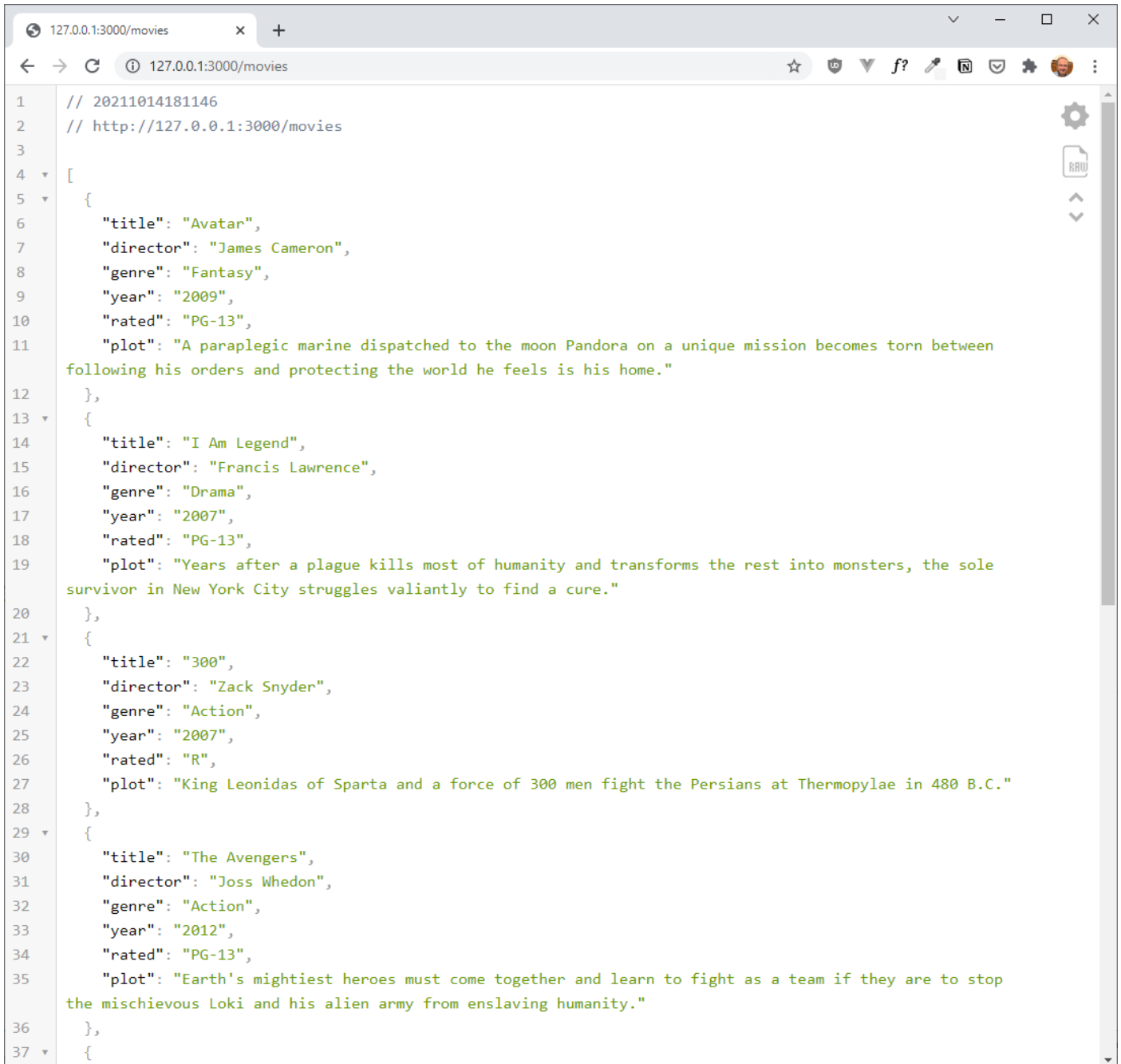


Robert Baumgartner

Lade die Datenbank mit der Datei **movies.sql**. Erstelle einen Server (wie in Unterricht gezeigt) und eine Route GET **/movies**.



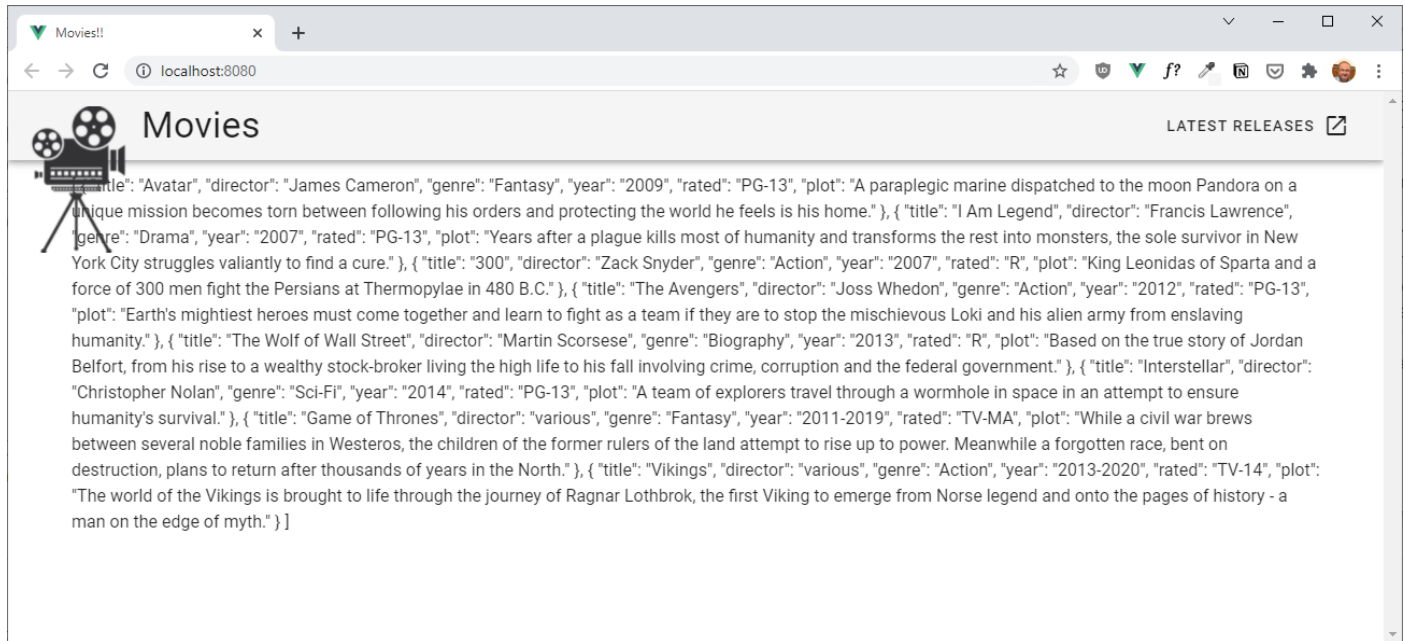
```
1 // 20211014181146
2 // http://127.0.0.1:3000/movies
3
4 [
5   {
6     "title": "Avatar",
7     "director": "James Cameron",
8     "genre": "Fantasy",
9     "year": "2009",
10    "rated": "PG-13",
11    "plot": "A paraplegic marine dispatched to the moon Pandora on a unique mission becomes torn between following his orders and protecting the world he feels is his home."
12  },
13  {
14    "title": "I Am Legend",
15    "director": "Francis Lawrence",
16    "genre": "Drama",
17    "year": "2007",
18    "rated": "PG-13",
19    "plot": "Years after a plague kills most of humanity and transforms the rest into monsters, the sole survivor in New York City struggles valiantly to find a cure."
20  },
21  {
22    "title": "300",
23    "director": "Zack Snyder",
24    "genre": "Action",
25    "year": "2007",
26    "rated": "R",
27    "plot": "King Leonidas of Sparta and a force of 300 men fight the Persians at Thermopylae in 480 B.C."
28  },
29  {
30    "title": "The Avengers",
31    "director": "Joss Whedon",
32    "genre": "Action",
33    "year": "2012",
34    "rated": "PG-13",
35    "plot": "Earth's mightiest heroes must come together and learn to fight as a team if they are to stop the mischievous Loki and his alien army from enslaving humanity."
36  },
37  {
```

Aktiviere im Server CORS.

Erstelle einen Vue Client mit Router und Vuetify.

In **App.vue** verwende die **v-app-bar**. Hole in **Home.vue** im Lifecycle Hook **created** die Daten vom Server und zeige sie zunächst so an, wie sie vom Server kommen.

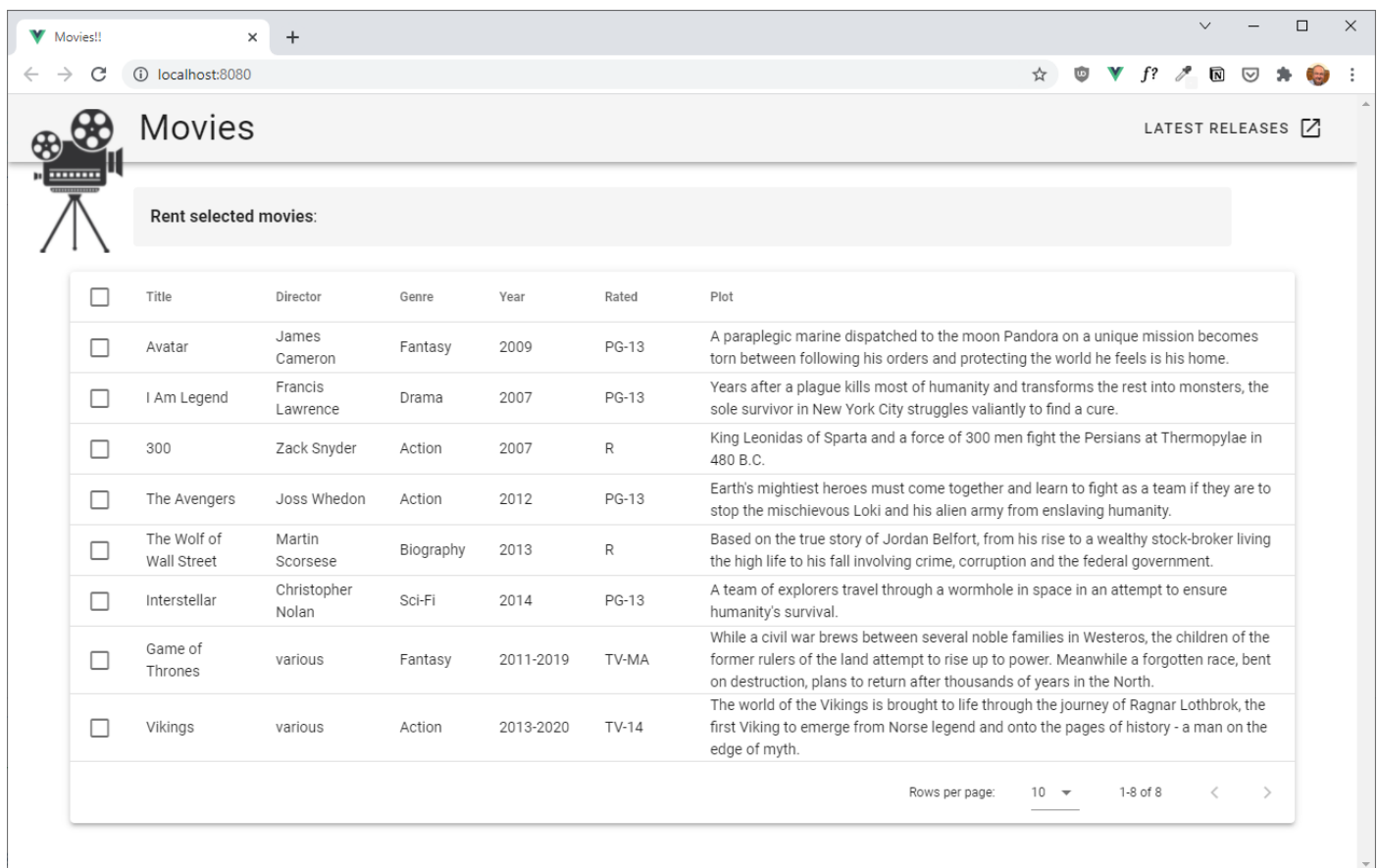
Robert Baumgartner



Sieht hässlich aus, aber funktioniert. Beachte, dass auch der Tab richtig benannt wurde.

Der Link rechts oben führt zu: <https://www.imdb.com/movies-in-theaters/>

Verwende dann `v-data-table`, um die Daten als Tabelle anzuzeigen. Lies dir dazu die Vuetify Dokumentation durch. Verwende `v-alert` für das Banner oberhalb der Tabelle.



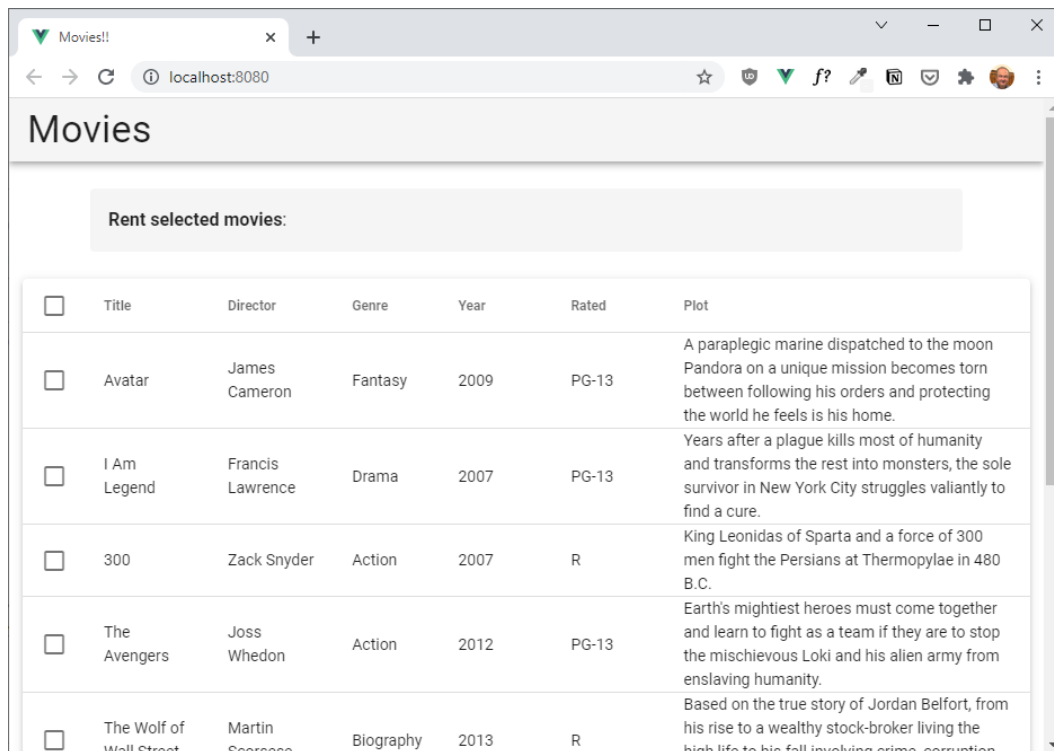
Setze eine Breite (`width`) von 100 für die Spalten `Year` und `Rated`. Beispiel:

```
{ text: 'Year', value: 'year', width: 100 },
```

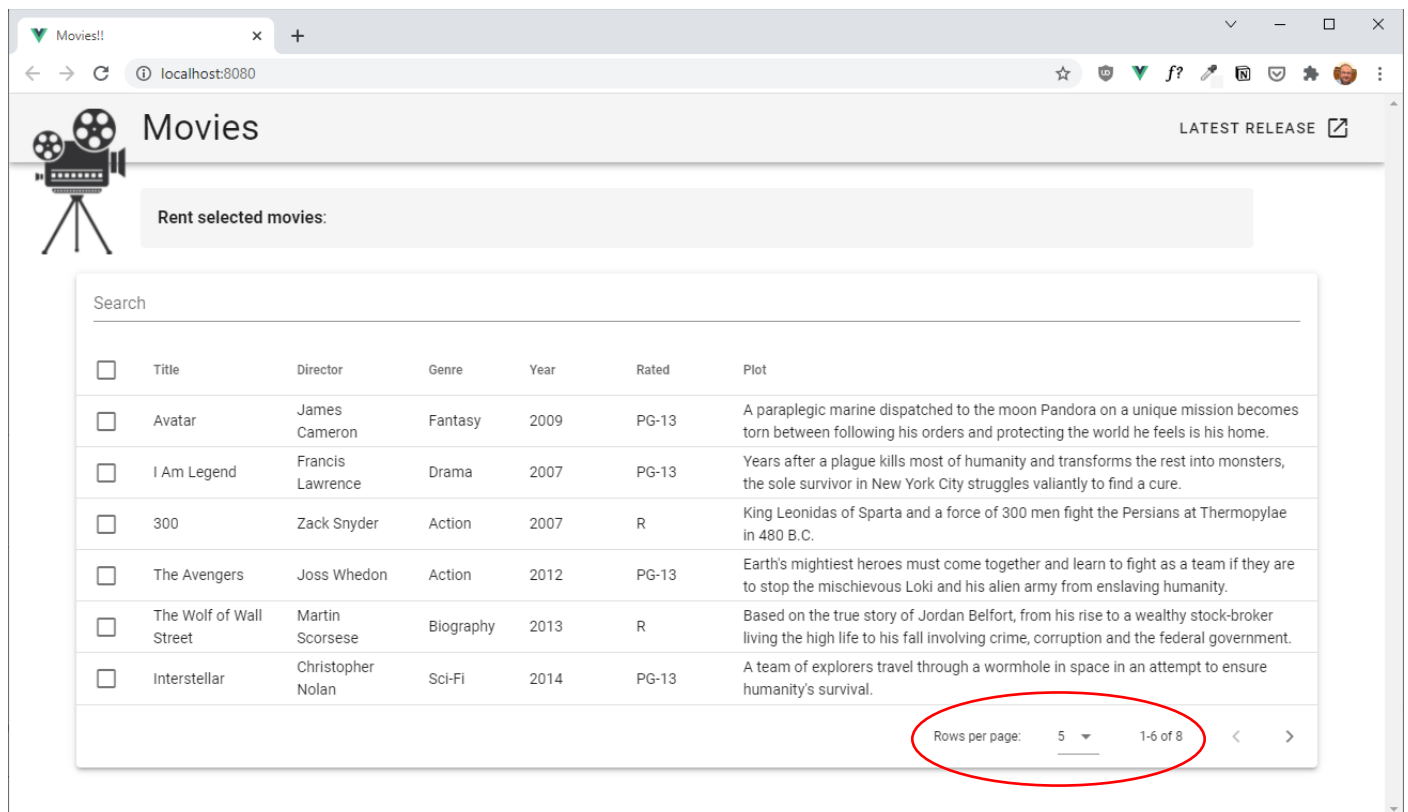
Robert Baumgartner

Schalte mit dem **v-data-table** Prop **show-select** die Möglichkeit ein, Zeilen auszuwählen.

Wird das Fenster verkleinert, so wird das Logo ab **md** nicht angezeigt.



Ändere mit dem **v-data-table** Prop **items-per-page** die Anzahl der Zeilen auf 6 pro Seite.



Aber nun stimmt der Footer nicht mehr! Dafür gibt es jedoch in **v-data-table** das Prop **footer-props**, mit dem du den Footer gestalten kannst.

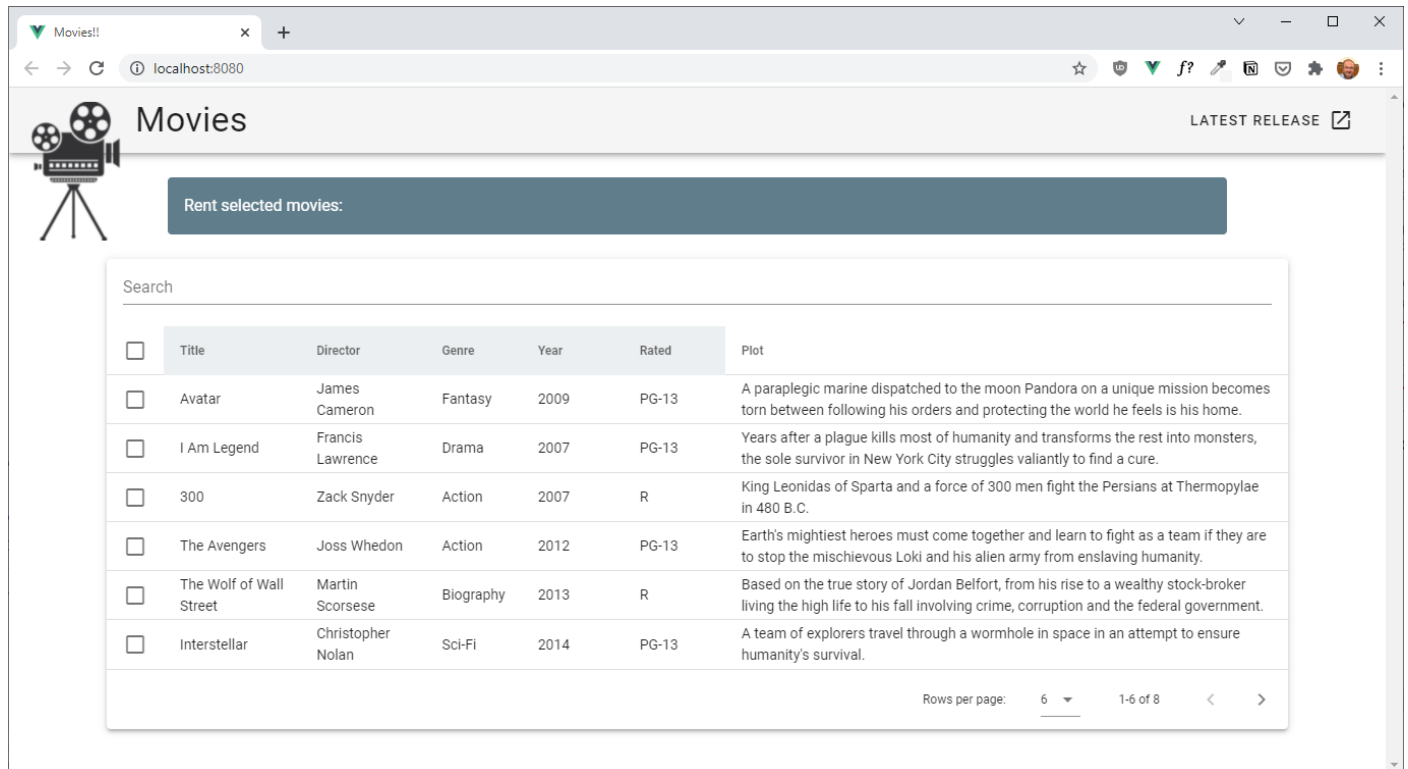
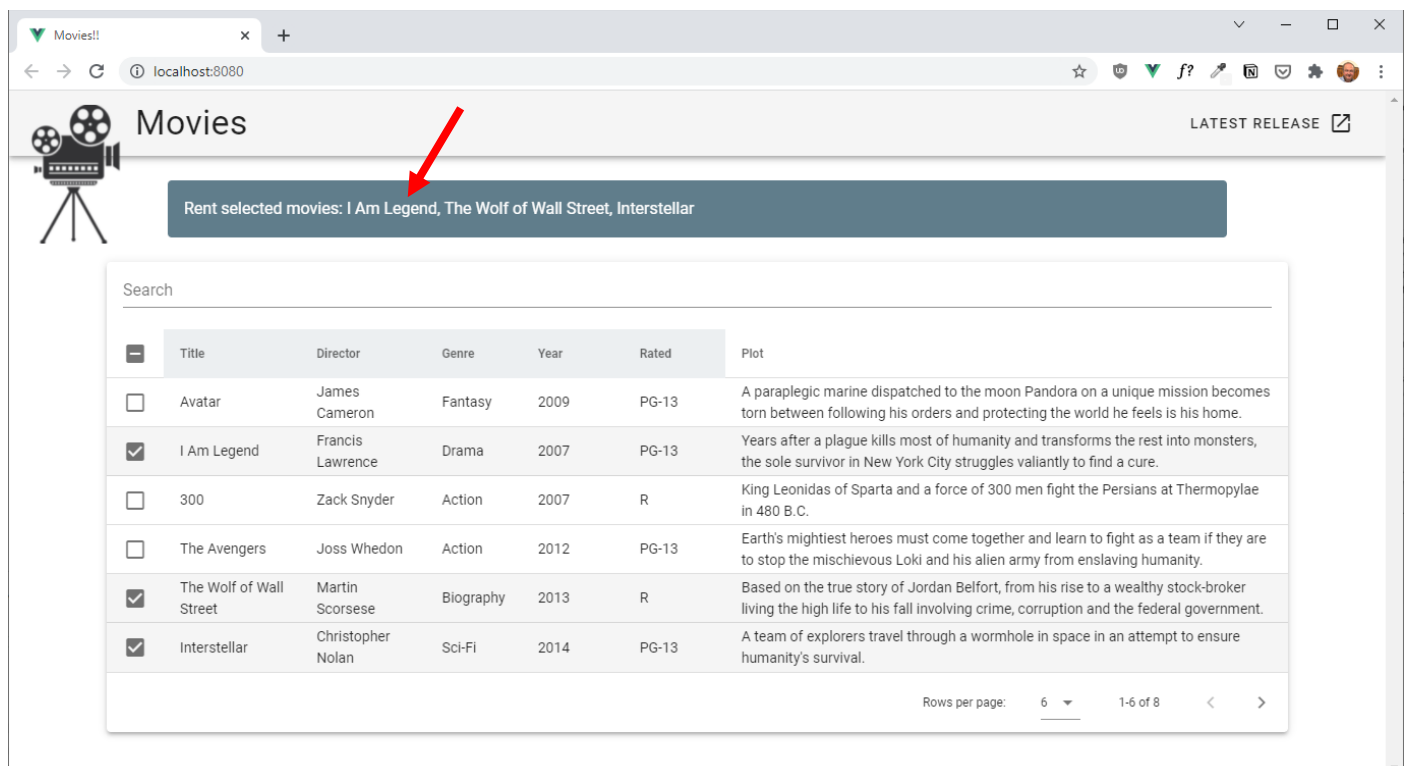
```
:footer-props= "{itemsPerPageOptions: [6,12,18]}"
```

Robert Baumgartner

Nun noch etwas Styling...

BTW: Die Header kannst du mit `class=""` einfärben. Beispiel:

```
{ text: 'Title', value: 'title', class: 'blue-grey lighten-5' },
```

Wenn der Benutzer eine oder mehrere Zeilen auswählt, sollen die Titel der Filme im **v-alert** oben angezeigt werden:Wird ein Movie wieder deselektiert, soll die Liste der ausgewählten Movies in **v-alert** entsprechend geändert werden.

Robert Baumgartner

Dazu gibt es zwei Events in `v-data-table`: `item-select` und `toggle-select-all`.

Verwende die Vue Dev Tools im Browser, um herauszufinden wie du sie verwenden kannst, um das oben beschriebene Verhalten zu implementieren.