Theming Omeka, METRO, July 12 2012

Default Theme Files

Located in Omeka/application/sites/views/scripts

- collections: theme files for displaying Omeka collections
- common: templates used across all theme files (like header.php and footer.php)
- css: CSS files
- error: error pages
- file: theme files for displaying files
- images: jquery-ui and other icons used when displaying images
- items: theme files for displaying Omeka items
- javascripts: jquery and tinymce files
- upgrade: pages used during Omeka's upgrade process
- users: pages used during registration and (public) login

Most important files are usually

- show.php -- used when displaying a single thing (collection, item, etc.)
- browse.php -- used when displaying several of a thing

Activities

- Copy the example-theme theme into your themes directory
- Copy over the items/show.php file
- Copy over the items/browse.php file
- Switch to the example theme and add alter the page title for items/show
- Add a class to the bodyclass

Displaying Item data

- item() : displays element set data
- loop_items() : used in a <?php while(); ?> structure to loop through items in browse

Activities

- Customize the metadata shown by getting rid of custom_show_item_metadata (some themes let you do this in their configuration settings)
- use item() to display the title and description
- Add an additional metadata field (e.g., identifier) to the display for each item in items/browse
- use collection('name') to display just the name of the collection in items/show
- use collection('name') to display just the name of the collection in items/browse
    - First use get_collection_for_item() to get the collection
    - Use collection('name', array(), $collection)

Inserting your CSS and Javascript

- queue_css() : use before head() to queue CSS files to be added
- queue_js() : use before head() to queue javascript files to be added

Activities

- Insert the css and javascript files into the page
- Change the css file to match with the bodyclass you introduced above

Overriding plugin views

- Plugins have a 'views/public' (or 'views/shared') folder that contains the views (pages) they generate. You can override those the same way you override the core views. Copy the first folder under 'public' or 'shared' to your theme

Activities

- Install SimplePages and hack the public/show page to alter the title

Overriding the front page

- index.php in default theme files directory

From Scratch Theme

- Contains files and folders for collections, items, header and footer, css, and more to get you up and theming quickly.

Activities

- Download From Scratch and look through style.css

Override the front page

Building a theme for release

Plugin hooks

Hooks http://omeka.org/codex/Hooks

Functions beginning with plugin_ provide mechanisms for plugins to add content to pages. Not including theme in your theme may cause plugins to lose functionality. Usually, it is safe to move them around though.

Hook public_theme_header() is fired by plugin_theme_header() in the theme. Similarly for all hooks.

Theme Configuration

Omeka has a robust mechanism for theme designers to easily create a configuration page for your themes. This happens via a config.ini file for the theme, and by calling get_theme_option() appropriately in your theme files.

config.ini file and the theme config page

The theme config.ini page lets you give instructions to Omeka about how to put together the theme config page. For each option that you want to create, it needs:

- input type
- label
- description
- optional settings specific to the input type
- optional validation settings

When using get_theme_option(), logo in config.ini becomes get_theme_option('Logo'). background_header becomes get_theme_option('Background Header')