

balde: Desenvolvendo aplicações web em C

Rafael G. Martins

10 de maio de 2014

Tópicos

- 1 balde?!
- 2 Como instalar?
- 3 Hello World
- 4 Views
- 5 Contexto de aplicação
- 6 Contexto de requisição
- 7 Contexto de resposta
- 8 Recursos estáticos
- 9 Templates

Quem sou?!

- Engenheiro de controle e automação
- Desenvolvedor do Gentoo Linux
- Desenvolvedor Python no Titans Group

balde?!

- O que quer dizer esse nome?!

balde?!

- O que quer dizer esse nome?!
- Alguém ainda programa em C?!

balde?!

- O que quer dizer esse nome?!
- Alguém ainda programa em C?!
- Isso não deveria estar sendo feito em Python/PHP/Ruby/(insira sua linguagem hipster preferida aqui)?!

Como instalar?

- `http://balde.io/`

Como instalar?

- `http://balde.io/`
- `https://github.com/balde`

Como instalar?

- `http://balde.io/`
- `https://github.com/balde`
- Poucas dependências: FastCGI toolkit, shared-mime-info, PEG.

Como instalar?

- `http://balde.io/`
- `https://github.com/balde`
- Poucas dependências: FastCGI toolkit, shared-mime-info, PEG.
- Funciona com qualquer servidor WEB que suporte CGI/FastCGI.

Como instalar?

- `http://balde.io/`
- `https://github.com/balde`
- Poucas dependências: FastCGI toolkit, shared-mime-info, PEG.
- Funciona com qualquer servidor WEB que suporte CGI/FastCGI.
- `$./autogen.sh && ./configure && make && sudo make install`

Hello world

Live coding!

Views

- Views são chamadas pelo framework para responder a uma requisição.

Views

- Views são chamadas pelo framework para responder a uma requisição.
- Uma view recebe um contexto de aplicação e um contexto de requisição, e retorna um contexto de resposta.

Views

- Views são chamadas pelo framework para responder a uma requisição.
- Uma view recebe um contexto de aplicação e um contexto de requisição, e retorna um contexto de resposta.
- A definição das rotas e métodos HTTP atendidos pela view é feita no momento do registro da view.

Contexto de aplicação

- Armazena toda a informação necessária durante todo o ciclo de vida da aplicação.

Contexto de aplicação

- Armazena toda a informação necessária durante todo o ciclo de vida da aplicação.
- Registra todas as views e realiza o roteamento das URLs e métodos HTTP.

Contexto de aplicação

- Armazena toda a informação necessária durante todo o ciclo de vida da aplicação.
- Registra todas as views e realiza o roteamento das URLs e métodos HTTP.
- Pode armazenar dados arbitrários da aplicação, para serem facilmente acessados pelas views, se necessário.

Contexto de requisição

- Armazena toda a informação proveniente de uma requisição HTTP feita por um cliente.

Contexto de requisição

- Armazena toda a informação proveniente de uma requisição HTTP feita por um cliente.
- Existem funções especiais para manipular os dados armazenados neste contexto.

Contexto de resposta

- Deve ser construído na view para armazenar toda a informação necessária para construir a resposta da requisição HTTP.

Contexto de resposta

- Deve ser construído na view para armazenar toda a informação necessária para construir a resposta da requisição HTTP.
- Existem funções especiais para inserir os dados neste contexto.

Recursos estáticos

- Recursos estáticos da aplicação, como arquivos CSS, imagens e scripts JavaScript, podem ser convertidos para código C e linkados com o binário final da aplicação.

Recursos estáticos

- Recursos estáticos da aplicação, como arquivos CSS, imagens e scripts JavaScript, podem ser convertidos para código C e linkados com o binário final da aplicação.
- Utiliza os objetos GResource da GLib e o utilitário glib-resources-compile.

Recursos estáticos

- Recursos estáticos da aplicação, como arquivos CSS, imagens e scripts JavaScript, podem ser convertidos para código C e linkados com o binário final da aplicação.
- Utiliza os objetos GResource da GLib e o utilitário `glib-resources-compile`.
- Todos os recursos estáticos presentes na aplicação são registrados automaticamente num endpoint específico (`/static/`).

Templates

- Engine de templates "logic-less".

Templates

- Engine de templates "logic-less".
- É capaz de manipular variáveis, strings, inteiros, reais e chamadas simples de funções.

Templates

- Engine de templates "logic-less".
- É capaz de manipular variáveis, strings, inteiros, reais e chamadas simples de funções.
- Todos os templates são convertidos em código C, que devem ser compilados e linkados com o binário final da aplicação.

Perguntas?!

- Email: rafael@rafaelmartins.eng.br
- Blog: <http://rafaelmartins.eng.br>
- IRC: rafaelmartins @ Freenode