

# FreePBX Internals

## What does Internal Mean?

Internal means any program that tries to access parts of FreePBX as a module that is found inside the admin/modules folder.

For now please checkout out development module located at: <https://github.com/FreePBX/helloworld>

## Creating a BMO Module

Create a file called Modulename.class.php in the root of your module directory. In this example we're creating a Pony Module, and we're going to call it ponies

To start with, the file `/var/www/html/admin/modules/ponies/Ponies.class.php` must contain at least:

```
<?php
// Ensure you namespace your module. Failing to do so may lead to
unexpected bugs.
namespace FreePBX\modules;

class Ponies implements \BMO {
    public function install() {}
    public function uninstall() {}
    public function backup() {}
    public function restore($backup) {}
    public function doConfigPageInit($page) {}
}
```

You also need to create a 'module.xml' file that contains the definition of the Module (See [module.xml](#))

The 'FreePBX\_Helpers' class is built in `BMO.interface.php`, and provides Autoloading of classes, the ability to use `$this->getConfig()` and `$this->setConfig()` - the new built in high-performance key/value store, and the management and mitigation of `$_REQUEST` responses.

Note that this by itself is not useful at all, and will in fact cause an error to be displayed, because it doesn't contain the public function 'doConfigPageInit()', which, while is not required by the BMO interface, is required by the Hooks class. See the [BMO Hooks page](#) for more information.

- [BMO Ajax Calls](#)
  - [BMO Built-In Classes 12+](#)
  - [BMO DB\\_Helper](#)
  - [BMO Hooks](#)
  - [BMO Request\\_Helper](#)
  - [BMO Self\\_Helper](#)
  - [BMO Unit Testing](#)
  - [FreePBX Helper](#)
  - [HTML Output from BMO](#)
- 
- [Adding Floating Right Nav to Your Module](#)
  - [Adding Form Buttons to Your Module \(Action Bar\)](#)
  - [BMO Search Provider](#)
  - [Database 13](#)
  - [FWConsole Development](#)
  - [Global Functions in 13 and Higher](#)
  - [Hooking Bulk Handler](#)
  - [Javascript Functions and Variables](#)
  - [PHP Console and Debugging](#)

- Process Management
- Third Party Module Hosting
- Uniformity Guidelines

## External libraries available for use in Framework

Please note some of these may be abstracted out into BMO functions. It is recommended that you use the BMO functions where available.

- <https://github.com/Respect/Validation>
- <https://github.com/rmccue/Requests> FreePBX::Curl()->requests(\$url)
- <https://github.com/educoder/pest> FreePBX::Curl()->pest(\$url)
- <https://github.com/guzzle/guzzle>
- <https://github.com/symfony/filesystem>
- <https://github.com/symfony/process>
- <https://github.com/briannesbitt/Carbon>
- <https://github.com/fightbulc/moment.php>
- <https://github.com/swiftmailer/swiftmailer> FreePBX::Mail();
- <https://github.com/ramsey/uuid>

- BMO Built-In Classes 14+
- BMO Methods
- Generating a UUID
- Hooking Calendar
- Sending Email

- Backup 15.x+
- BMO Methods 15
- Filestore Module
- Included composer Libraries Framework
- Module Hooks (Updates)
- PBX API