



CENTER FOR TEXTUAL STUDIES AND DIGITAL HUMANITIES

402 - Introduction to Digital Humanities Design and Programming

Spring Semester 2016

Week 12

Object Oriented Programming

Working framework so far...v0.4

[Home Page](#)

[Output Image Content](#)

[Output Text Content](#)

Object Oriented Programming

Working framework so far...v0.5

[Home Page](#)

[Output Image Content](#)

[Output Text Content](#)

402framework v0.5

- Compare and contrast v0.4 with v0.5
 - what has changed in the code and where?
 - what has changed in the view for the user?
 - update flow diagram

402framework resources

- Update SQL to v0.5 - [GitHub SQL v0.5](#)
 - use with 402framework v0.5
- [GitHub v0.4](#) | [GitHub v0.5](#)
- [v0.4 test site - image](#) | [v0.5 test site - image](#)

402mini

A fun, mini framework for testing basic PHP routing...

- Compare and contrast v0.1 with v0.2
 - what has changed in the code and where?
 - what has changed in the view for the user?
 - update flow diagram

402mini resources

- [GitHub v0.1](#) | [GitHub v0.2](#)
- [v0.1 test site](#) | [v0.2 test site](#)

Object Oriented Programming

Object Oriented Programming - Abstract overview of current framework structure

Recent updates to v0.5 include:

- add plugins for text, images etc...
- add some more error checking and reporting...
- add default content handler for non controller/format/params URI requests including index.php
- sanitise content request and returned content
- update DB tables, content, metadata...

and so on...

Object Oriented Programming

Object Oriented Programming - Abstract overview of current framework structure

v0.5 - handling default views - image/text etc

- default handlers for format in user requested route URI
 - content/image, content/text etc
 - eg: frame/view/image.php
- new constant in directory.php for location of images, texts etc
- viewer classes, eg: ImageViewer, extends BuildHTML class
 - allows easy building and output of HTML required for viewer
- viewer classes also need to check any add-on plugins
 - magnify, zoom etc for image
 - mapping
 - search highlighter
 - and so on...

Object Oriented Programming

Object Oriented Programming - Abstract overview of current framework structure

v0.5 - updated Loader class

- after checking for returned content, metadata and selected theme
 - we use \$format (image/text etc) to load required viewer
 - check viewer class exists
 - instantiate viewer object
 - set any required viewer or format attributes
- we then use the initial returned raw content data
- content is formatted and returned from the applicable format viewer
- content is passed to draw_theme() method

[GitHub Code](#)

Object Oriented Programming

Object Oriented Programming - Abstract overview of current framework structure

v0.5 - handling default views - ImageViewer class

- ImageViewer class in image.php
- abstracts rendering of images from DB
- uses constant for media/images directory
- extends BuildHTML class to allow us to easily build required HTML elements
- private static properties
- format_image_view() method called to create required HTML from content
 - 3 parameters for content, viewer attributes, image attributes
 - customised attributes for div parent wrapper
 - customised attributes for image element

Object Oriented Programming

Object Oriented Programming - Abstract overview of current framework structure

v0.5 - handling default views - TextViewer class

- uses similar pattern to ImageViewer class
 - format_text_view() is the main method for rendering the content
 - uses BuildHTML to return the required HTML elements
- abstracts rendering of initial text content from DB
- text currently stored in content table in DB, and not a file etc
 - this can be handled in a similar manner to the images
- format of txt in DB will be dependent upon import or edit plugins
 - eg: HTML format, txt, TEI etc...

[GitHub Code](#)

Object Oriented Programming

Object Oriented Programming - Abstract overview of current framework structure

v0.5 - initial plugin implementation

- new plugin directory for user developed and imported plugins
 - stored in root directory for framework
 - directory = /plugins
- new constant in directory.php for new plugin directory and admin plugins
- plugins initially added to DB tables 'plugins' and 'plugins_lookup'
- code is stored in plugin named directory in /plugins directory
 - eg: /plugins/image_zoom
 - /plugins/image_magnify

[GitHub Code](#)

Object Oriented Programming

Object Oriented Programming - Abstract overview of current framework structure

v0.5 - new plugins and plugins_lookup table in DB

- plugin table allows us to store basic information on available framework plugins
 - plugin_id, plugin_name, plugin_desc, and plugin_directory
- plugin_name can be used to display plugin title in framework options
- plugin_desc for tooltips etc
- plugin_directory informs the application where to find the files within the framework's plugin directory
- plugins_lookup table references plugin_id to plugin_type and content type
 - eg: plugin_id=1, plugin_type=content, and content_type=image

[GitHub Code](#)

Object Oriented Programming

Object Oriented Programming - Abstract overview of current framework structure

v0.5 - updated Loader class - adding content and format plugins

- first we check for available plugins relative to current controller and format in router URI
 - eg: content/image
- load plugin file 'plugin.php', set plugin_class, and check that class exists
- instantiate object for plugin_class and call get_plugins() method
 - this checks the available plugins for controller and format
 - checks the plugins_lookup and plugins tables
 - returns the details for the available plugins including plugin_id
- outputs plugin from code in defined plugin_directory
 - eg: css and js files

[GitHub Code](#)

Object Oriented Programming

Object Oriented Programming - Abstract overview of current framework structure

v0.5 - check for installed plugins available in new DB table - plugin.php

- new plugin.php file and PluginController class
 - file stored as a controller in /frame/controller
- requires query_builder.php file and extends parent class BuildQuery
- queries plugins and plugins_lookup tables in database to check for available plugins
 - check is relative to controller and format specified in router URI
- returns results to Loader class
- also method to check plugin details using plugin_id as a specified parameter

[GitHub Code](#)

Object Oriented Programming

402framework - v0.6

- update framework to v0.6 - available on GitHub
 - [GitHub v0.6](#)
- update database SQL to v0.6 - available on GitHub
 - [GitHub SQL v0.6](#)

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** table= organisation for taxonomy subsets eg: gallery, work, catalogue...
 - **taxonomy** table = definition of taxa names within our overall taxonomy eg: devon, cornwall...
 - **taxonomy_lookup** table = taxa_id and taxa_parent_id
 - eg: 'Brixham' with parent taxa set to 'Devon'
 - or 'Time Passes' with parent taxa set to 'Uniform Edition'
- *NB:* both taxa come from taxonomy table...
- 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

Current framework v0.6 examples

- [image page](#)
- [text page](#)
- [gallery](#)
- [book](#)