# VueJS Miss *Book*

## STEP 2

## Continue your application

- first - move to using only local components
- Make your BookService async!
- Add the UtilService and use it to save and store from localStorage
- To get autocompletion in JS files, you need to add this to your javascript settings.json: *"emmet.triggerExpansionOnTab": true*
  How to get there?
  - Ctrl+Shift+P, search for 'language'
  - Select: 'Configure Language Settings'
  - search for 'javascript'
  - add that line

Add the following pages to your app:

1. **/** - nice and welcoming homepage to your app.
2. **/about** - a simple page (nice picture and few lines of text).
   a. Bonus: This component should set an interval when created, and clear the interval when destroyed (It can just print something to the log)
   b. Bonus: Learn about Vue animations and use some animations in the page
3. **/book** – the previously built component - BookApp 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 this book

## New components

### <app-header>

- Page title and <nav> with links to the app pages, place it in the main

### <review-add>

- Rendered Inside the BookDetails component.
- Contains a <form> to allow the user to provide a review for the book:
    - Their full name (default: 'Books Reader')
        - Use Vue Ref to make this input focused when form appears
    - Rate: 1-5  - use a dropdown (Bonus: use stars)
    - Read at - datepicker (default now).
    - Textarea for free text
- Save the review: bookService.addReview(bookId, review)
- Show the list of reviews for the book
- Allow deleting a review (small x link for each review in the list)

### <user-msg>

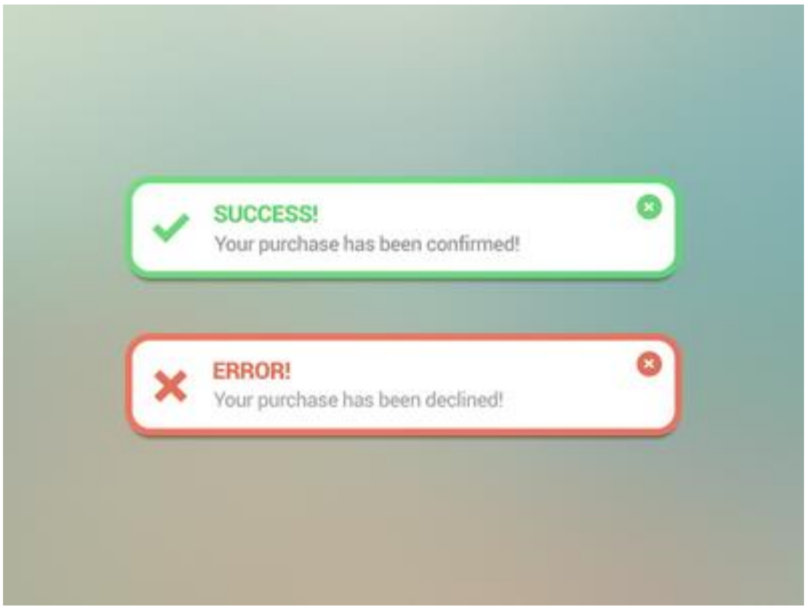
Whenever the user performs an action, we want to show a nice success/error message, to achieve this, we will use an event bug and $emit an event ('show-msg') whenever needed.

Build a simple component <user-msg/> and render it in your app (position absolute, top right).

This component

- is not seen by default

- listens ($on) on the eventBus for 'show-msg' events – when this event is fired (For example: {txt: 'Pet was successfully saved!', type:'success'}) show the message.
- Bonus: hide the msg after 3 seconds, or when use clicks the x button



SUCCESS!
Your purchase has been confirmed!

ERROR!
Your purchase has been declined!

When a book review is added, show the msg: 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:

```
<p>{{ msg.txt }} </p>
<router-link :to="msg.link">Check it Out</router-link>
```

### <book-app>

As is

### <book-filter>

As is

### <book-preview>

As is