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

eCommerce Software

- Magento
- Interchange6

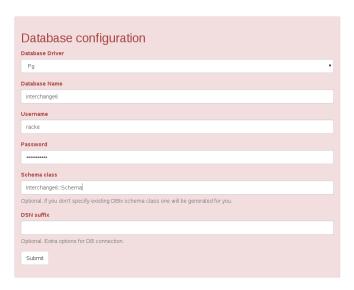
Datenbankadministration

- phpmyadmin
- phppgadmin
- TableEditor

TableEditor Features

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

Eingabe Datenbankparameter



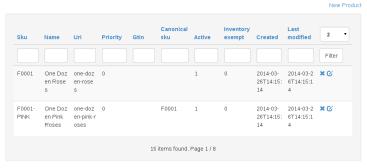
Ansicht Produkte

Configuration Status

Product - List of items

Tables Address Attribute Attribute Value Cart Group Pricing Media Media Display Media Product Media Type Merchandising Merchandising Navigation

Navigation Attribute Navigation Attribute Value Order



Ansicht Produkt

Configuration Status F0002 - Product Tables General Canonical Merchandising Product Related Merchandising Product Attribute Media Products **Group Pricings** Media Display Variant Attribute Value Sku Cart Group Pricing Name One Dozen Roses & Calla Lilv Media Display Short description Media Product What says I love you better than 1 dozen fresh roses with calla lily? Media Type Merchandising Description Attribute Surprise the one who makes you smile, or express yourself perfectly with this stunning bouquet of one dozen fresh red roses. This elegant Merchandising Navigation Price Navigation Attribute 49.95 Navigation Attribute Value Uri Order one-dozen-red-roses-calla-lilly



Weight

Payment Order

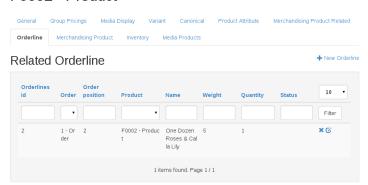
Beziehung Orderline



Tables Attribute Value Cart Group Pricing Media Display Media Product Media Type Merchandising Merchandising Navigation Navigation Attribute Navigation Attribute Value Order Payment Order

Product

F0002 - Product



Übersicht Dancer::Plugin::DBIC

- Anwendung
- Konfiguration
- UTF-8
- Schema erzeugen

DBIx::Class ohne Dancer Plugin

```
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
```

Mehrere Schemas

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

UTF-8 für MySQL

```
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

Übersicht Dancer::Session::DBIC

- Engines
- Konfiguration
- Serialisierung
- Sitzungsablauf

Engines

- 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};
```

Beispieltabelle

```
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;
```

Serialisierer

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

Sitzungsablauf

- alte Sitzungen aus Datenbank löschen
- Interchange6::Schema::Resultset::Session

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

Übersicht TableEditor

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

Installation

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

Treiber

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

Frontend

- Angular
 - Routes f
 ür das Frontend
 - XHR-Abfragen an 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});
};
```

Anmeldung

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

Beziehungen

- might_have
- has_many
- belongs_to
- many_to_many

Einschränkungen

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

- Auth::Extensible
- DBIC

Weitere Features

- Suche (Solr)
- Auswahl Schema

Entwicklung

https://github.com/interchange/TableEditor

Dancer2

Session::DBIC noch nicht portiert

TableEditor noch nicht portiert

Slides

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