

Packaging

Overview

It is possible to install a module manually by copying the files using an SFTP client and modifying the database tables. It is more efficient to create a package file in the form of a [composer.json](#) document that will allow the Installer to do this for you. This package file resides in the top-level of your module's directory and contains a variety of information:

- basic descriptive details about your module (i.e. name), and optionally, a description, copyright and license information.
- the extension type (component, module, plugin, template)
- optionally, a destined install directory

Composer Manifest

This `composer.json` file just outlines basic information about the module such as the owner, version, etc. for identification by the installer and then tells the installer which files should be copied and installed.

A typical component manifest:

```
{
  "name": "myorg/mod_example",
  "description": "Example module",
  "license": "MIT",
  "type": "hubzero-module"
}
```

The hub includes some extra code that tells Composer where/how to install extensions, so it's important to use the designated types. Available types are: `hubzero-component`, `hubzero-module`, `hubzero-plugin`, `hubzero-template`.

Structure

Packaging a module for distribution is relatively easy. The file and directory structure is exactly as it would be after installation. Here's what a typical package will look like:

```
/mod_{name}
```

```
mod_{name}.xml
mod_{name}.php
composer.json
/tmpl
    default.php
```

XML Manifest (deprecated)

All modules should include a manifest in the form of an XML document named the same as the module. This file lines out basic information about the module such as the owner, version, etc. for identification by the installer and then provides optional parameters which may be set in the Module Manager and accessed from within the module's logic to fine tune its behavior. Additionally, this file tells the installer which files should be copied and installed.

A typical module manifest:

```
<?xml version="1.0" encoding="utf-8"?>
<extension type="module" version="1.0.0">
  <!-- Name of the Module -->
  <name>mod_listnames</name>

  <!-- Name of the Author -->
  <author>HUBzero</author>

  <!-- Version Date of the Module -->
  <creationDate>2015-06-23</creationDate>

  <!-- Copyright information -->
  <copyright>All rights reserved by HUBzero 2015.</copyright>

  <!-- License Information -->
  <license>GPL 2.0</license>

  <!-- Author's email address -->
  <authorEmail>support@hubzero.org</authorEmail>

  <!-- Author's website -->
  <authorUrl>hubzero.org</authorUrl>

  <!-- Module version number -->
  <version>1.0.0</version>

  <!-- Description of what the module does -->
```

PACKAGING

```
<description>MOD_LISTNAMES_DESCRIPTION</description>

<!-- Listing of all files that should be installed for the module to
function -->
<files>
  <!-- The "module" attribute signifies that this is the main controll
er file -->
  <filename module="mod_listnames">mod_listnames.php</filename>
  <filename>index.html</filename>
  <filename>helper.php</filename>
  <filename>tmpl/default.php</filename>
  <filename>tmpl/index.html</filename>
</files>

<languages>
  <!-- Any language files included with the module -->
  <language tag="en-GB">en-GB.mod_listnames.ini</language>
</languages>

<!-- Optional parameters -->
<config>
  <fields name="params">
    <fieldset name="basic">
      <!-- parameter to allow placement of a module class suffix for the
module table / xhtml display -->
      <field name="moduleclass_sfx" type="text" default="" label="MOD_LI
STNAMES_PARAM_CLASS_LABEL" description="MOD_LISTNAMES_PARAM_CLASS_DESC
" />

      <!-- just gives us a little room between the previous paramter and
the next -->
      <field name="@spacer" type="spacer" default="" label="" descriptio
n="" />

      <!-- A parameter that allows an administrator to modify the number
of users that this module will display -->
      <field name="usercount" type="text" default="5" label="MOD_LISTNAM
ES_PARAM_USERCOUNT_LABEL" description="MOD_LISTNAMES_PARAM_USERCOUNT_D
ESC" />
    </fieldset>
  </fields>
</config>
</extension>
```

PACKAGING

Note: Notice that we DO NOT include a reference in the files section for the XML file.

Let's go through some of the most important tags:

EXTENSION

The extension tag has several key attributes. The type must be "module".

NAME

You can name the module in any way you wish.

FILES

The files tag includes all of the files that will be installed with the module.

CONFIG

Any number of parameters can be specified for a module.