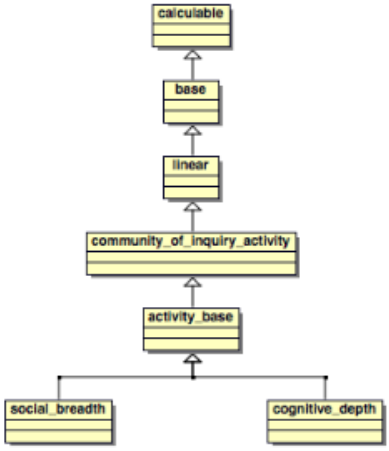
Open Forums V.3.4.3

"The Open forum activity enables students to post and reply to comments, suggestions, and questions asynchronously. Use the Open forum to provide a platform on which your students can communicate with each other." (From Blackboard Open LMS Documentation)

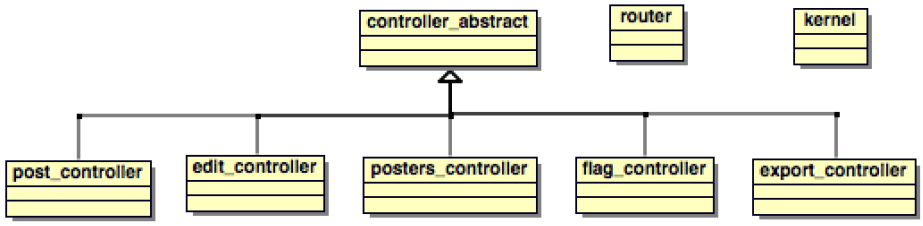
Classes Architecture

This section shows the classes and its structure for this plugin. Each group is separated in a folder depending on its functionality

Analytics



Controller



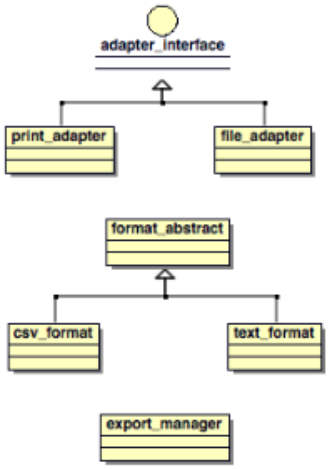
Event

In this folder, the list of events used in this plugin can be found.

- assessable_uploaded.php
- course_module_instance_list_viewed.php
- course_module_viewed.php
- course_searched.php
- discussion_created.php
- discussion_deleted.php
- discussion_moved.php
- discussion_pinned.php
- discussion_subscription_created.php
- discussion_subscription_deleted.php
- discussion_unpinned.php
- discussion_updated.php
- discussion_viewed.php
- post_created.php
- post_deleted.php
- post_updated.php
- readtracking_disabled.php
- readtracking_enabled.php
- subscribers_viewed.php
- subscription_created.php
- subscription_deleted.php
- user_report_viewed.php

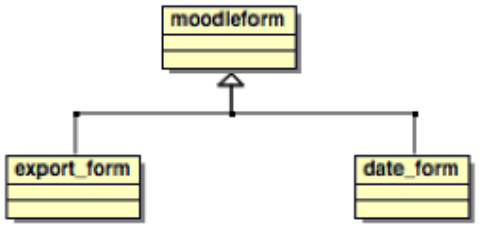
Export

This folder has various data export implementations.



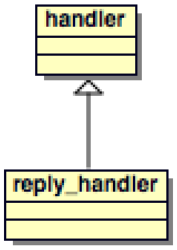
Form

Some specific forms can be found in this folder.



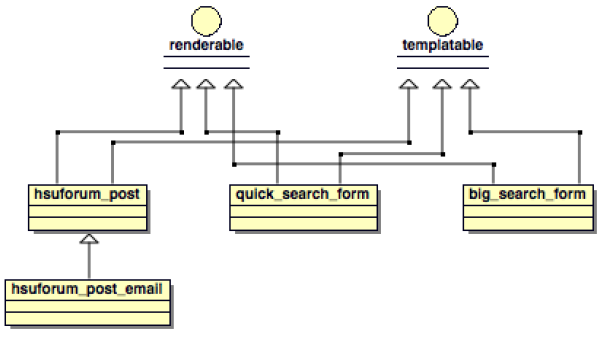
Message

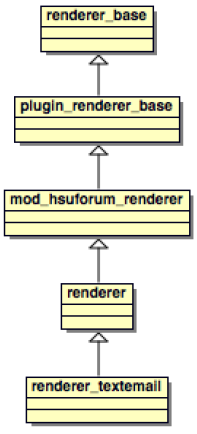
It has a class to handle the process of replying to a forum post.



Output

In this folder you'll find all the logic pertaining rendering the different UI elements.



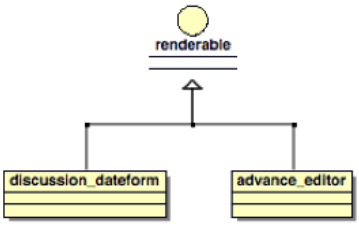


Privacy

This folder holds the privacy components for GDPR compliance.

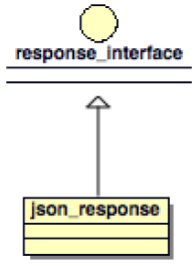
Renderable

This folder holds "renderables", which are classes that describe UI elements can be rendered several times.



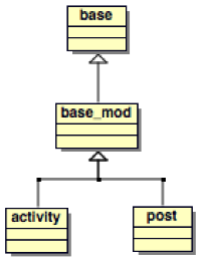
Response

This folder holds a WebService response implementation.



Search

This folder holds the search implementation pertinent to this plugin.



Service

This folder holds various services which allow accessing this plugin's functionality.

- discussion_service
- form_service
- post_service

Task

This folder holds the scheduled tasks to be executed for this plugin.

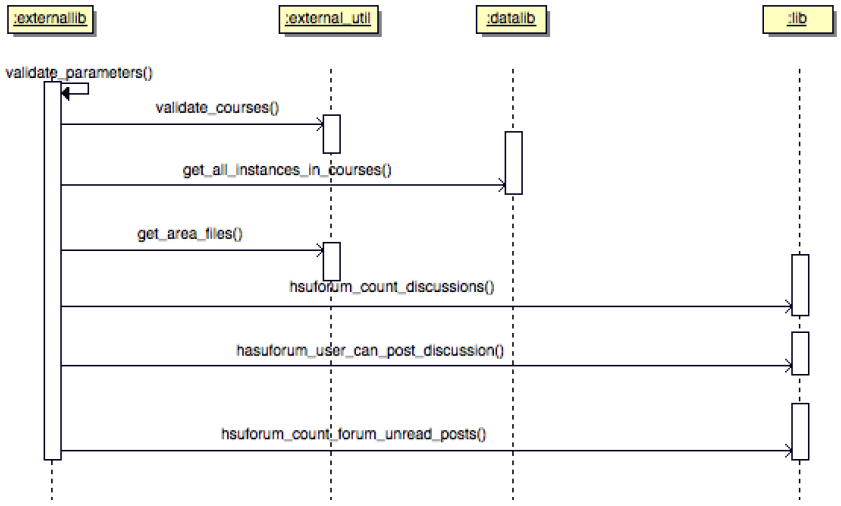
Web services

Open Forum uses a set of web services that are exposed to external components through its external lib. These services are:

Note: The following sequence diagrams do not depict the exact workflow of data. It's a description of some method calls between components.

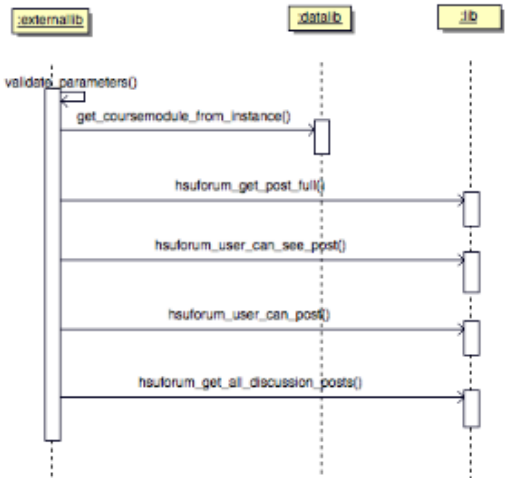
Get forums by courses

- Returns a list of forum instances in a provided set of courses, if no courses are provided then all the forum instances the user has access to will be returned
- Parameters:
    - Array of course IDs
- Returns a list of forums in a provided list of courses, if no list is provided all forums that the user can view will be returned.



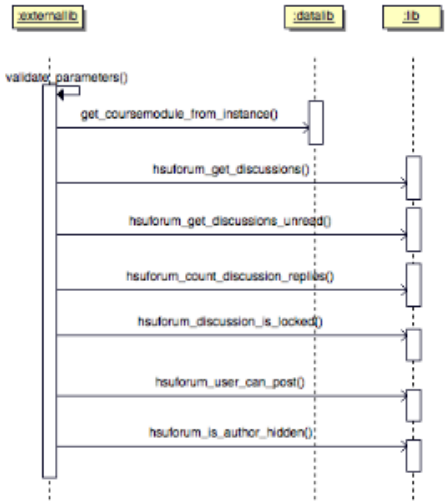
Get forum discussion posts

- Returns a list of forum posts for a discussion.
- Parameters:
    - Discussion ID



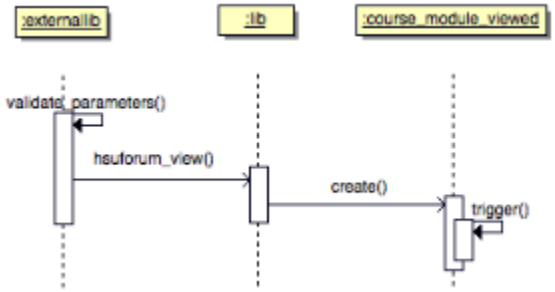
Get forum discussions paginated

- Returns a list of forum discussions optionally sorted and paginated.
- Parameters:
    - Forum id: hsuforum instance id
    - Sort by: Sort by this element: id, time modified, time start or time end
    - Sort direction: Ascend or Descend
    - Page: Current page
    - Per page: Items per page



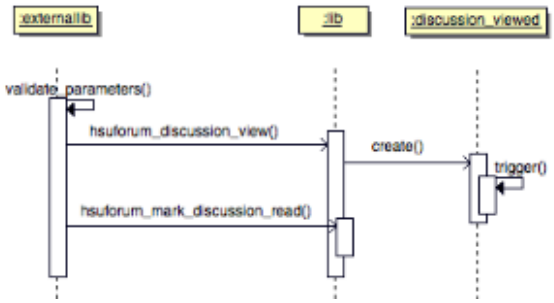
View forum

- Trigger the course **module viewed** event and update the module completion status.
  Parameters:
  - forum id
  Triggers:
  - Event \mod_hsuforum\event\course_module_viewed



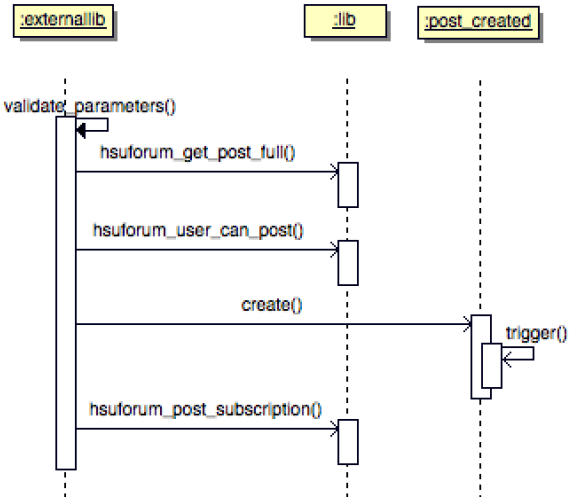
View forum discussion

- Trigger the forum discussion viewed event.
  Parameters
  - discussion id
  Triggers:
  - Event \mod_hsuforum\event\discussion_viewed



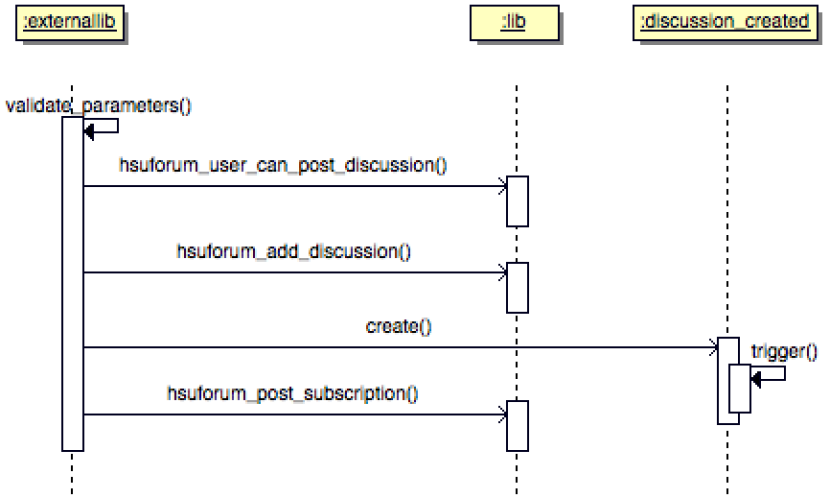
Add discussion post

- Create new posts into an existing discussion.
  Parameters
  - post id: the post id we are going to reply to (can be the initial discussion post)
  - subject: new post subject
  - message: new post message (only html format allowed)
  - options:
    - name: The allowed keys (value format) are:
      - discussionsubscribe (bool); subscribe to the discussion?, default to true
      - inlineattachmentsid (int); the draft file area id for inline attachments
      - attachmentsid (int); the draft file area id for attachments
    - value: The value of the option, this param is validated in the external function.
- Triggers
  - Event \mod_hsuforum\event\post_created



Add discussion

- Add a new discussion into an existing forum.
  Parameters:
    - forum id: Forum instance id
    - subject: New discussion subject
    - message: New discussion message (only html format allowed)
    - group id: The group, default to 0
    - options:
        - name: The allowed ,eys (value format) are:
            - discussionsubscribe: (bool); subscribe to the discussion?, default to true
            - discussionpinned: (bool); is the discussion pinned, default to false
            - inlineattachmensid: (int); the draft file area id for inline attachments
            - attachmentsid (int); the draft file area id for attachments
        - value: The value of the option, This param is validated in the external function.
- Triggers
    - Event: \mod_hsuforum\event\discussion_created



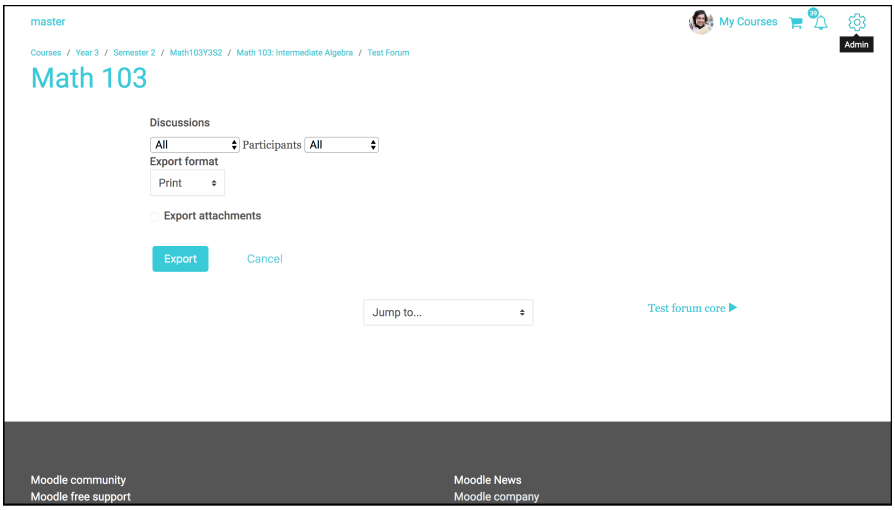
Can add discussion

- Check if the current user can add discussions in the given forum (and optionally for the given group).
  Parameters:
  - forum id: Forum Instance ID
  - group id: The group to check, default to the active group. Use -1 to check if the user can post in all the groups.

Open Forums features: Export

This document aims to give a good understanding of one of the key features of this activity module, the **export** featu re.

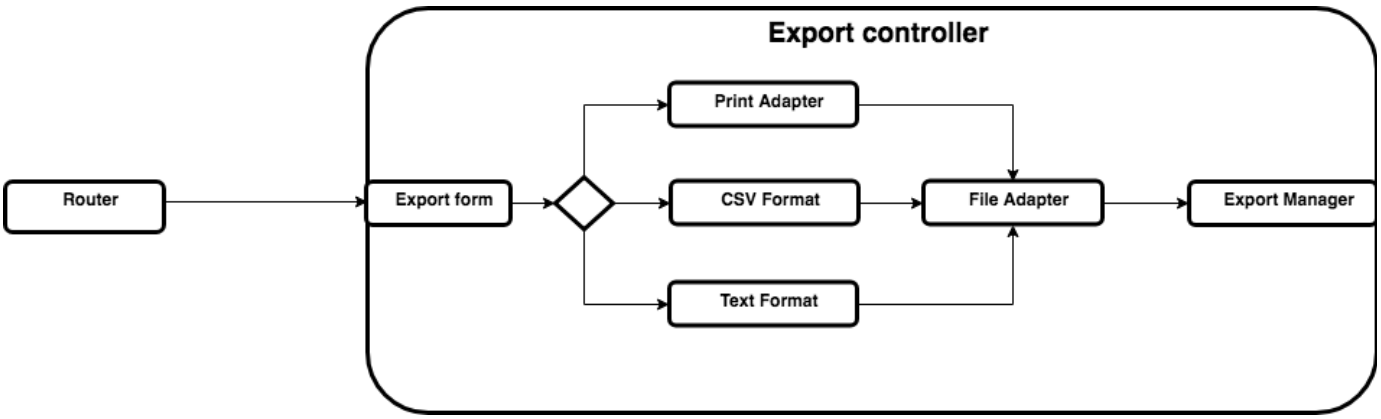
User UI



User UI has the following possible values regarding formats:

- Print: Shows the desired information in a new page.
- CSV
- Plain Text

How it works



Router

The router role is pretty simple, it just handles which action will be performed, in this case "**Export**".

Export controller

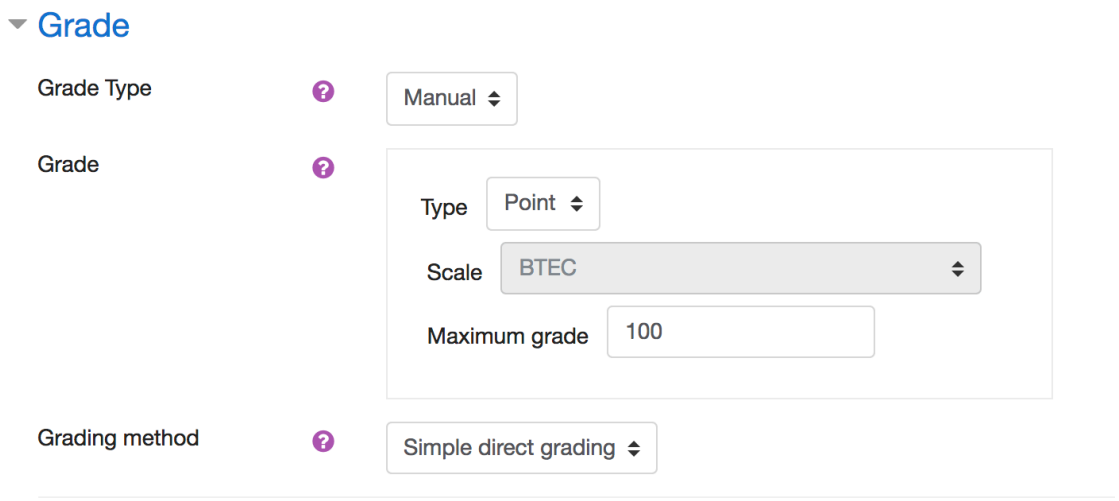
This controller is the responsible for handling the user requests which are processed using the following steps.

- Export form:
  The definition is pretty simple, it only has 3 select fields, a check box and submit and cancel buttons. However, its gathered data are the base on which the export process extracts the possible parameters of the user request. This means that the values on each select input depend of the forum settings, participants and users capabilities. Functions inside the class find each discussion and user who have posts on each one of those items. There is a select field to choose the export format with predefined values (print, CSV, Plain text).
- Formats and adapters:
  This classes have common functions among each other, **get_extension**, **export_discussion** and **export_post**.
  Depending on which format is chosen, the controller makes use of the following classes:
  - **print_adapter:** Uses the the render functionality from Open Forums.
  - **csv_format:** Uses predefined columns (id, discussion, subject, author, date, message, attachment and private reply) and gives a readable format to dates.
  - **text_format:** Gives a more structured look to the information, discussions are displayed first followed by posts and replies. It's a parent - child structure.
- **File Adapter** and **Export manager**: those two classes work very close. Export manager uses as input parameter an instance of **file_adapter** which holds an instance of the format class (**print_adapter**, **csv_format** or **text_format**).
  Flow overview:
  - The process starts with **export_discussion()** or **export_discussions()** from **export_manager.** An specific discussion can be passed, otherwise all existent discussion will be processed.
  - A new file is created using **initialization()** from **file_adapter.** The name of the file depends of the discussion passed to the function, if there is no value for this, then the forum name will be used. Each format is responsible for the required configuration when processing data, that happens when calling **init()** (on each format class). Otherwise, default configuration will be established by the abstract class each format implementation class extends from.
  - All discussions are retrieved and processed. For each discussion, all posts are retrieved and then are cleaned; E.G, if a user cannot see posts or groups of posts, the unwanted information would be removed
  - Once the information has been prepared, the method **send_discussion()** from the **file_adapter** goes through every discussion, so existing posts and their attachments will be placed in the new file using the given format (using format implementation class specific functions; **get_extension**, **export_discussion** and **export_post**.)

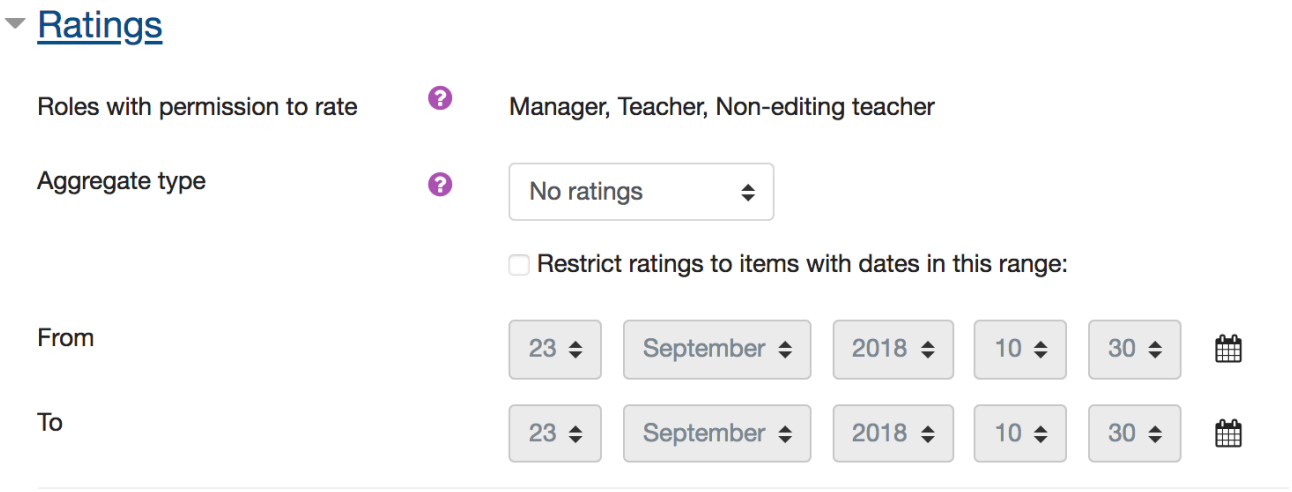
Open Forums features: Grades

This page describes the grading feature. Open Forums allow grading given it was configured during forum creation.

TEACHER UI



Grade has a separate section from rating in Open forums. Of course, they can also be rated if the instructor prefers it.



HOW GRADING WORKS?

There is a list of functions defined in the plugin's lib.php that handle the grading process. Everything is done using the core_grades standard.

- hsuforum_update_grades()

This function updates the forum grades in the database. After validating several variables as the grade type set in the forum creation and the user that needs to be graded, the function hsuforum_grade_item_update() is called.

- hsuforum_grade_item_update()

Don't let the name of the function distract you. This function creates or updates the grade item stored in the database. The only parameters requested are the forum object and the grade, of course. After validating a few parameters, this function calls the core method grade_update() defined in the gradelib.php

Both functions are called when the instances are created, updated or deleted. Covering every possible scenario where the grading method would be changed/stored.

There is also a function called hsuforum_reset_gradebook() that is in charge of deleting every grade related to Open forum, given a specific course ID. After executing a query to bring all forums in the course, hsuforum_grade_item_update() will be called once for each one of them.

Ratings

Ratings are used in the same way that core forums do. These ratings are aggregated to produce a grade for that activity for the student being rated. A student's posts in Open forums can be rated by his partners with those ratings being averaged later to produce a grade.

Manual Grading

Open forums can be manually graded via gradebook as any other plugin that support grades. The real deal here is that Open Forums also support advanced grading. Unfortunately, they cannot be graded with the core grader as assignments do. We need the help of the Open Grader plugin (see more: https://github.com/blackboard-open-source/moodle-local_joulegrader).

Open Forums Integrated With Open Grader

Open forums supports Moodle advanced grading methods as Rubrics, Marking guides and any other grading plugins developed by the community, but we need the Open Grader to grade the posts.

Workflow:

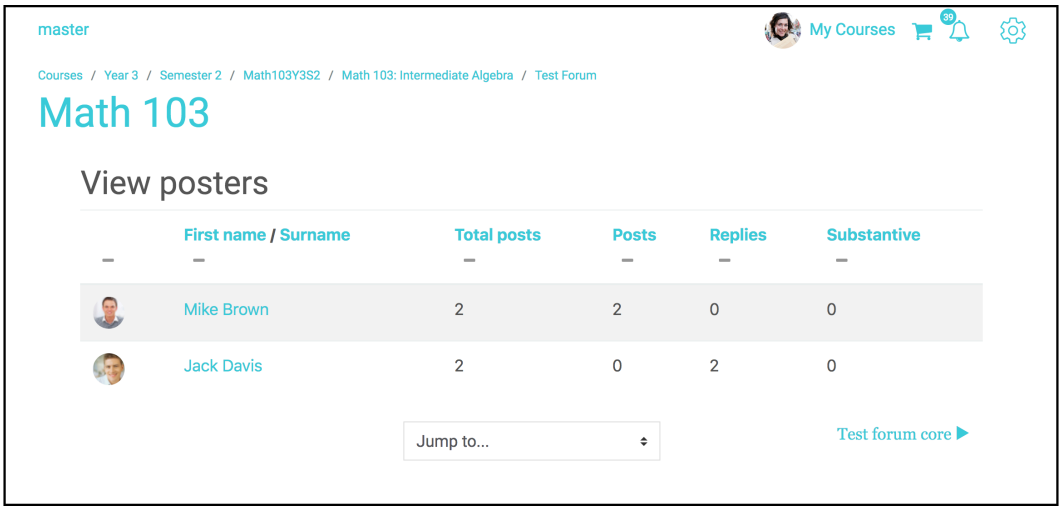
- A: When editing the forum settings, there is a field in the "Grade" section called "Grading method". The teacher can define which kind of grade (value or scale) will be calculated after selecting some advanced grading method to use.
- B: The teacher now has to navigate to the Forum admistration => Advanced grading and fill the form with the definition of the method to use.
- C. There is a function called hsuforum_add_instance() in the lib.php that will handle the creation of both grade_items and grade_grades records.

- D: If the teacher wants to edit the Forum and change the grading method, he is able to do it without problems. The only condition to be met is that every posts in that forum must be waiting for grading. There is a function called hsuforum_update_instance() that will handle the required update in the tables.
- E. Assuming that local_joulegrader is now installed, the instructor can access the Open Grader from the course administration and provide a grade for the posts of the forum.
- F. The Open Grader will handle the grading plugin and a grade will be provided to the forum.
- G: Now the teacher will be able to access the gradebook and review all the grades for the students if he needs it.

Open Forums features: View Posters

This document aims to give a good understanding of one of the key features of this activity module, the **view posters** feature.

User UI

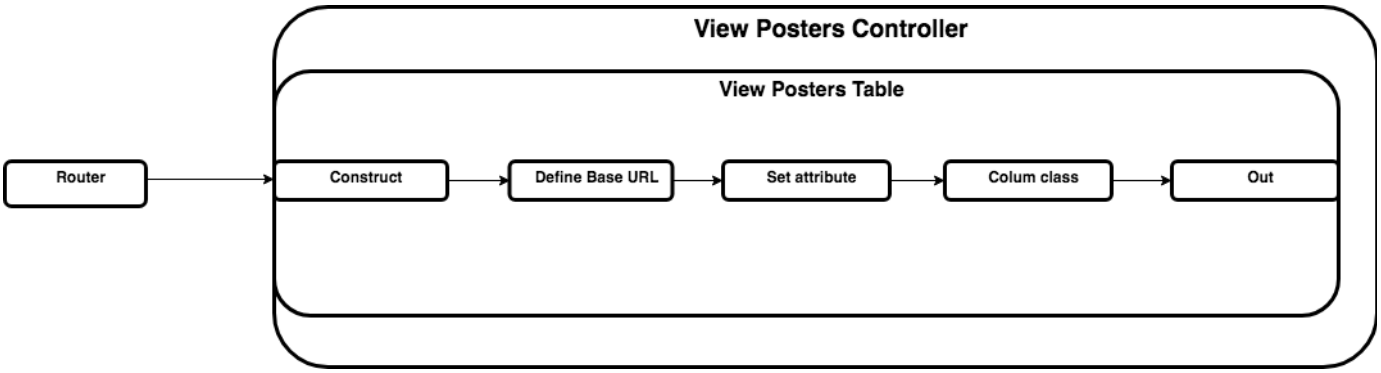


User UI has no restrictions regarding capabilities, so once the user reaches the activity module, it will be possible for them to access the "View posters" page.

It is possible to access users' profiles and rearrange the UI table.

How it works

The entire functionality is held in the **posters_controller.php** file. This controller is responsible of rendering the UI components which means calling the class that builds up the posters table.



This functionality takes advantage of one of the features from Moodle to build tables. The result is a type of table that is familiar to the user.

- **View Posters Table** is the main character for the **View Posters** feature. It's an extended class from **table_sql** and inside its **construct** method, it handles the table setup. This includes giving column names and setting what queries should run to get the number of posts per user (standard methods provided by Moodle). There is only a specific function for a column, which brings a user picture.
- After that, inside the controller the URL is defined, attributes are set and the user pic column class is set.
- A page header is configured.
- Finally the table is rendered.