

NgRx Fundamentals

Introduction



Duncan Hunter

Lead Frontend Developer

@dunchunter | duncanhunter.com.au

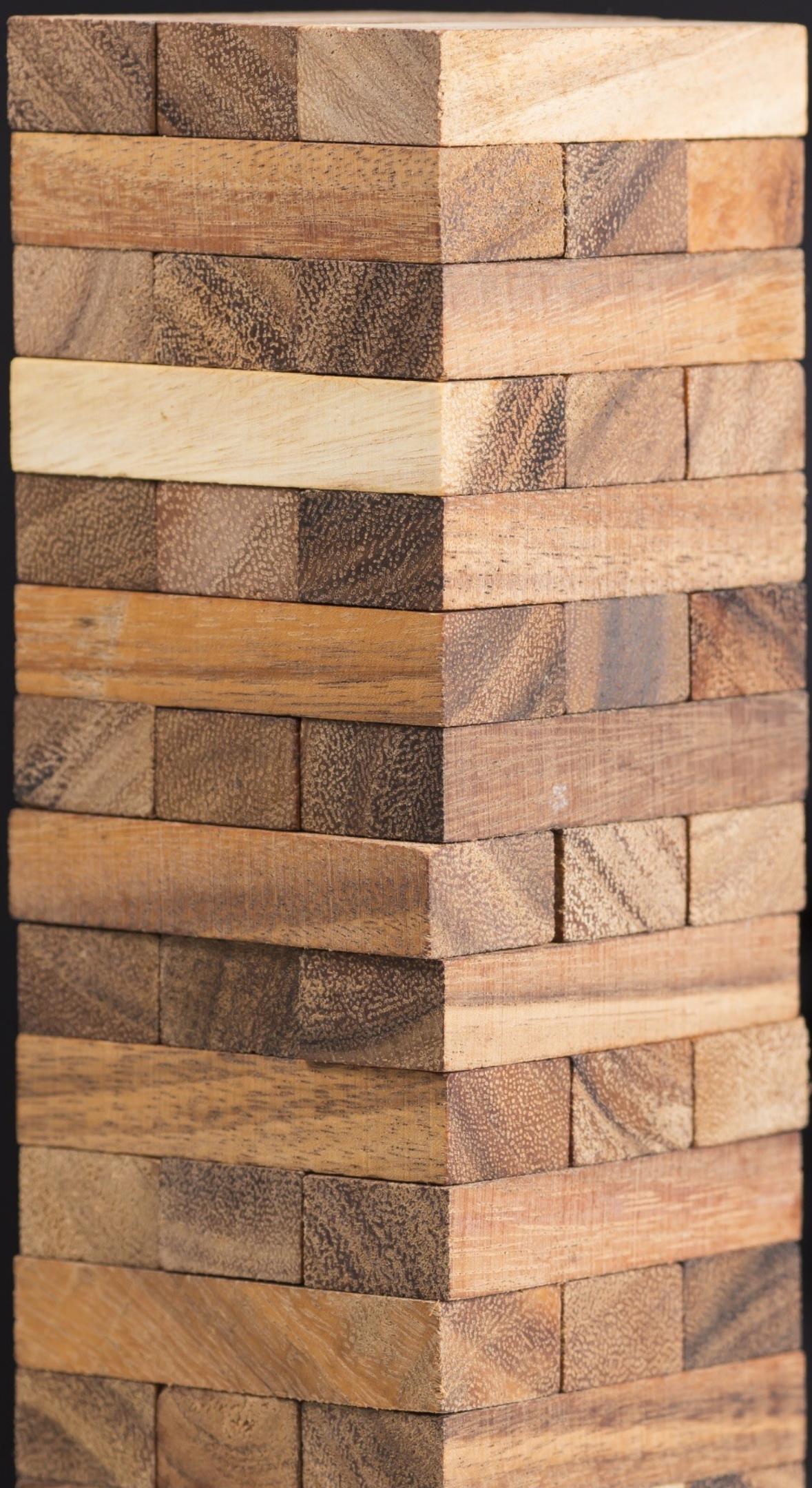




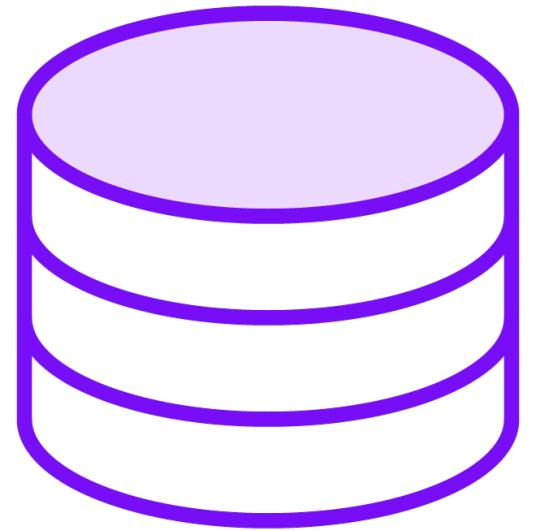




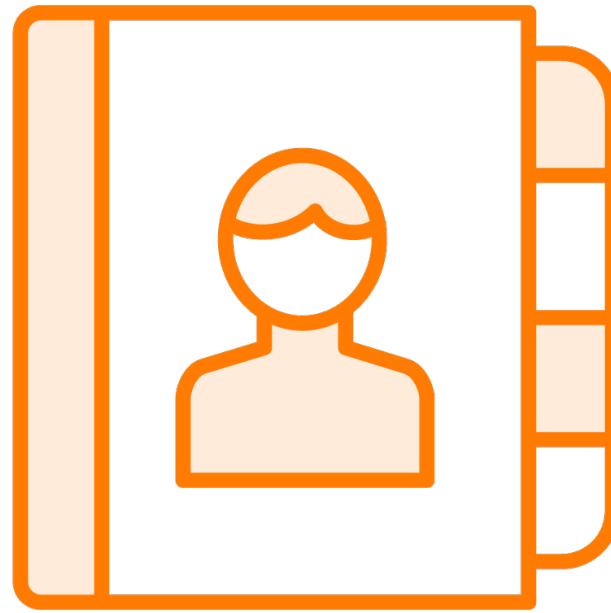




Types of State?



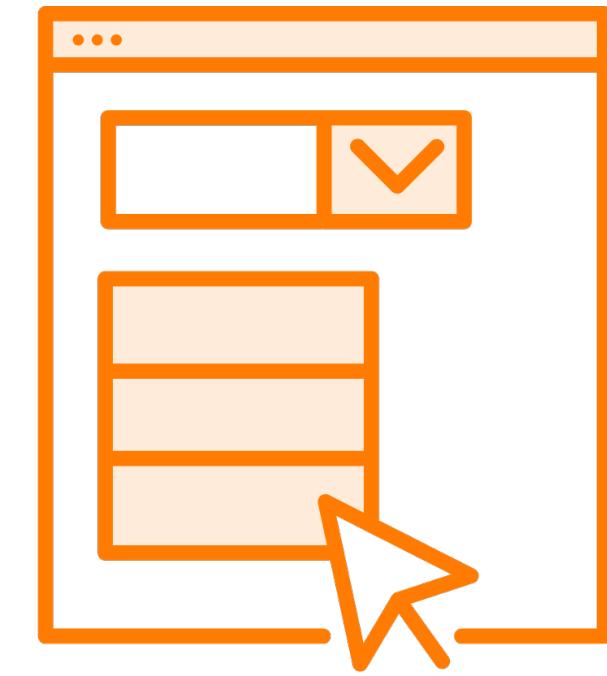
Persisted state



URL state



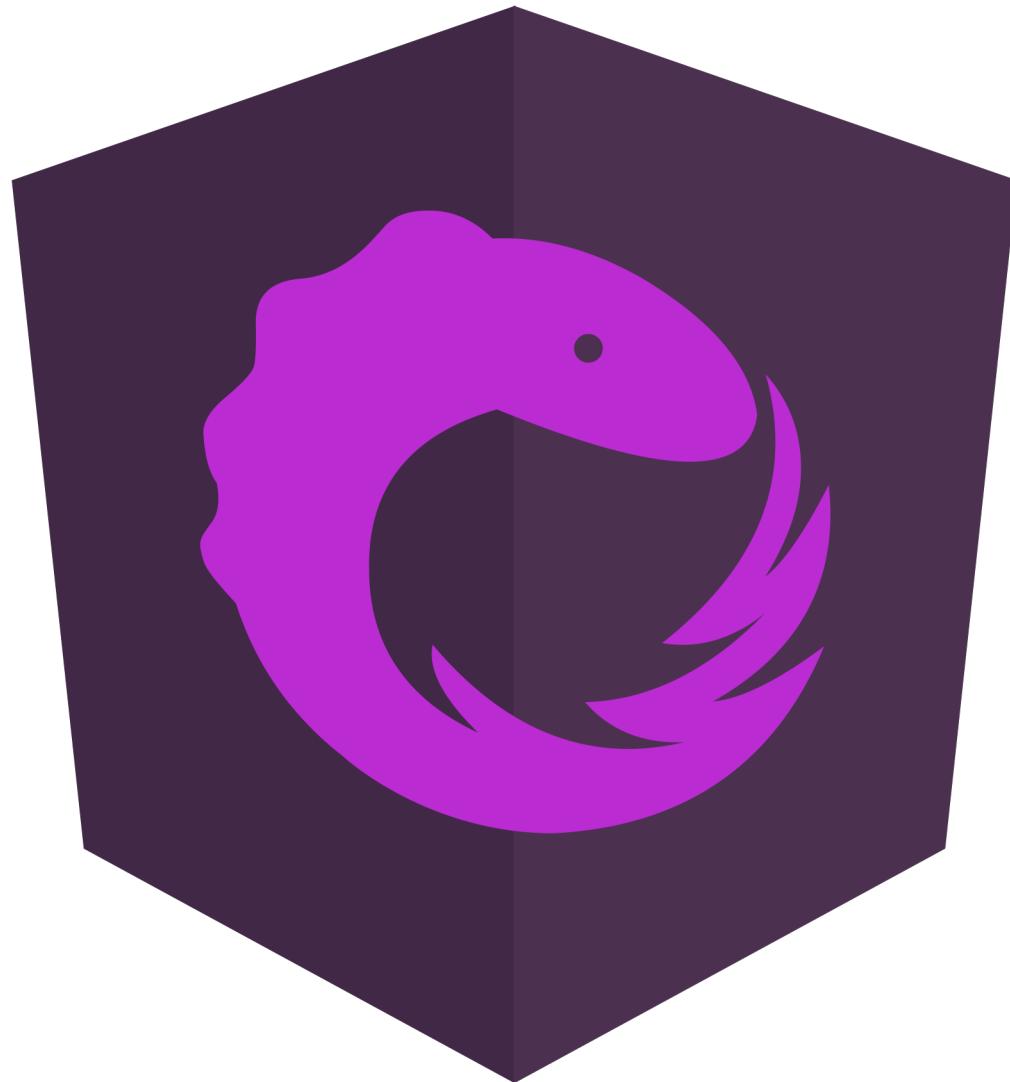
Client state



Local UI state



Purpose of NgRx



Provide a framework for building reactive applications in Angular

- Managing global and local state
- Isolation of side effects
- Entity collection management
- Integration with the Angular Router
- Developer tooling



Introduction

- State >
 - @ngrx/store >
 - @ngrx/effects >
 - @ngrx/router-store >
 - @ngrx/entity >
 - @ngrx/component-store >

What is NgRx?

NgRx is a framework for building reactive applications in Angular. NgRx provides libraries for:

- Managing global and local state.
 - Isolation of side effects to promote a cleaner component architecture.
 - Entity collection management.
 - Integration with the Angular Router.
 - Developer tooling that enhances developer experience when building many different types of applications.

Packages

NgRx packages are divided into a few main categories

State ↗

- **Store** - RxJS powered global state management for Angular apps, inspired by Redux.
 - **Effects** - Side effect model for `@ngrx/store`.
 - **Router Store** - Bindings to connect the Angular Router to `@ngrx/store`.
 - **Entity** - Entity State adapter for managing record collections.
 - **ComponentStore** - Standalone library for managing local/component state.

Data

- **Data** - Extension for simplified entity data management

View

- **Component** - Extension for building reactive Angular templates.

Module Introduction



Introduce NgRx

Examine the problems it solves

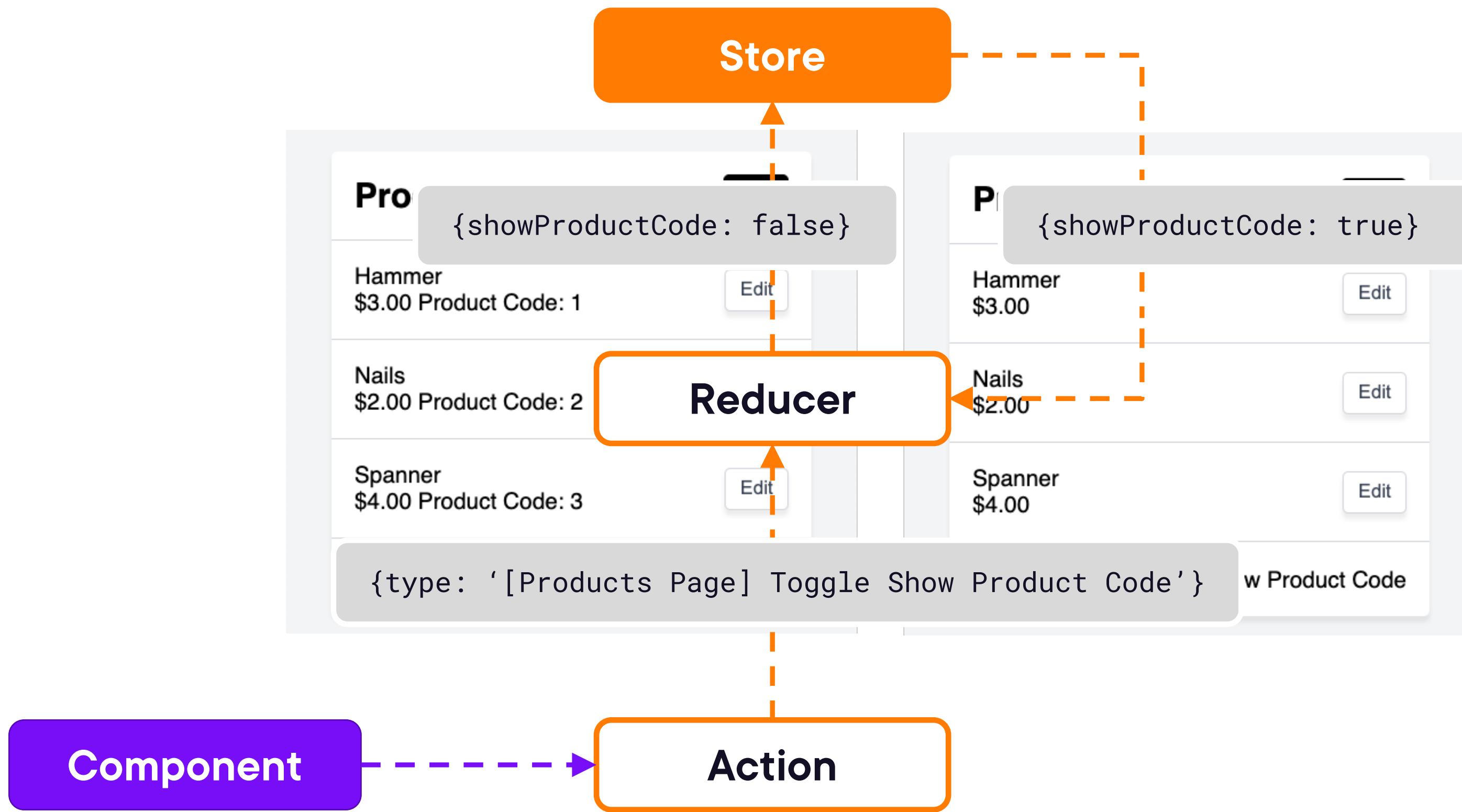
Getting the most from this course

Sample application

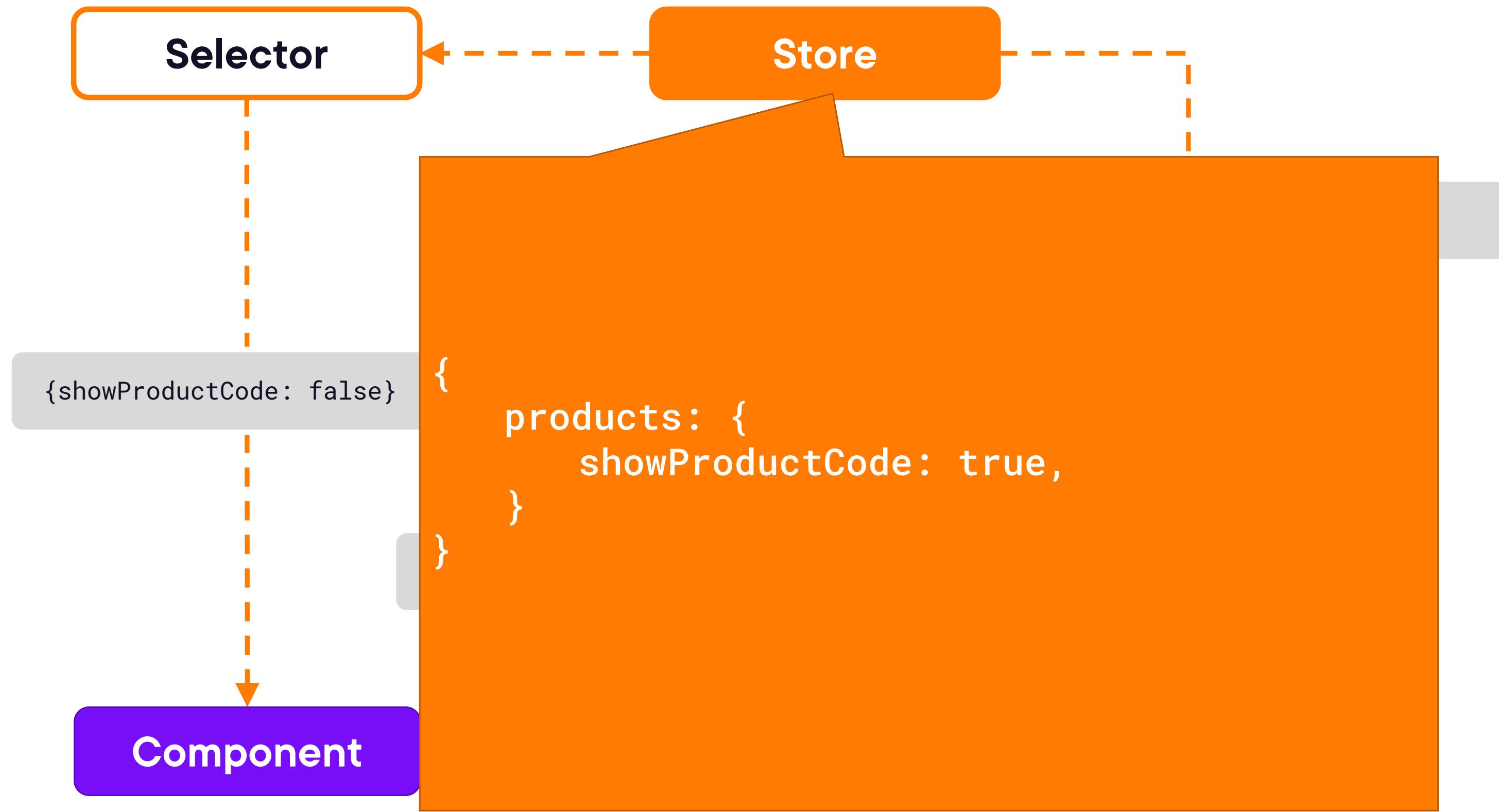
Course outline



State Lifecycle – Toggle Show Product Code



State Lifecycle – Toggle Show Product Code



NgRx Store Implements The Redux Pattern

There is only one single source of truth for application state called the store

State is read-only and the only way to change state is to dispatch an action

Changes to the store are made using pure functions called reducers



Use NgRx To Manage State When...

Your application has a lot of user interactions

Your application has a lot of different data sources

Managing state in services is no longer sufficient



The SHARI Principle

Shared: state that is accessed by many components and services.

Hydrated: state that is persisted and rehydrated from external storage.

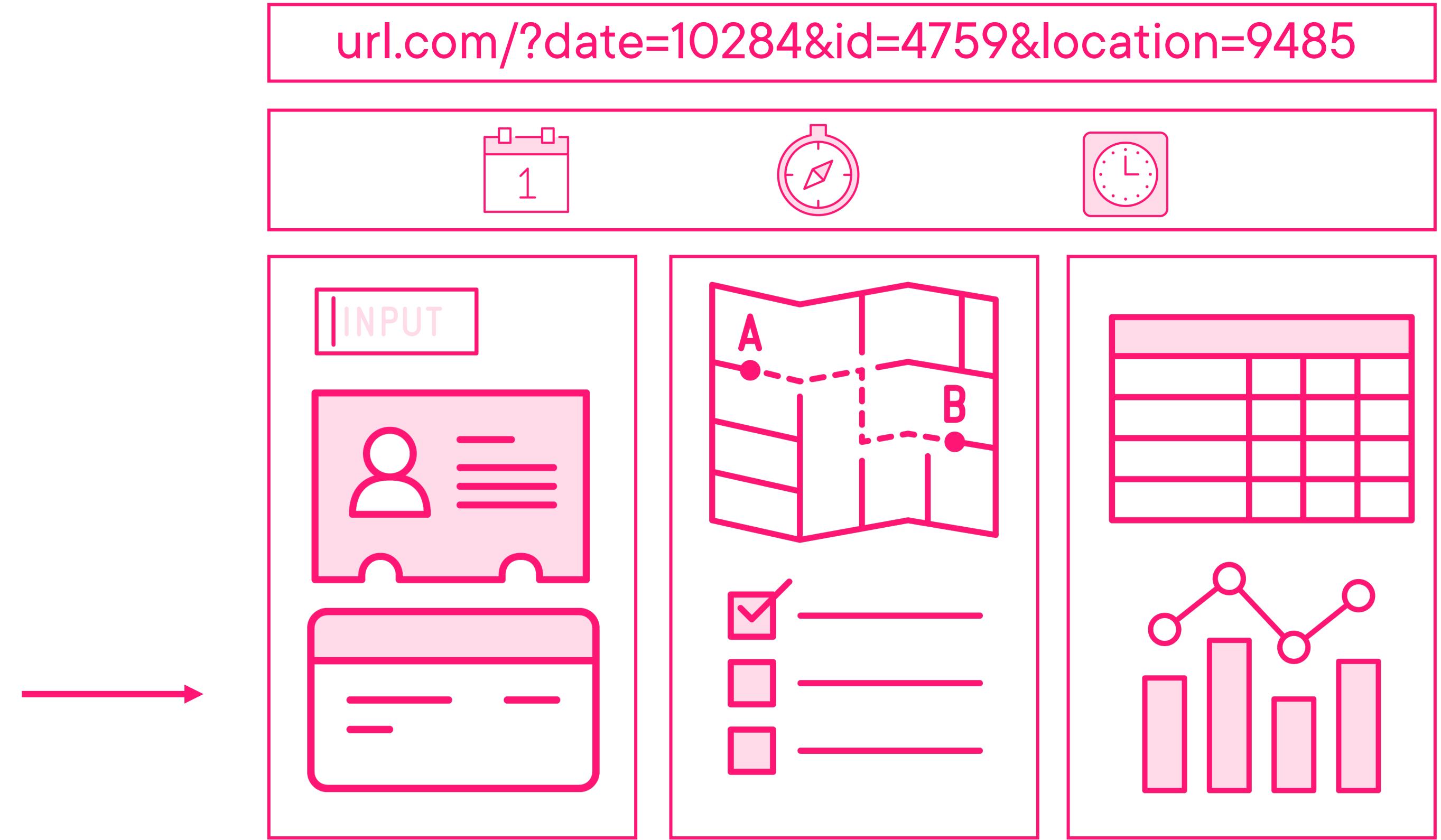
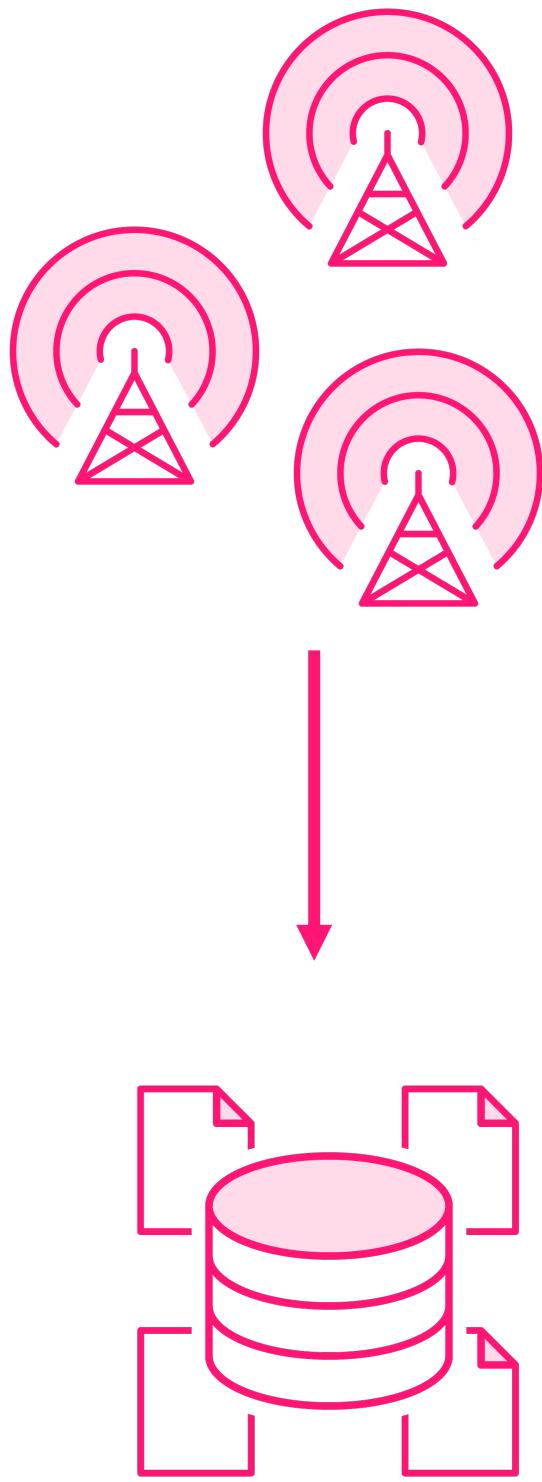
Available: state needs to be available when re-entering routes.

Retrieved: state that must be retrieved with a side effect.

Impacted: state that is impacted by actions from other sources.



Example NgRx Application



Key Concepts of NgRx

Type safety

Immutability and performance

Encapsulation

Serializable

Testable



Don't Use NgRx When...

Your team is new to Angular, RxJS Observables, and Redux

Your application is simple with little shared state

You need the shortest or quickest path to write an app



Prerequisites

Required	Suggested	Not Required
Angular modules, components, services, and RxJS observables	Angular Fundamentals Angular with RxJS Fundamentals	NgRx Experience



Discussion, Questions, and Feedback

[Table of contents](#) [Description](#) [Transcript](#) [Exercise files](#) [Discussion](#) [Learning Check](#) [Related Courses](#)

@dunchunter



Product Solutions Open Source Pricing

Search / Sign in Sign up

duncanhunter / pluralsight-ngrx-fundamentals Public

Notifications Fork 0 Star 0

Code Issues Pull requests Actions Projects Security Insights

master 1 branch 0 tags

Go to file Code About

duncanhunter 0604b99 2 weeks ago 14 commits

demo-0 2 weeks ago

demo-1-first-look 2 weeks ago

demo-2-developer-tools 2 weeks ago

demo-3-action-creators 2 weeks ago

demo-4-selectors 2 weeks ago

demo-5-effects 2 weeks ago

Readme MIT license 0 stars 1 watching 0 forks

Releases

<https://github.com/duncanhunter/pluralsight-ngrx-fundamentals>

NgRx Fundamentals[home](#) products

Pluralsight Angular NgRx Fundamentals

Duncan Hunter @dunchunter

Products

[Add](#)

Hammer
\$3.00

[Edit](#)

Nails
\$2.00

[Edit](#)

Spanner
\$4.00

[Edit](#)

Total: \$9.00

 Show Product Code

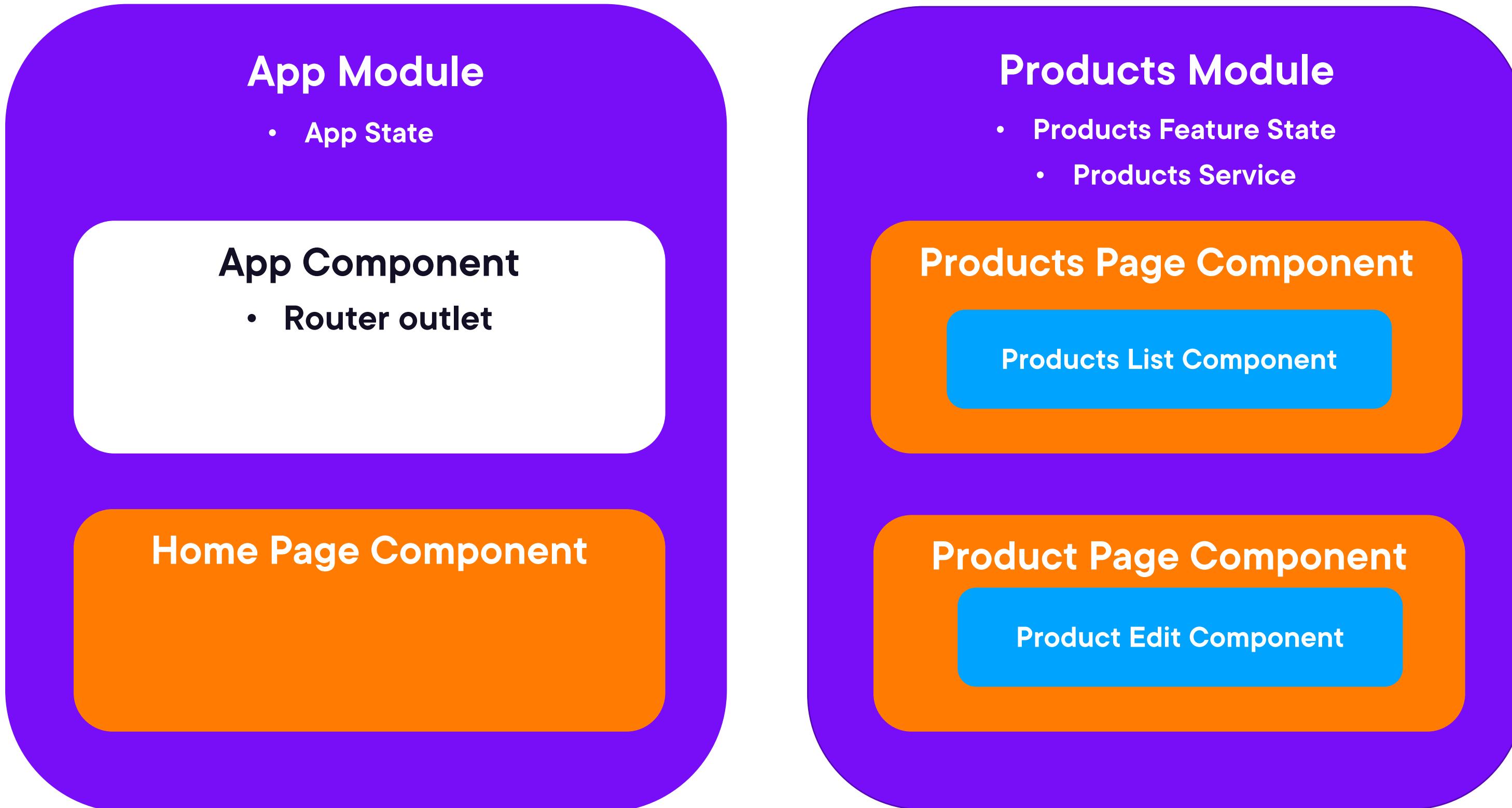
Product Detail

Name: Hammer

Price: 3

SaveDeleteCancel

Demo App Architecture





- demo-0
 - .angular
 - dist
 - node_modules
 - src
 - .editorconfig
 - .gitignore
 - angular.json
 - package-lock.json
 - package.json
 - tsconfig.app.json
 - tsconfig.json
 - demo-1-first-look
 - demo-2-developer-tools
 - demo-3-action-creators
 - demo-4-selectors
 - demo-5-effects
 - demo-6-router-store
 - demo-7-entity-adapter
 - demo-8-component-store
- CHANGELOG.md

Show All Commands

Go to File

Find in Files

Start Debugging

Toggle Terminal

Course Outline



First Look at NgRx

Developer Tools and Debugging

Strongly Typing Actions with Action Creators

Selecting State with Selectors

Working with Effects

Listen to Router State with Router Store

Manage Collections with Entity Adapter

Manage Local State with Component Store

Final Words

