

## **How Viewshare and Omeka Complement Each Other**

We get a fair number of questions from folks asking about similarities and differences between Viewshare and Omeka. Both are open source tools for working with digital collections. However, they are fundamentally different kinds of tools. This starts by laying out some of the key differences between the two tools, then shows how they can work together to complement each other for a single collection, and finally provides a walkthrough for what is currently the most efficient way to get the two tools to fit together.

## **Omeka and Viewshare Similarities and Differences**

The most central difference between Omeka and Viewshare is that Omeka is a content management system for digital cultural heritage collections while Viewshare is a tool for creating dynamic interfaces to digital cultural heritage collections. To this end, both of these tools are created for different but related reasons. Viewshare is a tool for creating interfaces that let you and your users explore trends and patterns in collections and Omeka serves as a tool for creating online exhibitions, with a particular focus on composing narratives involving collection objects. Here are some bullet points on each of their strengths.

### *Omeka: Online Exhibition and Content Management System*

Omeka....

- Creates a collections-driven website with editable design templates
- Offers a downloadable version for installation on your own Linux server or a cloud-based hosted service on Omeka.net
- Pushes users to use Dublin Core metadata standard to generate compatible data formats
- Supports space for file storage, and automatic resized image derivation
- Serves as a searchable database for an online collection
- Allows data to be shared in a variety of formats
- Provides robust tools for creating narrative online exhibitions based on collection objects
- Offers plugins to extend the basic functionality to engage public users.

### *Viewshare: Exploratory Visual Interface Platform*

Viewshare...

- lets you quickly create dynamic charts, maps, image galleries and timelines of items in digital collections.
- can enhance your collection information by looking up points of latitude and longitude for plain text place names, deriving ISO standardized dates, and breaking apart lists
- supports faceted navigation through historgams, weighted tag clouds, and numbered lists
- is valuable for exploring trends and emergent trends in collection data
- supports browsing collections by data and other numerical ranges

## **Fairfaxpostcards.org A Worked Example of Omeka and Viewshare Together**

To illustrate how Viewshare and Omeka's features work together to complement each other I

created Fairfaxpostcards.org. This small site runs an instance of the Omeka software but has a Viewshare view of the same collection data embedded in it. You can use the Viewshare view to explore the collection information and by clicking on any of the links you will end up on the Omeka pages for each item.

The screenshot displays the 'Fairfax Postcards Collection' website. At the top, there is a search bar and navigation links for 'Browse Items', 'Browse Collections', 'Browse Exhibits', and 'Viewshare View'. The main content area is titled '22' and includes a navigation menu with 'MAP', 'PIE CHART', 'TIMELINE', and 'GALLERY'. A map of Fairfax County, Virginia, is shown with various colored markers representing different categories of historical sites. A popup window is open over a marker, displaying the following information:

- Image URL:** [Image of Gunston Hall]
- Category:** Famous Residences
- City and:** Gunston Hall

Below the map, there is a legend with colored circles corresponding to the categories: Famous Residences (orange), Military History (blue), Places of Worship (light blue), Recreation (yellow), Townscapes (grey), Transportation (light blue), and mixed (white). On the left side of the page, there is a search bar and a list of categories: Famous Residences, Military History, Places of Worship, Recreation, Townscapes, and Transportation. Below this is a 'Related Date' section with a list of date ranges and their counts:

Count	Date Range
1	1670 - 1680
1	1750 - 1760
1	1760 - 1770
2	1790 - 1800
2	1800 - 1810
1	1850 - 1860
1	1860 - 1870
4	1970 - 1990

At the bottom of the page, there is a footer with navigation links: Home, Browse Items, Browse Collections, Browse Exhibits, and Viewshare View.

Together, you get all of the benefits of each of software tool.

By creating the site in Omeka I have created individual web pages for each of my individual postcards. When I uploaded the collection information Omeka prompted me to map the data over to common dublin core data fields. You can see that the site, and my view, are styled and themed by the particular Omeka theme. If I wanted to I could tweak this style however I like. I can also use Omeka's exhibit building functionality to create a narrative exhibition incorporating any of the collection items. You can see an example of this on the browse exhibits page. Lastly, I can use Omeka's native search functions to search through the collection database for individual items.

By incorporating a Viewshare view of the collection into the site I gain the following functionality. First off, I can provide a range of visualizations that would require third party tools or plugins to work in Omeka. In this case, a map, charts, a timeline and an image gallery. Beyond simply offering these views, the tool also lets any user use each of the widgets on the left side of the view to facet through the displayed collection in each view. Most importantly, I can use the related data facets to see all items between a given data range, a feature that is impossible in Omeka because it treats data information as text and not as numerical information. This all means that in each of the individual displays I can browse the items in the collection according to any of the parameters in the widgets. Collectively, this the Viewshare view becomes a potent way to visually explore trends and patterns in the collection. As one final benefit, by building my site this way I have duplicated my collection metadata. It now exists both in my Omeka database on the Viewshare site.

In short, this combination brings together each of the strengths of Omeka with each of the strengths of Viewshare. While there are lots of cases where you wouldn't need to bring both of these tools together, bringing them together is a very complementary process.

### **The Omeka Viewshare Workflow**

To create this site and this view I started from a single spreadsheet

1. **Set up your Omeka instance:** Either install the software or sign up for a hosted Omeka.net instance
  - a. **Install the CSV import Omeka plugin:** This is what is going to allow you to import the collection data spreadsheet.
2. **Prepare and Upload Collection Data**
  - a. **Create your collection spreadsheet:** Items in rows, each datafield in columns. If you want to import files that already exist online into the Omeka Exhibit you can include a column with permanent URIs for each of the items. If you will be manually uploading your files to Omeka you will need to add those files one by one or by using the Omeka dropbox plugin to bulk upload files.
  - b. **Save your spreadsheet as CSV:** This is a simple thing to do in Excel, just save it as a .csv file. Be sure to include URLs for images if you want Omeka to automatically import the image files and derive the various sized images.
  - c. **Upload your CSV file to Omeka:** Once you have the CSV plugin in place you can upload the collection data.
  - d. **Add Omeka Page URLs to your Spreadsheet:** Omeka generates URLs for pages for each of your items, they are automatically numbered from 1 to infinity. You are going to want to add a new column to the collection spreadsheet and put the base URL in it and fill it up with each of the sequential page numbers for your items. Save it as a .xls file to upload it to Viewshare. If you want to add in the URLs for the images you have uploaded to Viewshare you are going to need

to add the Image URLs Omeka creates as well.

### 3. Upload to Viewshare and Create Your View

- a. **Upload the data to viewshare:** Be sure to mark data fields as numbers, image URLs and URLs.
- b. **Augment your data:** You can then generate the points of latitude and longitude or break apart any of the data to use for building your view.
- c. **Build your view:** Just use Viewshare's drag and drop interface to pick which widgets and displays you want to use and customize what data you want to show in each display.

### 4. Embed the Viewshare View into an Omeka Page

- a. **Install the Simple Pages Omeka Plugin:** This plugin lets you add pages to your Omeka site
- b. **Create a page for your view and paste in the embed script:** Just paste in the one line script on the page you create for your view on the omeka site.

### A Quick Note on Continuing to Add to the Collection

With all this said, there are a few notes to keep in mind for updating your site in the future. You are going to need to add items to both the Viewshare View and the Omeka site. There are a few ways to go about this. First, you can manually add items to your Omeka site one at a time through the website. If you do this, you are just going to want to also add the items to the spreadsheet you used to create the Viewshare View. You can always refresh your viewshare view from the spreadsheet, so if you just keep adding them to the end of the list of items this should all work fine and won't involve much redundancy of effort. Alternatively, to avoid this double data entry effort, you can fill the items in on a new spreadsheet, upload them to your Omeka site via the CSV import plugin and then just copy them over to the sheet you are using for your Viewshare view and then refresh from there. In the long run, it would be better for Viewshare to be able to directly pull and update itself from Omeka's Atom feeds. This is something that might come about in the future and would make this a little less tricky but would work just fine for anyone who started using the approach outlined here as well.