ResourcSpace

**Modifications to source code**

* Commented out default easy tooltip javascript on /include/searchbar/ - line 590
* Added TMS field to be populated in Admin view

-line 192

**CRON Management**

Several aspects of the new integration involve utilizing cron jobs to schedule executed tasks. The basic structure of this cron system is that when a cron is needed (such as a failure to connect to external system) it is added by the requesting page to the temp directory. There is a scheduled cron job called getcron that looks every hour for new cron jobs in the temp directory. If new cron jobs are found it then moves them to the /etc/cron.d directory to be executed. When the cron is executed it is removed from the cron.d directory and if it fails to execute it is added back to the cron.d directory to be executed again. If 3 attemps to execute the cron job fails the cron job is ignored and an email is sent to the configured admin for the system to resolve the issue.

* Areas that utilize this cron system:
  + ElasticSearch
  + TMS
  + Media Preview Creation

**Plugins/**

* **Google\_Forms**

This plugin was designed to use google places API as an authoritative control for geographical naming conventions

* + Google\_forms.yaml

Used to configure ResourcSpace plugin definitions

* + Hooks/
    - All.php

This file utilizes the hook Editbeforesave to gather lat and long data submitted in $\_POST data and updates the `resource` tables geo\_lat and geo\_long fields in the MySQL database. This data is used by the geomaps location pins and location searching in ResourceSpace.

* + - Edit.php

This file uses the hook Autolivejs to add javascript to the bottom of the document. This script adds a new “location selector” input to the main edit form. This selector field utilizes google places autocomplete functionality to allow a user to input data and select a location based on their input. The selection is then broken down into location components that complete other fields in the form such as City, State, and Country components. Lat and Long data are also appended to the form as hidden fields.

* **TMS\_Connect**

This TMS plugin is used to connect externally to the TMS API. The plugin itself

* + Tms\_connect.yaml
  + Hooks/
    - Edit.php

This file utilizes the hook Autolivejs to place javascript in the bottom of the document. It sends a request to TMS.php when user input is detected in the specified “ObjectId” field and autopopulates the related Art Object fields with the returned results.

* + Pages/ajax/TMS.php

This file is responsible for the control in sending and receive data via the TMS API.

* + cron\_tms.php

This file adds a cronjob to the /tmp directory which is picked up by getcron.php and is responsible for updating an Assets TMS data in the event that a user entered an Object ID (and saved it) but the cURL request to retrieve the data from the API was down at the time.

* **Mia\_Styling**
  + Mia\_styling.yaml
  + Config/
    - Config.php
  + Css/
    - Style.css
  + Gfx/
    - Helpfiles/

This directory is home to the images for uploading help files for the system

* + - Homeanim/

This directory is home to the images for the homepage animated slider

* + - Interface/

This directory is home for any interface images such as icons

* + Hooks/
    - All.php

This page utilizes the hook ***Additionalheaderjs*** to add masonry.pkgd.min.js to the header. It utilizes the hook ***Searchbarbeforebuttons*** to compile a select list of all of the system users and adds it to the simple search sidebar for searching based who contributed assets to the system. It utilizes the hook **Footerbottom** to add javascript and logic to the footer for setting, active nav, display animation, and the sidebar checkbox icons. ***Swfplayer*** and ***Mp3player*** hooks are used to replace the default flash players with HTML5 audo and video and the ***Searchbarbottom*** hook to send and ajax request to /pages/kill\_cookie.php when the clear button in the sidebar is pressed.

* + - Edit.php

This file utilizes the ***Replacesaveerror*** hook to replace the default save errors that where echoed out on the pages as an alert of all errors compiled together.

* + - Help.php

This file utilizes the ***Replacehelp*** hook to replace the help default help content with the content from the help table in the database.

* + - Search.php

This plugin utlilizes the **Searchaftersearchcookie** hook to set a contrib key in the cookie to the value of the contributed by select list in the simple search sidebar.

* + - Team\_home.php

This file utilizes the hook Customteamfunction hook to add the manage “manage help section” link to the Team Center.

* + - Themes.php

This file utilizes the ***Additionalsmartthemetool*** hook to add html elements wraping the theme sections in order to layout the thems with a mason grid as well as turns the tool options to a popup to preserve space.

* + - View.php

This file utilizes the ***Beforeresourcetoolsheader*** hook to add a big red message if an asset flag is detected.

* + Pages/
    - Help/
      * Create.php

Used to create a new help section

* + - * deleteimage.php

Used to delete a help images from directory

* + - * delete\_page.php

Used to delete a help text entry from the database

* + - * getimages.php

Used to retrieve images from directory

* + - * manage.php

Used to manage/view all help-text entries

* + - * nav.php

navigation view to include on other pages

* + - * upload.php

Used to upload images from create.php via ajax.

* + - Killcookie.php

Used to clear the search cookie when requested.

* **Mia\_Upload**

Installation: Install plugin and activate. Two directories need to be created in filestore/tmp called upload\_log/ and uploads/. These serve as the temp directories to handle upload requests and for logging statuses back to the uploader.

* + Config/
    - Config.php
  + Css/
    - Upload\_styles.css
  + Hooks/
    - All.php
  + Pages/
    - abort\_upload.php

Used by upload class to abort upload and remove file from temp directory

* + - background\_previews.php

Used to process media preview creation as a background process and allow upload to continue.

* + - check\_existing.php

Used to handle an ajax request from upload class to check if a file exists before sending upload request.

* + - chunk.php

Used to take uploaded file input stream and write binary chunk to temp file in temp directory.

* + - tmp\_cleanup.php

Executed by cron job to make sure tmp directory cleans up old files on a set schedule.

* + - upload2.php

Used by single upload process to perform metadata extraction and TMS connection/extraction and return data for auto-population of main form.

* + - save.php

Used by single upload process for second stage of ingestion of creating resource, previews, thumbnails, and ingesting data.

* + - upload\_batch.php

Used by batch upload process for metadata extraction, TMS connection, preview creation, thumbnail creation and ingestion of data.

* **Elasticsearch**

The elastic search plugin utilizes two hooks in resource\_functions.php to run push and delete requests to ElasticSearch anytime a resource is uploaded, modified or deleted. All that needs to be done to setup is configure your ElasticSearch settings in the config file to connect to you ElasticSearch database. If there are any fields you do not want to pull across to your elasticsearch system: define them in the $dont\_pull. Any adjustments/formatting to the data that you want done pre-ingestion to ElasticSearch can be done in the mia\_results function located in include/elastic\_functions.php in the plugin directory.

* + Elasticsearch.yaml
  + Config/
    - Config.php
  + Hooks/
    - All.php

Utilizes ***beforedeleteresourcefromdb*** and ***aftersaveresourcedata*** hooks to tie into anytime a resource is saved or delete and pushes the request to ElasticSearch. \*\*Note that deleted resources are not removed from ElasticSearch until they are permanently deleted from the system, NOT when they are only marked for deletion.

* + Include/
    - Elastic\_functions.php

General function used for elastic search plugin. Key function in this file is mia\_results which takes metadata removes empty values and the data for elasticsearch (ie converting resource\_type id to a string value).

* + Pages/
    - Cron\_elastic\_delete.php

Used when a request to delete a file from ElasticSearch is unsuccessful and needs to be resolved. Scheduled to attempt the resolve 3 times at hour intervals and then notifies admin.

* + - Cron\_elastic.php

Used when a request to save a file from ElasticSearch is unsuccessful and needs to be resolved. Scheduled to attempt the resolve 3 times at hour intervals and then notifies admin.

* **Api\_download**
  + Index.php

This is used by external request sent to the system in order to allow downloads without being logged in if API authentication is successfull.

* + Api\_download.yaml
* **Api\_view**
  + Api\_view.yaml
  + Index.php

Used to allow external systems to validate via the API as MetaMia user if they are not logged into the system and view the as it would appear they where logged in. If the user is already logged in when the page is request they are re-directed to the actual page as their own user.

* **Contacts**
  + Config/
    - Config.php
  + Contacts.yaml
  + Dbstruct/
    - Table\_contacts.txt

Database structure for contacts MySQL contacts table

* + Hooks
    - Edit.php
    - Team\_home.php

Used to populate a link to manage.php for contacts plugin in Team Center.

* + Include
    - Contact\_functions.php
  + Pages/
    - manage.php

Used to create and edit contacts

* + - setup.php

Used to view all contacts in system with link to edit and delete contacts.

* Customizations
  + getcron.php
  + Exiftool\_Config
  + Function rsReady();