

# Building web apps with Buffalo



Paweł Słomka  
GoWroc, 21 IX 2017



# Agenda

- Building web apps in Go
- A few words about Buffalo
- Live demo
  - Initialisation
  - Package walkthrough
  - Templates
  - I18n
  - Models, resources
  - Authentication
  - Console
  - Production build

# Agenda

- Building web apps in Go
- A few words about Buffalo
- Live demo

+Dilbert comics  
by Scott Adams

- Initialisation
- Package walkthrough
- Templates
- I18n

- Models, resources
- Authentication
- Console
- Production build

# Building web apps

- Where do I start?

# Building web apps

- Where do I start?
  - *Use the standard library!*

# Building web apps

- Where do I start?
  - *Use the standard library!*
- Reinventing the wheel
  - Routing
  - Templating
  - Logging
  - Authentication
  - Sessions
  - etc.

# *Rolling your own*



# Buffalo

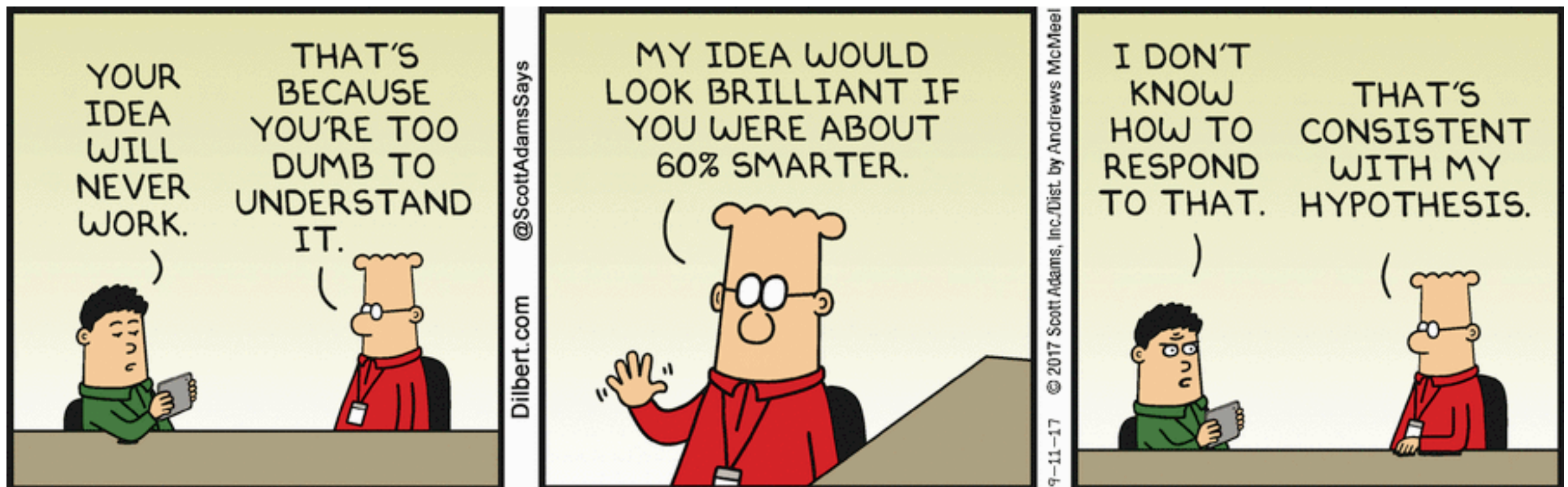
- by Mark Bates ([@markbates](https://twitter.com/markbates))
- Not a version 1.0 yet
  - 0.9.4 at the time of writing
  - Evolves dynamically
- From v1 they want to provide Go-like backward compatibility



# Buffalo

- It's a framework
- It's extracted, not invented
- Built from smaller blocks
  - Many cool, useful packages
- Makes development faster
  - Takes care of the boring stuff
  - You can focus on your business logic

# Buffalo



# Buffalo

- I'd recommend to use `development` branch
- If Gophercon 2018 ([@bketelsen](#) working on this one) page can be build with Buffalo, so can your webpage



# Getting started

- Starting a new project

```
$ buffalo new gowroc
```

# Walking through

- actions
- assets, public/assets
- gifts
- locales
- models
- node\_modules 🤔
- templates
- database.yml
- README.md
- webpack.config.js

# Walking through

- actions



- templates

- tmp

# Getting started

- Starting a new project

```
$ buffalo new gowroc
```

- Creating databases [all three]

```
$ buffalo db create [-a]
```

- Running the server

```
$ buffalo dev
```



# Templates

- Using `markbates/plush`
- Injecting values
  - `<%= SomeValue %>`
- Flow control
  - `<%= for (x) in xs { ... } %>`



# Internationalisation

- Locales files
- Messages in templates
- Page translations
  - Using `Accept-Language` HTTP header
  - Using `lang=xx-zz` cookie
- Values in messages

# Models

- Generating a model

```
$ buffalo db g model field:type
```

- Running migrations

```
$ buffalo db migrate [up]
```

# Resources

- Generating a resource

```
$ buffalo g resource field:type
```

- Running migrations

```
$ buffalo db migrate [up]
```

# Authentication

- Using `markbates/goth`



- Login callback

# Authentication

- Using markbates/goth

```
$ buffalo g goth <provider>
```

- Generating two endpoints:
  - Login
  - Login callback

# REPL Console

- Entering console

```
$ buffalo console
```

# Building a binary

- Entering console

```
$ buffalo build
```





# Thank you!

## Any easy questions?





# Links

- Buffalo
  - Docs - <https://gobuffalo.io/>
  - Sources - <https://github.com/gobuffalo>
- My blog
  - [Getting started](#)
  - [Internationalization](#)
  - [Models, database connection](#)
  - [Generating resources](#)
  - [Authentication with GitHub](#)