

# Dancer und DBIx::Class

Stefan Hornburg (Racke)

`racke@linuxia.de`

16. Deutscher Perl-Workshop, Hannover, 27. März 2013

# CGI.pm must die ...

... and we bury DBI alive!

# Dancer und DBIx::Class

# Anwendungen

- Magento
- Interchange6

- phpmyadmin
- phppgadmin
- TableEditor

- Unterstützung mehrerer Datenbanksysteme  
MySQL, PostgreSQL, ...
- höherer Abstraktionslevel
- modernes Frontend
- wenig Quellcode
- “einfache” Installation



# Eingabe Datenbankparameter

## Database configuration

### Database Driver

Pg

### Database Name

interchange6

### Username

racke

### Password

\*\*\*\*\*

### Schema class

Interchange6::Schema

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

### DSN suffix

Optional. Extra options for DB connection.

Submit

## 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](#)

Skü	Name	Uri	Priority	Gtin	Canonical sku	Active	Inventory exempt	Created	Last modified	2
										Filter
F0001	One Dozen Roses	one-doz-en-rose-s	0			1	0	2014-03-26T14:15:14	2014-03-26T14:15:14	<a href="#">✕</a> <a href="#">🔗</a>
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	<a href="#">✕</a> <a href="#">🔗</a>

15 items found. Page 1 / 8

## 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

## 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"/>	<input type="text" value="Filter"/>
2	1 - Or der	2	F0002 - Produc t	One Dozen Roses & Cal la Lily	5	1	 	

1 items found. Page 1 / 1

- Anwendung
- Konfiguration
- UTF-8
- Schema erzeugen

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

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

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

# DBIx::Class mit Dancer Plugin

```
use Dancer::Plugin::DBIC;
```

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

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

```
rset( 'User' )->search ( { .. } );
```

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



# Mehrere 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
```

```
use Dancer::Plugin::DBIC;
```

```
schema( 'legacy' )->resultset( 'UserDb' )->search ( { .. } );
```

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

# Schema dynamisch erzeugen

- `schema_class` fehlt in Konfiguration
- `DBIx::Class::Schema::Loader`
- Test und Entwicklung
- `TableEditor`

- Engines
- Konfiguration
- Serialisierung
- Sitzungsablauf

- Templates  
TT, Xslate, Flute, ...
- Sitzungen (Sessions)  
Storable, Database, DBIC
- Logger  
File, Syslog
- Serialisierer  
JSON, YAML, XML

`session` Name der Sessionengine, hier DBIC

`session_options` Optionen

`session_expires` Ablaufzeit der Session

```
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
```



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

```
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;
```

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

- alte Sitzungen aus Datenbank löschen
- `Interchange6::Schema::ResultSet::Session`

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

- Installation
- Frontend
- Anmeldung
- Beziehungen
- Einschränkungen
- Konfiguration

# Installation

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

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

- Angular
  - Routes für das Frontend
  - XHR-Abfragen an REST API
  - JSON
- Bootstrap
- Theme



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

- Dancer::Plugin::Auth::Extensible
- Provider
  - Unix
  - DBIC
- Datenbank (*geplant*)

- might\_have
- has\_many
- belongs\_to
- many\_to\_many

- Primärschlüssel für **eine** Spalte
- Geschwindigkeit (komplexe Schemas)
- Fehlerbehandlung

- Auth::Extensible
- DBIC

- Suche (Solr)
- Auswahl Schema

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

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

**Session::DBIC** noch nicht portiert

**TableEditor** noch nicht portiert



**Slides:** <http://www.linuxia.de/talks/pws2014/dancer-dbic-de-beamer.pdf>