Tanz!

Stefan Hornburg (Racke)
racke@linuxia.de

13. Deutscher Perl-Workshop, Frankfurt, 21. Oktober 2011

Tanzflur

- einfach
- ausdrucksstark
- effektiv
- minimale Abhängigkeiten

Projekte

- Dropbox-Klon
- LDAP Benutzerverwaltung
- Procurement für American Corners

Dropbox

- Autoindex
- Upload/Download
- Benutzerverwaltung

LDAP Benutzerverwaltung

- LDAP-Verzeichnis
- Benutzerverwaltung
  - Adminstrator
  - Referrer
  - Patron

Procurement für American Spaces

- Warenkorb/Wunschliste
- Genehmigungsprozesse
- LibraryThing
- ISBNDB

`https://vsc.state.gov/`

Rezept für Routes

```perl
#! /usr/bin/env perl

use Dancer;

get '/' => sub {
    template 'index';
};

dance;
```

Routes

- String
- String mit benannten Parametern
- String mit Wildcards
  - Splat
  - Megasplat
- Regulärer Ausdruck
- Regulärer Ausdruck mit Captures

Splat

```
get '/images/covers/*.jpg' => sub {
    my ($isbn) = splat;

    if (-f "public/images/covers/$isbn.jpg") {
        return send_file "images/covers/$isbn.jpg";
    }

    status 'not_found';
    forward 404;
}
```

Megasplat

```
https://vsc.state.gov/lostpwd/biz@linuxia.de/e642bd543b9907bd2c06aa4

get '/lostpwd/**' => sub {
    my ($email, $hash) = splat;

    form->fill(email => $email,
               hash => $hash);

    template('lostpwd_confirm', form => $form);
}
```

Captures

```
https://dropbox.nite.si/~racke/talks/dancer-beamer.pdf

any qr{^/~(?<user>[^/]+)/(?<file>.*?)/?$} => sub {
    my ($capts, $user, $file);

    $capts = captures;
    $file  = $capts->{file};
    $user  = $capts->{user};

    ...
};
```

Präfix

```
prefix '/wiki';

get qr{/?(?<page>.*)} => sub {
    my $capts = captures;

    template 'wiki', {content => wiki($capts->{page}),
                      page => $capts->{page}};
};
```

Filters

```
before sub {
};

after sub {
};
```

before

```
before sub {
    unless (session('user')
        || request->path eq '/login'
        || request->path =~ m%^/lostpwd%
        ) {
        redirect '/login';
    }
};
```

Hooks

```
hook before_template_render => sub {
    my $tokens = shift;
    my $menu;

    if (session('user')) {
        $menu = [{name => 'Logout', url => '/logout'}];
    }
    else {
        $menu = [{name => 'Login', url => '/login'}];
    }

    $tokens->{menu} = $menu;
};
```

Plugins

- Funktionalität
- Schlüsselwort
- Route
- Konfiguration

Plugins auf CPAN

- Dancer::Plugin::Database
  `database`
- Dancer::Plugin::Email
  `email`
- Dancer::Plugin::Ajax
  `ajax`

Dancer::Plugin::LibraryThing Konfiguration

```
plugins:
    LibraryThing:
        api_key: d231aa37c9b4f5d304a60a3d0ad1dad4
        directory: public/images/covers
        size: large
```

Dancer::Plugin::LibraryThing Code

```
use Dancer::Plugin::LibraryThing;

get '/images/covers/*.jpg' => sub {
    my ($isbn, @ret);
    $isbn = splat;

    unless (-f "public/images/covers/$isbn.jpg") {
        @ret = librarything_cover($isbn);
        if (@ret < 3) {
            status 'not_found';
            forward 404;
        }
    }

    return send_file "images/covers/$isbn.jpg";
}
```

Writing Plugins

- register
- register_plugin
- plugin_setting

Plugins und Hooks

```
Dancer::Factory::Hook
        ->instance
        ->install_hooks('before_cart_add');
```

Sessions

- Engine

  ```
  session: Storable
  ```

- Verzeichnis

  ```
  session_dir: /var/run/dancer-sessions
  ```

- Ablauffrist

  ```
  session_expires: 8 hours
  ```

Logs

- Engine
  ```
  logger: file
  ```
- Verzeichnis
  ```
  log_path: log
  ```
- Format
  ```
  logger_format: "%t [%P] %L @%D> %m in %f l. %l"
  ```

Szenarien

- Standalone

  ```
  ./bin/app.pl
  ```

- CGI, FastCGI
- plackup
  - Starman
  - Twiggy
  - Corona
- Reverse Proxy

Szenarien: Reverse Proxy

- Apache
- nginx
- Perlbal

Starman

```
plackup –E production
        –s Starman \
        —workers=5 \
        —pid /home/racke/Dropbox/run/Dropbox.pid \
        –p 5000 \
        –a bin/app.pl \
        –D
```

Perlbal

```
LOAD vpaths
XS enable headers
```

Perlbal: Pools

```
CREATE POOL prod_starman_dosqua
 POOL prod_starman_dosqua ADD 127.0.0.1:5000

CREATE POOL prod_starman_vsc
 POOL prod_starman_vsc ADD 127.0.0.1:5001
```

Perlbal: Reverse Proxy

```
CREATE SERVICE vsc_prod
  SET role                = reverse_proxy
  SET pool                = prod_starman_vsc
  SET buffer_uploads      = on
ENABLE vsc_prod
```

Perlbal: Selector

```
CREATE SERVICE vsc_selector
 SET listen              = 86.59.13.238:80
 SET role                = selector
 SET plugins             = vpaths
 VPATH .*                = vsc_prod
ENABLE vsc_selector
```

Perlbal: SSL Selector

```
CREATE SERVICE vsc_ssl_selector
 SET listen               = 86.59.13.238:443
 SET role                 = selector
 SET plugins              = vpaths
 SET enable_ssl           = on
 SET ssl_key_file         = /etc/ssl/private/vsc.state.gov.key
 SET ssl_cert_file        = /etc/ssl/certs/vsc.state.gov.pem
 VPATH .*                 = vsc_prod
ENABLE vsc_ssl_selector
HEADER vsc_ssl_selector INSERT X–Forwarded–Proto: HTTPS
```

Skripts

```
use Dancer ':script';

set logger => 'console';
set logger_format => '%m';
```

Dancer2

- keine globalen Variablen
- 100% OO Backend (Moo)
- Scoping for Sub-Applikationen
- überarbeitete Architektur

Community

Web: `http://perldancer.org/`
Github: `git://github.com/sukria/Dancer.git`
IRC: #dancer @ irc.perl.org
Mitarbeit: Dancer::Development

The End

Slides, Handout, Skripte:

`http://www.linuxia.de/talks/pws2011/`