

Contao und Symfony – Wie geht es weiter?

Datum: 24.04.2014

Uhrzeit: 19:00 Uhr bis 21:15 Uhr

<https://contao.org/de/news/einladung-zum-call-am-24042014.html>

Protokoll

- Der Featurebranch ist offiziell verfügbar: <https://github.com/contao/core/tree/feature/contao-with-symfony>
- Das ganze ist noch 100% rückwärtskompatibel (es hat sich an sich nicht viel verändert), die Frage ist, wie tief soll es gehen (die Veränderung)
- Erweiterungen weiter in system/modules oder in Symfony Manier einstellen?
- Bedenken: Wie sieht es mit der Dokumentation aus? Schnittstellen Contao <-> Symfony
- Formulare für das Backend soll weiterhin über das DCA gemacht werden
- Was Symfony bringen soll:
 - Beispiel: Die Datenbankinstalltool sollte über das Doctrine Schematool einfacher möglich sein (Tabellendefinitionen hat Zugriff drauf etc.)
 - Dokumentation für Neuentwickler ist einfacher
 - Geht halt zu Lasten der Rückwärtskompatibilität
 - Unit Tests wären möglich!
- Die große Frage: Rechnet sich die ganze Arbeit?
- Ursprünglich war erst einmal der Plan überhaupt ein schon bekanntes großes PHP-Framework zu nutzen -> bei weiteren Umfragen hat sich dann Symfony herauskristallisiert
 - Es ging dabei auch schon immer darum auch im Core komplett dann das neue Framework zu nutzen
 - Frage dabei: Abwärtskompatibilität -> alle Erweiterungen würden auf einen Schlag nicht mehr funktionieren bzw. ist die Frage, wie man da updaten könnte
 - Man kann nicht hin gehen und das von Grund auf neu aufbauen, da dann keine Erweiterungen mehr funktionieren würden, das Problem davon haben wir von Contao 2 auf Contao 3 gesehen. -> TYPO3 ist da mit Flow auch ein spannendes Beispiel
- Contao Framework als Shim ([http://en.wikipedia.org/wiki/Shim %28computing%29](http://en.wikipedia.org/wiki/Shim_%28computing%29)) nachbauen in Symfony
 - d.h. die Queries müssten so umgebaut werden, das Funktionen von früher genau so weiter abgebildet werden. Müsste man dann in konkreten Fällen genauer schauen
 - Es wird durchaus Probleme geben, die man aktuell noch nicht absehen kann. Das wird man dann erst feststellen, wenn es soweit ist.
- Es gibt ja bereits CMS'e die zu Symfony gewechselt sind. Die haben sich ja auch Gedanken gemacht zu der Rückwärtskompatibilität. Man sollte mal schauen, wie die das gemacht haben (beide haben Schritt für Schritt migriert, aus gutem Grund):
 - eZ Publish
 - <http://share.ez.no/blogs/ez/migrating-from-ez-publish-4-to-ez-publish-5-basic-concepts> / <https://confluence.ez.no/pages/viewpage.action?pageId=12780126>
 - Drupal
- Man könnte auch schauen, ob man nur Teile von Contao durch Komponenten von anderen Frameworks ersetzen könnte.
 - Vorteil: man kann nach und nach austauschen
 - Nachteil: Zeit! Das klappt einfach leider nicht aufgrund fehlender Manpower
- Einige Hauptkrankheiten von Contao liegen in der Softwarearchitektur selbst. Diese bekommt man auch erst einmal nicht weg, wenn man das ganze rüber kopiert. D.h. da müsste man von Grund auf ran gehen.
- Definitive Integration bzw. Wechsel der Input Klasse zu Symfony HTTP / Integration von

- Doctrine
- Das Symfony Framework entwickelt sich mit einer Geschwindigkeit weiter, mit der Contao nicht mithalten kann
- Man hätte bei einem Wechsel (Rückwärtskompatibel) dann 2 API's zur Verfügung (Old Way / New Way)
 - Wir wollen keinen harten Schnitt, sondern einen sanften Übergang! Sehr wichtig.
 - Es bleibt ein Problem von Zeit und Geld. Die Motivation ist da nicht die Frage (die Entwickler wären zwar sehr motiviert, haben aber weder Zeit noch Geld)
- Am besten so früh wie möglich IoC (Inversion of Control), DI (Dependencies Injection) und Events einbauen. Dann kann man als Entwickler komfortabel eigene Anpassungen vornehmen (als Idee).
- Rückwärtskompatibel wäre eher eine Contao 3.5 Version / Breakpoint wäre Contao 4 (da wäre das möglich allerdings so komplett nicht gewünscht)

Grundfragen

- Symfony Framework wird mit gebootet und ein paar Dinge aus Contao ersetzen
- Die Integration soll weiter gehen und Erweiterungen sollen dann darauf aufbauen
- Als Beispiel das Loginsystem -> aktuell ist es nicht einfach möglich Logins über andere Schnittstellen einzubinden. Das wäre dann mit Symfony wesentlich einfacher
- Routing in Symfony ist performanter -> der komplette Baum kann gedumpt werden, um ein Routing ohne zusätzliche Datenbankzugriffe zu ermöglichen
- Dependency Injection ist wesentlich einfacher bei Symfony
- Die Security Komponente (ist für Authentifizierung zuständig) ist extrem komplex, Einbindung mindestens erst langfristig, nicht sofort
- Mangelnde Transparenz der Security Komponente, da extrem verschachtelt (Tristan)
- Formulargenerator in Contao nutzt nur einen geringen Teil der Fähigkeiten von Symfony Forms
- Was wäre sinnvoll überhaupt von Symfony zu nutzen
 (<https://github.com/symfony/symfony/tree/master/src/Symfony/Component>)
 - Welche im ersten Schritt
 - Sinnig sind sicher alle auf Dauer
 - Symfony 2 ist eventuell zu komplex als Fullstack.. da wären Teile (Komponenten) einfacher zu Handhaben für Erweiterungsentwickler (nicht so komplex)

Argumente

Es gibt Argumente für den kompletten Wechsel und Argumente dafür erstmal nur Teile zu ersetzen. Es sollte eine Entscheidung gefällt werden.

- Symfony Ja/Nein? -> Ja
- Rückwärtskompatibel -> Ja
- Schritt wäre die komplette Lib umwandeln in Shims, Libraries in Shims, vorrangig erstmal DI, Events und IoC.
- Frage nach dem ersten Schritt -> Module nach -> Symfony Bundles Ordner? Wird das gebraucht? (Wichtige Frage der Abhängigkeit!) -> Nein
- Es wäre wichtig, das man definieren kann, in welcher Reihenfolge die Erweiterungen geladen werden sollen.

Bei vielen Erweiterungen ist es in Symfony nicht wichtig in welcher Reihenfolge sie geladen werden. Bei Contao Extension schon. Das beschränkt sich allerdings im Grunde auf das DCA von Contao. Das DCA müssten also umgestellt werden auf einen Configloader wie in Symfony, damit die Reihenfolge nicht mehr wichtig ist.

Lösungsansätze

Die Ordnerstruktur unter system/modules bleibt erhalten, dort kann man eine Bundle Klasse laden und die Reihenfolge der Bundles werden per initconfig.php bestimmt.

- ezPublish macht sowas (Backendintegration) mit Dual-Kernel View
- Soll Symfony voll integriert werden? -> Ja, das komplette Bundle
- Twig-Templates könnten schon in Contao 3.3 eingeführt werden (.twig), das könnte dann so in Symfony weiter genutzt werden.
- Der Core prüft ob XHTML genutzt werden soll, wenn nicht, dann nimm html5 bzw. dann eben .twig
- Schrittweise die Symfony Bundles reinziehen um zu sehen, was schon drin ist und was noch fehlt
- Man legt die Config an Symfony bekannten Stellen ab und schult die Contao Entwickler, damit diese sich auf die Änderung einstellen können

Nächste Schritte

- Trennung der Ressourcen soweit möglich
- Nutzung der Symfony-Komponenten im Contao-Framework
- Frühe und umfassende Integration des Dependency-Injection-Containers
- Nutzung des Event-Dispatchers für Contao-Hooks
- Entwicklung eines Bundle-Autoloaders für Symfony-Bundles

Termine für den nächsten Call

Montag 28.04 ab 19.00 Uhr

Mögliche Themen:

- Implementierungen in Contao 3.3 sind teilweise unklar, sollten noch mal besprochen werden (z.B. die Custom Sections umbauen)