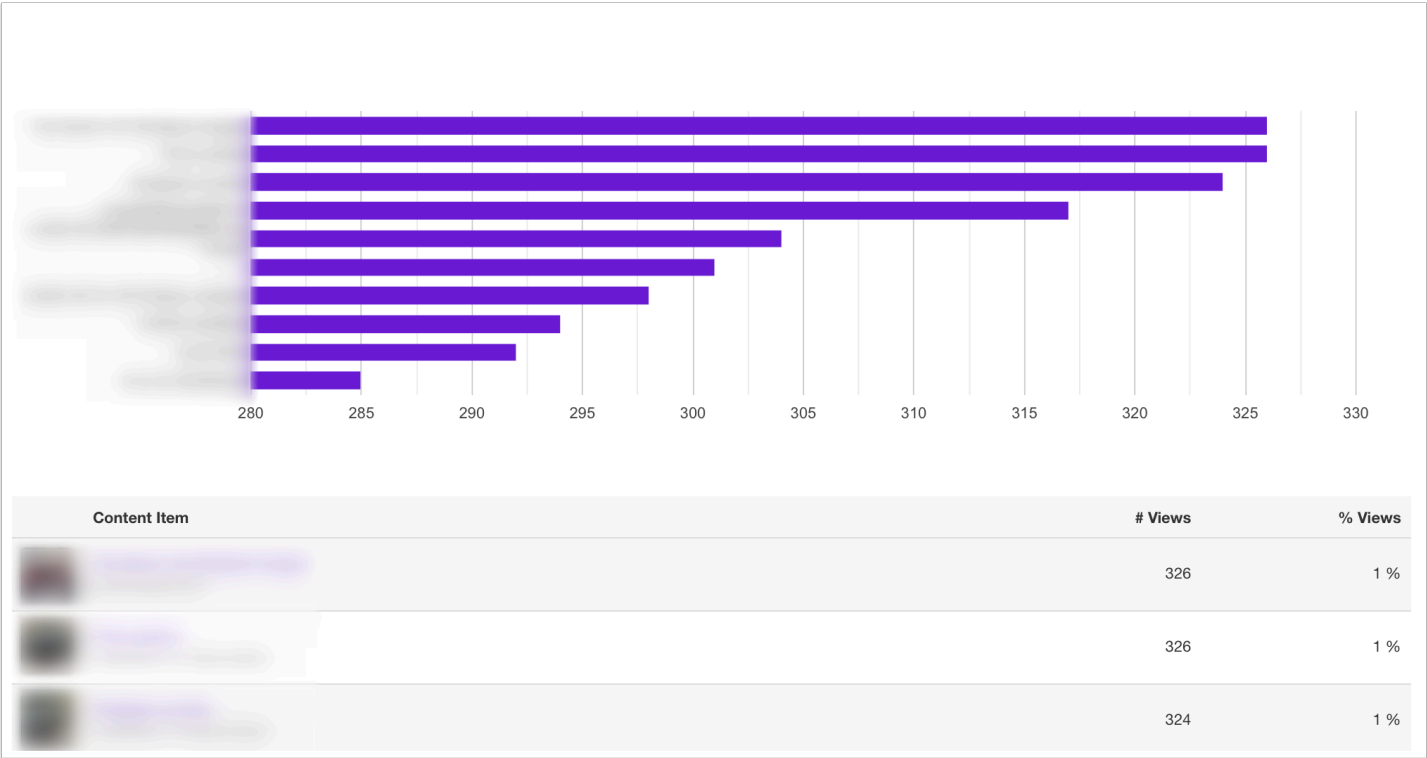
2.2.1. Top Collection views

This report displays a bar chart with the top 10 most popular collections for a given period. The report defaults to the last 30 days, but a date picker lets you change that. Below the chart is a list of collections by popularity, and the percentage they represent.



2.2.2. Top Content Item views

This report displays a bar chart with the most popular content items for a given period. The report defaults to the last 30 days, but a date picker lets you change that. Below the chart is a list of content items by popularity, and the percentage they represent.

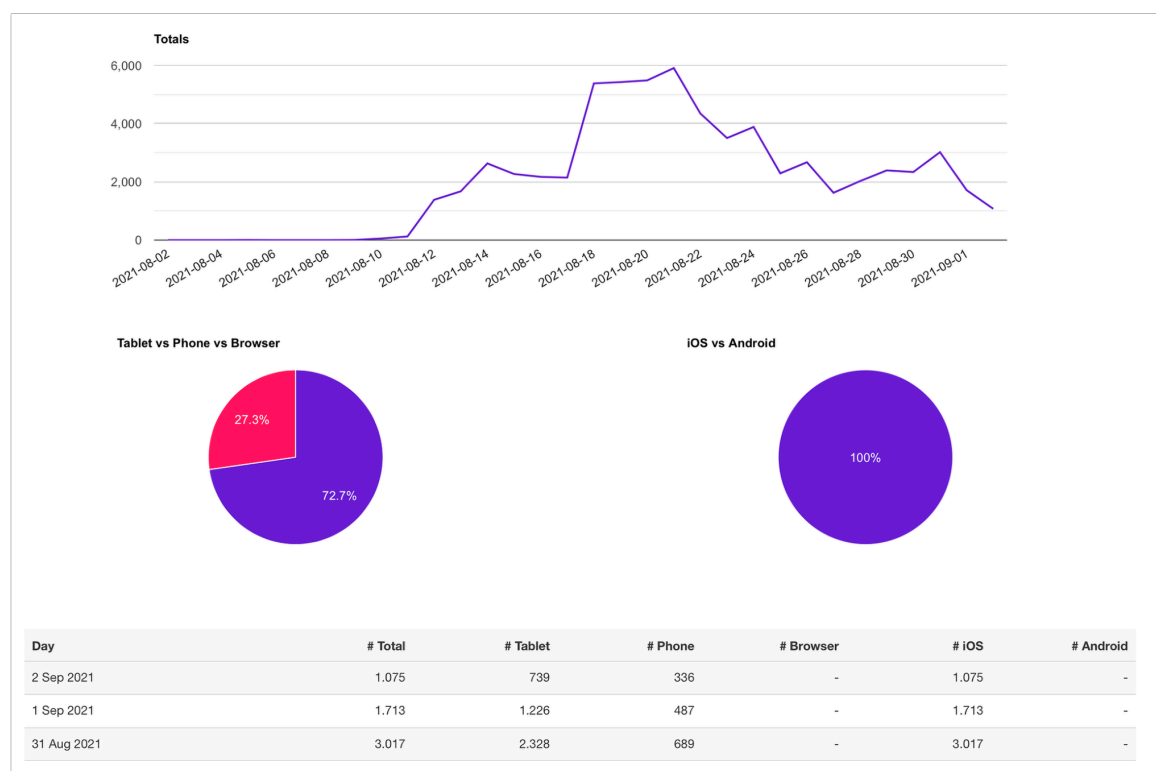


2.3. Views by ...

2.3.1. Device type

This report displays a line chart of the number of devices connecting to your app over a given period. The report defaults to the last 30 days, but a date picker lets you change that.

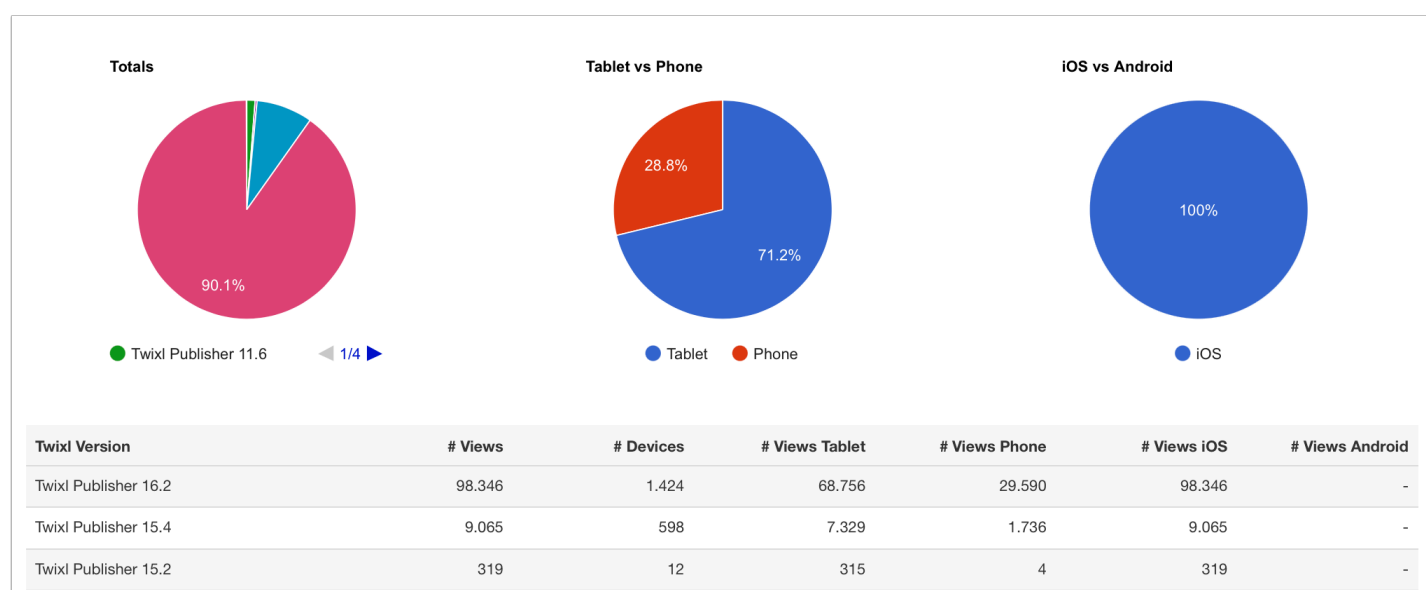
There are also two pie charts, one based on the percentages for tablet vs phone vs browser, another one on the distribution of iOS vs Android.



2.3.2. Twixl version

This report displays 3 pie charts of the Twixl versions that your users use over a given period. The report defaults to the last 30 days, but a date picker lets you change that.

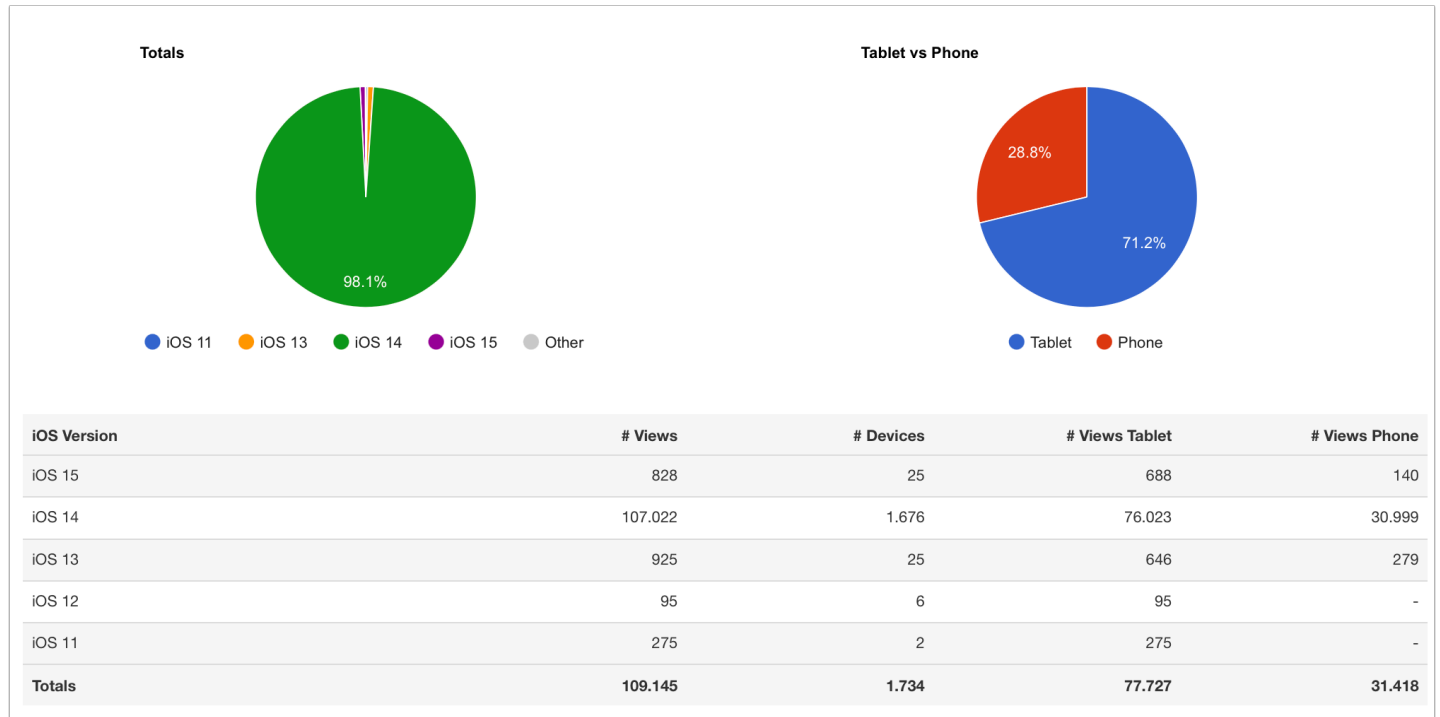
One chart shows the totals, a second one the comparison of tablet vs phone, and a third one the distribution of iOS vs Android.



2.3.3. iOS version

This report displays 2 pie charts of the iOS versions that your users use over a given period. The report defaults to the last 30 days, but a date picker lets you change that.

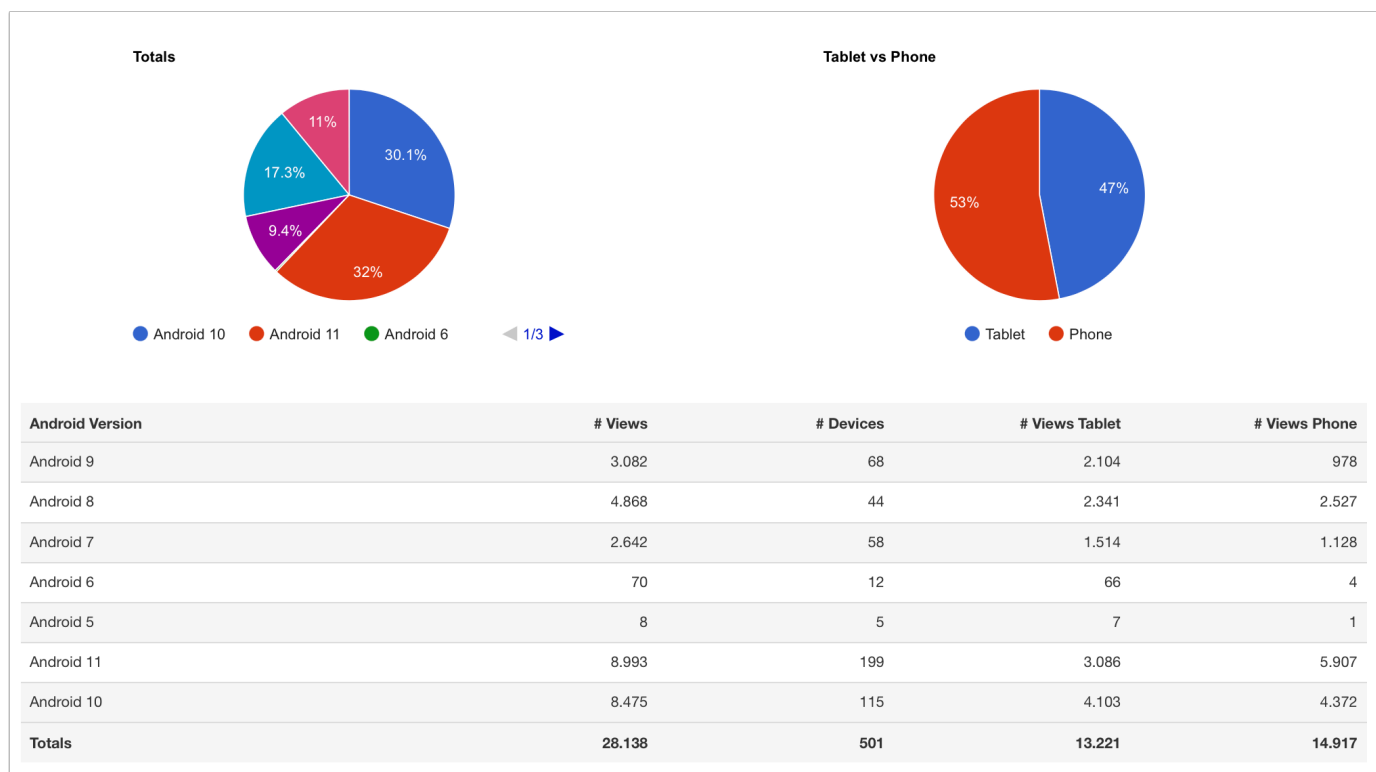
One chart shows the totals, a second one the comparison of tablet vs phone.



2.3.4. Android version

This report displays 2 pie charts of the Android versions that your users use over a given period. The report defaults to the last 30 days, but a date picker lets you change that.

One chart shows the totals, a second one the comparison of tablet vs phone.



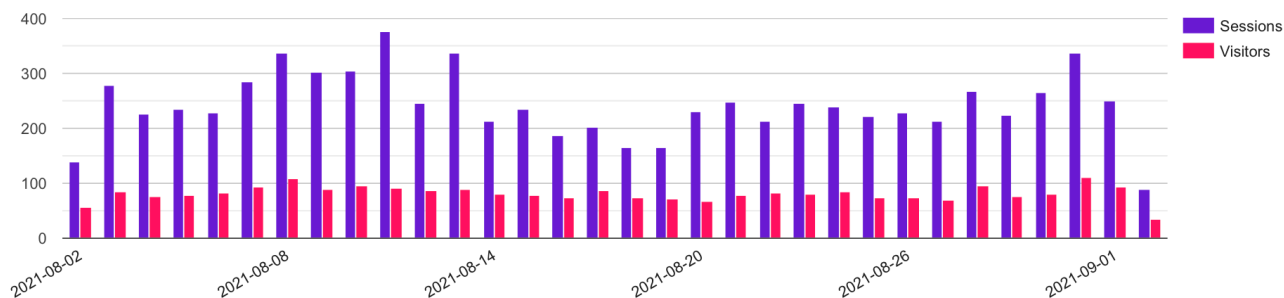
2.4. Sessions & Users

2.4.1. Application sessions

This report displays a column chart for the number of application sessions and visitors for a given period. The report defaults to the last 30 days, but a date picker lets you change that.

An application session is a period that starts the moment a visitor opens the app and ends either when the visitor leaves the app, or when the device goes to sleep.

The list below the chart provides extra information like the average number of sessions per visitor, and information about the duration of the sessions.

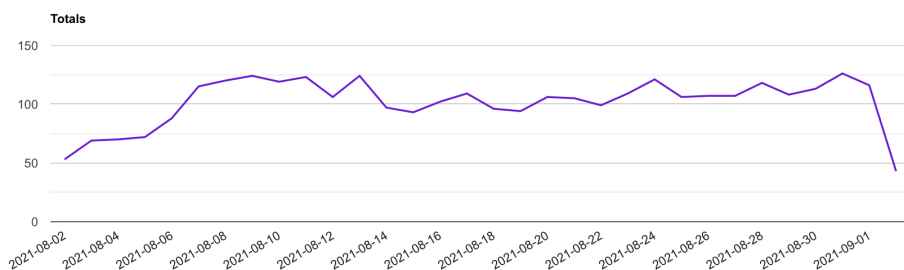


Date	# Sessions	# Visitors	Avg. # Sessions	Avg. Duration hh:mm:ss	Min. Duration hh:mm:ss	Max. Duration hh:mm:ss	Sum. Duration hh:mm:ss
02 September 2021	90	34	2,65	00:03:26	00:00:03	00:28:24	05:08:33
01 September 2021	250	93	2,69	00:05:07	00:00:03	07:01:20	21:20:43
31 August 2021	338	111	3,05	00:03:38	00:00:03	03:02:55	20:26:29
30 August 2021	265	80	3,31	00:10:16	00:00:03	20:23:57	45:20:26

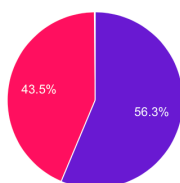
2.4.2. Active Users

This report displays a line chart of the number of users per day connecting to your app over a given period. The report defaults to the last 30 days, but a date picker lets you change that.

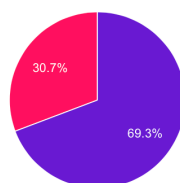
There are also two pie charts, one based on the percentages for tablet vs phone vs browser, another one on the distribution of iOS vs Android.



Tablet vs Phone vs Browser



iOS vs Android

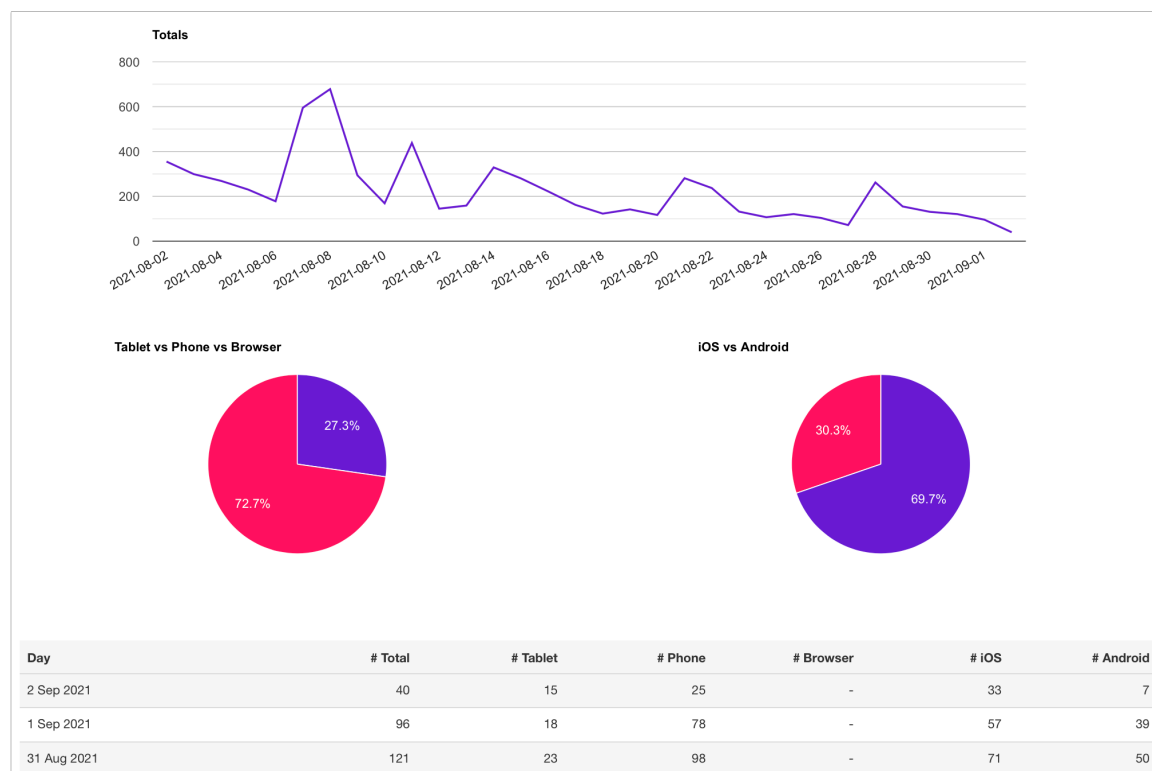


Day	# Total	# Tablet	# Phone	# Browser	# iOS	# Android
2 Sep 2021	43	22	16	5	27	11
1 Sep 2021	116	64	52	-	88	28
31 Aug 2021	126	71	55	-	96	30

2.4.3. First Time Users

This report displays a line chart of the number of users per day connecting to your app for the first time. The report defaults to the last 30 days, but a date picker lets you change that.

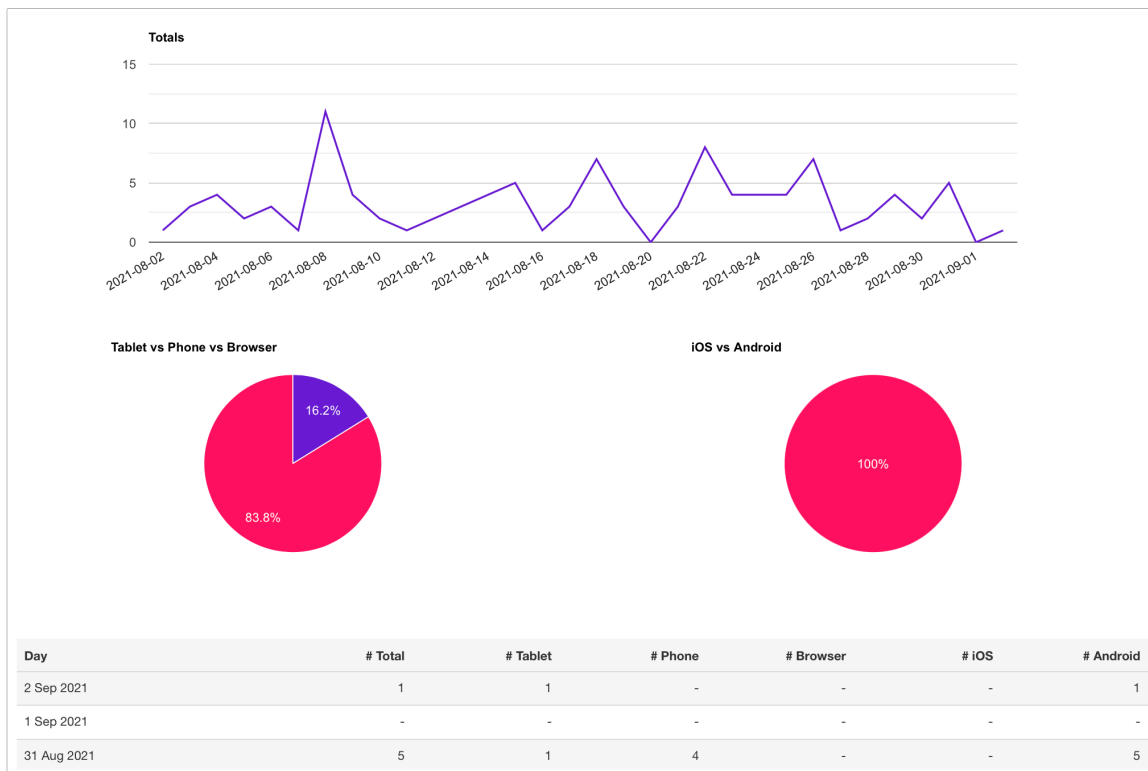
There are also two pie charts, one based on the percentages for tablet vs phone vs browser, another one on the distribution of iOS vs Android.



2.4.4. First Entitlement logins

If your app uses entitlements, this report will display a line chart of the number of users per day logging in to the app for the first time. The report defaults to the last 30 days, but a date picker lets you change that.

There are also two pie charts, one based on the percentages for tablet vs phone vs browser users, another one on the distribution of iOS vs Android users.



2.5. Sharing

2.5.1. Shared Content Items

This list displays a table of the most popular content items that have been shared by users of your app. The table also details the number of tablet vs phone vs browser shares, and the number of iOS vs Android shares.

2.5.2. Share Destinations

This report displays 3 pie charts of the channels to which your users have shared over a given period. The report defaults to the last 30 days, but a date picker lets you change that.

One chart shows the totals, a second one the comparison of tablet vs phone vs browser, and a third one the distribution of iOS vs Android.

2.6. Search

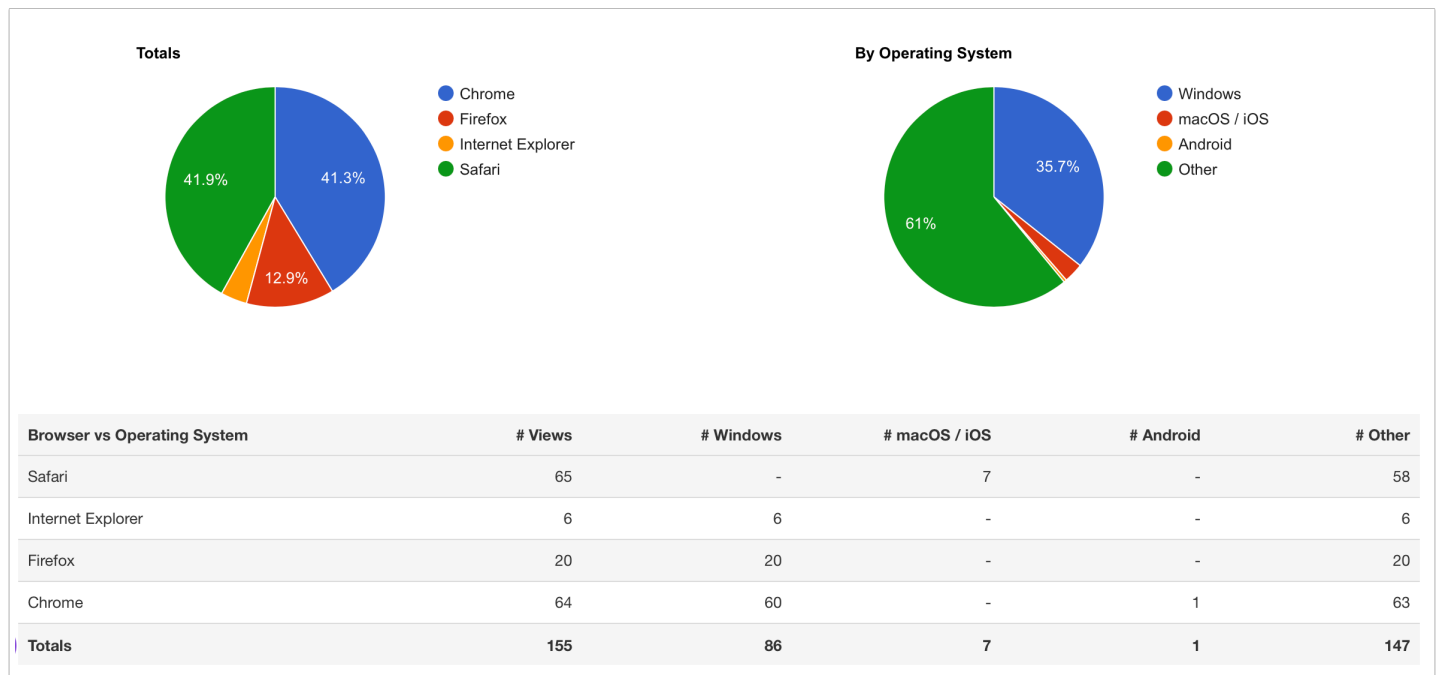
2.6.1. Search Queries

This list displays a table of the most popular queries that users where searching for in your app. The table also details the number of tablet vs phone vs browser shares, and the number of iOS vs Android shares.

2.7. Browser Client

2.7.1. Browser versions vs Operating System

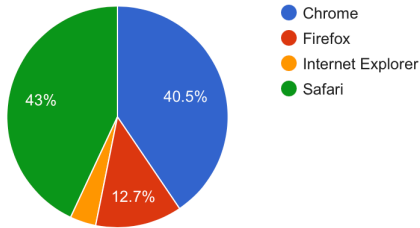
This report displays 2 pie charts: one based on the browser (Safari, Chrome, etc;) that users for the browser client use, a second one shows the distribution by operating system. The report defaults to the last 30 days, but a date picker lets you change that.



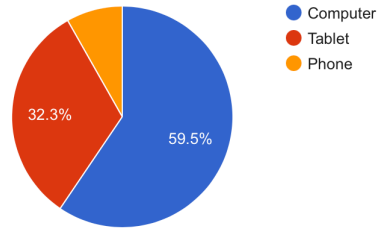
2.7.2. Browser versions vs Device Type

This report displays 2 pie charts: one based on the browser (Safari, Chrome, etc;) that users for the browser client use, a second one shows the distribution by device type (computer vs tablet vs phone). The report defaults to the last 30 days, but a date picker lets you change that.

Totals



By Device Type



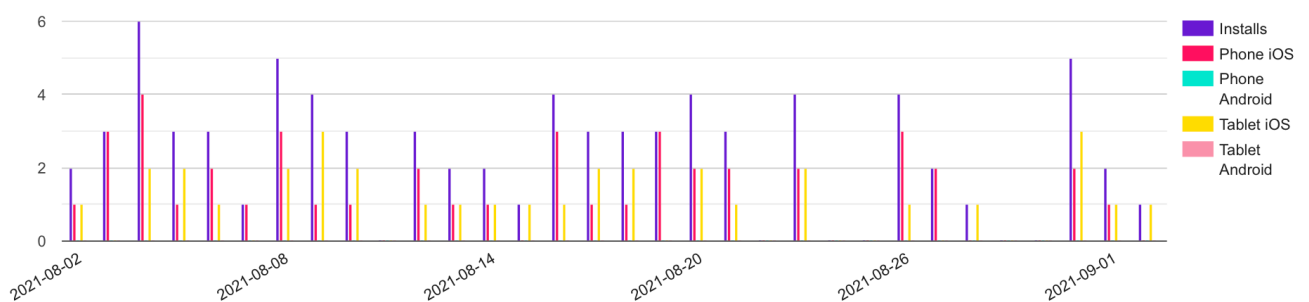
Browser vs Device Type	# Views	# Computer	# Tablet	# Phone	# Other
Safari	68	7	51	10	-
Internet Explorer	6	6	-	-	-
Firefox	20	20	-	-	-
Chrome	64	61	-	3	-
Totals	158	94	51	13	-

2.8. Installs

2.8.1. Fresh app installs

This report displays a column chart that shows a list of the number of users that have installed and opened your app for the first time.

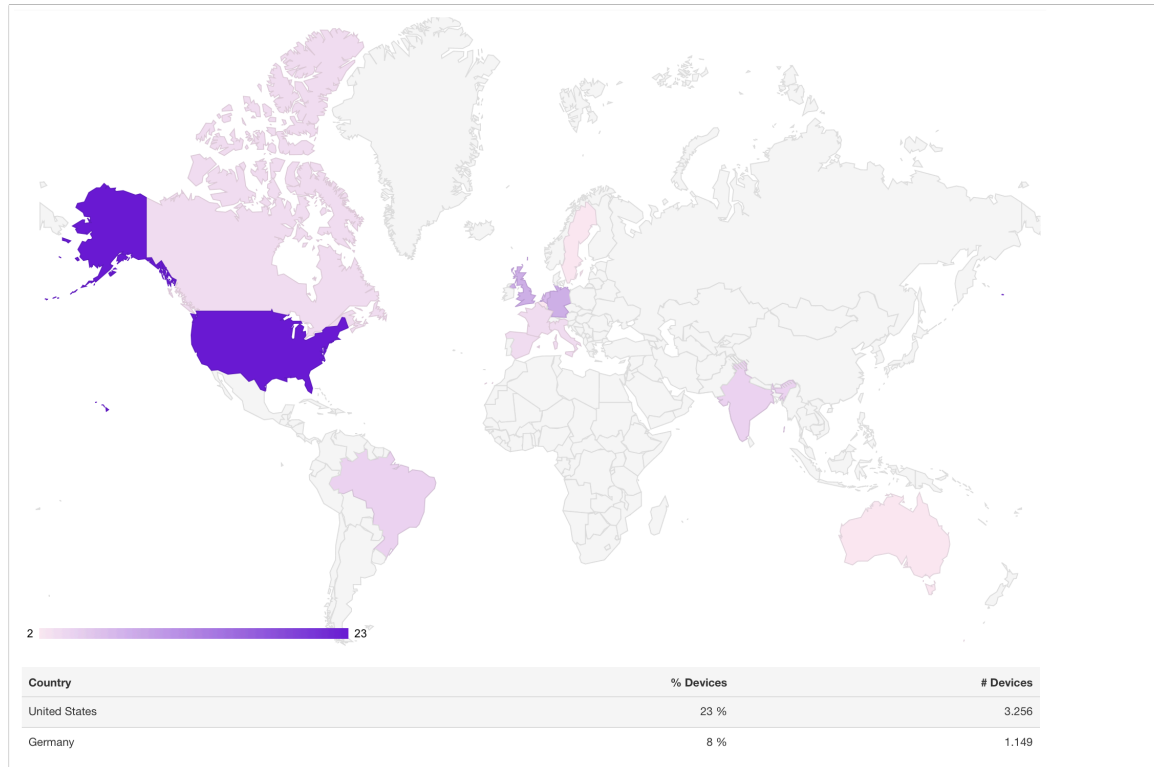
The table below provides more details about the numbers and percentages for tablets vs phones and iOS vs Android.



Date	# Installs	% Tablet iOS	% Tablet Android	% Phone iOS	% Phone Android	# Tablet iOS	# Tablet Android	# Phone iOS	# Phone Android
02 september 2021	1	100 %	-	-	-	1	-	-	-
01 september 2021	2	50 %	-	50 %	-	1	-	1	-
31 august 2021	5	60 %	-	40 %	-	3	-	2	-
30 august 2021	-	-	-	-	-	-	-	-	-

2.8.2. Geographical Overview

This show a world map with the distribution of your users by country. The table below the map displays a list of the countries and what they represent in terms of number of devices and percentages.

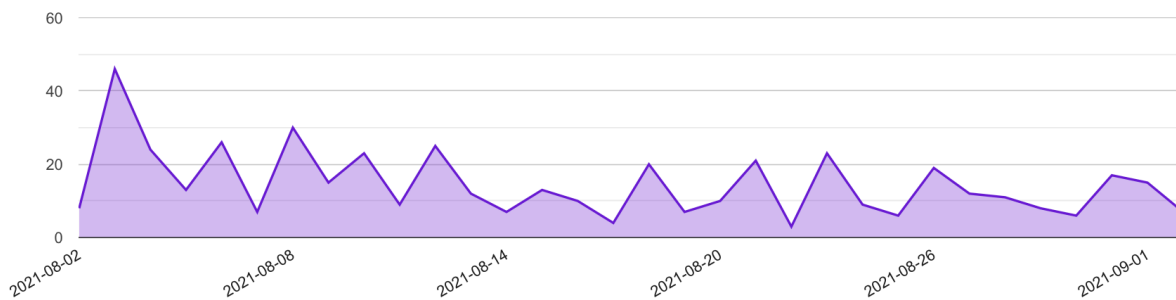


2.9. Downloads & Bandwidth

2.9.1. Content Item Downloads

This report displays a line chart of the number of content item downloads per day. The report defaults to the last 30 days, but a date picker lets you change that.

The table below the chart shows details about the number of downloads, the percentage of the total they represent and the total and average download sizes.

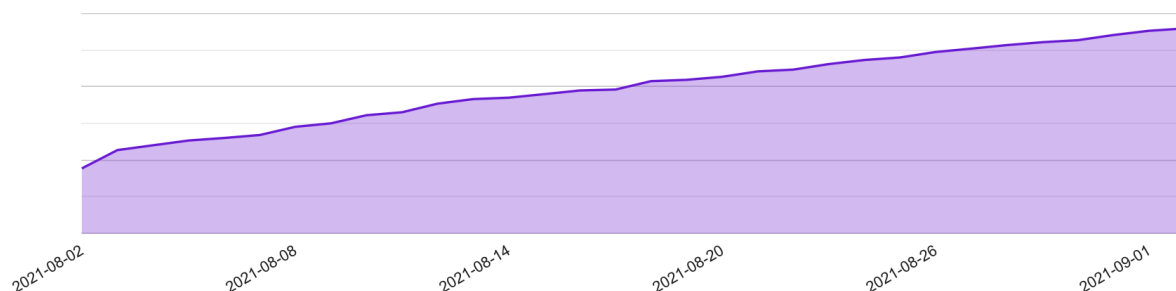


Date	# Downloads	% Downloads	Total Download Size	Average Download Size
02 September 2021	7	2 %	318,24 MB	45,46 MB
01 September 2021	15	3 %	577,49 MB	38,50 MB
31 August 2021	17	4 %	704,12 MB	41,42 MB
30 August 2021	6	1 %	277,03 MB	46,17 MB

2.9.2. Bandwidth Usage

This report displays an area chart that shows the cumulative bandwidth usage for all the downloads in your app.

The table below the chart also shows bandwidth usage per day.






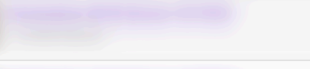






Date	Used Bandwidth	Cumulative Used Bandwidth
02 September 2021	318,24 MB	87,95 GB
01 September 2021	577,49 MB	87,64 GB
31 August 2021	704,12 MB	87,06 GB
30 August 2021	277,03 MB	86,35 GB

2.9.3. App Size

This report displays a table listing the size of your collections, from large to small. For PDFs and HTML content items the size will be the same for iPad, Android tablet and phone. For Twixl article created in InDesign, these may be different if you created different renditions of your content.

If you click on the name of a collection, a list of all its content items will be displayed.

Collection	iPad	Android Tablet	iPhone & Android Phone
 	167,28 MB	167,28 MB	167,28 MB
 	149,03 MB	149,03 MB	149,03 MB
 	134,34 MB	134,34 MB	134,34 MB
 	129,87 MB	129,87 MB	129,87 MB
 	122,79 MB	122,79 MB	122,79 MB

3. Entitlements Users

For the Entitlement Scenarios *Users & Groups* or *Username & Password* built-in analytics are also available. When you go to the list of groups or users, for each user / group, you will now see links to the *content item views*, *collection views* and *application sessions*. See screenshots

WARNING:

Only available for the built-in scenarios *Users & Groups* or *Username & Password*.

HINT

The same information is also available via the [Admin API v2](#).

TWIXL DISTRIBUTION PLATFORM

Production

Reports

Twixl Distribution Platform

Applications

Production

Entitlements

ENTITLEMENTS

PRODUCTION | PRODUCTION MODE

Edit Settings

ENTITLEMENTS

Entitlements Mode Users & Groups

Entitlements URL

URL Style Path: http://host/action?app_id=test

BEHAVIOUR

Login Button Title Login

Login Button Description

Logout Button Title Logout

Logout Button Description

GROUPS

Search Group

Search

Add Group

Group	# Users	Created On	
Admin_all	4	2017-07-13 08:07	content item views collection views app sessions edit delete
Argentina_ES	1	2017-07-13 08:12	
Australia_EN	11	2017-07-13 08:14	
Austria_DE	13	2017-07-13 08:16	
Belgium_EN_FR	4	2017-07-13 08:20	
Brazil_PT	31	2017-07-13 08:22	
Canada_EN_FR	13	2017-07-13 08:25	
China_CN	20	2017-07-13 08:27	
Colombia_ES	2	2017-07-13 08:28	
Croatia_EN	4	2017-07-13 08:30	
Czech_EN	1	2017-07-13 08:31	
Denmark_EN	4	2017-07-13 08:32	
DomRep_ES	1	2017-07-13 08:35	
Ecuador_ES	1	2017-07-13 08:37	
Estonia_EN	1	2017-07-13 08:38	
Finland_EN	2	2017-07-13 08:39	
France_FR	15	2017-07-13 08:41	
Germany_DE	20	2017-07-13 08:42	

Display a menu

TWIXL DISTRIBUTION PLATFORM

Production

Reports

Twixl Distribution Platform

Applications

Production

Entitlement Group

Content Item Views

CONTENT ITEM VIEWS

30 SEPTEMBER 2017 - 30 OCTOBER 2017

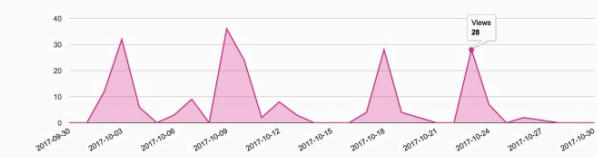
PRODUCTION | PRODUCTION MODE

30/09/2017 17:20 pm

30/10/2017 17:20 pm

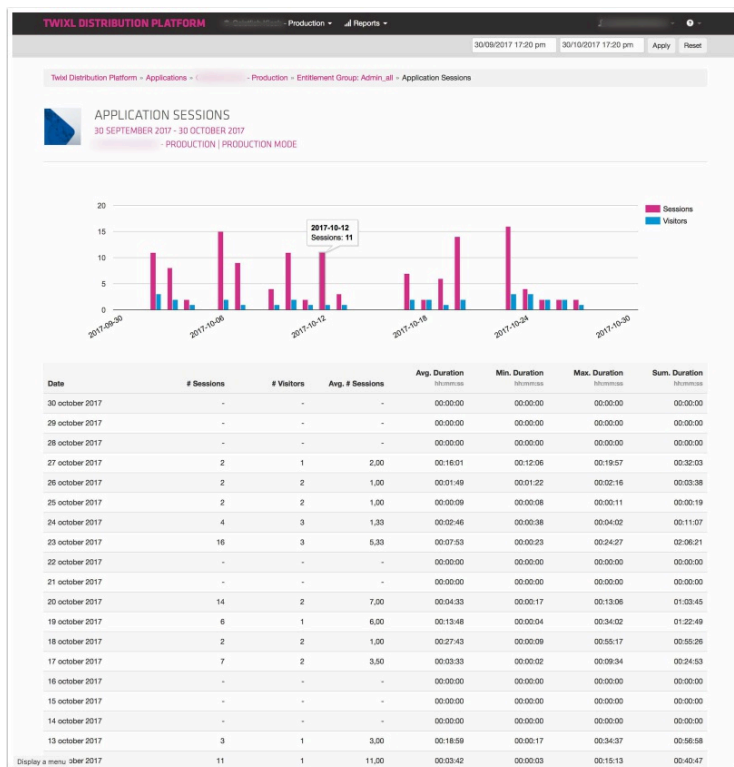
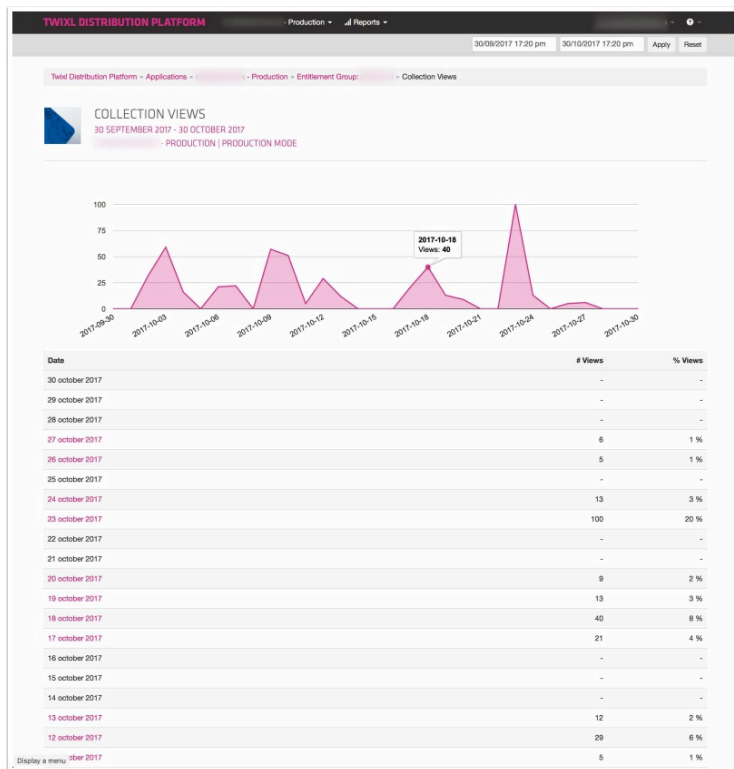
Apply

Reset



Date	# Views	% Views
30 october 2017	-	-
29 october 2017	-	-
28 october 2017	-	-
27 october 2017	1	0 %
26 october 2017	2	1 %
25 october 2017	-	-
24 october 2017	7	3 %
23 october 2017	28	13 %
22 october 2017	-	-
21 october 2017	-	-
20 october 2017	2	1 %
19 october 2017	4	2 %
18 october 2017	28	13 %
17 october 2017	4	2 %
16 october 2017	-	-
15 october 2017	-	-
14 october 2017	-	-
13 october 2017	3	1 %
12 october 2017	8	4 %
11 october 2017	2	1 %

Display a menu



Analytics Events Overview

The Twixl platform gathers anonymized information about how users/readers interact with your app and the Twixl platform. While we provide [a number of reports](#) based on this data, some of the events are available only through the Twixl Admin API, and can be imported in your own analytics presentations, tables, tools... Below is an overview of all events that are collected. The events are collected on **iOS, Android and the browser client** (unless specified differently below).



IMPORTANT NOTE:

You need to have the Integration API as part of your subscription if you want to access the raw data using our Admin API.

1. Events in the apps

1.1. App Install event

`app-install`

This event is logged every time a user launches the app for the very first time on his/her device.

iOS and Android only.

1.2. Download event

`download`

This event is logged every time a user performs a download of a content item.

1.3. Entitlement login & logout events

`entitlement-login` & `entitlement-logout`

This event is logged every time a user performs a login or logout via entitlements.

1.4. Notification opened event

`notification-open`

This event is logged every time a user opens a push notification on his device.

iOS and Android only.

1.5. Purchase events

`purchase-item`

This event is logged every time a user purchases a specific content item (PDF) or collection.

`purchase-restore`

This event is logged every time a user performs a restore of the previous purchases.

`purchase-subscription`

This event is logged every time a user purchases a subscription.

iOS and Android only.

1.6. Search event

`search`

This event is logged every time a user performs a search query.

1.7. Session events

`session-start`

This event is logged every time a user launches the app or bring the app to the foreground.

When the user logs in via entitlements, this is also considered as a new session.

`session-end`

This event is logged every time a user stops the app or when the app goes to the background.

When the user logs out from entitlements, this also should be considered as a session end.

iOS and Android only.

1.8. Social sharing

`social-share`

This event is logged every time a user successfully shares a content item on social media.

iOS and Android only.

1.9. View events

1.9.1. General view events

`view-collection`

This event is logged every time a user goes to a specific collection, regardless of the view mode (browse or detail). If the user first goes to the browse page and then navigates to the detail page, the event is logged twice.

If the user returns from e.g. a detail view of a collection back to the browse page, these count as separate events.

`view-content-item`

This event is logged every time a user visits a specific content item page. For each page, we log a separate event.

For PDF content items, every page that is displayed will be logged as well.

This event is only triggered after the content item has been downloaded and one of the pages is visible on the screen.

`view-pagelink`

This event is logged every time a user taps on a pagelink in a content item. For each page, a separate event will be logged.

`view-paywall`

This event is logged every time a user ends up on the paywall.

iOS and Android only.

`view-settings`

This event is logged every time a user opens the settings window.

iOS and Android only.

`view-weblink`

This event is logged every time a user taps a web link in a content item.

1.9.2. InDesign-specific view events

`view-button`

This event is logged every time a user taps a button in a Twixl InDesign article. For each page, a separate event will be logged.

`view-image-sequence`

This event is logged every time a user swipes or taps on an image sequence in a Twixl article. This event will be counted only once for each page view.

As long as the user stays on the same page and swipes multiple times on the image sequence, it counts as a single event.

`view-movie`

This event is logged every time a user starts the playback of a movie in a Twixl article (that does not autoplay).

If the user plays the movie multiple times during the same "page session", only one event will be logged.

`view-scrollable-content`

This event is logged every time a user swipes in a scrollable content item in a Twixl article.

This event will be counted only once for each page view. As long as the user stays on the same page and swipes multiple times on the scrollable, it will count as a single event.

`view-slideshow-slide`

This event is logged every time a user swipes to another slide in a slide show. If the slideshow is set to autoplay, the event will not be logged.

`view-sound`

This event is logged every time a user taps a sound element in a Twixl article (that does not autoplay).

If the user plays the sound multiple times during the same "page session", only one event will be logged.

`view-web-overlay-button`

This event is logged every time a user taps on a web overlay button in a Twixl article.

2. Admin events on the Twixl platform

`app-create`

This event is logged when a user creates a new app on the Twixl platform.

`app-delete`

This event is logged when a user deletes an app from the Twixl platform.

`collection-create`

This event is sent when a user creates a collection on the Twixl platform.

`collection-delete`

This event is sent when a user deletes a collection on the Twixl platform.

`content-item-create`

This event is sent when a user creates a content item on the Twixl platform.

`content-item-delete`

This event is sent when a user deletes a content item on the Twixl platform.

`notification-send`

This event is triggered after a notification for a specific app has been sent.

Google Analytics 4

Google decided some time ago to phase out Google Analytics UA and to replace this by Google Analytics 4. Starting with TP 19.0, Twixl will support both systems until Google ends UA support (planned for June 2023).

More specifically:

- If you are already using Google Analytics UA, your app will continue to feed data to Google Analytics UA until Google stops receiving data to UA. Google states that you will be able to consult your UA data for some time after new UA data will no longer be accepted.
- If you haven't used Google Analytics UA yet, it will only be possible to register your Google Analytics key for Google Analytics 4.
- If you upload a new app, it will only be possible to register a Google Analytics 4 key.

How

Main > Edit Application > Enter your key(s) in Google Analytics Keys.

Google Analytics 4

[More info](#)

API Secret

The Google Analytics API Secret


Measurement ID

The Google Analytics Measurement ID

Client ID

The Google Analytics Client ID

With Google Analytics 4, there are no different keys for different platforms (iOS, Android, Browser) as this is handled by Google Analytics 4 itself.

 Google stated that it isn't possible to collate data from Google Analytics UA and Google Analytics 4 easily. This requires custom development from the user.

Click [here](#) for more information about Google Analytics 4.

Google Analytics (legacy)

Google will end support for Google Universal Analytics by the end of June 2023.
Please use [Google Analytics 4](#) instead.

Setup in Google Analytics

You'll need to generate a Google Analytics property for each platform you want to have separate analytics for.

To generate a Google Analytics Key:

1. Login to [analytics.google.com](#).
2. Select **Admin** in the bottom left corner.
3. Create a new **Property**.
4. In **Property Setup**, make sure to select 'Show Advanced Options'.
5. Enable the option '**Create a Universal Analytics property**' (never mind that it says 'It is not recommended for new Analytics users, and it will not support app measurement')
6. Enter a 'fake' URL in the website URL field. This doesn't really matter as you will be using this for an app, not a web site.
7. Finish creating the property.
8. In the next step, under Tagging Instructions, you can link this new property to an older property (if you used one before). Look for the option **Use existing on-page tag**.

ADMIN USER

Create property

1 Property setup

Property details
A property represents a business's web and/or app data. An account can contain one or more properties. [Learn more](#)
**Navigate to your Firebase account if you want to create a new property for an existing Firebase project.*

Create a Google Analytics 4 property to measure your web and/or app data.

Property name
Twixl Project

Reporting time zone
United States - (GMT-07:00) Los Angeles Time -

Currency
US Dollar (USD \$) -

You can edit these property details later in Admin

[Hide advanced options](#)

Create a Universal Analytics property

The Universal Analytics property is a part of an earlier version of Google Analytics and only supports web measurement. It is not recommended for new Analytics users, and it will not support app measurement. Turn this on if you'd like to create a Universal Analytics property.

Website URL
https:// www.mywebsite.com

☒ **Create both a Google Analytics 4 and a Universal Analytics property**
This will create two properties with connected site tags, so you only need to tag your website once. [Learn more](#)

☒ **Enable enhanced measurement for Google Analytics 4 property**
Automatically measure interactions on your sites in addition to standard screen view measurement.
Data from on-page elements such as links and embedded videos may be collected with relevant events. You must ensure that no personally-identifiable information will be sent to Google. [Learn more](#)

☐ **Create a Universal Analytics property only**

Setup for iOS & Android apps & the Browser Client

If you want to activate Google Analytics for your mobile apps and/or the Browser Client, select 'Edit App' and enter the Google Analytics Key in the relevant fields. For an overview of all application options, you can check [this article](#).

Adding analytics information to InDesign interactive elements

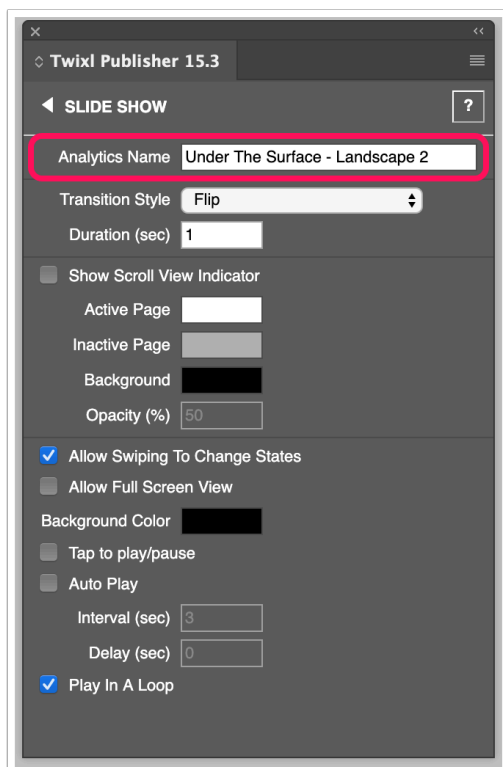
Introduction

Twixl Publisher allows you to gather analytics information in your apps.

How to add analytics information to InDesign interactive elements?

The Twixl platform offers [built-in support for analytics](#) with zero configuration.

If you want to make your analytics more readable, you can enter an analytics name for your interactive objects, such as slide shows, audio or video clips, and web overlays. If you don't enter your own analytics name, a default value will be used.



Which information gets tracked ?

The following items of your InDesign articles will be tracked:

- Visited articles
- Visited pages inside articles

These items will be logged as "fake URLs" and will use the following structure:

- Pages: `/<publication>/<article>/<orientation>/<page>`

Push Notifications

iOS: Setup key-based authentication in the Apple Developer Portal

1. Setup your App ID

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

1. Navigate to (**developer.apple.com** > **Account** > **Certificates, Identifiers & Profiles** > **Identifiers** > **App ID**)
2. If you want to edit an existing App ID, select *Configure*, otherwise select *New Apple 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.

Certificates, Identifiers & Profiles

[← All Identifiers](#)

Edit your App ID Configuration Remove Save

Platform
iOS, macOS, tvOS, watchOS

App ID Prefix
ZM7SDYSP39 (Team ID)

Description
Avantgand Magazine

Bundle ID
com.twixlmedia.avantgand (explicit)

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

Capabilities

ENABLED	NAME
<input type="checkbox"/>	Access WiFi Information ⓘ
<input type="checkbox"/>	App Attest ⓘ
<input type="checkbox"/>	App Groups ⓘ Configure
<input type="checkbox"/>	Apple Pay Payment Processing ⓘ Configure
<input type="checkbox"/>	Associated Domains ⓘ
<input type="checkbox"/>	AutoFill Credential Provider ⓘ
<input type="checkbox"/>	ClassKit ⓘ
<input type="checkbox"/>	Custom Network Protocol ⓘ

2. Activate Push Notifications for the App ID


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



WARNING:

For key-based authentication, there is no need to create a Development or Production SSL Certificate

3. How to create the authentication key

1. Navigate to <https://developer.apple.com/account/ios/authkey/> (**developer.apple.com** > **Account** > **Certificates, Identifiers & Profiles** > **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

3

Key Name

AvantGand Magazine

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

ENABLE	NAME	SERVICE	
4	<input checked="" type="checkbox"/>	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
	<input type="checkbox"/>	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
<input checked="" type="checkbox"/>	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)

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

iOS: Setup Push Notifications in Firebase (TP19 and up)

As of release 19.0, **new Twixl apps** will only be able to use Firebase-based configurations for sending out push notifications.

We chose to adopt Firebase fully 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.

1. Configure Firebase for a new Twixl app

New apps created with Twixl Publisher 19 and up will only allow you to configure push notifications through Google Firebase.

1. Create a Firebase Project

First create a Firebase project that will host the iOS and the Android applications. Follow the steps to [set up the Firebase project for Android](#).

2. 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 in your build setting on the platform.

3. Configure the iOS app to use the Apple Push Notification Servers (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 supported way to configure Firebase, because this key never expires.

1. Follow the steps to [create a p8-key](#). Take a note of the Key ID, the Team ID and save 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.twixlmedia.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. Browse to the location where you saved your key, select it, and click Open.
5. Add the Key ID and Team ID click the Upload button.

4. **Configuring the Twixl Platform to use Firebase.**

1. On the Twixl Platform page of your app, in the Main menu select "Push Notifications"
2. Upload the Service Account File (`GoogleService-Info.plist`) created in step 2.

Code Signing

OTA Deployment URL

Only applies to Ad Hoc and Enterprise builds.
Should be a https URL.

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](#).

5. Creating a build.

1. Create a build with build settings that have configurations for push notifications. 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](#).

2. 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 September 30th, 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 September 30th, it is mandatory to move to Firebase.

3. 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.

Upon 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.

From the Settings 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 (TP19 and up)

1. Get Started

1. Go to <https://console.firebase.google.com/>
2. **Signin** with a Google Account

TIP:

If you login with a *Google Account* that is linked to a *Google Play Developer account*, you can link your *Firebase Projects* (and apps) more easily to your Android apps in the *Google Play Developer Console*. However, this is not a requirement to setup your *Push Notifications* for a Twixl app.

2. Add a Project

1. Click on **Add Project**.
2. Choose a **Project Name** (e.g. the name of your app: `My Magazine`). You can also choose an existing Project to setup Firebase for.
3. Choose a **Project ID** or accept the proposed one.

3. 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'.
5. Select 'Manage service accounts'.
6. Click on one of the service accounts or create a new one.
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 to the platform.

ABOUT FIREBASE ANDROID AND IOS APPS:

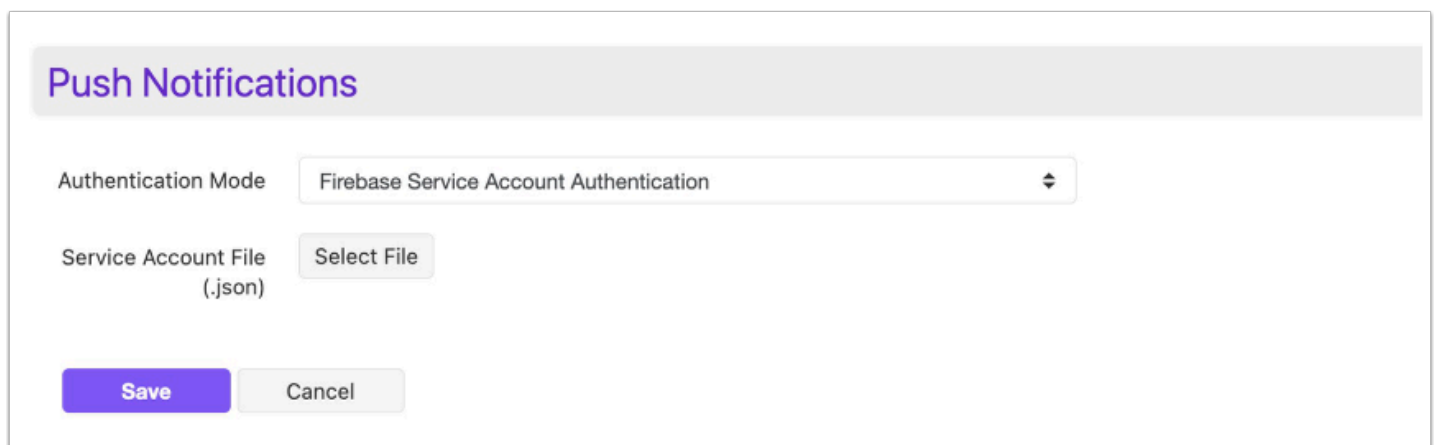
In Firebase, you can setup an Android and/or iOS app. For more details about configuring for iOS, check [here](#).

4. Generate your Google Services JSON file

1. In **Project Settings**, select the **General** tab.
2. Add an **Android** app.
3. Fill in the *Android package name* (a reverse DNS name like *com.mycompany.myapp*), and optionally specify an *App nickname*.
4. Click *Register* to continue.
5. Download the `google-services.json` file and save this file. You'll need it in your build setting on the platform.

5. Upload the JSON Key to the platform

1. Go to the detail view of your app on the Twixl platform, then select *Notifications* from the top menu.
2. Upload the JSON Key generated in [step 3](#).



Push Notifications

Authentication Mode: Firebase Service Account Authentication

Service Account File (.json): Select File

Save Cancel

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

In the build setting for your app on the platform, select the `google-services.json` file from [step 4](#) to add it.

Code Signing

OTA Deployment URL

Only applies to Ad Hoc and Enterprise builds.
Should be a https URL.

google-services.json

Select File

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

GoogleService-
Info.plist

Select File

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

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.

Sending Push Notifications

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

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.

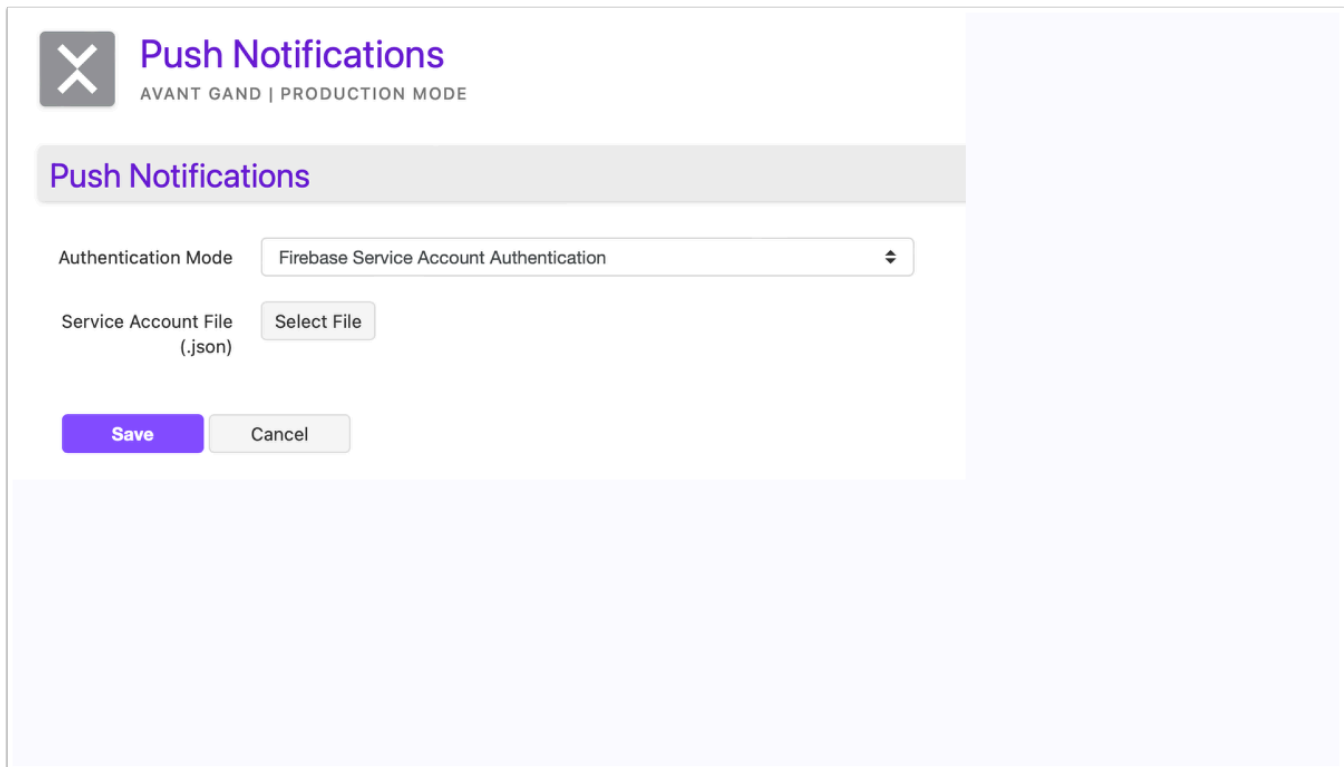
Check out the following articles for help on how to setup your app for Push Notifications:

- 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 an updated build of the app.



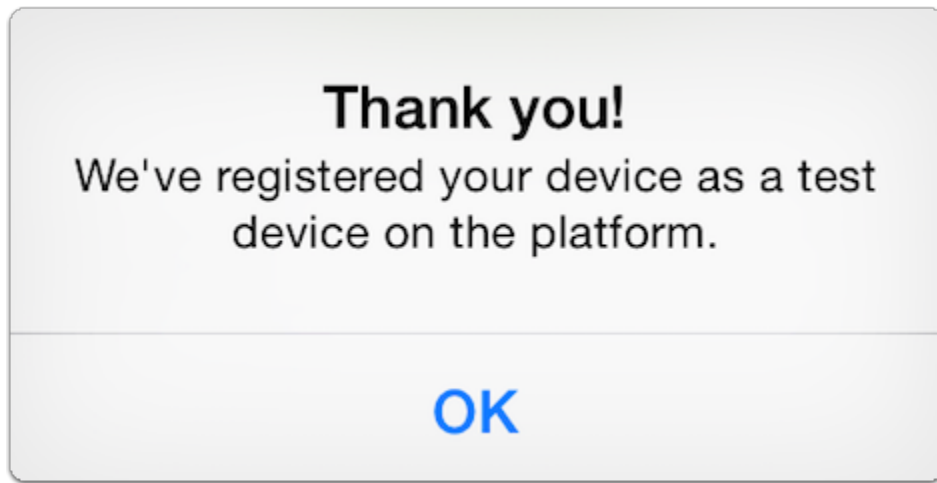
The screenshot shows a web interface for configuring push notifications. At the top left is a logo with a stylized 'X' inside a square, followed by the text 'Push Notifications' and 'AVANT GAND | PRODUCTION MODE'. Below this is a header bar with the text 'Push Notifications'. The main form area contains two fields: 'Authentication Mode' with a dropdown menu set to 'Firebase Service Account Authentication', and 'Service Account File (.json)' with a 'Select File' button. At the bottom of the form are two buttons: 'Save' (in purple) and 'Cancel' (in grey).

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 or updated 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 you 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.



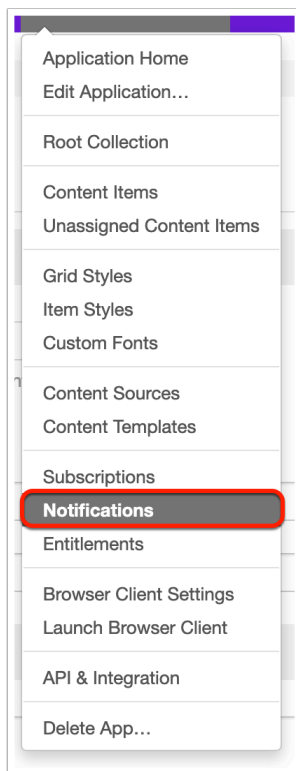
NOTE:

The test functionality on iOS devices is only available in [Test Flight](#) 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 app menu on the platform.

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



3.1. Available options

These are the available options:

1. **Type: Plain or Rich Push Notifications**
2. **Title:** Your *Rich Push Notification* can include a **Title**
3. **Message or Description:** Of course you want to include a **Message** in your *Plain Push Notification*. This **Message** can include 140 characters. 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. **Bundle Identifier:** Informs you about the targeted app. See the following article for more info: [iOS: Setup Key based authentication for Apple Push Notifications \(>8.0\)](#)
8. **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](#)

Push Notification

1
Type *
Rich
Rich Push Notifications requires a v12 or newer reader app

2
Title

3
Description

4
Image
Select File

5
Link
The "tp-1" link to send with the rich push notification

6
Send At
14/09/2020 10:57

Send To

7
Bundle Identifier

8
Send to *
All device(s)
All iOS device(s)
All Android device(s)
Specific test device(s)

Reach

Number Of Apple Devices 8
Number Of Android Devices 11
Total Number Of Devices 19

Send Cancel

3.2. Support Matrix Plain vs. Rich Push Notifications

What?	Supported in Rich Push Notifications	Supported in Plain Push Notifications
Title	yes	no
Description	yes	no
Message	no	yes
Image	yes	no
Link	yes	yes
Send At	yes	yes
Bundle Identifier	yes	yes
Send to	yes	yes

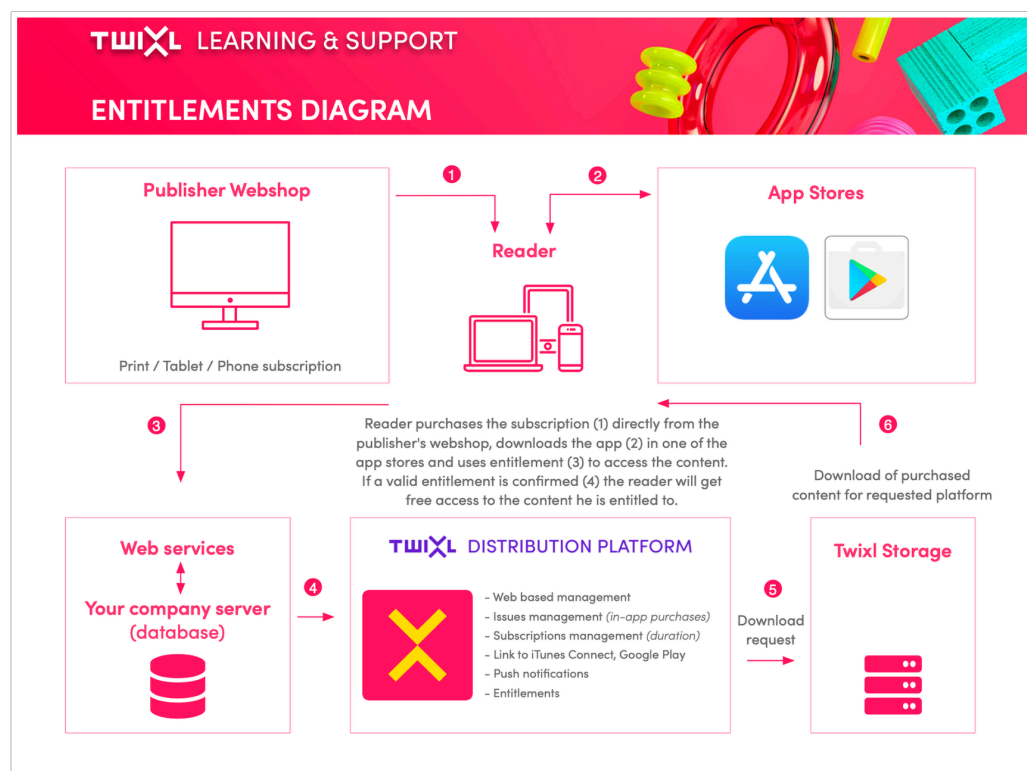
NOTE:

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

Entitlements

Entitlement Diagram: how it works

This diagram explains in a nutshell how Entitlements work.



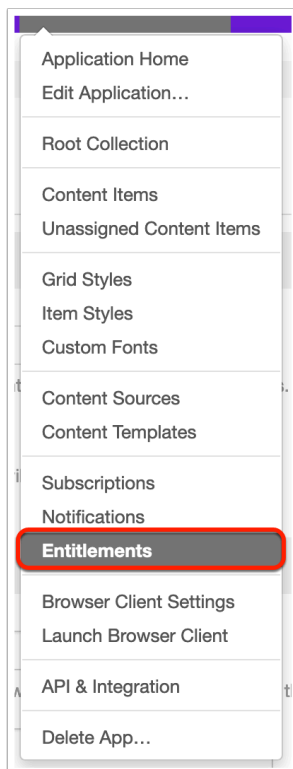
You can also download this diagram:

[Twixl Entitlements Diagram](#)

Using the Entitlements option

By using entitlement in your app, you can determine which content a specific reader/ user can have access to. In the world of magazines, this is most frequently used to allow print subscribers to get free access to the digital editions by logging in with a user name and a password. But entitlement is also frequently used by companies that want employees or partners to login to an app that can be distributed both internally or via the App Stores.

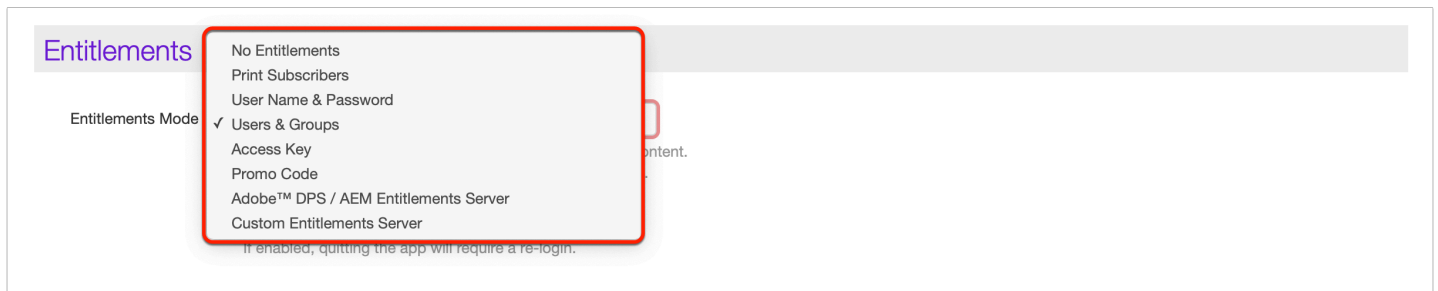
1. Configuring entitlements for your app



When navigating to the detail of your app, select Entitlements from the menu. You can then select one of the preconfigured scenario's:

1. **Print Subscribers:** for providing print subscribers free access to digital content – users can still see and purchase the different collections (issues) and also purchase a subscription through the App Store
2. **User name & password:** for a restricted access app, i.e. without a user name and password, users will not be able to access any content). This scenario also has a special default user called 'No Entitlements' that allows you to determine whether certain types of content will be displayed anyway when a user is not logged in.

3. **Users & Groups:** define access to your collections based on group access privileges – users belong to a group, collections can be visible in one or more groups.
4. **Access Key:** can be used for providing extra content to anyone with a valid access key
5. **Promo Code:** allows readers to redeem a promo code and get an issue for free that normally needs to be purchased
6. **Adobe DPS/AEM Direct Entitlement Server:** useful for customers migrating from this solution that were previously using Adobe's Direct Entitlement API.
7. **Custom Entitlements Server:** connect via a web service to an external database. See [this article](#).



When you select a specific scenario, you will also be able to configure the labels of the login form.

Home > La Grande Motte > Entitlements > Settings

Entitlements
La Grande Motte | Development Mode

Entitlements

Entitlements Mode: Print Subscribers
Allows free access to payed content for all valid print subscribers.

☐ Auto Logout
If enabled, quitting the app will require a re-login.

Behaviour

Login Button Title:
If filled in, the button will be shown in the payroll and will enable the user to trigger the entitlements sign form.

Login Button Description:

Logout Button Title:
If filled in, the button will be shown in the payroll and will enable the user to logout from the entitlements.

Logout Button Description:

Labels

Entitlements Form Title:

User Name Label:

Password Label:

Log In Label:

Error Button Label:

2. Auto Logout

Entitlements

Entitlements Mode Print Subscribers

Allows free access to payed content for all valid print subscribers.

☐ Auto Logout
If enabled, quitting the app will require a re-login.

If you enable this option, quitting the app will require a re-login. A re-login will also be required when your mobile device goes into sleep mode.

3. Collection Options

Once you have set the entitlement scenario in Application Settings, two collection options are very important in this respect:

Collection Options

☐ Root Collection?
If checked, this is the default collection that will be displayed when the app starts

☐ Requires entitlements
If enabled, the entitlements signin form will be shown for unentitled users

3.1. Free of charge or in-app purchase

Each collection is set by default as *Free of charge*. Unchecking this option will make it an in-app purchase. In that case, you need to make sure that this in-app purchase is also defined on your Apple and/or Google developer account using the same product identifier.

Purchase Info

Product Identifier

If the collection is a purchase, this should match the product identifier for the In-App Purchase in App Store Connect and Google Play.
Leave empty to use the suggested identifier.
Only characters, numbers and dots should be used.

☐ Free Of Charge
If enabled, all users can access the contents in this collection. If disabled, the users will be presented with a paywall before they can access the contents.

Purchase Title (iOS)

Purchase Info (iOS)

Purchase Title (Android)

Purchase Info (Android)

When a reader navigates to a collection that has been defined as an in-app purchase, it will trigger the paywall to be displayed. The paywall can display a number of different options:

- the standalone purchase of a collection (or 'issue')
- the purchase of a subscription (if you defined any)
- a *Login* button: tapping this will select the entitlement sign-in form to be displayed

3.2. Requires entitlement

If this setting is enabled for a collection, the entitlements sign-in form will be triggered when an unentitled user wants to navigate to that collection. If a reader then logs in with a valid user name and password, he will get access to the collection.

! IMPORTANT NOTE:

If you check *Requires Entitlement* for the root collection, then the login form will be displayed on startup, and anyone who's not entitled will not be able to access any content in the app.

4. PDF Content Items

PDF Content Items are something special: while they are indeed a type of *Content Item*, for the purpose of *Entitlements* they are treated as a *Collection*. **As such, you can set the Entitlements options for each PDF Content Item.**

SOME TIPS:

- PDF Content Items have a **Product Identifier** (used for Entitlements and In-App Purchases). All other Content Items don't have Product Identifiers.
- PDF Content Items can – as a result – be offered as a **purchase**.
- PDF content Items are – as a result – also checked against the **Entitlements** to see if they need to be offered for free or not.
- PDF Content Items are not shown in **Detail Mode** of a Collection. They are being presented in their own special Detail View.

5. Working with subscriptions

If you want to offer an App Store or Google Play subscription in an app, there are two options:

- **Standard Subscription:** this is the traditional type of subscription where you get access to new collections (i.e. issues) that are added during the period of your subscription.
- **All Access Subscription:** this can be compared to e.g. a Spotify or Apple Music type of subscription: you get access to all content in the app, as long as your subscription remains valid.

Using "reset password" for the "Users & Groups" scenario (beta)


Please note that the "password reset" feature can be used only with the entitlementment scenario "Users & Groups". This feature is not enabled by default, so If you want to use this functionality in your app, please create a support ticket.

1. How to configure

When you select the Entitlementment scenario "Users & Groups", under "Labels", you'll see an field appear that can be used to define the text that is shown for the "Forgot password" link. If you leave it empty, the link will simply not show up.

The screenshot shows the Twixl Entitlements configuration interface. At the top, there's a header with the Twixl logo and the text "Entitlements ISSUE 6313 - ENTITLEMENT PASSWORD RESET | DEVELOPMENT MODE". Below this is a section titled "Entitlements" with a dropdown menu set to "Users & Groups". Underneath, there's a checkbox for "Auto Logout" which is currently unchecked. The next section is "Behaviour", which contains fields for "Login Button Title", "Login Button Description", "Logout Button Title", and "Logout Button Description". The final section is "Labels", which contains fields for "Entitlements Form", "User Name", "Password", "Log In", "Error Button", and "Reset Password Button". The "Reset Password Button" field is highlighted with a red box and contains the text "Forgot Password?". At the bottom of the form are "Save" and "Cancel" buttons.

This is also reflected in the main entitlements page:


Entitlements
ISSUE 6313 - ENTITLEMENT PASSWORD RESET | DEVELOPMENT MODE

Edit Settings

Entitlements

Entitlements Mode

Users & Groups

Entitlements URL

<https://platform.twixlmedia.com/8c41989d05525bd8b65caa04e3b707a3/entitlements>

Auto Logout

No

URL Style

Path: https://host/action?app_id=test

Behaviour

Login Button Title

-

Login Button Description

-

Logout Button Title

-

Logout Button Description

-

Labels

Form Title

Log In

User Name

User Name

Password

Password

Log In Button

Log In

Error Button

Try Again

Reset Password Button

Forgot Password?

Groups

Search Group

Search

Add Group

Group	# Users	Created On
No results found.		

Users

Search User Name

Search

Add User Name

User Name	Belongs To Group	Created On
<input type="checkbox"/> Unentitled User	-	2020-12-10 15:29

select all

Copyright 2013-2020, Twixl media. All Rights Reserved.
 Twixl media • Kortrijksesteenweg 1144 N, 9051 Gent, Belgium
 VAT BE 0892.553.121 • T +32 493 25 26 77 • info@twixlmedia.com
[legal notices](#) • [general terms and conditions](#)

2. Reset flow

When a user is in the app and the entitlement login dialog is shown, you'll see a link that allows you to reset your password:

3:37

Done

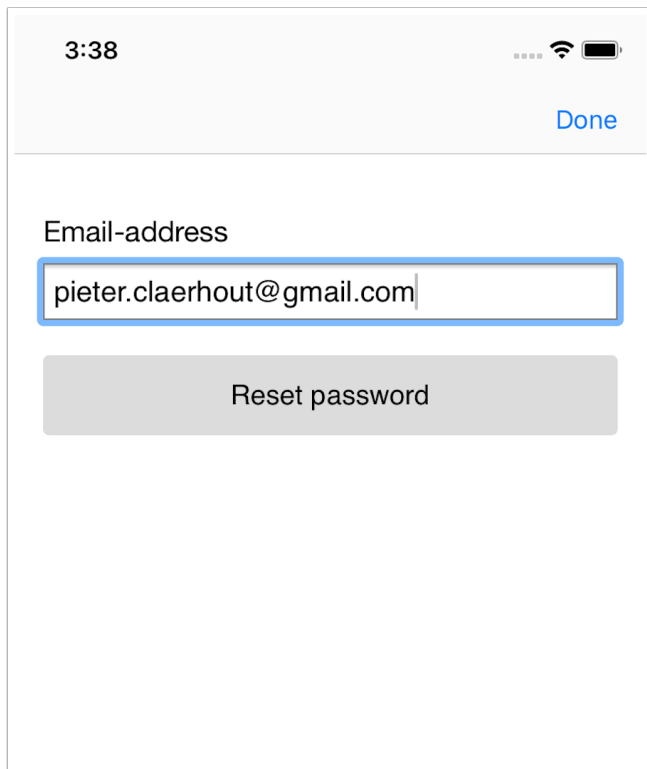
User Name

Password

Log In

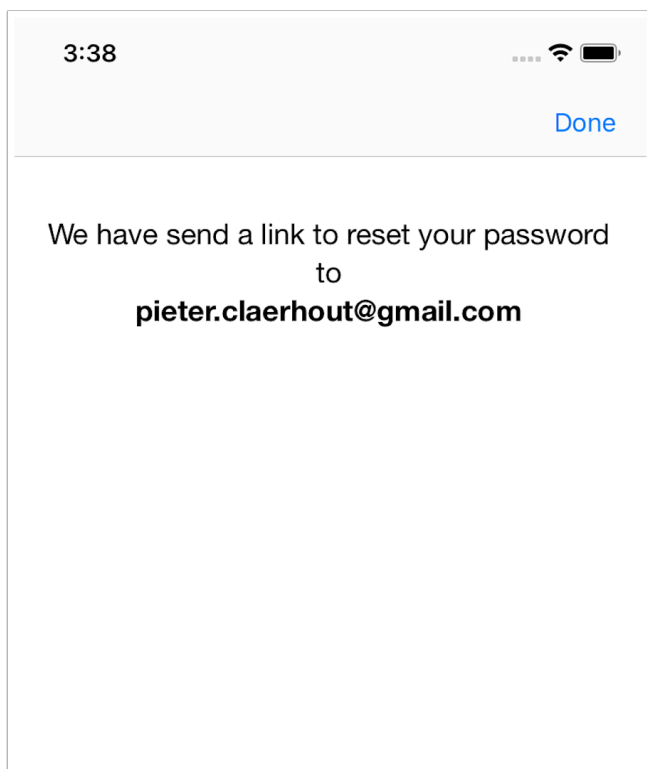
[Forgot Password?](#)

Tapping "Forgot Password?" takes you to a page where you can enter your email (or whatever you used as the label for the "username" on the platform):



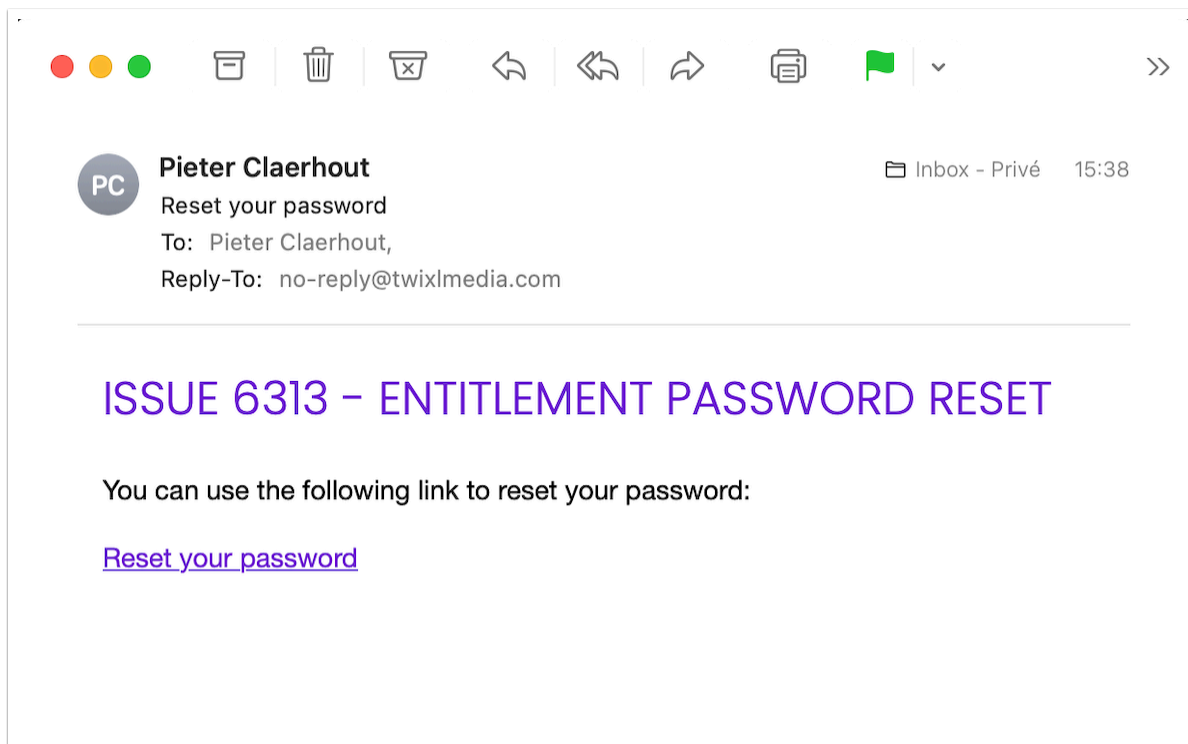
A screenshot of a mobile application interface for a password reset. At the top, the status bar shows the time 3:38, signal strength, Wi-Fi, and battery. Below the status bar is a header with a blue 'Done' link. The main content area has a label 'Email-address' above a text input field. The input field contains the email address 'pieter.claerhout@gmail.com' and is highlighted with a blue border. Below the input field is a grey button labeled 'Reset password'.

You can then fill in the username / email and tap "Reset password". This will trigger that an email is sent with a link to the form that allows you to reset the password:



A screenshot of a mobile application interface showing a confirmation message. At the top, the status bar shows the time 3:38, signal strength, Wi-Fi, and battery. Below the status bar is a header with a blue 'Done' link. The main content area displays the text: 'We have send a link to reset your password to pieter.claerhout@gmail.com'. The email address is bolded.

You will then receive an email similar to the one below:



The "from" address is the address that has been configured as the support email in the app settings. The title in the mail is the name of the app as configured on the platform.

Tapping the 'Reset your password' link takes you to the following form which allows you to reset the password:

The screenshot shows a password reset form. It has three input fields: 'Email-address' with the value 'pieter.claerhout@gmail.com', 'Password' with masked characters, and 'Repeat Password' with masked characters and a toggle icon on the right. Below the fields is a 'Reset password' button.

After resetting the password, a message explains you that you can now login to the app with this new password:

You can now login in the app with your new
password and the email-address
pieter.claerhout@gmail.com

Integrating a Custom Entitlements Server

- i** For some customers, the built-in entitlement scenario's don't always offer exactly what they are looking for... others require a connection to an external data source. In those cases, a custom entitlement server can be an alternative.

The implementation and deployment of a custom entitlement server is something that can be offered by Twixl or by Twixl's solution partners, and is based on the technical information below.

If you are looking for the Twixl dev team to implement a custom entitlement server, then please provide us as much details as possible about the project [via this form](#).

1. Prerequisites

To use a Custom Entitlements Server in your app, you will need:

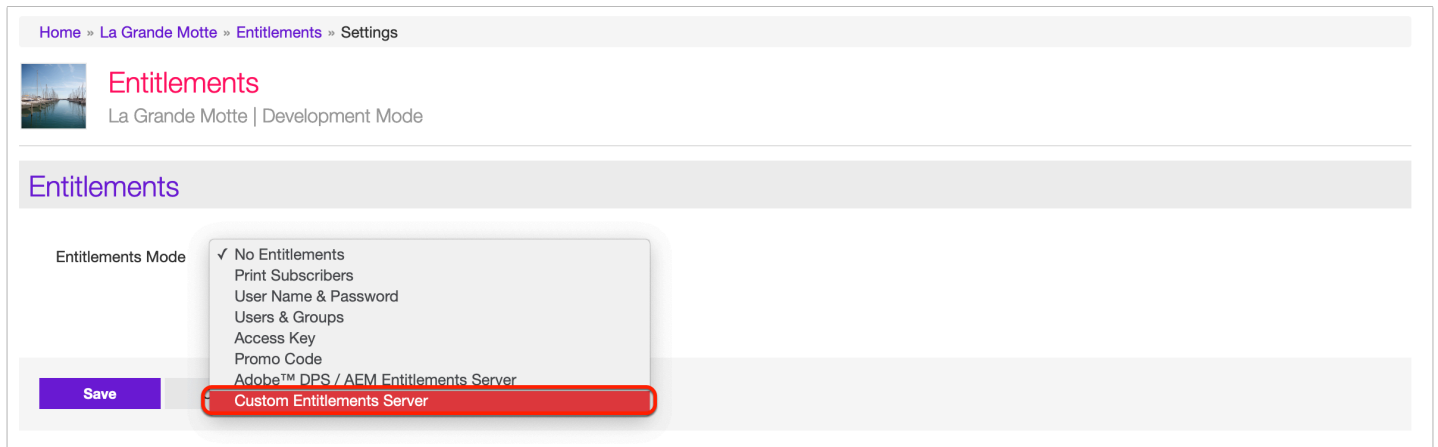
- An App Store app for iOS and/or Android.
- An account and app on the [Twixl platform](#).
- An external web application that implements the Entitlements API.

2. Terminology

There is some specific terminology you need to be familiar with before using entitlements:

- **Entitlements Server:** the external web application that implements the Entitlements API.
- **Token:** a unique identifier for the user on the entitlements server. This can e.g. be a print subscriber number, a region, ... This token will be used to verify which content a user has access to.
- **Device UDID:** in a Twixl app, a device is identified by a unique identifier that stays the same, even when you reinstall the application on the same device. This can be used to track a specific device. Note that this information remains completely anonymous.

3. Configuration



First of all, make sure you have done the necessary configuration on the Twixl platform, as described in the Entitlements [documentation](#).

4. API

4.1. Sample implementations

You can [download example implementations](#) of an Entitlements Server SDK in the following server-side programming languages:

- ASP.NET
- Java
- PHP


4.2. Syntax and payload

More info about the syntax and the payload can be found in the [API documentation](#) on the Twixl platform.

4.3. Entitlements Server URLs

Depending on the server side implementation of the Entitlements Server, there are two different ways the URLs can be constructed. This can be configured on the Twixl platform!

Home » La Grande Motte » Entitlements » Settings


Entitlements
La Grande Motte | Development Mode

Entitlements

Entitlements Mode: Custom Entitlements Server
You are implementing a custom entitlements server

Entitlements URL:

☐ Auto Logout
If enabled, quitting the app will require a re-login.

URL Style: ✓ Query: http://host?do=action&app_id=test
Path: http://host/action?app_id=test

Behaviour

Login Button Title:
If filled in, the button will be shown in the paywall and will enable the user to trigger the entitlements sign form.

Login Button Description:

Logout Button Title:
If filled in, the button will be shown in the paywall and will enable the user to logout from the entitlements.

Logout Button Description:

Save Cancel

4.4. URL Styles

The first style is passing the action in the query string parameter called “do”. This can be done by selecting the following url style in the build settings.

```
http://<url>?do=signin_form&app_id=test
```

This will generate URLs in the following style:

```
http://<url>?do=<action>&param=value
```

If you prefer the action to be a part of the url itself, you can choose the following url style in the build settings:

```
http://<url>/signin_form?app_id=test
```

This will generate URLs in the following style.

```
http://<url>/<action>?param=value
```



IMPORTANT NOTE:

For security reasons it is obviously a requirement to use the https protocol for your connections. Note that this requires that you have a secure SSL certificate installed on your web server.

4.5. signin_form

- **HTTP method:** POST
- **POST Parameters:** None
- **Description:** This API call shows the signin page. Using this page, you can e.g. ask users for their print subscribers credentials or allow them to choose e.g. a region from a list of regions. When you use a link to trigger the signin action, you can specify the link as follows:

```
<a href="?do=signin&region=uk">Select UK</a>
```

When you use a form, there are a few things you need to take into account:

- All form parameters should be passed as GET parameters
- The form should call the action `signin` (using a hidden form field)

The example HTML code for a form can be:

```
<form method="GET">
  <input type="hidden" name="do" value="signin" />
  <input type="text" name="email" />
  <input type="password" name="password" />
</form>
```

When the user clicks the link or submit the form, the application will intercept the URL and parse the query string. It will then add the following parameters to the dictionary of keys and values:

- **app_id**: the unique application identifier of the app (e.g. `com.twixlmedia.AvantGand`)
- **app_version**: the version number of the application (e.g. `1.0`)
- **udid**: the unique Twixl identifier of the device (e.g. `3efad737b4d845ffa6ddc4d484b279e9`).

4.6. signin

- **HTTP method:** POST
- **POST Parameters:**

- **app_id**: the unique application identifier of the app (e.g. `com.twixlmedia.AvantGand`)
- **app_version**: the version number of the application (e.g. `1.0`)
- **udid**: the unique Twixl identifier of the device (e.g. `3efad737b4d845ffa6ddc4d514b279e9`).
- all parameters from the `signin_form` action.
- **Description:**
 - This is the call that will check if the user is entitled or not. If the user is entitled, it should generate a so-called “entitlement token”. This is a unique ID identifying the user and will be used by Twixl Publisher to identify the user for all other entitlement requests.
 - The `signin` call should always return JSON data. There are two results that are possible, depending on the result.
 - If the user is entitled, the following JSON structure should be returned:

```
{"token": "the entitlement token"}
```

If the entitlement fails, the following JSON structure should be returned:

```
{"error": "a message explaining why the user is not entitled"}
```

4.7. `signin_succeeded`

- **HTTP method:** `POST`
- **POST Parameters:**
 - **token**: The token that was returned in the `signin` action.
- **Description:** This is the view that is shown when the `signin` returned a valid token (which means that the user is entitled). You can either show an HTML page with more details based on the token passed as a query string parameter. If you want to provide a link that allows the user to close the popup window, you need to use the following URL:

```
<a href="tp-close://self">Close Me</a>
```

If you want to automatically close the popup when the `signin` succeeds, you can also perform an HTTP redirect to the `tp-close` URL:

```
header('Location: tp-close://self')
```

4.8. `signin_error`

- **HTTP method:** `POST`

- **POST Parameters:**

- **error**: The error that was returned in the signin action.

- **Description:** This action is executed when the signin action returns an error message. This can be used to e.g. indicate that the user is not known or not entitled. The actual error message will be passed as a query string parameter.

4.9. entitlements

- **HTTP method:** POST

- **POST Parameters:**

- **requested_identifier**: the product identifier of the requested issue (will only be filled in when the user does a purchase. It will be empty when the user requests the contents of the kiosk).
 - **app_issues**: the list of issues defined for the app encoded as a json string
 - **app_id**: the unique application identifier of the app (e.g. `com.twixlmedia.AvantGand`)
 - **app_version**: the version number of the application (e.g. `1.0`)
 - **token**: the token that was returned in the signin action
 - **product_identifiers**: a json string containing the list of product identifiers the application is about to show.

- **Description:** This action is called every time the application wants to show the list of issues or when a user tries to purchase an issue. The post request will contain a parameter called "product_identifiers" which is a JSON string containing the list of product identifiers the application is about to show. To convert these to a real list, you need to parse the JSON string. In PHP, this can be done using the [json_decode](#) function:

```
$product_identifiers = json_decode($_POST['product_identifiers']);
```

The result of the entitlements call should be a JSON structure with the following content:

```
{
  "token": "my-entitlement-token",
  "entitled_products": [
    "com.myapp.product1",
    "com.myapp.product2"
  ],
  "mode": "hide_unentitled"
}
```

The parameter **entitled_products** should contain the list of issue identifiers to which the user is entitled.

The parameter **mode** defines how the Twixl Distribution Platform will interpret the results:

- `purchase_unentitled`: make all the products to which you are entitled free if you didn't get them for free yet
- `hide_unentitled`: hides all the items to which you are not entitled from the app

In this call, you should also make sure you properly handle the case where the "token" is empty. This basically means that the user is not entitled.

4.10. force re-login of the user

When the `/entitlements` call returns a token `__token_expired__`, it will force the app to clear the token, thus requiring the user to login again.

Note that this feature requires the Twixl app to be built with **version 15+**.

Custom Entitlements Servers Inquiry Form

For some of our customers, the built-in entitlement scenario's don't always offer exactly what they are looking for... others require a connection to an external data source. In those cases, a custom entitlement server may offer an alternative. The implementation and deployment of a custom entitlement server is something that can be offered either directly by Twixl or by Twixl's solution partners.

If you are looking to implement a custom entitlement server, then please provide us as much details as possible about the project via the form below.

[Yes, I'm interested in the Custom Development of a **Custom Entitlements Server**](#)

In-app purchases & subscriptions

In-app purchases for Android apps

How to configure in-app purchases for your app.

1. Adding in-app purchases in the Google Play Developer Console

In your app, content may be offered for free, or some or all content may be available as an in-app purchase.

If your app offers only free content, there's nothing else to configure but adding your content to the Twixl platform.

If you want to make collections or PDF content items available as an in-app purchase, you will need to create a separate in-app purchase entry for each of the collections you want to offer:

- **Google Play Developer Console** : here you will also set the price for the in-app purchase
- **Twixl platform** : here you will have your different paid collections or PDF content items.

1. Go to <https://play.google.com/apps/publish> and select your app from the list of apps
2. In the section **Monetize**, choose **In-app Products**.
3. Now select **Create Product**.
4. Add the **product ID**, **name** and **description**, set the price, and Save the information.
5. The in-app purchase has been saved, but it is still inactive at this point.
6. Select the in-app purchase you just created once more, and you'll see you can now activate it.

Google Play Console

Search Play Console

[In-App Purchases](#)

App integrity

Internal app sharing

Advanced settings

Grow

Store presence

Store performance

Quality

Ratings and reviews

Android vitals

Monetize

Products

App pricing

In-app products

Subscriptions

Promo codes

Financial reports

Monetization setup

← In-app products

3 Create in-app product

* – Required fields. Enter all fields in English (United States)

Product ID *

4 com.casablancapublishing.rickscafe.oct2020

42 / 136

Must start with a number or lowercase letter, and can contain numbers (0-9), lowercase letters (a-z), underscores (_), and periods (.).

You can't change or reuse a product ID after the product has been created. [Learn more](#)

Default – English (United States) – en-US

Manage translations

Product details

Name *

October 2020

12 / 55

Description *

Read the latest news from Rick's Cafe

37 / 200

In-app products

Offer products for sale in your app for a one-off charge, like extra lives, or access to premium content. [Show more](#)

Search products by name or ID

[Import](#)
[Export](#)
[Create product](#)

Product name	Product ID	Price	Last updated	Status	
October 2020	com.casablancapublist	EUR 10.00	May 5, 2021	Inactive	5 →

← In-app products

October 2020

Delete

Inactive · com.casablancapublishing.ricksafe.oct2020

* – Required fields. Enter all fields in English (United States)

Default – English (United States) – en-US Manage translations

Product details

Name *

October 2020
12 / 55

Description *

Read the latest news from Rick's Cafe
37 / 200

Price

Default price *

EUR 10.00

Select pricing template ▾ Edit price

6

Save changes **Activate**

2. Adding in-app purchases to the Twixl platform

For each of the in-app purchases defined in Google Play, you also need to have a collection or PDF content item on the Twixl platform. Add the same identifier under 'Purchase Info' for your collection/PDF.

You can also add the title and extra info (by platform) that will be displayed in the paywall.

Purchase Info

Product Identifier

com.casablancapublishing.ricksafe.oct2020

If the collection is a purchase, this should match the product identifier for the In-App Purchase in App Store Connect and Google Play. Leave empty to use the suggested identifier. Only characters, numbers and dots should be used.

☐ Free Of Charge

If enabled, all users can access the contents in this collection. If disabled, the users will be presented with a paywall before they can access the contents.

Purchase Title (iOS)

Purchase Info (iOS)

Purchase Title (Android)

Purchase Info (Android)

! IMPORTANT:

Make sure the product identifier in the Twixl platform matches exactly with the one you defined for the in-app Product in Google Play.

3. In-app purchase testing

After you added your in-app products for your app in Google Play, you can test your purchases:

1. Create a Google Play build of your app
2. Upload the app to Google Play as an alpha or beta
3. Create license test accounts for authorized users (in Developer Console, go to Settings > Account details, then in the License Testing section, add the addresses to the field "Gmail accounts with testing access")
4. Test your in-app purchases with one of the test accounts – these allow you to purchase any of your in-app products without being charged

! IMPORTANT NOTE:

Sometimes you may get the error "This version of the application is not configured for billing through Google Play. Check the help center for more

information". Note that Google takes a while to process applications and update them to their servers (anywhere from a couple of hours to a day). So after uploading the APK on Google Play, you may have to wait a few hours before the in-app products will allow to be purchased.

4. Adding subscriptions in your app

See this article: [Working with subscriptions for iOS & Android](#)

In-app purchases for iOS apps

How to configure in-app purchases for your Twixl app.

1. Adding in-app purchases in AppStore Connect

In your app, content may be offered for free, or some or all content may be available as an in-app purchase.

- If your app offers only free content, there's nothing else to configure but adding your content to the Twixl platform.
- If you want to make collections or PDF content items available as an in-app purchase, you will need to create a separate in-app purchase entry in AppStore Connect for each of the collections you want to offer. Note that the in-app purchase type for collections needs to be 'Non-Consumable'.

Select the in-app purchase you want to create:

☐ **Consumable**
A product that is used once, after which it becomes depleted and must be purchased again.
Example: Fish food for a fishing app.

☒ **Non-Consumable**
A product that is purchased once and does not expire or decrease with use.
Example: Race track for a game app.

☐ **Auto-Renewable Subscription**
A product that allows users to purchase dynamic content for a set period. This type of subscription renews automatically unless cancelled by the user.
Example: Monthly subscription for an app offering a streaming service.

☐ **Non-Renewing Subscription**
A product that allows users to purchase a service with a limited duration. The content of this in-app purchase can be static. This type of subscription does not renew automatically.
Example: One-year subscription to a catalog of archived articles.

[Learn more about In-App Purchases](#) [Cancel](#) [Create](#)

1. Enter a reference name (that will be used for reporting purposes only), e.g. October 2020
2. In the Product ID field, enter a unique identifier that will be used for reporting. It can be composed of letters and numbers. Usually this will also be a reverse DNS name like the app identifier: e.g. `com.casablancapublishing.ricksafe.oct2020`
3. Set pricing for the collection by selecting a price tier

2. Adding In-App Purchases

You can also add the title and extra info that will be displayed in the paywall.

! IMPORTANT:

Make sure the product identifier in the Twixl platform matches exactly with the one you defined in AppStore Connect.

3. In-app purchase testing

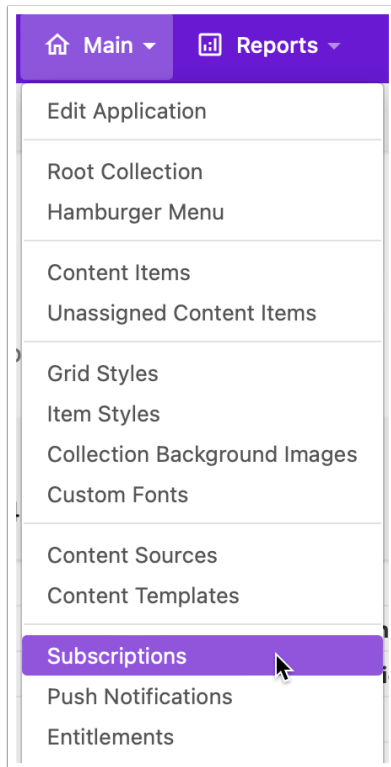
In order to test in-app purchases with an Ad Hoc build of your app, Apple provides a sandboxing environment. To do this:

1. Build and install an Ad Hoc version of your app.
2. Create a test user in AppStore Connect.
3. Logout from the App Store for your regular account.
4. Try to make a purchase and login with the test user you created.
5. You should be able to make the test purchase in the testing ('sandbox') environment.

4. Adding subscriptions in your app

For more info about this, see [Working with subscriptions for iOS & Android](#)

Subscriptions for iOS & Android



1. Standard or all access subscriptions

If you want to offer a subscription in your app, there are two options:

- **Standard Subscription:** this is the traditional type of subscription where you get access to new collections or PDF content items (i.e. issues) that are added during the period of your subscription.
- **All Access Subscription:** this can be compared to e.g. Spotify or Apple Music type of subscriptions: you get access to all content in the app, as long as your subscription remains valid.

Subscription Settings

Subscription Type * ☐ No Subscriptions
☒ Standard Subscription
☐ All Access Subscription

App-Specific Shared Secret

This can be found on [App Store Connect](#) under the "Manage In-App Purchases" section in your app.

Save

1.1. Generate App-Specific Shared Secret (iOS only)

1. Go to <https://appstoreconnect.apple.com> -> My Apps -> Select your app -> Manage In App Purchases -> Select '**App-Specific Shared Secret**'.
2. Click '**Generate App-Specific Shared Secret**'.
3. Copy the Shared Secret.
4. On the Twixl platform, select **Subscriptions** from the app's menu.
5. Edit the subscription settings, select the type you want to offer, and paste the shared secret in the corresponding field.

iOS App

1.0 Prepare for Submission

[Add macOS App](#)
[Add tvOS App](#)

General

[App Information](#)
[Pricing and Availability](#)
[App Privacy](#)
[Ratings and Reviews](#)
[Version History](#)

In-App Purchases

In-App Purchases

i

Your first in-app purchase must be submitted with a new app version. Create your in-app purchase, then select it from the app's In-App Purchases section under App Store and click Submit. [Learn More](#)

Once your binary has been uploaded and your first in-app purchase has been submitted for review, additional in-app purchases can be submitted using the table below.

In-App Purchases (5) [+](#)

Q Search

1

App-Specific Shared Secret

Reference Name ^	Type	Product ID	Status
------------------	------	------------	--------

App-Specific Shared Secret

The app-specific shared secret is a unique code to receive receipts for only this app's auto-renewable subscriptions. You may want to use an app-specific shared secret if you're transferring this app to another developer, or if you want to keep your primary shared secret private.

2 [Generate App-Specific Shared Secret](#)

Done

App-Specific Shared Secret

The app-specific shared secret is a unique code to receive receipts for only this app's auto-renewable subscriptions. You may want to use an app-specific shared secret if you're transferring this app to another developer, or if you want to keep your primary shared secret private.

Shared Secret

Generated

3

May 5, 2021

[Regenerate](#)

Done

Subscription Settings

4

Subscription Type *

☐ No Subscriptions

Use this if you don't want to support subscriptions in your app.

☒ Standard Subscription

Use this if you want to support magazine/newspaper-style subscriptions in your app.

Everything that is published while the user has a subscription will be offered free-of-charge.

☐ All Access Subscription

Use this if you want to offer free access to all content while the user is subscribed.

When the subscription ends, the user will have no access to the content anymore.

App-Specific Shared
Secret
This can be found on [App Store Connect](#) under the "Manage In-App Purchases" section in your app.

Save

Cancel

1.2. Latest content

New subscribers will automatically receive the most recent paid collection or PDF added to your app (for standard subscriptions only).

2. Adding a duration

To create a duration for an auto-renewable subscription, you need to specify the following details:

- **Duration:** The interval of the subscription. On iOS, it is either 1 week, 1 month, 2 months, 3 months, 6 months or one year.
- **Product identifier:** A unique reverse-DNS for this duration. This product identifier has to be the same as the one entered in AppStore Connect / Google Play Developer Console, for example `com.mycompany.mymagazine.subscription.1month`
- **Title (iOS & Android):** The name a reader will see in the list of subscriptions on the device.
- **Description (iOS & Android):** The description a reader will see in the list of subscriptions of the device.

Add Duration

Duration

Duration * 1 Month

Product Identifier com.mycompany.mymagazine.subscription.1month

This should match the product identifier for the In-App Purchase in App Store Connect and Google Play. Leave empty to use the suggested identifier. Only characters, numbers and dots should be used.

iOS

Title (iOS) *

Description (iOS)

Android

Title (Android)

Description (Android)

Add Cancel

3. Adding a subscription

3.1. In App Store Connect (iOS)

1. Go to <https://appstoreconnect.apple.com> -> **Manage Your Application** -> **Select your application** -> **Manage In-App Purchases** -> **Create New**
2. Choose the correct type of subscription (usually Auto-Renewable) and create a new subscription family or select an existing subscription family.
3. Just like on the Twixl platform, you have to add a duration for the subscription. You need to use the same product identifier & matching duration as on the Twixl platform, for example `com.mycompany.mymagazine.subscription.1month`

3.2. In the Google Play Developer Console (Android)

1. Go to <https://play.google.com> -> navigate to the In-App Products for your app.
2. Create the same subscription with the same product identifier & matching duration as on the Twixl platform, for example `com.mycompany.mymagazine.subscription.1month`

Metered Access

If your app has purchasable content and you want to offer a free preview of that content to the app user, you can use the metered access feature to make your life easier. Note that this feature requires TP19 or higher.

Principle

The metered access feature makes it easier to create links between preview and paid collections. The app will check whether the user has access to either the preview or the purchased collection. As a publisher you have complete control over the content that will be shown to the user because the linking is performed between collections. These collections can contain most types of Twixl content items*.

Workflow

1. [Enabling the metered access feature \(app setting\)](#)
2. [Defining collections as purchase or preview](#)
3. [Linking collections to a preview or purchase collection](#)
4. [Adding content to the collections](#)
5. [Adding purchase button](#)

1. Enabling the metered access feature


To be able to use the metered access feature, you have to enable this function in the app settings.

How:

Go to app settings via:

Main > Edit Application > Application Behavior > check the 'Metered Access' box > Save

Application Behavior

- ☐ **Prompt for App Store Rating**
Prompts the user to rate your app in the App Store.
- ☐ **Blur snapshot in iOS & Android app switcher**
Enable this option to hide potentially sensitive information in the app switcher screen.
- ☐ **Keep All Data Offline**
Requires users to download all app contents on first startup. Ideal for certain in-house apps, but not recommended for App Store apps, as such an app may be rejected by Apple.
- ☐ **Sharing on Social Media**
When allowing sharing on social media, users can share articles on social media.
- ☐ **Encrypt PDF Files**
Enables on-the-fly encryption to PDF content items.
- ☐ **Save read position in PDF files**
Save the last page read in PDF content items.
- ☐ **Externally Managed Content**
Enable this when you have an external system that manages the content.
This puts the content items and collections of your app in read-only mode on the Twixl Distribution Platform.
-  ☐ **Metered Access**
Enables metered access for collections.

2. Defining collections as purchase or preview

Once the metered access feature has been enabled, an extra 'preview' option appears when creating or editing a collection. When creating or editing a collection you have 3 options:

- Free of charge
- Purchase
- Preview

Collection Details

Collection Type	<div>Free of charge</div> <div> <div>Free of charge</div> <div>Purchase</div> <div>Preview</div> </div>
Name *	
Title	
Product Identifier	<div>7799715d1f2dd42e3f0af5299a5bd5da</div> <div>Leave empty to use the suggested identifier. Only characters, numbers and dots should be used.</div>
Published On	<div>10/01/2023 10:23</div> <div>This is used to determine if a collection is part of a subscription or entitlement.</div>

When a collection is marked as Preview, you will be able to link it to an existing Purchase Collection in Collection Details while creating the collection. A collection marked as Purchase will offer the possibility to link to an existing Preview collection via the dropdown. Linking the collection is not mandatory. You can link to a collection afterwards too. You can complete all collection fields as usual and save it.

Collection Details

Collection Type

Purchase

Free of charge: a free collection
Purchase: a purchasable collection
Preview: a preview collection for a purchase collection

Name *

Title

Product Identifier

Leave empty to use the suggested identifier.
Only characters, numbers and dots should be used.

Published On

This is used to determine if a collection is part of a subscription or entitlement.

Preview Collection

-- unassigned --

Select the collection to link to. A preview collection can only be linked to one purchase collection.

Whether a collection is free, preview or purchase is shown in the Type column in collection overview in Home or via Main > Collections. If there is already a collection linked to this collection, will also be shown in the 'Linked To' column in this overview.

Collections

Add Collection

	Title	Type	Options	Items	Grid Style	Linked To
<input type="checkbox"/>	<div>FREE</div> Free collection 1 didier.test.app.free_collection_1 Published 21 hours ago	Free of charge		1	Default	
<input type="checkbox"/>	<div>CONTENT</div> Other Content didier.test.app.othercontent Published 21 hours ago	Free of charge		3	Default	
<input type="checkbox"/>	<div>REVIEW</div> Preview collection 1 didier.test.app.preview_collection_1 Published 6 days ago	Preview		2	Default	
<input type="checkbox"/>	<div>PURCHASE</div> Purchase collection 1 didier.test.app.purchase_collection_1 Published 6 days ago	Purchase		1	Default	Preview collection...


3. Linking collections to a preview or purchase collection

Some possibilities to create links between collections:

- Via edit in a collection and select the matching collection
- By creating a new collection and selecting the matching collection

- By clicking on 'Create' in the Collection Details of a collection and creating a new collection

Collection details

Name	purchase collection 2	Created On	2023-01-10 10:32
Title	Purchase Collection 2	Published On	2023-01-10 10:31
Product Id	didier.test.app.purchase_collection_2	Monolithic Download	No
Twixl ID	27876	Requires entitlements	No
Twixl UUID	854aa44c2e367924abda165c116b89aa	Grid Style	Default
External ID	-	Open In	Browse Mode
Collection Type	Purchase	Preview Collection	Create 
Browser Client URL	https://browserclient-staging.twixlmedia.com/5b05a146eed6...		
Twixl Link	tp-collection://purchase%20collection%202		

4. Adding content to the collections

In Preview collections you can add your content as usual. For preview collections, the type of content items that are allowed are:

- HTML Article
- Twixl Article
- PDF
- Image
- Movie
- Vimeo Movie
- YouTube Movie
- Inline Web Viewer
- Embedded Web Viewer
- Placeholder
- Weblink

This means you are free to provide preview content that maximises purchase attraction.

5. Adding purchase button

How you integrate and design the purchase button is also completely customisable. If you create a Twixl article (InDesign), you can integrate the "tp-paywall://" scheme via the Twixl Plugin in your design. This will call the paywall function and once the content has been purchased, the app will only show the purchase collection. After the purchase of a pcollection has been completed, its preview collection will no longer appear.

Other content types will require to add a web link containing a "tp-paywall://" scheme.

Click [here](#) for more information on these custom url schemes.

**Final check:**

Before the go live of your publication, make sure both your content items both in the purchase and the preview collection are in the published status.