

## The Request

Our art faculty use our site, Trove, to create collections of images for use in class, and they'd like to order the images in the collection however they want, for class.

## The Problem

Fedora 4 will be too performance heavy to manage arbitrary sorting of hundreds of works and subcollections. So we have to use something else.

## The Solution

Use ActiveRecord to store the list of ids, accessible as a Rails model.

Hook into dashboards to update our collection order arrays when item membership changes.

Alter solr queries on collection pages to retrieve all records, and re-order as necessary.

## Implementation\*

1. Create a migration and model to store a list of ids as an array in the db, and create a module that saves and retrieves orders from the Collection objects. (app/models/collection\_order.rb, db/migrate/create\_collection\_orders.rb, app/models/concerns/collection\_order\_support)
2. Create routes/methods for updating CollectionOrders, and find the places you'll need to patch to insert item-order updating. (config/routes.rb, app/controllers/hyrax/dashboard/collections\_controller.rb)

3. Add a mechanism in the edit collection view to allow users to order the works manually. We used jQuery UI Sortable to create an AJAX interface. (/app/assets/javascripts/sortable-works, /app/assets/javascripts/sortable-subcollections)
4. Extend Hyrax::Collections::CollectionMemberService and alter query\_solr to retrieve all works and then obey the CollectionOrder. (app/services/collection\_member\_service.rb)

