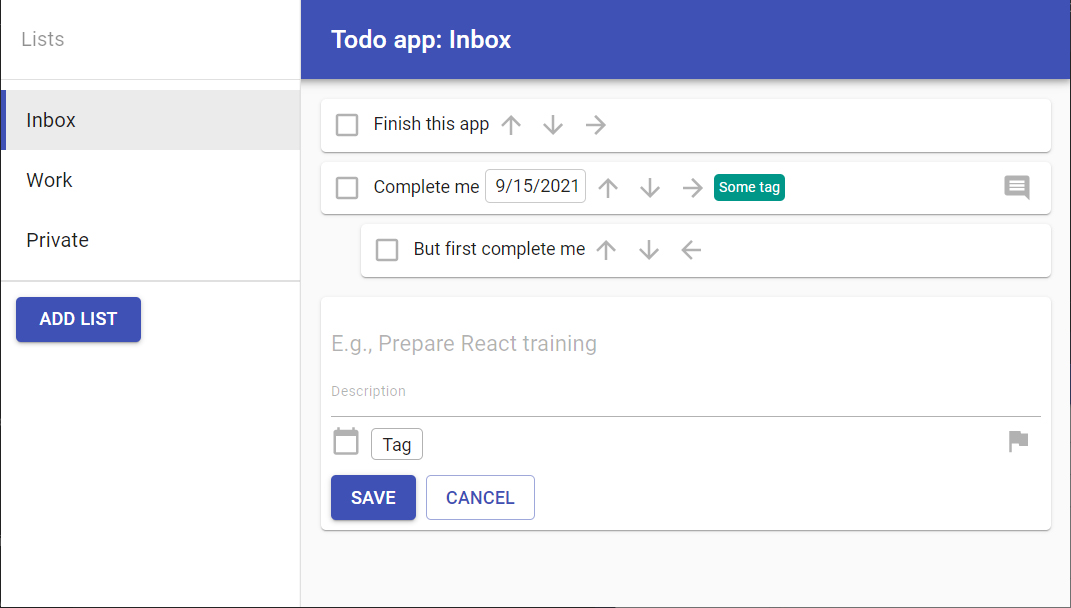
React case

# Case: Create a todo application

To get some practice using React we’ll make a todo app.

The goal is to create something that looks like this:



You have been provided with a starter application that contains some data.

## Lists

The app has multiple lists to put todos in. The default list is called ‘Inbox’. This list cannot be deleted and is always the top list. I should be able to create as many lists for todos as I want. For example, I want a ‘Work’ and ‘Personal’ list to keep things separate.

A list has the following properties (an interface IList has already been provided in the starter project)

* Id: string
* Name: string
* Color: string
* Order: number
* Default: boolean

## Todos

A todo is a task that I want to complete. Todos are always in a list, and a list can have 0 or more todos.

A todo has the following properties (an interface ITodo has already been provided in the starter project)

* Id: string
* List id: string (default is the ‘inbox’ list)
* Parent todo id (optional): string
* Title: string
* Description: string
* Due date (optional): date
* Priority: enum
* Tags (0 or more): ITag’s
* Complete: boolean
* Completed\_on (optional): date

A todo can be nested under another todo. It can be marked complete easily. It has a due date which is optional. If there is a due date, and it is in the past, the text of the todo should become red.

### Tags

Note that a todo can have tags, this entity has not been provided in the started files. You should create the model for tags and add tags to todos yourself.

A tag has the following properties:

* Id: string
* Name: string
* Color: string

## Requirements

### Functional

**List**

* I should be able to create, read, update and delete a list
* I should be able to reorder my lists
* I should be able to activate a list, so I can see and work with its todos.
* The default list is called ‘Inbox’
  + This list is always on the top
  + This list cannot be deleted

**Todo**

* I should be able to create, read, update and delete a todo
* I should be able to add and remove tags to a todo
* I should be able to reorder todos using up and down arrow icons
* I should be able to nest a todo under the todo above it using a right arrow icon
* I should be able to unnest a todo using a left arrow icon

**Tags**

* I should be able to create, read, update and delete a tag

### Technical

* The data should be stored in and be fetched from a redux store, a
* All actions should be performed using redux
* Each component should have unit tests

## Extra credit

If you have the todo app working, and you still want more, you can pick and choose some ideas below to implement.

### Local storage

Store the data in local storage so I can close my browser window and when I reopen it, my todo’s are still around.

### Upcoming

Create a special list called ‘upcoming’. This list should contain all todos from all lists that have a due date, ordered by due date ascending, and grouped by date. Todos with a due date in the past should be in a special group called ‘overdue’.

### Comments

Add the functionality to leave comments on a todo. I should be able to reply to comments. A comment has the following properties:

* id: string
* todoId: string
* parentCommentId: string (optional)
* comment: string
* date: Date

### Search

I should be able to search all todo’s for a simple string. The result list should show the todo’s that match my search term. Each todo in the search results should show the title, tags, and list the todo is in. I should be able to complete or edit a todo from this list.

### Multi-user

Add a login option so multiple users can use the app. By default, the todos are private to the current user. Add the option to make a list shared, so that all users can see all todos in that list. Todos in shared lists should say which user added the todo. A todo can be assigned to a user. Create a special list called ‘Mine’ that shows me all todos that have been assigned to me. All users can edit or complete a shared todo. Comments should be multi-user also.

You can simply create a list of users with name/username/password to login, no authentication server is required.

### Drag and drop

Replace the re-ordering, nesting and unnesting of todo’s with drag and drop. I also should be able to move a todo to a different list using drag and drop.

import \* as React from 'react';

import {connect} from 'react-redux';

import {RootState} from '../reducers/root.reducers';

import {loadTodos} from '../actions/Todo.action'

import {ITodo} from '../../model/interfaces/ITodo'

export interface ITodoState {

todos: ITodo[]

}

export interface ITodoActions {

loadTodos(): void;

}

const mapStateToProps = (state: RootState) => {

todos: state.todos.todos

}

const mapDispatchToProps: ITodoActions = {

loadTodos

}

interface ITodoProps extends ITodoState, ITodoActions {

}

export const withTodos = () => (Component: React.ComponentType) => {

class TodoContainer extends React.PureComponent<ITodoProps> {

constructor(props: ITodoProps) {

super(props);

if (props.todos.length === 0) {

props.loadTodos()

}

}

public render() {

return <Component {...this.props}/>;

}

}

return connect(mapStateToProps, mapDispatchToProps)(TodoContainer)