



- Home
- Documentation
- Downloads
- Demo
- Tracker
- Development
- Translation
- Moodle.net
- Search

You are here

- Main Page
- ► Moodle Mobile custom apps

Moodle Mobile custom apps

Moodle Mobile

- Moodle Mobile Customization
- Setting up your development environment for Moodle Mobile 2
- Moodle Mobile development using Chrome or Chromium
- Mobile support for plugins
- Moodle Mobile Themes
- Moodle Mobile debugging WS requests
- Moodle Desktop
- Moodle Mobile FAQ

Warning: This page is no longer in use. The information contained on the page should NOT be seen as relevant or reliable.

Contents

- 1 Overview
- 2 Requirements
- 3 Steps for creating your Custom App
 - 3.1 Setting up your computer
 - 3.2 Fork the app
 - 3.3 Clone the repo in your desktop
 - 3.4 Create a new branch
 - 3.5 Edit the app settings file
 - 3.6 Edit another app files
 - 3.7 Replace the app icons, splashscreen
 - 3.8 Add your custom styles
- 4 Building your Custom App
 - 4.1 Building with Phonegap Build
 - 4.1.1 GitHub
 - 4.1.2 Phonegap
 - 4.2 Remove existing features
 - 4.3 Change the app strings
 - 4.4 Change the initial screen (add site)
 - 4.5 Keep your custom version up2date

Overview

You will need a custom app if you want:

- Force the users to log in to your site URL
- Custom corporate image (name, app icon, theme, store entry)
- Add additional features or static pages
- Remove or change existing features
- Custom translation of the app

- Use your own notifications infrastructure

Requirements

- Code editor (Sublime, WebStorm, Eclipse, Netbeans...)
- Github account (<http://github.com/>)
- Github desktop app (<http://windows.github.com/>)
- PhonegapBuild account (<https://build.phonegap.com/>)
- Chromium browser as described here: Moodle Mobile development using Chrome or Chromium

Steps for creating your Custom App

Setting up your computer

Follow these guides:

- Setting up your development environment for Moodle Mobile 2
- Moodle Mobile development using Chrome or Chromium

Fork the app

Use the Fork tool in: <https://github.com/moodlehq/moodlemobile2> (You must be logged with your Github user!)



Clone the repo in your desktop

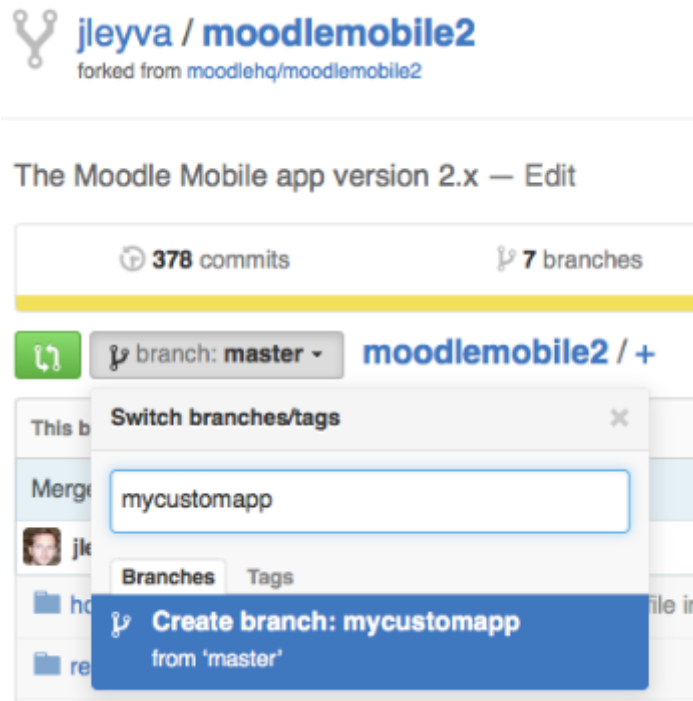
You can use your favourite GIT client (or GitHub Desktop clients)

```
git clone git@github.com:jleyva/moodlemobile2.git
```



Create a new branch

Having the code in a separate branch in the forked repository will make very easy to maintain our customizations using git tools (merge, rebase)



Edit the app settings file

/config.xml

- Change the widget id value (instead com.moodle.moodlemobile your own id)

- Change the app name and description

/www/config.json

- Change versionname
- Add a pre-fixed URL: Adding a field like “siteurl”: “http://mysite.abc”
- Change gcmprn (if you are using your custom Google Push infrastructure)
- Change the appid (required for making Push notifications work)
- Remove demo sites (“demo_sites”: “”)

Edit another app files

/www/core/component/settings/lang/x.json (x means all the languages)

- Change the “appname” string to match your custom app name (Not necessary in latest version Moodle Mobile 3.1.3), you can change it in config.json

/www/errorreport.js

- Change: var reportUrl to point to your custom error reporting system (if you want to use your custom one)
- Same for appVersion

Replace the app icons, splashscreen

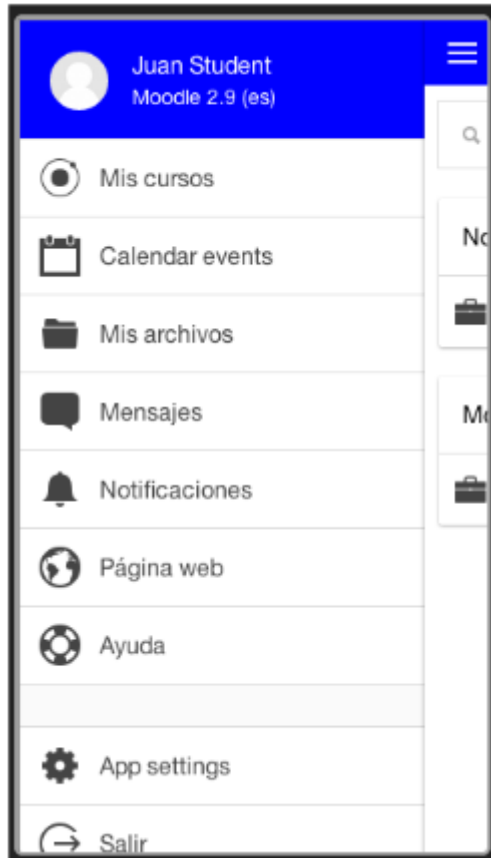
- Replace the resources/ icon.png and splash images
- Replace also the www/img/moodle.png with an image matching your logo
- **In a console:**
 - cd mycustomappdir/
 - ionic resources

This will create your custom icon and splash images in all the required sizes

More information about Ionic’s image generation: <http://ionicframework.com/docs/cli/icon-splashscreen.html>

Add your custom styles

- Add your custom styles in scss/app.scss
- Sample theme available at: Moodle Mobile Themes



Building your Custom App

There are two options for building the app:

- Ionic CLI
 - Local building (require installation of Android and iOS SDK and emulators)
 - See <http://ionicframework.com/docs/cli/run.html>
- Phonegap Build
 - Remote building (cloud service)
 - This is the recommended way because it doesn't require to install all the SDK, emulators (or having to buy a Mac for developing)

Building with Phonegap Build

GitHub

- First of all, you need to create a new repository in GitHub. This is because Phonegap Build requires a different directory structure than Ionic CLI and we need a branch per platform.
- In your forked repository, run *gulp* so the build files are updated.
- Copy all the code inside the *www/* in your forked repository directory to this new repo/branch.
- Also copy the */resources* directory from the ionic project to a new */resources* directory in the phonegap build repository
- You also have to place the */resources/icon.png* and */resources/splash.png* you used to create your resource files in the root folder in the phonegap build repository
- Use this repository as an example: <https://github.com/moodlehq/moodlemobile-phonegapbuild/tree/learn-moodle>
- In the Phonegap Build repo root folder you need to add a special *config.xml* file (different to the Ionic one), use this file as a template. Please, replace there the app name, versions, etc.
- Then, in the Phonegap build repo you need to create two additional branches based on master: *android* and *ios*
- In the *android* and *ios* branch you need to apply these changes:

Disable debugging in the app:

2		config.xml
		@@ -118,7 +118,7 @@
118	118	<gap:plugin name="ch.ti8m.documenthandler" version="0.2.1" />
119	119	
120	120	<gap:config-file platform="android" parent="/manifest">
121		- <application android:debuggable="true" />
	121	+ <application android:debuggable="false" />
122	122	</gap:config-file>
123	123	
124	124	<!-- See MOBILE-892 -->

2		core/lib/log.js
		@@ -14,7 +14,7 @@
14	14	
15	15	angular.module('mm.core')
16	16	
17		-.constant('mmCoreLogEnabledDefault', true) // Default value for logEnabled.
	17	+.constant('mmCoreLogEnabledDefault', false) // Default value for logEnabled.
18	18	.constant('mmCoreLogEnabledConfigName', 'debug_enabled')
19	19	
20	20	/**

Add android or ios base class (so the app use native style widgets):

2  index.html		
@@ -14,7 +14,7 @@		
14	14	<script src="lib/angular-md5/angular-md5.min.js"></script>
15	15	<script src="build/mm.bundle.js"></script>
16	16	</head>
17	-	<body ng-app="mm">
	17	+ <body ng-app="mm" class="platform-android platform-cordova platform-webview">
18	18	<ion-nav-view></ion-nav-view>
19	19	</body>
20	20	</html>

- The rest of the commits can be safely omitted:
 - Add reporting library
 - Add official tag

Phoneygap

- Create an account in: <https://build.phonegap.com/>
- For iOS you need to create certificates/provisioning files: http://docs.build.phonegap.com/en_US/
- For Android, you can download and install the .apk for testing, for production follow the previous guide.
- In order to build, add your repository https link and also the branch in the Phonegap Build settings (you will have to do two different builds: one for the android branch and another one for the ios branch)
- In order to update your app for phonegap build you have to (this will apply all the specific android/ios commits to your master branch):
 - Upload new versions of the app to the www/ directory and then execute the commands:
 - git rebase master android
 - git push -f
 - git rebase master ios
 - git push -f

Remove existing features

- You can only remove add-ons.
- In order to disable an add-on you need to delete the folder.
- Add-ons are located in the www/addons folder.

- In case you want to remove a particular functionality of an add-on you will need to edit the files inside the addon. Please notice that removing a state is not enough for a feature to be disabled, you also need to remove all the links to that state to prevent errors.
- Then run **gulp** or **ionic serve** in the project root folder so the build files are created.

Change the app strings

- Language strings are located in:
 - `www/core/lang` folder (a different file for each language)
 - `www/core/components/COMPONENT_NAME/lang` folder
 - `www/addons/ADDON_NAME/lang` folder
- Language files are `.json` files, you can edit it using a text editor. Please, use jsonlint.com to validate that the new language file is correct.
- Then run **gulp** or **ionic serve** in the project root folder so the build files are created

Change the initial screen (add site)

- To make the app use a fixed site URL you need to add a new setting in the `config.json` file: `"siteurl": "http://yourmoodle.com"`
- Then run **gulp** or **ionic serve** in the project root folder so the build files are created.

You can see an example from our Learn Moodle app (that is a custom version of Moodle Mobile) here: <https://github.com/jleyva/moodlemobile2/blob/learn-moodle/www/config.json>

Keep your custom version up2date

- Sync your fork: <https://help.github.com/articles/syncing-a-fork/>
- Rebase your changes (and resolve conflicts):
 - `git rebase master yourcustombranch`
- Upgrade your version number
- Commit your changes
- Apply the changes in your Phonegap Build specific repository
- Rebuild with Phonegap build for both android and ios branches
- Publish in the store

Retrieved from "https://docs.moodle.org/dev/index.php?title=Moodle_Mobile_custom_apps&oldid=52091"

Categories: Mobile | Obsolete

-
- This page was last modified on 29 March 2017, at 10:33.

- Content is available under GNU General Public License unless otherwise noted.