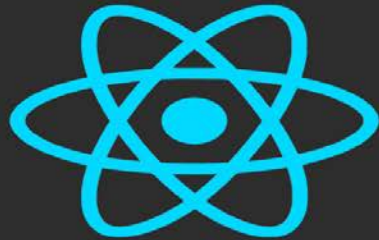


React Codelab Day 2



+



@Chalath.mo @Vattanan.bu

React Codelab Day 2

- Remind Letter
- Firebase Realtime Database
- Firebase Hosting
- Reference

Remind letter

To Do List

- 1.
- 2.
- 3.
- 4.
- 5.



add

Eating

edit

delete

Leaning

edit

delete

Playing

edit

delete

add

Eating

edit

delete

Learning

save

cancel

Playing

edit

delete

App

CreateTodo

add

Eating

edit

delete

Learning

save

cancel

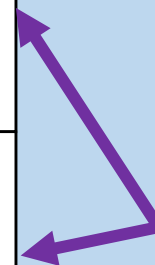
Playing

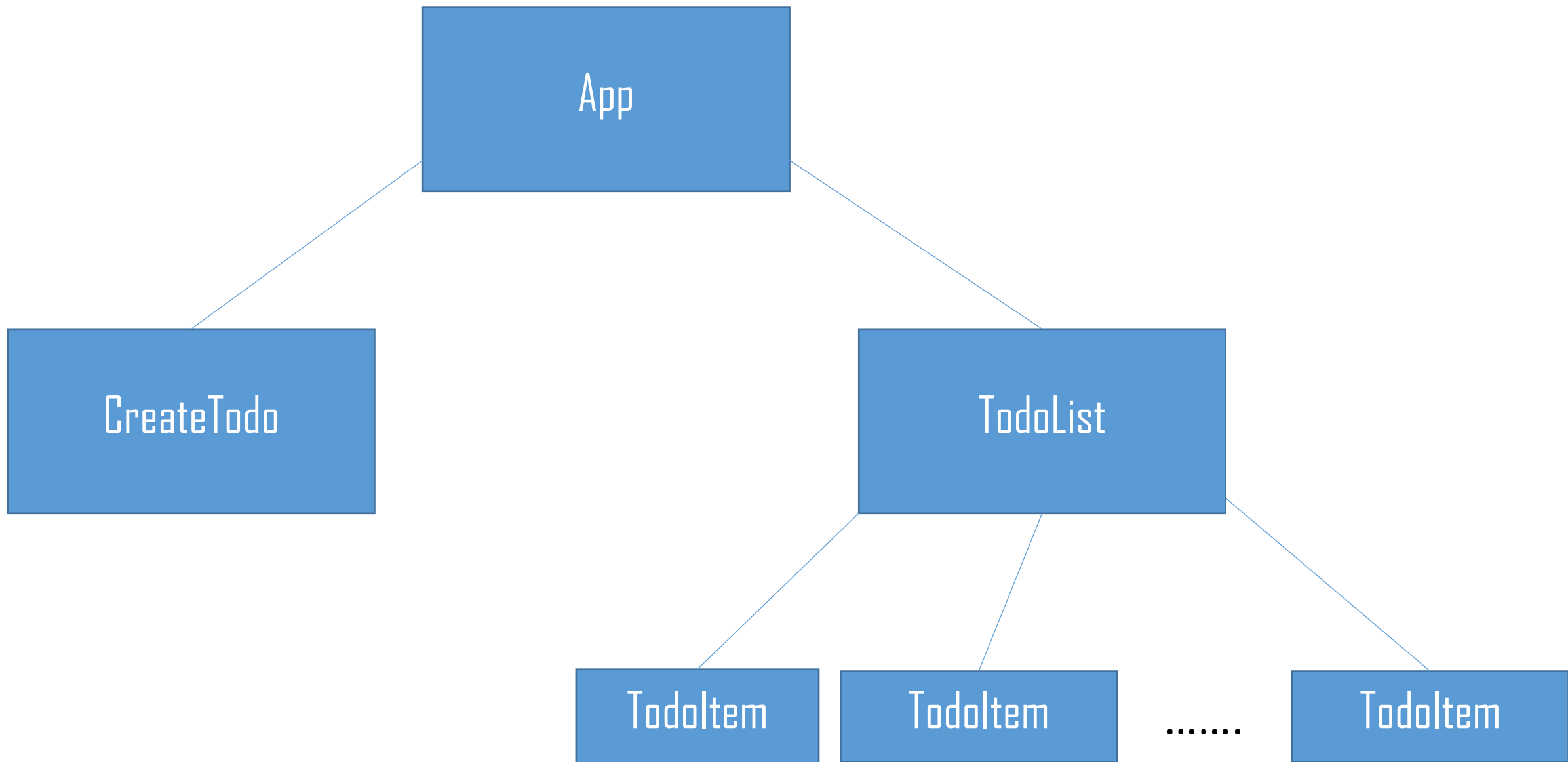
edit

delete

TodoList

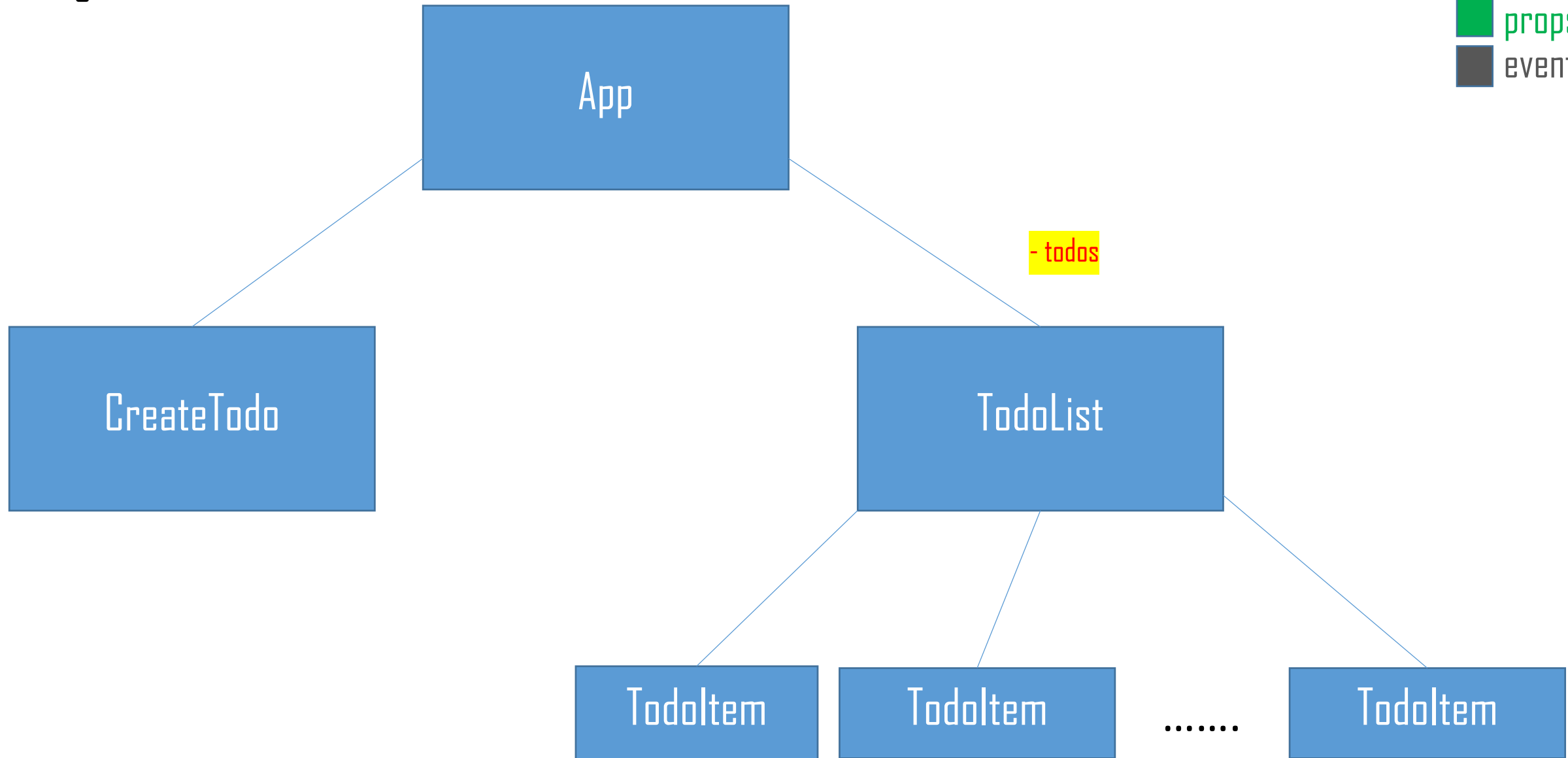
TodoItem





Manage Todo

state
props
event

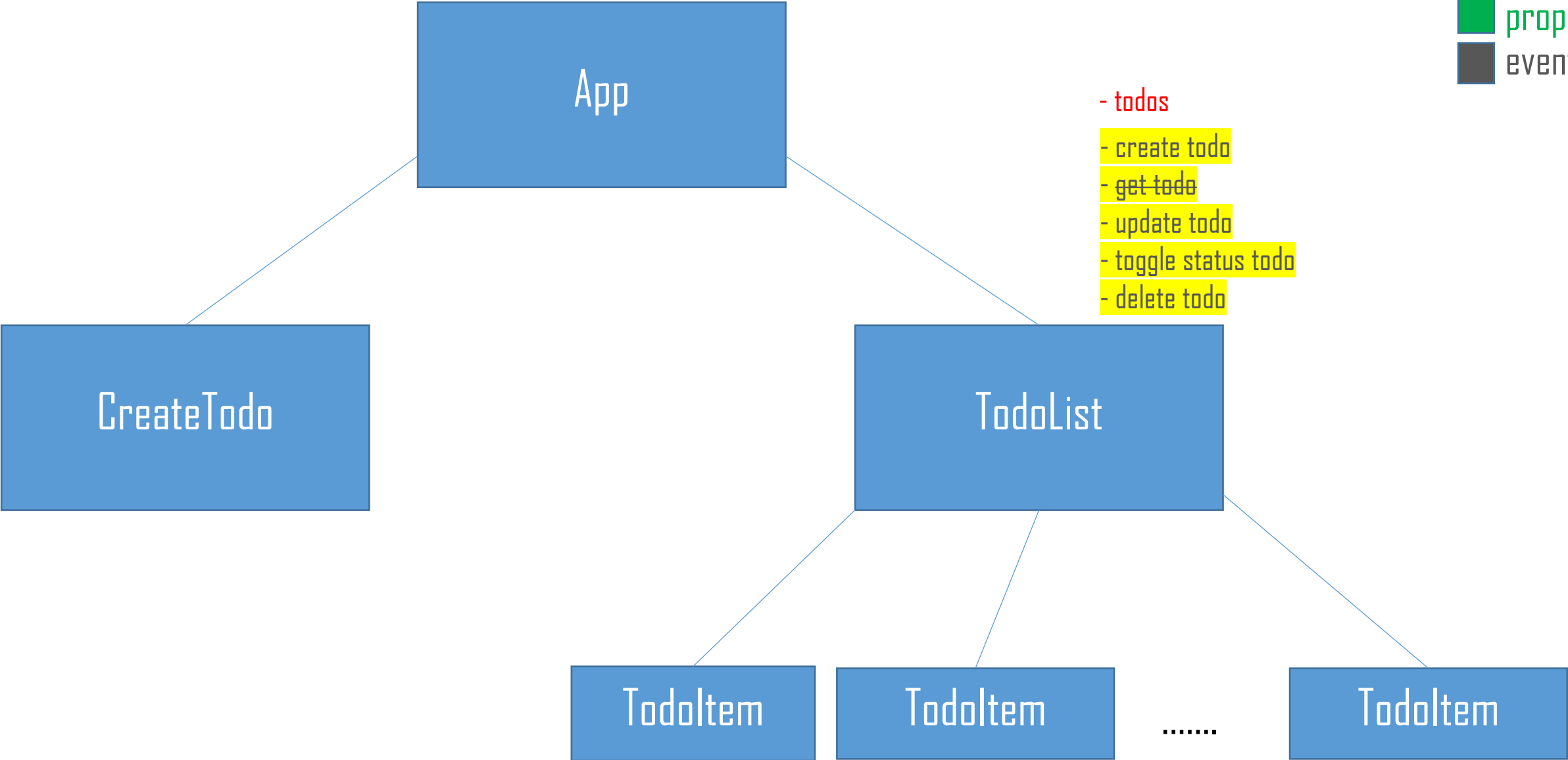


Manage Todo

state

props

event

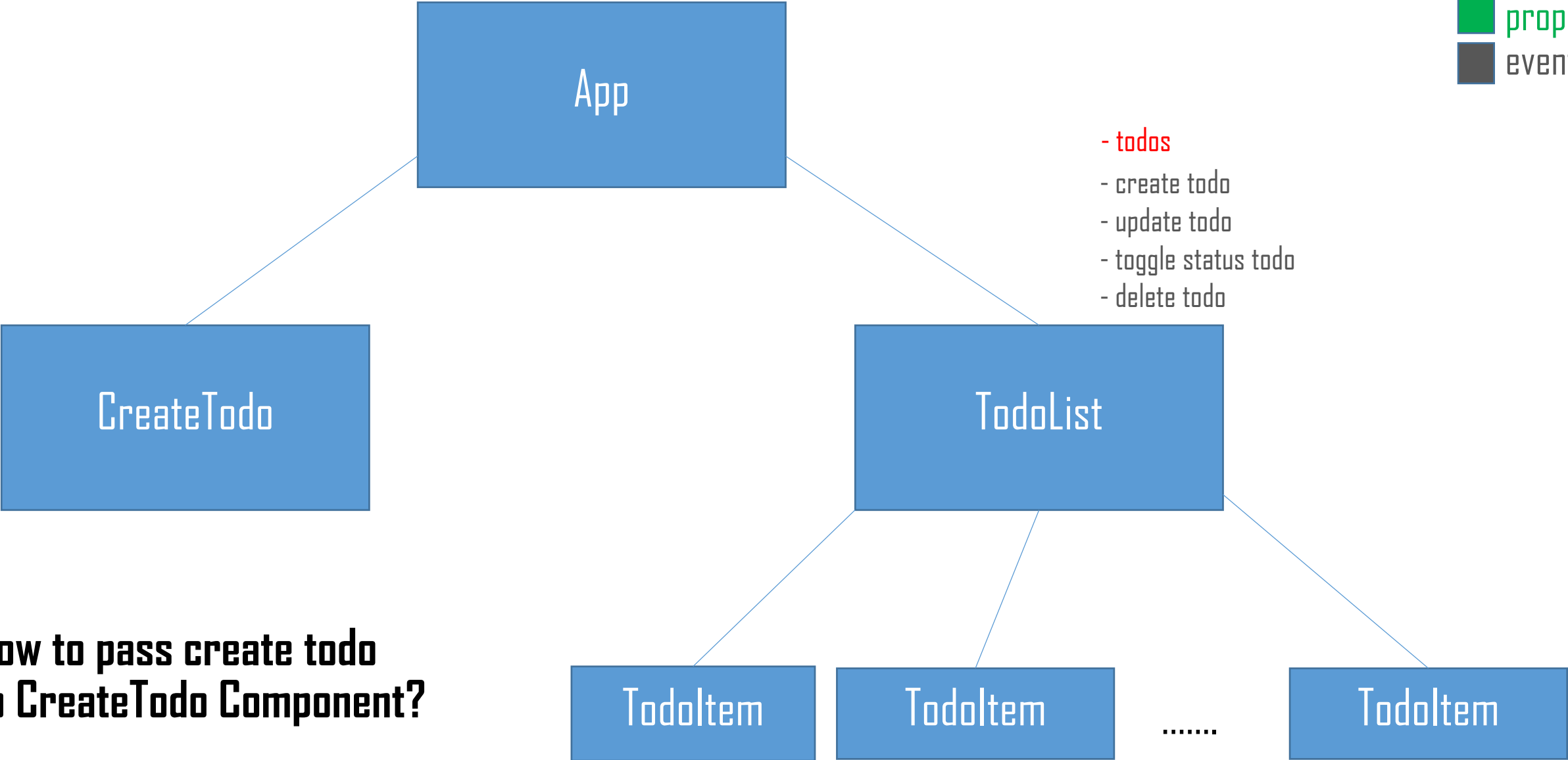


Manage Todo

state

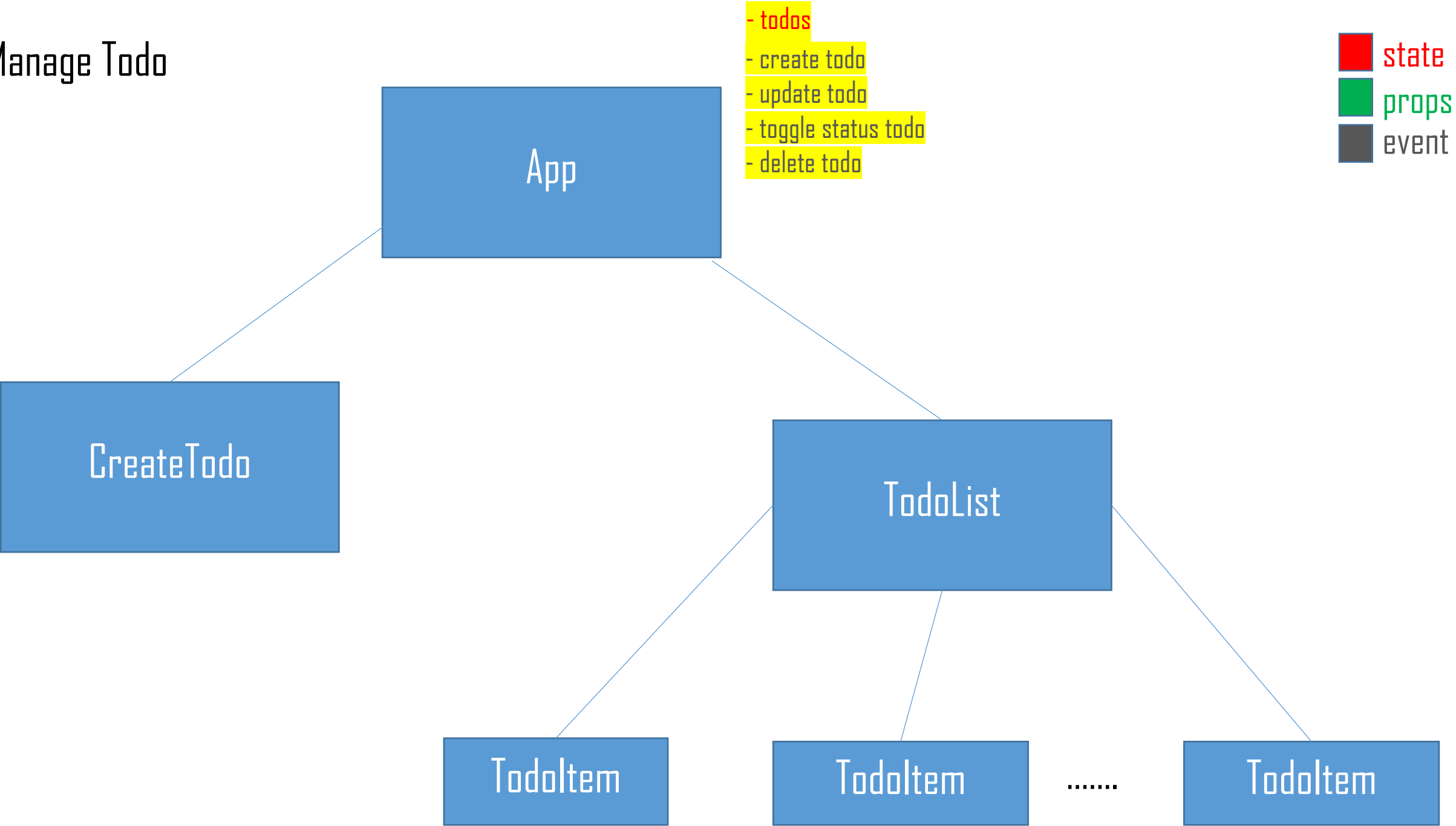
props

event

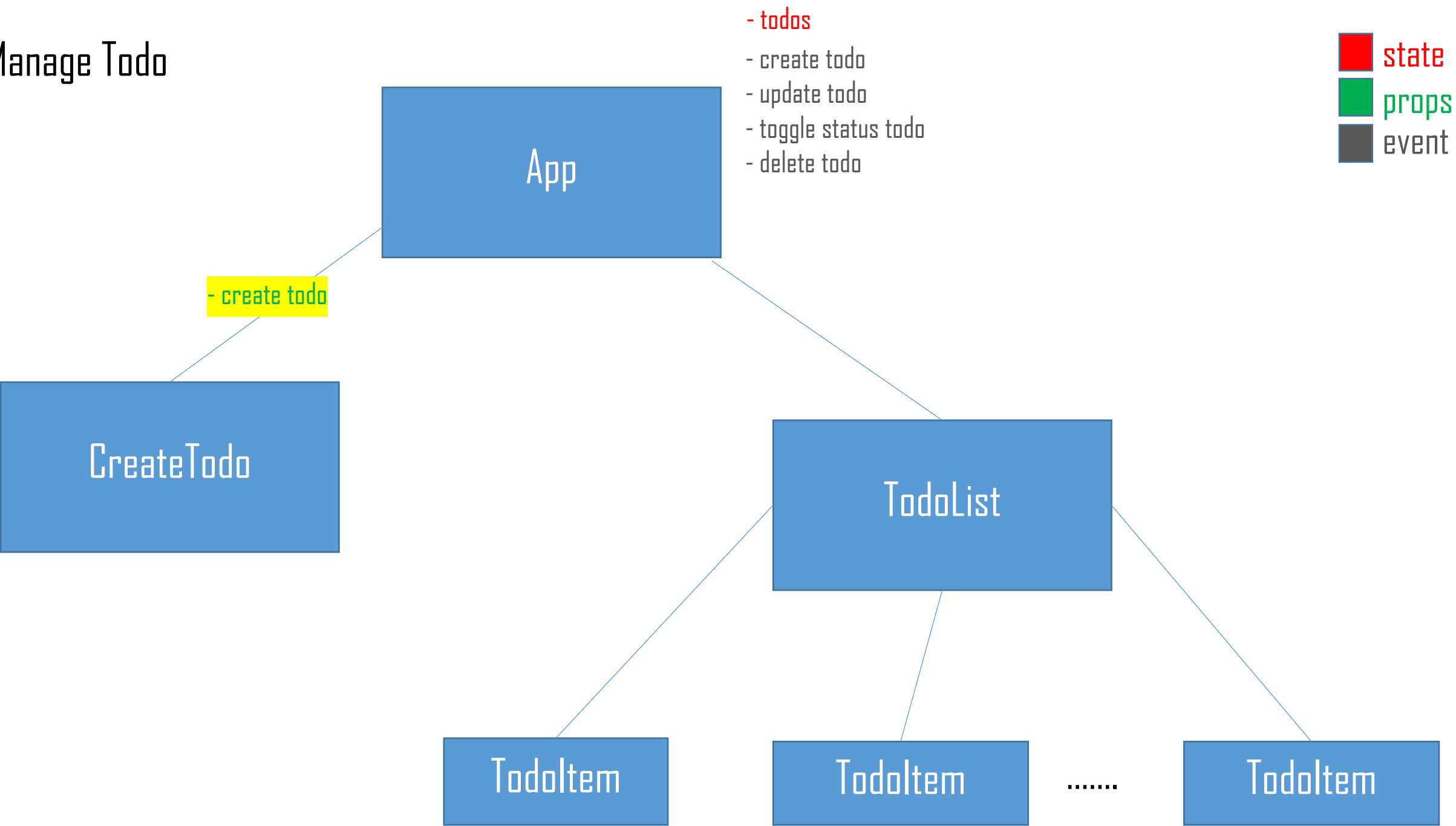


How to pass create todo
to CreateTodo Component?

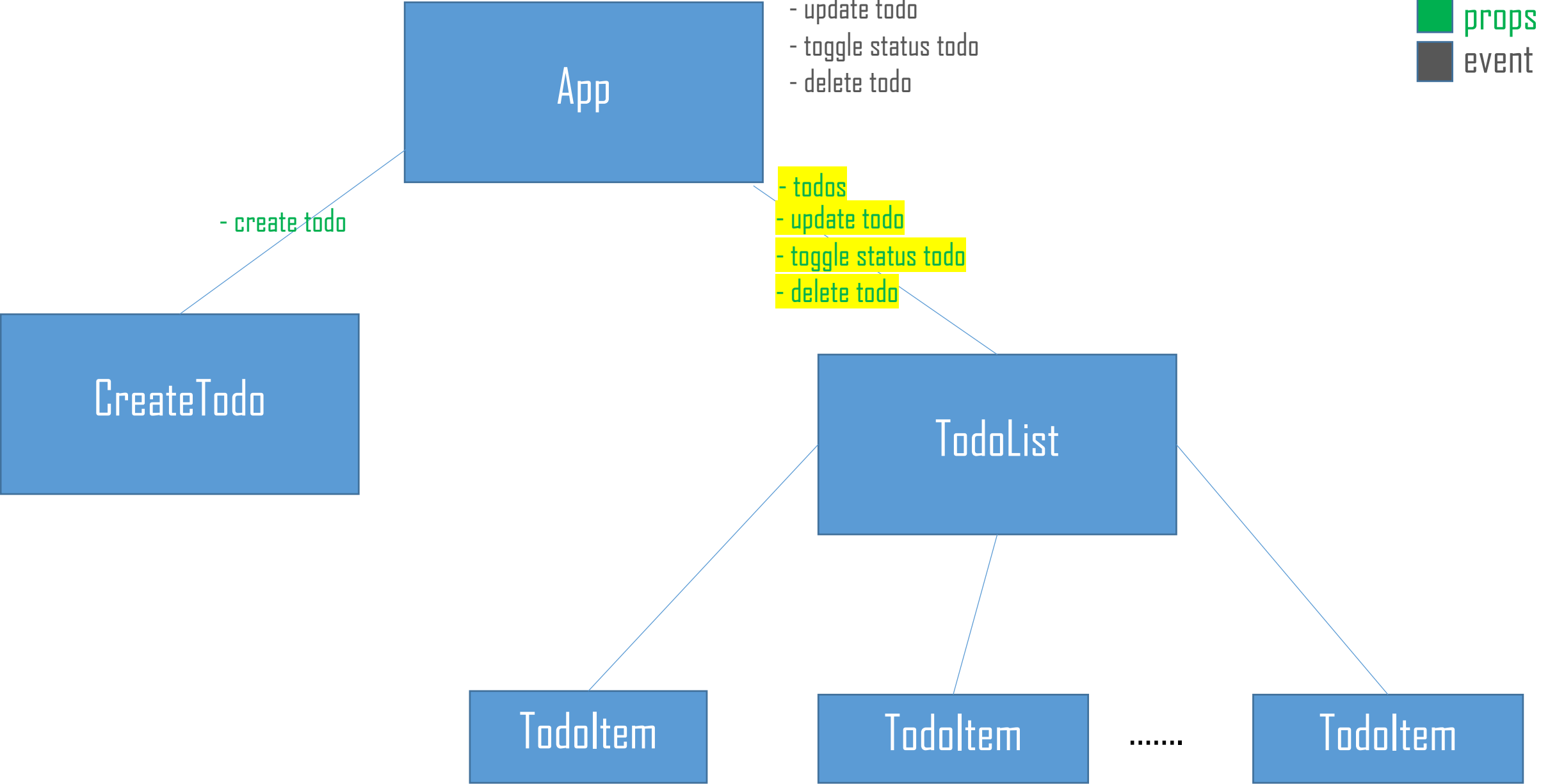
Manage Todo



Manage Todo



Manage Todo

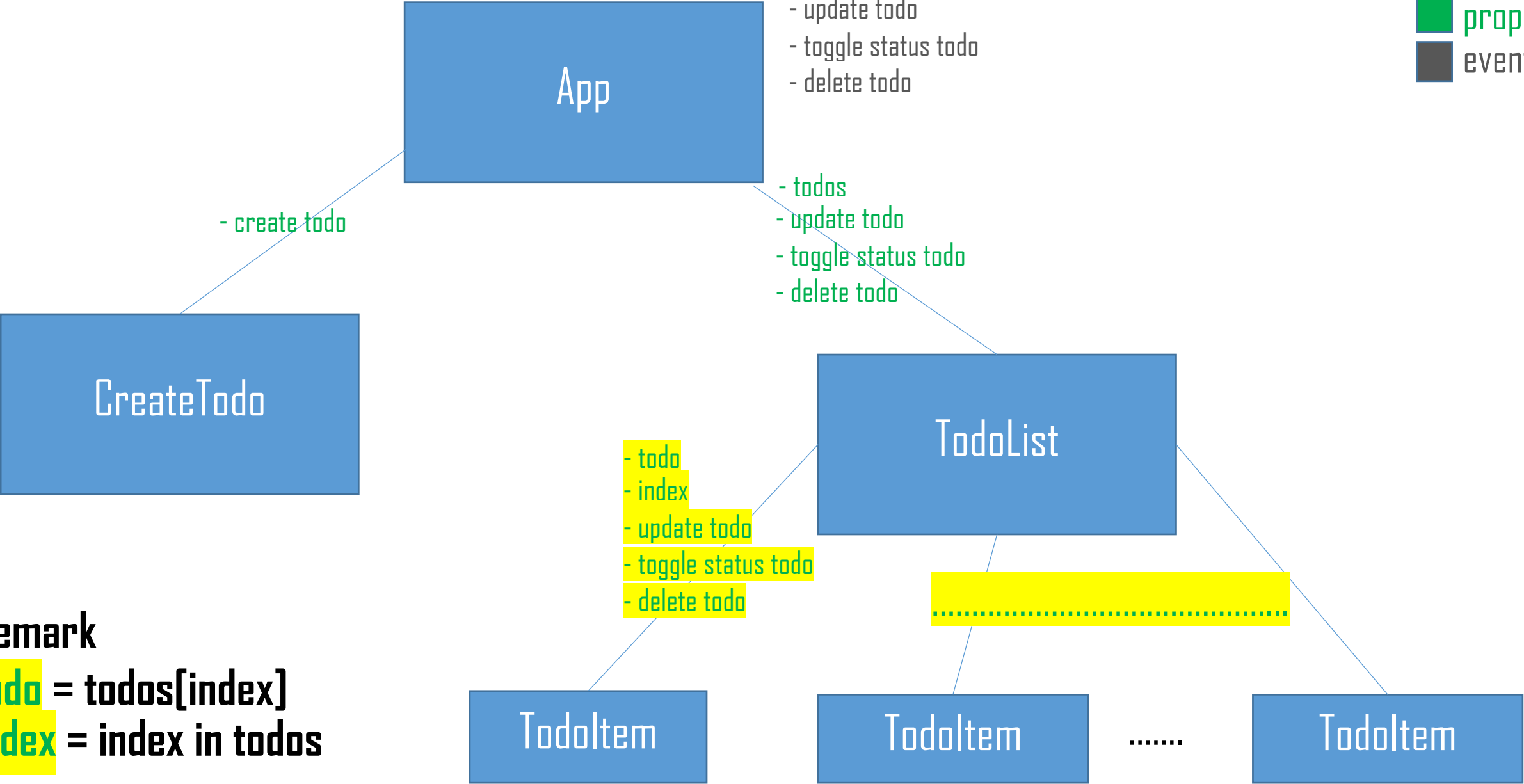


Manage Todo

state

props

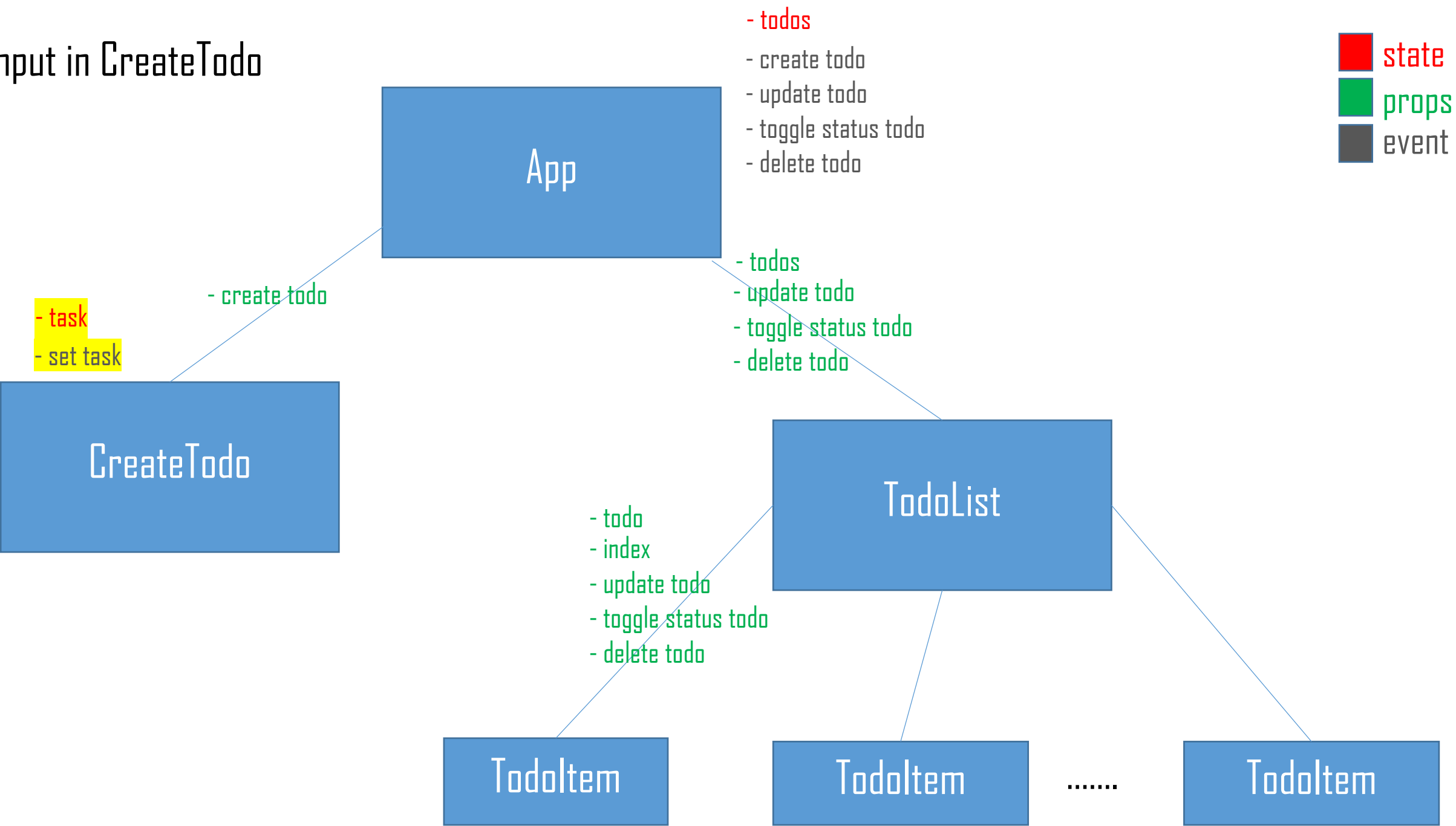
event



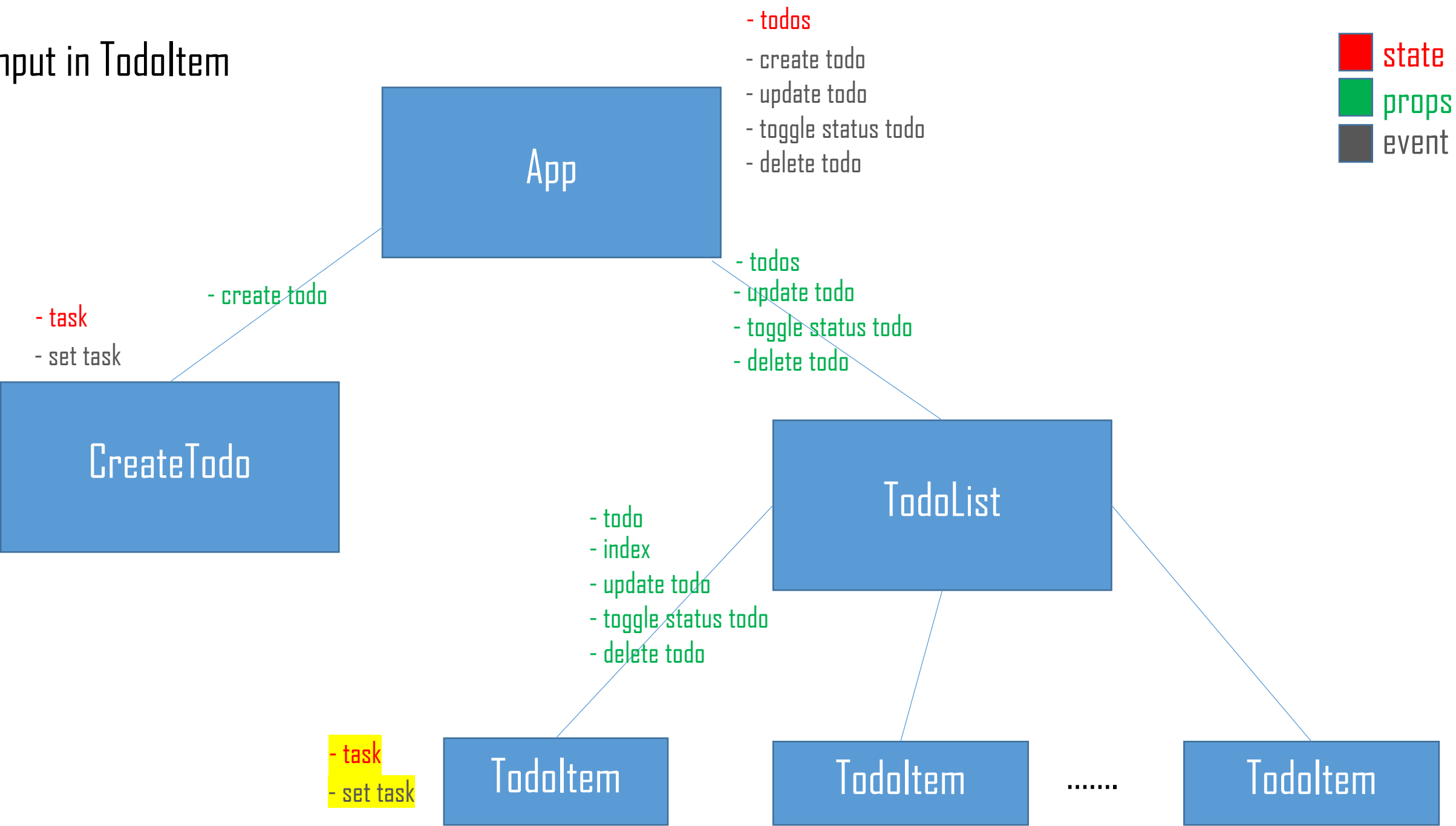
Remark

todo = todos[index]
index = index in todos

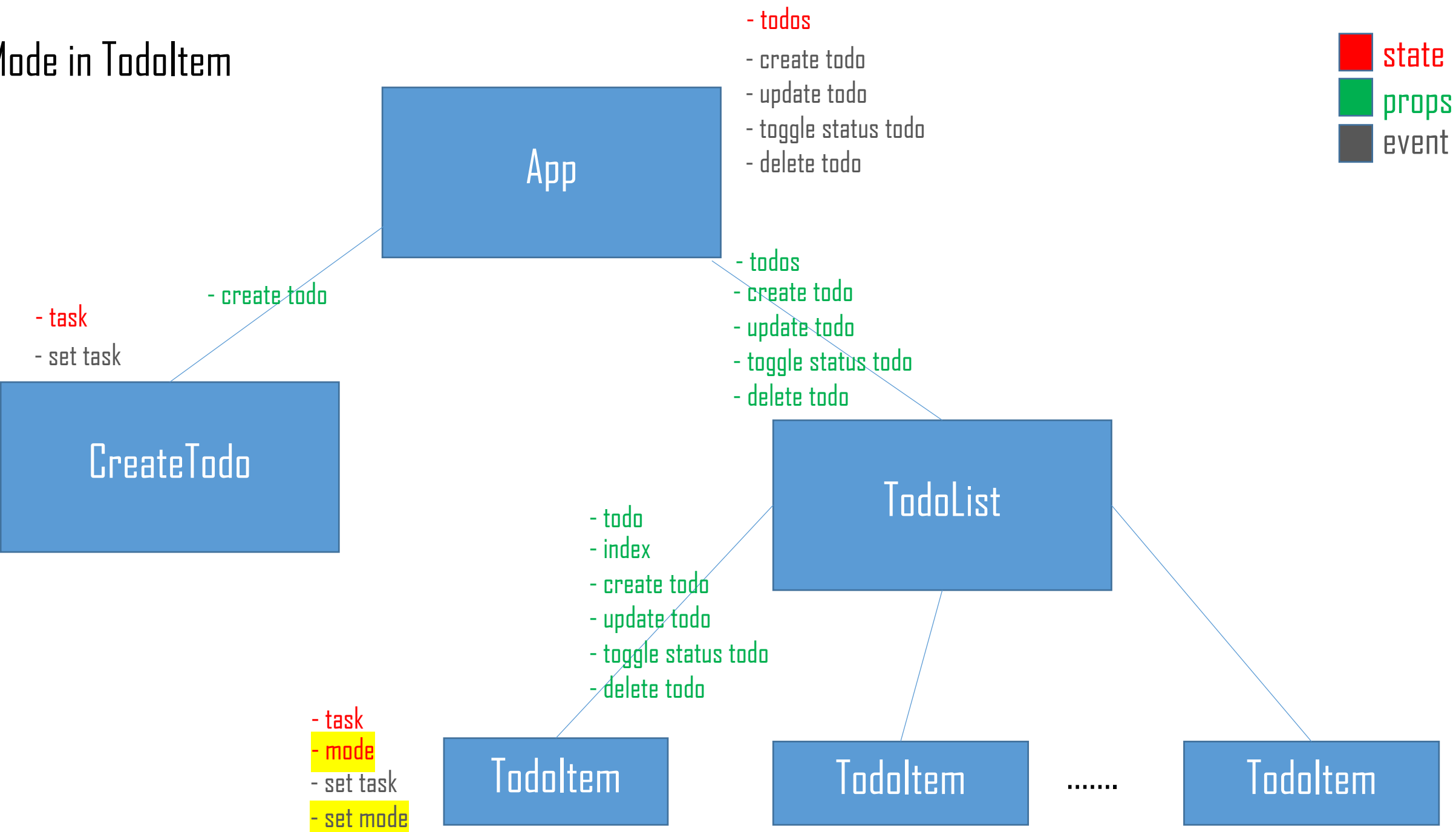
Input in CreateTodo



Input in TodoItem



Mode in TodoItem



App Component

State

```
- todos = [  
  { task: 'Eating' , isComplete: false } ,  
  { task: 'Learning' , isComplete: false } ,  
  { task: 'Playing' , isComplete: true }  
]
```

Event

```
- create todo is function :: (newTask) => { ... }  
- update todo is function :: (indexUpdate, newTask) => { ... }  
- toggle status todo is function :: (indexToggle) => { ... }  
- delete todo is function :: (indexDelete) => { ... }
```

CreateTodo Component

Props

- `create todo` is function (prop from App)

State

- `task` = ""

Event

- `set task` is function :: (e) => { ... } // will use e.target.value

- TodoList Component

Props

- `todos` is array (prop from App)
- `update todo` is function (prop from App)
- `toggle status todo` is function (prop from App)
- `delete todo` is function (prop from App)

- TodoItem Component

Props

- `todo` = { task: 'Eating' , isComplete: true } // `todos[index]`
- `index` = 0 // index in todos
- `update todo` is function (prop from App)
- `toggle status todo` is function (prop from App)
- `delete todo` is function (prop from App)

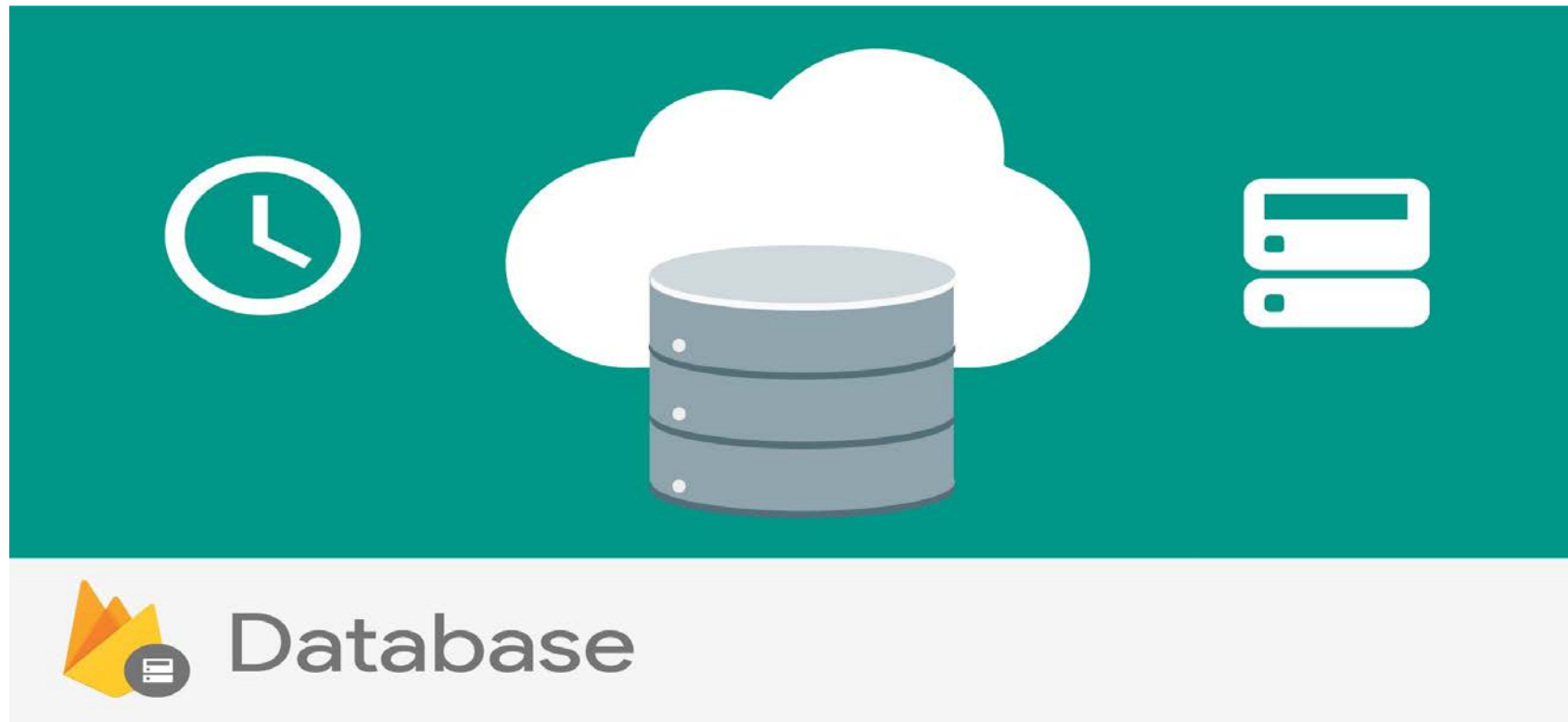
State

- `task` = '' // initial value by `todo.task`
- `mode` = 'SHOW' or 'EDIT'

Event

- `set task` is function :: (e) => { ... } // will use `e.target.value`
- `set mode` is function :: (mode) => { ... }

Firestore Realtime Database



Firestore Realtime Database

- Create Project in Firestore
- Design Database
- Re-Design React Component
- Recommend
- Coding

- Design Database

- Q: What data is stored in the database?

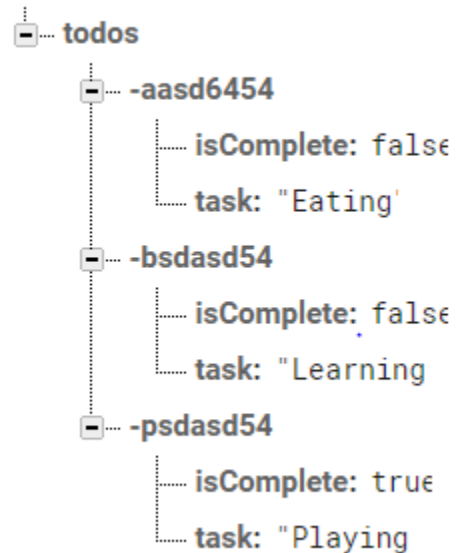
- A: todos

- Q: What is the format of the data?

- A:

- Design Database

todo-list-9ed46



JSON FORMAT

```
{
  todos: {
    "-aasd6454": { task: "Eating" , isComplete: false },
    "-bsdasd54": { task: 'Learning' , isComplete: false },
    "-psdasd54": { task: 'Playing' , isComplete: true }
  }
}
```

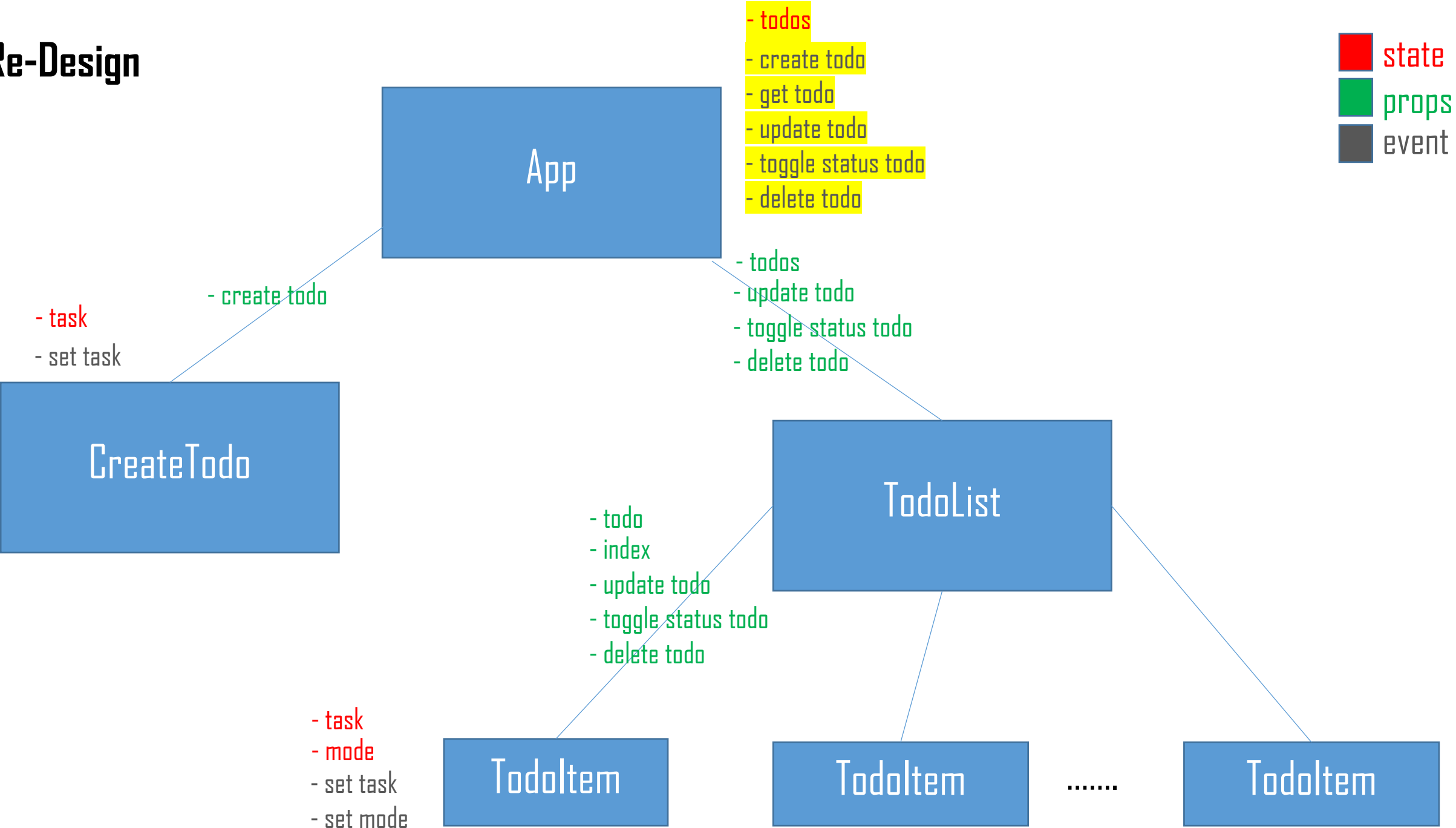
**** Firebase Realtime Database is not have Array**

- Re-Design React Component



If You want

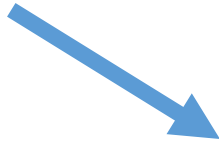
Re-Design



- Recommend

From Firebase

```
todos = {  
  "-aasd6454": { task: "Eating", isComplete: false },  
  "-bsdasd54": { task: 'Learning', isComplete: false },  
  "-psdasd54": { task: 'Playing', isComplete: true }  
}
```



In React

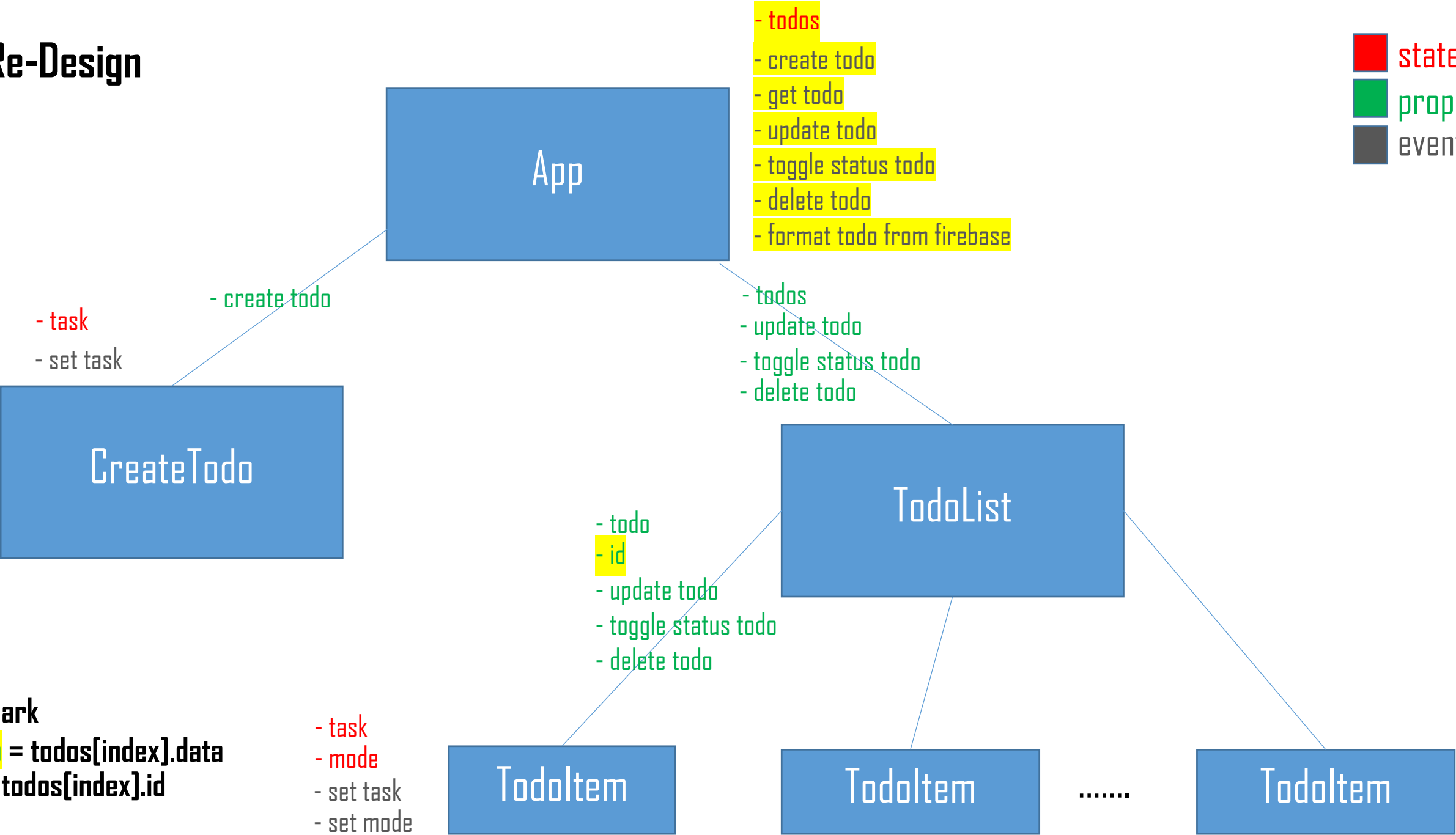
```
todos = [  
  { id: '-aasd6454', data: { task: 'Eating', isComplete: false } },  
  { id: '-bsdasd54', data: { task: 'Learning', isComplete: false } },  
  { id: '-psdasd54', data: { task: 'Playing', isComplete: true } }  
]
```

Re-Design

state

props

event



Remark
todo = todos[index].data
id = todos[index].id

App Component (Re-Design)

State

```
todos = [  
  { id: '-aasd6454', data: { task: 'Eating', isComplete: false } },  
  { id: '-bsdasd54', data: { task: 'Learning', isComplete: false } },  
  { id: '-psdasd54', data: { task: 'Playing', isComplete: true } }  
]
```

Event

- create todo is function :: (newTask) => { ... }
- get todo is function :: () => { ... } // realtime get data (subscribe)
- update todo is function :: (idUpdate, newTask) => { ... }
- toggle status todo is function :: (idToggle) => { ... }
- delete todo is function :: (idDelete) => { ... }
- format todo from firebase is function :: (todosFromFirebase) => { ...; return newTodos; }

- TodoItem Component (Re-Design)

Props

- `todo` = { task: 'Eating', isComplete: true } // `todos[index].data`
- `id` = 'aasd6454' // `todos[index].id`
- `update todo` is function (prop from App)
- `toggle status todo` is function (prop from App)
- `delete todo` is function (prop from App)

State

- `task` = '' // initial value by `todo.task`
- `mode` = 'SHOW' or 'EDIT'

Event

- `set task` is function :: (e) => { ... } // will use `e.target.value`
- `set mode` is function :: (mode) => { ... }

- Coding



```
24 <?php language_attributes(); ?>
25 <?php bloginfo( 'charset' ); ?>
26 <?php wp_title( '|', true, 'right' ); ?>
27 <?php rel="profile" href="http://gmpg.org/xfn/11" ?>
28 <?php rel="pingback" href="php bloginfo( 'pingback_url' ); ?&gt;
29 &lt;?php fruitful_get_favicon(); ?&gt;
30 &lt;?php wp_head(); ?&gt;
31 &lt;?php body_class(); ?&gt;
32 &lt;div id="page-header" class="hfeed site"&gt;
33     $theme_options = fruitful_get_theme_options();
34     $logo_pos = $menu_pos = "";
35     if (isset($theme_options['logo_position']))
36         $logo_pos = esc_attr($theme_options['logo_position']);
37     if (isset($theme_options['menu_position']))
38         $menu_pos = esc_attr($theme_options['menu_position']);
39     $logo_pos_class = fruitful_get_class_position($logo_pos);
40     $menu_pos_class = fruitful_get_class_position($menu_pos);
41     $responsive_menu_type = fruitful_get_responsive_menu_type($menu_pos);
42     $responsive_menu_type = fruitful_get_responsive_menu_type($menu_pos);</pre
```


Firestore Hosting



Hosting

Firestore Hosting

- Create Project in Firestore
- What is Firestore Hosting
- Step to Deploy

- What is Firebase Hosting

Firebase Hosting provides fast and secure static hosting for your web app.

- Step to Deploy

1. Install the Firebase CLI

```
npm install -g firebase-tools
```

2. Initialize your app

```
firebase init
```

 and edit file `firebase.json` in your project

3. Add a file

```
npm run build
```

- Step to Deploy

4. Deploy your website

```
firebase deploy
```

5. Manage and rollback deploys

From the **Hosting** panel in the [Firebase Console](#) you can see a full history of your deploys. To rollback to a previous deploy, hover over its entry in the list, click the overflow menu icon, then click "Rollback".

Reference

- **Firebase Realtime Database:**
<https://firebase.google.com/docs/database/>
- **Firebase Hosting:**
<https://firebase.google.com/docs/hosting/>

