

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
.. /components
.. .. /com_example
.. .. .. /admin
.. .. .. /api
.. .. .. /helpers
.. .. .. /migrations
.. .. .. .. /Migration20190301102219ComExample.php
.. .. .. .. /Migration20190301102256ComExample.php
.. .. .. .. /Migration20190301102301ComExample.php
.. .. .. /models
.. .. .. /site
.. .. .. /tests
```

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

Components typically have one to three initial migrations: one for registering the component with the CMS, one for installing any tables specific to the component, and one for installing any default or sample data. While all of this can be done in one migration, it's typically broken into the three to allow for easier (re-)running of the migrations in steps. Next, we'll go through examples of each common migration.

A migration for registering the component with the CMS typically just involves calling the `addComponentEntry` helper method:

```
<?php

use Hubzero\Content\Migration\Base;

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

/**
 * Migration script for registering the example component
```

MIGRATIONS

```
/**/
class Migration20190301102219ComExample extends Base
{
    /**
     * Up
     */
    public function up()
    {
        // Register the component Note the 'com_' prefix is optional.
        //
        // @param string $name (required) Component name
        // @param string $option (optional) com_xyz
        // @param int $enabled (optional, default: 1) Whether or not the component should be enabled
        // @param string $params (optional) Component params (if already known)
        // @param bool $createMenuItem (optional, default: true) Create an admin menu item for this component
        $this->addComponentEntry('example');
    }

    /**
     * Down
     */
    public function down()
    {
        // Provide the name of the component. Note the 'com_' prefix is optional.
        $this->deleteComponentEntry('example');
    }
}
```

That's all there is to it! The `addComponentEntry()` adds the necessary entries to the needed database tables for the CMS to be aware of the component's existence. Next, we'll look at a typical table installation migration.

```
<?php

use Hubzero\Content\Migration\Base;

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

MIGRATIONS

```
/**
 * Migration script for registering the example component
 **/
class Migration20190301102256ComExample extends Base
{
    /**
     * Up
     **/
    public function up()
    {
        if (!$this->db->tableExists('#__example_entries'))
        {
            $query = "CREATE TABLE `#__example_entries` (
                `id` int(11) unsigned NOT NULL AUTO_INCREMENT,
                `title` varchar(255) NOT NULL DEFAULT '',
                `alias` varchar(255) NOT NULL DEFAULT '',
                `content` text NOT NULL,
                `created` datetime DEFAULT NULL,
                `created_by` int(11) unsigned NOT NULL DEFAULT '0',
                `state` tinyint(2) NOT NULL DEFAULT '0',
                `params` tinytext NOT NULL,
                `access` tinyint(3) NOT NULL DEFAULT '0',
                PRIMARY KEY (`id`),
                KEY `idx_created_by` (`created_by`),
                KEY `idx_alias` (`alias`)
            ) ENGINE=InnoDB DEFAULT CHARSET=utf8;";

            $this->db->setQuery($query);
            $this->db->query();
        }
    }

    /**
     * Down
     **/
    public function down()
    {
        if ($this->db->tableExists('#__example_entries'))
        {
            $query = "DROP TABLE IF EXISTS `#__example_entries`;";
            $this->db->setQuery($query);
            $this->db->query();
        }
    }
}
```

MIGRATIONS

In the `up()` method, we check if the table exists first and, if not, we add it. Existence checks for tables, keys, columns, etc. are essential to ensure migrations can be run and re-ran without conflicts. Also note that in all our migrations we undo everything from the `up()` in the `down()` method. This allows for rolling back a migration should the need arise.

Finally, we'll take a look at a migration that installs sample data.

```
<?php

use Hubzero\Content\Migration\Base;

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

/**
 * Migration script for registering the example component
 */
class Migration20190301102301ComExample extends Base
{
    /**
     * Up
     */
    public function up()
    {
        if ($this->db->tableExists('#__example_entries'))
        {
            // Check if the entry already exists
            $query = "SELECT * FROM `#__example_entries` WHERE `alias`
='sample' AND `created_by`='1000'";
            $this->db->setQuery($query);
            $id = $this->db->loadResult();

            if (!$id)
            {
                $now = new HubzeroUtilityDate('now');

                $query = "INSERT INTO `#__example_entries` (
                    `title`, `alias`, `content`, `created`, `created_by`,
                    `state`, `params`, `access`) VALUES ('Sample', 'sample', 'Sample cont
ent!', $now->toSql(), '1000', '1', '', '1');

                $this->db->setQuery($query);
                $this->db->query();
            }
        }
    }
}
```

MIGRATIONS

```
    }
}

/**
 * Down
 **/
public function down()
{
    if ($this->db->tableExists('__example_entries'))
    {
        $query = "DELETE FROM `__example_entries` WHERE `alias`='
sample' AND `created_by`='1000';
        $this->db->setQuery($query);
        $this->db->query();
    }
}
}
```