

VueJS Miss Book
STEP 3
Route your App and Get Some Real Books!



Add the following pages to your app:

1. **homepage** - nice and welcoming homepage to your app.
 2. **about** - a page about yourself and the application (nice picture and few lines about you and about the application for example framework you used to build the app.).
 3. **book** – the previously built component with list of books and the filter
 4. **book/bookId**
 - a. Reuse the BookDetails component
 - b. Get bookId from the routing, use `BookService.getBookById(bookId)` that returns a promise to this book.
 5. **add-book** - `<BookAdd>` - Support adding a book to our database from google books api, see details below
-
- **NavBar** - All the pages should contain a navbar with links to the other pages.
 - In this nav bar also show how many books system has
 - Show success/error messages for user actions

Build the component: <BookAdd>

Find and Add new Books:

Show an input for search (use google API, here is a [sample call](#))

Important: to avoid the hassle of creating an API key, issue a sample call, save the results locally for the development time, and then when ready switch back to the real API.

Show the options arriving from the API:

×

- The Three Yugoslavias ⊕
- The Three U.S.-Mexico Border Wars ⊕
- The Three Musketeers ⊕
- The Three Gifts ⊕
- The Three Bears ⊕
- Alexandre Dumas' the Three Musketeers ⊕

When + is clicked, add the book to our database using

`BookService.addGoogleBook(googleBook)`

This function should convert the google book to our format and store in our books entry in the localStorage and return a resolved promise.

It also shows a success message: **Book XXX was successfully added**, with a link allowing us to navigate to this book.

- Improve the user-msg component, to also accept a link to render in a router-link inside the msg :

```
<router-link :to=msg.link>Check it Out</router-link>
```