

---

# Mobilepartners Documentation

*Release 1.0*

**Mozilla**

**Sep 27, 2017**



---

## Contents

---

<b>1</b>	<b>About MobilePartners</b>	<b>3</b>
<b>2</b>	<b>Contents</b>	<b>5</b>
2.1	Installing MobilePartners (FirefoxOS Scaling Website (FXOSS)) . . . . .	5
2.2	Groups & Permissions . . . . .	6
2.3	Content Authoring . . . . .	8
2.4	Site Translations . . . . .	13
2.5	Rosetta Translations . . . . .	14



This documentation explains how to install and use [mobilepartners.mozilla.org](http://mobilepartners.mozilla.org).



# CHAPTER 1

---

## About MobilePartners

---

<https://mobilepartners.mozilla.org/> is the self service website for OEMs and Operators.





## Installing MobilePartners (FirefoxOS Scaling Website (FXOSS))

### Pre-Requisites

Below you will find basic setup and deployment instructions for the FXOSS project. To begin you should have the following applications installed on your local development system:

```
- Python >= 2.6 (2.7 recommended)
- `pip >= 1.1 <http://www.pip-installer.org/>`_
- `virtualenv >= 1.7 <http://www.virtualenv.org/>`_
- `virtualenvwrapper >= 3.0 <http://pypi.python.org/pypi/virtualenvwrapper>`_
- PostgreSQL >= 9.1.11
- git >= 1.7
```

### Getting Started

To setup your local environment you should create a virtualenv and install the necessary requirements:

```
mkvirtualenv fxoss -r $PWD/requirements/dev.txt
```

Then create a local settings file and set your `DJANGO_SETTINGS_MODULE` to use it:

```
cp $PWD/project/settings/local.py-dist $PWD/project/settings/local.py
```

Create the PostgreSQL database and run the initial syncdb/migrate:

```
createdb fxoss
python manage.py syncdb
python manage.py migrate
```

You should now be able to run the development server:

```
python manage.py runserver
```

## Deployment

TODO

## Fixtures

Load the default fixtures for the site. Currently, the only fixtures that exist define a *Content Authors Group*. To run:

```
python manage.py loaddata fixture_groups.json
```

## Groups & Permissions

Below you will find documentation on groups and permissions levels.

### Granting Admin Permissions

Users who need to access the admin must have the following permissions:

Permissions
<input checked="" type="checkbox"/> <b>Active</b> Designates whether this user should be treated as active. Unselect this instead of deleting accounts.
<input checked="" type="checkbox"/> <b>Staff status</b> Designates whether the user can log into this admin site.
<input type="checkbox"/> <b>Superuser status</b> Designates that this user has all permissions without explicitly assigning them.

### Groups: Content Author

This group grants the user permission to edit most of the content (Pages, Forms, Media) within the admin.

### Groups: Signed Agreement Viewer

Granted to people who need to be able to view the signed agreements. The permissions for this group are shown below:

**Name:** Signed Agreement Viewer

Permissions: Hold down "Control", or "Command" on a Mac, to select more than one.

**Available permissions**

Filter

- admin | log entry | Can add log entry
- admin | log entry | Can change log entry
- admin | log entry | Can delete log entry
- auth | group | Can add group
- auth | group | Can change group
- auth | group | Can delete group
- auth | permission | Can add permission
- auth | permission | Can change permission
- auth | permission | Can delete permission
- auth | user | Can add user
- auth | user | Can change user
- auth | user | Can delete user
- comments | comment | Can add comment

Choose all

**Chosen permissions**

- protected\_assets | signed agreement | Can add signed a
- protected\_assets | signed agreement | Can change sign
- protected\_assets | signed agreement | Can delete signed

Remove all

## Groups: Translators

Granted to people who need to perform translations. This grants users permissions to access/edit Rosetta and Todos. The permissions for this group are shown below:

**Name:** translators

Permissions: Hold down "Control", or "Command" on a Mac, to select more than one.

**Available permissions**

Filter

- admin | log entry | Can add log entry
- admin | log entry | Can change log entry
- admin | log entry | Can delete log entry
- auth | group | Can add group
- auth | group | Can change group
- auth | group | Can delete group
- auth | permission | Can add permission
- auth | permission | Can change permission
- auth | permission | Can delete permission
- auth | user | Can add user
- auth | user | Can change user
- auth | user | Can delete user
- comments | comment | Can add comment

Choose all

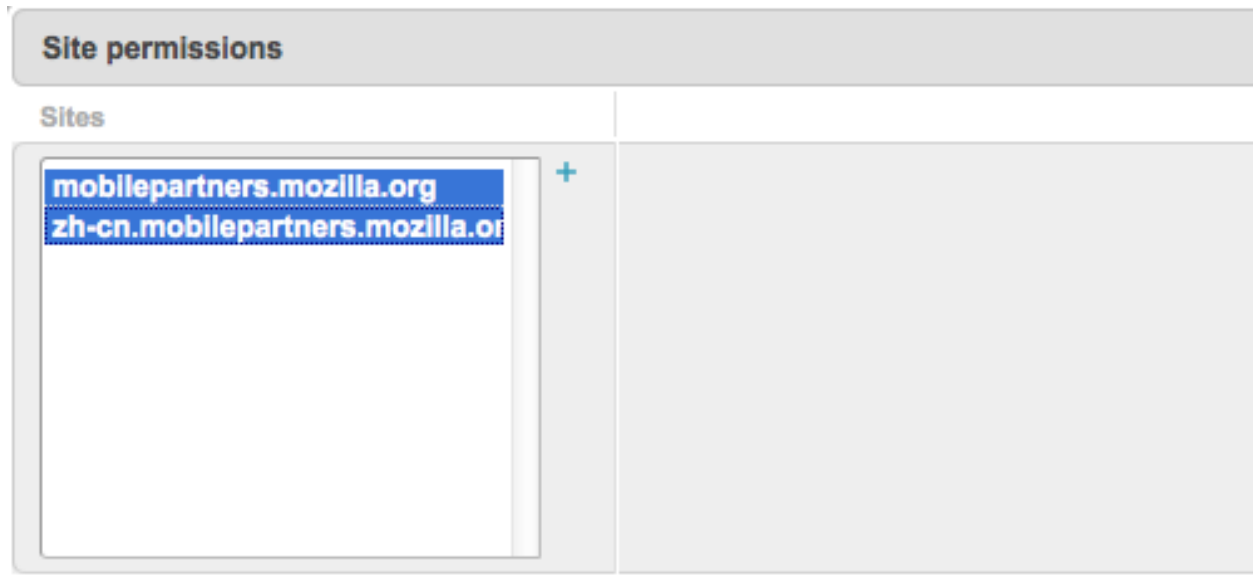
**Chosen permissions**

- translations | todo item | Can add todo item
- translations | todo item | Can change todo item
- translations | todo item | Can delete todo item

Remove all

**You must manually add users to each site they need to translate.**

**Important:** When adding translators in addition to adding them to the correct group you must also select all the sites they need to access. Translators must be given access to the US site to view that content.



## Content Authoring

Below you will find basic content authoring instructions for the FirefoxOS Scaling Website (FXOSS) project.

### Authenticating

In order to author content in the FXOSS site, you will need to:

1. Navigate to **mobilepartners.mozilla.org/admin**
2. Supply the **username** (email) and **password** you
3. Select the **Admin Interface**
4. Press **Login**

### Dashboard

The dashboard provides quick access to various site components. The pieces to focus on for Content Authoring are the *Pages* and *Media Library* sections under the *Content* category.

### Pages

Pages are the primary piece of authored content as the site's navigation and structure are derived from the created pages. To add a Page to the root of the site, click **Add+** in the Pages Row. This will open a form that allows you to create a Rich Text Page:

- **Title:** The title that will appear in the navigation node(s) for the Rich Text Page.
- **Status:** Select Draft to hide the page from all but admin users. Publishing the page will allow all users to view the page.
- If publication dates are required, set the optional **Published from** and **Expires on** fields

- **Show in menus** allows the author to indicate which menu(s) the navigation element should appear in. The FxOSS theme requires *Top navigation bar* and *Left-hand tree* menus to be checked in order for the node to render in a responsive manner. The footer nav is unused in this theme.
- **Subtitle** of the page
- **Intro Paragraph**
- **Content:** This field supports WYSIWYG editing, and easy insertion of related media
- **Closing Paragraph**
- **Login required** protects content from anonymous users.
- To explicitly declare additional meta data, expand the **meta data** tab and populate the optional fields accordingly.
- Be sure to press **Save** to store the Page in the database.

Creating a Child Page is as simple as:

1. Navigate to the dashboard
2. Click **Pages**
3. Selected the **Add...** dropdown next to the Page you would like to append a child too, and select **Rich Text Page**.
4. Add your content as above, and **Save**.

## Media Library

The Media Library stores upload images, pdfs, and other related media that will be linked to from within a Rich Text Page. It is accessible either from the Dashboard, or via the WYSIWYG editor while editing a Page. The Media Library supports Folders and uploading multiple files at once.

If you have all of your related media ready to upload before creating any Pages:

1. Navigate to the **Media Library** from the Dashboard
2. Create Folders (if needed) to structure your media via the **New Folder** button
3. Navigate to the target folder and select **Upload**
4. Press **Select Files** to browse your local file system for files to upload

Alternately, one may upload related media during Page authoring.

1. Navigate to (or create a) **Page**
2. In the **Content** field, type and select some text
3. Click **Insert/Edit Link** in the toolbar
4. Click the **browse icon** in the popup next to Lin url
5. The Media Library popup will render, and one can create folders/upload per the above instructions.
6. Once file(s) have been uploaded, click the **Select File** button next to the file you want to insert.
7. Make any additional configuration changes in the **Insert/Edit Link** popup
8. Click **Insert**
9. Be sure to **Save** changes to the Page

## Other Content Types

If you need to generate an External Link as a navigation element, this can be accomplished by creating a Link object from the Pages admin section

1. Click **Pages**
2. Click **Add... Link** at the root of the site or as a child of an existing content node.
3. Ensure the **URL** field is a valid URL
4. Edit other fields as needed
5. Press **Save**

It is possible to generate a Form from within the Admin. This is beyond the normal use case for this site. In short, similar to creating a Page, an author can generate a form to collect data from end users.

## Download Agreement

The download agreement is located in the **Protected\_Assets** section, under **Agreements**. Here you can see a list of all past and current agreements.

When the download agreement changes, you will need to add a new entry in the agreement list:

1. Click **Agreements**.
2. Click **Add Agreement** in the top right of the listing.
3. Enter a version name and select the PDF file for the new agreement.
4. Click **Save**.

You will also need to update the **Download Agreement Version** setting to match the version of the agreement you just added:

1. Navigate to the **Site > Settings** from the Dashboard.
2. Change the **Download Agreement Version** under **Miscellaneous** to match **exactly** the version of the current agreement.
3. Press **Save**.

---

**Important:** The **Download Agreement Version** is under **Site > Settings** which means it is **not** a global setting. Different sites correspond to different languages, and if the version specified is available in English but not the site language, users will be presented with the English version instead.

---

You may also see a list of signed agreements, including which specific version was signed, under the **Protected\_Assets > Signed Agreements** section.

### Updating the version of an agreement will prompt users to re-sign

---

**Important:** When the version number of the agreement is changed, all users who previously agreed to the agreement will be prompted to resign (when they try to access restricted content). If the change to the agreement does not require a resign, simply edit the current download and upload a new version without changing the version number.

---

## Protected Downloads

Protected downloads are media assets which require the user to sign the user agreement prior to accessing. These are a special case of assets managed under the **Media Library**. These assets are uploaded into a **protected** folder within the **Media Library**.

1. Navigate to the **Media Library** from the Dashboard
2. Create Folder a folder named **protected** (if needed) via the **New Folder** button
3. Navigate to the **protected** folder and select **Upload**
4. Press **Select Files** to browse your local file system for files to upload

## TinyMCE Snippets

The rich content fields in the CMS use TinyMCE for editing the HTML. The templates plugin for TinyMCE allows creating reusable patterns for content. These snippets can be created and modified in the CMS via **Snippets > Tiny MCE Snippets** from the Dashboard. Each snippet has a title and description to help select and understand the purpose of the snippet along with the actual snippet content itself.

Once some snippets have been created they can be used by any rich text field. The template button is second to last on the right with the hover text “Insert Predefined Template Content”. When pressed it will open a pop-up with a drop-down to select a snippet based on its title. After selecting the snippet, the description and content preview should be shown. Clicking the “Insert” button on the bottom left hand corner of the pop-up will insert the previewed content into the original rich text field at the cursor location. These snippets are meant to only contain the expected layout and not the content itself. Once the snippet has been inserted you will likely need to modify the content which was inserted with the relevant text rather than the snippet placeholder.

## Default Snippets

Requirements Table:

```
<h3 class="mini-title">Category Name</h3>
<table border="0" class="mini-table">
<tbody>
<tr>
<td>Manufacturer</td>
<td>Versions Supported</td>
</tr>
<tr>
<td>Foo</td>
<td>Bar</td>
</tr>
</tbody>
</table>
```

CATEGORY NAME	Manufacturer	Versions Supported
	Foo	Bar

Offset Title Block:

```
<h3 class="mini-title">Title</h3>
<p class="offset-block">Lorem Ipsum <br> <a class="follow" href="http://google.com">
↪Read More</a></p>
```

TITLE	Lorem Ipsum <a href="#">Read More »</a>
-------	--

## CSS Classes and Tables

There are a number of CSS classes defined that will allow tabular content to conform to the standard site look and feel. All of these are accessible when using TinyMCE's *Insert/Edit Table* functionality. The popup window has a *class* selector from which you can choose one of the following options:

**Default:** Extends the full width of the content well.

Groups and Feeds	IRC
<a href="#">Mailing List »</a>	<a href="#">IRC: #b2g »</a>
<a href="#">News Group »</a>	<a href="#">IRC: #gaia »</a>
<a href="#">Google Group »</a>	<a href="#">IRC: #openwebapps »</a>
<a href="#">RSS Feed »</a>	<a href="#">IRC: #marketplace »</a>
	<a href="#">IRC: #webapi »</a>

**Two Column:** Extends the full width, first column set to 30%.

Groups and Feeds	IRC
<a href="#">Mailing List »</a>	<a href="#">IRC: #b2g »</a>
<a href="#">News Group »</a>	<a href="#">IRC: #gaia »</a>
<a href="#">Google Group »</a>	<a href="#">IRC: #openwebapps »</a>
<a href="#">RSS Feed »</a>	<a href="#">IRC: #marketplace »</a>
	<a href="#">IRC: #webapi »</a>

**Mini:** Extends 70% of full width.



Groups and Feeds	IRC
Mailing List »	IRC: #b2g »
News Group »	IRC: #gaia »
Google Group »	IRC: #openwebapps »
RSS Feed »	IRC: #marketplace »
	IRC: #webapi »

**Simple Bordered:** A simple outlined table. All table cells and table headers will have solid 1px border. No additional font related styling for td or th cells.

Groups and Feeds	IRC
Mailing List »	IRC: #b2g »
News Group »	IRC: #gaia »
Google Group »	IRC: #openwebapps »
RSS Feed »	IRC: #marketplace »
	IRC: #webapi »

## Site Translations

Below is a description of how the site is translated and how those translations are managed for both existing languages and adding new ones.

### Architecture Overview

To translated the site into multiple languages, multiple copies of the site are created and managed using the existing mutli-tenancy features in Mezzanine: <http://mezzanine.jupo.org/docs/multi-tenancy.html>

**Pages are considered to be the same on each site if they have the same slug/URL.**

**Important:** Pages in locale's other than english must have the same slug. That is the existing URLs (/learn/, /market/, etc) on the front end are the same across sites. The slug sets up the relationship between pages on different locales. If there is a page specific to a locale that does not appear in english, it does not need to share the slug.

The site/language to used is detected by the browser's Accept-Language header or the django\_language cookie which can be set using the language selection in the site footer. The admin also uses localized URLs to detect the current site/language. See the Django documentation for a more verbose description of this detection process <https://docs.djangoproject.com/en/1.6/topics/i18n/translation/#how-django-discovers-language-preference>

## For Admins/Developers

The set of available languages is determined by the `LANGUAGES` setting but simply adding a new language here is not sufficient for creating a new language version. As previously noted, each language will have its own `Site` record from `django.contrib.sites` and corresponding CMS content for that site. When a new language needs to be added, a new site record and a copy of the current CMS content needs to be made. This is handled via the `build_language_sites` management command:

```
python manage.py build_language_sites <language-code>
# Adding French (fr)
python manage.py build_language_sites fr
# Adding all languages defined in the LANGUAGES setting
python manage.py build_language_sites --all
```

By default this will copy all of the CMS content (Pages, Forms, Links) from the English site for any new site which is created. This can be skipped via the `--skip-content` option. However, this is not recommended because there is currently nothing in place to copy content for an existing site. When testing/developing locally you can easily reset the site content for a non-default language by deleting the site in the admin which will cascade to delete all of the CMS content. Re-running the management command will create a fresh copy for the language.

For non-superusers, Mezzanine has a concept of site permissions. That is users can be staff for one or more sites. Since languages are handled as separate sites it is possible to restrict users to only a single language/site admin. However, if translators don't have access to the English version of the site they won't be able to view the original content for making translation updates. Care should be taken to ensure that content authors have sufficient site permissions for their translations. Site permissions can be managed in the User admin.

The language switching is enabled/disabled via a django-waffle flag. This flag can be created on the command line via:

```
python manage.py flag language-switching --staff --create
```

By default this flag will be disabled unless the flag is created. That can be configured with the `WAFFLE_FLAG_DEFAULT` setting. For more information on managing feature flags, see the django-waffle documentation <http://waffle.readthedocs.org/en/latest/index.html>

## For Translators

When logging into the admin you can select the language you would like to see. Once in the admin you can change this language at any time using the selector in the upper right-hand corner. When content is updated on the English version of the site, TODO items will be created under the "Translations" header. This will give a brief description of what was updated so that the translated content can be updated. When viewing a translated page (non-English site version), there will be a link on the upper right-hand side to open a popup window with the English version to aid the translation.

## Non CMS Translations

Non CMS content (text marked for translation in code and templates) is translatable via *Rosetta*

## Rosetta Translations

Below is a description of how the text marked as translatable in code and templates is translated and how those translations are managed for both existing languages and adding new ones.

## i18n Overview

If you have not read through the official [translation documentaion](#), you should. The Django documentation provides an excellent foundation to understand the translation infrastructure.

## For Admins/Developers

This adds an admin section (Rosetta) which allows for the translation of the non-database content (which is tracked by the .po files). Content such as the registration email bodies and subjects should be editable through this interface.

PO files are generated manually for each new locale. This is done via a management command:

```
$ python manage.py makemessages -l zh_CN
```

It is important to note that the [locale name](#) format differs from the [language code](#). Be mindful of this when setting up PO files for new locales and enabling language support for the new language code.

For example, simplified Chinese support requires adding an entry to the LANGUAGES setting with the proper language code:

```
('zh-cn', _('Simplified Chinese'))
```

While the locale name, as shown above, is `zh_CN`.

## For Translators

When logging into the admin you can navigate to **Rosetta PO File Translation** and select a PO File in the desired language to start editing. Upon save, a new compiled .MO file will be written on the server.

**In order for the translations to be visible, a server restart will be required.**

---

**Important:** If you have made any changes to the Rosetta translations a developer must restart the server for the changes to appear. To request this please contact a developer in irc #oss or file a bug Bugzilla: Websites / mobilepartners.mozilla.org

---

While it is possible in some circumstances to auto reload the MO file, it is ill advised in production environments for performance reasons. More configuration options can be read about [here](#).

## Translating Tips

Some of the content in Rosetta will have embedded html.

```
<p>
  You can also
  <a href="% (password_reset_url) s?next=% (profile_update_url) s">reset your password</a>
  if you've forgotten it.
</p>
```

In the above example you should copy the entire block but only translate the strings, so your copy should look like the following where the Xs are translated:

```
<p>
  XXX XXX XXX
  <a href="% (password_reset_url)s?next=% (profile_update_url)s">XXXXX XXXX XXXXXXXX</
↪a>
  XX XXXXX XXXXXXX XX.
</p>
```

If you see content with 2 text boxes, these are for handling the singular (Box 0) and plural (Box 1) forms of words.