

Steps for Installing Course Management

- Preparing for the Installation
- Creating Content Types and Views for Courses
 - OPTION 1: Making use of the feature in Course Management package (If you want to easily install content type and view for Courses...)
 - OPTION 2: Creating content types and views manually (if you need to configure them differently)
- Enable Migrate Courses module (in the package)
- Enable Courses UI module (in the package)
- Set URL Alias Patterns
- Manually Create Taxonomy Terms
- Check Permissions
- First Import of Courses
- Copy template files in the package to your 'template' folder.
- Copy CSS code in the package to your 'global.css' file.
- Set Cron jobs (Important)
- Set Views Appropriately
 - (Example) Create a View for Courses
 - (Optional) SLLC-specific Configurations: in a case where a website has many courses page
 - (Optional) ARHU-specific Configurations
 - (Optional) Requirements from English

Preparing for the Installation

- Repository of Course Management Package: <https://git.apollo.arhu.umd.edu/myeong/course-management-module>
- Modules required: enable them
 - Entity API (>= v.1.3)
 - Field Collection (most recent "dev" version. The stable version does not have required functions.)
 - User Reference
 - References
 - Migrate
 - Elysia Cron
- Check the permissions for the modules after enabling them.
- Module enabling order is very important, so follow the steps as follows (if you enable modules and features randomly, there will be conflict in the database).

Creating Content Types and Views for Courses

OPTION 1: Making use of the feature in Course Management package (If you want to easily install content type and view for Courses...)

- Copy "course_management" folder in the package to your "modules" folder (where features are located)
- Enable the feature "course management"
 - **drush en course_management**
 - **drush fr course_management**
- If you enable the feature, content types (Course, Syllabus), a field collection (Section), and the views for Courses will be set up.

OPTION 2: Creating content types and views manually (if you need to configure them differently)

1. Create a content type called "Course"
 - a. Machine name is "course"

Name *
 Machine name: course [Edit]
The human-readable name of this content type. This text will be used in URLs and labels. It can contain letters, numbers, and spaces. This name must be unique.

- b. Set Title Field Label as "Course Code"

Submission form settings

Course Code

Publishing options

Published

Display settings

Display author and date information.

Add this settings

Compare revisions

Title field label *

Course Code

Preview before submitting

☐ Disabled

☒ Optional

☐ Required

c. Display settings: Hide the author and date

Submission form settings

Course Code

Publishing options

Published

Display settings

Don't display post information

☐ Display author and date information.

Author username and publish date will be displayed.

d. Fields (machine names should be same as follows)

LABEL	MACHINE NAME	FIELD TYPE	WIDGET
+ Course Code	title	Node module element	
+ Title	field_title	Text	Text field
+ Credits	field_credits	Text	Text field
+ Grade Method	field_grade_method	Text	Text field
+ Description	field_description	Long text	Text area (multiple rows)
+ Extra Description	field_extra_description	Long text	Text area (multiple rows)
+ Course Details	field_course_details	Long text	Text area (multiple rows)
+ Permission Required	field_permission_required	Boolean	Single on/off checkbox
+ URL redirects	redirect	Redirect module form elements	
+ URL path settings	path	Path module form elements	
+ Sections	field_sections	Field collection	Embedded
+ Syllabus	field_syllabus_file	File	File
+ Tags	group_tags	Fieldset	fieldset collapsible
+ Department	field_department	Term reference	Select list
+ Level	field_level	Term reference	Check boxes/radio buttons
+ Term	field_term	Term reference	Autocomplete term widget (tagging)

e. Click into the "Edit" of "Sections" field and check if it is set "unlimited" in the "Number of Values" field. It should be checked as "Unlimited".

Sections field settings

These settings apply to the *Sections* field

FIELD VISIBILITY AND PERMISSIONS

☒ Public (author and administrators can
☐ Private (only author and administrators
☐ Custom permissions

NUMBER OF VALUES

Unlimited ▼

☒ Hide blank items

f. Display settings: follow this

FIELD	LABEL	FORMAT	
+ Title	<Hidden> ▼	Default ▼	
+ Grade Method	Inline ▼	Default ▼	
+ Credits	Inline ▼	Default ▼	
+ Sections	Above ▼	Field collection items ▼	Links: none View mode: Full content
+ Description	Above ▼	Default ▼	
+ Course Details	Above ▼	Default ▼	
+ Permission Required	<Hidden> ▼	Default ▼	
+ Department	Inline ▼	Plain text ▼	
+ Level	Inline ▼	Plain text ▼	
+ Term	Inline ▼	Plain text ▼	

2. Create a content type called "**Syllabus**": this is for embedding syllabus page for each faculty member.

a. The machine name is "syllabus".

Name *

Syllabus

Machine name: syllabus [Edit]

b. Set the fields as follows:

LABEL	MACHINE NAME	FIELD TYPE	WIDGET
+ Title	title	Node module element	
+ Instructor	field_instructor	User reference	Select list
+ Course ID	field_course_id	Node reference	Select list
+ Syllabus File	field_syllabus_file	File	File

3. Set up the views for the courses as you need.

Enable Migrate Courses module (in the package)

1. If you are working on ARHU multisite, don't worry. courses_ui module is already there. Otherwise, copy the folder from the package to your folder.
2. In the command line, enable "migrate_courses" module:

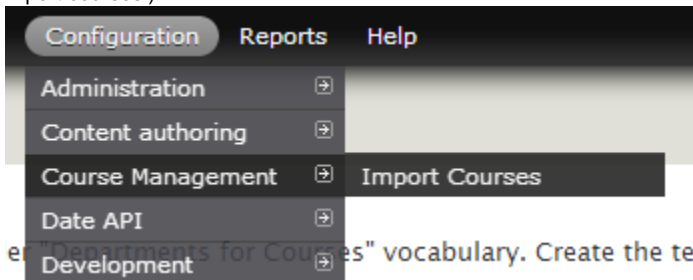
```
drush en -y migrate_courses
```

Enable Courses UI module (in the package)

1. In the command line, enable "Courses UI" module:

```
drush en -y courses_ui
```

2. Then, you can see Course Management menu under Configuration (This menu is only for the first migration purpose. After that, cron jobs import courses.)

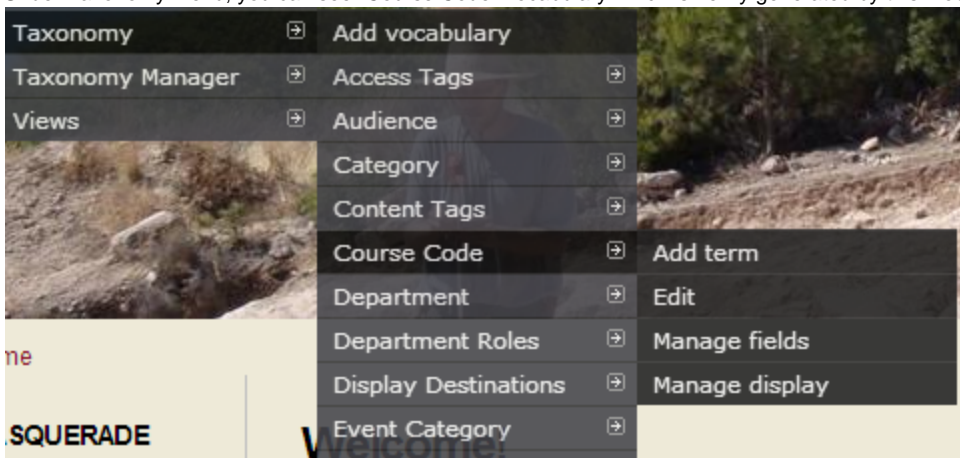


3. For now, "Import Courses" page shows only a text, since Taxonomy terms are not set yet: If you see this text, it is successful so far.

Import Courses +

In order to use Course Management, you need taxonomy terms under "Departments for Courses" vocabulary. Create the terms.

4. Under Taxonomy menu, you can see "Course Code" vocabulary which is newly generated by this module.



5. After you enable the module, you should be able to see these taxonomy vocabularies. These are automatically generated, and required to use the module (so, don't delete them).
 - a. Course Code
 - b. Term
 - c. Level
 - d. Department (this taxonomy term is required for Course Management package, but normally it already exists in most ARHU websites.)

Set URL Alias Patterns

In order to show correct URL for your courses, you need to set up a URL alias.

- Go to Configuration --> Search and metadata --> URL Aliases --> Patterns (Pathauto module is needed for this menu)
- Set your desired URL for Course content type.
- For example, Classics department's course URL is like this.

Pattern for all Course paths

academicprograms/courses/[node:title]

Manually Create Taxonomy Terms

- Taxonomy terms for "Course Code" are parameters for Courses UI module.
- You can put any 4-letter department names under the vocabulary.
- For example, if you want to pull "Korean" department's courses, you add a term "KORA" since it stands for Korean department. (You can check this in <http://ntst.umd.edu/soc/>)

Departments for Courses

Name *

KORA

Description



Check Permissions

- Field Permission
 - The permissions are already set in the module, so you don't have to do anything here – but just in case, check out the permissions.
 - Course and Section content type have fields that are regularly imported: these fields should not be modifiable by users – so need to set custom permissions for each field.
 - e.g. since "Description" field cannot be modified by users, the permissions were set as follows:

Field visibility and permissions

☐ Public (author and administrators can edit, everyone can view)

☐ Private (only author and administrators can edit and view)

☒ Custom permissions

PERMISSION	ADMINISTRATOR	DRUPAL ADMIN	SITE ADMIN	SITE EDITOR	EDITOR	FEEDS IMPORTER	AUTHENTICATED USER	ANONYMOUS USER
Create own value for field <i>Description</i>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Edit own value for field <i>Description</i>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Edit anyone's value for field <i>Description</i>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
View own value for field <i>Description</i>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
View anyone's value for field <i>Description</i>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>

First Import of Courses

- For the first import of courses, you need to use the interface (this is only for the first time import. After that, cron jobs are responsible for

updating them).

1. If you go to the "Configuration --> Course Management --> Import Courses" page, you see the page with Import button.

Import Courses

Select term(s) and department, and click "Import" button to import course data from Testudo.


Select a term *



A dropdown menu with a light gray background and a thin border. The top item is "- Select -" in a darker gray font. Below it are several terms in a standard black font: "2014 Fall", "2014 Summer", "2014 Spring", "2013 Winter", "2013 Fall", "2013 Summer", and "2013 Spring". Small upward and downward arrow icons are visible on the right side of the menu.

Select a term that you want to import.

Select a department *



A dropdown menu with a light gray background and a thin border. The top item is "- Select -" in a darker gray font. Below it is the text "KORA" in a standard black font. Small upward and downward arrow icons are visible on the right side of the menu.

Select a department that you want to import.

Import

2. Select a term and a department, click Import button.
(if you need to import multiple departments, select "ALL" in the department select box. It automatically appears when there are more than 2 departments)
3. If the Import is successful, a message indicates the success.

Import Courses +



Courses have been successfully imported!

- Processed 3 (3 created, 0 updated, 0 failed, 0 ignored) in 2.5 sec (73/min) – done with 'Courses'
- Processed 22 (0 created, 0 updated, 0 failed, 0 ignored) in 1.7 sec (758/min) – done with 'Sections'

Select term(s) and department, and click "Import" button to import course data from Testudo.

Select a term *

Select a term that you want to import.

4. If you see any other error messages, it means the import is not successful. In this case, you need to contact the developer or administrator.
5. If it was successful, you can see the imported contents.
6. If you need to import courses from multiple semesters, you need to repeat the above steps. The unit you can pull is only one semester.

Copy template files in the package to your 'template' folder.

- Copy these two files to your template folder located in the theme folder
 - `template/node--courses.tpl.php`
 - `template/views-view-table--courses--page.tpl.php`
- The first template file is for each Course page (generated by the content type).
- The second template file is for the course listing page (generated by the view).
- Clear cache. "drush cc all".

Copy CSS code in the package to your 'global.css' file.

- Example code of CSS is in the package.
- Copy the code to your global.css file.
- If you are using the feature in the package, the css code will be directly applied.

Set Cron jobs (Important)

- There are 2 cron jobs coded in the modules.
 - Need to set different period for Course Management since its computation load is high (using Elysia cron, etc).
1. Cron job in **courses_ui**
 - This cron job is for importing courses for new semester (for the first time).
 - The default setting is that
 - "First Import" cron runs every 10th of October, November, March, and June (2 months before each semester starts).
 - Set the period of running the cron job to "Once a day".
 2. Cron job in **migrate_courses**
 - This is for updating existing courses (only the most recent semester).
 - Set the period to "Once a day".

Set Views Appropriately

- This is very specific to each site. Some examples are presented below.

(Example) Create a View for Courses

1. Set View Name: "Courses"
2. Fields are like this

TITLE

Title: [Courses](#)

FORMAT

Format: [Table](#) | [Settings](#)

FIELDS

[Add](#) ▼

[Course Code \(Course\)](#)

[Content: Title \(Title\)](#)

[Content: Sections \(Sections\)](#)

[Content: Sections \(Instructor\)](#)

FILTER CRITERIA

[Add](#) ▼

[Content: Published \(Yes\)](#)

[Content: Type \(= Course\)](#)

[Content: Has taxonomy term \(exposed\)](#) | [Settings](#)

[Content: Has taxonomy term \(exposed\)](#) | [Settings](#)

3. Content: Sections (Sections) is as follows:

Label☒ Place a colon after the label☐ Exclude from display

Enable to load this field as hidden. Often used to group fields, or to u:

Formatter**Edit link title**

Leave the title empty, to hide the link.

Delete link title

Leave the title empty, to hide the link.

Add link title

Leave the title empty, to hide the link.

☐ Show the field description beside the add link.

If enabled and the add link is shown, the field description is shown in

View mode

4. Content: Sections (Instructor) is as follows. If you don't have "Revision comparison" display in the "View mode", you can set other display that you have set through Display Suite.

Label☒ Place a colon after the label☐ Exclude from display

Enable to load this field as hidden. Often used to group fields, or to use a

Formatter**Edit link title**

Leave the title empty, to hide the link.

Delete link title

Leave the title empty, to hide the link.

Add link title

Leave the title empty, to hide the link.

☐ Show the field description beside the add link.

If enabled and the add link is shown, the field description is shown in fro

View mode

Select the view mode

5. Sections are making use of Content Type's different views to show each field.
6. Final view looks like this

Courses

Term/Year

- Any -

Course Level

- Any -

Apply

Course	Title	Sections	Instructor
CLAS170	Greek and Roman Mythology	0101	Gregory Staley
		0102	Gregory Staley
		0103	Gregory Staley
		0104	Gregory Staley
		0105	Gregory Staley
		0106	Gregory Staley
		0107	Gregory Staley
		0201	Gregory Staley
		0202	Gregory Staley
		0203	Gregory Staley
		0204	Gregory Staley
		0205	Gregory Staley
		0206	Gregory Staley
		0207	Gregory Staley
		0208	Gregory Staley
CLAS309	Special Topics in Classical Literature; Inventing Ancient Greek Culture	0101	Eric Adler
CLAS320	Women in Classical Antiquity	0101	Lillian Doherty
CLAS330	Ancient Greek Religion: Gods, Myths, Temples	0101	Jorge Bravo

(Optional) SLLC-specific Configurations: in a case where a website has many courses page

1. Create a View using Contextual Filter (taxonomy term for the filter is department): This enables us to create a courses page in each department.

Page

Unit Courses

+ Add

edit view name/description

▼ Unit Courses details

Display name: Unit Courses

clone Unit Courses

TITLE

Title: Courses

FORMAT

Format: Table | Settings

FIELDS

Course Code (Course)

Content: Title (Title)

Content: Sections (Sections)

Content: Sections (Instructor)

Add

FILTER CRITERIA

Content: Published (Yes)

Content: Type (= Course)

Content: Has taxonomy term (exposed) | Settings

Content: Has taxonomy term (exposed) | Settings

Add

SORT CRITERIA

Content: Post date (desc)

Add

PAGE SETTINGS

Path: /unit%/courses

Menu: No menu

Access: Permission | View published content

HEADER

Add

FOOTER

Add

PAGER

Use pager: Full | Paged, 10 items

More link: No

▼ Advanced

CONTEXTUAL FILTERS

Content: Has taxonomy term ID

Add

RELATIONSHIPS

Add

NO RESULTS BEHAVIOR

Add

EXPOSED FORM

Exposed form in block: No

Exposed form style: Basic | Settings

OTHER

Machine Name: page_1

Comment: No comment

Use AJAX: No

Hide attachments in summary: No

Hide contextual links: No

Use aggregation: No

Query settings: Settings

- Existing system path *

https://devel.sllc.umd.edu/

unit/arabic/courses

Specify the existing path you wish to alias. For example: node/28, forum/1, taxonomy/term/1

Path alias *

https://devel.sllc.umd.edu/

arabic/courses

Specify an alternative path by which this data can be accessed. For example, type "about" when

Save

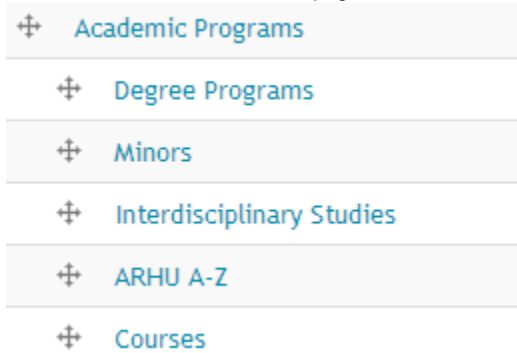
Delete

- ▶ About
- ▶ Undergraduate
- ▶ Flagship
- ▶ Summer Institute
- ▶ Beyond the Classroom
- ▶ News
- ▶ Events
- ▶ Contact
- ▼ Courses

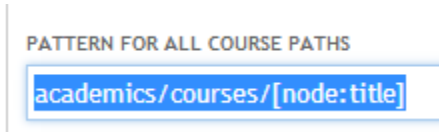
4. Style the pages using CSS and "node--courses.tpl.php" file. You can make use of the example CSS code in the package.

(Optional) ARHU-specific Configurations

1. Create a menu item for Courses page



2. Set URL alias for Courses



3. Modify the path in Courses View to an appropriate path
4. Style the pages using CSS and "node--courses.tpl.php" file. You can make use of the example CSS code in the package.

(Optional) Requirements from English