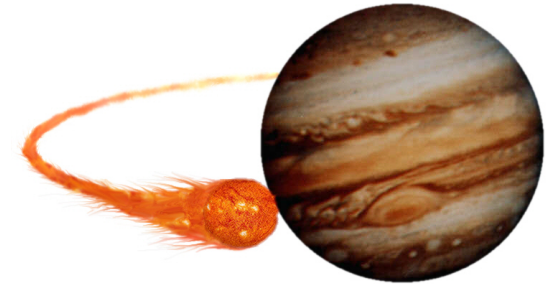


Perl5::Spartanic

Für Perl auf Android und anderen
spartanischen Plattformen

Ralf Peine

www.jupiter-programs.de



1. Motivation

Experiment

Perl Module und Programme auf dem
Android Tablet entwickeln.

**Das sollte doch nicht so
schwierig sein?**

Leider doch!

Motivation

Perl auf Android

- Keine CPAN-Shell
- Kein Standard-Make
- Kein Standard-C-Compiler verfügbar
- Kaum Unix-Commands verfügbar

Motivation

Perl auf Android (2)

- Keine IDE
- Kein ctags
- Kein Emacs (mein Lieblingseditor)
- Vim als einziger professioneller Editor

2. Was nun ?

**CPAN-Module finden,
die ohne Compile
und Make auskommen**

Was nun?

Module ohne Compile und Make finden

Auf CPAN existiert leider kein Attribut für:

- Modul muss kompiliert werden
- Modul benötigt unbedingt Make



Idee



```
use Perl5::Spartanic;
```

```
9 package Perl5::Spartanic;
0
1 use strict;
2 use warnings;
3
4 use vars qw($VERSION);
5 $VERSION = '0.110';
6
7 # no methods!
8
9 1;
```

Was nun?

Perl5::Spartanic (1)

Mit "use Perl5::Spartanic" legt man
folgende Eigenschaften für sein Modul
fest:

Was nun?

Perl5::Spartanic (2)

- Modul muss **NICHT** kompiliert werden
- Modul benötigt **KEIN** Make
- Modul verwendet **AUSSCHLIESSLICH**
Spartanic-Module
- und **PERL-CORE** Module

use Perl5::Spartanic::Strict;

Was nun?

Perl5::Spartanic::Strict

- Modul muss **NICHT** kompiliert werden
- Modul benötigt **KEIN** Make
- Modul verwendet **AUSSCHLIESSLICH PERL-CORE** Module

3. Spartan Lib

Perl Lib

Hilfreiche Core-Module

Archive::Tar;

IO::Compress::Gzip;

Spartanic Lib

HTML::Template; # strict

Trace::Log; # strict

Alive; # strict

PQL::Cache; # strict

Report::Porf; # strict

Scalar::Validation; # strict

Perl5::MetaInfo::DB;

spartanic.lib.gz

Spartanic Lib

`spartanic.lib.gz`

- enthält alle Spartanic Module
- Installation mit `install_spartanic.pl`
 - nur auspacken
 - und ablegen

spartanic_strict.lib.gz

Spartanic Lib
spartanic_strict.lib.gz

- enthält alle Strict-Spartanic Module
- Verwendung einzelner Module möglich

Vorteile

für

Spartanic Scripts

Spartanic Lib

Vorteile für Spartanic Scripts (1)

- vorkonfiguriertes Paket möglich
- laufen in **jeder** Perl-Umgebung
- können vom USB-Stick gestartet werden
 - keine Admin-Rechte notwendig
 - Großunternehmen
 - High Security

Spartanic Lib

Vorteile für Spartanic Scripts (2)

- plattformübergreifend
 - Android
 - MS-Windows
 - Apple
 - Unix/Linux
 - ...
 - Nintendo DS
 - ...

4. Spartan IDE

Spartanic IDE

Info-DB aufbauen

Perl-Module scannen

ms . sh

ms . bat

Info speichern mit

Perl5::MetaInfo::DB;

Spartanic IDE

Info-DB nutzen

Statisches HTML

Perl-Modul-Report

mr . sh

mr . bat

Interaktiv

Shell

mi . sh

mi . bat

Statisches HTML

Perl Module/Class Overview

Mon Feb 1 08:23:36 2016

Spartanic Lib

31 modules / classes listed

PerlVers	#subs	Name (Methods)	Doc	Location	Namespace	#Line	File
	8	Alive	SPRT	SPRT		12	{SPRT}/Alive.pm
	20	DB	-	SPRT		66	{SPRT}/Log/Trace.pm
	0	Template	SPRT	SPRT	HTML	1	{SPRT}/HTML/Template.pm
	32	Template	SPRT	SPRT	HTML	1059	{SPRT}/HTML/Template.pm
	11	COND	-	SPRT	HTML::Template	1045	{SPRT}/HTML/Template.pm
	1	COND	-	SPRT	HTML::Template	3351	{SPRT}/HTML/Template.pm
	1	DEFAULT	-	SPRT	HTML::Template	3292	{SPRT}/HTML/Template.pm
	1	ESCAPE	-	SPRT	HTML::Template	3372	{SPRT}/HTML/Template.pm
	0	FAQ	SPRT	SPRT	HTML::Template	3	{SPRT}/HTML/Template/FAQ.pm
	1	JSEscape	-	SPRT	HTML::Template	3381	{SPRT}/HTML/Template.pm
	2	LOOP	-	SPRT	HTML::Template	1041	{SPRT}/HTML/Template.pm
	2	LOOP	-	SPRT	HTML::Template	3299	{SPRT}/HTML/Template.pm
	1	NOOP	-	SPRT	HTML::Template	3363	{SPRT}/HTML/Template.pm
	3	PRINTSCALAR	-	SPRT	HTML::Template	3400	{SPRT}/HTML/Template.pm
	1	URLEscape	-	SPRT	HTML::Template	3390	{SPRT}/HTML/Template.pm
	1	VAR	-	SPRT	HTML::Template	3285	{SPRT}/HTML/Template.pm
	4	Trace	SPRT	SPRT	Log	1	{SPRT}/Log/Trace.pm
	30	Cache	SPRT	SPRT	PQL	11	{SPRT}/PQL/Cache.pm
5.10	19	DB	SPRT	SPRT	Perl5::MetaInfo	13	{SPRT}/Perl5/MetaInfo/DB.pm
	0	Spartanic	SPRT	SPRT	Perl5	19	{SPRT}/Perl5/Spartanic.pm
	2	Porf	SPRT	SPRT	Report	16	{SPRT}/Report/Porf.pm
	24	Framework	SPRT	SPRT	Report::Porf	53	{SPRT}/Report/Porf/Framework.pm
	107	Simple	SPRT	SPRT	Report::Porf::Table	47	{SPRT}/Report/Porf/Table/Simple.pm
	6	AutoColumnConfigurator	SPRT	SPRT	Report::Porf::Table::Simple	25	{SPRT}/Report/Porf/Table/Simple/AutoColumnConfigurator.pm
	10	CsvReportConfigurator	SPRT	SPRT	Report::Porf::Table::Simple	45	{SPRT}/Report/Porf/Table/Simple/CsvReportConfigurator.pm

Spartanic IDE

HTML-Infozeile

1	<u>Cache</u>	<u>SPRT</u>	SPRT	<u>LOG</u>
30	<u>Cache</u>	<u>SPRT</u>	SPRT	<u>PQL</u>
10	nn	nnnn	nnnn	nnnnnn

Spartanic IDE

Subs (top)



PQL::Cache

Home

{SPRT}/PQL/Cache.pm

Mon Feb 1 08:23:37 2016

Documentation: [SPRT](#)

[30 Methods/Subs have been found in .pm file](#)

[Modules/Classes are used by this one.](#)

[Modules/Classes are using this one.](#)

Spartanic IDE

Subs (Liste)

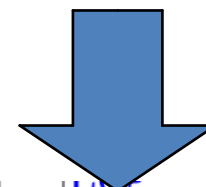


Methods / Subs of Cache

30 methods/subs have been found by the parser in the *.

ID	#Line	***	Name
34	595		_find_matches
35	696		_handle_where
13	80		_implement_compare_subs
11	59		_init
26	386		_insert
14	209		_prepare_cache
33	564		_select_columns
28	437		_update_index_key
29	458		build_index_string
15	230		clear
30	481		delete

Spartanic IDE Namespace



30	Cache	SPRT	SPRT	PQL
10	DD	DDDT	DDDT	DDDT

PQL

Home

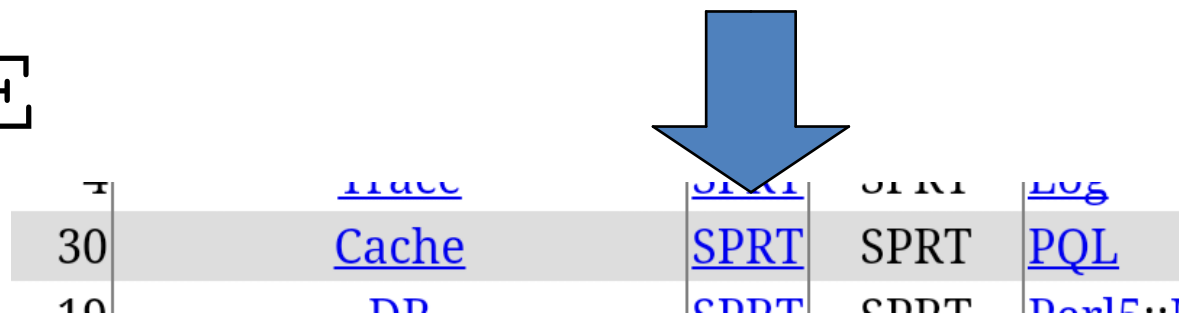
Modules Of Namespace

Generated: Mon Feb 1 08:23:37 2016

Modules listed: 1

PerlVers	#subs	Name (Methods)	Doc	Location	Namespace	#Line	File
	30	Cache	SPRT	SPRT	PQL	11	{SPRT}/PQL/Cache.pm

Spartanic IDE Manpage



NAME

PQL::Cache - A mini in memory database using pure perl and PQL - PerlQueryLanguage -

PQL is similar to DBIx::Class API, so switching to DBIx and SQL databases is possible.

It uses pure Perl (no compilation needed), so you can run it from a USB stick.

VERSION

This documentation refers to version 0.800 of PQL::Cache

SYNOPSIS

```
use PQL::Cache qw(:all);
```

Create Cache

```
my $cache = PQL::Cache->new();
$cache->set_table_definition
    ('person',
     {
         keys    => ['ID'],
         columns => [pname => surname => birth => gender => 'perl_level']
     });
```

Interaktive Abfrage

Danke!

Wozu validieren ?

```
sub add {  
    my $sum = 0;  
    for (@_) { $sum += $_; }  
    return $sum;  
}
```


Fazit

Vergleich Fazit (1)

Für Moose

- `MooseX::Params::Validate`
- `Type::Params`

Danke!