Fetch data from backend

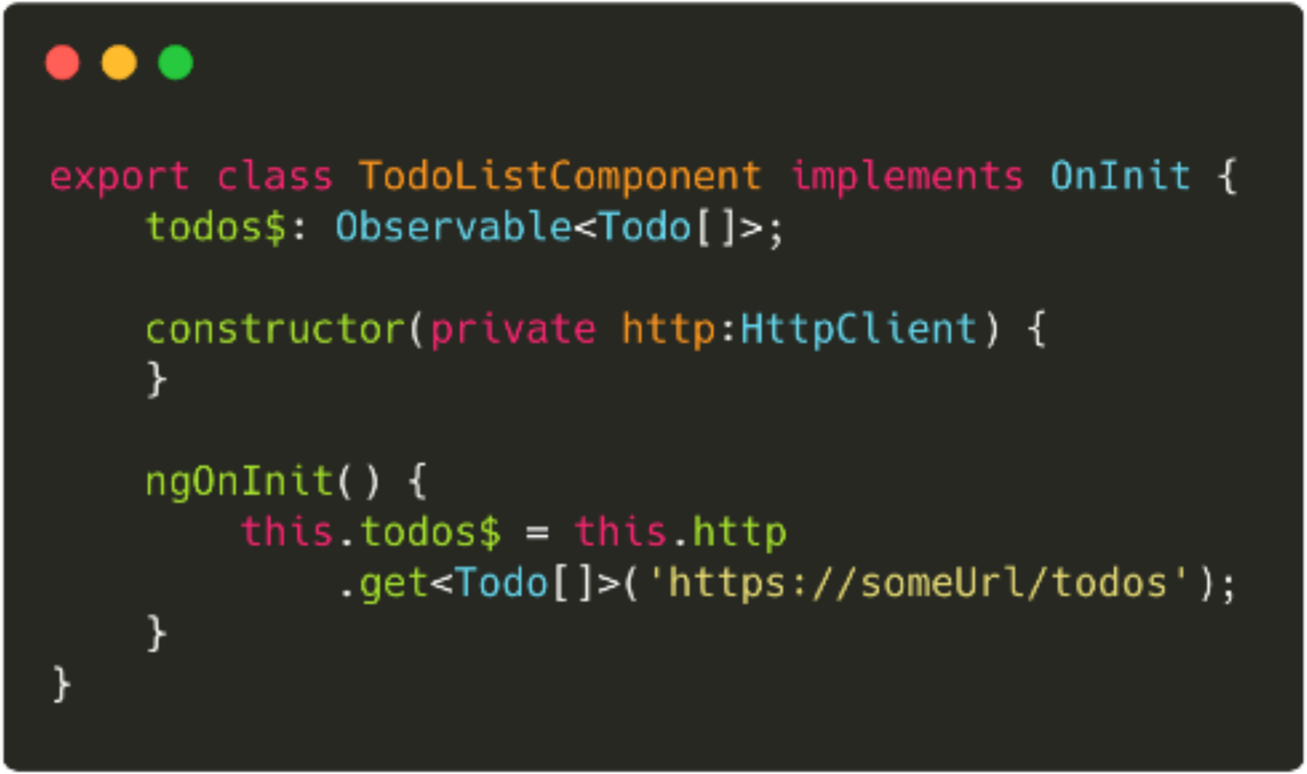
❌ Not part of vue

📦 i.e. axios or vue-resource

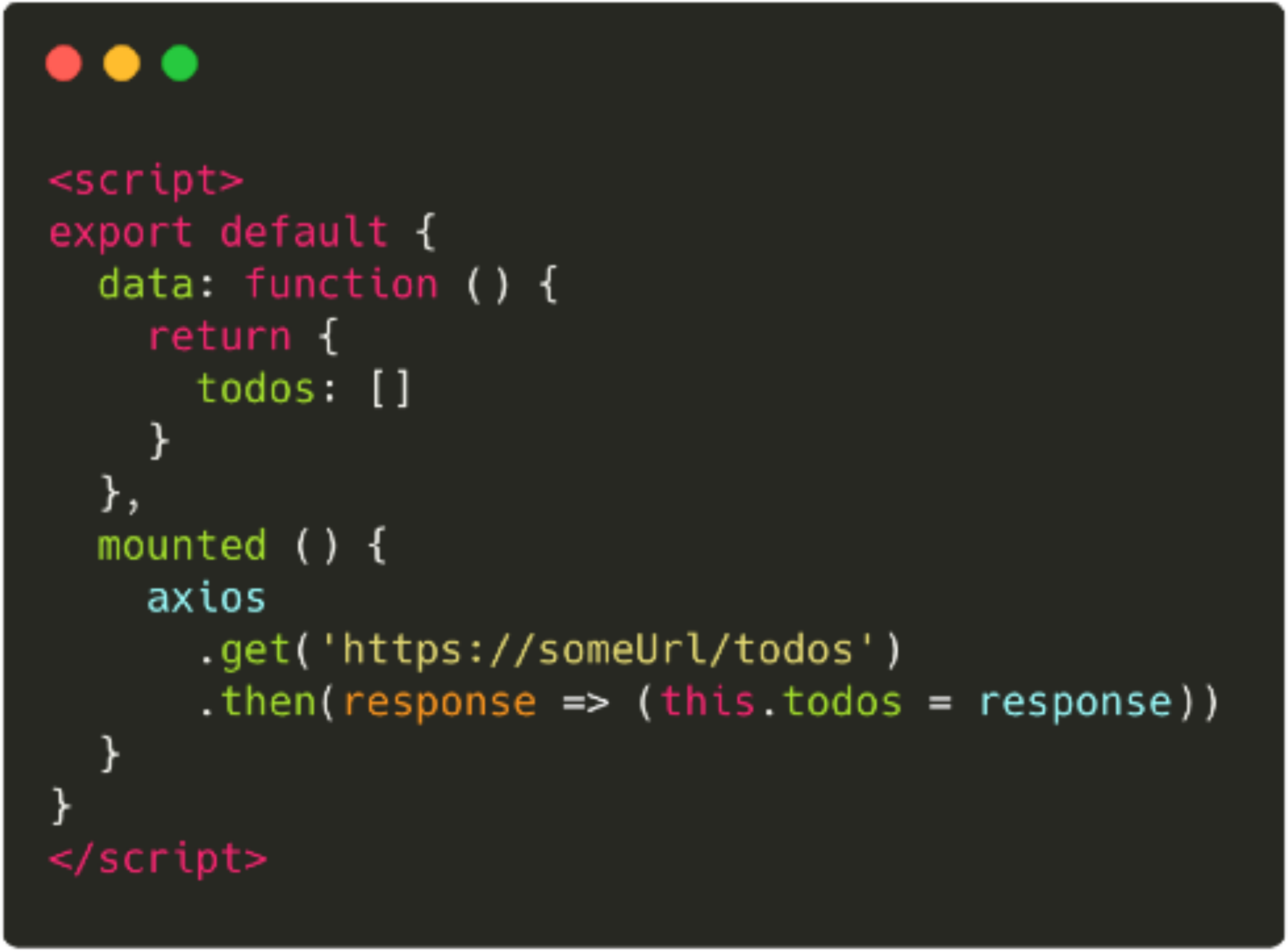
📋 https://github.com/vuejs/awesome-vue#http-requests

✅ Part of angular

📦 @angular/common

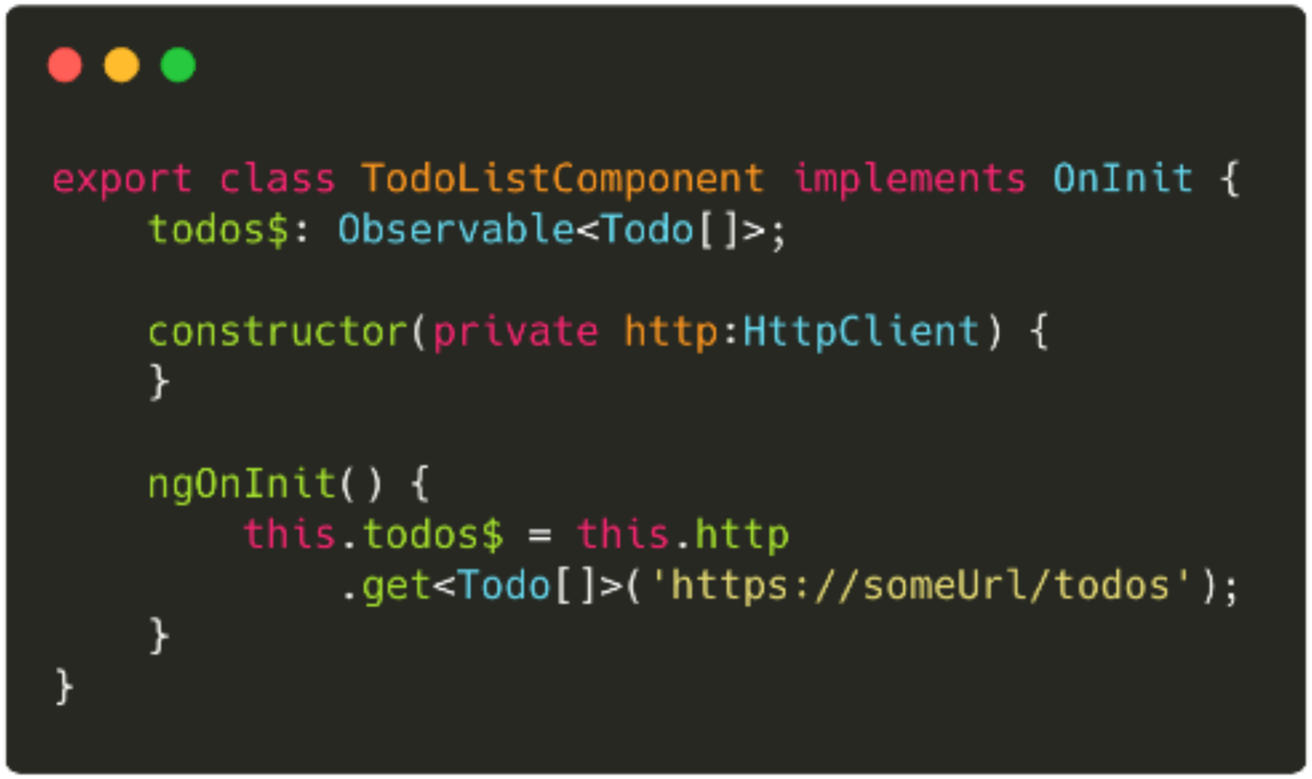


```
export class TodoListComponent implements OnInit {
    todos$: Observable<Todo[]>;

    constructor(private http:HttpClient) {
    }

    ngOnInit() {
        this.todos$ = this.http
            .get<Todo[]>('https://someUrl/todos');
    }
}
```

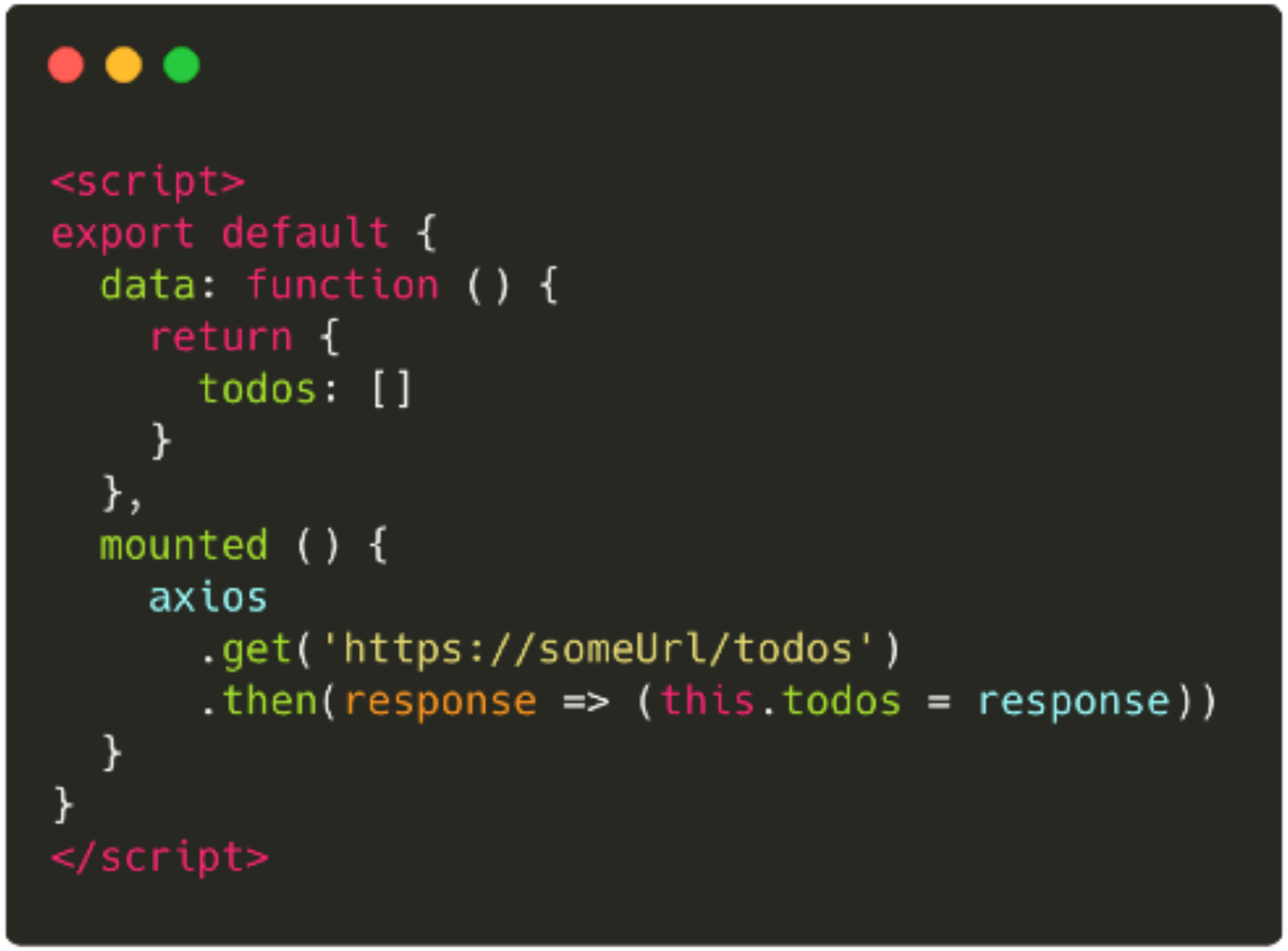


```
<script>
export default {
  data: function () {
    return {
      todos: []
    }
  },
  mounted () {
    axios
      .get('https://someUrl/todos')
      .then(response => (this.todos = response))
  }
}
</script>
```

Fetch data from backend



```
export class TodoListComponent implements OnInit {
    todos$: Observable<Todo[]>;

    constructor(private http:HttpClient) {
    }

    ngOnInit() {
        this.todos$ = this.http
            .get<Todo[]>('https://someUrl/todos');
    }
}
```

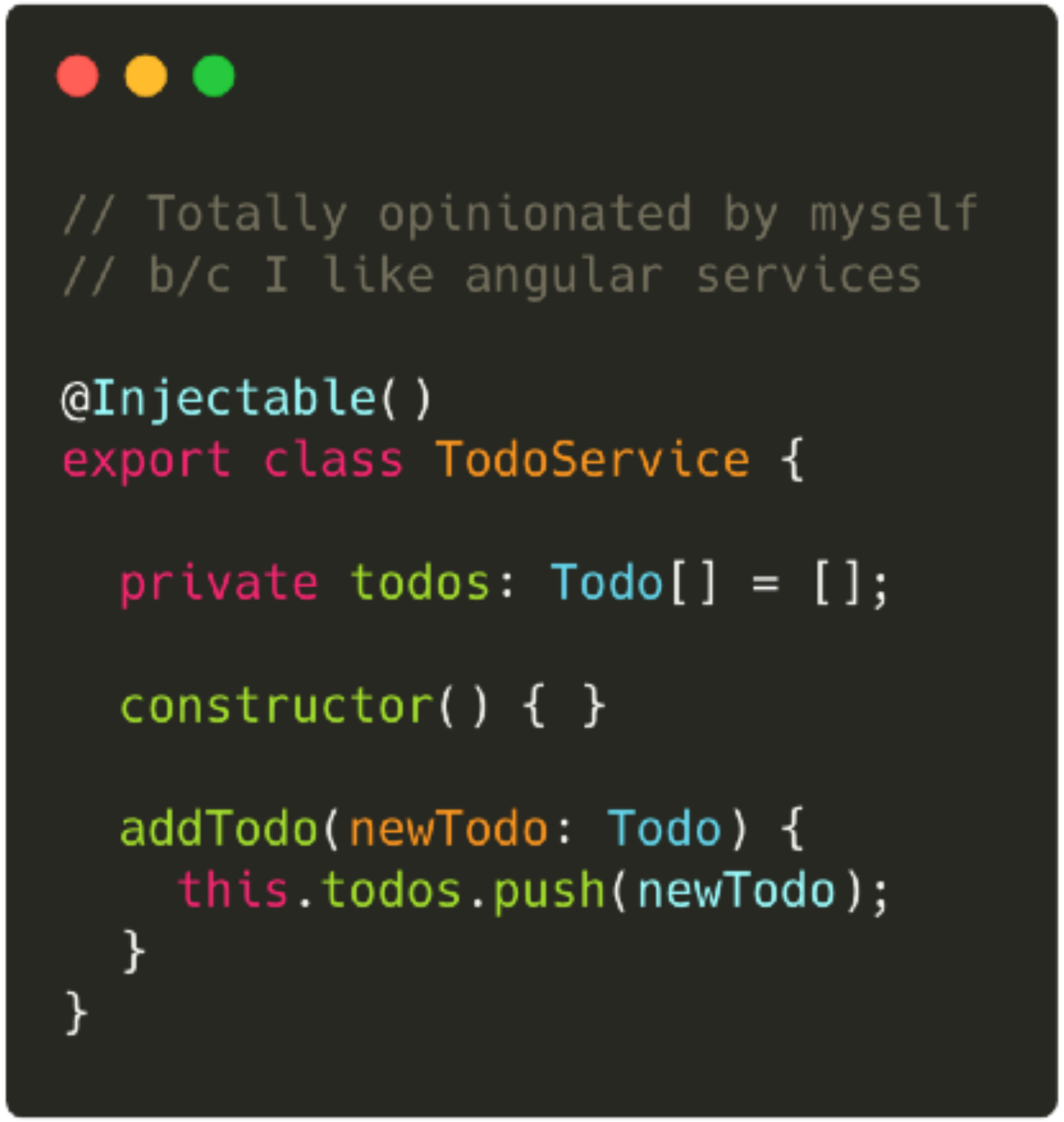


```
<script>
export default {
  data: function () {
    return {
      todos: []
    }
  },
  mounted () {
    axios
      .get('https://someUrl/todos')
      .then(response => (this.todos = response))
  }
}
</script>
```

✅ Part of angular
📦 @angular/common

❌ Not part of vue
📦 i.e. axios or vue-resource
📋 https://github.com/vuejs/awesome-vue#http-requests

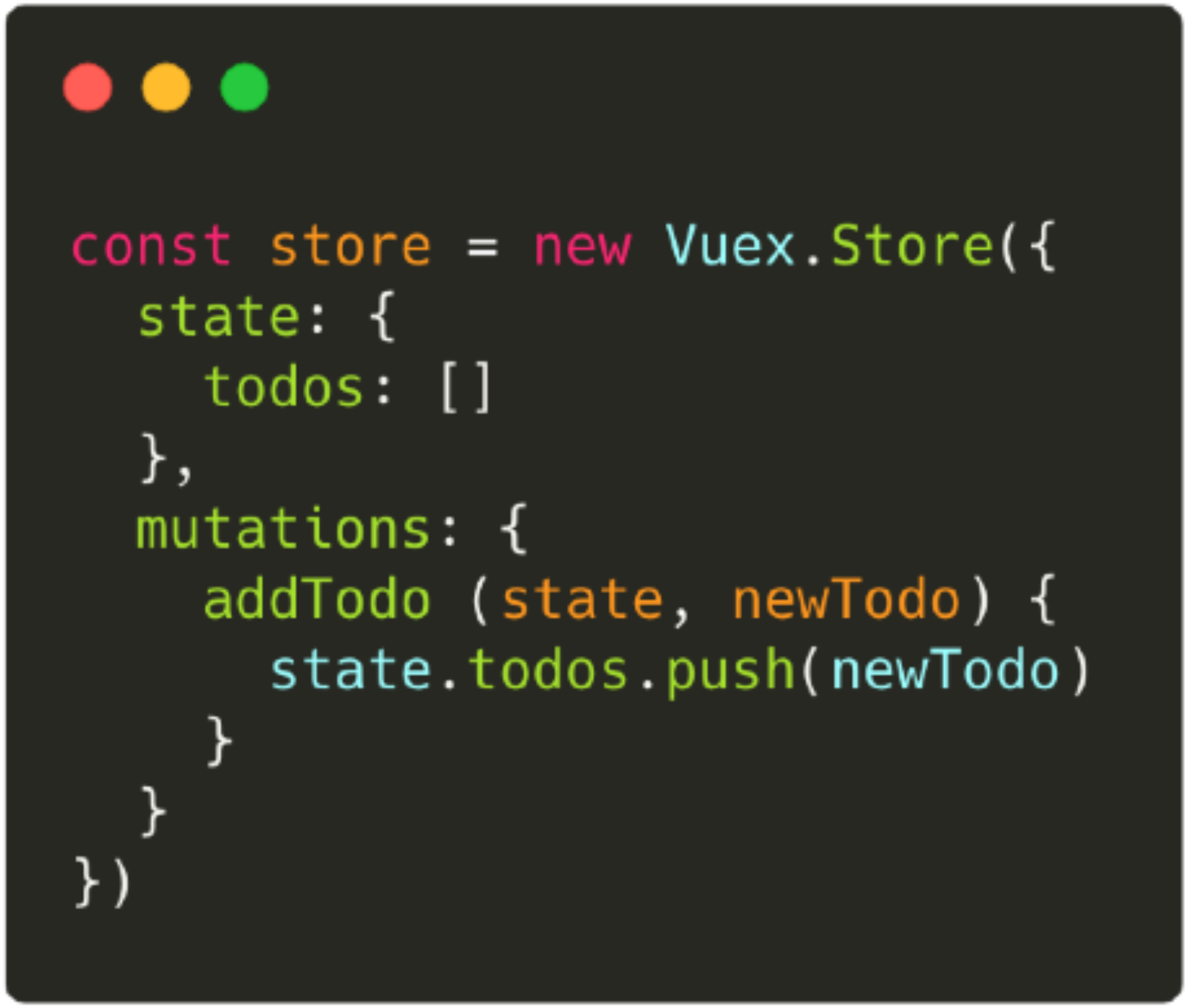
Manage state



```
// Totally opinionated by myself
// b/c I like angular services

@Injectable()
export class TodoService {

  private todos: Todo[] = [];

  constructor() { }

  addTodo(newTodo: Todo) {
    this.todos.push(newTodo);
  }
}
```

✅ Part of angular, but depends on you

📦 @angular/core or @ngrx/platform



```
const store = new Vuex.Store({
  state: {
    todos: []
  },
  mutations: {
    addTodo (state, newTodo) {
      state.todos.push(newTodo)
    }
  }
})
```

✅ Part of vue

📦 vuex

📋 https://github.com/vuejs/vuex