Configuration

Bower on GitHub
@bower
Get support on discord
Support on Bountysource

Bower can be configured using JSON in a .bowerrc file. For example:

```
{
  "directory": "app/components/",
  "timeout": 120000,
  "registry": {
    "search": [
      "http://localhost:8000",
      "https://bower.herokuapp.com"
    ]
  }
}
```

Placement & Order

The config is obtained by merging multiple configurations by this order of importance:

- CLI arguments via `--config`
- Environment variables
- Local `.bowerrc` located in the current working directory
- All `.bowerrc` files upwards the directory tree
- `.bowerrc` file located in user's home folder (`~`)
- `.bowerrc` file located in the global folder (`/`)

Example of CLI arguments:

- `--config.endpoint-parser=<parser>`
- `--config.storage.packages=<packages_cache_dir>`

Example of valid environment variables:

- `bower_endpoint_parser` is evaluated as `endpoint-parser`
- `bower_storage__packages` is evaluated as `storage.packages`

Example of valid enviroment variables with Array convention:

- `export bower_registry__search='[http://localhost:8080, http://bower.herokuapp.com]'; bower install`

.bowerrc specification

Available configuration variables, in `.bowerrc` format:

```
{
  "cwd": "~/.my-project",
  "directory": "bower_components",
  "registry":
"https://bower.herokuapp.com",
  "shorthand-resolver":
"git://github.com//.git",
  "proxy": "http://proxy.local",
  "https-proxy": "http://proxy.local",
  "ca": "/var/certificate.pem",
  "color": true,
  "timeout": 60000,
  "save": true,
  "save-exact": true,
  "strict-ssl": true,
  "storage": {
    "packages" : "~/.bower/packages",
    "registry" : "~/.bower/registry",
    "links" : "~/.bower/links"
```

```
  },
  "interactive": true,
  "resolvers": [
    "mercurial-bower-resolver"
  ],
  "shallowCloneHosts": [
    "myGitHost.example.com"
  ],
  "scripts": {
    "preinstall": "",
    "postinstall": "",
    "preuninstall": ""
  },
  "ignoredDependencies": [
    "jquery"
  ]
}
```

A detailed description of available configuration variables can be found in bower/spec repository.

Environment variables in .bowerrc

One can use environment variables in `.bowerrc`, using the following syntax `${ENV_VAR}`.

```
"storage" : {
  "packages":
"/path/to/${USER}/packages"
}
```

Hooks

Bower provides 3 separate hooks that can be used to trigger other automated tools during Bower usage. Importantly, these hooks are intended to allow external tools to help wire up the newly installed components into the parent project and other similar tasks. These hooks are not intended to provide a post-installation build step for component authors. As such, the configuration for these hooks is provided in the `.bowerrc` file in the parent project's directory.

In `.bowerrc` do:

```
{
  "scripts": {
    "preinstall": "<your command
here>",
    "postinstall": "<your command
here>",
    "preuninstall": "<your command
here>"
  }
}
```

The value of each script hook may contain a % character. When your script is called, the % will be replaced with a space-separated list of components being installed or uninstalled.

Your script will also include an environment variable `BOWER_PID` containing the PID of the parent Bower process that triggered the script. This can be used to verify that a `preinstall` and `postinstall` steps are part of the same Bower process.

Help improve these docs. Open an issue or pull request.