

VueJS Miss Book



Build a vue application that manages a **book** entity with the following structure:

```
{
  "id": "GXj93K0kqZoC",
  "title": "Hacking",
  "subtitle": "Digital Media and Technological Determinism",
  "authors": [
    "Tim Jordan",
    "Puki Ben David"
  ],
  "publishedDate": 2008,
  "description": "Hacking provides an introduction to the community of hackers and an analysis of the meaning of hacking in twenty-first century societies.",
  "pageCount": 160,
  "categories": [
    "Computers",
    "Hack"
  ],
  "thumbnail": " http://coding-academy.org/books-photos/1.jpg",
  "language": "en",
  "listPrice": {
    "amount": 19,
    "currencyCode": "ILS",
    "isOnSale": true
  }
}
```



First you will need to learn about computed-properties, see [here](#)

You only need to read the basic example.

Build the BookService with a query function, that returns an array of books
(copy from books.json and paste to the service file – do not use AJAX yet)

Build the following components:

<book-app>

- Has a *books* data - an array of books, on created – get the books from the service.
- Has a *filterBy* object – null by default
- Has a computed property *booksToShow* that returns books based on the current filter

renders:

```
<book-filter @filtered="setFilter"></book-filter>
<book-list :books="booksToShow" @selected="selectBook"></book-list>
<book-details :book="selectedBook"></book-details>
```

<book-preview>

- Props: *book*
- Displays the book title and price, use computed prop for currency icon

<book-list>

- Props: *books*, Emits: *selected*
- Render a list of *<book-preview>* passing down a *book* prop
Show the books as cards (side by side)
- When a book is clicked – emit the selected book id (only one book can be selected)
NOTE: you will have to use **click.native** in this case

<book-details>

- Props: *book*
- Show entire details
- Use computed: based on *pageCount*, display also the text:
 - > 500 – Long reading
 - > 200 – Decent Reading
 - < 100 – Light Reading
- Use computed: based on *publishedDate*, display also the text:
 - > 10 years ago – Veteran Book
 - < 1 Year – New!
- Use class binding: based on price, show in color:
 - > 150 – Red
 - < 20 – Green
- If the book is on sale – show a nice SALE sign
- Show only 100 characters of the description and add read more/less that toggles the rest of the text
TIP: build a component that can be used like this: `<long-text v-bind:txt="book.desc">`



<book-filter>

- Allow the user to filter the books list by name and by price range
- Emits: `filtered` with a `filterBy` object: `{byName: 'xx', fromPrice:0, toPrice:Infinity}`
- Emits the current filter when Filter button is clicked