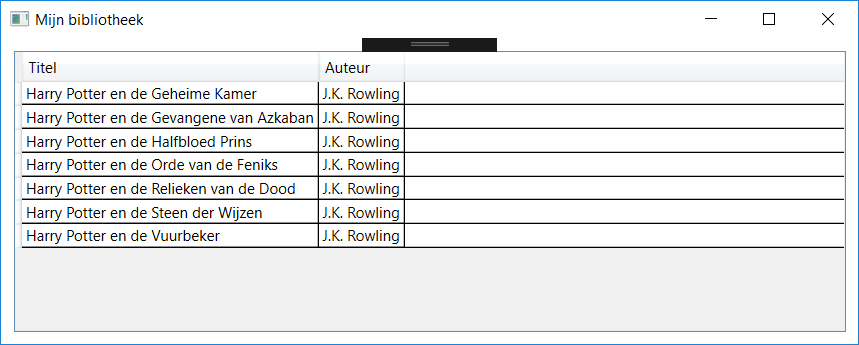
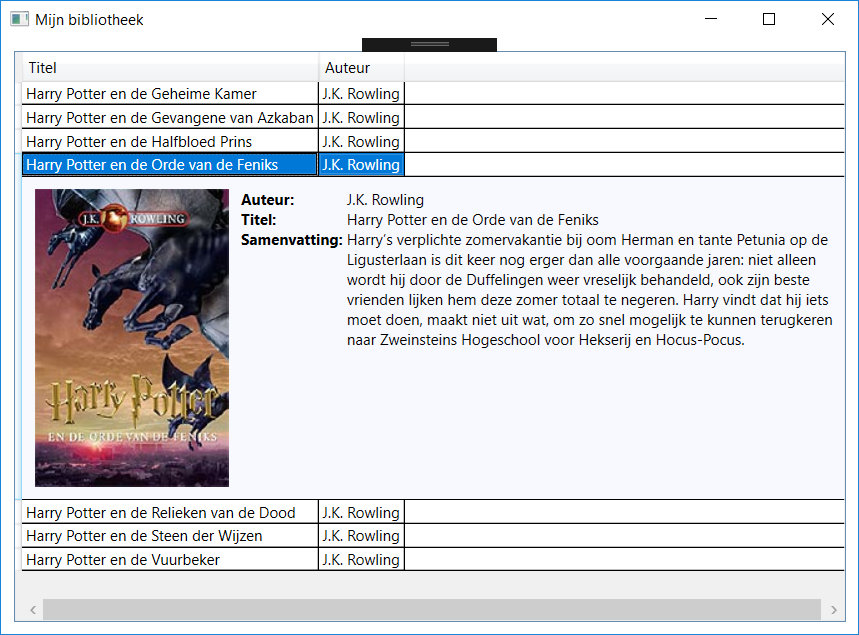
***MVVM Boeken***

***A. Opgave***

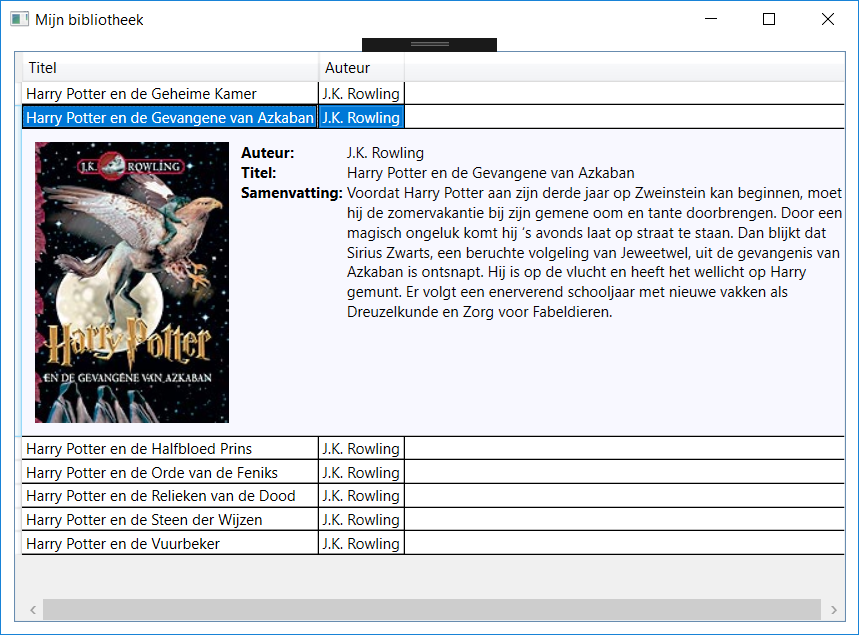
Doel van de oefening is het tonen van de verschillende boeken uit een txt-bestand. Bij opstarten krijg je:



Wanneer de gebruiker een boek selecteert, dan wordt van dat boek de cover en de samenvatting getoond. Je krijgt dan bijvoorbeeld:



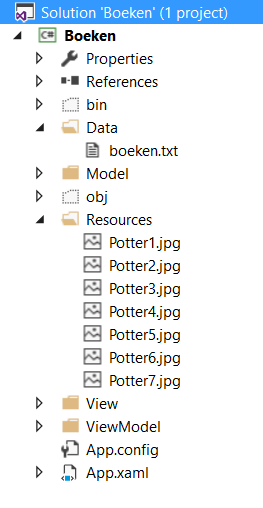
Of:



***B. Uitwerking***

***1. Creatie van het project***

Maak een nieuwe WPF-applicatie *Boeken* aan. Verwijder *MainWindow*. Voorzie onderstaande structuur. De nodige bestanden vind je in de startbestanden.

Maak onderstaande folders aan:

* Data
* Resources
* View
* Model
* ViewModel

Voeg het bestand *boeken.txt* toe aan de folder *Data*. Elke record uit dit bestand is een boek en bevat de auteur, de titel, een image en een korte samenvatting van het boek.

Plaats de images in de *Resources*-map van het project.

Maak van zowel het txt-bestand als de images *Embedded Resources* en stel in dat ze *If newer* gekopieerd worden naar de outputdirectory.

Maak in de *View*-map een nieuwe *MainWindow* aan. Je kan alvast een gedeelte van de *Xaml*-code uit de startbestanden halen.

***2. MVVM-designpattern***

Je werkt via het MVVM-designpattern.

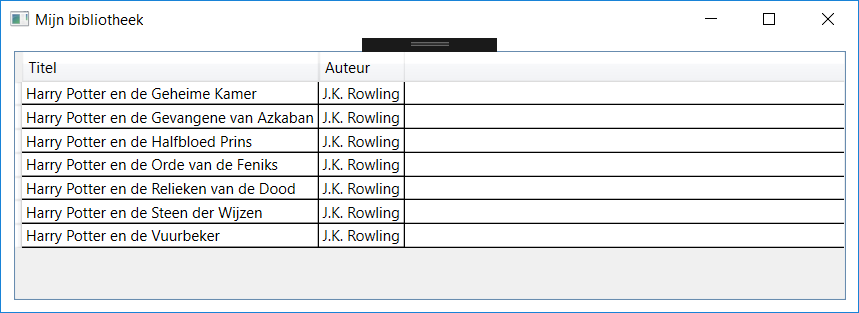
Dit betekent dat je start met het uitwerken van de klasse *Boek* (kopieer ook de klasse *BaseModel* uit het derde voorbeeld) in de folder *Model*.

Vervolgens ga je de klasse *BoekenViewModel* uitwerken (kopieer ook *BaseViewModel* en *BaseCommand* uit het derde voorbeeld). Deze klasse genereert een collectie van *Boeken* die vervolgens via databinding kan gekoppeld worden aan de view. De collectie boeken wordt opgehaald uit het txt-bestand. In tegenstelling tot bij Windows-applicaties moet je telkens je een bestand via een *StreamReader* wil aanspreken in de outputdirectory het volledige pad naar dat bestand opgeven. Onderstaande code kan je gebruiken om *dynamisch* te verwijzen naar de bin-debug directory:

AppDomain.CurrentDomain.BaseDirectory + "/Data/boeken.txt"

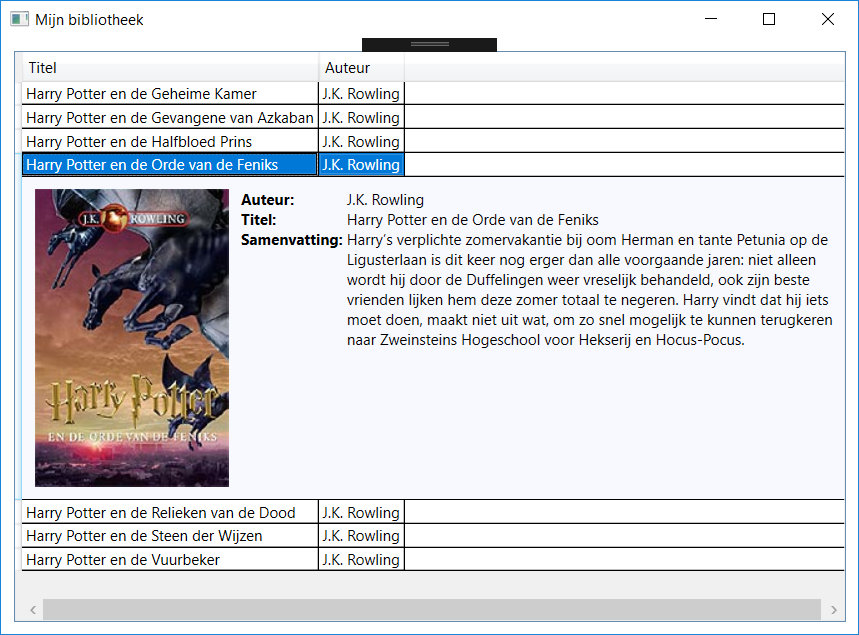
***3. DataBinding***

Stel nu in *MainWindow* de *DataBinding* in (zoek op het internet hoe je de *DataBinding* doet met een *DataGridTextColumn*). Je krijgt je eerste resultaat



***4. Coderen van de RowDetailsTemplate***

Codeer nu de *RowDetailsTemplate*. Deze *template* wordt door de *DataGrid* automatisch getoond wanneer je een rij selecteert. Wanneer de gebruiker een boek in de *DataGrid* selecteert, dan wordt van dat boek de cover en de samenvatting getoond. Je krijgt dan:



***5. Extra***

Codeer dat de data niet alleen uitgelezen worden maar dat deze ook kunnen aangepast worden. Bij sluiten van de applicatie schrijf je het aangepaste bestand weg naar bin/debug. Je overschrijft het bestaande txt-bestand.