

Dateityp-Einschränkungen für studentische Datei- Einsendungen

Erlaubte Dateiendungen und MIME-Types für Datei-Upload-
Felder in Aufgabenseiten festlegen

Autor:

Immo Schulz-Gerlach,
FernUni Hagen, ZMI, FeU-
Anwendungen

Version:

1.1 vom 19.05.2014

Inhaltsverzeichnis

Motivation	3
Wirkungsweise	4
Filterung auf Basis von Dateinamens-Extensionen und MIME-Types	4
Client- und serverseitige Auswertung	4
Auswirkung auf den Browser	5
Auswirkung im Online-Übungssystem	7
Zuverlässigkeit	8
Einrichtung für ein Dateiupload-Formularfeld	10
Assistentengestützte Aufgabenerstellung	10
Fortgeschrittene Aufgabenerstellung	10
HTML-Datei bearbeiten	10
Hinweis zu Teilaufgaben-Zerlegung	11
HTML-Datei auswerten	12
Zulässige Werte	12

Motivation

Je nach Einrichtung können Aufgaben des Online-Übungssystems verschiedene Arten von Eingaben entgegennehmen, z.B. „Ankreuzen“ von Checkboxen oder Radiobuttons, Klartext-Eingaben in reinem oder (HTML-)formatiertem Text oder auch Datei-Uploads. In letzterem Fall erstellen Studenten lokal Lösungen in Dateiform (z.B. als PDF) und senden diese dann ein, indem sie die Aufgabenseite im Online-Übungssystem öffnen und dort im entsprechenden Formularfeld die lokale Datei auswählen und einen Button zum Hochladen der ausgewählten Datei betätigen. Dazu muss natürlich zuvor der Aufgabenautor in dem Aufgabenformular ein Dateiupload-Feld eingefügt haben.

Standardmäßig erlaubt ein solches Dateiupload-Feld in Aufgabenseiten das Hochladen beliebiger Dateien. Es können aber im Aufgabentext gewünschte/unterstützte Dateiformate genannt werden, zusammen mit dem Hinweis, dass Einsendungen in anderen als den laut Aufgabenstellung erlaubten Formaten nicht korrigiert und bewertet werden.

Seit Mai 2014 gibt es nun auch eine technische Option, akzeptierte Dateiformate einzuschränken und den Upload anderer Formate vom Übungssystem ablehnen zu lassen. Der Vorteil ist, dass man (normalerweise¹) keine unerlaubten Dateitypen zur Korrektur erhält. Ein Nachteil ist, dass technische Probleme (oder auch Fehler bei der Einrichtung durch den Aufgabenautor), die zur Annahmeverweigerung einer eigentlich erlaubten Datei führen könnten, nicht zu 100% auszuschließen sind.

Die folgenden Kapitel beschreiben die Wirkungsweise und Einrichtung dieser Maßnahme.

Wirkungsweise

Filterung auf Basis von Dateinamens-Extensionen und MIME-Types

Um den Dateityp einzuschränken, legen Sie eine Liste von Dateinamens-Endungen und von [Internet Media Types](http://de.wikipedia.org/wiki/Internet_Media_Type) <http://de.wikipedia.org/wiki/Internet_Media_Type> (im Folgenden mit dem geläufigeren Namen *MIME-Types* bezeichnet) fest, die akzeptiert werden sollen. Alles Aufgezählte ist erlaubt, d.h. eine Ein-sendung wird akzeptiert, wenn der Name der Datei auf einer der genannten Extensionen endet *oder* wenn der Browser als Content-Type der Datei einen der in der Liste genannten MIME-Types mit-sendet.

Um zum Beispiel nur PDF-Dokumente und Jpeg-Bilddateien zu akzeptieren, könnte eine solche Liste wie folgt aussehen:

```
.pdf, .jpg, .jpeg, application/pdf, image/jpeg
```

- Genauere Informationen zum Aufbau dieser Liste finden Sie im Abschnitt [Zulässige Werte](#).
- [Liste offizieller MIME-Types](http://www.iana.org/assignments/media-types/media-types.xhtml) <<http://www.iana.org/assignments/media-types/media-types.xhtml>> laut [Internet Assigned Numbers Authority \(IANA\)](http://www.iana.org) <<http://www.iana.org>>

Theoretisch können Sie sich auf eine dieser Angaben (Dateiextension oder MIME-Type) beschränken, es wird jedoch dringend empfohlen, beide Angaben zu kombinieren.² Die Kombination erhöht einer-seits die Zuverlässigkeit im Sinne der Reduktion der Wahrscheinlichkeit, dass Dateien fälschlicher-weise vom Übungssystem als unzulässig abgelehnt werden – gerade der Einsatz von MIME-Types alleine ist nicht besonders zuverlässig. Andererseits erhöht der Einsatz von MIME-Types die Flexibilität: Es gibt Formate, bei denen Dateien mit gleichem MIME-Type verschiedene Endungen zur Unterscheid-ung von „Unter-Formaten“ aufweisen können. Z.B. sind, um obiges Beispiel wieder aufzugreifen, für JPEG-Bilddateien sowohl die Endung `.jpg` als auch `.jpeg` üblich. Hätten wir oben die zweite Schreibweise vergessen, also nur `.pdf, .jpg, application/pdf, image/jpeg` notiert, sollte ein Einsenden von Dateien mit Endung `.jpeg` in der Regel dennoch funktionieren – sofern der Browser korrekterweise den `image/jpeg`-MIME-Type mitsendet.

Ein anderes Beispiel sind Dateien mit Endungen wie `.xsl`, `.xsd`, `.wsdl` etc., die sämtlich bes-timmte „Spezialfälle“ von XML-Dateien sind und meist mit Content-Type `text/xml` gesendet wer-den. Eine Angabe nur der Dateiendung `.xml` würde solche Dateien also ausschließen, eine Angabe wie `.xml, text/xml` würde sie dagegen normalerweise zulassen. Um sicher zu gehen, sollten aber auch alle Dateiendungen, die man explizit erlauben will, zusätzlich aufgezählt werden.

Client- und serverseitige Auswertung

Die oben beschriebene Liste von Dateiextensionen und MIME-Types wird an zwei Stellen im Prozess-ablauf ausgewertet: Bereits beim Ausfüllen des Formulars im Browser sowie vom Übungssystem bei Verarbeitung eines Uploads:

Auswirkung auf den Browser

Die Liste mit zu akzeptierenden Dateiendungen und MIME-Typen wird als Teil des Web-Formulars an den Webbrowser übermittelt. Dieser *kann* damit bereits eine Vorabprüfung vornehmen und den Upload oder sogar die Auswahl von nicht auf dieses Muster passenden Dateien unterbinden. Ob und wie zuverlässig dies funktioniert, hängt aber sehr stark vom Browser ab.

Safari 7 z.B. wertet diese Angaben aus, wann immer ein Nutzer den Dialog zur Auswahl der hochzuladenden Datei öffnet. Hier kann der Benutzer nur „passende“ Dateien selektieren, alle anderen sind ausgegraut. Allerdings lässt Safari 7 es zu, beliebige Dateien per Drag & Drop direkt auf den Auswahl-Button im Web-Formular zu ziehen, und lädt diese dann komplett ohne Abgleich mit dem Namens- oder MIME-Type-Schema hoch.

Internet Explorer 11 bietet ebenfalls standardmäßig nur „passende“ Dateien an, „unpassende“ werden sogar nicht nur ausgegraut, sondern überhaupt nicht angezeigt. Jedoch bietet der Datei-Öffnen-Dialog des Internet-Explorers unten rechts eine Auswahlmöglichkeit, mit der sich statt der „benutzerdefinierten Dateien“ wieder „alle Dateien“ anzeigen und auswählen lassen, vgl. Abb. 1.1:

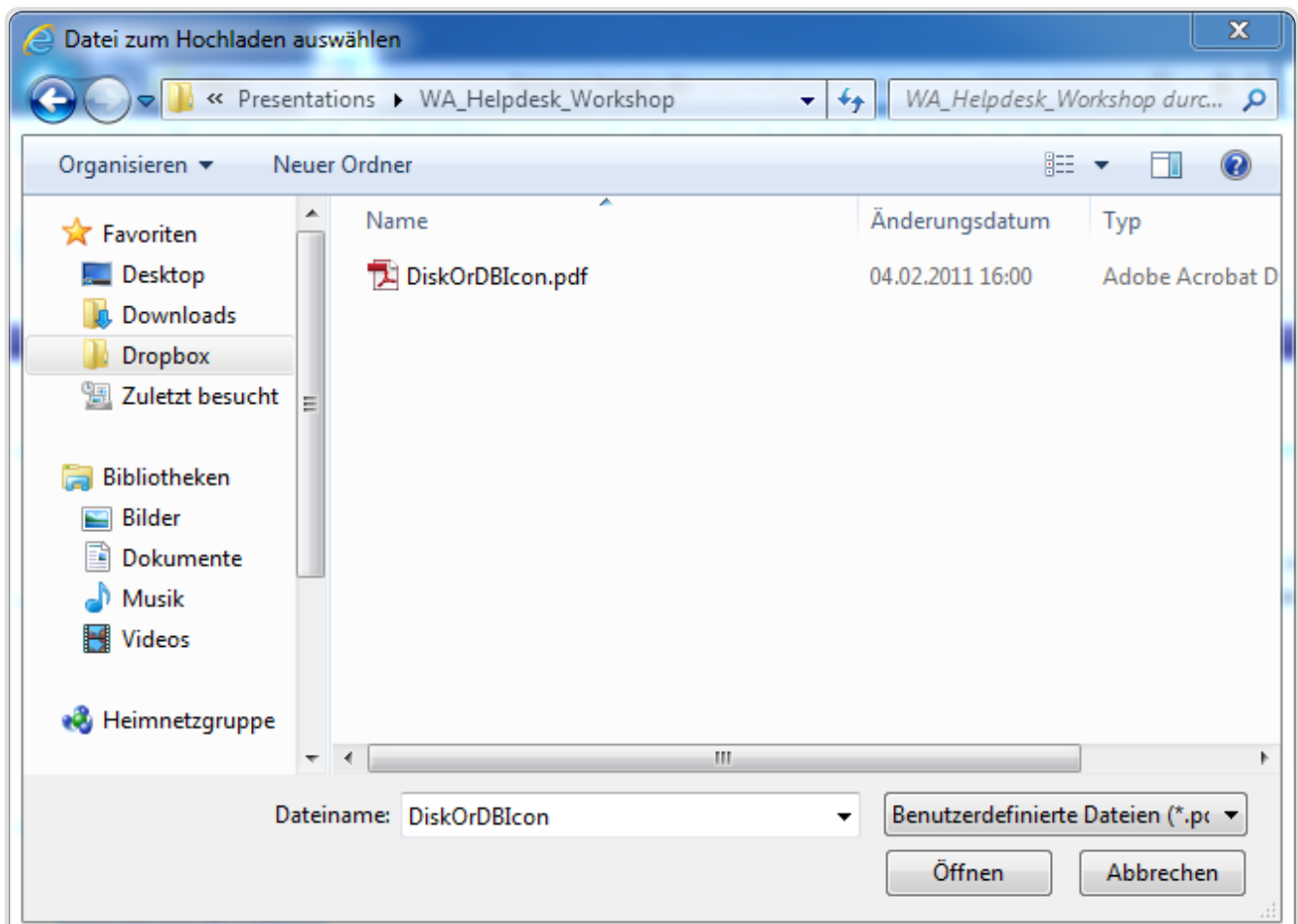


Abb. 1-1: Internet-Explorer-Dateiauswahldialog

Firefox 29 arbeitet ähnlich, stellt die Format-Auswahl unten im Öffnen-Dialog aber sogar standardmäßig auf „Alle Dateien“ ein. Firefox ignoriert also die Format-Einschränkungen praktisch per Default, der Nutzer kann höchstens *manuell* explizit eine Einschränkung des Auswahldialogs auf einen akzeptierten Typ aktivieren.

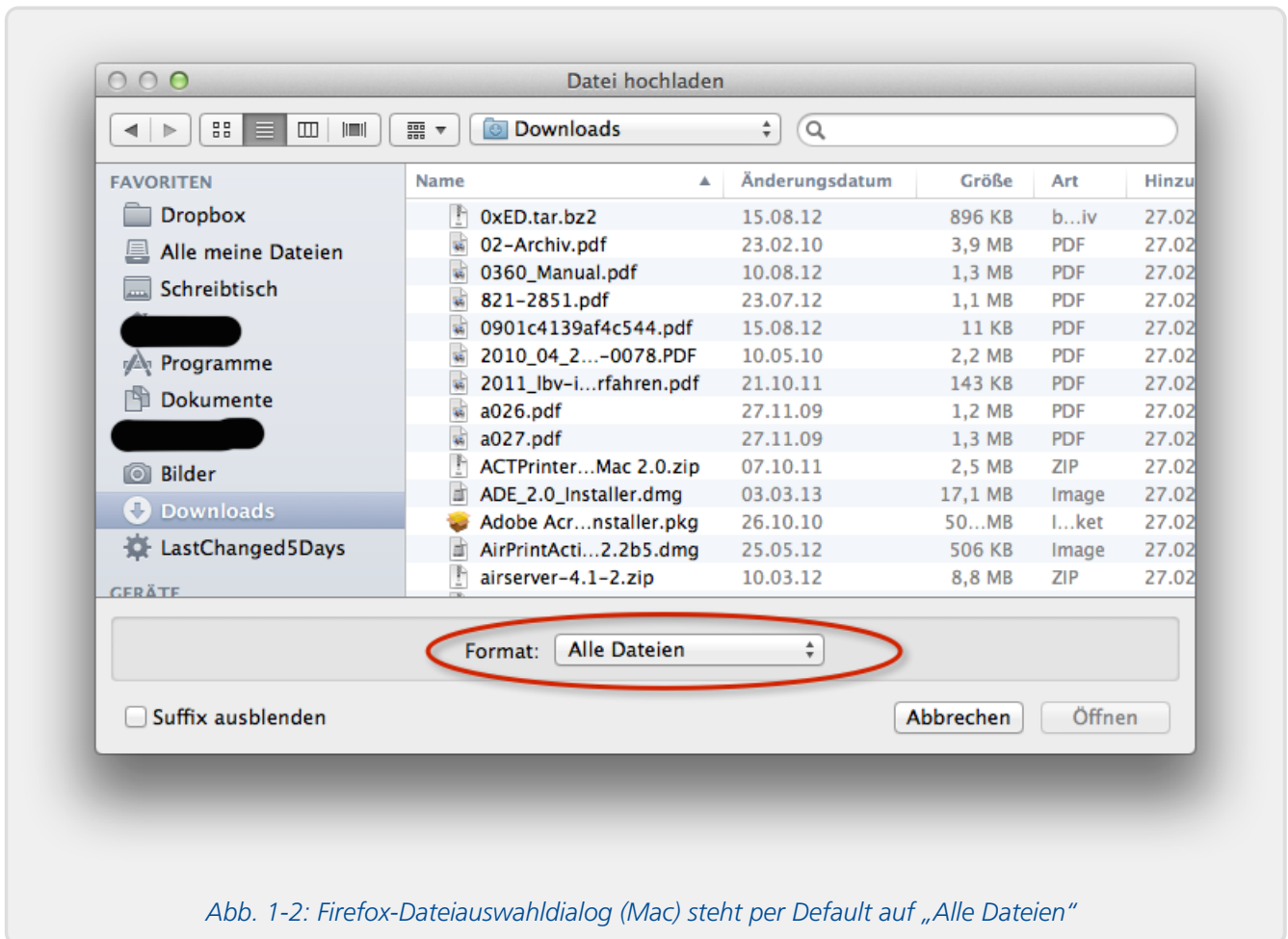


Abb. 1-2: Firefox-Dateiauswahldialog (Mac) steht per Default auf „Alle Dateien“

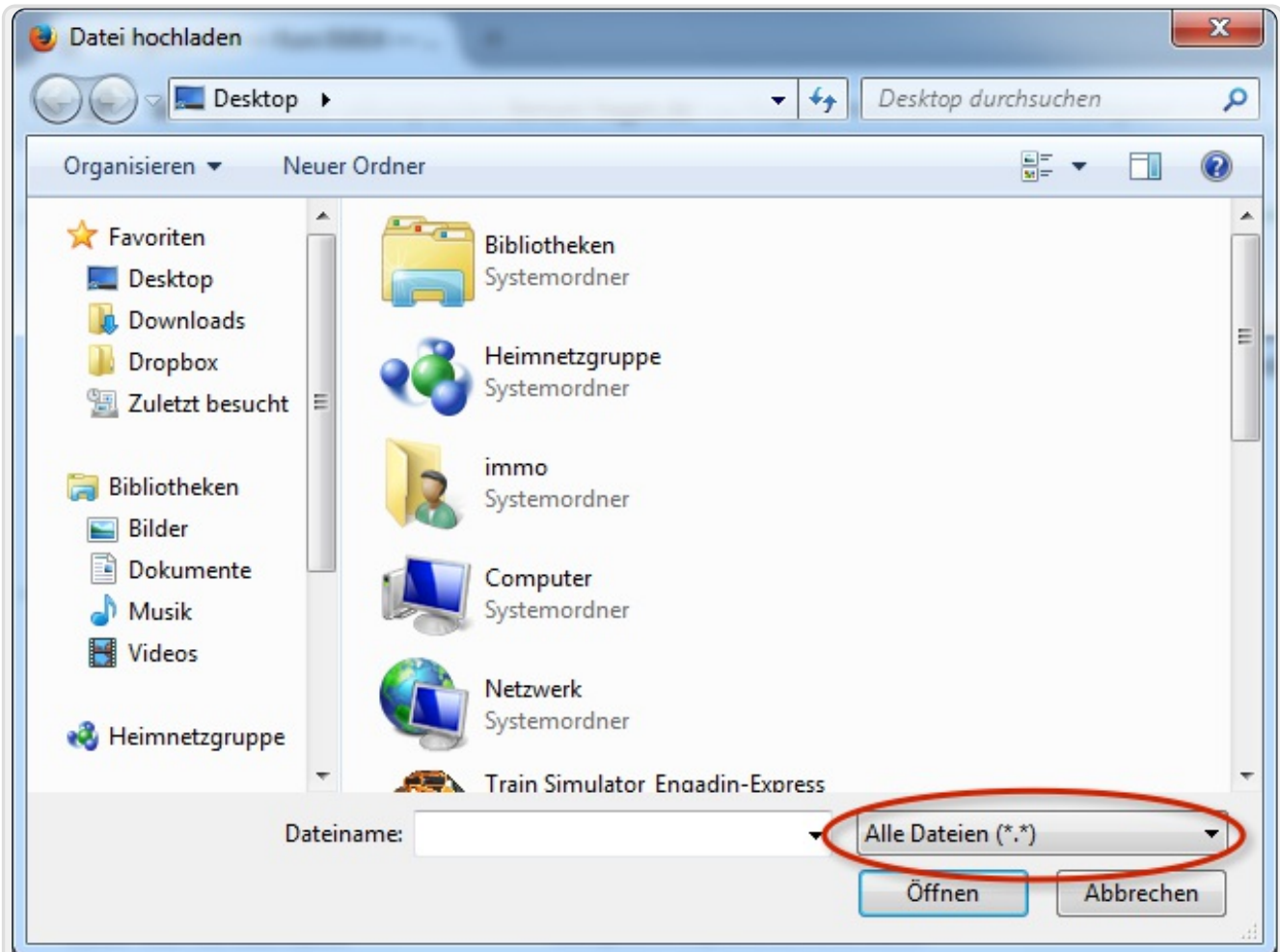


Abb. 1-3: Firefox-Dateiauswahldialog (Windows), ebenfalls auf „Alle Dateien“ voreingestellt

Das Verhalten dieser Browser könnte sich mit zukünftigen Versionen ändern, weshalb oben jeweils die Versionsnummer des letzten Tests mit angegeben wurde.

Auswirkung im Online-Übungssystem

Da man sich nicht darauf verlassen kann, dass der Browser nur zum Muster passende Dateien ans Übungssystem einsendet bzw. den Studenten am Hochladen unerlaubter Dateien hindert (siehe oben), prüft anschließend auch noch das Online-Übungssystem, ob der Browser zur hochgeladenen Datei einen der genannten MIME-Typen mitgesendet hat oder der Name der Datei auf einer der genannten Extensionen endet. Ist das nicht der Fall, wird die gesamte Einsendung nicht akzeptiert; dem Studenten wird eine entsprechende Fehlermeldung angezeigt:

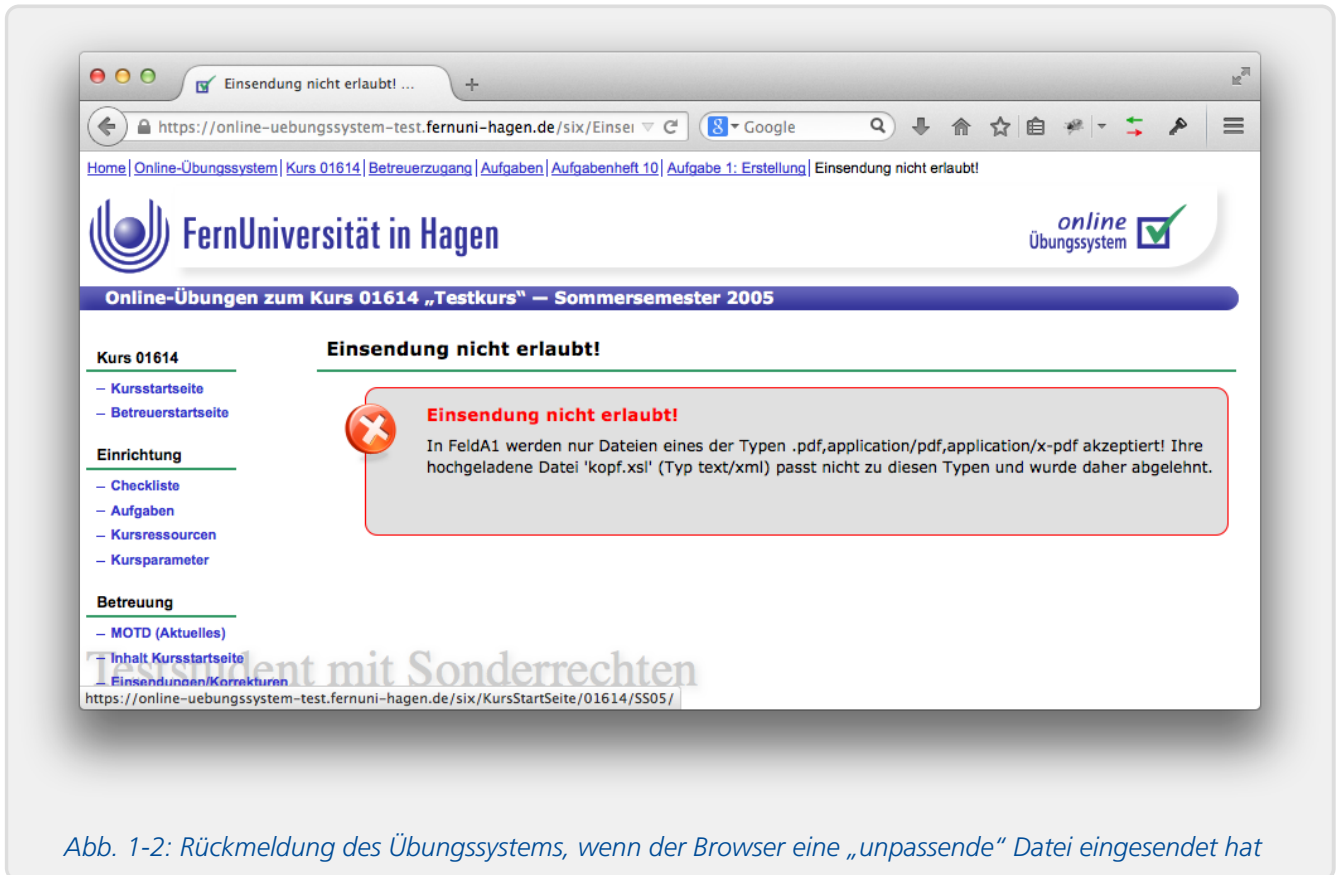


Abb. 1-2: Rückmeldung des Übungssystems, wenn der Browser eine „unpassende“ Datei eingesendet hat

Zuverlässigkeit

Diese Wirkungsweise auf Basis einer Liste von Dateierweiterungen und MIME-Typen kann prinzipbedingt nicht absolut zuverlässig funktionieren, da lediglich Metainformationen zur Datei untersucht werden, aber keine Dateiinhalte. Es ist z.B. möglich, eine Word-Datei so umzubenennen, dass ihr Name nicht mehr auf `.doc` bzw. `.docx` endet, sondern auf `.pdf`. Es bleibt (vom Inhalt her) zwar immer noch eine Word-Datei, System und Browser halten sie aber i.d.R. dann wegen ihres Namens für eine PDF-Datei. Entsprechend funktioniert dann in der Regel der Upload dieser umbenannten Word-Datei in einer Übungssystem-Aufgabe, die eigentlich ausschließlich PDF-Dateien akzeptiert. In solchen Fällen kann eine Korrektur der „defekten PDF-Datei“ aber ruhigen Gewissens vom Korrektor verweigert werden.

Umgekehrt ist es natürlich auch möglich, dass eine Datei, die eigentlich akzeptiert würde, nicht hochgeladen werden kann. Ein triviales Beispiel ist z.B., dass ein Benutzer seine PDF-Datei `loesung2.pdf` nennen möchte, sie aber versehentlich `loesung.2pdf` nennt. Eine reine Dateinamens-Beschränkung des Upload-Feldes auf Dateien, deren Name mit `.pdf` endet, würde diese Datei somit ablehnen. Etwas höher ist die Chance, wenn neben der Dateierweiterung `.pdf` auch der MIME-Type `application/pdf` akzeptiert wird: Falls das hochladende System bzw. der Browser intelligent genug sein sollte, zu erkennen, dass es sich bei `loesung.2pdf` um eine PDF-Datei handelt und diese also als Datei des Typs `application/pdf` einsendet, würde sie akzeptiert. Allerdings erfolgt die Bestimmung des vom Browser gesendeten MIME-Typs meistens auch nur auf Basis der Dateierweiterung, und da `.2pdf` unbekannt ist, würde ein generischer MIME-Type für Binärdateien unbekanntem Type wie `application/octet-stream` ([Referenz](http://www.iana.org/assignments/media-types/application/octet-stream)) `<http://www.iana.org/assignments/media-types/application/octet-stream>` gesendet.

Wenn Sie also in solchen Fällen von falsch benannten Dateien die Datei im Zweifel lieber akzeptieren als ablehnen möchten, so können Sie den MIME-Type `application/octet-stream` zur Liste der akzeptierten Typen hinzufügen.

Erschwerend kommen Probleme wie die MIME-Type-Verwaltung des Browsers *Firefox* hinzu, die „selbstlernend“ eine Zuordnung von MIME-Types zu Dateinamens-Endungen verwaltet und dabei auch manchmal falsche Zuordnungen „lernt“. So kann es z.B. vorkommen, dass Firefox PDF-Dateien fälschlicherweise immer mit Content-Type `text/html` sendet ([Externer Blog-Artikel <http://techblog.procurios.nl/k/news/view/15872/14863/mimetype-corruption-in-firefox.html>](http://techblog.procurios.nl/k/news/view/15872/14863/mimetype-corruption-in-firefox.html)).

Um die Wahrscheinlichkeit fälschlicher Ablehnungen zu minimieren, auch wenn dabei die Wahrscheinlichkeit fälschlicher Akzeptanz steigt, sollten also in jedem Fall beide Maßnahmen (Angabe von MIME-Types und von Dateinamens-Extensionen) kombiniert eingesetzt werden. Damit wird insbesondere auch das oben genannte Firefox-Problem entschärft.

Einrichtung für ein Dateiupload-Formularfeld

Dieses Kapitel beschreibt, wie Sie eine solche Liste von Dateierweiterungen und MIME-Typen in ein Aufgabenformular konkret einbinden.

Assistentengestützte Aufgabenerstellung

Wenn Sie die Aufgabenseite mit dem Aufgabenerstellungs-Assistenten für handbewertete Aufgaben zusammenstellen, fügen Sie zunächst ein Element für einen studentischen Datei-Upload in die Aufgabenseite ein. Darin können Sie direkt die Dateityp-Einschränkung einstellen:

Abb. 2-1: Datei-Upload-Element im Aufgabenerstellungs-Assistenten

Unter der Auswahl „Vordefinierte Profile“ sind die wichtigsten Einschränkungen schon vorgegeben, insbesondere „PDF-Dateien“ und „Bilddateien“ sowie die Voreinstellung „Beliebige Dateien“. Wählen Sie letztere aus, um keine Einschränkung zu definieren, sondern wieder den Upload beliebiger Dateien zu erlauben. Wählen Sie eine andere Voreinstellung wie „PDF-Dateien“ aus, so wird das Filter-Feld automatisch auf einen vordefinierten Wert gesetzt. (Neben dem Filter-Feld werden bei Auswahl eines vordefinierten Profils auch die anderen Werte, also Beschriftung und Checkbox „... Einsendung grundsätzlich als Link anzeigen“, auf die typischen zum gewählten Dateityp passenden Werte gesetzt³. Diese Vorschläge können Sie aber anschließend ändern.)

Um einen eigenen Filter zu definieren, tragen Sie unter „Filter (HTML-Accept-Attribut)“ die gewünschte Liste von Dateierweiterungen und MIME-Typen direkt ein. (Die Profilauswahl springt dann um auf den Eintrag „Benutzerdefiniert“.)

Genauere Informationen zu zulässigen Eintragungen in diesem Feld finden Sie im Abschnitt [Zulässige Werte](#).

Fortgeschrittene Aufgabenerstellung

Wenn Sie nicht mit dem Assistenten arbeiten, sondern im fortgeschrittenen Modus die Aufgabenseite als HTML-Datei bearbeiten, sind zwei Schritte nötig: Anpassung der HTML-Aufgabenseite und anschließend eine Auswertung dieser Seite durchs Online-Übungssystem.

HTML-Datei bearbeiten

In einer HTML-Seite wird ein Datei-Upload-Element durch ein Input-Element mit Attribut `type="file"` realisiert. Hier können Sie Ihre Liste von Dateierweiterungen und MIME-Typen im speziellen Attribut `accept` mit eintragen. Das folgende Listing z.B. zeigt den entsprechenden Teil der Aufgabenseite, wie er vom Assistenten in [Abb. 2-1](#) erzeugt wurde:

```
<p>PDF hochladen:
  <input type="file" name="FeldA1" accept=".pdf,application/pdf,application/x-
pdf">
  <input type="submit" name="einsendenA" value="PDF einsenden"><br>
  $IfExistsA1(Ihre zuletzt hochgeladene Datei: )$FeldA1HREF
</p>
```

- Spezifikation zum `accept`-Attribut in der [HTML5-Spezifikation](http://www.w3.org/TR/html5/forms.html#attr-input-accept)
<<http://www.w3.org/TR/html5/forms.html#attr-input-accept>>
- Weitere Informationen zu zulässigen Eintragungen im `accept`-Attribut finden Sie im Abschnitt [Zulässige Werte](#).

Hinweis zu Teilaufgaben-Zerlegung

Falls Ihre Aufgabe nicht nur ein Dateiupload-Feld als einzige Eingabemöglichkeit enthält, sondern die Studenten mehrere Felder ausfüllen sollen (mehrere Dateiuploads oder ein Dateiupload und zusätzlich Textfelder o.ä.), so sollte jeder Datei-Upload jeweils eine eigene Teilaufgabe mit eigenem Submit-Button bilden! In obigem Listing z.B. hat das Feld den Namen `FeldA1`, ist also das erste Eingabefeld zur Teilaufgabe A. Dazu findet sich ein Submit-Button namens `einsendenA`, der den Upload der Datei aus `FeldA1` auslöst. Nun sollte es in der Aufgabe keine weiteren Inputs mit Namen `FeldAn` geben (wobei `n` eine Nummer $\neq 1$ ist)! Soll nun z.B. der Student zusätzlich noch ein Textfeld ausfüllen und seine Eingaben dort einsenden können, so sollte dies in einer neuen Teilaufgabe B erfolgen, d.h. das Text-Eingabefeld soll nicht `FeldA2`, sondern `FeldB1` heißen, und dazu wird ein gesonderter Submit-Button `einsendenB` benötigt.

Der Hauptgrund für diese Empfehlung ist – unabhängig von Dateityp-Einschränkungen – folgender:

- Der Submit-Button einer Teilaufgabe sendet die Daten aller Felder derselben Teilaufgabe gleichzeitig ans Übungssystem.
- File-Upload-Felder müssen bei jedem Aufruf der Aufgabenseite neu ausgefüllt werden. Wird eine Aufgabenseite erneut geöffnet und eine Teilaufgabe mit Fileupload-Element eingesendet, zu dem keine Datei ausgewählt wurde, so wird eine ggf. in einem früheren Submit zu diesem Feld bereits eingesendete Datei wieder gelöscht. Damit kann ein Student seine Datei-Einreichung zurückziehen.
- Würde man nun aber z.B. einen File-Upload `FeldA1` und ein Textfeld `FeldA2` in einer Teilaufgabe A zusammenfassen und ein Student hätte auch bereits in beiden Feldern etwas eingesendet, rufe nun aber die Aufgabenseite erneut auf, um den Text in `FeldB1` nachzubearbeiten, so müsste er bei jeder Änderung des Textes wieder in `FeldA1` die einzusendende lokale Datei auswählen und mit der Textänderung zusammen erneut einsenden. Bearbeitete er dagegen nur den Text und ließe das Dateiupload-Feld leer, so würde zwar die Textänderung gespeichert, aber die zuvor eingesendete Datei wieder gelöscht!
- Um derartige Fehlbedienungen zu vermeiden, sollen also Dateiupload-Felder immer „atomare“ Teilaufgaben bilden.

Im Kontext mit Dateityp-Einschränkungen gibt es einen weiteren Grund für diese Empfehlung: Bildet ein mit `accept`-Attribut ausgestattetes Dateiupload-Feld eine Teilaufgabe zusammen mit weiteren Eingabefeldern (Dateiupload, Texteingaben, ...) und wird eine nicht zulässige Datei zusammen mit weiteren Eingaben eingesendet, so wird die gesamte Einsendung vom Übungssystem abgelehnt, also nicht nur die Datei nicht gespeichert, sondern gar keine der eingesendete Daten.

HTML-Datei auswerten

Wenn Sie Änderungen an Ihrer Aufgaben-HTML-Datei (`aufgabex.y.html`, wobei `x` die Aufgabenheft- und `y` die Aufgabennummer darstellt) vorgenommen haben – egal ob online oder durch Upload einer offline erstellten neuen Version über den Kursressourcen-Upload –, so muss im Übungssystem eine Auswertung dieser Datei manuell angestoßen werden. (Bei [assistentengestützter Aufgabenerstellung](#) erfolgt dieser Schritt automatisch.)



Abb. 2-2: Fortgeschrittene Aufgabenerstellung: Aufgabenseite analysieren und DB anpassen

In der Aufgabenerstellungsseite finden Sie dazu zwei Buttons (siehe Abb. 2-2): Der „Vorlage analysieren“-Button startet eine reine Analyse und zeigt Ihnen das Ergebnis an, z.B.:

Die Aufgabe enthält 2 Teilaufgaben und 2 Submit-Buttons für die Teilaufgaben A, B.

- * Die Teilaufgabe A enthält 1 Eingabefelder
 - * Das Eingabefeld 'FeldA1' ist ein File-Upload mit `accept=".pdf,application/pdf,application/x-pdf"`.
- * Die Teilaufgabe B enthält 1 Eingabefelder
 - * Das Eingabefeld 'FeldB1' ist eine Textarea.

Hiermit können Sie also z.B. kontrollieren, ob das Übungssystem die Datei korrekt verarbeiten kann, insbesondere auch ob das Accept-Attribut korrekt identifiziert wurde, und ob der Aufbau / die Gliederung in Teilaufgaben sinnvoll ist.

Wenn alles in Ordnung ist, drücken Sie den Button „Teilaufgaben und Eingabefelder“ erzeugen. Dieser bereitet im Wesentlichen die Datenbank vor, damit diese zu jedem Eingabefeld auch Einsendungen von Studenten speichern kann. Die Ausführung dieser Funktion ist aber insb. auch nötig, um den Wert des Accept-Attributs aus dem HTML-Code in die Datenbank zu übernehmen, damit die [serverseitige Auswertung](#) des Accept-Filters funktionieren kann.

Zulässige Werte

Laut [HTML-5-Spezifikation](http://www.w3.org/TR/html5/forms.html#attr-input-accept) (Stand Mai 2014) darf das `accept`-Attribut folgende Werte annehmen:

- Eine komma-separierte Liste von...
 - `image/*` für beliebige Bilddateien⁴,
 - konkreten MIME-Types (vgl. [MIME-Type-Liste der IANA](http://www.iana.org/assignments/media-types/media-types.xhtml) <<http://www.iana.org/assignments/media-types/media-types.xhtml>>),
 - mit `.` beginnenden Strings (Dateinamens-Extensionen).

Das Übungssystem akzeptiert für seine serverseitige Prüfung neben `image/*` auch analoge Muster mit anderen Präfixen als `image/`. Der `*` wird aber tatsächlich nur als Wildcard interpretiert, wenn er in der Form `/*` am Ende einer MIME-Type-Angabe steht.

Fußnoten

1. Das Verfahren ist streng genommen nicht 100%ig zuverlässig, d.h. es können sowohl Dateien „durchkommen“, die kein gewünschtes Format haben, als auch Einsendungen von Dateien abgelehnt werden, deren Format eigentlich akzeptiert würde. Mehr dazu im [Abschnitt zur Zuverlässigkeit](#).
2. Eine Ausnahme von der Empfehlung, zu MIME-Types immer auch die bekanntesten / üblichen Dateierweiterungen mit anzugeben, ist die Angabe `image/*` für beliebige Bilddateien. Diese wird vom HTML5-Standard ausdrücklich als möglicher Eintrag vorgesehen. Natürlich kann man auch diese um eine Auflistung verschiedener Bilddatei-Endungen ergänzen, z.B. `image/*, .jpg, .jpeg, .svg, .gif, .png`, aber mit der Angabe `image/*` allein sollten Browser im Allgemeinen auch problemlos klarkommen.
3. So wird z.B. für Bilddateien die Beschriftung „Bild“ gewählt und die Checkbox nicht aktiviert, so dass eingesendete Bilder, sofern in einem geeigneten Format, vom Browser direkt in der Quittungs- bzw. Korrekturseite angezeigt werden, während z.B. für Text- und XML-Dateien die Checkbox markiert wird, so dass immer ein Download-Link zum Herunterladen der eingesendeten Datei angezeigt werden soll und nicht versucht werden soll, die Textdateiinhalte in die Quittungs- oder Korrekturseite direkt einzubetten.
4. Laut HTML-5-Standard ebenfalls zulässig sind `audio/*` und `video/*`, aber das Übungssystem sollte nicht zur Einsendung großer Dateien wie insbesondere ganzer Videodateien eingesetzt werden!