

Gulp build system for Joomla!

This build system works exactly the same as the [joomla-gulp](#) system developed by Roberto Segura, in fact I have based my development on it to create this system. The difference can be found when adding new extensions to the system. You don't need to create new code for each extension, ~~just copy and paste the foo.js file and rename it with your extension name~~ when a new extension is added to the **extensions-config.json** file it will be added automatically to the file system.

This development works with gulp version 4

Getting started

- The first thing you need is node. For installation you can go to [node.js](#).
- Next you need gulp. Once you have node installed you can follow the installation instructions [here](#).
- Now you must clone this repository. Path to the repository folder and type `npm install`, it will automatically install all the dependencies needed to make the system work.

At this point everything is ready for you to start working on your new project with Joomla!.

Configuring your system

Rename the file `config.json.dist` to `config.json`. This is what the file contents looks like:

```
{
  "extName": "the_in_use_extension_name",
  "extsConfigDir": "main_path_for_all_your_extensions"
}
```

Add your local installation configuration. Can be something like:

```
{
  "extName": "myExtension",
  "extsConfigDir": "/home/username/develop/myExtensions"
}
```

Using this configuration the build system will look at

`/home/username/develop/myExtensions/myExtension/` for a `extensions-config.json` file and a `joomla-gulp-config.json` file to get your extension configuration data.

Rename the file `joomla-gulp-config.json.dist` to `joomla-gulp-config.json` and if we continue with the previous example move it to `/home/username/develop/myExtensions/myExtension/`. This is what the file contents look like:

```
{
  "wwwDir": "path_to_joomla_site",
  "extDir": "your_extensions_path",
  "browserProxy": "probably_localhost"
}
```

You just have to add your local installation configuration. Can be something like:

```
{
  "wwwDir": "/var/www/html/mi_joomla",
  "extDir": "/home/mike/my_joomla_extensions",
  "browserProxy": "localhost"
}
```

Rename the file `extensions-config.json.dist` to `extensions-config.json` and also move it to the your extension path as done with the previous json file. Here is where you tell the system what extensions to look for. This is the structure of this configuration file

```
{
  "components": ["foo"],
  "plugins": {
    "bar": ["foo"],
    "system": ["foo"]
  },
  "modules": {
    "site": ["foo"],
    "admin": ["foo"]
  },
  "packages": ["foo"]
}
```

This is an example configuration and corresponds to the **"joomla-scaffolding-example"** I offer you in this package. You can remove or change the "foo" strings with your real extensions name.

How to add my own extensions?

For these instructions we will assume that we use the configuration described above.

Components

The component name will be "mycomponent".

- I must have a folder named "mycomponent" with the component content at `/home/mike/my_joomla_extensions/components/mycomponent`.
- The `extensions-config.json` must look like:

```
{
  "components": ["mycomponent"]
}
```

- I must copy, paste and rename the ~~./gulp-joomla-extensions/components/foo.js~~ file to ~~./gulp-joomla-extensions/components/mycomponent.js~~

Thats all.

Modules

In Joomla! the modules can be developed for both the backend and the frontend. I'll show you how to add a module for each case. The module name will always be "mymodule".

Backend

- I must have a folder named "mod_mymodule" (It must be preceded by "mod_") with the module content at `/home/mike/my_joomla_extensions/modules/admin/mod_mymodule`
- The `extensions-config.json` must look like:

```
{
  "components": ["mycomponent"],
  "modules": {
    "admin": ["mymodule"]
  }
}
```

- I must copy, paste and rename the ~~./gulp-joomla-extensions/modules/admin/foo.js~~ file to ~~./gulp-joomla-extensions/modules/admin/mymodule.js~~ **Watch out!!**. Even if the folder name is `mod_mymodule`, the file name must be `"mymodule.js"`

Thats all.

Frontend

- I must have a folder named "mod_mymodule" with the module content at `/home/mike/my_joomla_extensions/modules/site/mod_mymodule`
- The `extensions-config.json` must look like:

```
{
  "components": ["mycomponent"],
  "modules": {
    "admin": ["mymodule"],
    "site": ["mymodule"]
  }
}
```

- ~~I must copy, paste and rename the `./gulp-joomla-extensions/modules/site/foo.js` file to `./gulp-joomla-extensions/modules/site/mymodule.js`~~

Thats all.

Plugins

In Joomla! the plugins are grouped by type of plugin, where the system plugins are added to the system folder, the authentication plugins are added to the authentication folder and so on. The name of the plugin will be "myplugin" and it will be of the type system.

- I must have a folder named "myplugin" with the plugin content at `/home/mike/my_joomla_extensions/plugins/system/myplugin`
- The `extensions-config.json` must look like:

```
{
  "components": ["mycomponent"],
  "modules": {
    "admin": ["mymodule"],
    "site": ["mymodule"]
  },
  "plugins": {
    "system": ["myplugin"]
  }
}
```

- ~~I must copy, paste and rename the `./gulp-joomla-extensions/plugins/system/foo.js` file to `./gulp-joomla-extensions/plugins/system/myplugin.js`~~

Thats all.

More

I offer you a test **joomla-scaffolding-example** to serve as a basis for the construction of your packages. Here you can see the folder and file structure required for each of the extensions.

Conclusion

I hope this system is useful to you, and if anything goes wrong let me know here. Enjoy it.