

Menu

1. Overview
2. Project Structure
3. Getting Start
4. Entrypoint
5. Widgets
6. Brightspace APIs Used

1. Overview

The CetlBarWidget is a set of dashboard embedded as D2l widget using iframe. It provides quick access of course contents to students and instructors. This project consist of three main php files EntryPoint.php, student.php and instructor.php. EntryPoint.php deals with D2l Authentications and redirect users to different widgets according to their roles.

2. Project Structure

The project consists of these following folders/files:

- config: Storing appId, appKey, host, etc.
- css: bootstrap library
- js: js libraries
- lib: Brightspace API PHP libraries
- resources: Images
- widgets: The widgets that is actually shown on the website.
- doRequest.php: A wrapper api of Brightspace APIs(mainly used by ajax in this project).
- cacert.pem: Used for authentication when getting professor information.
- EntryPoint.php: The endpoint that first handles D2l Authentication and then redirect to different widgets.

3. Getting Start

Server Side:

To use this project, copy it to a server and setup the php running environment. One would also need to get a appId and appKey from the D2l Platform and replace the appId and appKey in the config.php file.

If the project is running on different location(host), you would also need to change the URL,host and port in the config.php file. Also, make sure you change the corresponding URLs in the EntryPoint.php file.

On D2l Course Website:

Please refer to the "Configuration and Setup" part in the user document to setup the widget.

4. Entrypoint

When a user visit the iframe of this widget, the browser send a get request to the URL that is set in the iframe. The URL should be pointing to the EntryPoint.php file. The entrypoint will first do a Brightspace id-key authentication as it is described in: <https://docs.valence.desire2learn.com/basic/auth.html>.

Before it redirects the browser for authentication, it also gets the url query parameters in the GET request and passes them back by putting them into the x_state parameter. Without this step, all query parameters will be lost after the authentication process.

After the user completed the authentication, he/she will be redirect back to the entrypoint(this redirection URL is passed in the previous step). Then the entrypoint will do two api calls to the Brightspace api. It first get the user id and then use it to get the role id of the user to determine which widget this user should be redirected to. It also pass the query parameter stored in the x_state parameter to the widget.

5. Widgets

After the authentication is done in the entrypoint, it redirect browser to different widgets according to the user's role.

Note: To add new widgets, just simply setup different redirection conditions to your new widget and send some query parameters in the iframe as conditions.

Student Dashboard:

If the user is a student, he/she will be redirect to the student.php file. The OrgUnitId(ou) must be passed through the iframe url. The student dashboard contain three functions that do Brightspace api calls using ajax.

- emailprof: Get the professor information of the course that the user is visiting.
- gradevalue: Get the user's grade value in this course.
- grade: Get the user's grade in this course.

Instructor Dashboard:

If the user is an instructor, he/she will be redirect to the instructor.php file. The OrgUnitId(ou) must be passed through the iframe url. The instructor dashboard contain three functions that do Brightspace api calls using ajax.

- EnrollmentNumber: Get the classlist of this course
- checkCourse: Check if the course is currently active or not.
- checkOrgid: Get the OrgId of this course.

6. Brightspace APIs Used

This project uses several Brightspace apis and it take the advantage of the doRequest.php file(a wrapper api of the Brightspcae api) to send the request. The detailed information of the following api can be found on:

<https://docs.valence.desire2learn.com/>.

- GET /d2l/api/lp/(version)/users/whoami
 - Retrieve the current user context's user information.
 - **Parameters:** version (D2LVERSION) – API version.
 - **Return:** This action returns a WhoAmIUser JSON block for the current user context.
- GET /d2l/api/lp/(version)/enrollments/orgUnits/(orgUnitId)/users/
 - Retrieve the collection of users enrolled in the identified org unit.
 - **Parameters:** version (D2LVERSION) – API version, orgUnitId (D2LID) – Org unit ID.
 - **Return:** This action returns a paged result set containing the resulting OrgUnitUser data blocks for the segment following your bookmark parameter (or the first segment if the parameter is empty or missing).
- GET /d2l/api/le/(version)/(orgUnitId)/grades/values/myGradeValues/
 - Retrieve all the grade objects for the current user context assigned in a particular org unit.
 - **Parameters:** version (D2LVERSION) – API version, orgUnitId (D2LID) – Org unit ID.
 - **Return:** This action returns a JSON array of GradeValue blocks.
- GET /d2l/api/le/(version)/(orgUnitId)/grades/
 - Retrieve all the current grade objects for a particular org unit.
 - **Parameters:** version (D2LVERSION) – API version, orgUnitId (D2LID) – Org unit ID.
 - **Return:** This action returns a JSON array of GradeObject blocks.
- GET /d2l/api/le/(version)/(orgUnitId)/classlist/
 - Retrieve the enrolled users in the classlist for an org unit.
 - **Parameters:** version (D2LVERSION) – API version, orgUnitId (D2LID) – Org unit ID.
 - **Return:** This action returns a JSON array of ClasslistUser data blocks. Note that the data blocks returned here respect the org unit's privacy settings with respect to the UserName, OrgDefinedId, Email, FirstName, LastName and the contents of the DisplayName properties (typically, the display-name gets formed from the user's first and last names, and privacy settings govern the visibility of those items).
- GET /d2l/api/lp/(version)/courses/(orgUnitId)
 - Retrieve a course offering.
 - **Parameters:** version (D2LVERSION) – API version, orgUnitId (D2LID) – Org unit ID.

- **Return:** This action returns a CourseOffering JSON block with the provided course's information.
- GET /d2l/api/lp/(version)/orgstructure/childless/
 - Retrieve all org units that have no children.
 - **Parameters:** version (D2LVERSION) – API version
 - **Return:** This action returns a paged result set containing the resulting OrgUnitProperties blocks for the segment following your bookmark parameter (or the first segment if that parameter is empty or missing).