# Canvas++ Programmer’s Documentation
# March 10, 2023

**Authors: Ethan Aasheim and Juan Rios**

## 1. Introduction

The Programmer’s Documentation describes the important aspects of how the Canvas++ system functions at a high level. This provides any person(s) who wants to make future modifications to the source code with the guidance to do so.

The following document will provide an overview of the source files. Along with this, it provides a rundown of the main functions in the code.

Source code can be found in this GitHub repository.
https://github.com/johnriverz/CanvasPlusPlus
See User Documentation or the README for installation instructions.

Code is written mainly in HTML, vanilla Javascript, and CSS, although there are several API calls that use Canvas Live API’s.

## 2. Canvas++ Source Files

### 2.1. Canvas++.html

Creates the components of the system. The login div shows by default, which contains the sign-in form and buttons. After signing in, the authed div is displayed and the login div disappears, which contains tabs for each task the user can do as well as displays for each course the user is taking and the appropriate data.

2.2. Canvas++.css

Provides the styling for the extension user interface. The styling is arranged mostly by the module to which the styling applies.

2.3. Canvas++.js

Hosts the main code and functions for the system, and coordinates with the additional .js files: login.js, assignments.js, notifications.js, and grade_calculator.js.

The functions reloadCourseIds(), reloadAssignments(), and reloadNotifications() populate the data structures for each kind of data respectively. loadAllCourses() rerenders the application to display new data, and the function itself calls loadCourse() for each course which makes the appropriate rendering calls to each of the tab modules.

loadCourse() calls loadCourseAssignments() from assignments.js, loadCourseNotifications from notifications.js, and loadCourseGrades() from grade_calculator.js to update each module's display.

2.4. assignments.js

2.4.1. getAssignments(post_url, options, token, course, course_id)

This function requests the Canvas API to get a list of upcoming assignments for a course. The post_url and options parameters specify the base URL and options for the API request. The token parameter is a Canvas API access token that is used to authenticate the request. The course_id parameter is the Canvas ID for the course. And this function returns a promise that resolves to an array of assignment names.

2.4.2. fetchAssignments(ass_url, options)

This function makes a GET request to the Canvas API to get the list of assignments for a specific course. The ass_url parameter is the URL for the API request. This function returns a promise that resolves to an array of assignment objects.

2.4.3. loadCourseAssignments(courseName, assignments)

This function takes in an assignments object containing a list of course assignments and displays them on the web page.

2.5. notifications.js

2.5.1. getNotifications(post_url, options, token, course_id)

This function takes the Canvas API URL, the request options, the user's access token, and the ID of the course to retrieve notifications. It makes a GET request to the Canvas API to retrieve the activity stream for the specified course and returns the JSON response as a string. If an error occurs, it will return an error message.

2.5.2. loadCourseNotifications(courseName, notifications)

This function takes in an announcements object containing a list of course announcements and displays them on the web page.

2.6. grade_calculator.js

2.6.1. getGrades(post_url, options, token, course_id)

This (obsolete) function takes the Canvas API URL, the request options, the user's access token, and the ID of the course to retrieve grades. It makes a GET request to the Canvas API to retrieve the activity stream for the specified course and returns the JSON response as a string. If an error occurs, it will return an error message.

2.6.2. loadCourseGrades(courseName, grades)

This function takes in a Grade object containing a list of course assignments and their respective scores and displays them on the web page.

**3. Extension Interface Code -** Code can be found under Canvas++.html and Canvas++.js:

The Canvas++ JavaScript file implements the interface functionality. It does so by calling functions from each tab's respective JS file. It uses the responses from these calls to populate the interface with a user's data.

**4. Assignments Tab** - Code can be found under assignments.js:

The Assignments Tab displays all assignments the student must complete in the next week. This is useful because Canvas does not have a way to filter assignments by date, and shows all assignments at once in a sometimes overwhelming display.

The module consists of a single function that takes in the ID of a course that a student is currently enrolled in, and outputs all assignments for that course. The function is run in a loop, however many times it needs to in order to successfully grab the data from each individual course.

**5. Notifications Tab -** Code can be found under notifications.js:

The Notifications Tab displays recent updates for courses that a student is currently enrolled in. It can display recent Announcements from a course to make it easy for a student to see any new information that a teacher may be trying to send to the students.

This is useful because many students do not receive alerts from Canvas when a new announcement is posted by a teacher. The Notifications Tab allows them to easily check to see if there has been any update from their teacher.

The module has an asynchronous function that takes in the ID of a course that a student is currently enrolled in, and outputs recent activity from that course. The function is run in a loop, however many times it needs to in order to successfully grab the data from each individual course.

The rest of the functions for this module handle the notification data that is passed in (a list of all notifications for a particular course). loadCourseNotifications() creates new div objects to populate the display, and openNotification() expands a notification panel to display the actual message when the panel is clicked on.

**6. Grade Calculator Tab -** Code can be found under grade_calculator.js:

The Grade Calculator Tab displays all assignments for all courses that a student is currently enrolled in, along with the score, maximum possible score, and percentage of those assignments. It also displays the overall grade for that course at the top of the panel.

Users can input hypothetical scores for each assignment, and the percentage for that particular assignment as well as the overall grade will change in the display according to the user input and the weight of the assignment's score.

The module has an (obsolete asynchronous) function that takes in the ID of a course that a student is currently enrolled in, and outputs recent activity from that course. The function is run in a loop, however many times it needs to in order to successfully grab the data from each individual course. Unfortunately, Canvas API will not return grade data (likely a security feature), so in this version of Canvas++ we populate the display with fake assignments and fake scores.

The rest of the functions for this module handle the notification data that is passed in (a list of all notifications for a particular course). loadCourseNotifications() creates new div objects to populate the display, and openNotification() expands a notification panel to display the actual message when the panel ops are clicked on.

**7. Major Data Structures**

There are four major dictionaries used by the system: one for courses, one for assignments, one for notifications, and one for grades. They are each populated by a function in Canvas++,js with data fetched via Canvas API calls.

The assignments, notifications, and grades dictionaries use course IDs for their keys, with the value of each being a list of all the appropriate data. The data that is rendered to the application is taken directly from these lists, which are each rendered one at a time.

**8. Known Issues**

We try to write the grade calculator function, but when we try to use the Canvas API to get the grade, we can not find any data in the Canvas API which is about the graded return. We suspect that this is a security measure Canvas implements to protect student privacy.

Another prominent issue was the Canvas servers' security measures. We found that we were not able to make any API requests without first prepending the URL request with a CORS proxy. Simply put, the proxy adds an *Access-Control-Allow-Origin* header to the request. We would then also need to gain temporary access to this demo CORS server in order to use it.

**9. References**

Canvas++ uses Canvas Live APIs that can be found here:
https://canvas.instructure.com/doc/api/live#!/assignments.json

## 10. Acknowledgements

Canvas++ was created by Ethan Aasheim, Spike Chen, Alder French, Juan Rios, and Blake Skelly at the University of Oregon for the course CS 422: Software Methodologies, taught by Anthony Hornof.