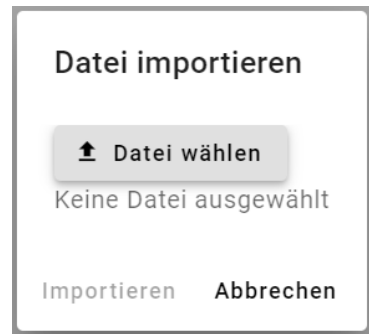


Exkurs: Angular Datei-Upload

Dialog-Komponente für den Upload – Das Template

```
<h1 mat-dialog-title>Datei importieren</h1>
<mat-dialog-content>
  <input hidden (change)="onFileSelected($event)"
    #fileInput type="file">
  <button mat-raised-button (click)="fileInput.click()">
    Datei wählen
    <mat-icon>upload</mat-icon>
  </button>
  <p>
    {{ !currentFile ? 'Keine Datei ausgewählt' : 'Gewählte Datei ' + currentFile.name }}
  </p>
</mat-dialog-content>
<mat-dialog-actions>
  <button mat-button [disabled]="!currentFile" (click)="import()">Importieren</button>
  <button mat-button mat-dialog-close>Abbrechen</button>
</mat-dialog-actions>
```



Dialog-Komponente für den Upload – Die Klasse

```
...
@Component({ ... })
export class ImportDialogComponent {
  currentFile!: File;
  constructor(
    private dialogRef: MatDialogRef<ImportDialogComponent>,
    private uploadService: UploadService,
  ) {}
  onFileSelected(event: any) {
    this.currentFile = event.target.files.item(0);
  }
  import() {
    this.movieService.importFile(this.currentFile)
      .then(response => this.dialogRef.close())
      .catch(error => ... );
  }
}
```

Die Upload-Service-Klasse

```
...
const URL = '...';
@Injectable({ ... })
export class UploadService {
  constructor(private http: HttpClient) { }
  importFile(file: File) {
    const formData: FormData = new FormData();
    formData.append('importfile', file);
    return lastValueFrom(this.http.post<any>(`${URL}/import`, formData));
  }
}
```

HINWEIS: Wie Sie serverseitig die Datei lesen können, wird Ihnen im „Exkurs: Datei-Upload in Node.js durch *express-fileupload*“ gezeigt.