form4 Fileuploader

Abnahmeaufforderung

Stand: 28. Oktober 2020

Inhalt

1 Abnahmeaufforderung

Sehr geehrte Damen und Herren,

die Realisierung des Auftrags ist teilweise abgeschlossen und ich bitte Sie um die Abnahme des unter 1.1 aufgeführten Abnahmegegenstandes. Die Abnahme erfolgt auf Basis der Testinstanz unter http://fileuploader-test-dev72.form4.de/. Die Repositories für den Source Code befinden sich im GitLab:

- Backend https://git.form4.de/form4/typo3/typo3.extensions/form4_fileuploader
- Frontend https://git.form4.de/form4/frontend/form4fileuploader.js

1.1 Abnahmegegenstand

1.1.1 Komptabilität

Der PHP-Code läuft unter PHP 7.2 und die Extension ist kompatibel sowohl mit TYPO3 8.7 als auch TYPO3 9.5 und höher. Die Lauffähigkeit im IE11 muss geprüft werden, da dies unter macOS nicht möglich war.

1.1.2 Allgemein

Es wurde eine Extension angelegt und konfiguriert. Implementiert wurden ein Ajax Handler, Fileupload Service, die Konfigurationsschnittstelle und einige Processors zur Verarbeitung/Validierung der Daten. Für den Ajax Handler wurde im TypoScript ein Page Type definiert, der mittels userFunc die handleRequest-Methode des Handlers aufruft. Der Ajax Handler bereitet die Daten für den Fileupload Service vor und ruft dann eine entsprechende Methode dessen auf und übergibt die Daten.

1.1.3 Fileupload Service

Der Fileupload Service bildet den Kern des Fileuploaders und stellt nach außen eine API zur Verfügung. Von dort aus wird alles gesteuert und die Daten verarbeitet und validiert. Außerdem kümmert dieser sich um die Integritätssicherung der Daten.

1.1.4 Integritätssicherung

Die Integrität wird mittels einer Checksumme sichergestellt, die berechnet wird, wenn Dateien für das Preloading hinzugefügt werden, Dateien hochgeladen oder entfernt werden. Außerdem wird eine API zur Validierung der Checksumme bereitgestellt.

1.1.5 Konfigurationsschnittstelle

Die Konfigurationsschnittstelle wurde als Singleton implementiert. Eine Konfiguration kann mittels eines eindeutigen Keys registriert und Subkonfigurationen vorgenommen werden. Subkonfigurationen erhalten ebenfalls einen eindeutigen Key und es können ein Processor und Settings definiert werden.

1.1.6 Verarbeitung/Validierung

Die Validierung wurde mittels der zuvor genannten Processors realisiert. Diese besitzen mehrere Hooks, mit welchen Dateien verarbeitet bzw. validiert werden können. Dafür wurde eine Abstrakte Klasse implementiert, von der jeder Processor erben muss. Folgende Processors wurden im Core implementiert:

- Destination — Für das Verschieben der Dateien in ein Zielverzeichnis
- Max File Size — Zur Validierung der Dateigröße einzeln
- Max Upload Size — Zur Validierung der Gesamtgröße aller Dateien
- Min Files — Zur Validierung der Mindestanzahl der Dateien
- Max Files — Zur Validierung der Maximalanzahl der Dateien
- MIME-Type — Zur Validierung von MIME-Types
- File Extension — Zur Validierung von Dateiendungen

1.1.7 Viewhelper

Der Viewhelper wurde umgesetzt und rendert zwei HTML Formular-Felder. Zum einen das File-Feld, das zum Hochladen von Dateien via Ajax dient und zum anderen ein Hidden-Feld, welches in das jeweilige Extbase-Formular integriert wird und die Daten enthält, die beim Submit an das Backend gesendet werden sollen. Der Viewhelper rendert außerdem die Konfiguration in einem Data-Attribut zur Frontend-Validierung.

1.1.8 Frontend

Das Frontend besteht im Kern aus einer Fileuploader Klasse deren Zustand in einem observable Store verwaltet wird. Zudem aus einer Processors Registry zum Registrieren von Processors und einigen vorimplementierten Processors, wie im Backend. Processors werden im Frontend anders als im Backend global registriert und erhalten ihre entsprechenden Einstellungen erst bei Ausführung.

Die Fileuploader Klasse hat eine API zum Lauschen auf Änderungen ihres Zustandes und zum Entfernen von Elementen. Die Bibliothek stellt außerdem eine API zur Initialisierung und zum Registrieren von Processors unter dem globalen Namespace Form4Fileuploader zur Verfügung.

Darüber hinaus gibt es neben einer statischen Bibliothek die Möglichkeit, das Frontend in einem Modulbasierten Frontend als NPM Package einzubinden.

1.2 Nicht/Teilweise umgesetzt

Folgender Features bzw. Funktionalitäten wurden nicht oder nur teilweise umgesetzt:

1.2.1 Fehlerbehandlung

Die Fehlerbehandlung ist noch nicht vollständig durchdacht und umgesetzt worden und daher noch sehr rudimentär.

1.2.2 API zur abschließenden Validierung

Die API zur abschließenden Validierung im Fileupload Service (Backend) ist vorhanden, aber nicht vollständig implementiert.

1.2.3 Entwicklerhandbuch/-dokumentation

Es gibt noch keine vollständige Entwicklerdokumentation. Es sind einige PHPDoc-Kommentare im Backend und JSDoc-Kommentare im Frontend vorhanden. Außerdem gibt es für das Frontend teilweise eine Entwicklerdokumentation im HTML-Format, welche in das Repository eingecheckt ist.