

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
  }
}
```



Build the BookService with a query function, that returns an array of books, for now, generate an array of 20 books based on this object with modifications such as:

- Random (string) id
- Title – capitalized lorem (1-3 words 3-5 length)
- Price (10-500)
- Published date (pick one from: 1978, 1999, 2011, 2018)
- isOnSale – true/false
- Random language from (en, he, sp)
- Random image (1-20.jpg)
- Random author from:
'Agatha Christie', 'William Shakespeare', 'Barbara Cartland', 'Danielle Steel', 'J. K. Rowling', 'Dr. Seuss', 'Leo Tolstoy', 'Jin Yong'

To make the data stable, save it in localStorage, from now on only regenerate if not in localStorage.

Build the following components:

`<book-app>`

- Has a *books* data - an array of books
- Has a *filter* 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-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-preview>`

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

`<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

<book-filter>

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