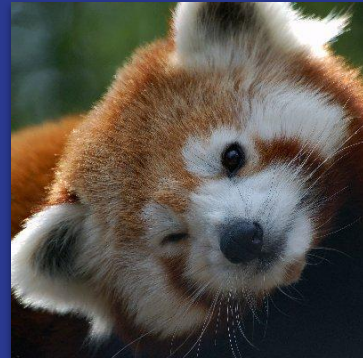# Basalt: Bash Package Management

Edwin Kofler

Twitter: @hyperupcall
GitHub: @hyperupcall

Problem: Code reuse in Bash

- It is difficult to reuse logic across Bash projects
    - Most of the time, "libraries" are simply standalone files
- Existing package managers like bpkg and Basher partially solve the problem
    - Some do not enable 'local' dependencies
    - Some do not properly distinguish between different versions of the dependency
    - Some do not enable declaring properties of a "Bash package" declaratively

```
[package]
name = 'bash-object'
slug = 'bash_object'
version = '0.6.5'
authors = ['Edwin Kofler" <edwin@kofler.dev>']

[run]
dependencies = ['https://github.com/hyperupcall/bats-common-utils.git@v3.0.0']
sourceDirs = ['pkg/lib', 'pkg/lib/public', 'pkg/lib/util']
builtinDirs = []
binDirs = []
completionDirs = []
manDirs = []

[run.shellEnvironment]

[run.setOptions]

[run.shoptOptions]
```

Use Case 1: Installing Global dependencies

1. For example, [Bats](#) is a testing library for Bash
   - Simply globally install the package
2. This works out of the box with nearly all popular Bash projects, including [bash-preexec](#), [shdoc](#), [bashmarks](#), [bashtop](#), [bash2048](#), etc.

```
> basalt global add JosefZIla/bash2048 github.com/bats-core/bats-core
 Downloaded github.com/JosefZIla/bash2048@37da5210d6d70713bc8630cf45c907102a53e3cf (cached)
  Extracted github.com/JosefZIla/bash2048@37da5210d6d70713bc8630cf45c907102a53e3cf (cached)
Transformed github.com/JosefZIla/bash2048@37da5210d6d70713bc8630cf45c907102a53e3cf
 Downloaded github.com/bats-core/bats-core@v1.4.1 (cached)
  Extracted github.com/bats-core/bats-core@v1.4.1 (cached)
Transformed github.com/bats-core/bats-core@v1.4.1
> bats --version
Bats 1.4.1
```

Use Case 1: Local dependencies

```bash
#!/usr/bin/env bash

eval "$(basalt-package-init)"
basalt.package-init
basalt.package-load
basalt.load 'github.com/hyperupcall/bats-common-utils' 'load.bash'
```

1. Downloading local Bash dependencies
   - Use dependencies as you would with `npm`, `cargo`, etc.

```
> basalt init --bare
...
> basalt add hyperupcall/bats-common-utils
...
> basalt install
...
> tree .basalt
.basalt
├── generated
│   └── source_package.sh
└── packages
    └── github.com
        └── hyperupcall
            └── bats-common-utils@v3.0.0 ->
/home/edwin/.local/share/basalt/store/packages/github.com/hyperupcall/bats-com
```

```toml
[package]
name = 'bash-object'
slug = 'bash_object'
version = '0.6.5'
authors = ['Edwin Kofler" <edwin@kofler.dev>']

[run]
dependencies = ['https://github.com/hyperupcall/bats-common-utils.git@v3.0.0']
sourceDirs = ['pkg/lib', 'pkg/lib/public', 'pkg/lib/util']
builtinDirs = []
binDirs = []
completionDirs = []
manDirs = []

[run.shellEnvironment]

[run.setOptions]

[run.shoptOptions]
```

Roadmap and Ecosystem

- Extend support to other runtimes
  - Ksh, Zsh, Fish, Gawk
- Custom builtins
  - Improves performance, potentially better platform agnostic
- `basalt bundle`
  - If sourcing multiple files, concatenate them all into one file (Webpack-esque)

- hyperupcall/basalt
- hyperupcall/awesome-basalt
- hyperupcall/bash-object