Dancer and DBIx::Class

Stefan Hornburg (Racke) racke@linuxia.de

Dutch Perl Workshop, Utrecht, 25th April 2014



Dancer::Session::DBIC TableEditor Tips and Tricks Forecast and Contribution Dancer
DBIx::Class
Database Administration

Easy to start with

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

Dancer::Session::DBIC
TableEditor
Tips and Tricks
Forecast and Contribution

Dancer DBIx::Class

Database Administration

Easy to expand

- Plugins
- ► Hooks
- Engines

Dancer::Plugin::DBIC
Dancer::Session::DBIC
TableEditor
Tips and Tricks
Forecast and Contribution

Dancer
DBIx::Class
Database Administration

Solid

- Stable
- Keep behaviour
- Community

Tips and Tricks Forecast and Contribution

Applications

Simple Dropbox https:
 //metacpan.org/pod/Dancer::Plugin::Dropbox

- .state.gov Websites
 https://eshop.state.gov/
- Monitor for Power Plant

Introduction

Dancer::Plugin::DBIC
Dancer::Session::DBIC
TableEditor
Tips and Tricks
Forecast and Contribution

Dancer
DBIx::Class
Database Administration

DBIx::Class

- ORM
- Objects instead of SQL
- Performance

Dancer::Plugin::DBIC
Dancer::Session::DBIC
TableEditor
Tips and Tricks
Forecast and Contribution

Dancer DBIx::Class Database Administration

Database Administration

- phpmyadmin
- phppgadmin
- TableEditor

Dancer DBIx::Class Database Administration

TableEditor Features

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



Dancer DBIx::Class Database Administration

Input Database Parameters



View Products

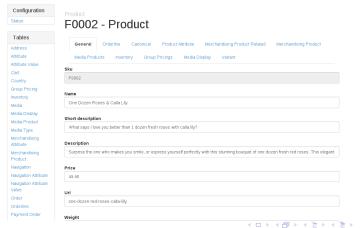


Product - List of items

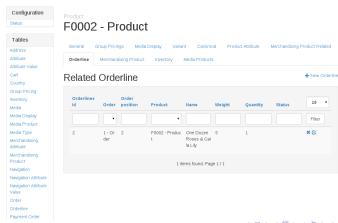


Dancer DBIx::Class Database Administration

View Product



Relationship Orderline



Usage Configuration UTF-8 Create schema dynamically

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({..});
```

Usage Configuration UTF-8 Create schema dynamically

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

Engines Example table Serializer Session expiration

Overview Dancer::Session::DBIC

- engines
- configuration
- serialization
- session expiration

Engines

- Templates TT, Xslate, Flute, ...
- SessionsStorable, Database, DBIC
- LoggerFile, Syslog
- Serializer JSON, YAML, XML



Engines Example table Serializer Session expiration

Configuration

session name of session engine (DBIC) session_options options session_expires expiration date

Engines Example table Serializer Session expiration

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

Engines Example table Serializer Session expiration

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');
```

Installation Frontend Routes Login Relationships Limitations Configuration

Overview TableEditor

- Installation
- Frontend
- Routes
- Login
- Relationships
- Limitations
- Configuration



Installation Frontend Routes Login Relationships Limitations Configuration

Installation

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

Installation Frontend Routes Login Relationships Limitations Configuration

Driver

- DBD::mysql
- DBD::Pg
- **.**..

Frontend

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



Installation
Frontend
Routes
Login
Relationships
Limitations
Configuration

Routes

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

Installation Frontend Routes Login Relationships Limitations Configuration

Login

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



Installation Frontend Routes Login Relationships Limitations Configuration

Relationships

- might_have
- has_many
- belongs_to
- many_to_many

Installation Frontend Routes Login Relationships Limitations Configuration

Limitations

- Primary key for one column only
- Speed (complex schemas)
- Error handling

Installation Frontend Routes Login Relationships Limitations Configuration

Configuration

- Auth::Extensible
- DBIC
 - default



Installation Frontend Routes Login Relationships Limitations Configuration

Planned Features

- Search (Solr)
- Select schema
- Debian packages

User role

User role

```
$role = $schema->resultset('Role')->find(3);
# primary key
$role->id;
# name and label
$role->name;
$role->label;
```

Forecast and Contribution

Object vs Hashref

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

Development Dancer2 Slides

Development

https://github.com/interchange/TableEditor

Dancer2

Development Dancer2 Slides

Slides

Slides: http://www.linuxia.de/talks/nlpw2014/dancer-dbic-en-beamer.pdf