

Advanced State management in Angular

.....

Angular Oslo meetup
14th, Feb 2018
Majid Hajian

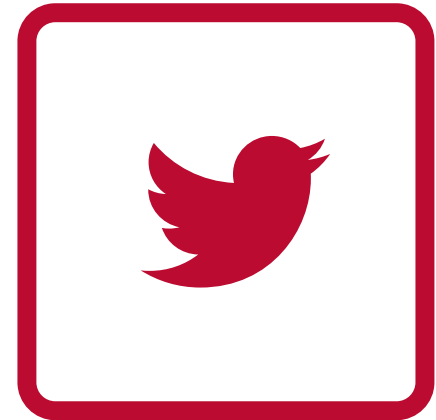




Majid Hajian

ABOUT ME

- Passionate web developer
- Angular enthusiast, Over 4+ experiences
- Open Source lover and contributor
- Public speaker,
- Trainer @PacktPub



@mhadaily

Redux



THREE PRINCIPLES

1. Single source of truth
2. State is read-only
3. Pure functions update state

SINGLE SOURCE OF TRUTH

- One state tree inside Store
- Predictability, maintainability
- Universal apps (SSR)
- Testing and debugging

STATE IS READ-ONLY

- Derive specific props from state
- Dispatch actions to change the state
- Immutable update patterns

PURE FUNCTIONS UPDATE STATE

- Pure functions (reducers)
- Reducers look for action types
- Return new state

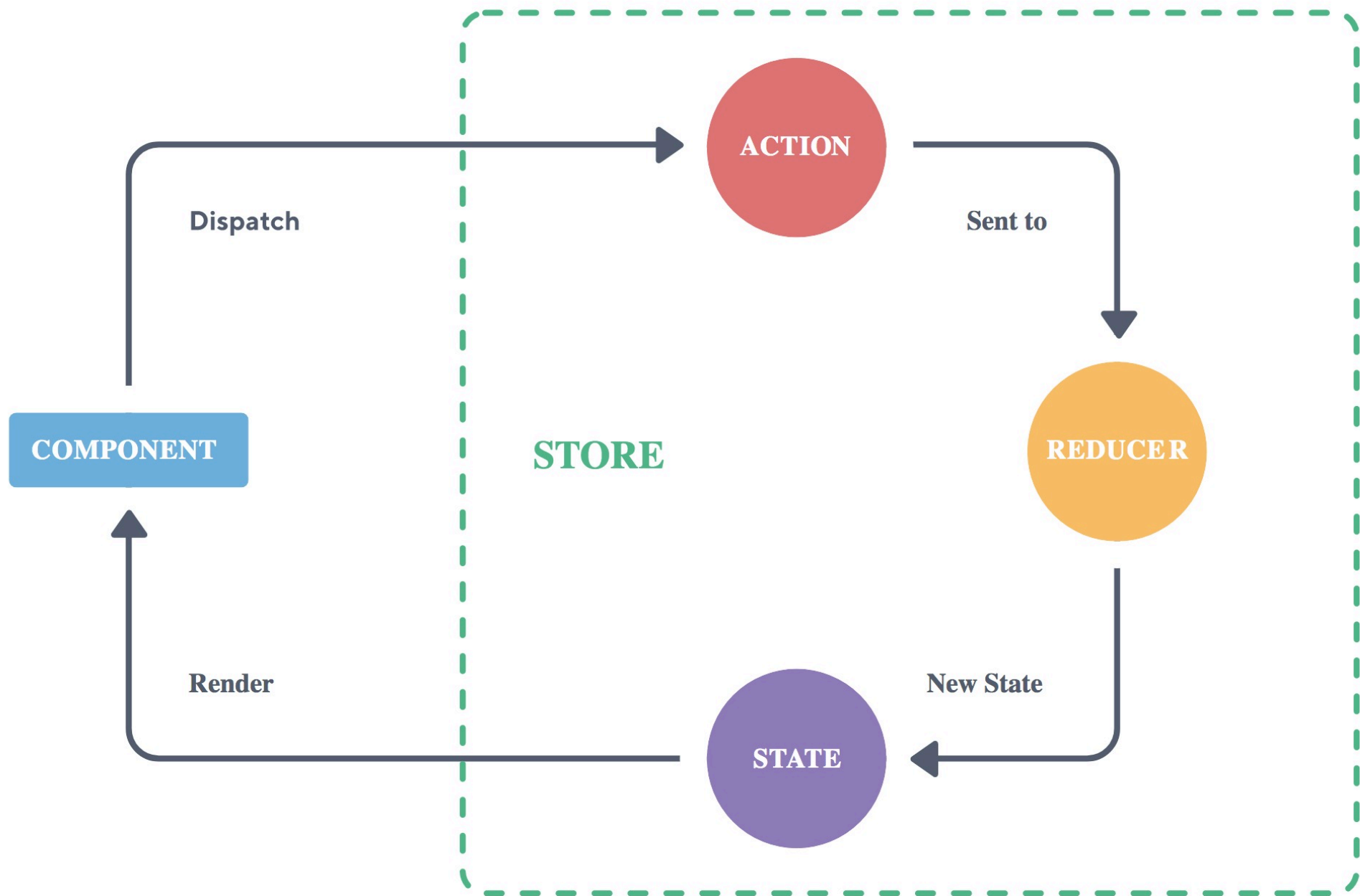
CORE CONCEPTS

1. Single state tree
2. Actions
3. Reducers
4. Store
5. One-way dataflow


```
const state = {  
  products: []  
}
```

```
const action = {  
  type: 'ADD_PRODUCT',  
  payload: {  
    id: 1,  
    title: 'Awesome Angular'  
  }  
}
```

```
function reducer(state, action) {  
  switch (action.type) {  
    case 'ADD_PRODUCT':  
      {  
        return {  
          products: [  
            ...state.products,  
            action.payload  
          ]  
        };  
      }  
    }  
  }  
  return state;  
}
```



NGRX



+



+



=



redux pattern

reactiveX Js

Angular

ngRx

[@ngrx/store](#) - RxJS powered state management for Angular applications, inspired by Redux

[@ngrx/effects](#) - Side Effect model for @ngrx/store to model event sources as actions.

[@ngrx/router-store](#) - Bindings to connect the Angular Router to @ngrx/store

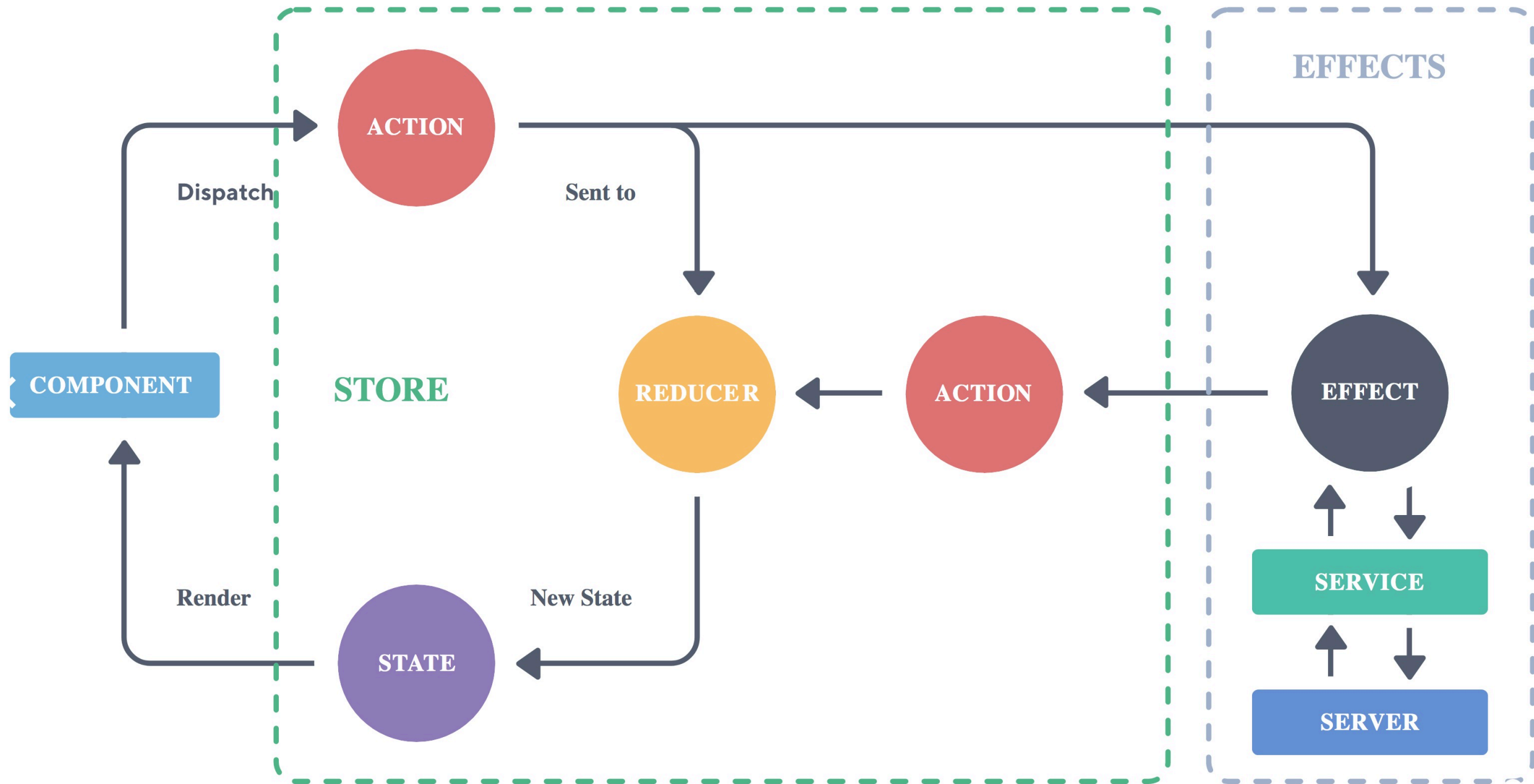
[@ngrx/store-devtools](#) - Store instrumentation that enables a powerful time-travelling debugger.

[@ngrx/entity](#) - Entity State adapter for managing record collections.

[@ngrx/schematics](#) - Scaffolding library for Angular applications using NgRx.

NGRX/STORE

- ✓ Single source of truth
- ✓ Testability
- ✓ Performance benefits
 - ✓ ChangeDetectionStrategy.OnPush
 - ✓ Immutable @Inputs
 - ✓ Object reference checks are fast
- ✓ Root and feature module support
 - ✓ Eagerly loaded modules
 - ✓ Lazily loaded modules



NGRX/EFFECTS

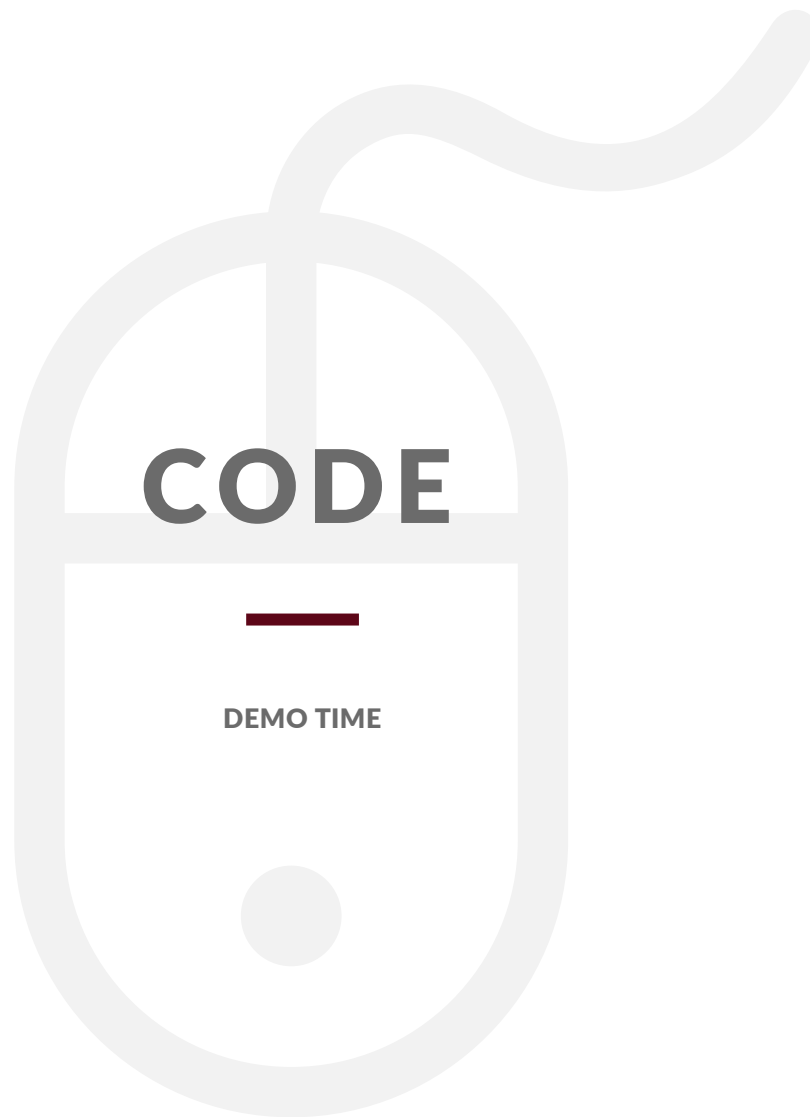
- Listen for @ngrx/store actions
- Isolate side effects from components
- Communicate outside of Angular

NGRX/ENTITY

- Reduces boilerplate for creating reducers that manage a collection of models.
- Provides performant CRUD operations for managing entity collections.
- Extensible type-safe adapters for selecting entity information.

NGRX/SCHEMATICS

- generating files when building out feature areas using NgRx. Built on top of Schematics, it integrates with the Angular CLI to make setting up and expanding NgRx in Angular applications easier.



THANK YOU

See You Next Time

MAJID HAJIAN



@mhadaily