Masaryk University
Faculty of Informatics

Refactoring of Kdyby packages

Bachelor's Thesis

Filip Procházka

Brno, Spring 2017

Replace this page with a copy of the official signed thesis assignment and a copy of the Statement of an Author.

Declaration

Hereby I declare that this paper is my original authorial work, which I have worked out on my own. All sources, references, and literature used or excerpted during elaboration of this work are properly cited and listed in complete reference to the due source.

Filip Procházka

Advisor: RNDr. Jaroslav Bayer

Acknowledgement

This is the acknowledgement for my thesis, which can span multiple paragraphs.

Abstract

This is the abstract of my thesis, which can
span multiple paragraphs.

Keywords

package, kdyby, nette, doctrine, orm, composer, packagist

Contents

List of Tables

List of Figures

1 Introduction

The Kdyby is an Open-source software (OSS) [1] project that I, Filip Procházka, lead and maintain. It is a set of PHP [2] libraries, that aim to ease writing of web applications.

Through my carer, I've been able to use the Kdyby in core business applications of companies such as Damejidlo.cz and Rohlik.cz. A lot of people consider my work useful enough, to incorporate it to their own applications as well.

As of writing this, the more popular libraries have hundreds of thousands of downloads. Five of Kdyby libraries have over quarter million downloads and one is approaching half a million with staggering amount of 470 thousands of downloads [3]. In conclusion, a sober estimate would be, that Kdyby libraries are used in hundreds of real production applications.

If I'll account only for the two biggest projects that I can confirm are using the Kdyby packages, over a billion Czech crowns [4] has literary flowed through the Kdyby. That is a big responsibility.

Over the years, I've had problems keeping up with the demand and the packages began to get obsolete. I wanna use this thesis as way to fix the situation.

I'm going to review the state of each library, decide it's future, which means I'll either deprecate it and provide the users a suggestion for a better alternative, or fix the problems and refactor the library.

2 Theory

2.1 A brief history of Kdyby

At 2006, few months after I've started attending my high school, I've began learning PHP and a friend introduced me to a concept called Content Management System (CMS) [5]. I've immediately started working on my own CMS, as a way to learn PHP and also as a way to make some money.

I've managed to make a working prototype, that was used in production on few websites. The oldest preserved version is archived at `https://github.com/fprochazka/kdyby-cms-old`. And as expected, it is full of security holes and badly written code.

Then the concept of Open-source software (OSS) [1] was introduced to me and I've decided to start working on everything openly, under a free license [6]. Sadly, since then, no new working version of the Kdyby CMS was ever released, because I've rewritten it from scratch exactly 10 times.

Around year 2012, I've realized this is not a good way to continue and split all useful code to separate libraries, that could be used more or less independently and have their own release cycle.

2.2 Technologies used

2.2.1 Composer

Composer is a tool for dependency management [7] in PHP. It allows you to declare the libraries your project depends on and it will manage (install or update) them for you [8].

Packages are usually published using Github 2.4.3 as a Git [9] repository with metadata in a file named composer.json, that is written in JSON [10] format.

2.2.2 Nette Framework

Nette Framework is an OSS framework for creating web applications in PHP [11].

2.2.3 Doctrine 2 ORM

Doctrine ORM is an Object-Relation Mapper (ORM) [12], which means it allows the programmer to create PHP classes called entities, that represent relational data in a database and are used to actually map the data from the database to the classes and back. In conclusion, it allows the programmer to write a fully Object-oriented (OOP) [13] applications.

2.2.4 Symfony Framework

Symfony is a PHP web application framework and a set of reusable PHP components/libraries, similar to Nette. [14]

2.2.5 Monolog

Monolog is a logging library that sends your logs to files, sockets, inboxes, databases and various web services. This library implements the PSR-3 [15] interface that you can type-hint against in your own libraries to keep a maximum of interoperability. [16]

2.2.6 RabbitMQ

RabbitMQ is OSS message broker software (sometimes called message-oriented middleware) that implements the Advanced Message Queuing Protocol (AMQP). The RabbitMQ server is written in the Erlang programming language and is built on the Open Telecom Platform framework for clustering and failover. [17]

2.2.7 ElasticSearch

Elasticsearch is a search engine based on Lucene. It provides a distributed, multitenant-capable full-text search engine with an HTTP web interface and schema-free JSON documents. Elasticsearch is developed in Java and is released as open source under the terms of the Apache License. It is the most popular enterprise search engine . [18]

2.2.8 Redis

Redis is an in-memory database OSS project, that is networked, in-memory, and stores keys with optional durability. [19]

2.2.9 PhpStan

PHPStan focuses on finding errors in your code without actually running it. It catches whole classes of bugs even before you write tests for the code. [20]

2.2.10 Nette\Tester

Nette\Tester is an unit testing [21] framework for the PHP. [22]

2.3 Techniques and design patterns

2.3.1 Dependency Injection

Inversion of control is a design principle in which custom-written portions of a computer program receive the flow of control from a generic framework.

Dependency injection is a technique whereby one object supplies the dependencies of another object. Passing the service to the client, rather than allowing a client to build or find the service, is the fundamental requirement of the pattern. [23]

2.3.2 Aspect Oriented Programming

In computing, aspect-oriented programming (AOP) is a programming paradigm that aims to increase modularity by allowing the separation of cross-cutting concerns. It does so by adding additional behavior to existing code (an advice) without modifying the code itself, instead separately specifying which code is modified via a "pointcut" specification, such as "log all function calls when the function's name begins with 'set'". This allows behaviors that are not central to the business logic (such as logging) to be added to a program without cluttering the code core to the functionality. [24]

2.4 3rd party services

2.4.1 Packagist

Packagist is the main Composer repository. It aggregates public PHP packages installable with Composer. [25]

2.4.2 Facebook

Facebook is a social network, that can be used as an authentication tool for web services. If you have an account at Facebook and are logged in, some 3rd party service you'd like to use can use that, to allow you to log in to their website, without forcing you to go through registration process, or having to remember password.

2.4.3 Github

Github is a collaboration platform for software development using Git [9]. Kdyby is hosted and developed here, such as many other Composer packages.

It can also, same as Facebook, be used as authentication tool.

2.4.4 Google

Google is a company behind the most popular search engine google.com, but we will talk only about it's OAuth 2 functionality.

2.4.5 PayPal

PayPal is a payment service, that you can use to pay on 3rd party websites, using your debit or credit card with very simple process.

2.4.6 CSOB Payment Gateway

CSOB Payment Gateway is a standard card payment gateway provider.

3 Current state of the project

To be able to lay out the roadmap, first we have to know the current state of each Kdyby package, the original purpose and the current requirements. We shall only review those packages, that actually made it to production and at least one usable version was released.

Few years back I was really eager to solve all the problems around developing web applications in PHP and I've created few GitHub repositories as a reminder for me, to start working on those problems. And I've actually started to work on some, for example DoctrineForms is one of them, but it was never "officially released". The rest I've not even started working on.

3.1 State of the project

Lorem ipsum.

3.2 State of each package

Let's review each relevant package separately.

3.2.1 Doctrine

Lorem ipsum dolor sit amet.

3.2.2 Console

Lorem ipsum.

3.2.3 Events

Lorem ipsum.

3.2.4 Annotations

Lorem ipsum.

3.2.5 DoctrineCache

Lorem ipsum.

3.2.6 DoctrineMagicAccessors

Lorem ipsum.

3.2.7 DoctrineCollectionsReadonly

Lorem ipsum.

3.2.8 DoctrineCollectionsLazy

Lorem ipsum.

3.2.9 DoctrineDbalBatchImport

Lorem ipsum.

3.2.10 DoctrineForms

Lorem ipsum.

3.2.11 Curl

Lorem ipsum.

3.2.12 CurlCaBundle

Lorem ipsum.

3.2.13 Autowired

Lorem ipsum.

3.2.14 FormsReplicator

Lorem ipsum.

3.2.15 Translation

Lorem ipsum.

3.2.16 Validator

Lorem ipsum.

3.2.17 RabbitMq

Lorem ipsum.

3.2.18 Money

Lorem ipsum.

3.2.19 DoctrineMoney

Lorem ipsum.

3.2.20 Aop

Lorem ipsum.

3.2.21 Clock

Lorem ipsum.

3.2.22 Redis

Lorem ipsum.

3.2.23 ParseUseStatements

Lorem ipsum.

3.2.24 RedisActiveLock

Lorem ipsum.

3.2.25 TesterParallelStress

Lorem ipsum.

3.2.26 Monolog

Lorem ipsum.

3.2.27 ElasticSearch

Lorem ipsum.

3.2.28 DoctrineSearch

Lorem ipsum.

3.2.29 Geocoder

Lorem ipsum.

3.2.30 CsobPaygateNette

Lorem ipsum.

3.2.31 CsobPaymentGateway

Lorem ipsum.

3.2.32 Wkhtmltopdf

Lorem ipsum.

3.2.33 FakeSession

Lorem ipsum.

3.2.34 RequestStack

Lorem ipsum.

3.2.35 StrictObjects

Lorem ipsum.

3.2.36 Facebook

Lorem ipsum.

3.2.37 Google

Lorem ipsum.

3.2.38 Github

Lorem ipsum.

3.2.39 NettePhpServer

Lorem ipsum.

3.2.40 TesterExtras

Lorem ipsum.

3.2.41 HtmlValidatorPanel

Lorem ipsum.

3.2.42 BootstrapFormRenderer

Lorem ipsum.

3.2.43 PayPalExpress

Lorem ipsum.

3.2.44 PresentersLocator

Lorem ipsum.

3.2.45 SvgRenderer

Lorem ipsum.

3.2.46 QrEncode

Lorem ipsum.

4 Roadmap

4.1 Common requirements

Lorem ipsum.

4.2 Roadmap for each package

4.2.1 Doctrine

Lorem ipsum.

4.2.2 Console

Lorem ipsum.

4.2.3 Events

Lorem ipsum.

4.2.4 Annotations

Lorem ipsum.

4.2.5 DoctrineCache

Lorem ipsum.

4.2.6 DoctrineMagicAccessors

Lorem ipsum.

4.2.7 DoctrineCollectionsReadonly

Lorem ipsum.

4.2.8 DoctrineCollectionsLazy

Lorem ipsum.

4.2.9 DoctrineDbalBatchImport

Lorem ipsum.

4.2.10 DoctrineForms

Lorem ipsum.

4.2.11 Autowired

Lorem ipsum.

4.2.12 FormsReplicator

Lorem ipsum.

4.2.13 Translation

Lorem ipsum.

4.2.14 Validator

Lorem ipsum.

4.2.15 RabbitMq

Lorem ipsum.

4.2.16 Money

Lorem ipsum.

4.2.17 DoctrineMoney

Lorem ipsum.

4.2.18 Aop

Lorem ipsum.

4.2.19 Clock

Lorem ipsum.

4.2.20 Redis

Lorem ipsum.

4.2.21 ParseUseStatements

Lorem ipsum.

4.2.22 RedisActiveLock

Lorem ipsum.

4.2.23 TesterParallelStress

Lorem ipsum.

4.2.24 Monolog

Lorem ipsum.

4.2.25 ElasticSearch

Lorem ipsum.

4.2.26 DoctrineSearch

Lorem ipsum.

4.2.27 Geocoder

Lorem ipsum.

4.2.28 CsobPaygateNette

Lorem ipsum.

4.2.29 CsobPaymentGateway

Lorem ipsum.

4.2.30 Wkhtmltopdf

Lorem ipsum.

4.2.31 FakeSession

Lorem ipsum.

4.2.32 RequestStack

Lorem ipsum.

4.2.33 StrictObjects

Lorem ipsum.

4.2.34 Facebook

Lorem ipsum.

4.2.35 Google

Lorem ipsum.

4.2.36 Github

Lorem ipsum.

4.2.37 NettePhpServer

Lorem ipsum.

4.2.38 TesterExtras

Lorem ipsum.

4.2.39 HtmlValidatorPanel

Lorem ipsum.

5 Implementation

5.1 Doctrine

Lorem ipsum.

5.2 Console

Lorem ipsum.

5.3 Events

Lorem ipsum.

5.4 Annotations

Lorem ipsum.

5.5 DoctrineCache

Lorem ipsum.

5.6 DoctrineMagicAccessors

Lorem ipsum.

5.7 DoctrineCollectionsReadonly

Lorem ipsum.

5.8 DoctrineCollectionsLazy

Lorem ipsum.

5.9 DoctrineDbalBatchImport

Lorem ipsum.

5.10 DoctrineForms

Lorem ipsum.

5.11 Autowired

Lorem ipsum.

5.12 FormsReplicator

Lorem ipsum.

5.13 Translation

Lorem ipsum.

5.14 Validator

Lorem ipsum.

5.15 RabbitMq

Lorem ipsum.

5.16 Money

Lorem ipsum.

5.17 DoctrineMoney

Lorem ipsum.

5.18 Aop

Lorem ipsum.

5.19 Clock

Lorem ipsum.

5.20 Redis

Lorem ipsum.

5.21 ParseUseStatements

Lorem ipsum.

5.22 RedisActiveLock

Lorem ipsum.

5.23 TesterParallelStress

Lorem ipsum.

5.24 Monolog

Lorem ipsum.

5.25 ElasticSearch

Lorem ipsum.

5.26 DoctrineSearch

Lorem ipsum.

5.27 Geocoder

Lorem ipsum.

5.28 CsobPaygateNette

Lorem ipsum.

5.29 CsobPaymentGateway

Lorem ipsum.

5.30 Wkhtmltopdf

Lorem ipsum.

5.31 FakeSession

Lorem ipsum.

5.32 RequestStack

Lorem ipsum.

5.33 StrictObjects

Lorem ipsum.

5.34 Facebook

Lorem ipsum.

5.35 Google

Lorem ipsum.

5.36 Github

Lorem ipsum.

5.37 NettePhpServer

Lorem ipsum.

5.38 TesterExtras

Lorem ipsum.

5.39 HtmlValidatorPanel

Lorem ipsum.

6 Conclusion

We made it!

Bibliography

[1] Wikipedia. *Open-source software — Wikipedia, The Free Encyclopedia*. 2017.

[2] Wikipedia. *PHP — Wikipedia, The Free Encyclopedia*. 2017.

[3] *Packages from kdyby - Packagist*. 2017.

[4] Matt Jedlička. *Rohlik.cz loni prodal zboží za miliardu, letos chce konečně zisk*. 2017.

[5] Wikipedia. *Content management system — Wikipedia, The Free Encyclopedia*. 2017.

[6] Wikipedia. *Free software license — Wikipedia, The Free Encyclopedia*. 2017.

[7] Wikipedia. *Package manager — Wikipedia, The Free Encyclopedia*. 2017.

[8] Composer Community. *Introduction - Composer*. 2017.

[9] Wikipedia. *Git — Wikipedia, The Free Encyclopedia*. 2017.

[10] Wikipedia. *JSON — Wikipedia, The Free Encyclopedia*. 2017.

[11] Wikipedia. *Nette Framework — Wikipedia, The Free Encyclopedia*. 2017.

[12] Wikipedia. *Object-relational mapping — Wikipedia, The Free Encyclopedia*. 2017.

[13] Wikipedia. *Object-oriented programming — Wikipedia, The Free Encyclopedia*. 2017.

[14] Wikipedia. *Symfony — Wikipedia, The Free Encyclopedia*. 2017.

[15] PHP Framework Interop Group. *PHP Standards Recommendations*. 2017.

[16] Jordi Boggiano. *Monolog - Logging for PHP*. 2017.

[17] Wikipedia. *RabbitMQ — Wikipedia, The Free Encyclopedia*. 2017.

[18] Wikipedia. *Elasticsearch — Wikipedia, The Free Encyclopedia*. 2017.

[19] Wikipedia. *Redis — Wikipedia, The Free Encyclopedia*. 2017.

[20] Ondřej Mirtes. *PHPStan - PHP Static Analysis Tool*. 2017.

[21] Wikipedia. *Unit testing — Wikipedia, The Free Encyclopedia*. 2017.

[22] David Grudl. *Nette Tester – enjoyable unit testing*. 2017.

[23] Martin Fowler. *Inversion of Control Containers and the Dependency Injection pattern*. 2004.

[24] Wikipedia. *Aspect-oriented programming — Wikipedia, The Free Encyclopedia*. 2016.

[25] *Packages from kdyby - Packagist*. 2017.