# Creating Re-usable Content

In today’s online ecosystem, the content you create isn’t always just for your website. You may have content that appears on both on your website and your iOS/Android app or you may want to use a list of content from an Agility instance on some other external website altogether.

The way in which you architect and manage your content can affect how this works.

Creating your content with this in mind can set you up for success when you are ready to syndicate your content to channels other than your website.

## Sharing Content

Everything from Pages to Content in Agility is accessible via the Agility Content Fetch REST API. This enables you to retrieve Agility CMS content using any programming language. You can even pull content from multiple instances.

Common Uses:

* Mobile app (iOS, Android)
* Digital signage
* IoT devices
* Kiosks
* Other websites (any programing language)

Why create duplicate content when you can create, manage, and publish content from one central location?

## Content Architecture - Pages vs Shared Content

There are two ways to query your CMS data in Agility using the Content Fetch API. You can query *Pages* or *Shared Content*.

*Pages* provide a way to manage the presentation layer for a website. They are perfect for websites that are powered by Agility and have built-in page routing. They have lots of metadata that is specific to pages such as template, content zones, modules, title, URL, SEO information, tracking scripts and more.

*Shared Content* provides an interface for developers to query entire lists or retrieve specific items such as Articles, Blog Posts, Category Lists, etc. In other words, just the data.

The best way to share content with other sources is to store your content in Agility’s *Shared Content* area. Each content item/list is easily queryable by the Content Fetch API and it provides only the content you asked for. It makes no assumptions on how the content will be used or displayed.

## Keep your Content Clean

Because your content could be used on platforms other than a website, you can’t rely on the device being able to understand or render HTML/CSS/JS. It’s important to keep your content clean and maintain a clear separation of concerns between the content and how it is presented.

While iOS and Android do support some basic HTML elements, it is recommended to limit Rich Text Area editor usage and features for managing content that will be shared across other non-web platforms.

Some ways to limit HTML/CSS/JS usage:

1. Use *Long Text* fields as opposed to *HTML* fields in your Content Definitions. You could allow usage of Markdown for formatting in absence of HTML.
2. Implement a custom validation of your *HTML* fields to ensure only approved elements are being used.
3. Implement manual *Approval Workflows* to ensure the content being created is safe.
4. Create a *Custom HTML* field using *Custom Field Types* that only allow specific HTML elements.
5. If you need to implement *tables*, *videos*, *image galleries*, or other UI elements, consider using specific fields in the CMS for this as opposed to embedding in the Rich Text Area.

## Guiding Principals

1. Use the Agility Content Fetch REST API.
2. Store content in *Shared Content*.
3. Maintain a separation between your content and your presentation layer by limiting HTML/CSS/JS.