HLLN TECH MANUAL

WordPress Destiny Integration

* Introduction
* HLLN WordPress website
* Summary of plugin capabilities
* Setting up the plugin
* Data field map between WordPress and Destiny
* Data requirements within Destiny
* Data requirements within WordPress
* Support and development
* Recommendations
* Challenges
* Required Skills
* Future Development
* Contacts

eClass Destiny Intergration

* Summary of plugin capabilities
* Setting up the plugin
* eClass Requirements
* Workflow
* Support and development (Technical/non-technical)
* Common eclass issues
* Required Skills
* Future Development
* Contacts

Eclass Course Settings

* General Course Settings
* Course Completion Settings
* Attendance Tracking
* Assignment/Grade Item Settings

# WordPress Destiny Integration

https://hlln.dev.yorku.ca/

The Destiny Scraper plugin handles all course content displayed on the HLLN website. Information that can be controlled or manipulated through the plugin include:

* All data displayed on program pages
* All data displayed on the program directory
* Program categories (filters)
  + Personas
  + Subject areas
  + Certificate types
  + Custom programs

HLLN WordPress Website

The new HLLN WordPress website has been developed by Sandbox Solutions. The website refresh was a much-needed change to keep the department up to date with e-commerce. The design incorporates many features needed by HLLN but needed to abide by the YorkU guidelines and restrictions by the web developers. Communication with Sandbox Solutions went through Jacqueline Taylor, the Web Team Lead. The main web developer was Matt Graham, the Senior Web Developer. Oversite by York University was mainly handled by Jason Miller. Sandbox created a course template, and the plugin dynamically populates the page on demand with the most recent data retrieved from Destiny. The plugin relies on editing prebuilt components on the template page.

Summary of Plugin Capabilities

The Destiny Scraper plugin is a WordPress plugin designed and developed specially for HLLN@YorkU’s needs. The plugin helps automate Destiny integrations with WordPress. Using Destiny Webservices, the plugin scrapes Destiny for courses, sections, and certificates. The plugin also provides a number of short codes to automatically populate WordPress templates with Destiny Data for client view.

Destiny Scrapper Settings

The Destiny Scrapper relies on the admin settings for data collection. Proper set up of the plugin is critical for the plugin to function correctly.

The plugin settings page can be found on the collapsible menu on the left of the WordPress Admin view.

A screenshot of a computer

Description automatically generated with medium confidence**Error Alert Email** – In event of a plugin error, HLLN is notified through the specified email with information/instructions

**Destiny One Domain** – Crucial for requesting data from the Destiny One System

**Destiny One Public View Domain** – Required to redirect students to Destiny One for registration or application to programs through the website

**Destiny One Program Office** – Required to retrieve all programs offered by HLLN in the Destiny System. This is a Destiny field

**Destiny One Username/Password** – Required to logon to Destiny for program data retrieval

Data field map between WordPress and Destiny

|  |  |  |  |
| --- | --- | --- | --- |
| **Destiny** | | **New Website** | |
| **Level** | **Field** | **Page** | **Field** |
| Course | Course Title | Program page | Program Title |
| Course | Video Caption | Program page | Banner Image |
| Course | \*Official Description (Internal) | Course directory | Enrollment Details |
| Course | \*Official Description (Internal) | Course directory | Course Introduction |
| Course | Course Description (Public) | Program page | Program Overview |
| Course | Course Outline | Program page | Program Content |
| Course | Additional Information | Program page | Job Titles |
| Course | Learner Outcomes (Public) | Program page | What you will learn |
| Course | Notes | Program page | How you will benefit |
| Course | Prerequisites (Public) | Program page | Entry Requirements |
| Course | Recommendations (Public) | Program page | Certificate of Completion |
| Course | Testimonials (Public) |  |  |
| Instructor | Biography | Program page | Faculty |
| Section | Scheduling | Program page | Section Dates |
| Section | Instruction Methods | Program page | Section Instruction Method |
| Course | Associated Application | Program page | Section Enrollment Button |
| Fees | Tuition Amount | Program page | Price |
| Certificate | Video Caption | Certificate page | Banner Image |
| Certificate |  | Certificate page | Certificate Title |
| Certificate | User Defined Field 1 | Certificate page | How you will benefit |
| Certificate | \*Official Description | Course directory | Enrollment Details |
| Certificate | \*Official Description | Course directory | Course Introduction |
| Certificate | \*Official Description | Certificate page | Program Overview |
| Certificate | \*Official Description | Certificate page | Earning Process |
| Certificate | \*Official Description | Certificate page | EELM |
| Certificate | \*Official Description | Certificate page | How you will benefit |
| Certificate | \*Official Description | Certificate page | Entry Requirements |
| Certificate | \*Official Description | Certificate page | How to apply |
| Bundle | Amount | Certificate page | Price |
| Bundle | Description | Certificate page | Price Description |

\*Destiny One field that needs to be split into different sections of the website. Format by placing the delimiter [split] between the separate texts within the Destiny text box. Example:

Some course enrolment details.[split]Some course introduction.

Data requirements within Destiny

Text in Destiny should be free from headings, since the structure of the page is already set.

* Do not include any h1 or h2 tags.
* If section subheadings are required, stick to bold, italic, and underlined text.

If required, paragraphs, lists and tables are permitted

In the program directory, there are further limitations:

* The two sections (enrolment details and course introduction) come from the same Destiny textbox. In order to split the data in two sections the text must be separated using the delimiter “[split]”
* Enrolment details is limited to **115** characters, and striped of all formatting tags.
* Course introduction is limited to **115** characters, and striped of all formatting tags.

Destiny Certificates do not provide as many data fields as courses. Similar to the program directory, sections are separated using the delimiter “[split]”

* 8 sections include: Enrollment Details, Course Introduction, Program Overview, Earning Process, EELM, How you will benefit, Entry Requirements and How to apply
* Separate each section using the delimiter “[split]”

Data requirements within WordPress

Destiny One is not designed to be a complete customer facing website. Populating the HLLN website using Destiny has some draw backs. A few pieces of data cannot be housed in Destiny One.

**Program Images:** All program images must be uploaded and stored in the Wordpress Media Library to maintain original size and quality. The images need to be linked to a program in Destiny. To achieve this, the link for each program image should be placed in the video caption section of the Destiny course profile. To update program images, upload and save a new image in the Media Library and update the link in Destiny. If no image link is provided in Destiny a default image will be displayed.

A screenshot of a computer

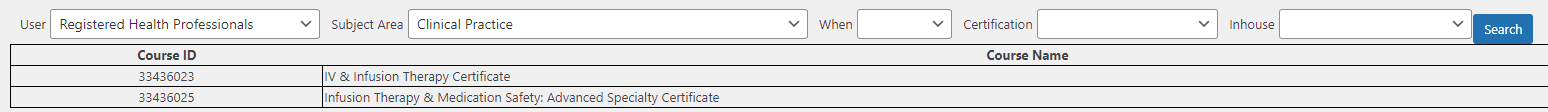
Description automatically generatedA screenshot of a computer

Description automatically generated with low confidence**Program Filters:** In order for the directory to function, all programs need to be tagged with applicable filters. For all new programs added to Destiny One, a WordPress admin must provide tags through the plugin settings page.

2 Check off all appropriate tags and save.

3 Enter the Destiny Course Id to retrieve the program.

Tags can be searched to find a list of matching courses.



Support and development

The HLLN WordPress site was designed and built with Sandbox Solutions. All templates, formatting and styling was controlled and handled by Sandbox Solutions following requirements by YorkU. The plugin provides shortcode to insert plain content onto prebuilt templates with formatting. In some cases, such as the accordions on the course pages, javascript injection is required in order to maintain proper block function. Sandbox provides a limited amount of coverage for changes or fixes to the website. Upkeep of the plugin is entirely the responsibility of HLLN.

Changes to the webpages can be controlled through the plugin to some degree within the given templates.

The Destiny Scrapper plugin has been developed internally by HLLN and is the responsibility of HLLN. Maintenance will not be carried out by York IT services.

Support during the development process was provided by the Information Technology Coordinator at the School of Continuing Studies. The support provided included advise on the plugin design and guidance for navigating the Destiny system.

Recommendations

The plugin is limited by the Destiny system as well as the HLLN website templates. The piece that needs to be improved is the settings page of the plugin. The program filter tag “editor” was built quickly on short notice. It is functional but clunky. It could be separated from the main settings page and made more user friendly.

The plugin event log is also quite primitive and built to function during development. It should be properly built for staff use.

Challenges

There are limited fields available in Destiny. In order to house all content for the website, some fields need to be combined and separated by the scraper code. Many fields are not being used as they were designed. Please refer to the content map to achieve proper integration.

Required Skills

Plugin maintenance requires programming skills. The plugin is written mostly in PHP, with a little HTML needed for formatting. Basic knowledge of the WordPress environment is recommended. A good understanding of the Destiny system is needed, although it is difficult to have this knowledge without working with the system. WordPress guides are available online, and community discussions are very helpful in learning about WordPress.

Future Development

Switching system providers should not require too much change to the system. As long as the vendor provides access to their API, which allows data retrieval, minor changes will be required. If the API allows course and course bundle retrieval, only the API calls and the data transformation need to be changed. Depending on the availability of fields in the SIS system, the formatting might also need to be worked on.

From the vetting process of vendors to date, AmpEducator provides an API with very similar capabilities to Destiny One.

Contacts

[Drew Edwards](mailto:drew.edwards@sandboxsoftware.ca)  Director of Web Services Sandbox Solutions

[Jacqueline Taylor](mailto:jacqueline.taylor@sandboxsoftware.ca) Web Team Lead Sandbox Solutions

[Matt Graham](mailto:matt.graham@sandboxsoftware.ca)  Senior Web Developer Sandbox Solutions

[Riley Edwards](mailto:riley.edwards@sandboxsoftware.ca)  Digital Project Assistant Sandbox Solutions

[Jason Miller](mailto:jmiller4@yorku.ca)  Digital Specialist York University

[Paul A Elo](mailto:pae@yorku.ca) Information Technology Coordinator York University

# eClass Destiny Integration

Summary of plugin capabilities

The Destiny Student Management Block is a custom plugin for HLLN’s integration with Destiny One. The plugin aims to automatically manage students in eClass. Plugin options allows for a more manual approach through reports and manual approvals. Tasks handled by the plugin include; getting class lists from Destiny, managing student access to courses, tracking students’ progress, calculating course completion, exporting final results to Destiny, and resetting the course.

Setting up the plugin

Plugin set up must be done once for each different course. The plugin is designed to handle concurrent sessions. It is recommended to review settings between sessions.

A screenshot of a computer

Description automatically generated with medium confidence

**General settings:** This must include the corresponding Destiny course number, section number and section id. An eclass cohort name must also be provided.

**Date Conditions:** A session start date (locks automatic adding of students to course page), a session export date (required, date to export records to Destiny), a refresh date (optional, prepares course for next cohort)

**Activity Completion:** Select all assignments/quizzes and activities to monitor for final completion calculation.

**Approval:** Toggle auto enrolments and grade approval. If disabled, manual approval required.

**Notifications:** Enable and specify email addresses for plugin notifications to course approver and editor.

**Destiny Account Information:** Provide the correct login information for the Destiny System. Default set by plugin.

eClass Requirements

Attendance – There must be exactly one assignment named “Attendance” in order to properly extract student records. The assignment must be selected in the Activity Completion section of the plugin settings.

Cohort – For each course (not session), a cohort must be created. This allows students to be automatically added or dropped, when appropriate. The cohort created must be specified in the general section of the plugin settings.

Plugin settings – All plugin settings must be reviewed before each session to ensure proper function of class automation.

Workflow

**Initial setup –** The plugin can be set up after the proper creation of a program page with all required items. Plugin settings must be set after adding the tracking plugin to the page.

**Student access –** If automatic student access is allowed through the settings, students will automatically be added and granted access to the course page upon completing their payment and registration through Destiny. If manual approval is required, a staff member must approve access for each student through the plugin student approval page.

**Grading and Attendance –** As the course progresses, students must be graded on all required assignments and attendance must be recorded.

**Grade Approval –** If automatic grade approval is allowed, student records for the course will be exported to Destiny on the specified course end date. Otherwise, staff approval is required for each student for grades to be exported.

**Refresh –** If automatic refresh is enabled, the plugin will increment the section id and dates, remove all previous students and begin the plugin workflow again (Staff are strongly recommended to review all settings upon refresh). Otherwise, staff are required to edit plugin dates and remove students manually from the cohort.

**Notifications –** Email reminders are sent to the course approver and course editor at all steps to help with the timely completion of required steps.

Support and development

The plugin was built for and by HLLN. All functions are controlled by HLLN.

Eclass is controlled by the York IT team. All plugins are added by the IT team and must go through an approval process. Eclass updates are done monthly. Any updates to the plugin should be provided to York IT in advance. Upkeep of the plugin is the responsibility of HLLN.

Advice and guidance through the development process was provided by Patrick, the Director of IT Innovation & Academic Technologies at University Information Technology, York University. Patrick was also the contact through which the plugin went through for approval.

Infromation about the Destiny system was provided by Paul Elo, the Information Technology Coordinator.

Common eClass Issues

eClass course completion is not ideal for HLLN. Enabling completion tracking forces a progress bar on student dashboards. York IT does not allow the bar to be removed as it impacts all York departments. The progress bar has caused a lot of confusion among students and frequent emails asking for clarification. The plugin tracks completion without the built in tracker. The built-in completion report is updated on a schedule. Many times, the built-in completion report was not up to date when staff needed to review the student reports. The plugin completion report is generated on demand, so it is always up to date.

The attendance module has caused problems for HLLN instructors. The module also requires a lot of work for setup and maintenance. Furthermore, the module is unable to easily show if a student has attended the first session of the course. To get around these problems, an assignment has been used to let the instructors record attendance via a rubric. The plugin calculates the number of days attended as well as whether the student attended the first session of the course.

Required Skills

Plugin development and maintenance requires programming skills. The plugin is written mostly in PHP, with a little HTML needed for formatting.

It is important to learn and understand the Moodle system. The Moodle developers guide is comprehensive and a fast way to learn about Moodle components. Community discussion posts are also helpful to navigate problems and complex Moodle components.

Since the plugin bridges eClass and Destiny, it is important to understand how Destiny works and how the systems interact. The Destiny API guide helps with understanding what fields can be accessed and manipulated.

Future Development

Improvements can be made to the notification system of the plugin.

Switching away from Destiny can be a bit tricky. All of the communication via API would need to be changed for the new vendor.

The plugin relies on an integration on between Destiny One and York University to enroll students into courses. The integration, built by Continuing Studies, automatically creates and links a PPY to Destiny student accounts. The creation of the PPY results in the student automatically existing in the eClass database, with a default password. The plugin can automatically enroll students since they exist in eClass. Without the integration, the student LMS account will not be linked to their SIS account. The plugin will not be able to enroll students as easily and as seamlessly as planned.

In the ideal case, the new vendor would provide the same, or very similar, API endpoints. The plugin uses a few basic endpoints:

* Login to a session from the SIS
* Retrieve the class list from the SIS
* Check if each student has paid in full for their course
* Update the student record in the SIS (attendance and final grade)
* Log out

Contacts

[Paul A Elo](mailto:pae@yorku.ca) Information Technology Coordinator York University

[Patrick Thibaudeau](mailto:thibaud@yorku.ca) Director, IT Innovation & Academic Technologies York University