

Designing by Principle

A Case Study

Rack

Dockerfile
Gemfile
Gemfile.lock
README.md
Rakefile
app/
bin/
config/
config.ru
db/
docker-stack.yml
docs/
extra/
lib/
log/
node_modules/
package.json
public/
spec/
surveyor.iml
tmp/
vendor/
yarn.lock

Rack

```
# This file is used by Rack-based servers to start the application.
```

```
require_relative 'config/environment'
```

```
run Rails.application
```

Rack...Why?



Principles

- Well-defined interfaces (good fences make good neighbors)
- Extensibility (don't try to predict the future)
- Composition (we're better together)

Well-Defined Interface

```
# my_rack_app.rb
```

```
require 'rack'
```

```
app = Proc.new do |env|  
  ['200', {'Content-Type' => 'text/html'}, ['A barebones rack app.']]  
end
```

```
Rack::Handler::WEBrick.run app
```

```
# config.ru
```

```
run Proc.new { |env| ['200', {'Content-Type' => 'text/html'}, ['get rack\'d']] }
```

Extensibility

```
# config.ru
```

```
use Rack::CommonLogger
```

```
use Rack::Session::Cookie
```

```
run App
```

Composition

References

- Inventing on Principle - Bret Victor
<https://vimeo.com/36579366>
- Simplicity Matters - Rich Hickey
<https://www.youtube.com/watch?v=rI8tNMsozo0>