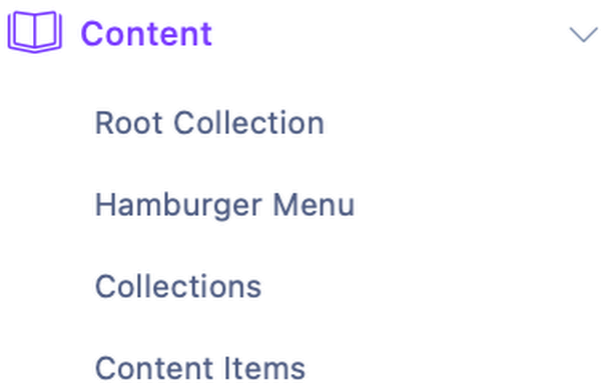
Content in your app

Table of Contents

How content is structured in Twixl

Getting started with content for your app

Now that you have designed the interface for your app, you can start adding content to your app. In this chapter, we will discuss all elements that you can find in the '**Content**' section (in the menu bar on the left) on the Twixl platform. As already mentioned, content is added to collections in the Twixl ecosystem.



Though all listed items on this page match the order of the content section on the Twixl platform, we advise to first think about the structure your app needs. Start by creating collections, add content items and finally link to these collections from the root collection and hamburger menu.

Root Collection

The Root Collection is a special collection type as it can be considered to be the Home page of the app. When a user opens the app, the Root Collection will be displayed and it is the page where one goes back to if one wants to go back to the first entry point. Depending on the purpose of your app, the Root Collection can list different navigation options (e.g. a sales app) or new items (e.g. news app) and is mandatory for every app published with Twixl Publisher.

Hamburger Menu

The Hamburger Menu is another special collection type as it will display a hamburger icon in the right corner of the menu bar throughout the app. This makes the Hamburger

Menu ideal to navigate quickly to other sections of the app or add some quick action buttons (login/logout, search, library, downloads...). By default, the Hamburger Menu is disabled as it is an optional collection.

Collections

Collections group all your content items that belong together. A collection can be a issue of a magazine where the pages are all different content items. But a collection can also be a set of similar magazines (e.g. one year per collection) where every content item is an entire magazine (depending on your content item type). Collections can also target specific users by handling different subjects (e.g. an HR section in an enterprise app). Finally, Collections can display a list of the content items that are uploaded to the collection (Browse view) or can start by showing the first content item in the collection (Detail view).

Content Items

All content that you want to offer to your user, can be found in Content Items. Here you can upload different types of files like InDesign export files, HTML files, PDF files, movies, images,... and add them to collections.

Advanced

You can also think about uploading content automatically via:

- Content Sources
- API integration

Collections

1. Organizing your content in 'collections'

Collections let you organize the content in your app as you can upload content items (and aliases, links, web content) into them. Especially when your app naturally becomes filled with more content, having all of your content items in the Root Collection becomes cluttered fast. So it is a good thing to think about your app structure and start using different collections from the beginning.

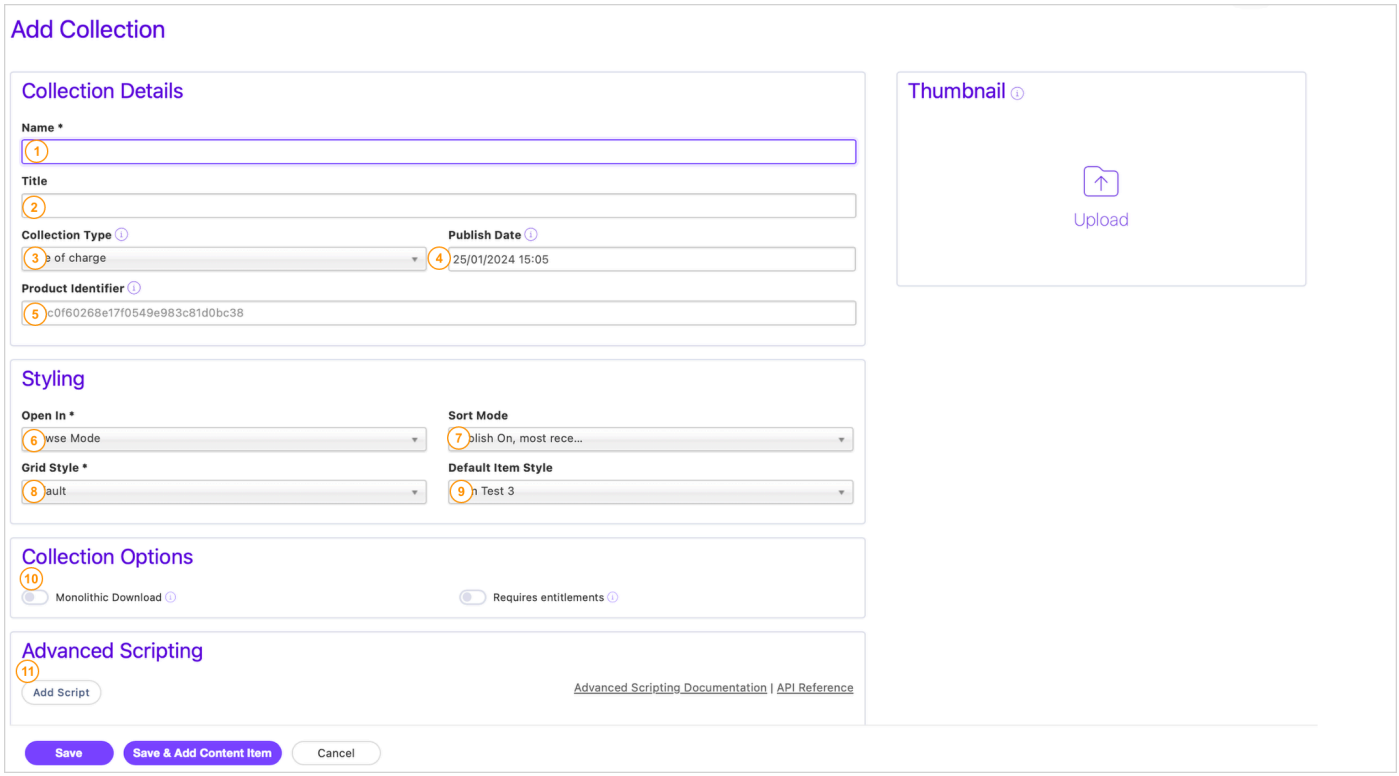
Here are some use cases for collections:

- 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 always has to be present and cannot be deleted. Therefore, the Root Collection is a special kind of collection.

2. Creating a collection

- Select 'Collections' in the left menu bar.
- Click on "+" next to the 'Collections' title.



Different sections:

1. **Name:** This is a required field and needs to be unique as it will define your collection throughout the project.
2. **Title:** The title will be shown to the end user and needs to be clear for them.
3. **Collection type:** There are 2 types of collections:
   Free of charge: One doesn't need to purchase these collections nor has to have a running subscription. However, you can still add an entitlement request if you want to hide your content for non-entitled users.
   Purchase: Here the paywall will be called if the user hasn't bought this collection already.

> If your app has metered access enabled, you can also select a 'preview' type here. Preview allows you to attract your reader to buy a magazine by offering them some free content. Read more about metered access here.

4. **Publish date:** If you have subscriptions in your app, the publish date will determine whether a collection is included in the subscription of the reader or not.
5. **Product identifier:** Use the presented identifier in your store or copy the one from the store here so purchases can be identified.
6. **Open in:** There are 2 modes as to how a collection will open:
   Browse mode: Will show the content items of that collection by their thumbnails in a grid so the user can browse through the items and select the ones they want to read.

Detail mode: Will open the first content item in the list directly and the user needs to swipe to the next content item to continue reading.

7. **Sort mode:** Different modes are available here and these will define the order the content items will be displayed for the user both in browse and detail mode. By selecting 'Manual' you can manually change the order of the content items in the list using drag and drop.
8. **Grid style:** Select the grid style you want to use for a collection that will be opened in browse mode. Leave to 'Default' when your collection is in detail mode.
9. **Default item style:** When adding your content items to the collection, you will be able to select an item style for each of your content items. By selecting a default item style here, this item style will be presented as the default that can be changed if needed. You can select 'default' when your collection is in detail mode.
10. **Collection options:** You can enable 2 special options for each collection:
    Monolithic Download: Downloads all of the content items in the collection at once so the user can read this offline too. This takes up space on the user's device however, so something to take into account. A user can always download a complete collection by long pressing its thumbnail in the browse grid.
    Requires entitlement: If Entitlement is enabled for your project, you can offer the content in your collection to a selected audience via a login entitlement pop-up.
11. **Advanced scripting** (optional): If you have the optional advanced scripting mode enabled in your account, this allows you to filter specific content based on certain criteria. More details here.

You can also add a thumbnail to identify your collection on the platform or to use it in a collection link. If left empty, Twixl will use the thumbnail of the first content item.

When you are ready with the collection settings you can either 'Save' or 'Save and add content item' which will allow to immediately upload content items to your new collection.

> All collections can be offered as In-App Purchases (except for the Root Collection). In that case, make sure your product identifier matches the one in App Store Connect Connect and/or Google Play.

3. Uploading content to your collection

1. Add content items to your collection by clicking '+' next to your Collection Content title.
2. You can also immediately create another collection and link to this one via this option.
3. Once your content item is uploaded to the collection, it will appear in the Recently Updated list. This makes it easy if you still want to change settings or metadata in your

content items. The most recently updated content items are shown first so you don't have to search in your list of content items if they are sorted differently.

4. The sort order of your content items, i.e. how they will be presented to the end user, is shown in Collection Content.
5. Need to change anything in the settings of your collection? Then click on 'Collection Settings'.

4. Related articles & chapters

- Content Items
- Root Collection
- Hamburger Menu
- Designing your app
- Sharing on social media
- Advanced Scripting
- Entitlements

Content Items

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

Click on "+" in one of these Platform locations:

- Content > Collections > Collection > Content Items
- Content > Content Items
- Content > Unassigned Content Items

Which content types can be uploaded?

- **HTML articles (`.html` or `.zip`)**
- **Twixl InDesign exports (zipped `.article` or `.publication`)**
- **PDF files (`.pdf`)**
- **Images (`.jpg` or `.png`)**
- **Movies (`.mp4)`**
- **YouTube movies (URL)**
- **Vimeo movies (URL)**
- **Web links (URL)**

How to prepare your content for upload

For Content Items

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` (after the upload the publication will be split into different content items for each of the articles). Before uploading to the platform, make sure to zip your .article or .publication file.
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 errors 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. **Collection Link:** Create a link to an existing collection.
9. **Content Item Alias:** Links to the same content item from different locations, even with different Item Styles.
10. **Web link:** Links to a web page - limited to 255 characters. If you have web links that are longer, you can use a URL shortener like Tiny URL or Bitly.

For Browse Grid Items

1. **Inline Web Viewer:** The URL of the content that you want to display inline.
2. **Embedded Web Viewer:** Will display uploaded web content inline. The web content should be uploaded as a zipped file in the same way as an HTML article (see above).
3. **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.

Add content items to your app

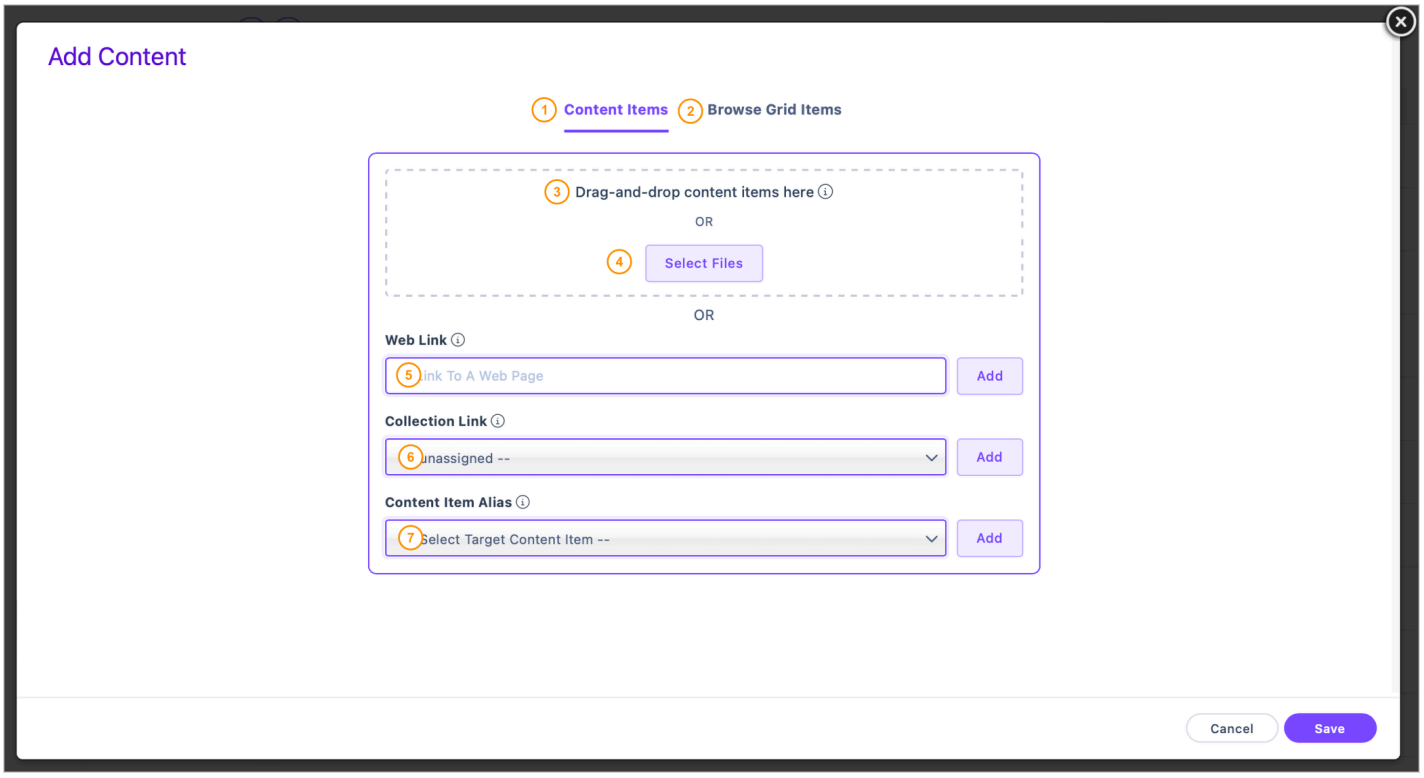
1. Start the upload

Click "+" here:

- Content > Collections > [Collection Name] > Content Items
- Content > Content Items

2. Upload or link to your content

A pop-up opens...



Select where your content items will be shown:

1. In detail mode

2. In a browse grid

Then select the content items you want to add:

3. By drag and drop in this zone

4. By selecting files from your computer

And/or:

5. Insert a web link & click "Add"

6. Select a collection to link to & click "Add"

7. Select a content item to create an alias here & click "Add"

And/Or in Browse grid items:

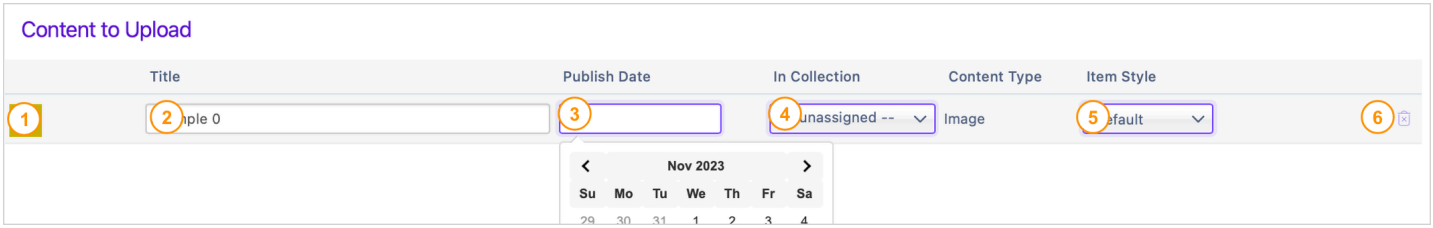
1. Via drag and drop in this zone

2. Via selecting files from your computer, either a Placeholder image, or an HTML Embedded Web viewer (zipped).

3. By entering the URL for an Inline Web Viewer & click "Add"

The items you added will appear at the bottom.

Here you can already add some additional metadata, if you wish:



1. Update the thumbnail by clicking on the proposed thumbnail

2. Update the title

3. Select a date by clicking in the Publish Date field & click "Save"

4. Select a collection (in case you didn't start the upload from within a Collection)

5. Select an item style

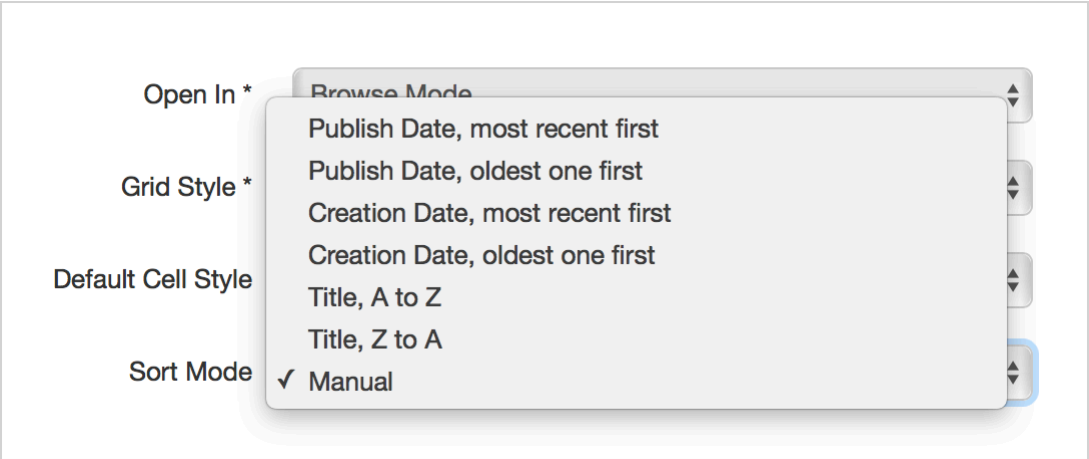
6. Delete this item

> ⚠ The items are now added and ready for uploading. However, they will only be uploaded if you confirm via the "Save" button.
>
> Click "Cancel" or "X" if you want to exit without uploading.

Push content from an external solution

Push content from solutions like WoodWing Inception, Canvasflow, vjoon K4 etc.
Click here for more information.

Sorting content items in a collection



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.

Related

- Collections
- Item Styles

Special collections

Root Collection

The 'Root Collection' is a special type of collection as it defines what the user sees first when opening the app. That's why the 'Root Collection' cannot be deleted.

However, all settings and uploads of content items are similar to those of normal collections.

You can access the Root Collection page on the Twixl platform via Content > Root Collection.

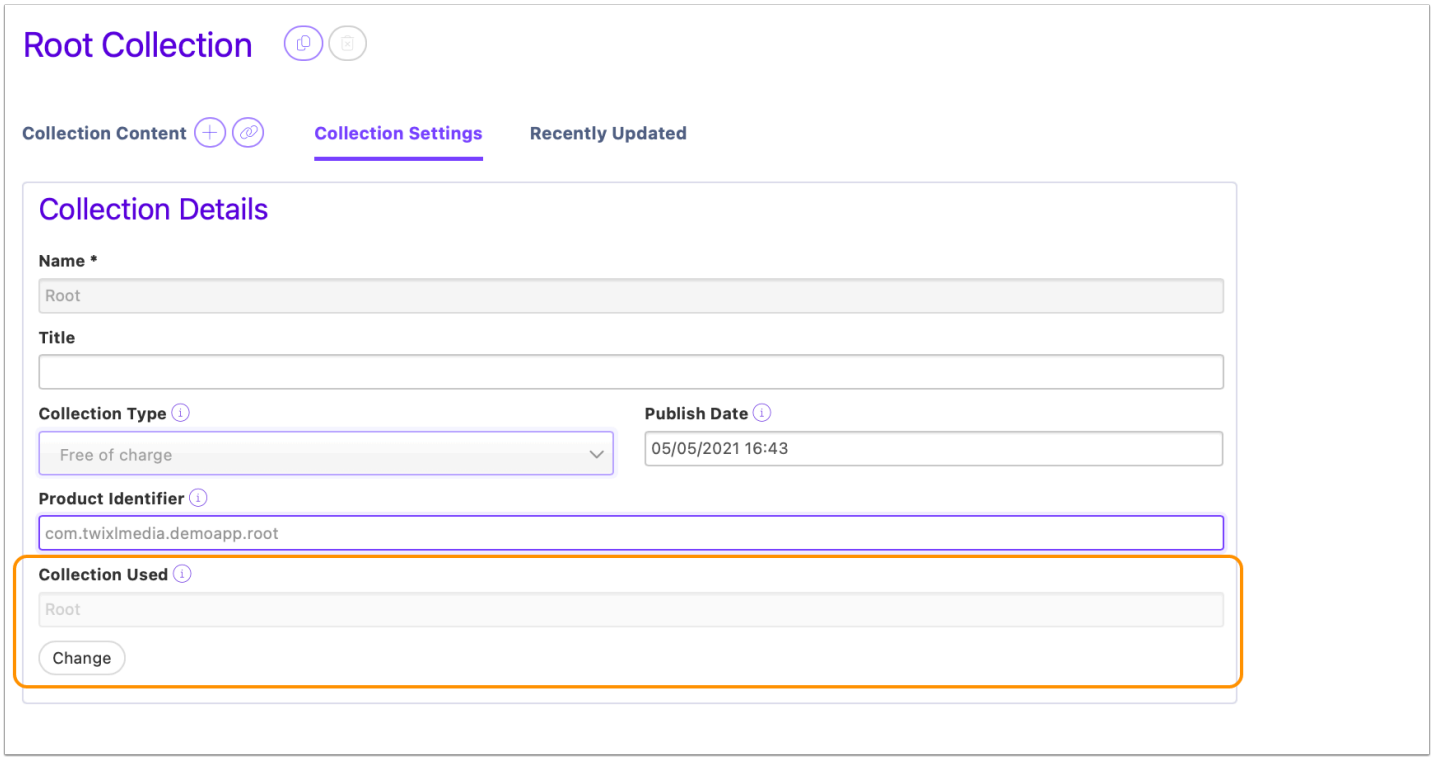
As the Root Collection is the first thing users will see, it is clear that this entry point defines your app's success as we want to attract the user to start using the app.

You can upload content items (also browse grid items) to your Root Collection directly. But we advise to think about Collection links, Content Item aliases... in stead of uploading the same content multiple times. If you ever need to change the content in a content item, you will only need to change this once if you link to that content item or collection via links and aliases.

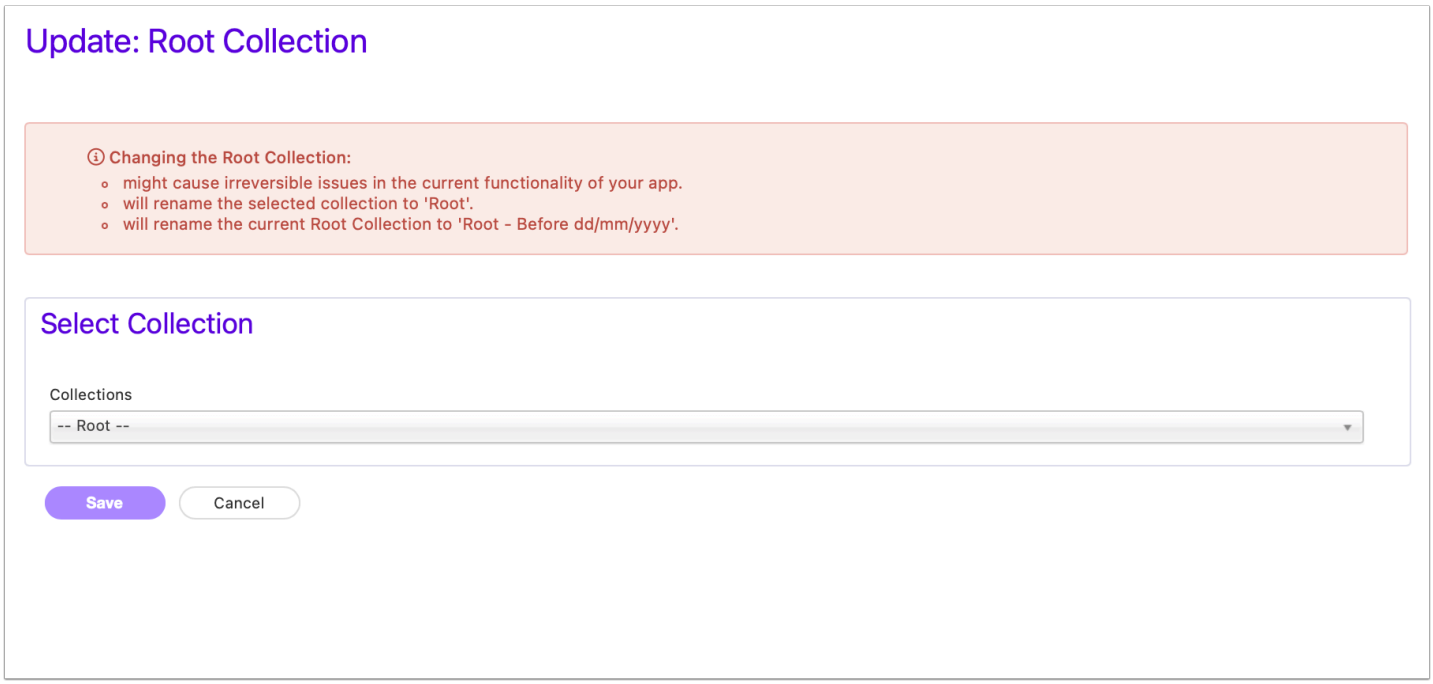
Changing your Root Collection

In the life cycle of your app, you might need to update your Root Collection on a regular basis (e.g. new approach for the app). At that point, you can create a normal collection and add the new structure and links in there. After testing, you can decide to define that collection as the new Root Collection.

Go to Content > Root Collection > Collection Settings > Collection used > Change



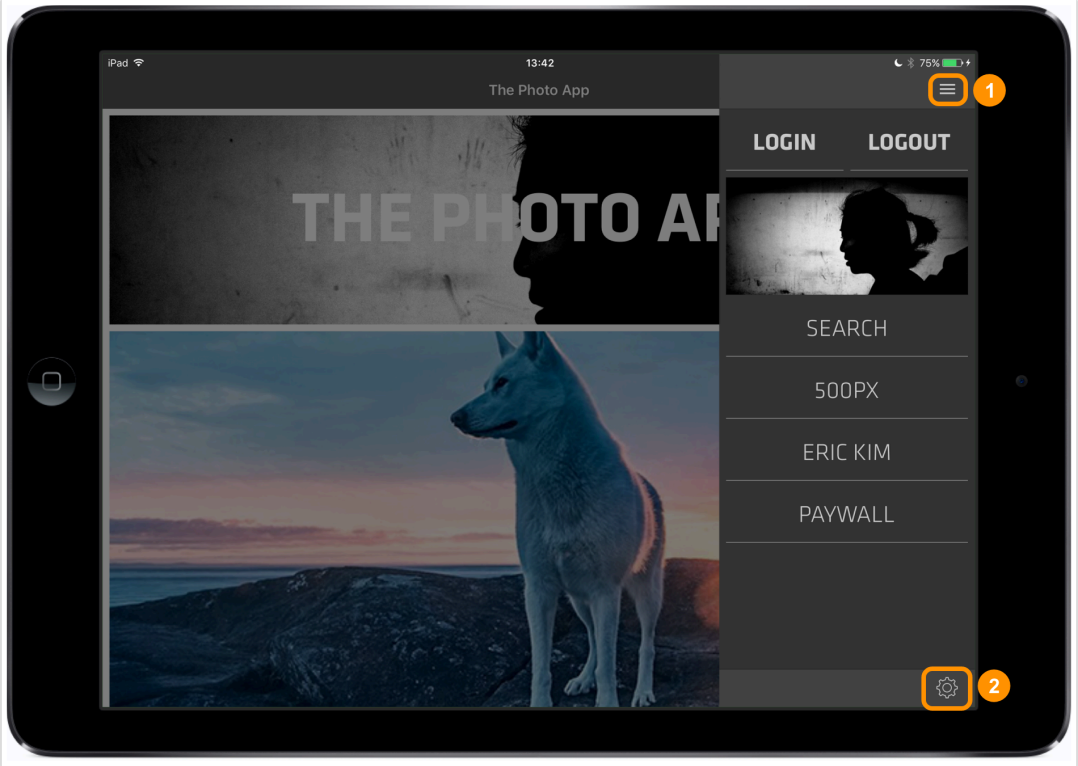
You will get an extra warning explaining the consequences. The collection you select will be renamed to 'Root'. The former root collection will be renamed to 'Root - Before dd/mm/yyyy'.



After your final approval, the new Root Collection will be displayed. Make sure all your content items in it are set to 'published'!

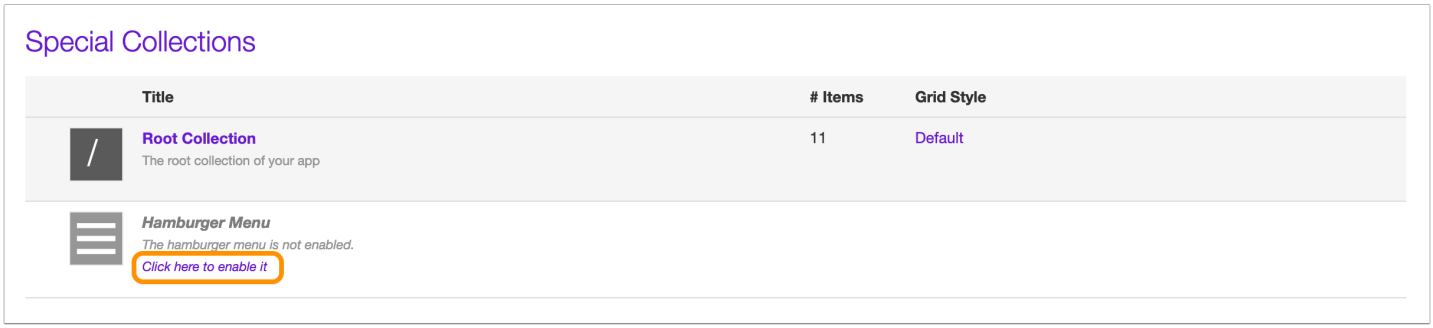
Hamburger Menu

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?

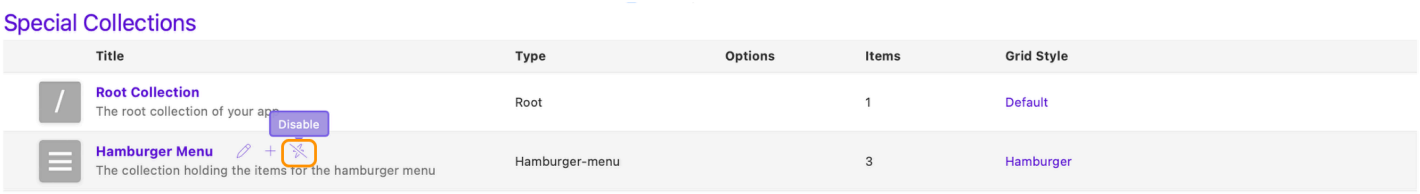
The Hamburger Menu isn't enabled by default.
You can enable it on Twixl Platform > Home > Special Collections > Hamburger Menu.



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 the **disable icon**.



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.

5. Upload Content Items

For more information on uploading Content Items, click here.

6. 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 Item 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.

7. 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.

8. Related

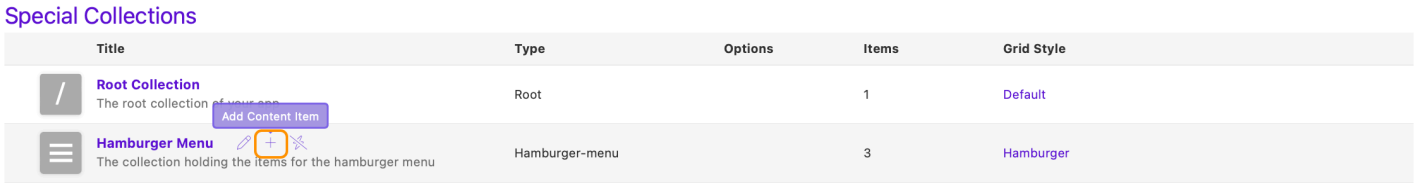
Uploading content items to the Hamburger Menu

Hamburger Menu: Uploading content items

There are 2 ways to upload Content Items to the Hamburger Menu:

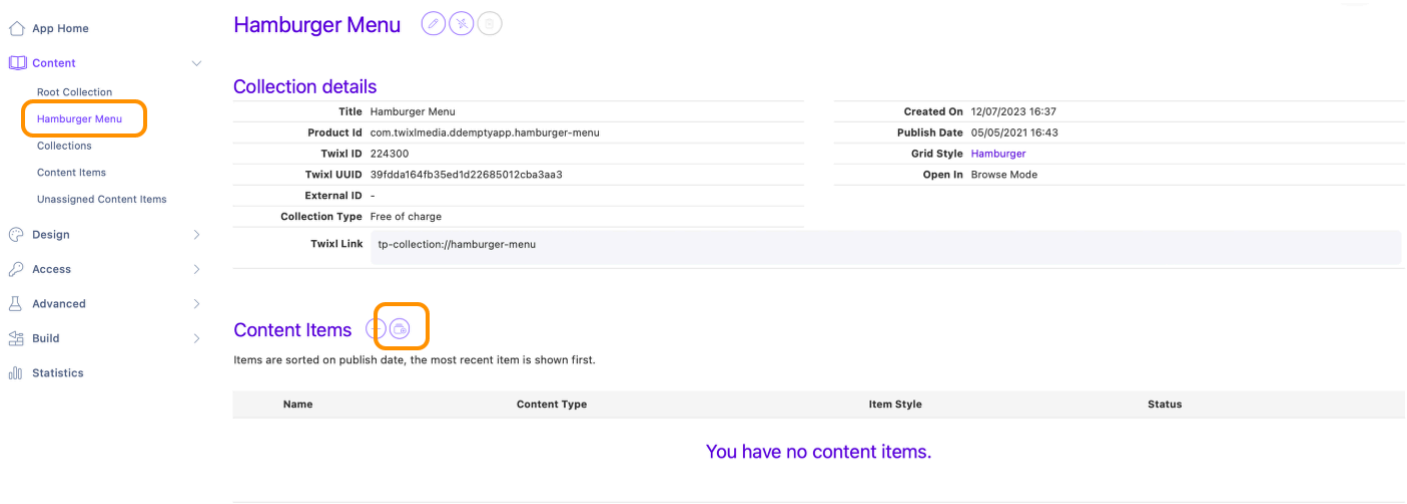
1. Via Application Home

Click the 'add' button in Special collections > Hamburger Menu

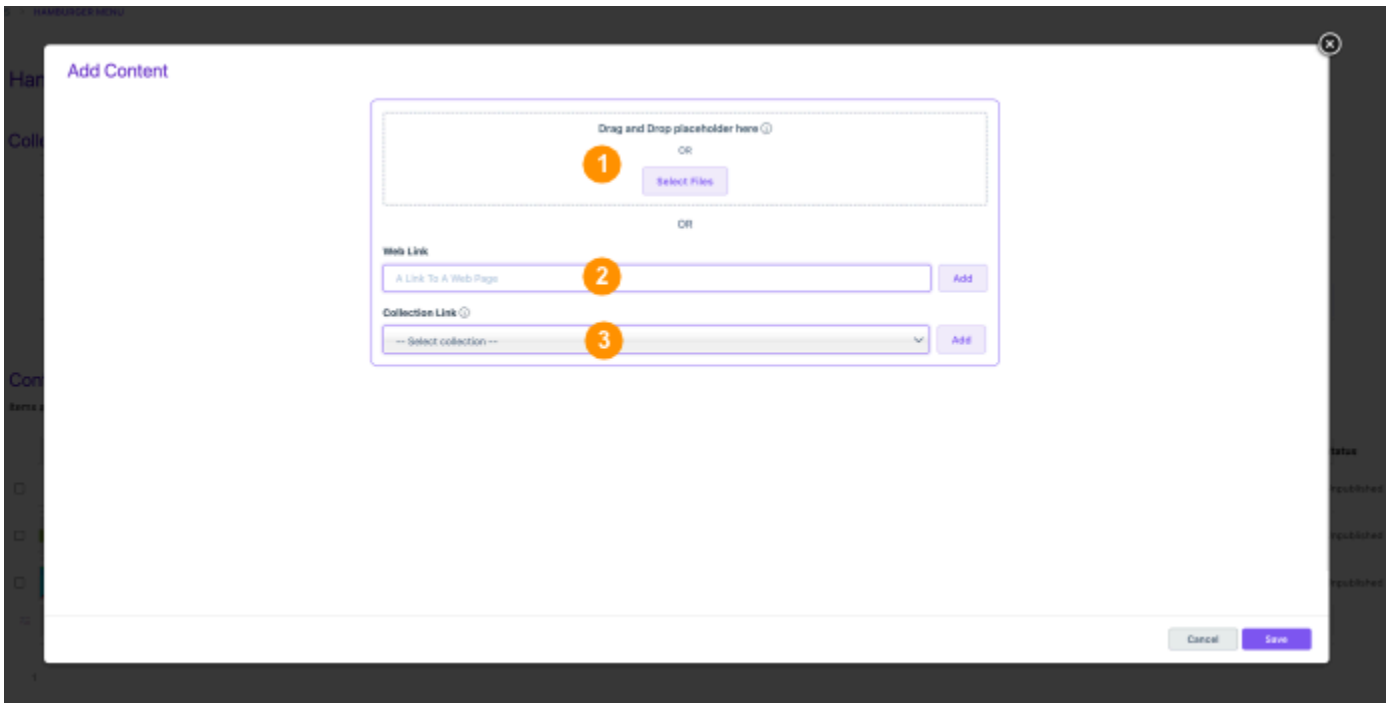


2. Via Menu

Select Content > Hamburger Menu and click the 'add multiple' button, regardless of the number of items or the type of content you want to add.

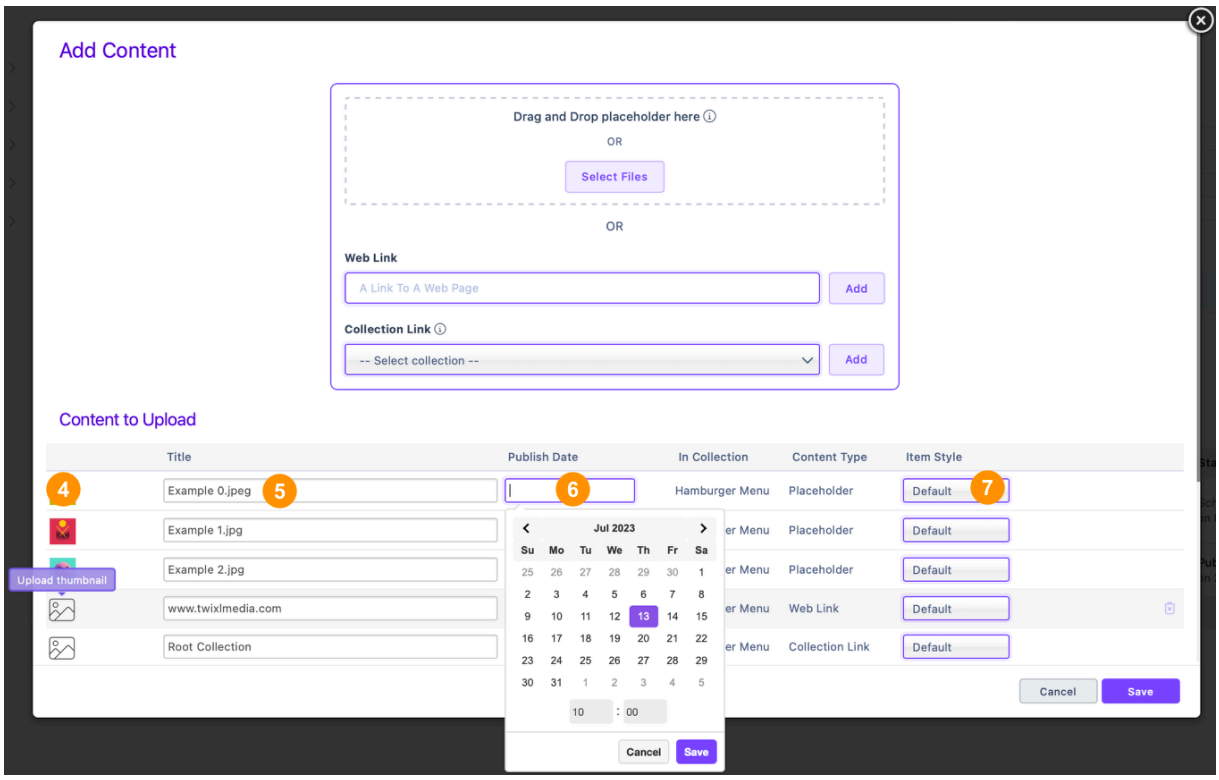


A pop up appears in which you can upload a Placeholder(s), Web Link(s) and/or Collection link(s).



1. **Placeholder**
   You can either drag&drop a file on this area or click 'select files' to add files from your drive.
2. **Weblink**
   Enter a link to a web page and confirm by clicking 'Add'.
3. **Collection link**
   Click on the dropdown icon, select the collection link and confirm by clicking 'Add'.

All the items you add will be listed below in a practical table that allows you to change some of the most important fields.



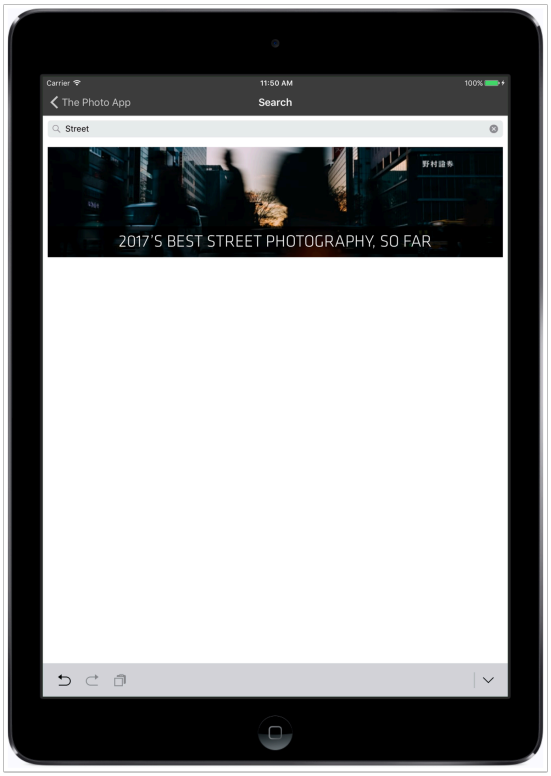
4. **Thumbnail**
   You can upload a different thumbnail by clicking on the thumbnail and selecting a thumbnail from your files.
5. **Title**
   Change the title of your content item by simply typing anything in this field
6. **Publish date**
   By default, the publish date field will be empty. In that case, the content item will remain unpublished. If one adds a publish date in the past, the content item will be published immediately and that date will count for subscription periods. When a date in the future is selected, the content item will show the scheduled status in the content item list until the publish date equals the actual date. At that moment, the content item will be published automatically.
7. **Item style**
   By default, the default item style will be selected but one can change the item style by clicking the field and selecting the desired item style.

⚠ All the content items will only be added to the Hamburger Menu by confirming via 'save'.

More features

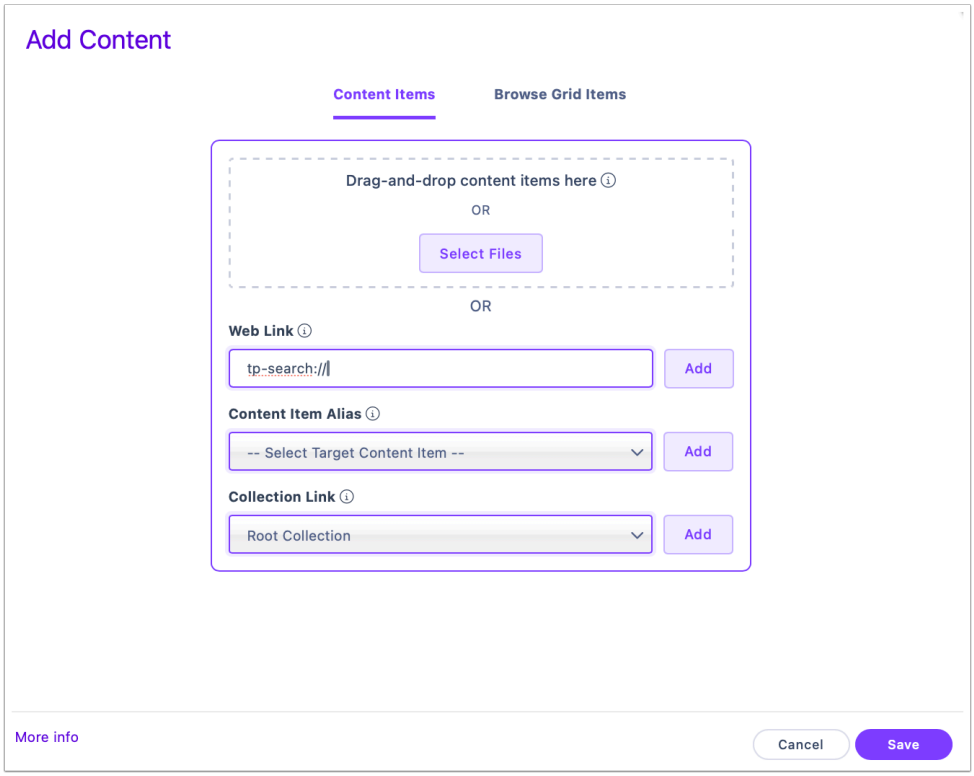
Searching in your app

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.

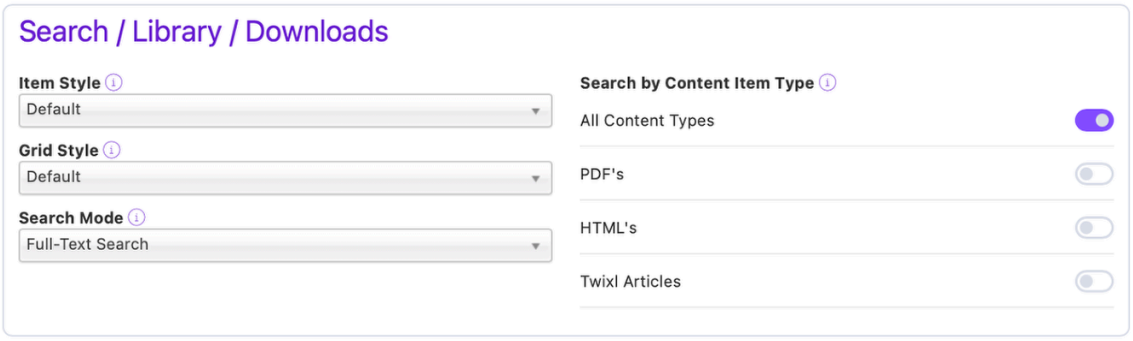


2. Determine the look & feel of the search results

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.



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.

In '**Search Mode**', one 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.

Search by...

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

"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.

Manage content downloading

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

- **On the fly:** A content item is only downloaded and opened as the user swipes to the item.
- **Collection on demand:** A collection can be viewed on the fly or monolithic by long-pressing the thumbnail of a collection.
- **Monolithic collection:** A collection is downloaded entirely before displaying, ideal for offline reading of a magazine when travelling.
- **Keep all data offline (Full offline modus):** App is downloaded entirely before displaying, ideal for offline app usage (e.g. a sales app).

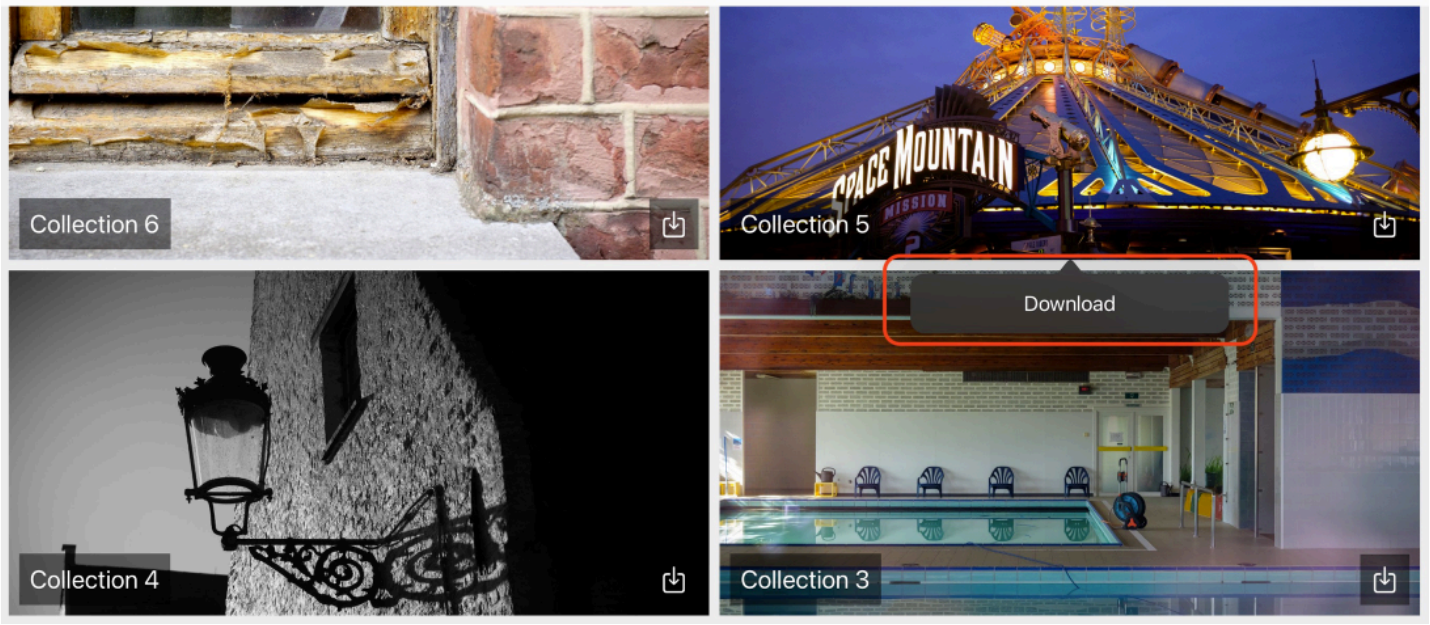
While the first 2 options are triggered by the user, the last 2 are managed and enabled by the App Publisher. Downloaded items are never deleted automatically by the system of the device.

1. Collection on demand

When a user wants to be able to read a collection offline (e.g. reading a magazine when travelling), they can long-press the thumbnail of that collection and confirm to download. The only difference with a monolithic collection is that the user is in full control to download a collection or not.

By long-pressing the collection again, the user can permanently delete the collection from the device.

As a publisher, you can help the user to discover this feature by adding a download icon (see further) and by adding a tp-download url scheme to your app.

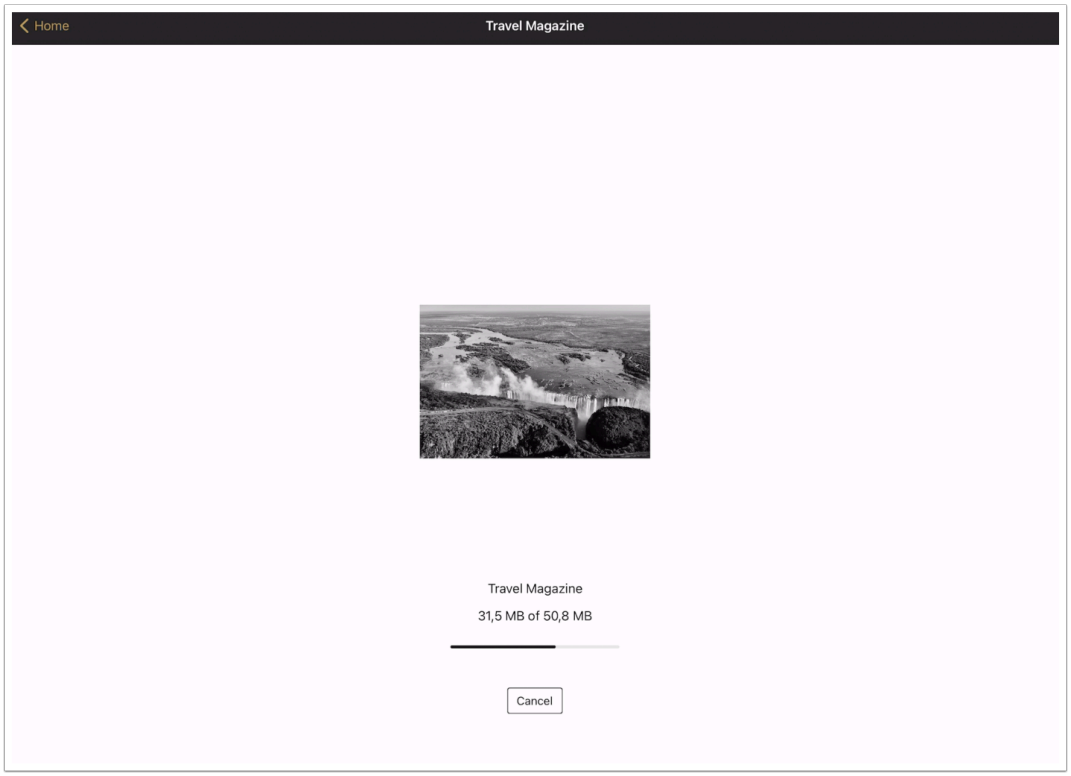


2. Monolithic Download

When you enable the monolithic option in a collection, the user is forced to download a collection before the browse view or first content item will be shown. This option ensures quick and smooth flows between 2 content items as these are on the device already.

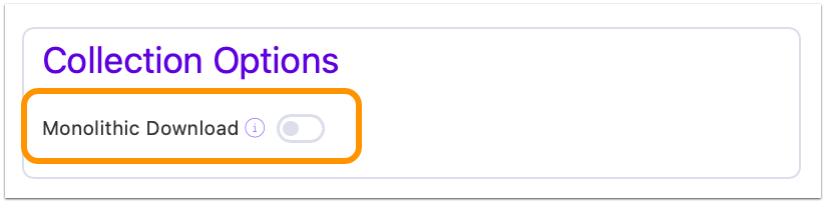
As this is not a general project setting, you can decide for each collection individually if you want to offer it as an on-the-fly or as a monolithic collection.

By long-pressing the collection, the user will permanently delete the collection from the device. A next visit to that collection will force the download again.



2.1. How to activate this feature?

Via Collection > Your collection > Collection Settings > Collection Options > Enable 'Monolithic Download'



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.

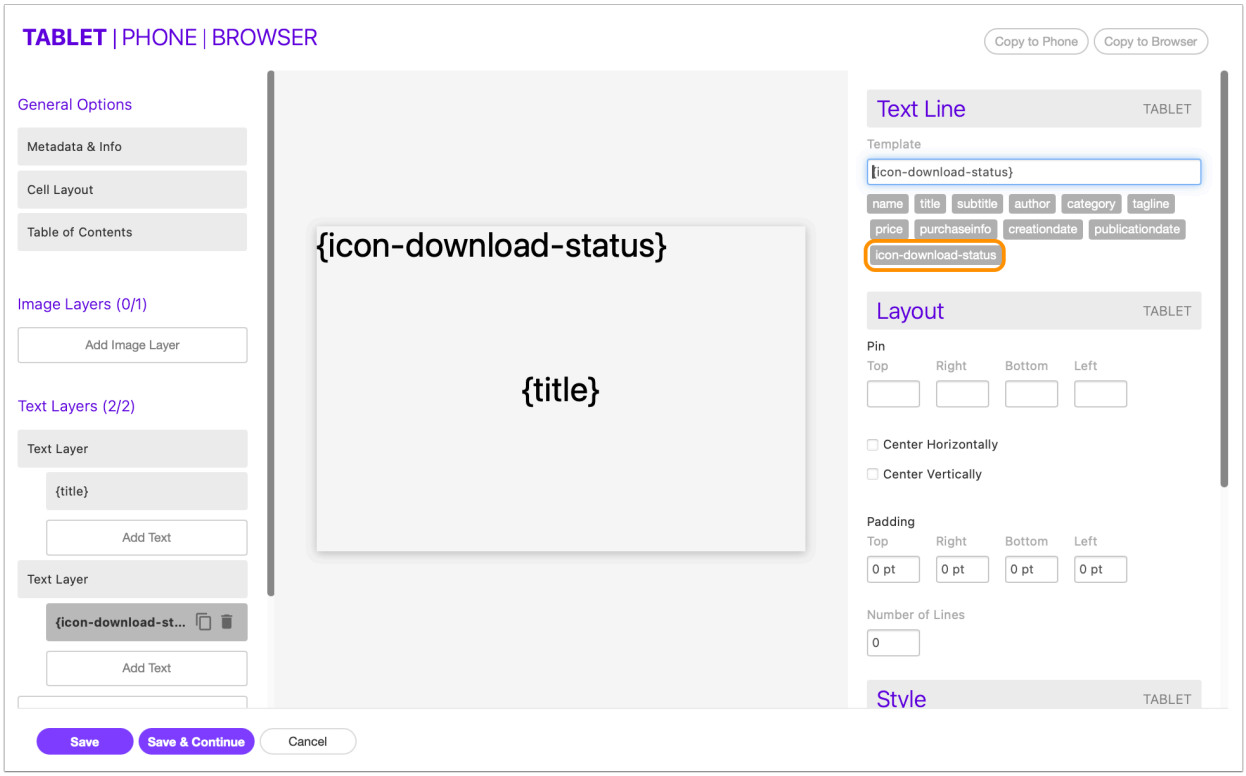
2.2. How can the reader use this?

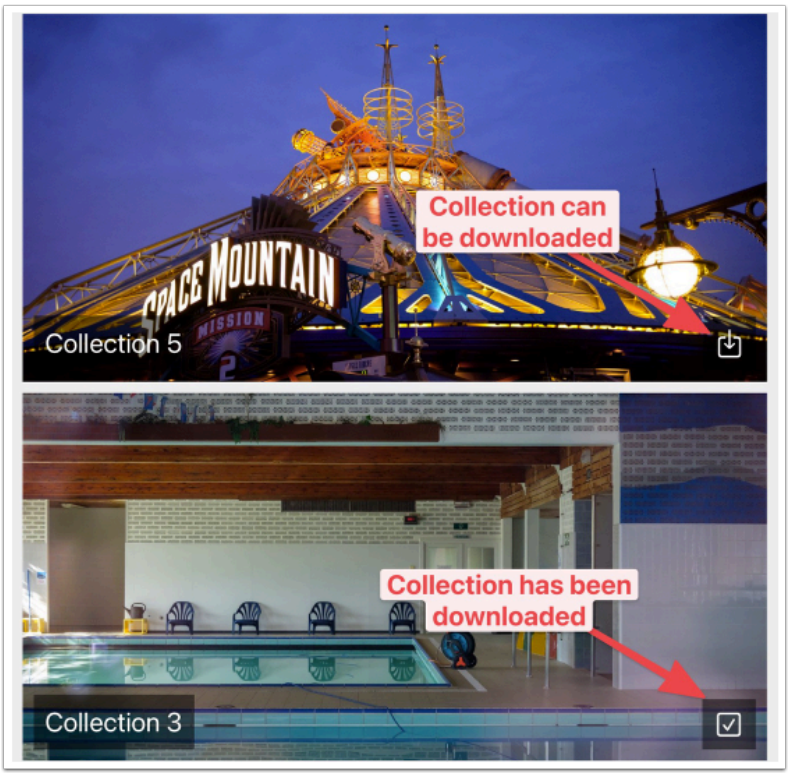
1. For collections that have the option 'Monolithic download' enabled, the process will be fully transparent, and a progress bar will be displayed during the download.

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.

2.3. 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:

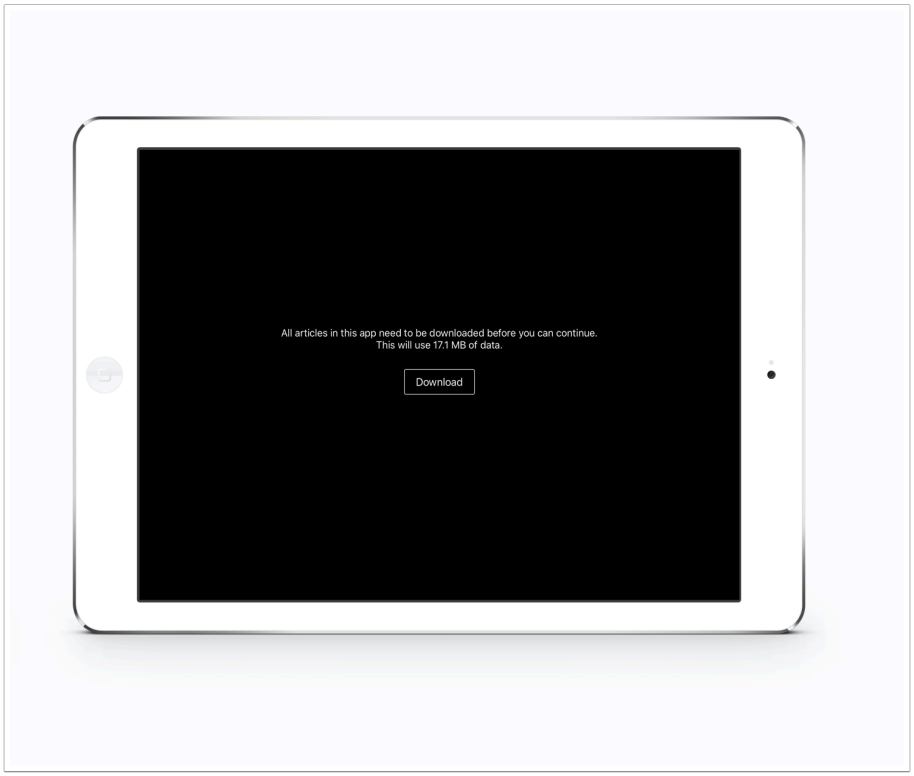
When downloading a (monolithic) collection, the content items that are downloaded will only go one level deep. This means that, if the collection contains e.g. collection links (to other collections), these will not be downloaded.

3. Full Offline Mode

We provide a configuration 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.

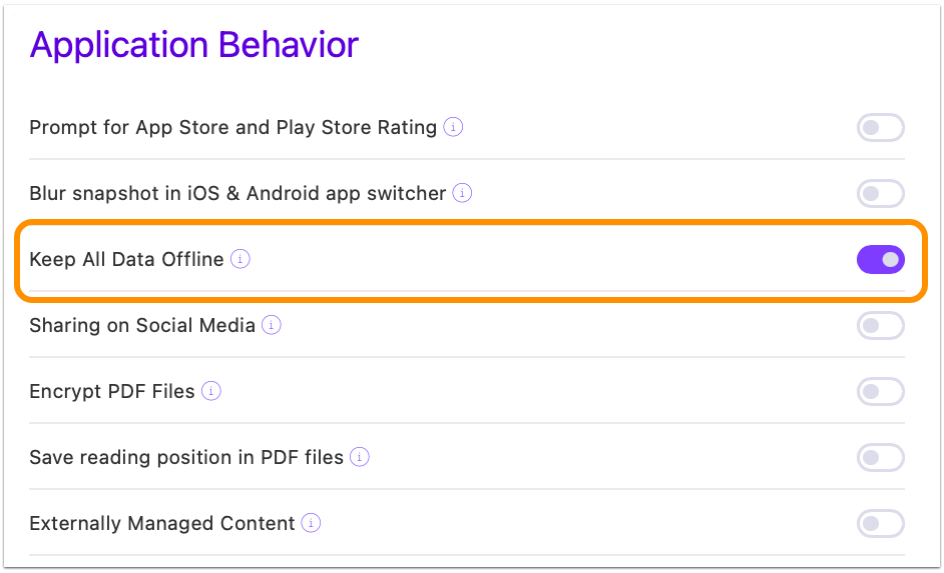
WARNING:

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?

Go to Menu > Design > Configuration > Application Behavior and enable the 'Keep All Data Offline' feature.



We advise to thoroughly test the navigation and functionality of your app first before enabling this feature as you constantly need checking during the project designing phase.

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 they are online, the app will check whether there is new and/or updated content that needs to be downloaded or old content that needs to be deleted. 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.

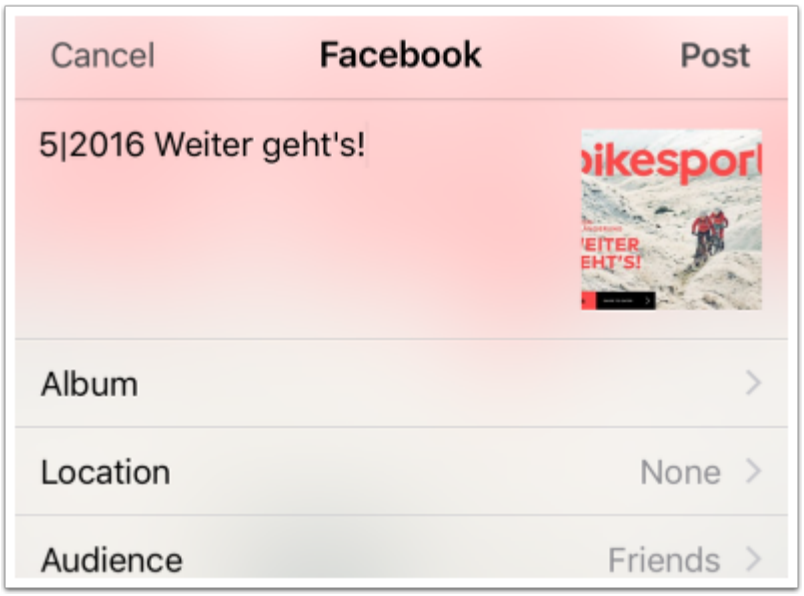
When the user is offline, the app will not check for updates and the user can just start exploring the app.

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.



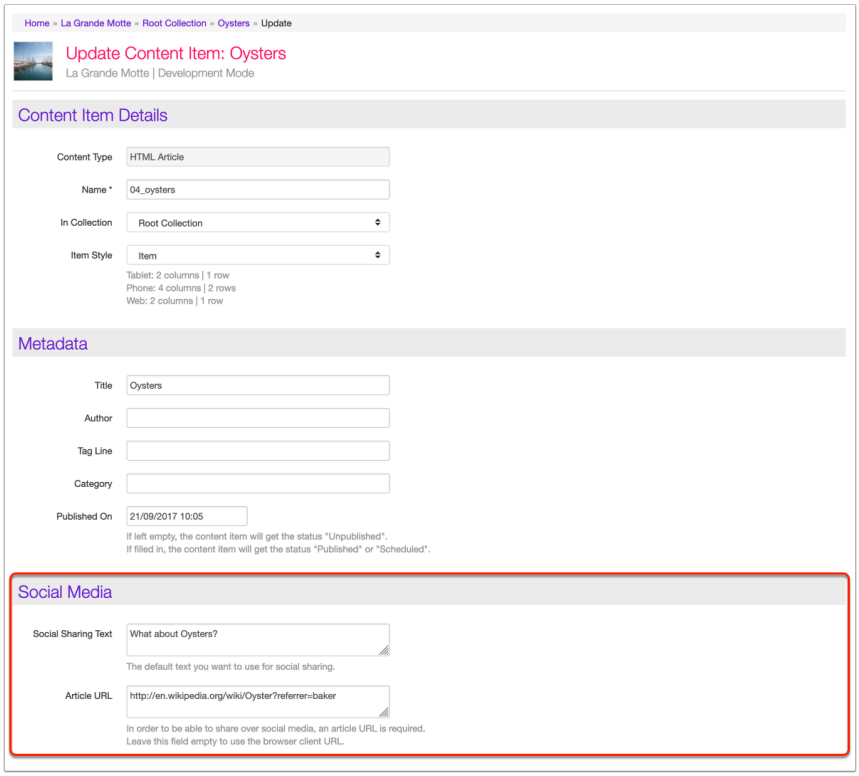
How to configure for your app

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.



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.

Using custom URL Schemes in your app

Custom URL schemes are a powerful way to control navigation from just about anywhere in your app! This article provides a complete overview of what you can do with the different URL schemes.

1. Navigating to a collection or article

From within your article content, you can use special URL schemes to link to a collection or a particular article within a collection. These can also be used in InDesign content using a Hyperlink, a Web viewer or a Web Overlay Button.

You can use the following scheme:

```
tp-collection://[target_collection_name]/[target_article_name]
```

Click to copy

The syntax is as follows (always make sure to refer to the collection or article `name`, not the `title`):

`tp-collection://[collection-name]`	Will link to the first article in a different collection.
`tp-collection://[collection-name]/[article-name]`	Will link to a specific article in a different collection.
`tp-collection://hamburger-menu`	Will open the Hamburger Menu (provided the Hamburger Menu is enabled).
`tp-collection://parent`	Will link to the parent collection of your current article (not supported in the Browser Client).
`tp-collection://root`	Will link to the root collection of your app.

2. Navigating to an article or page

The links below can also be used in your InDesign content, in a hyperlink, a web viewer or a web overlay. It can be used to go to another article in the same collection.

```
tp-pagelink://[article_name]
```

Click to copy

If you want to add a link to a particular page in an article in the same collection from within a browse page or article, you can do so creating an `HREF` like the one below:

```
tp-pagelink://[article_name]/[page_number]
```

Click to copy

An HTML example would be:

```
<a href="tp-pagelink://TOC/3">
```

Click to copy

For InDesign content the article name needs to be the name of the InDesign file. So, in case of the example above, the name of the InDesign document would be TOC.indd.

A number of other URL schemes for relative article and page navigation are also available:

`tp-next-article://`	Go to the next article.
`tp-previous-article://`	Go to the previous article.
`tp-first-article://`	Go to the first article of the publication.
`tp-last-article://`	Go to the last article of the publication.
`tp-next-page://`	Go to the next page.
`tp-previous-page://`	Go to the previous page.

`tp-first-page://`	Go to the first page of an article.
`tp-last-page://`	Go to the last page of an article.
`tp-article-top://`	Go to the top of a long-page article.
`tp-article-bottom://`	Go to the end of a long-page article.

3. Show/hide the toolbar (for InDesign content)

(Available in an InDesign Hyperlink, a Web viewer or a Web Overlay Button)

`tp-toolbar://hide`	Hides the toolbar with the Table of Contents icon, and optional sharing and bookmarking icons.
`tp-toolbar://show`	Displays the toolbar with the Table of Contents icon, and optional sharing and bookmarking icons.
`tp-toolbar://toggle`	Toggles the current view, whether visible or invisible.

4. Show Table of Contents (for InDesign content)

(Available in an InDesign Hyperlink, a Web viewer or a Web Overlay Button)

```
tp-toc://
```

Click to copy
When the `tp-toc://` url is triggered, it will show the Table of Contents dropdown.

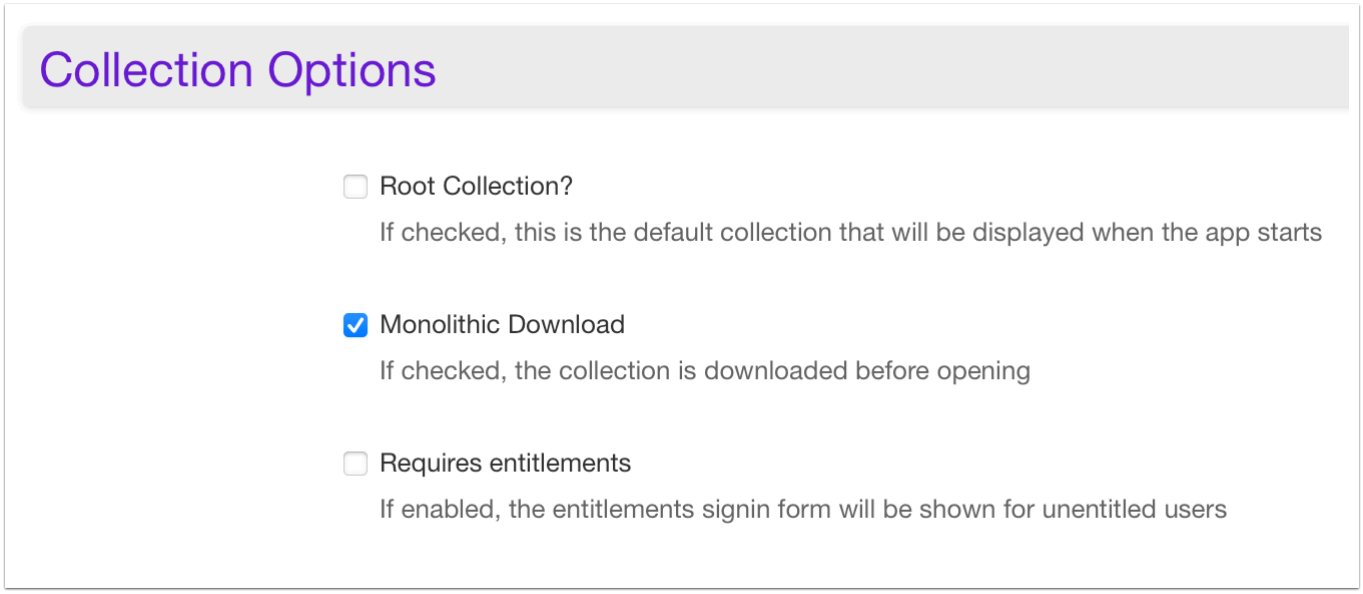
5. Downloads overview

(Can be used in e.g. an InDesign hyperlink, a Web Link content item, or an HTML article.)

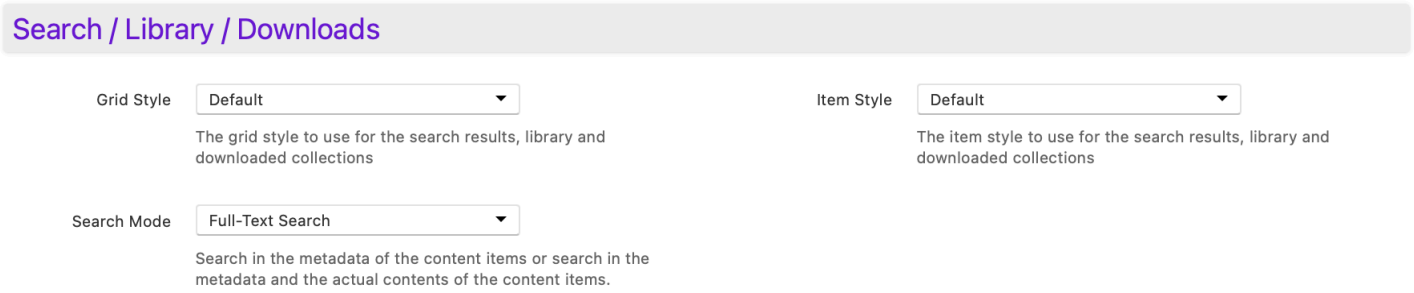
```
tp-downloads://
```

Click to copy
If you have an app where (some or all) collections are marked as **Monolithic Download**, this URL scheme will trigger the display of an overview of the collections that have been downloaded.

5.1. Watch a short 'How to' video...



- The grid & item styles that are used for this collection are defined in your app settings under 'Search / Downloads' (the same styles are used for both the Downloads collection and the Search results).
- Sorting is done by most recent publish date.



Users can also manage the content that has been downloaded: long-pressing the collection icon of the collection brings up a dialog that allows the user to delete the downloaded content. It can always be re-downloaded later.

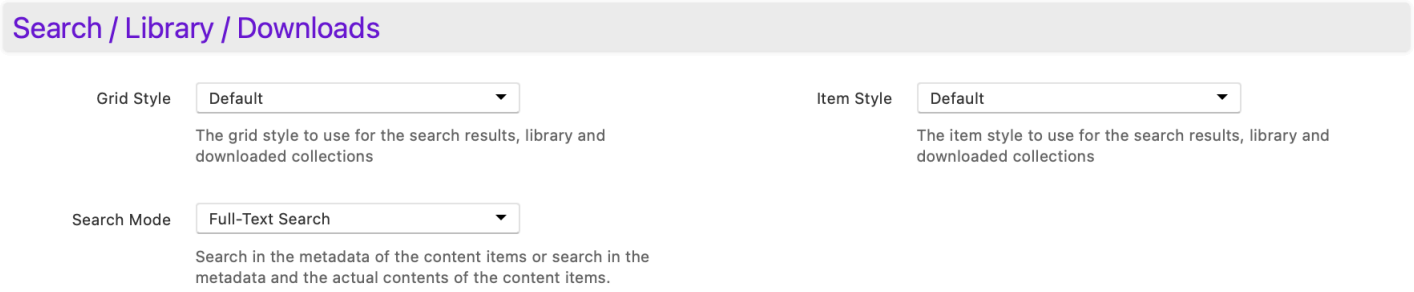
6. Library overview

The Library feature allows the publisher to include a section where all the free collections for an app user are displayed, i.e. the free, the purchased and the entitled collections. This feature can be used in e.g. an InDesign hyperlink, a Weblink content item or an HTML article.

```
tp-library://
```

Click to copy

- The grid & item styles that are used for this collection are defined in your app settings under 'Search / Downloads' (the same styles are used for both the Downloads collection and the Search results).
- Sorting is done by most recent publish date.



The Library-feature differs from the Downloads feature as tp-library doesn't show all the offline available (=downloaded) collections. It only shows the for a user freely available collections and this difference should be considered before implementing this feature.

7. Searching

(Can be used in e.g. an InDesign hyperlink, a Web Link content item, or an HTML article.)

To trigger the search-dialog in your app, you use the following url scheme:

Custom URL Scheme	Function
`tp-search://`	Shows an empty dialog
`tp-search://[keyword]`	Executes a search-command with the specified keyword

Example (search for content with the word `twixl`):

```
tp-search://twixl
```

Click to copy

- The grid & item styles that are used for the search result are defined in your app settings under 'Search / Downloads' (the same styles are used for both the Downloads collection and the Search results - see above).

8. Paywall & subscriptions

```
tp-paywall://
```

Click to copy
Enables you to trigger the paywall, showing both purchases and different subscriptions that you have defined.

```
tp-subscriptions://
```

Click to copy
Lets you trigger the paywall, showing only the different subscriptions you have defined.

```
tp-restore-purchases://
```

Click to copy
Lets you trigger the 'Restore purchases' functionality.

9. Entitlements

Show the Entitlements sign-in form

```
tp-entitlements-signin://
```

Click to copy
Triggers the entitlements sign-in form.

Show the Entitlements register form

```
tp-entitlements-register://
```

Click to copy
Triggers the entitlements register form. This only works if the entitlements server provides a "register" action. Available in the kiosk info cell - infoCell.html

Get entitlement token

```
tp-entitlements-get-token://<callback_function>
```

Click to copy
Get the current entitlement token, passing it as the argument to the `callback_function`. If no callback function is specified, it will default to `twixlKioskOnGetEntitlementToken`.

Clear entitlement token

```
tp-entitlements-clear-token://
```

Click to copy
Clears the entitlements token (can be used for testing purposes).

10. Sending e-mail

Please refer to the article Advanced Mailto hyperlinks to learn how to use the `mailto:` URL scheme, to send e-mail from your app.

11. Making a phone call

```
callto:[number] or tel:[number]
```

Click to copy
Allows you to trigger a phone call.

Example:

```
callto:+32493252577

tel:+32493252577
```

Click to copy

12. Sharing on social media

To trigger the sharing sheet, you use the following url-scheme:

```
tp-share://
```

Click to copy

13. Going back in Browsing History

This special url scheme can be used in to go back in the browsing history in an app. The syntax is as follows:

```
tp-history-back://
```

Click to copy
A history will be saved when going though the different content items. This means that if you are in **Article1** and you scroll to **Article2**, if you then press the custom URL tp-history-back:// it will go back to **Article1**.

The history will be saved when:

- Scrolling horizontally to other content items.
- Scrolling horizontally/vertically to pages of the same content item.
- Going to another content item or collection with a custom url (tp-pagelink://article2, tp-next-article, tp-collection://collection2...)
- Going to different pages in the same content item with a custom url (tp-next-page://, tp-last-page://, tp-first-page://...)

14. Going back in Navigation History

This special url scheme can be used in all the same places as `tp-collection` links and goes back in the navigation stack for a specific number of steps. The syntax is as follows:

```
tp-navigate-up://[number-of-steps]
```

Click to copy
The `number-of-steps` indicates how many steps you want to go back.

Example:

Application *XYZ* has the following structure:

- **Root Collection** (browse mode)
  - **Languages Collection** (browse mode)
    - **English Collection** (browse mode)
      - **English Document 1** (detail mode)

Some specific uses and what the result will be (in this case):

- English document 1: `tp-navigate-up://1` takes you to the **English Collection**.
- English document 1: `tp-navigate-up://2` takes you to the **Languages Collection**.
- English document 1: `tp-navigate-up://3` takes you to the **Root Collection**.
- English document 1: `tp-navigate-up://200` also takes you to the **Root Collection**.

IMPORTANT NOTE:

`tp-navigate-up://` is supported only on iOS and Android. It cannot be used in the browser client.

15. Launching an app

It is possible to trigger launching an app from within a Twixl app using the following scheme:

Please note that the implementation is slightly different on iOS vs Android.

- **iOS:** the URL scheme is the same as the application identifier:

  e.g. if the app identifier is `com.twixlmedia.myapp` -> then the URL to launch the app would be: `com.twixlmedia.myapp://`

- **Android:** the URL scheme is the same as the app identifier without the dots, dashes, underscores and all lowercase:

  e.g. if the app identifier is `com.twixlmedia.myapp` -> the URL to launch the app would be: `comtwixlmediamyapp://`

IMPORTANT NOTE:

This will only work if the app is already installed on the device.

16. Registering a test device for push notifications

`tp-register-test-device://`

Triggering this custom URL scheme will add the current device to the list of test devices for push notifications. Note that once push notifications for your app has been configured, you'll need to create a new build of your app before you can use this URL scheme. More details here.

17. Launching third-party apps

In this scenario, it depends on the url schemes supported by the app you want to launch. E.g. an app which can be opened using the url scheme `com.mycustomapp://` can be launched by simply creating a hyperlink as follows:

```
<a href="com.mycustomapp://">Launch My Custom App</a>
```

Click to copy

IMPORTANT NOTE:

This will only work if the app has already been installed on the device.

18. Opening a hyperlink in the device browser

```
tp-open-in-device-browser=1
```

Click to copy

Although in most cases, you want to keep readers in the embedded browser, sometimes you may want to open a link directly in the default browser on the device (e.g. Safari, Chrome, Brave...). A good use case for this is if you want to link to a PDF that you want readers to be able to download.

You can do this by adding this extra parameter to your URL. Available for Hyperlinks.

Example:

The link you want to open is e.g.:

```
http://www.website.com/file2.pdf
```

Click to copy

While just adding this link will open the URL in the embedded app browser, you can also make it open in the default browser on the device by adding the custom URL link. The link becomes:

```
http://www.website.com/file2.pdf?tp-open-in-device-browser=1
```

Click to copy

When your original link already has a query parameter, you need to change the `?` into a `&` when adding the custom URL link.

Example for a link with a query parameter:

E.g. the link with a query parameter looks like this:

```
http://www.website.com/file.pdf?id=1
```

Click to copy

To open this link in the default device browser, the custom URL link will be added like this:

```
http://www.website.com/file.pdf?id=1&tp-open-in-device-browser=1
```

Click to copy

19. Open in embedded web browser

```
tp-open-in-web-browser=1
```

Click to copy

Adding this url parameter to a hyperlink will open the link in the embedded web browser instead of opening inline.

20. Get the device info via JavaScript

Adding the following JavaScript code to your Content Item, Web Viewer, Web Overlay Button, ...

```
window.location.href = "tp-device-info://";
```

Click to copy

will execute the following JavaScript function:

```
function twixlOnGetDeviceInfo(deviceUDID, appVersion, appIdentifier, entitlementToken) {
}
```

Click to copy

This then gives you access to the Device UDID, the app version, the app identifier and the entitlement token. You can also customize the name of the function that gets called:

```
window.location.href = "tp-device-info://myCustomFunctionName";
function myCustomFunctionName(deviceUDID, appVersion, appIdentifier, entitlementToken) {
}
```

Click to copy

SAMPLE FILES:

For more info, see the sample files on this GitHub-page.

21. Custom Variables

Custom variables allow you to set or get one or more variables with the following properties:

- They are saved on the device
- They can be re-used in advanced scripting

A sample scenario is e.g. to store a user preference and change the content based on that variable.

```
tp-set-custom-vars://
```

Click to copy

```
tp-get-custom-vars://
```

Click to copy
Example:

```
tp-set-custom-vars://?name=&product=
```

Click to copy

MORE INFO ABOUT CUSTOM VARS:

Custom Variables rely on Advanced Scripting. For more info about Advanced Scripting and an example, see:

- Advanced Scripting: Introduction
- Advanced Scripting | Sample App 6: Custom Vars

Advanced Scripting: Introduction

With Advanced Scripting you can decide to filter Content Items or Collections based on selected criteria.

1. Advanced Scripting, what is it?

Using Advanced Scripting, you can link a number of properties available in the app (bundle ID, language, geolocation, ...) and decide whether the Content Item or Collection should be visible or not.

ENTITLEMENT PACK REQUIRED:

To make use of the Advanced Scripting functionality, you need to have the Entitlement Pack activated. Contact the sales department for details.

2. General info about Advanced Scripting

Advanced scripting is a collection of JavaScript functions that will be evaluated for each collection and/or content item in the context of an application.

It can be used to filter the content items shown in a collection based on a custom set of business rules.

Where can I activate Advanced Scripting?

Although Advanced Scripting is all about filtering Content Items and Collections, you define Advanced Scripting on a Collection level. As such you need to edit a Collection in order to add a script.

Her's a sample script:

```
// Executed once for every collection before the filtering of the items
function setupFilter(collection, environment) {
}

// Executed for once for every collection
function shouldShowCollection(collection, environment) {
```

```
    return true;
}

// Executed for every single content item
function shouldShowItem(contentItem, collection, environment) {
    return true;
}

// Executed once for every collection after the filtering of the items
function teardownFilter(collection, environment) {
}
```

Click to copy

IMPORTANT:

Each of the 4 base functions that get executed in the advanced filter currently have a maximum execution time of 2 seconds.

TIP:

You can find a complete syntax and lifecycle overview here. This info also can be found on the Twixl platform, below the Advanced Scripting editor window.

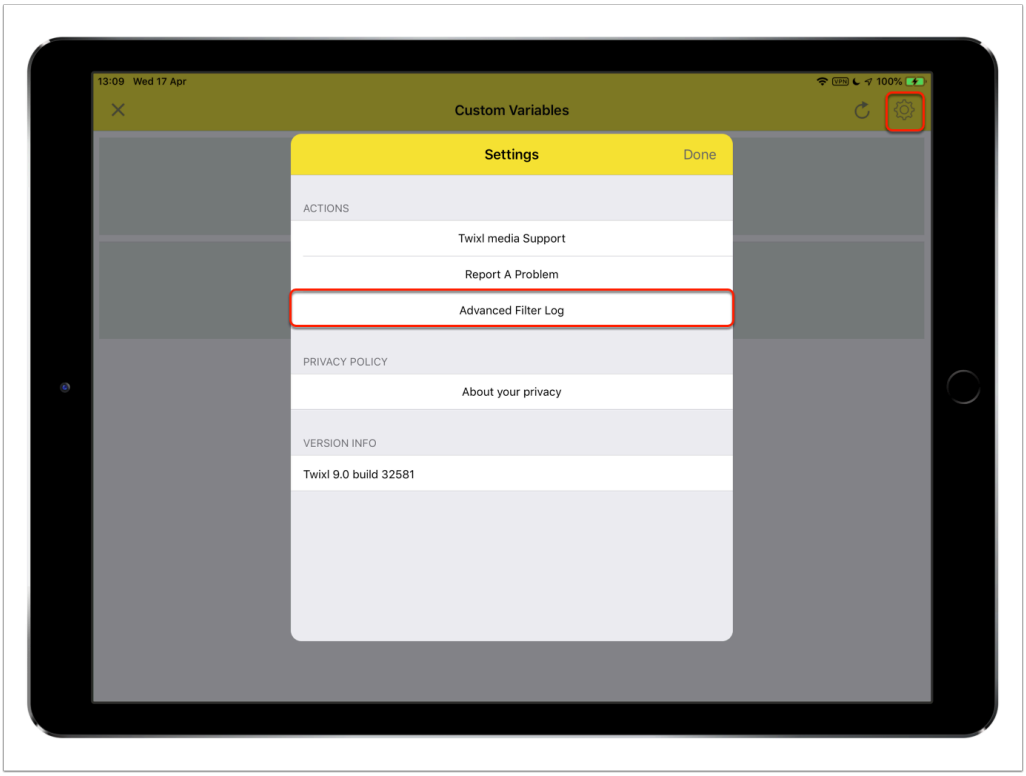
Debugging options

In a mobile app

2.1. App without a Hamburger Menu

For apps without a *Hamburger Menu*:

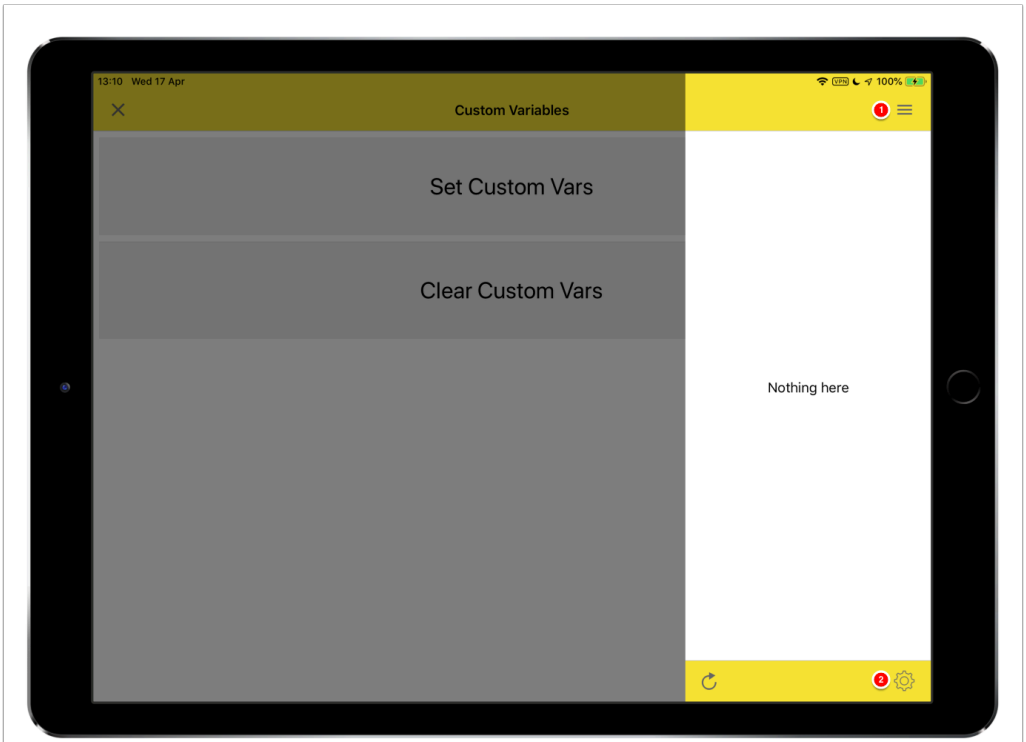
1. Tap the *Gear Menu* first
2. Then select *Advanced Filter Log*

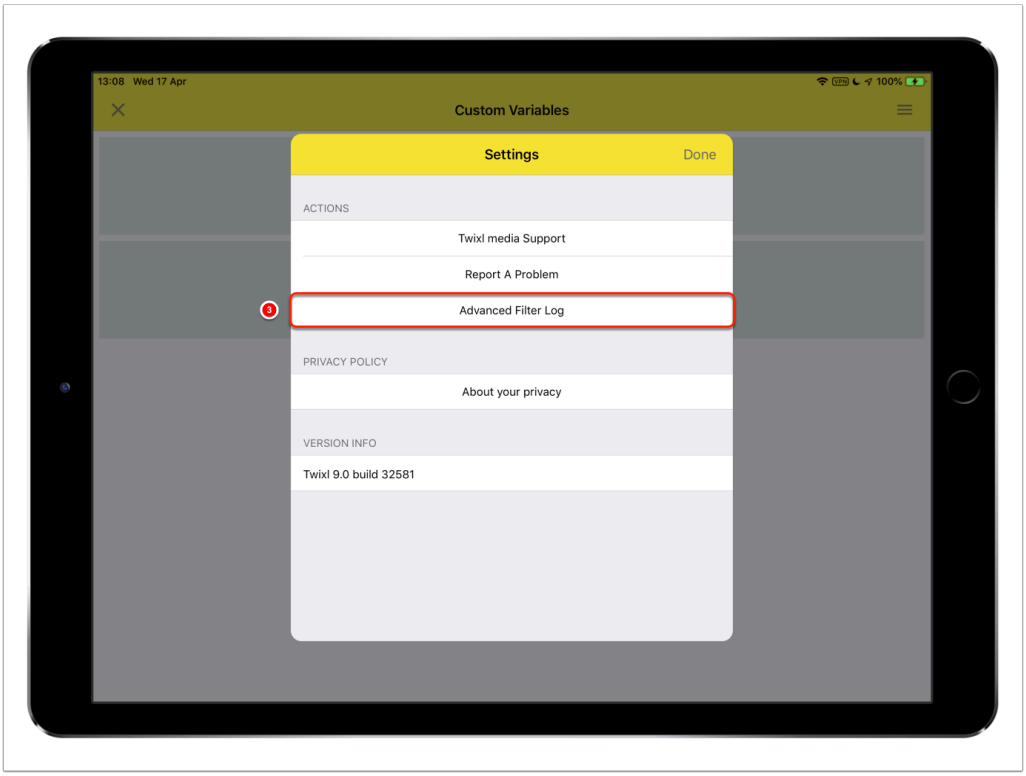


2.2. App with a Hamburger Menu

For apps with a *Hamburger Menu*:

1. Activate the Hamburger Menu first
2. Select the *Gear Menu*
3. Then select *Advanced Filter Log*





In the Browser Client

For The Browser Client, you need to add the suffix ?debug=1 and then you'll notice the debugger icon in the toolbar.

So this means, if your Browser Client url is:

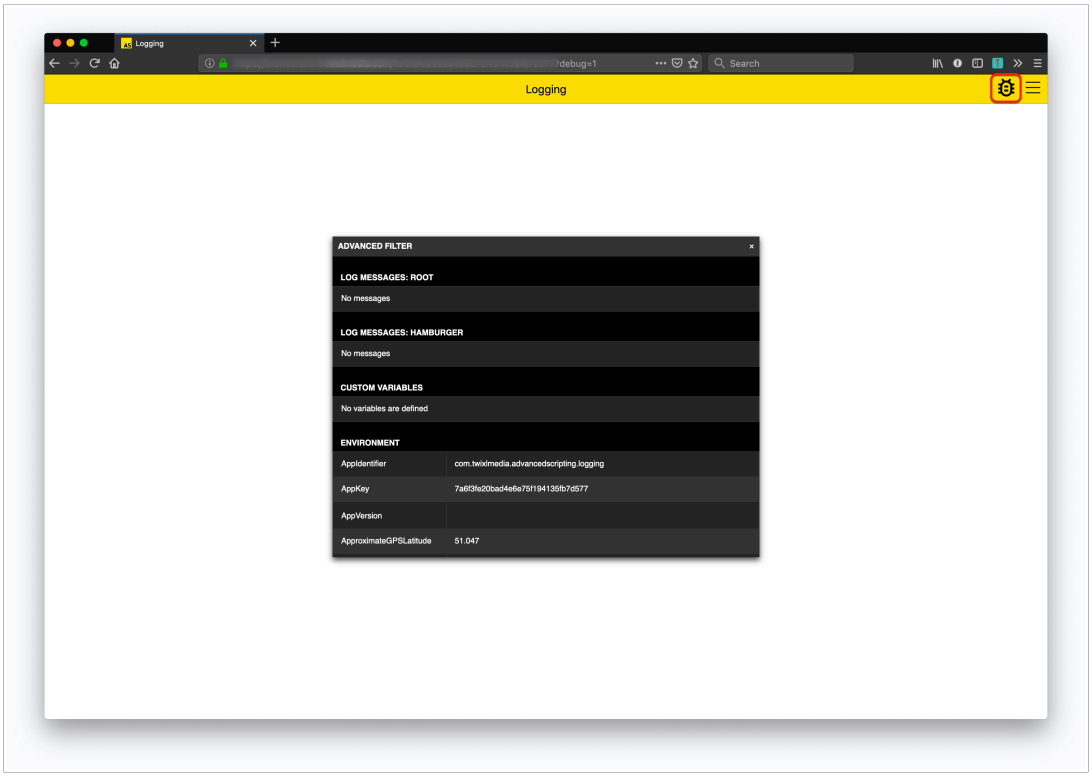
```
https://browserclient.twixlmedia.com/<app>
```

Click to copy

Then you'll need to add ?debug=1 in the following way:

```
https://browserclient.twixlmedia.com/<app>?debug=1
```

Click to copy



3. Advanced Scripting Sample Apps

INFO:

We've created several Advanced Scripting Sample Apps for you, each with a specific case that could be useful. They are a good start to explore the immense list of possible scenarios for Advanced Scripting.

- Advanced Scripting | Sample App 1: Login / Logout Button
- Advanced Scripting | Sample App 2: Different content for phone vs tablet vs Browser Client
- Advanced Scripting | Sample App 3: Using twxlog
- Advanced Scripting | Sample App 4: Using JSON and XML
- Advanced Scripting | Sample App 5: Using twxhttp

- Advanced Scripting | Sample App 6: Custom Vars
- Advanced Scripting | Sample App 7: Multiple Languages
- Advanced Scripting | Sample App 8: Privacy Policy

Best practices

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 article.

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:

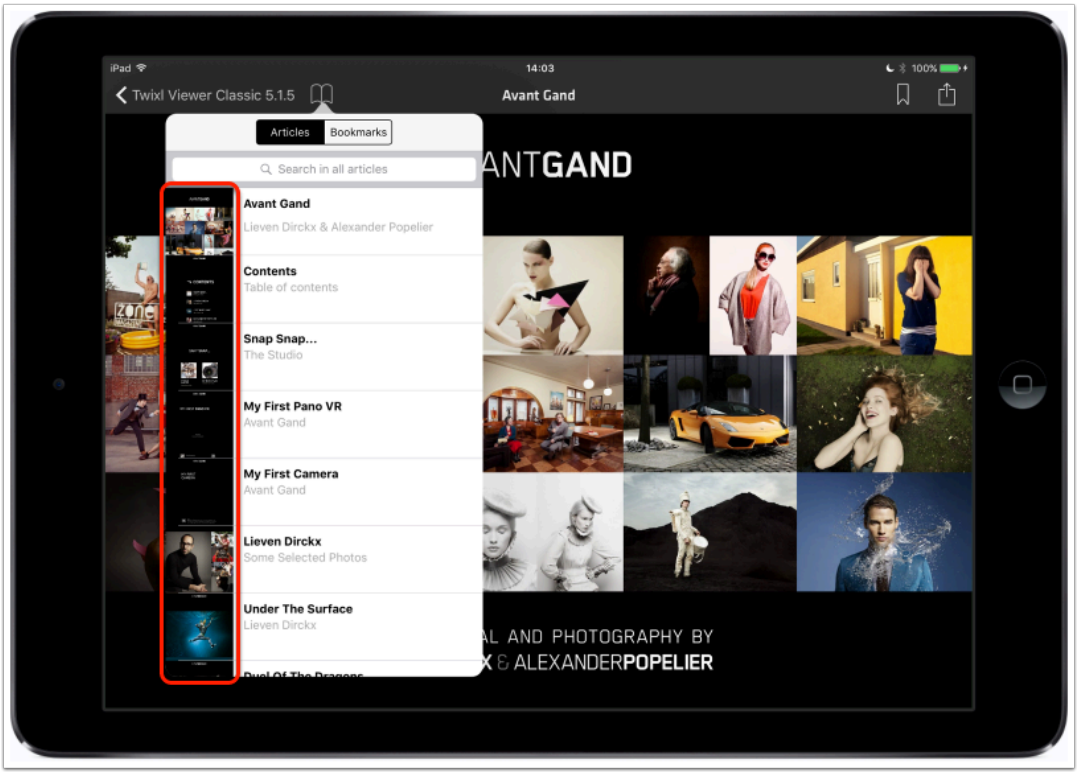
How your PDFs have been generated is not important by itself, but there are a few things to take into account when creating these PDF files:

- 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 RGB and not CMYK.
- 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.

Using custom thumbnails in the Table of Contents viewer

When using web viewers, panorama VR or image sequences on a page, these will not be rendered when the Twixl Publisher export creates the thumbnails that are used for the Table of Contents (TOC) overview.

There is a way to customize the thumbnails that are being displayed though...

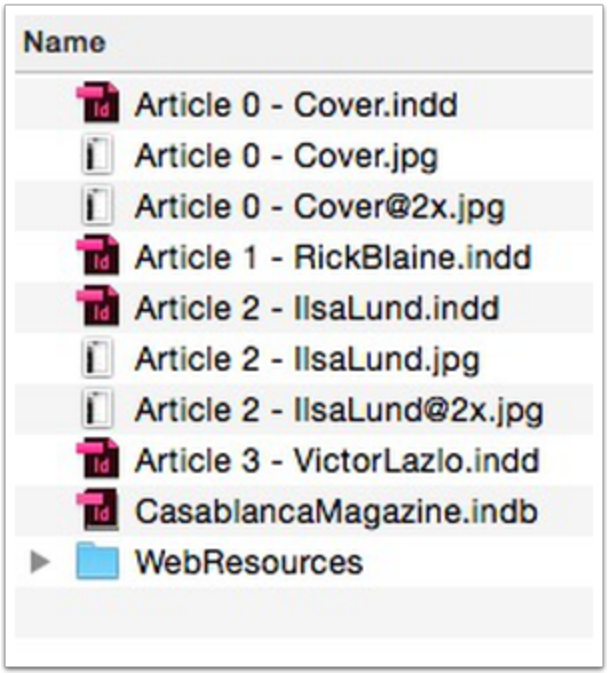


For InDesign-based content

1. The thumbnails that you want to customize need to be placed in the same folder as the InDesign files for the article. If you use a publication-based workflow, it is best practice to keep both the different articles and the publication (InDesign book) file within the same folder.
2. The thumbnail files need to have the same name as the article, but with .jpg as the extension instead of .indd, and have a size of **1024x1024** pixels (or a larger, preferably square, size).
3. The retina thumbnails should add a suffix `@2x` to the file name, e.g. `ArticleName@2x.jpg`, and have a size of **2048x2048** pixels (or a larger, preferably square, size).
4. For articles where no custom thumbnail has been defined, the Twixl Publisher plug-in will generate one on the fly during the Export process.

A sample folder structure should then look as follows:

- `Article1.indd`
- `Article1.jpg` (1024 × 1024 pixels)
- `Article1@2x.jpg` (2048 × 2048 pixels)
- `Article2.indd`
- `Article2.jpg` (1024 × 1024 pixels)
- `Article2@2x.jpg` (2048 × 2048 pixels)
- `MyPublication.indb`



For HTML-articles

The Twixl Distribution Platform will generate thumbnails. Here are the rules you need to take in account:

- If there is an `index.jpg` file in the `.zip` file on the same level as the `index.html` file, that image will be used as a thumbnail
- If there is an `index.jpeg` file in the `.zip` file on the same level as the `index.html` file, that image will be used as a thumbnail
- If there is an `index.png` file in the `.zip` file on the same level as the `index.html` file, that image will be used as a thumbnail
- If no images are present, the Twixl Distribution Platform will look at the first image it can find in the HTML-file.