

Contao Formulare

-

**Daten speichern und weiter
verarbeiten**

Inhalt

- Über
- Formularfelder und ihre Eigenheiten
- Analyse der Anforderungen
- Verschiedene Ansätze
 - Speichern mit EFG
 - Speichern in Contao Tabelle
 - Speichern in Datei
 - Speichern in Leads
 - Speichern in MetaModels
- Fazit
- Fragen / Diskussion

Über

Cliff Parnitzky

- Softwareentwickler seit > 15 Jahren
- Contao seit > 7 Jahren
- liebt die Extension Entwicklung
- außerdem: leidenschaftlicher Triathlet

Web Profile

- Forum: [cliffen](#)
- Twitter: [@ContaoCliff](#)
- GitHub: [cliffparnitzky](#)
- Packagist: [cliffparnitzky](#)



Formularfelder und ihre Eigenheiten

- Datum:
 - Formatieren
 - Sortieren
 - Filtern
- Text:
 - Schreibrichtung
 - Längenbegrenzung
 - Zeilenumbruch/HTML
- Checkbox/Radio Button/Select:
 - Value vs. Label
 - Ausgabe von Listen
- Referenzen:
 - ID vs. Daten

Analyse der Anforderungen

Fragen

- Welche Daten sollen gespeichert werden?
- Wie sollen die Daten gespeichert werden?
- Wie sollen die gespeicherten Daten dargestellt werden?
- Wie sollen die gespeicherten Daten modifiziert werden?
- Wie sollen die gespeicherten Daten weiter verarbeitet werden?
- usw.

Randparameter

- Verschlüsselung
- Zugriff auf die gespeicherten Daten
- Datenaufkommen
- usw.

Speichern mit EFG



Speichern in Contao Tabelle

- Benötigt: Boardmittel
- Schwierigkeit: mittel
- Szenario: anlegen von Blogbeiträgen aus dem Frontend

Vorteile

- Daten im Backend einsehbar/änderbar/löschbar
- Inhaltselemente/Module zur Anzeige im Frontend sind vorhanden

Nachteile

- Default Werte müssen z.B. per Verstecktem Feld vorgegeben werden (unsicher)
- Technisches KnowHow wird benötigt (DB Struktur der Tabelle)
- Nur Felder der Tabelle stehen zur Verfügung
- semantisch vielleicht nicht immer richtig

Speichern in Datei

- Benötigt: [terminal42/notification_center](#)
- Schwierigkeit: einfach
- Szenario: Formulardaten erfassen und für Fremdsystem verfügbar machen ([Live Beispiel](#))

Vorteile

- schnell & einfach
- verschiedene Möglichkeiten die Datei zu benennen/aufzubauen (Inserttags, an Datei anhängen)
- Dateiinhalt frei formatierbar (CSV, XML, JSON, etc.)
- Zugriff auf Datei via FTP/Web

Nachteile

- Datei kann einfach gelöscht werden (Datenverlust)
- Daten nicht via View im Backend einsehbar/änderbar/löschbar (außer per Dateiverwaltung)
- Dateihandling kann in Verarbeitung kompliziert sein

Speichern in Leads

- Benötigt: [terminal42/contao-leads](#)
[phpoffice/phpexcel](#)
- Optional: [hofff/contao-leads-encryption](#)
- Schwierigkeit: mittel
- Szenario: Bankdaten erfassen und per CSV exportieren ([Live Beispiel](#))

Vorteile

- Daten im Backend einsehbar/löschbar (Änderung technisch möglich, funktional aktuell nicht vorgesehen)
- umfangreiche Konfiguration (Backend Integration, welche Felder sollen gespeichert werden)
- komfortabler, flexibler CSV/Excel Export
- es gibt Zusatzextensions

Nachteile

- keine Inhaltselemente/Module zur Anzeige im Frontend vorhanden (aktuell nicht vorgesehen)
- je Formularversand wird pro gespeichertem Feld ein DB Eintrag erzeugt ([SQL komplex](#))

Speichern in MetaModels

- Benötigt: [metamodels/core](#)
- Optional: [aktuell 92 Dependents](#)
- Schwierigkeit: komplex
- Szenario: Erfassung von Anmeldungen zu einem Sport-Event

Vorteile

- Daten im Backend einsehbar/änderbar/löschbar (volle, flexible Backend Integration)
- Inhaltselemente/Module zur Anzeige im Frontend vorhanden (Filter, Listen, ...)
- volle Frontend Integration durch "FE Editing" im Beta-Stadium
- volle Kontrolle über Datenstruktur und gespeicherte Daten

Nachteile

- komplex und umfangreich in der Einrichtung
- benötigt technisches Verständnis und u.U. Programmierkenntnisse

Fazit

- möglichst viele Fragen zur Analyse beantworten
- vorausschauend planen und gesamten Business Case betrachten
- rechtliche/technische Randbedingungen prüfen (lassen)

Vermeidung von Migrationsaufwänden