Installing

Installing From Package

Warning: Most HUBs do **not** have public write access to the various extensions directories. Using this method may fail as a result. Contact your system administrator for any necessary changes.

The CMS provides a convenient Installer utility in the administrative back-end. From here, one can install new extensions that have been packaged with composer.json files. The installer downloads the code and moves all the necessary files to their appropriate locations.

1. Log in to the administrative back-end of the HUB you wish to install the extension on.
2. Once logged-in navigate to "Extensions Manager" in the drop-down under "Extensions". From there, you should be presented with a choice of "Packages" and "Repositories".

3. First and foremost, a composer.json file must be added to your hub's /app directory. If the system does not find one, it should provide the necessary code sample and instructions on where to place the file. This composer.json file is separate from the one you added to your extension and is, instead, used to tell Composer where to get the necessary libraries used to install hub extensions.

   The contents of that file should look like the following:

   ```
   {
    "name": "hubzero/hubzero-app",
    "description": "App directory",
    "repositories": {
     "hz-installer": {
      "type": "github",
      "name": "Hubzero Extension Installer",
      "description": "Custom installer for Hubzero extensions",
      "url": "https://github.com/hubzero/extension-installer",
      "options": {
      }
     }
    },
    "config": {
     "github-protocols": ["https"],
     "github-domains": ["www.github.com"],
     "preferred-install": {
      "*": "dist"
     },
     "cache-dir": "tmp/cache"
   ```

```
      },
      "require": {
       "hubzero/extension-installer": "dev-master"
      }
     }
```

4. To install a package, go to Administrator > Extensions > Extension Manager and find "Packages" in the sub-menu. From there, you should be presented with a choice of "Packages" and "Repositories". First, we need to register a repository and then select a package from the branches or tags listed for the repository we added.
5. Go the "Repositories" sub-menu item and click the "New" toolbar button. Here you will provide some information about your repository (it's considered good practice to name it the same as the extension you're installing) and the URL to that extension's GitLab or GitHub repository.
6. Once the repository has been added, go to the "Packages" sub-menu and click the "New" toolbar button. From here you will be able to select the repository you just added. Click "install package" and Composer will download the code and move it to the correct location.

Note: The most common issue found when installing extensions is incorrect file permissions. If you have any issues or an extension fails to install, first check that permissions are set accordingly.

Installing By Hand

Installing an extension by hand requires a few more steps than the Extensions Installer but is still a fairly easy and quick process.

1. If the extension is packaged as a .zip file, extract the files to a location on your local machine.

2. Upload the entire contents of the extension, except language files, via SSH/sFTP to the /yourhub/app/{ExtensionType}/ directory.

Extension Type	Install Location
Component	/yourhub/app/components/{ExtensionName}
Module	/yourhub/app/modules/{ExtensionName}
Plugin	/yourhub/app/plugins/{PluginType}/{PluginName}
Template	/yourhub/app/templates/{ExtensionName}

See Accessing Files for further details on how to use SSH/sFTP.

3. Log in to the administrative back-end of the HUB.

4. **Components**

    1. Components do not technically need a database entry to function in their simplest form. However, an entry is needed if one wishes to use parameters or have the component appear under the "Components" list in the administrative back-end. The preferred method is to create a migration for your extension and to then run migrations via the command-line utility muse. The alternative is to have it done by hand via MySQL command-line, some form of MySQL database GUI, or executing a PHP script. A sample SQL is provided below:

```
INSERT INTO #__extensions(
 `extension_id`,
 `name`,
 `type`,
 `element`,
 `folder`,
 `client_id`,
 `enabled`,
 `access`,
 `protected`,
 `manifest_cache`,
 `params`,
 `custom_data`,
 `system_data`,
 `checked_out`,
 `checked_out_time`,
 `ordering`,
 `state`
)
VALUES(
 '',
 'com_mycomponent',
 'component',
 'com_mycomponent',
 '',
 0,
 1,
 1,
 0,
 '',
 '',
 '',
 '',
 0,
 '0000-00-00 00:00:00',
 0,
 0
);
```

See Direct Database Access for further details on how to access a HUB's database via command-line or GUI utility.

Modules

1. Once logged-in navigate to the Modules Manager. This can be found from the main menu by following the "Modules Manager" option found in the drop-down under "Extensions".
2. Click the "New" button in the toolbar. This will present you with a list of all available modules, including those with existing directories but no database entries (such as the one you just copied to /yourhub/app/modules/).
3. Find the name of your newly added module and click its radio button. Once selected, click the "Next" button in the toolbar. This will take you to an "edit module" screen where you may enter a title, adjust parameters, select a position, etc.
4. Enter a title, adjust parameters, select a position, and enter any other necessary information. Click "Save" in the toolbar.

Plugins

1. Unlike modules, there is no convenient Joomla! utility to create the necessary database entry for us. This must be done by hand via MySQL command-line, some form of MySQL database GUI, or executing a PHP script. A sample SQL is provided below:

   ```
   INSERT INTO #__extensions(
    `extension_id`,
    `name`,
    `type`,
    `element`,
    `folder`,
    `client_id`,
    `enabled`,
    `access`,
    `protected`,
    `manifest_cache`,
    `params`,
   ```

```
 `custom_data`,
 `system_data`,
 `checked_out`,
 `checked_out_time`,
 `ordering`,
 `state`
)
VALUES(
 '',
 'System - Hello World',
 'plugin',
 'helloworld',
 'system',
 0,
 1,
 1,
 0,
 '',
 '',
 '',
 '',
 0,
 '0000-00-00 00:00:00',
 0,
 0
);
```

See Direct Database Access for further details on how to access a HUB's database via command-line or GUI utility.

Templates

1. Once logged-in navigate to the Templates Manager. This can be found from the main menu by following the "Template Manager" option found in the drop-down under "Extensions".
2. Here you will be presented with a list of available templates. Your newly added template should be available. To make it the default template of the site, select it by clicking the radio button next to its name.
3. Click the "Default" button to make the template the default.