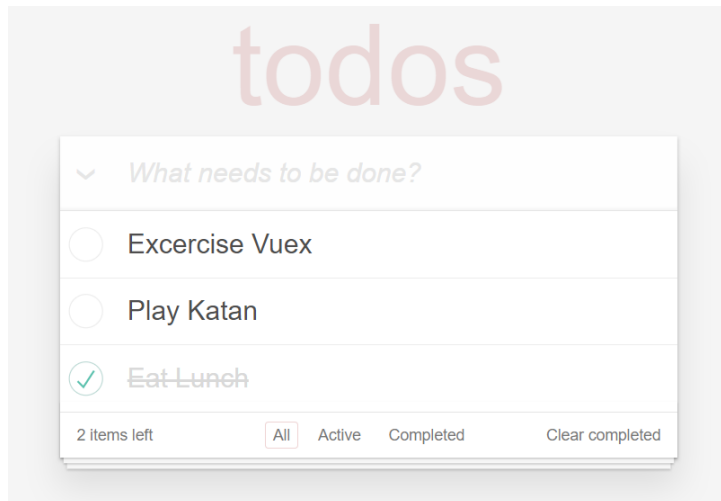




Todo, Todo



Build a [todo-app](#) application with that uses a store to manage the state

Here, we will use this terminology:

- Dumb – a component that do not dispatch to the store, usually also get the data from parent by props
- Smart – a component that dispatches to the store

Build the app from the following components:

- TodoApp (Routable, Smart component)
 - TodoList (Dumb)
 - TodoFilter (Dumb)
- TodoPreview (Dumb)
- TodoEdit (Routable, Smart)
- TodoDetails (Routable, Smart)

State Management

1. Store should manage the following state:
 - a. List of todos
 - b. isLoading
 - c. Current filterBy
 - d. User object
2. use a todoService from the components and commit to the store
 - a. The service uses the localStorage to store and load data
3. Implement the following features:
 - Add todo, Delete todo
 - Filter: All | Active | Done,
Add search functionality
 - Build an `<AppHeader>` component and render it at the top, show the name of the user and a todos progress-bar (percent of done todos)

userService

Export the following API:

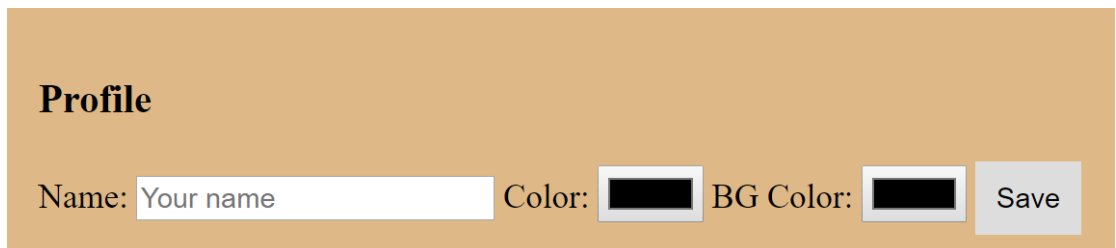
```
export const userService = {  
  login,  
  logout,  
  signup,  
  getLoggedInUser,  
  addActivity  
}
```

The service manages a user entity:

```
{  
  
  username: 'muki',  
  password: 'muki1',  
  fullname: 'Muki Ja',  
  balance: 10000,  
  activities: [{txt: 'Added a Todo', at: 1523873242735}]  
}
```

<UserProfile>

- Add a second page to your app: <user-profile> with an option to edit the user fullname and set preferences.



The image shows a user profile form with a light orange background. At the top, the word "Profile" is written in bold. Below it, there are three input fields: "Name:" followed by a text box containing "Your name", "Color:" followed by a black color picker, and "BG Color:" followed by a black color picker. To the right of these fields is a "Save" button.

- Also, render the user activities list, example
 - 2 minutes ago: Added a Todo: 'Wash the dishes'
 - Couple of hours ago: Removed the Todo: 'Talk to grandma'
- Add to user object: `prefs: {color: 'black', bgColor: 'white'}`

More features

- Show a loading indication when todos are being loaded
- Show no "no todos to show..", when there aren't any todos.

- Show Success and Failure messages in the component
- Implement filtering, sorting and paging in the todoService
- Show a progress-bar of todos completion at the header and at the footer of the app
- When filtering by txt, support debouncing
- Make it look good and responsive
- Push it to github-pages

PART 2

- Split your store to 2 modules: userStore and todoStore
- Create the needed actions to support the asynchronous calls
- Complete the app

Bonus

- Build a backend to your app
- TIP: copy and modify the backend from the carProj / missBug)
- Push the app to a cloud hosting (render.com)