



CENTER FOR TEXTUAL STUDIES AND DIGITAL HUMANITIES

DIGH 402 - Introduction to Digital Humanities Design and Programming

Spring Semester 2014

Week 13

Object Oriented Programming

Object Oriented Programming - Abstract overview of current framework structure

use and rendering of user installed plugins - viewer

- pass returned plugin_ids to load_plugins() method in Loader class
- load_plugins() prepares plugin details and sets plugins array for use with the View class
 - new \$plugins property added to Loader class
- plugins array can now be passed to draw_middle() via draw_theme() method
- draw_plugins() method in View class allows us to output available plugins including
 - plugin name, description, directory
 - required files from plugin where applicable
- these files can now be drawn within the framework template as part of the draw_middle() method

[GitHub Code](#)

Object Oriented Programming

Object Oriented Programming - Abstract overview of current framework structure

use and rendering of user installed viewer plugins - updated View class

- new draw_plugins() method
 - accepts \$plugins array parameter
 - checks \$plugins array is not empty and then builds HTML for plugin_options
 - loops through \$plugins array and formats output for plugin_options
 - uses draw_js() methods to output plugin files and code
 - this loads the plugin for the user
- updated draw_middle() method
 - accepts a new parameter for \$plugins
 - calls draw_plugins() method with \$plugins parameter before draw_main() method
 - hierarchy allows us to ensure that plugin is above content in DOM

[GitHub Code](#)

Object Oriented Programming

Object Oriented Programming - Abstract overview of current framework structure

viewer plugins - how to build

- built around JQuery Javascript library (or plain javascript if you prefer)
- allows injection of JS code within framework content
- allows manipulation, editing etc of existing content
- AJAX manipulation and editing of DOM, texts, images...
- quick and easy plugin development
- easily tested outside of framework before deployment as a plugin

Object Oriented Programming

Object Oriented Programming - Abstract overview of current framework structure

viewer plugin - v.basic test image_magnify.js

- create a new JQuery function
- associate mouse click event with element id #image_magnify
- associate toggle event with element id #content and image element within
- hide and show image element for each click

[GitHub Code](#)

Object Oriented Programming

Object Oriented Programming - Abstract overview of current framework structure

helpful Javascript - JQuery UI

- added js and css files to constants
- added JQuery UI css and js files to css and js arrays in draw_theme() method in Loader class
- css and js files are now loaded in the head section of our framework template
- JQuery UI is a very useful UI library for quick deployment of JQuery effects, options...
- eg: dialog/modal windows, accordion effects, autocomplete etc...

[GitHub Code](#)

Object Oriented Programming

Object Oriented Programming - Abstract overview of current framework structure

adding metadata - image, text etc...

- metadata can be specified for any of the content types
- enter required metadata in content_meta table completing each field as necessary
- enter meta_id in content_lookup for associated content_id...
- content_id can be replicated to match different content_type_id for image, text etc... for the same item
- metadata output in viewer will match user selected controller/format and param
 - eg: metadata for [content/text&id=6](#)
[content/image&id=6](#)

Object Oriented Programming

Object Oriented Programming - Abstract overview of current framework structure

updated metadata.php

- updated meta_row_query() method to check for specific controller and format
- allows differentiation in the database between specified metadata per format per content record
 - eg: different metadata records for contenttext and contentimage for the same content id
- output is the same to the draw_meta() method in the View class

[GitHub Code](#)

Object Oriented Programming

Object Oriented Programming - Abstract overview of current framework structure

organisation of site content - taxonomy structure

- new DB tables added to handle taxonomy and content structuring
 - content_group = organisation for taxonomy subsets such as gallery, work, catalogue...
 - taxonomy = definition of taxa names within our overall taxonomy
 - taxonomy_lookup = taxa_id and taxa_parent_id
 - eg: 'Brixham' with parent taxa set to 'Devon'
 - or 'Time Passes' with parent taxa set to Uniform Edition
- two new columns in content_lookup table for 'content_group_id' and 'taxa_id'
 - each item of content can be assigned to a taxa and group
 - eg: 'gallery' belonging to 'Brixham'
- updated constants for new taxonomy DB tables

Object Oriented Programming

Object Oriented Programming - Abstract overview of current framework structure

updated Loader class - handling 'taxonomy'

- load_controller() method updated to include taxonomy check based upon specified 'id' parameter in URI
- taxonomy details are now retrieved from the DB using the new TaxonomyController class
- after checking for 'group' parameter taxonomy details are set in \$content_meta
 - shows title and description for specified taxonomy in URI
 - these are output instead of format, eg: image or text, title and description
 - output and rendered in sidebar

[GitHub Code](#)

Object Oriented Programming

Object Oriented Programming - Abstract overview of current framework structure

simple new class for handling 'taxonomy'

- standard getter method for retrieving taxonomy row query results from DB
- taxonomy_row_query() method queries the DB to retrieve all taxonomy results for a specified taxonomy id
- taxonomy id is passed as a parameter from the now standard router URI
 - eg: content/image/gallery&id=3
 - id=3 is the taxonomy id we are searching for in the DB

[GitHub Code](#)

Object Oriented Programming

Object Oriented Programming - Abstract overview of current framework structure

updated Router class - handling 'group'

- Router class can now handle additional user selected option in URI
 - eg: content/image/gallery&id=3
 - group option is the third part of the URI eg: gallery
- params['id'] now refers to taxa id
 - eg: taxa_id = 3 in the DB table content_lookup
- value of 'group' is retrieved using a getter method in auto_load_controller() method in Loader class
- additional 'group' parameter now passed to load_controller() method in Loader class

[GitHub Code](#)

Object Oriented Programming

Object Oriented Programming - Abstract overview of current framework structure

updated Loader class - handling 'group'

- updated `auto_load_controller()` method to set `$group` property
- `load_controller()` method updated with `$group` parameter
 - new group view is set based on group specified in URI eg: gallery
 - group class is loaded eg: gallery.php
 - group and taxonomy attributes are passed to new `get_group_content()` method in group class eg: in gallery.php
 - formatted group content is returned and passed as parameter to standard `draw_theme()` method
- plugins can also be passed to `draw_theme()` method for groups...

[GitHub Code](#)