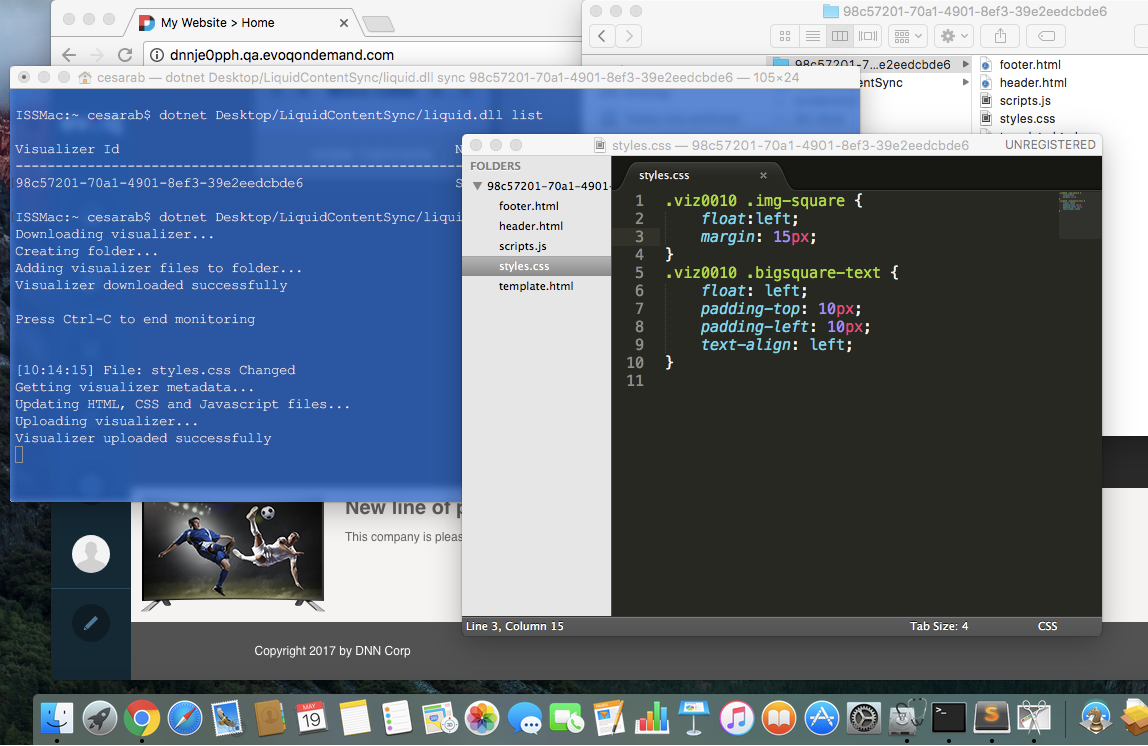
Liquid Content Synchronizer



# Introduction

Liquid Content has a rich set of APIs that allows you to integrate your applications with your site’s content.

In this tutorial, we’re going to develop a simple console application to manage the visualizers provided in Liquid Content. The app will let you get the files used by the visualizers, so that you can modify them in your desktop and re-upload them automatically to your site once you finish your changes.

To make it more interesting, the app will work on Windows, Mac OS and Linux, since we’re going to build it in [.NET Core](https://www.microsoft.com/net/core/platform).

# Motivation

I love working with my favorite web editor. Liquid Content has a nice web editor, but, when I’m editing the look and feel of my visualizers, I always end up copying the CSS, HTML and JavaScript in my web editor and then I tweak those files locally, in my laptop. I do this especially when those files are made up of several of lines of code.

Wouldn’t be great if I would be able to modify those files in my favorite web editor and keep them synchronized automatically in my site every time I save the changes?

It turns out that Liquid Content has an API that allows you to get and update those files very easily so, why don’t we take advantage of it and create a synchronizer which can download the visualizer’s files and keep them updated in your site, while you’re changing them?

And to make this more fun, why don’t we do it in a platform like .NET Core, which allows you to run the app, not only on Windows, but also on Mac OS and Linux?

# How to use it

This is a very simple console application, which allows you to download the files related to a specific visualizer and also lets you upload those files back to your site in both ways, on demand, and automatically.

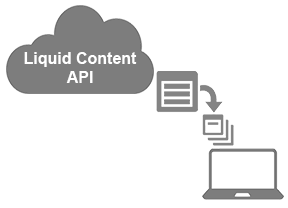
To see the list of visualizers that are in your site, just type:

|  |
| --- |
| dotnet liquid.dll list |

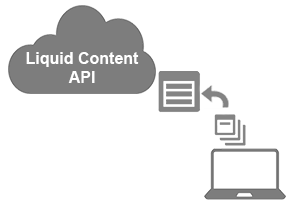
That list will show the names and Ids of each visualizer.

To edit a visualizers’ files (CSS, HTML, JavaScript), just execute the app with the visualizer’s id:

|  |
| --- |
| dotnet liquid.dll sync |



The command will download the files of that particular visualizer, put them in a folder and starts monitoring that folder, so that it will re-upload those files back to Liquid Content, once it detects file changes.



# How to build it

If you want to see how to build an application like this, follow the steps on the [GitHub repository](https://github.com/dnnsoftware/Dnn.Evoq.LiquidContent.Samples.Public/tree/master/OfflineVisualizerSync), or take a look at the source code on that same location.