

# Anforderungskatalog: HTML5- Dateibrowser

Ausbildungsbereich: Technik  
Studiengang: Informatik  
Vorlesung: Web-Engineering I  
Dozent: Martin Spörl

Maximale Punktzahl: 70

## 1 Allgemeine Hinweise

- Die Webpage soll als Single-Page-App erstellt werden. D.h. die Applikation besteht nur aus einer HTML-Datei (+ ggf. mehrere JavaScript & CSS-Dateien)
- Die Webpage darf nur JavaScript, HTML und CSS verwenden
- Keine Frameworks, Plugins, etc.
- Die Dokumentation muss reflektieren, welche Anforderung wie im Ergebnis umgesetzt ist

# Inhalt

1	Allgemeine Hinweise.....	1
2	Übersicht der Punkte je Anforderungen.....	3
3	Anforderungsdetails.....	4
3.1	Laufzeitumgebung .....	4
3.1.1	Sprachen .....	4
3.1.2	Keine Bibliotheken .....	4
3.2	Sicherheit .....	4
3.2.1	Authentifizierung .....	4
3.2.2	Logout .....	4
3.3	Darstellung.....	4
3.3.1	Anpassung an Browserfenstern .....	4
3.3.2	Usability.....	4
3.3.3	Fehlererkennung.....	5
3.3.4	Verzeichnisse & Dateitypen .....	5
3.4	Navigation .....	5
3.4.1	Verzeichnis öffnen .....	5
3.4.2	Ebene Wechseln.....	5
3.4.3	Browser-History .....	5
3.4.4	Wiedererkennung .....	5
3.5	Datei-Handling .....	6
3.5.1	MP4-Video-Datei abspielen .....	6
3.5.2	Audio-Datei abspielen.....	6
3.5.3	Bild-Datei ansehen .....	6
3.5.4	Text-Datei bearbeiten .....	6
3.5.5	Datei herunterladen.....	6
3.5.6	Datei löschen.....	6
3.5.7	Text-Datei erstellen.....	7
3.5.8	Datei hochladen .....	7
3.6	Verzeichnis-Handling .....	7
3.6.1	Verzeichnis löschen.....	7
3.6.2	Verzeichnis erstellen .....	7

## 2 Übersicht der Punkte je Anforderungen

Nr.	Kategorie	Anforderung	Punkte
1	Laufzeitumgebung	Sprachen	3
2		Keine Bibliotheken <b>Achtung: Bei Missachtung werden die Punkte der betroffenen Features nicht gewertet!</b>	4
3	Sicherheit	Authentifizierung	2
4		Logout	2
5	Darstellung	Anpassung an Browserfenster	4
6		Usability	3
7		Fehlererkennung	3
8		Verzeichnisse & Dateitypen	3
9	Navigation	Verzeichnis öffnen	3
10		Ebene Wechseln	3
11		Browser-History	4
12		Wiedererkennung	3
13	Datei-Handling	MP4-Video-Datei abspielen	3
14		Audio-Datei abspielen	3
15		Bild-Datei ansehen	3
16		Text-Datei bearbeiten	4
17		Datei herunterladen	4
18		Datei löschen	3
19		Text-Datei erstellen	4
20		Datei hochladen	3
21	Verzeichnis-Handling	Verzeichnis löschen	3
22		Verzeichnis erstellen	3
			<b>70</b>

## 3 Anforderungsdetails

### 3.1 Laufzeitumgebung

#### 3.1.1 Sprachen

Die Applikation muss in einem modernen Webbrowser ohne zusätzliche Plugins lauffähig sein. Das bedeutet, dass die Applikation nur mit HTML, CSS und JavaScript geschrieben werden muss.

#### 3.1.2 Keine Bibliotheken

Der verwendete Code muss eigenhändig erstellt worden sein. Die Applikation darf nicht auf externe Bibliotheken oder Frameworks aufsetzen noch diese benutzen. Lediglich Standard-HTML5-APIs sind erlaubt.

### 3.2 Sicherheit

#### 3.2.1 Authentifizierung

Bevor der User Inhalte sehen darf, muss sich dieser durch Benutzername und Passwort authentifizieren. Das Passwort darf in der Benutzeroberfläche niemals im Klartext sichtbar werden.

#### 3.2.2 Logout

Der User muss eine Möglichkeit haben, sich aus der Applikation auszuloggen. Dabei sollen auch alle Dateien bzw. Verzeichnis, die er betrachtet hat nicht mehr dargestellt werden. Außerdem sollen alle sicherheitsrelevanten Informationen aus seiner Session gelöscht werden.

### 3.3 Darstellung

#### 3.3.1 Anpassung an Browserfenstern

Die Applikation muss sich an unterschiedliche Fenstergrößen des Browsers anpassen. Dabei sollte, soweit möglich und sinnvoll auf horizontales Scrollen und dem Einsatz von Javascript verzichtet werden.

#### 3.3.2 Usability

Die Darstellung der Applikation soll intuitiv und nach den gängigen Best-Practise erfolgen. Dabei ist vor Allem auf Position der einzelnen UI-Elemente sowie auf die Farbgebung zu achten.

### 3.3.3 Fehlererkennung

Der User soll im Falle von Fehlern seitens der API über diese informiert werden. Also Beispiel soll hier das Aufrufen eines nicht vorhandenen Verzeichnisses dienen. In diesem Fall muss der User über den Fehler informiert werden. Ähnlich sollen Fehler beim Laden von Dateien, Hochladen von Dateien oder dem Löschen von Verzeichnissen sowie beim Erstellen von Element angezeigt werden.

Als Grundlage für die Fehlermeldungen dürfen die zurückgegeben Fehlermeldung der API dienen.

### 3.3.4 Verzeichnisse & Dateitypen

Die Applikation muss die Elemente des ausgewählten Verzeichnisses so darstellen, dass zwischen Verzeichnissen und Dateien unterschieden werden kann.

Außerdem sollen die einzelnen Dateitypen unterschiedlich dargestellt werden.

Hinweis: An dieser Stelle genügt es auf den Mediatyp zu achten. So sollten alle Dateien mit MIMEType „image/XXX“ als Bild-Datei, alle mit „audio/XXX“ als Audio-Datei, usw. betrachtet werden.

## 3.4 Navigation

### 3.4.1 Verzeichnis öffnen

Der User soll die, wenn er ein Verzeichnis auswählt, die Möglichkeit haben diese zu öffnen, sodass er die Elemente in diesem Verzeichnis sieht. Im Folgenden wird diese Ansicht als „Verzeichnisansicht“ bezeichnet.

### 3.4.2 Ebene Wechseln

Innerhalb eines Verzeichnisses soll der User die Möglichkeit haben eine Ebene höher zu gehen – d.h. in den Eltern-Verzeichnis zu wechseln, sofern es sich nicht um das Wurzelverzeichnis handelt.

### 3.4.3 Browser-History

Der User soll über die Vor- und Zurück-Buttons des Browsers ebenso durch das Verzeichnis navigieren können.

### 3.4.4 Wiedererkennung

Der User soll jederzeit wissen, in welchem Verzeichnis er sich aktuell befindet.

## 3.5 Datei-Handling

### 3.5.1 MP4-Video-Datei abspielen

Der User soll die Möglichkeit haben, MP4-Videos direkt im Browser wieder zu geben. Dabei soll der User die entsprechende Video-Datei in der Verzeichnisansicht auswählen können.

### 3.5.2 Audio-Datei abspielen

Der User soll die Möglichkeit haben, Audio-Dateien im MP3 oder OGG Format direkt im Browser wieder zu geben. Dabei soll der User die entsprechende Audio-Datei in der Verzeichnisansicht auswählen können.

### 3.5.3 Bild-Datei ansehen

Der User soll die Möglichkeit haben, Bilder direkt im Browser anzusehen. Dabei soll der User das entsprechende Bild in der Verzeichnisansicht auswählen können.

### 3.5.4 Text-Datei bearbeiten

Der User soll in der Lage sein in der Verzeichnisansicht eine Text-Datei auszuwählen. Diese soll er anschließend anzeigen und bearbeiten können. Die Änderungen soll der User entweder verwerfen können oder mit der Aktion „Speichern“ an den Server übertragen.

Ein etwaiges verfallen des Tokens muss dabei nicht beachtet werden!

### 3.5.5 Datei herunterladen

Unabhängig des Dateityps, soll der User in der Lage sein, die Dateien herunterladen zu können. Dabei sollte der originale Dateiname erhalten bleiben.

### 3.5.6 Datei löschen

Unabhängig des Dateityps, soll der User in der Lage sein, eine existierende Datei zu löschen. Der User soll die Datei in der Verzeichnisansicht auswählen können.

### 3.5.7 Text-Datei erstellen

Im aktuellen Verzeichnis soll der User eine neue Text-Datei erstellen können. Dazu soll der User den Dateinamen und den Inhalt definieren können. Durch die Aktion „Speichern“ soll die Datei erstellt werden.

### 3.5.8 Datei hochladen

Der User soll die Möglichkeit haben eine neue Datei, unabhängig vom Dateityp, hochladen zu können. Dabei darf die Seite weder neu geladen werden noch darf der User auf eine neue Seite verwiesen werden.

Es ist ausreichend, wenn nur 1 Datei gleichzeitig hochgeladen werden kann. Multifile-Upload ist nicht gefordert.

Das Upload-Limit muss ebenfalls nicht beachtet werden.

## 3.6 Verzeichnis-Handling

### 3.6.1 Verzeichnis löschen

Der User soll in der Lage sein, ein Verzeichnis löschen zu können. Dazu soll der User das Verzeichnis in der Verzeichnisanzeige auswählen. Eine Prüfung ob der zu löschende Verzeichnis leer ist, ist nicht notwendig und sollte vom User selbst durchgeführt werden.

### 3.6.2 Verzeichnis erstellen

Der User soll in der Lage sein, ein Verzeichnis erstellen zu können. Dazu soll der User den Verzeichnisnamen angeben können. Durch die Aktion „Speichern“ soll das Verzeichnis erstellt werden.

