



Submitty API Refinement

Cameron Peterson

Introduction



Submittity is an open-source course management, assignment submission, exam and grading system created by the Rensselaer Center for Open Source Software (RCOS), at the Department of Computer Science at Rensselaer Polytechnic Institute.

Goals



Upload JSON



Download JSON



Student API



Requirements for Upload



- Accept JSON
- Fail gracefully
- Support most of the parameters
- Be usable by both API and UI

Requirements for Download



- Produce JSON
- Fail gracefully
- Support most of the parameters
- Be usable by both API and UI

Requirements for Student API



- Be secure
- Allow instructors to use
- Give correct information

- Backend written in PHP
- Twig for UI
- Tested using Cypress written in JS

Proposed Schedule



Winter Quarter

Weeks 1-4:

- Create function to upload JSON to create gradeable
- Create test for the function

Weeks 5-8:

- Create UI to accept JSON
- Create function to Download JSON

Proposed Schedule



Spring Quarter

Weeks 1-4:

- Create API route to save/send the JSON file.
- Create UI button to save to JSON.
- Create tests for the UI.

Weeks 5-8:

- Catch up time, or other refinements.
- Weeks 1-2's items might take longer than expected, so accounting for that.
- Start thinking about poster.
- Prepare final poster and report.

Actual Schedule



Actual Schedule

Winter Quarter

- Create API for both Upload and Download
- Create UI for both Upload and Download
- Create tests for everything

Spring Quarter:

- Allow Student access to the API
- Create endpoint for Submissions
- Create endpoint for Requesting grades
- Create tests for everything

Challenges



1. Refining the API
2. Working with outside contributors
3. Security
4. Learning PHP Routing using Symfony
5. Legacy Code – Refactor Downloads

Upload UI



New Gradeable

Upload Gradeable JSON

[Need help with creating/editing a gradeable?](#)

General Rubric Grader Assignment Dates

Title

What is the title of this gradable?
(Required)

Assignment URL

What is the URL to the assignment instructions? (shown to student)

Unique ID

What is the unique id of this gradeable? (e.g., hw01, Lab_12, or midterm)

Gradeable Type
[What is the type of the gradeable?](#)

- (Required)
- Checkpoints (simple data entry: full/half/no credit)
 - Numeric/Text (simple data entry: integer or floating point and/or short text strings)

New Gradeable

Upload Gradeable JSON

[Need help with creating/editing a gradeable?](#)

General Rubric

Upload JSON for Gradeable

Close

Title

What is the title of this gradable?
(Required)

No file chosen

Assignment URL

What is the URL to the assignment instructions?

Unique ID

What is the unique id of this gradeable?

Gradeable Type

[What is the type of the gradeable?](#)

- (Required)
- Checkpoints (simple data entry: full/half/no credit)
 - Numeric/Text (simple data entry: integer or floating point and/or short text strings)
 - Students will submit one or more files by direct upload to the Submitty website
 - Students will submit by committing files to a version control system (VCS) repository
 - TA/Instructor will (bulk) upload scanned .pdf for online manual grading

Submit

Download UI



Edit Gradeable

Download Gradeable JSON

[Need help with creating/editing a gradeable?](#)

General

Submissions/Autograding

Rubric

Grader Assignment

Dates

Title

What is the title of this gradable?

Bulk Upload Scanned Exam

Assignment URL

What is the URL to the assignment instructions? (shown to student)

(Optional)

Unique ID

What is the unique id of this gradeable? (e.g., hw01, Lab_12, Or midterm)

bulk_upload_test

Gradeable type: TA/Instructor will (bulk) upload scanned .pdf for online manual grading

Will any or all of this assignment be manually graded (e.g., by the TAs or the instructor)?

Yes No

Should grade inquiries be enabled for this assignment?

Yes No

Should grade inquiries be associated with a specific component of the rubric?

Yes No

Is there a discussion component for this assignment?

Yes No

Team Settings: This is an individual assignment

Student API



```
if ($graded_gradeable->hasActiveVersion()) {
    $gradeable_version = $graded_gradeable->getAutoGradedVersions()[$graded_gradeable->getActiveVersion()] ?? null;
    if ($gradeable_version !== null) {
        // Gets arrays, and cleans empty arrays
        $testcase_array = array_filter(array_map(function (AutoGradedTestcase $testcase) {
            $testcase_config = $testcase->getTestcase();
            if ($testcase->canView()) {
                return [
                    'name' => $testcase_config->getName(),
                    'details' => $testcase_config->getDetails(),
                    'is_extra_credit' => $testcase_config->isExtraCredit(),
                    'points_available' => $testcase_config->getPoints(),
                    'has_extra_results' => $testcase->hasAutochecks(),
                    'points_received' => $testcase->getPoints(),
                    'testcase_message' => $testcase_config->canViewTestcaseMessage() ? $testcase->getMessage() : ''
                ];
            }
        } else {
            return [];
        }
    }, $gradeable_version->getTestcases()));
}

return JsonResponse::getSuccessResponse([
    'is_queued' => $graded_gradeable->isQueued(),
    'queue_position' => $graded_gradeable->getQueuePosition(),
    'is_grading' => $graded_gradeable->isGrading(),
    'has_submission' => $graded_gradeable->hasSubmission(),
    'autograding_complete' => $graded_gradeable->isAutoGradingComplete(),
    'has_active_version' => $graded_gradeable->hasActiveVersion(),
    'highest_version' => $graded_gradeable->getHighestVersion(),
    'total_points' => ($graded_gradeable->hasActiveVersion() ? $graded_gradeable->getTotalPoints() : null),
    'total_percent' => ($graded_gradeable->hasActiveVersion() ? $graded_gradeable->getTotalPercent() : null),
    'test_cases' => $testcase_array
]);
```

What's Next?



- Gradeable Creation API is finished
- Student API can be expanded
- Testing is always needed