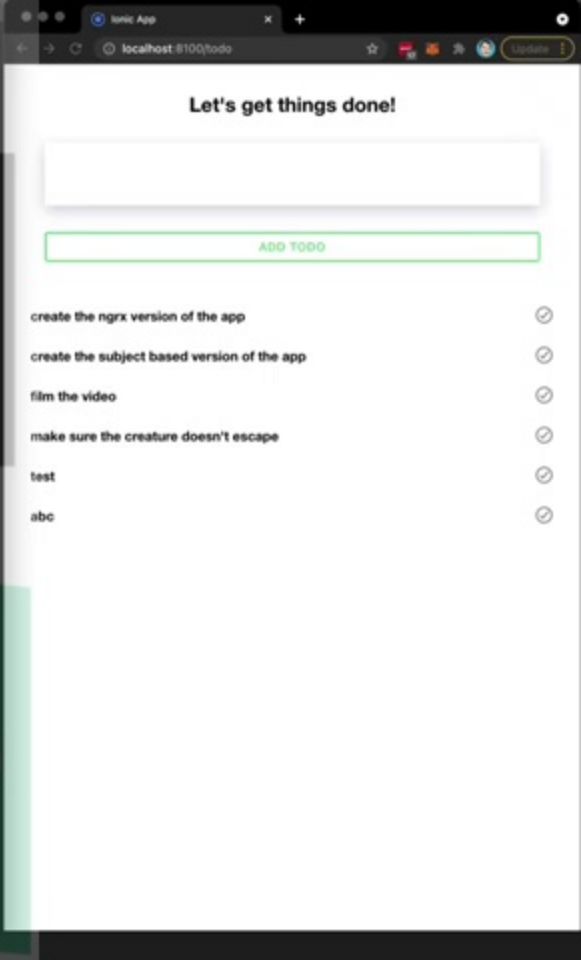


- app
- state
- todo
- app-routing.module.ts
- app.component.html
- app.component.scss
- app.component.spec.ts
- app.component.ts
- app.module.ts
- assets
- environments
- theme
- global.scss
- index.html
- main.ts
- polyfills.ts
- test.ts
- zone-flags.ts
- .browserslistrc
- .editorconfig
- .eslintrc.json
- .gitignore
- angular.json
- capacitor.config.ts
- ionic.config.json
- karma.conf.js
- package-lock.json
- package.json
- tsconfig.app.json
- tsconfig.json
- tsconfig.spec.json





<https://vizle.offnote.co>

Contact us: vizle@offnote.co

This document was generated automatically by **Vizle**

Your **Personal Video Reader Assistant**

Learn from Videos **Faster** and **Smarter**

VIZLE PRO / BIZ

- Convert *entire* videos ^{PDF, PPT}
- *Customize* to retain all essential content
- Include Spoken *Transcripts*
- Customer support

Visit <https://vizle.offnote.co/pricing> to learn more

VIZLE FREE PLAN

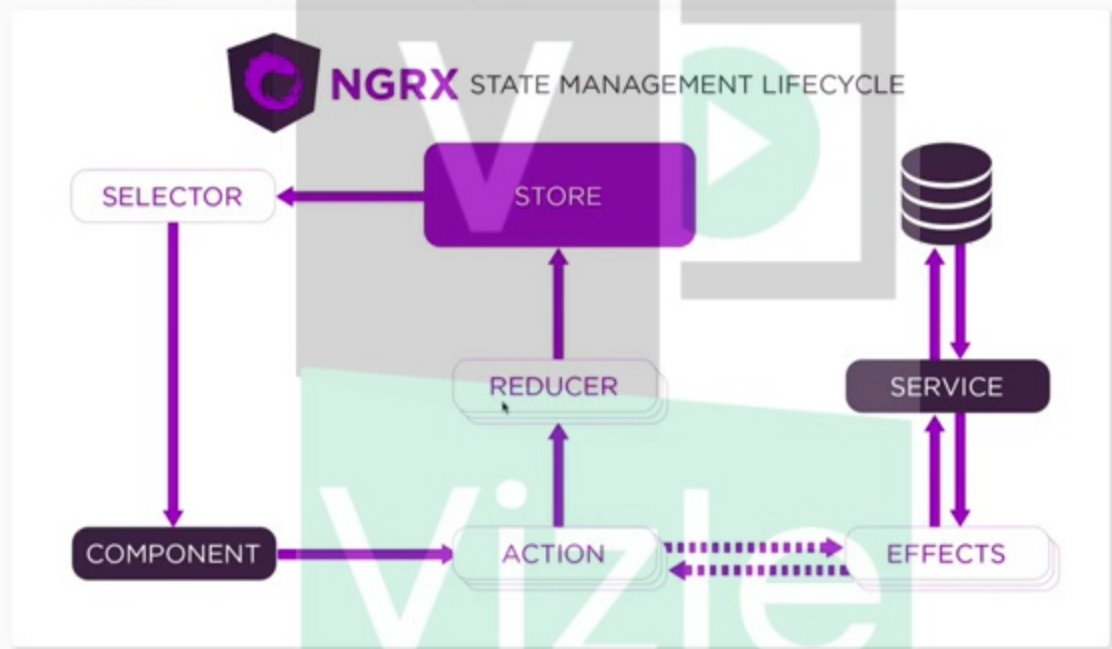
- Convert videos *partially* ^{PDF only}
- Slides may be *skipped**
- Usage restrictions
- No Customer support

Visit <https://vizle.offnote.co> to try free

Login to Vizle to unlock more slides*

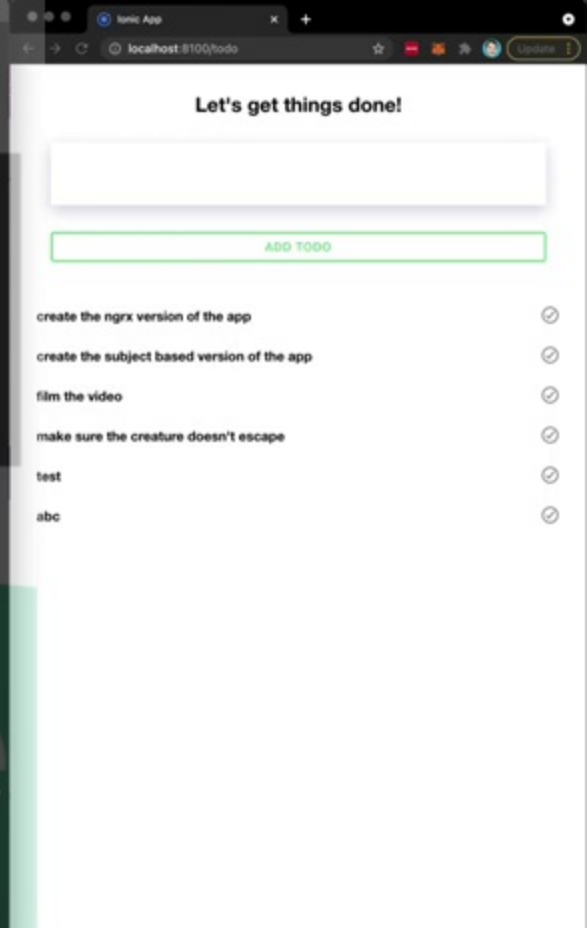
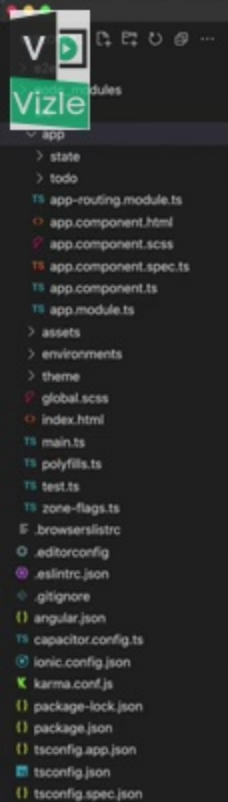
- Introduction
- State
 - @ngrx/store
 - Why use Store?
 - Getting Started
 - Walkthrough
 - Installation
 - Architecture
 - Advanced
 - Recipes
 - Configuration
 - Testing
 - @ngrx/store-devtools
 - @ngrx/effects
 - @ngrx/router-store
 - @ngrx/entity
 - @ngrx/schematics
 - @ngrx/component-store
- Data
- View
- Recipes
- Migrations
- API
- Contributing
- stable (v13.0.1)

The following diagram represents the overall general flow of application state in NgRx.

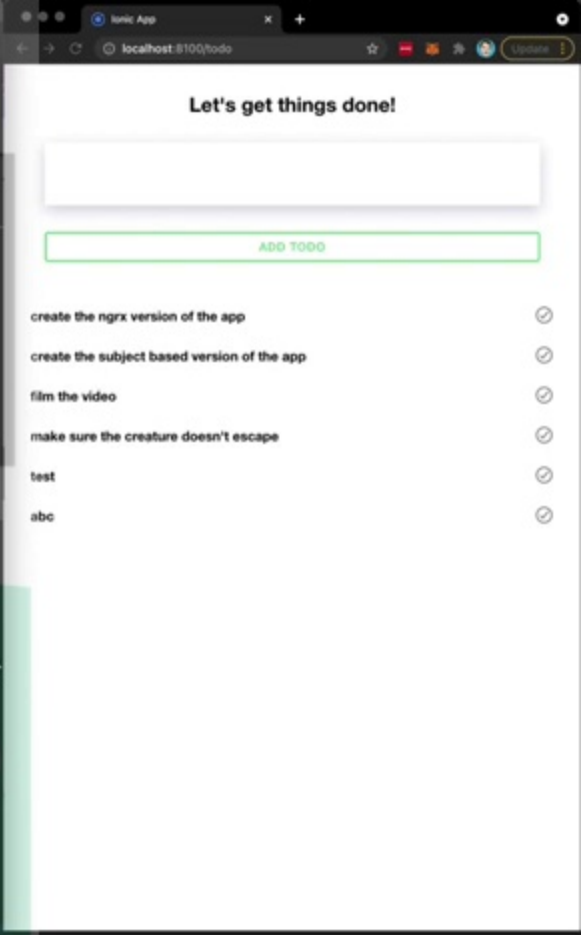


Note: All Actions that are dispatched within an application state are always first processed by the Reducers before being handled by the Effects of the application state.

- @ngrx/store
- Key concepts
- Local state management
- Installation
- Diagram
- Tutorial
- Next Steps



```
src > app > state > todos > TS todo.reducer.ts > @TodoState > status
8 } from './todo.actions';
9 import { Todo } from '../..../todo/todo.model';
10
11 export interface TodoState {
12   todos: Todo[];
13   error: string;
14   status: 'pending' | 'loading' | 'error' | 'success';
15 }
16
17 export const initialState: TodoState = {
18   todos: [],
19   error: null,
20   status: 'pending',
21 };
22
23 export const todoReducer = createReducer(
24   // Supply the initial state
25   initialState,
26   // Add the new todo to the todos array
27   on(addTodo, (state, { content }) => ({
28     ...state,
29     todos: [...state.todos, { id: Date.now().toString(), content: content }
30   })),
31   // Remove the todo from the todos array
32   on(removeTodo, (state, { id }) => ({
33     ...state,
34     todos: state.todos.filter((todo) => todo.id !== id),
35   })),
36   // Trigger loading the todos
37   on(loadTodos, (state) => ({ ...state, status: 'loading' })),
38   // Handle successfully loaded todos
```



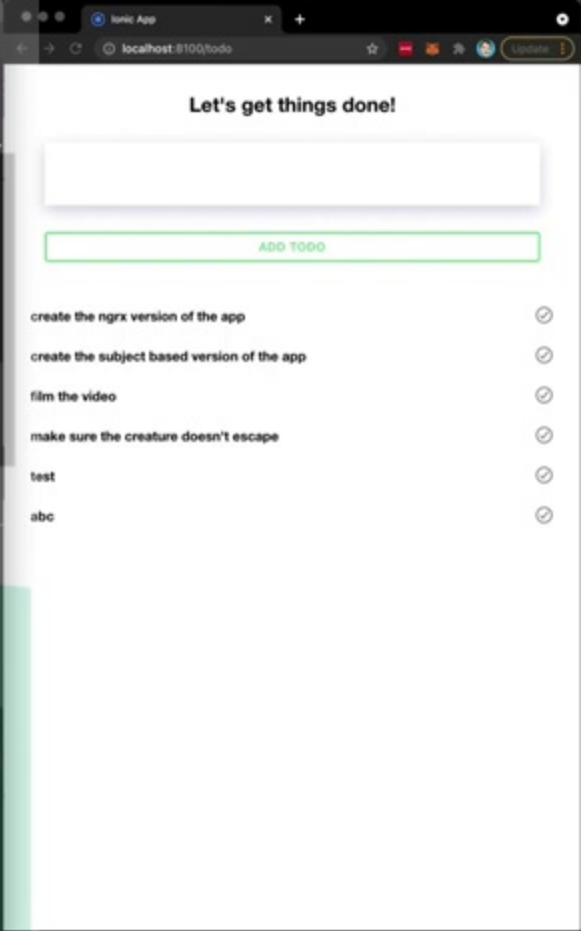
```
src > app > todo > todo.page.ts > @Todos$
1 import { Component, OnInit } from '@angular/core';
2 import { Store } from '@ngrx/store';
3 import { Observable } from 'rxjs';
4 import { addTodo, removeTodo, loadTodos } from '../state/todos/todo.actions';
5 import { selectAllTodos } from '../state/todos/todo.selectors';
6 import { Todo } from './todo.model';
7
8 @Component({
9   selector: 'app-todo',
10  templateUrl: 'todo.page.html',
11  styleUrls: ['todo.page.scss'],
12 })
13 export class TodoPage implements OnInit {
14   public allTodos$: Observable<Todo[]> = this.store.select(selectAllTodos);
15   public todo = '';
16
17   constructor(private store: Store) {}
18
19   ngOnInit() {
20     this.store.dispatch(loadTodos());
21   }
22
23   addTodo() {
24     this.store.dispatch(addTodo({ content: this.todo }));
25     this.todo = '';
26   }
27
28   removeTodo(todo: Todo) {
29     this.store.dispatch(removeTodo({ id: todo.id }));
30   }
31 }
32
```

localhost:8100/todo

Let's get things done!

- create the ngrx version of the app
- create the subject based version of the app
- film the video
- make sure the creature doesn't escape
- test
- abc

```
src > app > state > todos > todo.reducer.ts | @todo.reducer > on() callback > error
27 on(addTodo, (state, { content }) => ({
28   ...state,
29   todos: [...state.todos, { id: Date.now().toString(), content: content }
30 })),
31 // Remove the todo from the todos array
32 on(removeTodo, (state, { id }) => ({
33   ...state,
34   todos: state.todos.filter((todo) => todo.id !== id),
35 })),
36 // Trigger loading the todos
37 on(loadTodos, (state) => ({ ...state, status: 'loading' })),
38 // Handle successfully loaded todos
39 on(loadTodosSuccess, (state, { todos }) => ({
40   ...state,
41   (property) TodoState.error: string
42   error: null,
43   status: 'success',
44 })),
45 // Handle todos load failure
46 on(loadTodosFailure, (state, { error }) => ({ ...state, error: error }));
47
48
```



```
src > app > state > todos > TS todo.effects.ts > TS loadTodos$ > createEffect() callback
20 private actions$: Actions,
21 private store: Store<AppState>,
22 private todoService: TodoService
23 ) {}
24
25 // Run this code when a loadTodos action is dispatched
26 loadTodos$ = createEffect(() =>
27   this.actions$.pipe(
28     ofType(loadTodos),
29     switchMap(() =>
30       // Call the getTodos method, convert it to an observable
31       from(this.todoService.getTodos()).pipe(
32         // Take the returned value and return a new success action containing
33         map((todos) => loadTodosSuccess({ todos: todos })),
34         // Or... if it errors return a new failure action containing the error
35         catchError((error) => of(loadTodosFailure({ error })))
36       )
37     )
38   )
39 );
40
41 // Run this code when the addTodo or removeTodo action is dispatched
42 saveTodos$ = createEffect(
43   () =>
44     this.actions$.pipe(
45       ofType(addTodo, removeTodo),
46       withLatestFrom(this.store.select(selectAllTodos)),
47       switchMap(([action, todos]) => from(this.todoService.saveTodos(todos)
48     )),
49     // Most effects dispatch another action, but this one is just a "fire and
50     { dispatch: false }

```

localhost:8100/todo

Let's get things done!

- create the ngx version of the app
- create the subject based version of the app
- film the video
- make sure the creature doesn't escape
- test
- abc



<https://vizle.offnote.co>

Contact us: vizle@offnote.co

This document was generated automatically by **Vizle**

Your **Personal Video Reader Assistant**

Learn from Videos **Faster** and **Smarter**

VIZLE PRO / BIZ

- Convert *entire* videos ^{PDF, PPT}
- *Customize* to retain all essential content
- Include Spoken *Transcripts*
- Customer support

Visit <https://vizle.offnote.co/pricing> to learn more

VIZLE FREE PLAN

- Convert videos *partially* ^{PDF only}
- Slides may be *skipped**
- Usage restrictions
- No Customer support

Visit <https://vizle.offnote.co> to try free

Login to Vizle to unlock more slides*