

Migrations

All the common extension types for HUBzero can include their own migrations directory. Migrations are used for installing the extension into the required tables for the CMS to know about said extension's existence, installing any needed tables, installing sample data, etc.

To illustrate the typical component directory structures and files:

```
/app
.. /modules
.. .. /mod_example
.. .. .. /migrations
.. .. .. .. /Migration20190301102219ModExample.php
.. .. .. /tmpl
.. .. .. .. default.php
.. .. .. helper.php
.. .. .. mod_example.php
.. .. .. mod_example.xml
```

See the [Migrations documentation](#) for more about naming conventions, setup, etc.

Modules typically have at least one initial migration for registering the extension with the CMS. This migration typically just involves calling the addModuleEntry helper method:

```
<?php

use HubzeroContentMigrationBase;

// No direct access
defined('_HZEXEC_') or die();

/**
 * Migration script for registering the example plugin
 */
class Migration20190301102219ModExample extends Base
{
    /**
     * Up
     */
    public function up()
    {
        // Register the module
```

MIGRATIONS

```
//
// @param string $element (required) Module element
// @param int $enabled (optional, default: 1) Whether o
r not the module should be enabled
// @param string $params (optional) Plugin params (if alr
eady known)
// @param int $client (optional, default: 0) Client [site=0,
admin=1]
$this->addModuleEntry('mod_example');
}

/**
 * Down
 */
public function down()
{
    $this->deleteModuleEntry('mod_example');
}
}
```

That's all there is to it! The `addModuleEntry` method adds the necessary entries to the needed database tables for the CMS to be aware of the module's existence.

For a module to appear anywhere, a new instance of that module must be created and assigned to a position. This above migration just makes the CMS aware of the module's presence so that a new instance *can* be created.