

Dancer and DBIx::Class

Stefan Hornburg (Racke)
racke@linuxia.de

Czech Perl Workshop, Prague, 21st May 2014

Easy to start with

- ▶ Application ready to go
- ▶ Syntax easy to understand
- ▶ Routes and Keywords

Easy to start with

- ▶ `cpanm Dancer YAML`
- ▶ `dancer -a Dropbox`
- ▶ `cd Dropbox`
- ▶ `./bin/app.pl`

Program

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

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

```
use Dancer;
```

```
use Dropbox;
```

```
dance;
```

Module

lib/Dropbox.pm

```
package Dropbox;  
use Dancer ':syntax';  
  
our $VERSION = '0.1';  
  
get '/' => sub {  
    template 'index';  
};  
  
true;
```

Templates

Layout

`views/layouts/main.tt`

Content

`views/index.tt`

Templates

- ▶ Normal Layout

```
template 'index', {name => 'Test'}
```

- ▶ Specific Layout

```
template 'index', {name => 'Test'}, {layout => 'test'}
```

- ▶ No Layout

```
template 'index', {name => 'Test'}, {layout => undef}
```

Routes and Keywords

- ▶ HTTP method
 - ▶ get
 - ▶ post
 - ▶ ...
 - ▶ any
- ▶ Path
- ▶ Subroutine

Routes

- ▶ String
- ▶ Named parameters
- ▶ Wildcards
 - ▶ Splat
 - ▶ Megasplat
- ▶ Regular expression

String

```
get '/home' => sub {  
    my $files = autoindex('/');  
  
    template 'filebrowser', {directory => 'Home',  
                             files => $files,  
                             };  
};
```

Named parameters

```
get '/home/:file' => sub {  
    my $files = autoindex(param('file'));  
  
    template 'filebrowser', {directory => param('file'),  
                             files => $files,  
                             };  
};
```

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://eshop.state.gov/lostpwd/biz@linuxia.de/e642bd543f`

```
get '/lostpwd/**' => sub {  
  my ($email, $hash) = splat;  
  
  form->fill(email => $email,  
             hash => $hash);  
  
  template('lostpwd_confirm', form => $form);  
}
```

Regular Expression

Catch-All (last route!)

```
any qr{.*} => sub {  
    ...  
};
```

Keywords

- ▶ get, post, any, put, del, ...
- ▶ request, params, param
- ▶ redirect, forward, status, header
- ▶ config, var, session
- ▶ from_json, to_json, from_xml, to_xml

var(s) and session

Storing and retrieving data for the current request:

```
var bar => 'pivo';  
$bar = var 'bar';  
$bar = vars->{bar};
```

Storing and retrieving data from the session:

```
session username => 'racke@linuxia.de';  
  
if (! session('username')) {  
    redirect uri_for('/login');  
}
```


Introduction

Dancer::Plugin::DBIC

Dancer::Session::DBIC

TableEditor

Forecast and Contribution

Dancer

DBIx::Class

Database Administration

Easy to expand

- ▶ Plugins
- ▶ Hooks
- ▶ Engines

before Hook

Password protected site:

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

Solid

- ▶ Stable
- ▶ Keep behaviour
- ▶ Community

Applications

- ▶ Simple Dropbox `https://metacpan.org/pod/Dancer::Plugin::Dropbox`
- ▶ .state.gov Websites
`https://eshop.state.gov/`
- ▶ Monitor for Power Plant

DBIx::Class

- ▶ ORM
- ▶ Objects instead of SQL
- ▶ Performance

DBIx::Class

- ▶ Database => Schema
- ▶ interchange6 => Interchange6::Schema
- ▶ Table => Result classes
- ▶ users => Interchange6::Schema::Result::User
- ▶ Queries => Result sets

User and Roles

- User
- ▶ racke@linuxia.de
 - ▶ info@nite.si
 - ▶ test@linuxia.at
- Role
- ▶ user
 - ▶ editor
 - ▶ admin
 - ▶ guest

Tables

- ▶ users
 - ▶ users_id
 - ▶ email
 - ▶ first_name
 - ▶ ...
- ▶ roles
 - ▶ roles_id
 - ▶ name
 - ▶ label
- ▶ user_roles
 - ▶ users_id
 - ▶ roles_id

Roles for an user

```
mysql> select R.name from users U
       join user_roles UR on (U.users_id = UR.users_id)
       join roles R on (UR.roles_id = R.roles_id)
       where U.email = 'racke@linuxia.de';
```

+	-----	+
	name	
+	-----	+
	user	
	editor	
+	-----	+

User with DBIx::Class

```
$rs = $schema->resultset( 'User' );  
$user = $rs->find(1);  
$user = $rs->find({ email => 'racke@linuxia.de' });  
  
$first_name = $user->first_name;  
  
$users_linuxia = $rs->search({  
    email => { like => '%@linuxia.de' } });
```

Roles with DBIx::Class

```
$rs = $schema->resultset( 'User' );  
$user = $rs->find( { email => 'racke@linuxia.de' } );  
  
$roles = $user->roles;
```

User Result Class

```
package Interchange6 :: Schema :: Result :: User;
```

```
__PACKAGE__->table("users");
```

```
__PACKAGE__->add_columns(...);
```

```
__PACKAGE__->set_primary_key("users_id");
```

User Result Class

```
package Interchange6::Schema::Result::User;
```

```
__PACKAGE__->has_many( "UserRole",  
    "Interchange6::Schema::Result::UserRole",  
    { "foreign.users_id" => "self.users_id" },  
);
```

```
__PACKAGE__->many_to_many( "roles", "UserRole", "Role" );
```

Object inflation

```
debug "Role: ", $role;
```

```
debug "Role: ", {$role->get_inflated_columns};
```

```
Role: { 'label' => 'User',  
        'name' => 'user',  
        'roles_id' => '3' }
```

Object vs Hashref

- ▶ Debug / Logs
- ▶ Templates
- ▶ API / JSON
- ▶ Speed

Database Administration

- ▶ phpmyadmin
- ▶ phppgadmin
- ▶ TableEditor

TableEditor Features

- ▶ Different database systems
MySQL, PostgreSQL, ...
- ▶ higher level of abstraction
- ▶ modern frontend
- ▶ concise source code
- ▶ “simple” installation

Input Database Parameters

Database configuration

Database Driver

Pg

Database Name

interchange6

Username

racke

Password

Schema class

Interchange6::Schema

Optional. If you don't specify existing DBx schema class one will be generated for you.

DSN suffix

Optional. Extra options for DB connection.

Submit

View Products

Configuration

Status

Tables

Address

Attribute

Attribute Value

Cart

Country

Group Pricing

Inventory

Media

Media Display

Media Product

Media Type

Merchandising

Attribute

Merchandising

Product

Navigation

Navigation Attribute

Navigation Attribute

Value

Order

- . .

Product - List of items

[New Product](#)

Sk	Na	Ur	Pr	Gt	Can	Ac	In	Cr	La	
Sk	Name	Uri	Priority	Gtin	Canonical sku	Active	Inventory exempt	Created	Last modified	2
										Filter
F0001	One Dozen Roses	one-doz-en-roses	0			1	0	2014-03-26T14:15:14	2014-03-26T14:15:14	✕ 🔗
F0001-PINK	One Dozen Pink Roses	one-doz-en-pink-roses	0		F0001	1	0	2014-03-26T14:15:14	2014-03-26T14:15:14	✕ 🔗

15 items found. Page 1 / 8

1 2 3 4 5 6 »

View Product

Configuration

Status

Tables

Address

Attribute

Attribute Value

Cart

Country

Group Pricing

Inventory

Media

Media Display

Media Product

Media Type

Merchandising

Attribute

Merchandising

Product

Navigation

Navigation Attribute

Navigation Attribute

Value

Order

Orderline

Payment Order

Product

F0002 - Product

General

Orderline

Canonical

Product Attribute

Merchandising Product Related

Merchandising Product

Media Products

Inventory

Group Pricings

Media Display

Variant

SKU

F0002

Name

One Dozen Roses & Calla Lily

Short description

What says I love you better than 1 dozen fresh roses with calla lily?

Description

Surprise the one who makes you smile, or express yourself perfectly with this stunning bouquet of one dozen fresh red roses. This elegant

Price

49.95

URI

one-dozen-red-roses-calla-lilly

Weight

Relationship Orderline

Configuration

Status

Tables

Address

Attribute

Attribute Value

Cart

Country

Group Pricing

Inventory

Media

Media Display

Media Product

Media Type

Merchandising

Attribute

Merchandising

Product

Navigation

Navigation Attribute

Navigation Attribute
Value

Order

Orderline

Payment Order

Product

F0002 - Product

General

Group Pricings

Media Display

Variant

Canonical

Product Attribute

Merchandising Product Related

Orderline

Merchandising Product

Inventory

Media Products

Related Orderline

[+ New Orderline](#)

Orderlines id	Order	Order position	Product	Name	Weight	Quantity	Status	10 ▾
<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>	Filter
2	1 - Or der	2	F0002 - Produc t	One Dozen Roses & Cal la Lily	5	1		✕ 🔍

1 items found. Page 1 / 1

Overview Dancer::Plugin::DBIC

- ▶ Usage
- ▶ Configuration
- ▶ UTF-8
- ▶ Create schema dynamically

DBIx::Class without Dancer Plugin

```
use Interchange6 :: Schema;
```

```
$schema = Interchange6 :: Schema->connect (...);
```

```
$schema->resultset('User')->search({..});
```

DBIx::Class with Dancer Plugin

```
use Dancer::Plugin::DBIC;  
  
schema->resultset( 'User' )->search ( { .. } );  
  
resultset( 'User' )->search ( { .. } );  
  
rset( 'User' )->search ( { .. } );
```


Configuration

```
plugins :  
  DBIC :  
    default :  
      dsn: dbi:mysql:interchange6  
      user: racke  
      pass: nevairbe  
      schema_class: Interchange6::Schema
```

Multiple Schemas

```
plugins :  
  DBIC :  
    default :  
      dsn: dbi:mysql:interchange6  
      user: racke  
      pass: nevairbe  
      schema_class: Interchange6::Schema  
    legacy :  
      dsn: dbi:mysql:interchange5  
      user: racke  
      pass: nevairbe  
      schema_class: Interchange5::Schema
```

Multiple Schemas

```
use Dancer::Plugin::DBIC;  
  
schema('legacy')->resultset('UserDb')->search({..});
```

UTF-8 for MySQL

```
plugins :  
  DBIC :  
    default :  
      dsn: dbi:mysql:interchange6  
      user: racke  
      pass: nevairbe  
      schema_class: Interchange6::Schema  
      options :  
        mysql_enable_utf8: 1
```

Create schema dynamically

- ▶ `schema_class` missing in configuration
- ▶ `DBIx::Class::Schema::Loader`
- ▶ test and development
- ▶ TableEditor

Overview Dancer::Session::DBIC

- ▶ engines
- ▶ configuration
- ▶ serialization
- ▶ session expiration

Engines

- ▶ Templates
TT, Xslate, Flute, ...
- ▶ Sessions
Storable, Database, DBIC
- ▶ Logger
File, Syslog
- ▶ Serializer
JSON, YAML, XML

Configuration

`session` name of session engine (DBIC)

`session_options` options

`session_expires` expiration date

Configuration

```
session: "DBIC"  
session_options:  
  dsn: dbi:mysql:interchange6  
  user: racke  
  pass: nevairbe  
  schema_class: Interchange6::Schema  
  resultset: Session  
  id_column: sessions_id  
  data_column: session_data  
session_expires: 12 hours
```

Configuration

```
set session => 'DBIC';  
set session_options => {schema => schema};
```

Example table

```
CREATE TABLE 'sessions' (  
  'sessions_id' varchar(255) NOT NULL,  
  'session_data' text NOT NULL,  
  'created' datetime NOT NULL,  
  'last_modified' datetime NOT NULL,  
  PRIMARY KEY ('sessions_id')  
) ENGINE=InnoDB;
```

Serializer

```
set 'session_options' => {  
  schema      => schema,  
  serializer   => sub { YAML::Dump(@_); },  
  deserializer => sub { YAML::Load(@_); },  
};
```

Session expiration

- ▶ remove old sessions from database
- ▶ `Interchange6::Schema::Resultset::Session`

```
$schema->resultset( 'Session' )->expire( '12 hours' );
```

Overview TableEditor

- ▶ Installation
- ▶ Frontend
- ▶ Routes
- ▶ Login
- ▶ Relationships
- ▶ Configuration

Installation

```
git clone https://github.com/interchange/TableEditor
cd TableEditor
cpanm .
./bin/app.pl
```

Driver

- ▶ DBD::mysql
- ▶ DBD::Pg
- ▶ ...

Frontend

- ▶ Angular
 - ▶ Routes for the frontend
 - ▶ XHR requests to REST API
 - ▶ JSON
- ▶ Bootstrap
- ▶ Theme

Routes

```
get('/:class/:id' => require_login sub {  
  # retrieve database record and add relationships  
  ...  
  
  return to_json($data, {allow_unknown => 1});  
};
```

Login

- ▶ Dancer::Plugin::Auth::Extensible
- ▶ Provider
 - ▶ Unix
 - ▶ DBIC
- ▶ Database (*planned*)

Relationships

- ▶ belongs_to
- ▶ has_many
- ▶ might_have
- ▶ has_one
- ▶ many_to_many
needs to be configured

Configuration

- ▶ Auth::Extensible
- ▶ DBIC
 - ▶ default

Planned Features

- ▶ Search (Solr)
- ▶ Select schema
- ▶ Debian packages

Development

`https://github.com/interchange/TableEditor`

Dancer2

Plugin::DBIC <https://metacpan.org/pod/Dancer2::Plugin::DBIC>

Session::DBIC <https://metacpan.org/pod/Dancer2::Session::DBIC>

TableEditor not yet ported

Slides

Slides: <http://www.linuxia.de/talks/czpw2014/dancer-dbic-en-beamer.pdf>