Skip to content

Creating a Static Extensions Repository

To host your own extensions and leverage Blender update system all that is required is to host a static JSON file that is generated by Blender.

JSON

To generate a valid JSON file use the server generate Blender command-line tool:

```
blender --command extension server-generate --repo-dir=/path/to/packages
```

This creates an `index.json` listing from all the .zip extensions found in the *–repo-dir* location.

For more details, read the generated JSON API.

Testing

To test the generated repository, add a new Remote Repository from the Preferences:

- Get Extensions → Repositories → [+] → Add Remote Repository
- Paste the location of the generated JSON as the URL. So the example `/path/to/packages` would use the:
  - `file:///path/to/packages/index.json` on Linux and macOS.
  - `file:///C:/path/to/packages/index.json` on Windows.
  - `file://HOST/share/path/to/packages/index.json` for network shares on Windows.

Tip

Open `file:///` in a web browser and navigate to the repository location and copy that as the remote repository URL.

Extensions Listing HTML

The `server-generate` command can optionally create a simple website using the `--html` argument.

```
blender --command extension server-generate --repo-dir=/path/to/packages --html
```

This creates an `index.html` file ready to use, listing extensions which can be dropped into Blender for installation.

Download Links

In order to support drag and drop for installing from a remote repository, there are a few optional ways to prepare the URLs.

The only strict requirement is that the download URL must end in `.zip`.

You can pass different arguments to the URL to give more clues to Blender about what to do with the dropped URL.

repository:
Link to the JSON file to be used to install the repository in Blender. Supports relative URLs.

platforms:
Comma-separated list of supported platforms. If omitted, the extension will be available in all operating systems.

blender_version_min:
Minimum supported Blender version, e.g., `4.2.0`.

blender_version_max:
Blender version that the extension does not support, earlier versions are supported.

Tip

The more details you provide, the better the user experience.

With the exception of the `repository`, all the other parameters can be extracted from the extensions manifest. Those arguments are to be encoded part of the URL.

Expected format:

```
{URL}.zip?repository={repository}&blender_version_min={version_min}&blender_max=
{version_max_exclusive}&platforms={platform1,platform2}
```

Example from self-hosted repository:

```
http://my-site.com/my-addon.zip?
repository=.%2Findex.json&blender_version_min=4.2.0&platforms=windows-x64
```

Example from the Extensions Platform:

```
https://extensions.blender.org/download/sha256:57a6a5f39fa2cc834dc086a27b7b2e572c12fd14
on-amaranth-v1.0.23.zip?
repository=%2Fapi%2Fv1%2Fextensions%2F&blender_version_min=4.2.0&platforms=linux-
x64%2Cmacos-x64
```

> Note
>
> `%2F` and `%2C` are simply the url-encoded equivalent of `/` and `,` respectively.