The Moodle mobile app for LMS developers

Mobile Solutions Team

Key aspects of the Moodle mobile app

- Designed for students
- Works offline
- Supports [LMS plugins](...) (if the plugin author made it compatible)
- Supports multiple sites
- Does not support Bootstrap classes
- Supports Moodle LMS 3.5 onward (LMS 3.9 starting app 4.5 onward)
- Requires Android 5.1 (with Webview 61 or higher) and iOS 11 (requirements will change soon, [see forum post](...))

Which LMS features are supported

- Most of the student functionality is supported
- Functionality still not supported (version 4.4):
  - RTE plugins and attachments
  - Edit user profile and preferences
- For knowing more, download the
  [PDF with a comparison LMS/app](#)

Site Features

Site features	Moodle LMS (web)	Moodle App
Site home		
View custom content	✔	✔ 🛜
View blocks	✔	✔ 🛜
View default dashboard	✔	✔ 🛜
Dashboard		
View blocks	✔	✔ 🛜
Customise dashboard	✔	X
Order and filter courses by completion status	✔	✔ 🛜
Star courses	✔	✔

Architecture

- Stack: Angular 17 and Ionic 7 (version 4.4)
- [Hexagonal architecture](#)
  - Core modules that include the main functionality
  - Addon modules that provide additional features
- Modules are defined as Angular Modules
- Modules are resolved at runtime using Angular's dependency injection framework.
- Pages and navigation are defined using Angular Router, making heavy use of lazy loading.
- Testing:
  - Behat for Acceptance testing (225 scenarios in 69 features), Moodle Tracker Jenkin integration (pre-launch remote branch)
  - Jest for Unit Testing

How the app talk to LMS

- The REST protocol and the Mobile app service is automatically enabled on installation (only if site using https)
- The app is basically a Web Service client using WS tokens for access
- Authentication:
  - In most of the cases the login/token.php is used to obtain the token and securetoken
  - It is also possible to perform the authentication outside the app and launch the app via Custom URL schemes
- Performance:
  - The data retrieved from some Web Services is cached automatically
  - Some WS request are grouped into single requests using a custom WS

How the app talk to LMS

- Cache:
  - Cache strategy depends on the device connection (Wi-Fi or data)
- Security:
  - The app tries to use the tokenpluginfile.php script, using a different access key to download files (more secure than using webservice/pluginfile.php)
  - CORS is allowed in webservice/ endpoints
  - A different token is used when an automatic login (from app into browser) process is launched by the app
  - Requests to embed content from the site into the app are automatically allowed (LMS checks the User Agent for it)

How the app renders content

- In most of the cases, the app just download structured data via WebServices from the Moodle site and uses local templates using Ionic web components to display content
- SCORM and H5P zip packages are downloaded and extracted in the app
- In rare cases it is necessary to retrieve the content already rendered from the server:
  - Question types
  - Lesson
  - Gradebook table visualisation
  - Some Blocks
  - Plugins

"If you are working on a bug fix or improvement that changes the UI of a page that students have access to, it is very probable that the change is going to affect the mobile app"

The Mobile app rule of thumb

The Mobile app rule of thumb

- Examples of the previous are:
  - Adding or removing elements from the interface (adding a new field in the user profile, removing some options that were already visible)
  - Re-positioning elements (indentation)
  - Changes in the course index (communication tools)
  - Changes in the default course formats (subsections)
  - Changes in the UI of activity modules (assignment, quiz)
  - Icons in general (Font Awesome, activity icons)

Good practices

- If you are working on a new functionality for students, you'll need to reach the apps team and consider that:
    - Your new feature should implement new Web Services or expand existing ones to return all the data that would be required to display the information to the student (Think of this as creating a web service that returns the same sort of data as used in a mustache template)
- Changes in the rendering of question types usually break the app (accessibility)
- Upgrading H5P library or changing its code might require changes in the app
- Test your changes in the app just before sending it to IR
- In case of doubt, we are available for any question/doubt
- Please use the affects_mobileapp label

Useful links

- [Moodle App dev docs](#)
- [Testing LMS issues in the app](#)
- [Integration review guidelines](#) (special mentions to the Mobile app)
- [Peer review guidelines](#) (special mentions to the Mobile app)