How-To Guide

for the Interactive e-Learning Platform for the Eummena Prima

Developers' Guide

Riyadh, Saudi Arabia: 25 January 2022

Synopsis Guide for Developers

Bespoke Development for this Platform

Bespoke Development on Registration Form

For the sign up page, we adapted Moodle Core code, for the /login/signup_form.php file. To mark these changes we have added in the code the comment “Eummena hack start” and we mark the end of the changes with the “Eummena hack end” comment.

English / Arabic Translations for blocks on Landing Page

In order to have the landing page (and other pages e.g. the FAQs page, contact page, etc) displayed according to the selected language, the content has to be available in both English and Arabic. To do so, the developer needs to turn the editing on by clicking on the “Options” button and selecting “Turn editing on”. After that, click on the “gear” icon for any content block and choose the first option. Now you can change the content of these blocks. To add the English and Arabic content,, you will have to enclose the text in HTML tags like this:
English contentArabic content

Example:
How to Enrollكيفية الالتحاق

Notes:

- Make sure no other attributes go inside these tags, e.g. style=” ”
- Make sure there are no extra spaces between the different elements, e.g.: English content Arabic content

Developer guidelines

Moodle is provided freely as Open Source software, under the GNU General Public License. Anyone can adapt, extend or modify Moodle for both commercial and

non-commercial projects without any licensing fees and benefit from the cost-efficiencies, flexibility and other advantages of using Moodle.

Coding

Moodle tries to run on the widest possible range of platforms, for the widest possible number of people, while remaining easy to install, use, upgrade and integrate with other systems. Moodle has a general philosophy of modularity. There are nearly 30 different standard types of plugins and even more sub-plugin types, however all of these plugin types work the same way. Blocks and activities are the only small exceptions.

Consistent coding style is important in any development project, and particularly so when many developers are involved. A standard style helps to ensure that the code is easier to read and understand, which helps overall quality. It's important that Moodle produces strict, well-formed HTML 5 code (preferably backwards compatible with XHTML 1.1 if possible), compliant with all common accessibility guidelines (such as W3C WAG 2.0, ARIA).

CSS should be used for layout. Moodle comes with several themes installed. Beginning with version 2.7, only the 'Clean' theme comes in the base Moodle code. The 'standard' theme, which should be a plain theme suitable to act as a building block for other themes. That should contain the minimal styling to make the platform look OK and be functional. Then Moodle comes with several other default themes that look good and demonstrate various techniques for building themes.

Moodle has a powerful database abstraction layer that we wrote ourselves, called XMLDB. This lets the same Moodle code work on MySQL/MariaDB, PostgreSQL, MS SQL Server and Oracle. There are known issues when using Oracle, it is not fully supported and is not recommended for production sites.

Moodle Tracker

The Moodle tracker keeps track of the status of all bug fixes and new features. Moodle uses a workflow that ensures that new code receives multiple reviews by different people before it is included into the core Moodle code.

Stable Maintenance Cycles

Moodle releases regular updates of the stable version of the software to fix bugs and other issues. Releases like 2.2.1, 2.2.2, 2.2.3 etc only include fixes based on the latest

major release (2.2) and never any significant new features or database changes. At Moodle HQ there are teams of developers using the Scrum framework to work on these issues (as well as new features for major releases).

Major release cycles

Since Moodle 2.0, there has been a policy of releasing major versions (eg 2.1, 2.2) every six months in May and November. Each release can be different, but generally the cycles work as follows.

- Define Roadmap
- Planning and Development Sprints
- Testing

Material on plugin development

Activity Modules

Activity modules reside in the /mod directory. Each module is in a separate subdirectory and consists of a number of 'mandatory files' and any other files the developer is going to use. The below image is an example of the certificate module's file structure. Please note, any reference to <modname> in this documentation should be replaced by the name of your module.

Blocks

You can follow the creation of an "HTML" block from scratch in order to demonstrate most of the block features at our disposal. Our block will be named "SimpleHTML". This does not constrain us regarding the name of the actual directory on the server where the files for our block will be stored, but for consistency we will follow the practice of using the lowercased form "simplehtml" in any case where such a name is required.

Themes

What is a theme? A theme in Moodle is just another type of plugin that can be developed. Themes are responsible for setting up the structure of each page and have the ability to customise the output of any page in Moodle.

Course Formats

Course formats are plugins that determine the layout of course resources. Course formats determine how the course main page looks like (/course/view.php) in both

view and editing mode. They are also responsible for building navigation tree inside the course (displayed to users in the navigation block and breadcrumb). They can organise the course content in sections. Course creator or teacher can specify course format for the course in course edit form.

Moodle Core APIs

Access API

The Access API gives you functions so you can determine what the current user is allowed to do. It also allows modules to extend Moodle with new capabilities.

Data Manipulation API

These are used to access data in the Moodle database. You should exclusively use these functions in order to retrieve or modify database content because these functions provide a high level of abstraction and guarantee that your database manipulation will work against different RDBMSes.

Navigation API

The Navigation API allows for the manipulation of the navigation system used in Moodle.

Page API

The Page API is used to set up the current page, add JavaScript, and configure how things will be displayed to the user.

Persistent API

Moodle persistents are equivalent to models (or active records). They represent an object stored in the database and provide the methods to create, read, update, and delete those objects. Additionally, the persistents validate their own data against automatic and custom validation rules. Persistents are made by extending the abstract class core\persistent.

Developer tools for Moodle

Linting

In Moodle development code linters are used to help ensure consistent coding conventions and help prevent common errors in code. Linting is used to ensure consistent coding style, detect common errors and enforce best practices. At it's best linting helps developers learn, with editor integration and fast feedback meaning that human code reviews can focus on the non-trivial details while linter can be the tireless mistake-preventing machine.

Eclipse

Eclipse is an IDE originally designed for Java, but now with plugins for many languages including PHP. It has lots of very powerful features, and it is the editor that some Moodle developers like to use. Other (more) popular choices are vim and emacs.

Netbeans

NetBeans has got good PHP support. You find a host of information on the website (tutorials, developer blog, screen casts, etc.). You can install the optional JIRA module in order to be able to interact with the Moodle Tracker from within NetBeans. This has the advantage of avoiding constantly switching back and forth from the browser and works quite well. Go to Tools->plugins->available plugins and search for JIRA. Once installed, right click 'issue trackers' in the 'services' pane to make a new tracker instance, then enter your details.