

Exkurs: Datei-Upload in Node.js durch `express-fileupload`

QUELLE: <https://attacomsian.com/blog/uploading-files-nodejs-express>

Zur einfachen Realisierung eines Datei-Uploads in Node.js unter Express kann die *Middleware* `express-fileupload` verwendet werden. Diese muss zuerst importiert werden.

```
$ npm install express-fileupload
```

Der Datei-Upload wird in `index.js` folgendermaßen aktiviert:

```
const fileUpload = require('express-fileupload');
app.use(fileUpload());
```

WICHTIG: Diese zwei Zeilen müssen unmittelbar untereinander eingefügt werden.

Dadurch werden in *Request* in der Eigenschaft `files` die hochgeladenen Dateien nebst Eigenschaften und Inhalt bereitgestellt. Der Zugriff auf die Datei erfolgt dabei über den Namen der dem Eingabefeld im Formular gegeben wurde (`importfile`):

<code>request.files.importfile.name</code>	Name der hochgeladenen Datei
<code>request.files.importfile.mv()</code>	Methode zum Verschieben der Datei auf dem Server
<code>request.files.importfile.mimetype</code>	Typ der Datei
<code>request.files.importfile.size</code>	Größe der Datei in Bytes
<code>request.files.importfile.data</code>	Buffer der den Inhalt der Datei in codierter Form enthält

Das *Upload-Formular* hat folgendes Aussehen:

```
<form action="/import" method="post" enctype="multipart/form-data">
  <label for="importfile">Import-Datei:</label>
  <input type="file" id="importfile" name="importfile">
  <input type="submit" value="Importieren">
</form>
```

Zum Lesen der übermittelten Datei und zur unmittelbaren Verarbeitung deren Inhalte ohne Abspeicherung am Server kann folgendermaßen vorgegangen werden:

```
router.post('/import', (request, response) => {
  try {
    const movies = JSON.parse(request.files.importfile.data.toString('ascii'));
    response.send(movies);
  } catch (error) {
    response.send('Falsches JSON-Format');
  }
})
```

Da die Datei in codierter Form vorliegt, muss diese zuerst in eine Zeichenkette umgewandelt und dann ins JSON-Format konvertiert werden.