THATCamp Prime the Fifth Omeka Plugin Workshop

Simplest plugin -- Using Hooks and Filters

(15 min)

Overview of hooks and filters in Omeka

We'll append text to an items show page, and add a new tab to the items edit page.

- Install HookFilterExample plugin
- The `Omeka_Plugin_Abstract` class (new in 1.5) ** Compare to pre-1.5 style function calls
- Difference between a hook and a filter

Activities

- Change the text displayed in the hook
- Follow the same pattern to alter collection pages

Models

(15 min)

How Models work in Omeka

We'll work with the skeleton of a plugin that records additional information of collectors for collections.

- Install ModelExample plugin

- Describing your model to Omeka: `models/Collector.php`

- Setting up the database table with the `install` hook
- Creating and saving records
    - Faking it in the install hook
- /models/Collector.php vs models/CollectorTable.php
    - `get_db()->getTable('Collector');`
- /application/libraries/Omeka/Db/Table.php
- /application/libraries/Omeka/Record.php

Activities

- Change the plugin to use the Omeka pattern of `findBy()`
- Fix the bug! Non-public Collectors still show up on the public page
    - Add a filter for public in CollectorTable and alter the hook to use it.
- Add the same data to the admin show page

Controllers and Views

(20 min)

How Controllers and Views work with URL paths

We'll work with a plugin skeleton that creates a new URL path to display data

- Install ControllerViewsExample plugin

- What a Controller does

- Grabs data about a Model from database and make it available to a view
- based on an Action
- `controller/action` URL paths
- shared, public, and admin directories and URL paths
- `index` is automatic Controller in Zend
- watch things break at other URLS!
- `application/libraries/Omeka/Controller/Action.php`

Demo

- create some data in a variable in the indexAction and display it in the view

Activities

- create some data in a variable in the indexController and display it in the view
- create a new view and action at a different URL
- create different admin and public views

Putting it all together with Omeka defaults

Take all that we've learned, and add controllers and views for the ModelExample plugin.

Omeka has browse, show, edit, delete Actions

1. Create a Controller and a View to display information about each Collector
2. Set the default 'scaffolding' class in init() method
3. Build the `show` view for the public side
4. Build a `browse` page for Collectors

Extra bonus activities, probably as exercises for after our session

- Build an `add` and `edit` page for the admin side to add Collectors, and add them to a Collection. Bonus points if you use existing user and/or collector data
- Add form to the Collection add and edit pages, using the appropriate hooks, to add or remove existing Collectors to the Collection

Further exploration

- Explore more hooks and filters
- Look at examples of using pagination in browse views in other plugins
- Play with the implied hooks, e.g. `hook_after_save_item` to do things when an item is saved
- Explore ACL and permissions

Watch [Updating Plugins for 2.0](#) for upcoming changes. In most cases, the changes for 2.0 are backward-compatible.