



TYP03

Munich User Group

Frontend Formulare in TYP03 8 LTS

TYP03 User Group Munich

7. Februar 2017



Peter Kraume

Senior TYPO3 Developer

Mitglied im Vorstand der TYPO3 Association

Twitter: @cybersmog

Mail: peter.kraume@bgm-gmbh.de

Slides: <http://de.slideshare.net/pk77/>



w e b s o l u t i o n s

Evolution von EXT:form

- Form Wizard gibt es schon lange als mitgelieferte Systemextension in TYPO3
- Umfangreicher Rewrite in TYPO3 4.6 durch Patrick Broens
- Seit TYPO3 7.4 (Sommer 2015) hat die Jenaer Agentur TRITUM die Pflege der Extension übernommen
- August 2015: Code Sprint bei TRITUM

Dilemma

- Zu viele offene Bugreports
- Viele Hürden durch veraltete Architektur
- Kein Vertrauen in der Community in EXT:form

Plan:

- Codebasis auf Extbase und Fluid heben => realisiert in TYPO3 7.5
- Form Wizard neu erstellen => geplant für TYPO3 8.x

Realität:

- Rewrite des Form Wizard allein nicht zielführend
- Plan zum kompletten Rewrite von EXT:form

Zielgruppe von EXT:form

	Redakteure	Integratoren	Entwickler / Admins
formhandler		✓	✓
formz			✓
powermail	✓	✓	✓
form	✓	✓	✓













Architektur von EXT:form

- Analyse von anderen CMS und 3rd Party Formular Erweiterungen
- Grundlage ist das Flow Form Framework von Sebastian Kurfürst
- EXT:form wurde noch generischer ausgelegt
- Umsetzung des Editors nur mit jQuery um nicht wieder ein weiteres JavaScript Framework in den TYPO3 Core integrieren zu müssen

Struktur

- Forms (Backend Modul)
 - Form Manager mit Assistent zum Kickstart neuer Formulare
 - Form Editor zum Bearbeiten der Formulare
- Inhaltselement „Mail form“

Form Manager

Edit existing forms					
+ Form name			Location		References
 Demo Contact Form				1:/user_upload/demoContactForm.yaml	1
 Job Application Form				1:/user_upload/jobApplicationForm.yaml	0
 Save to database example				EXT:my_theme/Resources/Private/Forms/my_theme.yaml	1

Create new form

Form name

Advanced settings

Step 1 of 4

Advanced settings

Start template

Step 2 of 4

Cancel Next

Form Editor

The image shows a web-based form editor interface. At the top, there is a toolbar with icons for a new page, close, save, settings, and navigation. The main interface is divided into three sections:

- Left Sidebar:** A tree view showing the project structure. It includes a folder 'Demo Contact Form' containing 'Seite 1 (Page)' and 'Seite 2 (Page)'. Under 'Seite 1', there are elements: 'Name (Text)', 'Anrede (Radio button)', 'Geburtsdatum (Date picker)', and 'E-Mail Adresse (Text)'. Under 'Seite 2', there are 'Kommentar (Textarea)' and 'Summary page (Summary page)'. A 'Create new page' button is at the bottom.
- Center Canvas:** Displays 'Demo Contact Form' and 'Seite 1'. It shows a form with a blue header 'Text: text-1' and a 'Name *' field. Below it are 'Anrede *' (radio buttons for 'Herr' and 'Frau'), 'Geburtsdatum *' (date picker), and 'E-Mail Adresse *' (text field). A 'Create new element' button is at the bottom.
- Right Panel:** Properties for the selected 'Name' element. It includes fields for 'Element name' (Name), 'Placeholder' (Ihr Name), 'Default value', and a checked 'Required field' checkbox. A 'Validators' dropdown menu is set to 'Add validator'.

Form Editor

- hoch konfigurierbar
- komplett anpassbar
- JavaScript API zur Erweiterung des Editors

New element ×

Basic elements

Text Textarea Password

Advanced password

Select elements

Checkbox Single select Radio button

Multi select Multi checkbox

Advanced elements

Date picker File upload Image upload

Static text Content element

Container elements

Fieldset

Element name

Are you a TYPO3 association member?

Hubspot Property Name

company.1.assocmemberlevel

Hubspot Property Type

Contact ▼

Choices

	Label	Value	Selected	
≡	Platinum	Platinum	<input type="checkbox"/>	
≡	Gold	Gold	<input type="checkbox"/>	
≡	Silver	Silver	<input type="checkbox"/>	

Konfiguration

- YAML Dateien für Formulare und den Editor selbst
- Datei basiert => Formulare sind versionierbar
- Konfiguration kann überschrieben werden
- Konfiguration kann vererbt werden

```
1
2 type: Form
3 identifier: demoContactForm
4 label: 'Demo Contact Form'
5 prototypeName: standard
6 finishers:
7   -
8     options:
9       message: 'thx {text-2}'
10      identifier: Confirmation
11 renderables:
12   -
13     type: Page
14     identifier: page-1
15     label: 'Seite 1'
16     renderables:
17       -
18         properties:
19           placeholder: 'Ihr Name'
20           defaultValue: ''
21           type: Text
22           identifier: text-1
23           label: Name
24           validators:
25             -
26               identifier: NotEmpty
27             -
28               options:
29                 regularExpression: '/^(?i)php[34]/'
30               identifier: RegularExpression
31             -
32               properties:
33                 options:
34                   Herr: Herr
35                   Frau: Frau
36                 type: RadioButton
37                 identifier: radiobutton-1
38                 label: Anrede
39                 validators:
40                   -
41                     identifier: NotEmpty
```

Speicherorte für Formulare

- FAL
- site package / theme extension

Best Practice für Integriatoren:

- Formular im Editor erstellen
- YAML Datei in Site Package / Theme Extension verschieben
- Theme Extension braucht Basiskonfiguration, damit Formulare gefunden werden

PHP API

- Methoden zum Zugriff auf jeden Schritt während der Lebensdauer eines Formulars bzw. der Daten
- Eigene Renderer
 - Ausgabe als PDF Formular
 - TCA Ausgabe (Extension Kickstarter)
- Formulare via API
 - dynamisch erstellte Formulare
- Formulare mit Bedingungen



Formular Elemente

- Eigene Formularelemente
 - Google Maps Einbindung zur Positionsbestimmung
 - Bewertung mit Sternen
- alle HTML5 Attribute werden unterstützt
- Datei Uploads werden durch FAL verarbeitet

Validatoren

- Standard Validatoren werden mitgeliefert
- Eigene Validatoren möglich
- Es gibt keine mitgelieferte browserbasierte Validierung

Validators


▼  Regular expression 

Regular expression

Enter a valid PHP PCRE regular expression here.

Finisher

- Mitgelieferte Finisher
 - Speichern in Datenbank
 - Versand von Mails
 - Weiterleitungen
- Finisher können aufeinander aufbauen
 - Datenübernahme aus Vorgänger
- Eigene Finisher
- Finisher können überschrieben werden

Send email (sender) 

Subject

Neue Nachricht von Webseite {+} ▾

Recipient address

name@domain.tld {+} ▾

Recipient name

Musterfirma {+} ▾

Sender address

{text-2} {+} ▾

Sender name

{text-1} {+} ▾

Reply-to address

{+} ▾

CC address

{+} ▾

BCC address

{+} ▾

Format

HTML ▾

Weitere Features

- Mehrsprachige Formulare
- Formulare mit mehreren Schritten
- mehrere Formulare auf einer Seite
- Spam Schutz
- Starke Voreinstellungen für benutzerfreundlichen Editor
- TypoScript Unterstützung

Migration alter Formulare

- bislang gibt es keine Migration
- Alte EXT:form Extension wird ins TER wandern
- Parallelbetrieb möglich!

Nächste Schritte bis TYPO3 8 LTS

- Integration in TYPO3 8.5 ✓
- Viele Verbesserungen in TYPO3 8.6 (erscheint am 14. Februar 2017)
- Design / UI / UX Verbesserungen des Editors
- EXT:form-legacy ins TER
- Dokumentation
- Snippet Datenbank
- Beispiel Extensions
- Testen, Feedback geben!

Links

- Beispiele für Extensions:
<https://code.tritum.de/explore/projects>
- Slack Channel:
<https://typo3.slack.com/archives/ext-form>
- Demo Video:
<https://www.youtube.com/watch?v=F9sTA0EcTI0>

Vielen Dank!

Twitter: @cybersmog

Mail: peter.kraume@typo3.org

Slides: <http://de.slideshare.net/pk77/>



This work is licensed under a [Creative Commons Attribution-ShareAlike 4.0 International License](https://creativecommons.org/licenses/by-sa/4.0/).