

# The Redux Pattern

---



**Duncan Hunter**

CONSULTANT | SPEAKER | AUTHOR

@dunchunter duncanhunter.com.au











# Redux

Redux is a predictable state container for JavaScript apps



# Redux Principles

1. Single source of truth called the store
2. State is read only and only changed by dispatching actions
3. Changes are made using pure functions called reducers



# Module Overview



**3 principles of Redux**

**Store**

**Actions**

**Reducers**

**Advantages of Redux**



# The Store Is the Single Source of Truth

**Store (State)**

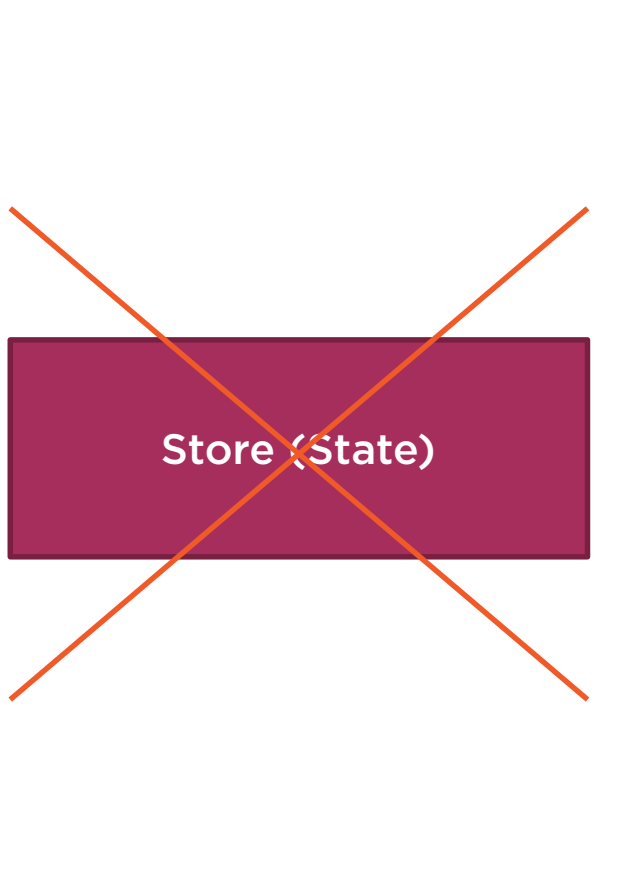
```
{
  showProductCode: true,
  currentProductId: 5,
  products: [
    {
      id: 1,
      productName: 'Leaf Rake',
      productCode: 'GDN-0011',
      description: 'Leaf rake with wooden handle',
      starRating: 3.2
    },
  ]
}
```



Do you have to put every  
piece of state in the store?



# What Should Not Go in the Store?



**Unshared state**

**Angular form state**

**Non serializable state**



# Dispatch an Action to Change State

## Actions

**Login action after a login form submission**

**Toggle side menu action after a button click**

**Retrieve data action when initializing a component**

**Start global spinner action when saving data**



# Dispatch an Action to Change State

Actions

```
{  
  type: 'LOGIN',  
  payload: {username: 'Duncan', password: 'secret'}  
}
```



# Use Reducers to Change State

Reducer

Set a userDetails property on login

Toggle a sideMenuVisible property to true on button click

Set successfully retrieved data on component initialization

Set a globalSpinnerVisible property to true while saving data



# Actions with Side Effects



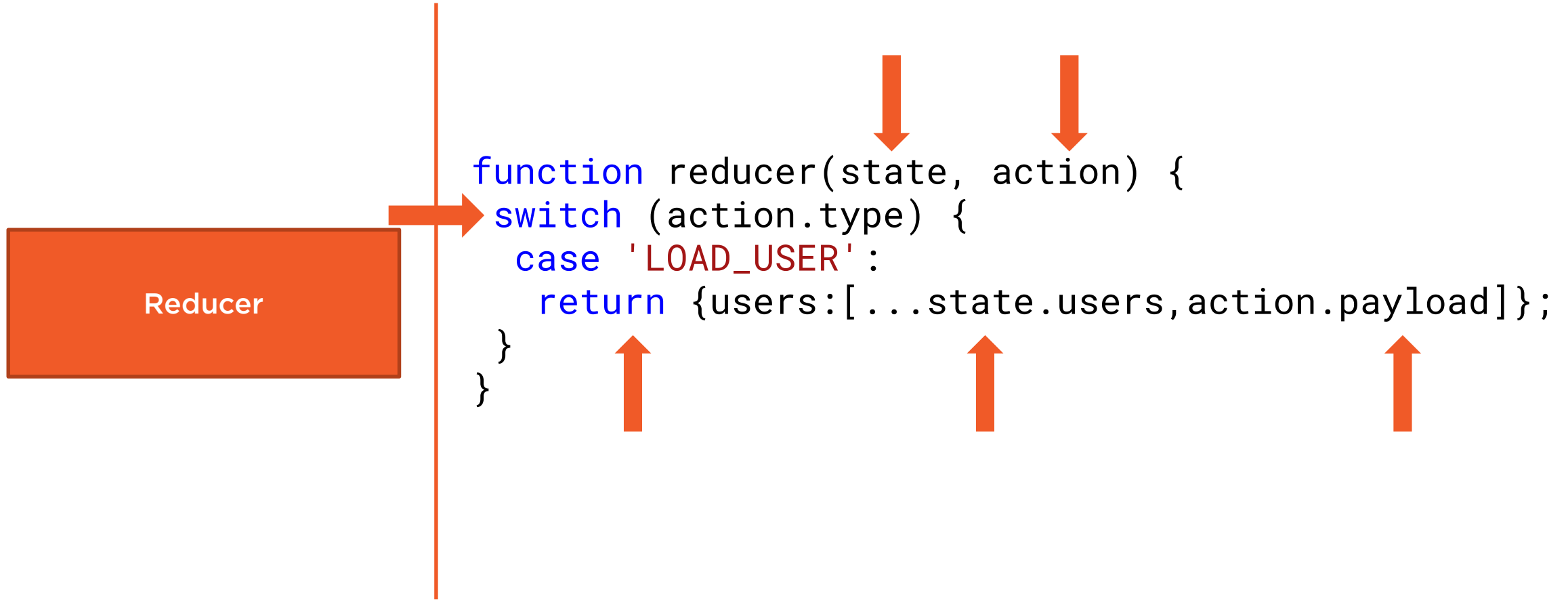
Reducer

**Manage side effects with the NgRx Effects library**





# Reducers Are Pure Functions



# What is a pure function?



# Pure and Impure Functions

Pure  
Function

```
function sum(a, b) {  
  const result = a + b;  
  return result;  
}
```

Impure  
Function

```
let c = 1;  
  
function sum(a, b) {  
  result = a + b + c;  
  return result;  
}
```



# Advantages of the Redux Pattern



Store (State)

**Centralized immutable state**

**Performance**

**Testability**

**Tooling**

**Component communication**





Redux is not great for making simple things quickly. It's great for making really hard things simple.

**Jani Evakallio**



# Checklist: Redux Principles



\_\_\_\_\_



\_\_\_\_\_



\_\_\_\_\_

Single source of truth called the store

State is read-only and changed by dispatching actions

Changes are made using pure functions called reducers

