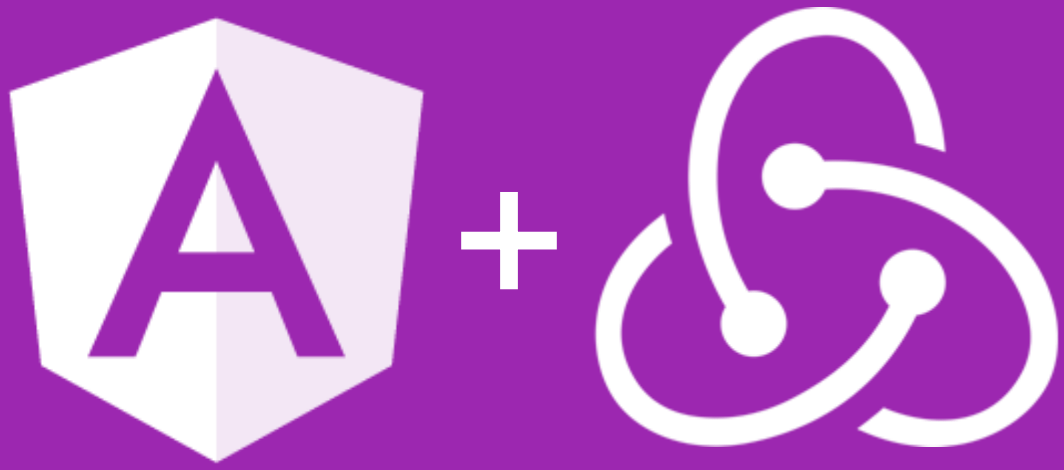


MANAGE STATE

Angular Redux

Christina Kayastha
Senior Software Engineer
Vistaprint, Cimpress
@christikaes

HELLO WORLD!

My name is Christina (:

Christina Kayastha
Senior Software Engineer
Vistaprint, Boston, MA
@christikaes

I'm all about:

- Icecream
- Community Events
- Bleeding Edge Technology

vistaprint®

1

BUILD FOR MOBILE

Progressive Web App

2

MANAGE STATE

Redux

3

USER INTERFACE

Material

4

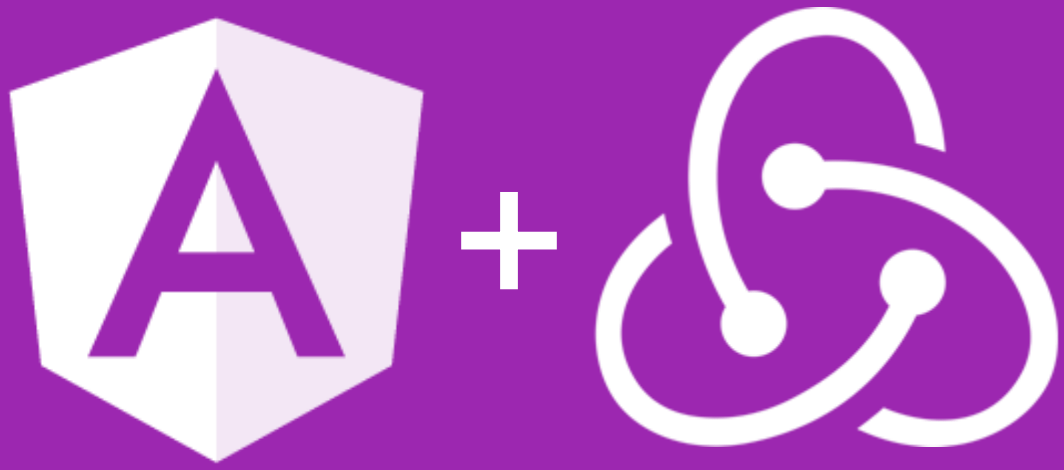
AUTH & DATABASE

Firebase

5

ANGULAR APP SPEED RUN!!

Everything



MANAGE STATE

Angular Redux

Christina Kayastha
Senior Software Engineer
Vistaprint, Cimpress
@christikaes

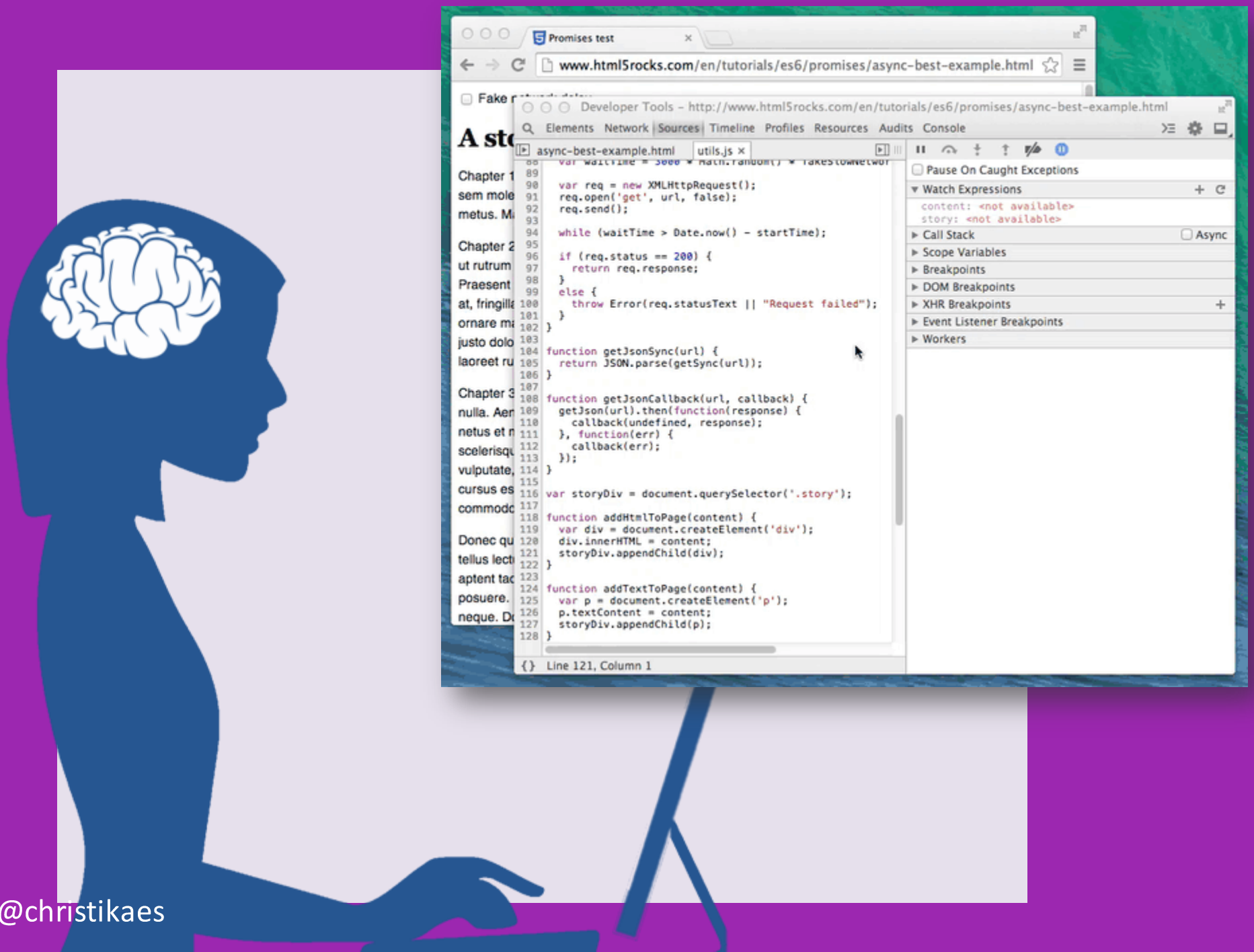
Angular Redux
WHAT is it ?
HOW can I use it ?
WHY should I care ?

What is STATE ?

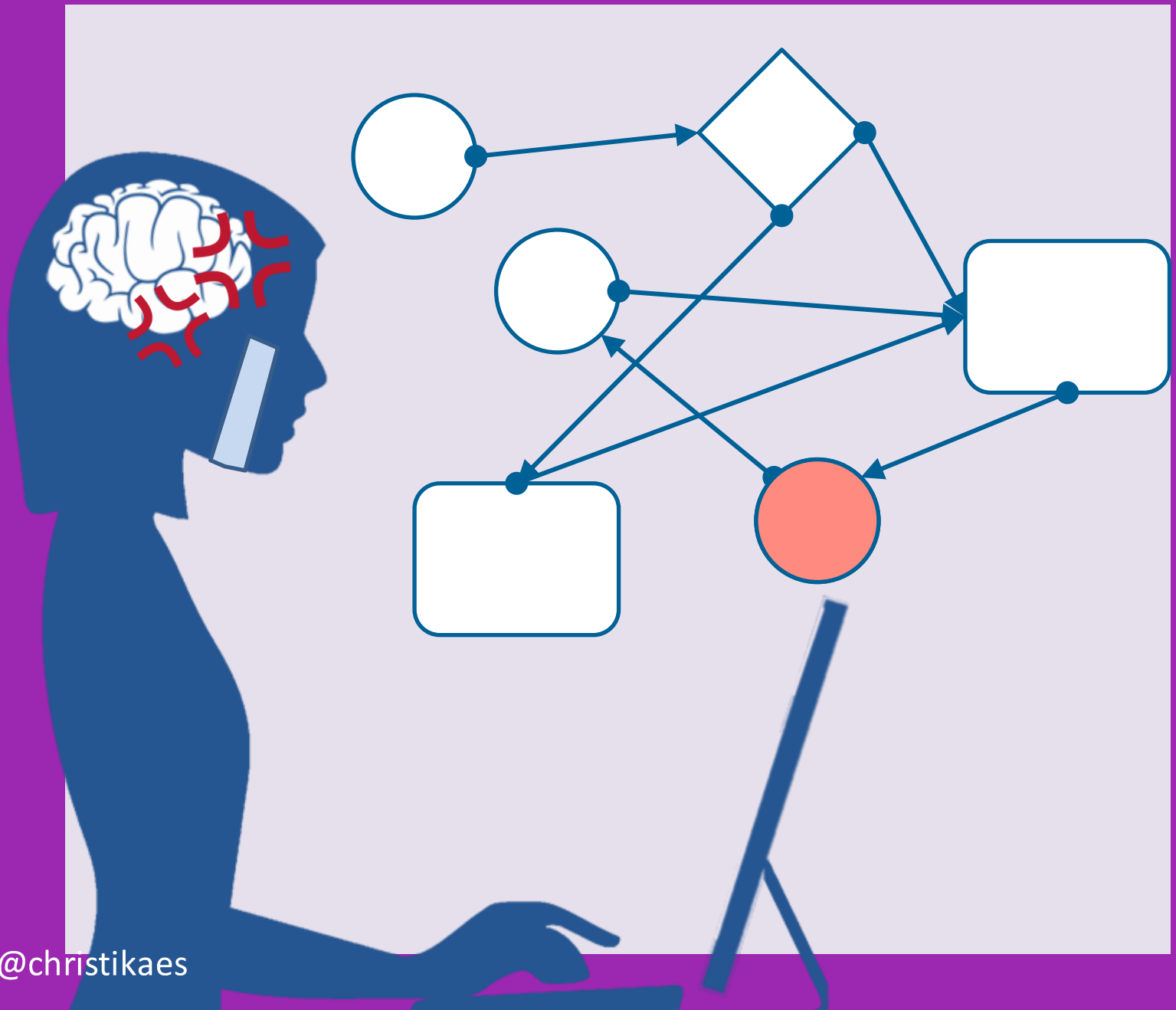
State is what an application knows about the user, interactions, and other pieces of global information

Why should I care about MANAGING STATE ?

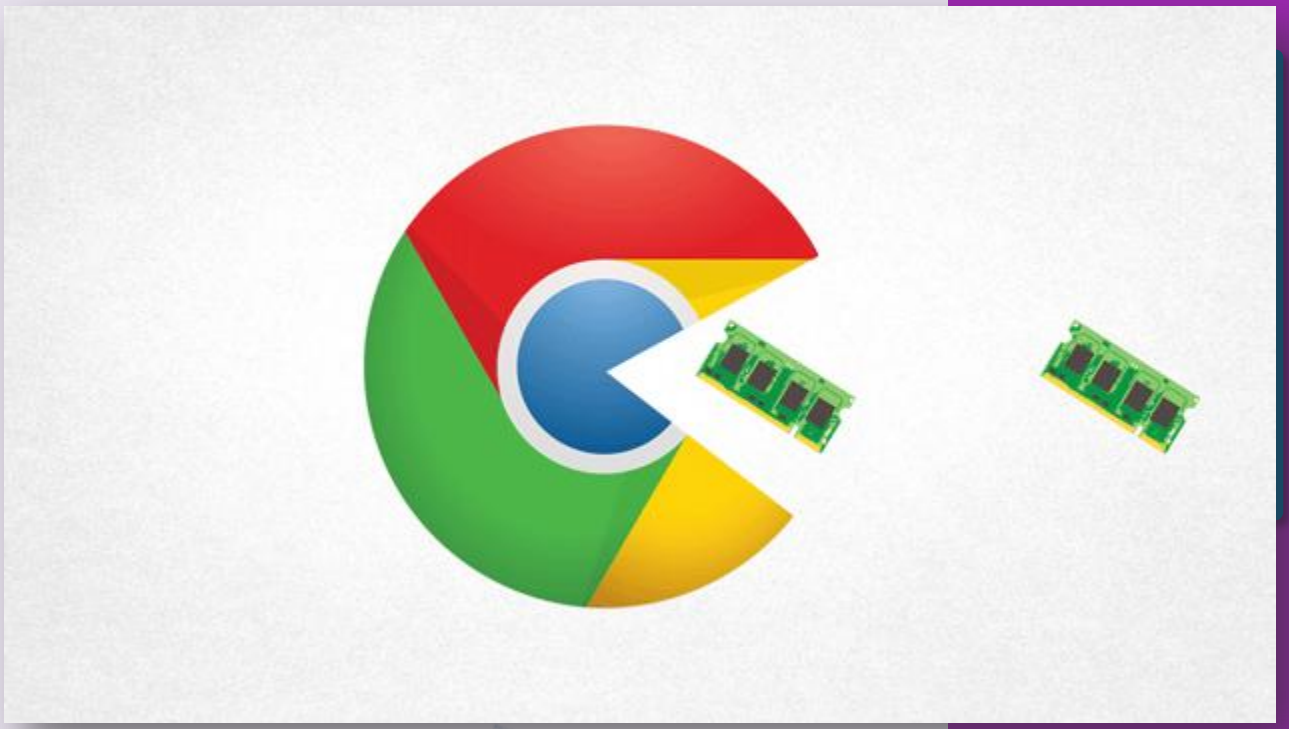

@christikaes



@christikaes

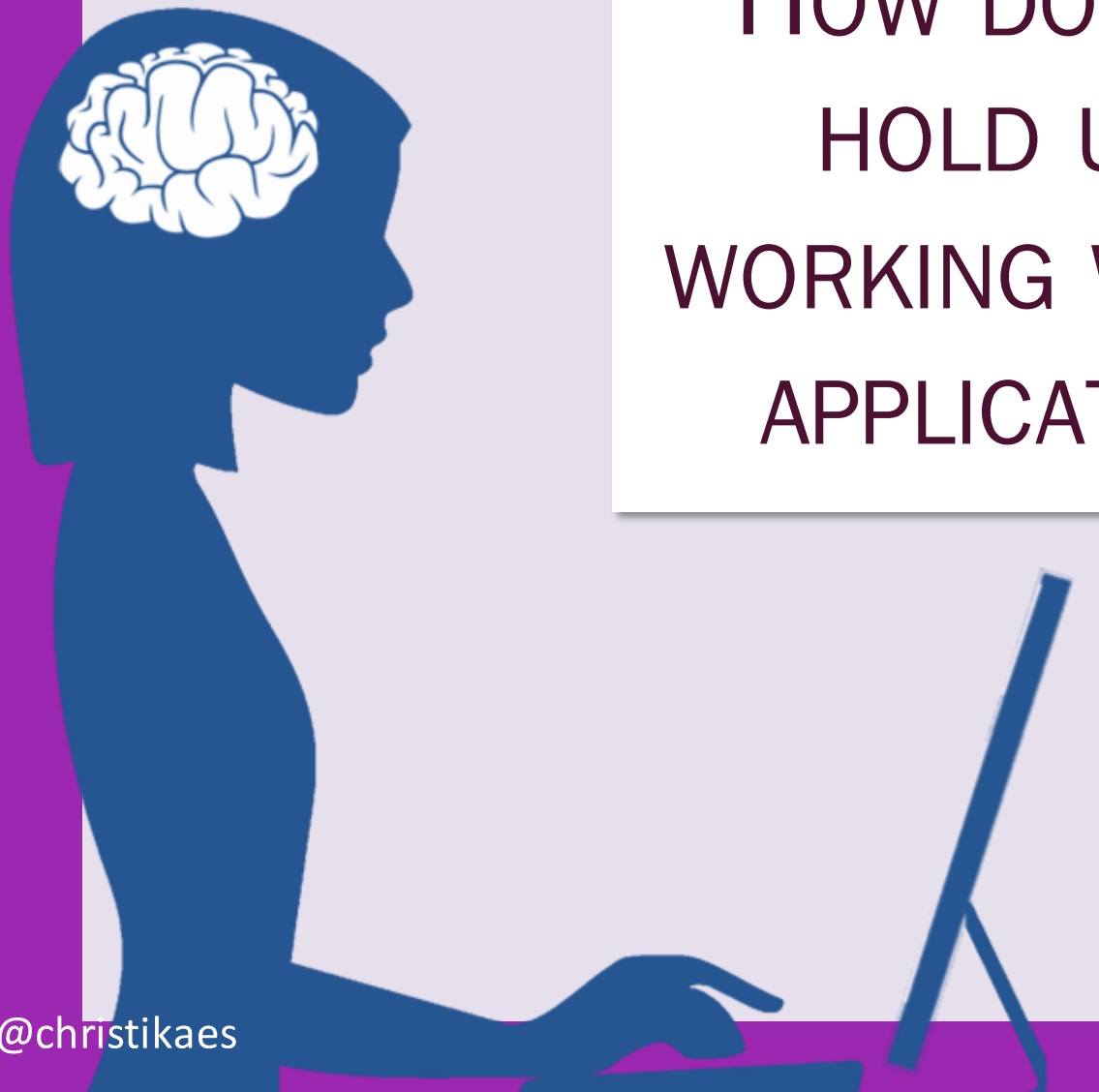


@christikaes



@christikaes

How do our brains hold up against working with complex application logic?



STAND BACK

I'M GOING TO TRY SCIENCE

@christikaes

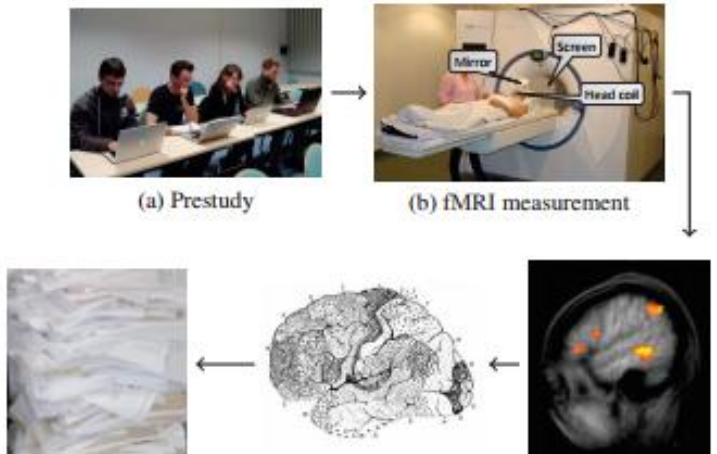
What parts of the Brain do Programmers activate?

Understanding Understanding Source Code with Functional Magnetic Resonance Imaging

Janet Siegmund[π], [*] Christian Kästner[ω], Sven Apel[π], Chris Parnin[β], Anja Bethmann[θ], Thomas Leich[δ], Gunter Saake[σ], and André Brechmann[θ]
[π]University of Passau, Germany [ω]Carnegie Mellon University, USA
[β]Georgia Institute of Technology, USA [θ]Leibniz Inst. for Neurobiology Magdeburg, Germany
[δ]Metop Research Institute, Magdeburg, Germany [σ]University of Magdeburg, Germany

ABSTRACT

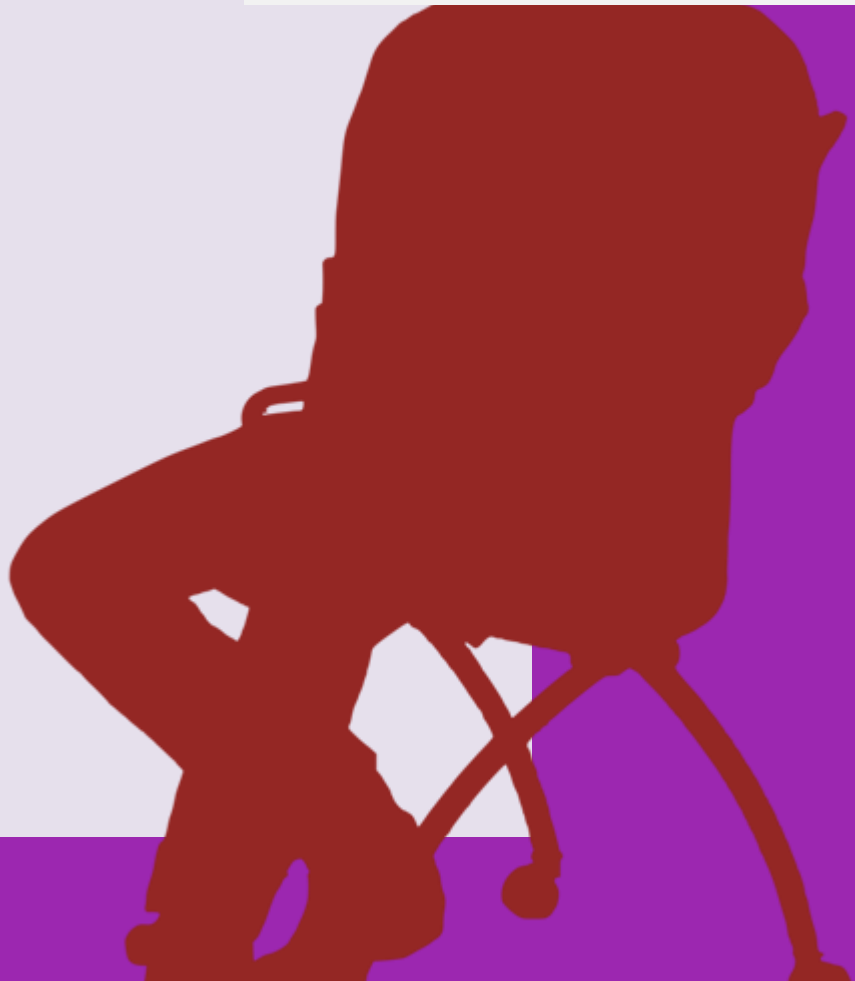
Program comprehension is an important cognitive process that inherently eludes direct measurement. Thus, researchers are struggling with providing suitable programming languages, tools, or coding conventions to support developers in their everyday work. In this paper, we explore whether *functional magnetic resonance imaging (fMRI)*, which is well established in cognitive neuroscience, is feasible to soundly measure program comprehension. In a controlled experiment, we observed 17 participants inside an fMRI scanner while they were comprehending short source-code snippets, which we contrasted with locating syntax errors. We found a clear, distinct activation pattern of five brain regions, which are related to working memory, attention, and language processing—all processes that fit well to our understanding of program comprehension. Our results encourage us and, hopefully, other researchers to use fMRI in future studies to measure program comprehension and,



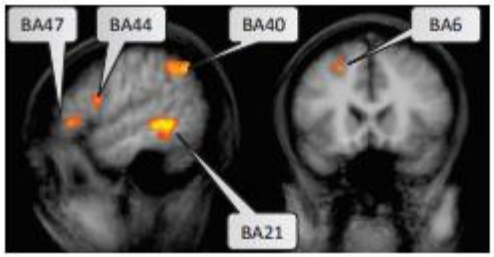
(a) Prestudy (b) fMRI measurement

@ch

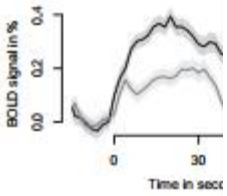

@christikaes



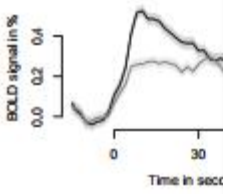
@christikaes



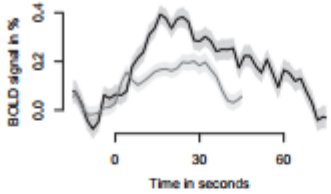
BA 6: Middle frontal gyrus
(Talairach coord.: -26, 17, 52; cluster size: 1279)



BA 21: Middle temp
(Talairach coord.: -55, -3



BA 40: Inferior parietal lobule
(Talairach coord.: -51, -49, 41; cluster size: 3368)



Working memory
Verbal/numeric
Problem solving

BA 44: Inferior frontal gyrus
(Talairach coord.: -50, 11, 16; cluster size: 698)

ATTENTION
WORKING MEMORY
SEMANTIC MEMORY
LANGUAGE

@christikaes

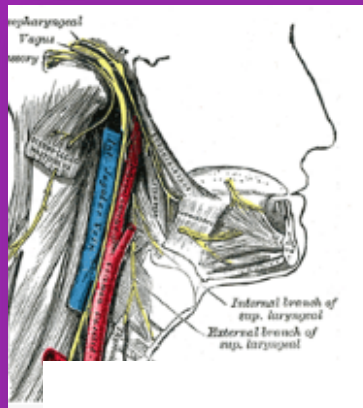
When do we use the most Attention/Working Memory?

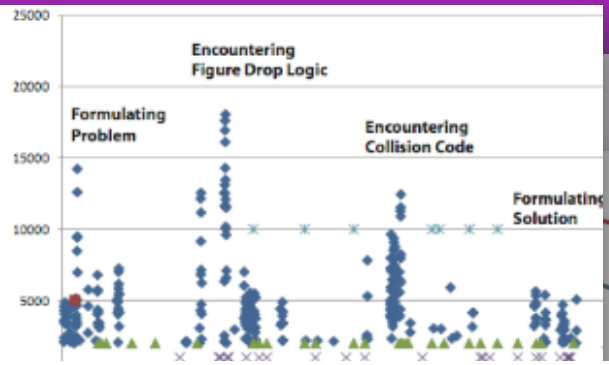
Understanding and Developing Models for Detecting and Differentiating Breakpoints during Interactive Tasks

Shamsi T. Iqbal and Brian P. Bailey
Department of Computer Science
University of Illinois
Urbana, IL 61801 USA
{siqbal, bpbailey}@cs.uiuc.edu

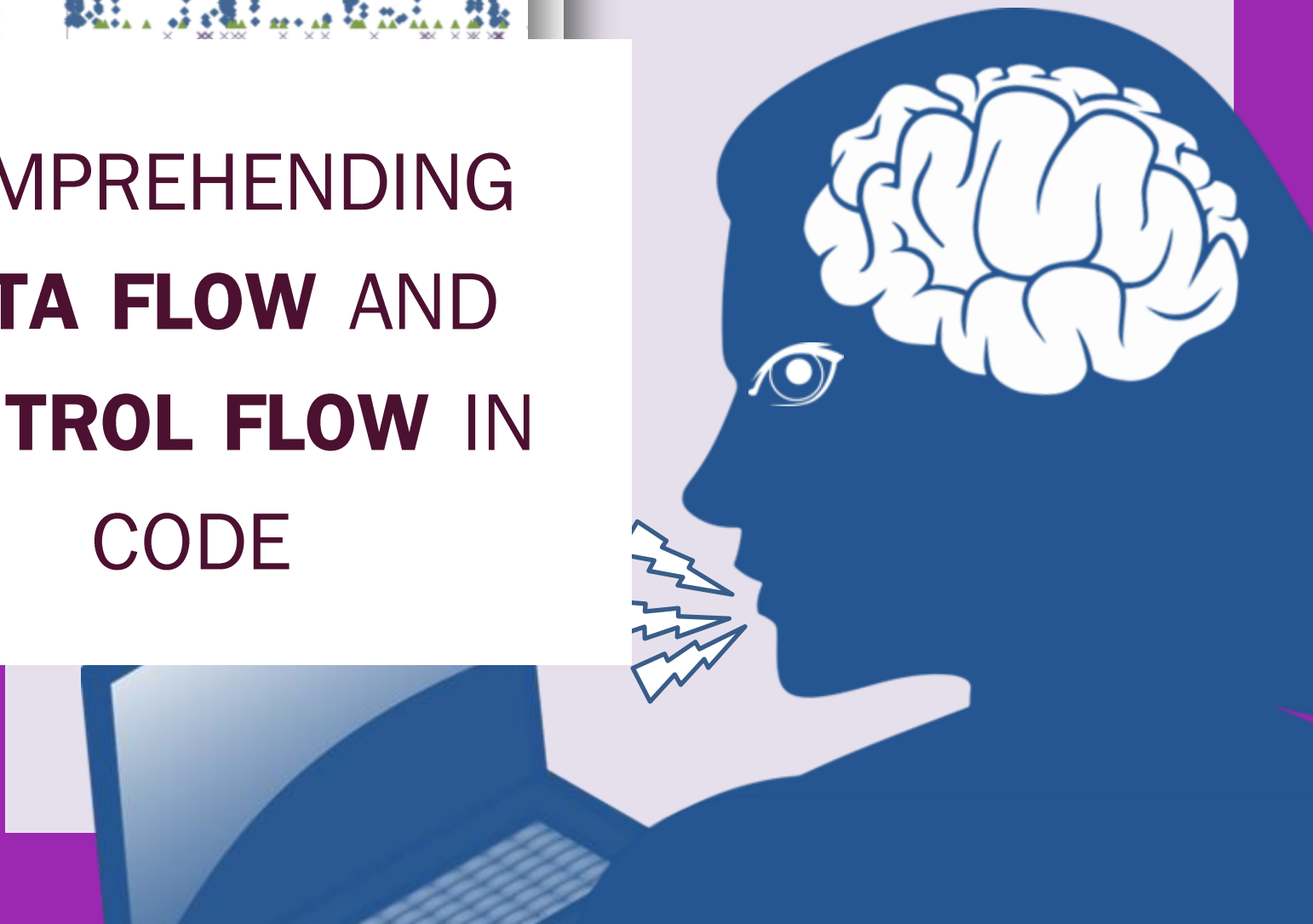
ABSTRACT
The ability to detect and differentiate breakpoints during task execution is critical for enabling defer-to-breakpoint policies within interruption management. In this work, we examine the feasibility of building statistical models that can detect and differentiate three granularities (types) of perceptually meaningful breakpoints during task execution, without having to recognize the underlying tasks. We

One common method for detecting breakpoints is to match users' ongoing interaction to specifications of tasks defined a priori [4]. Although this allows breakpoints to be easily detected within tasks that are fairly prescribed, it is much more difficult to leverage these types of static specifications to detect breakpoints within tasks that have highly variable interaction, i.e., *free-form* tasks, yet these are by far the most common type of computing task performed [8]. This





COMPREHENDING **DATA FLOW** AND **CONTROL FLOW** IN CODE



@christikaes

Complex application logic is taxing on our human brains!

WE LIVE IN A WORLD OF

INCREASINGLY COMPLEX

FRONTEND APPLICATIONS

@christikaes

How can we manage **increasingly complex** frontend applications

?

REDUX TO MANAGE **INCREASINGLY COMPLEX** FRONTEND APPLICATIONS

@christikaes

Managing State is makes our app much easier to Learn and Debug!

What does
REDUX
mean anyway?

Redux?

Brought back, revived

ReactFlux

Redux is a predictable state container for web applications

HOW do I use Redux?

Hello World

Redux Dev Tools

Undo/Redo

Syncing states

Hello Real World

What's next?

Hello World

Redux Dev Tools

Undo/Redo

Syncing states

Hello Real World

What's next?

github.com/christinakayastha/StudioReduxExample

Hello World+

Let's get started!

Redux is a **predictable state container** for Javascript Apps

```
(state, action) => newState
```

ACTION

REDUCER

STORE

```
(state, action) => newState
```

Action

Reducer

Store

All updates to the application state are caused by **actions**

JS Object that describes **What Happened?**

```
{
    type: “ADD_TODO”,
    payload: {
        text: “Learn Redux!”
    }
}
```

Best Practice: Use **Action Creators** to generate actions

```
addTodoAction = () =>
    return {
        type: “ADD_TODO”,
        payload: {
            text: “Learn Redux!”
        }
}
```

```
(state, action) => newState
```

Action

Reducer

Store

All updates to the application state are caused by **actions**

```
(state, action) => newState
```

Action | **Reducer** | Store

The **reducer** takes the current state and an action, and returns a new state

Design the State Shape

```
state = [
    {
        text: “Learn Redux”
    }
    …
]
```

Your entire state in 1 JS blob

A function that describes: What's the **new state**?

```
reducer = (state, action) => {
    switch(action.type) {
      case “ADD_TODO”:
        return [
            …state,
            action.payload
        ];
      default: return state;
    }
}
```

Best Practice: Use **CombineReducers**

```
reducer = combineReducers({
    reducer1,
    reducer2,
    …
})
```

```
(state, action) => newState
```

Action | **Reducer** | Store

The **reducer** takes the current state and an action, and returns a new state

```
(state, action) => newState
```

Action

Reducer

Store

THE **STORE** HOLDS THE STATE OF THE APPLICATION, IT CALLS THE REDUCER WHEN ACTIONS ARE DISPATCHED

Create a store with a reducer

```
store = Redux.createStore(reducer)
```

Get state through store

```
store.getState()
```

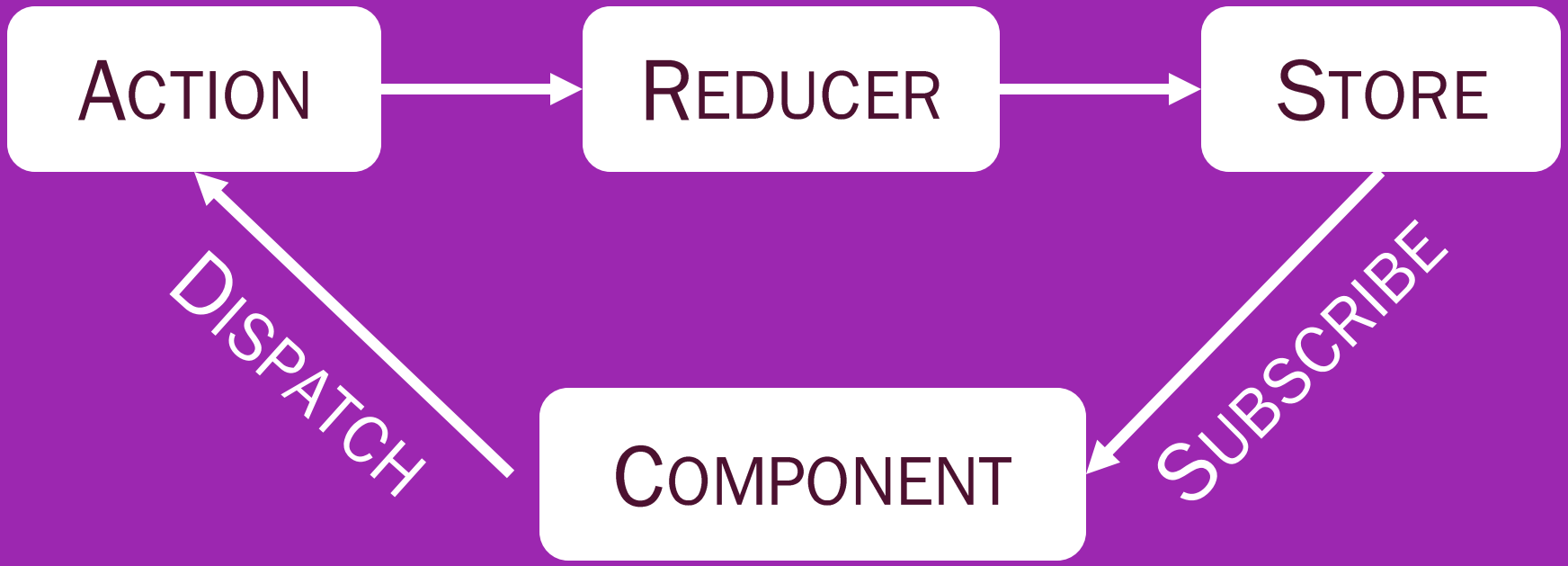
Dispatch actions through store

```
store.dispatch(addTodoAction())
```

Subscribe to changes on store

```
store.subscribe(() => {…})
```

```
(state, action) => newState
```



Hello World+

Let's get started!

Hello World

Redux Dev Tools

Undo/Redo

Syncing states

Hello Real World

What's next?

Hello World

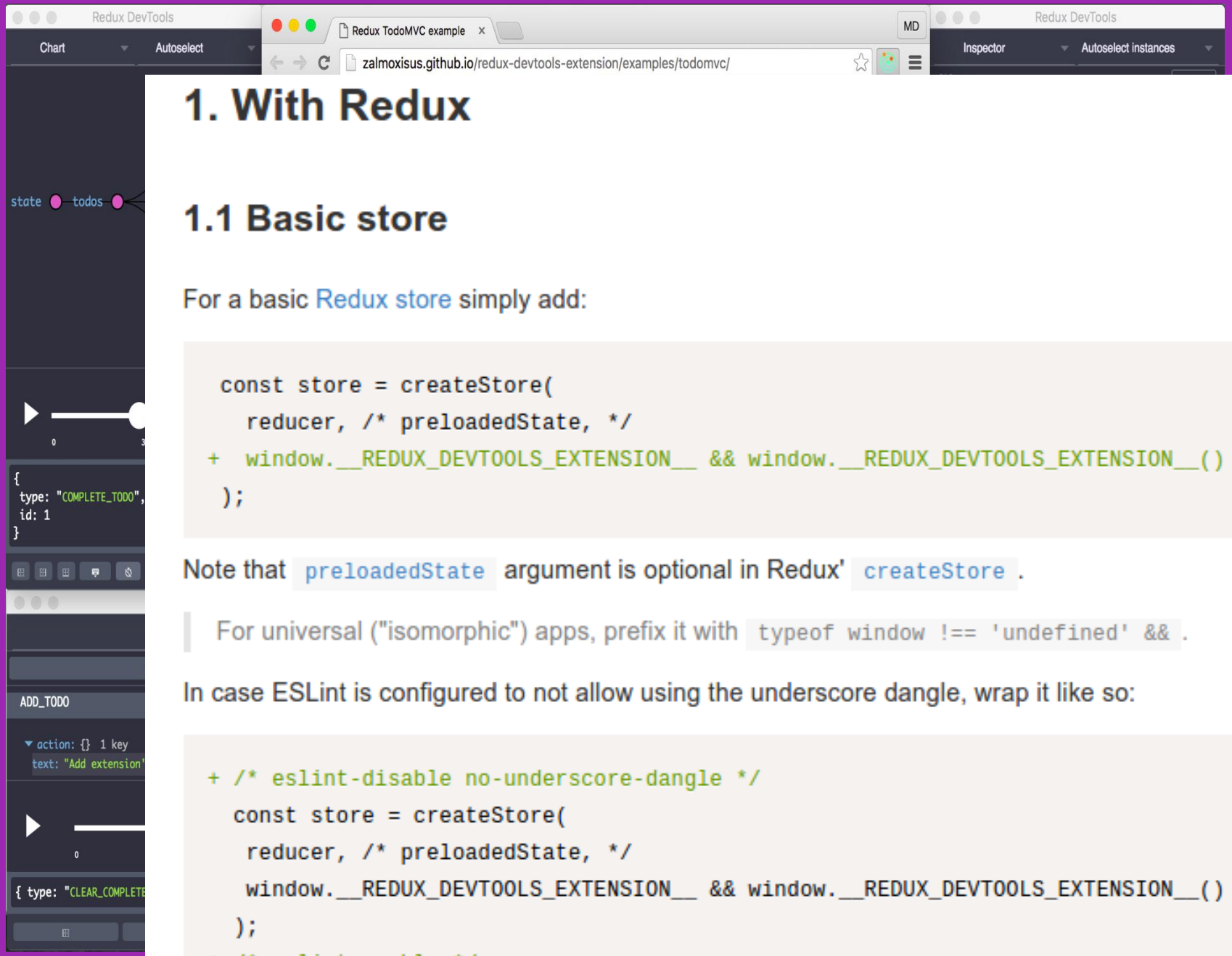
Redux Dev Tools

Undo/Redo

Syncing states

Hello Real World

What's next?


Redux DevTools
Chart
Autoselect
state
todos
{
type: "COMPLETE_TODO",
id: 1
}
ADD_TODO
action: {} 1 key
text: "Add extension"
{ type: "CLEAR_COMPLETE
Redux TodoMVC example
MD
zalmoxisus.github.io/redux-devtools-extension/examples/todomvc/
Redux DevTools
Inspector
Autoselect instances
1. With Redux
1.1 Basic store
For a basic Redux store simply add:
const store = createStore(
reducer, /* preloadedState, */
+ window.__REDUX_DEVTOOLS_EXTENSION__ && window.__REDUX_DEVTOOLS_EXTENSION__()
);
Note that preloadedState argument is optional in Redux' createStore .
For universal ("isomorphic") apps, prefix it with typeof window !== 'undefined' && .
In case ESLint is configured to not allow using the underscore dangle, wrap it like so:
+ /* eslint-disable no-underscore-dangle */
const store = createStore(
reducer, /* preloadedState, */
window.__REDUX_DEVTOOLS_EXTENSION__ && window.__REDUX_DEVTOOLS_EXTENSION__()
);

Hello World

Redux Dev Tools

Undo/Redo

Syncing states

Hello Real World

What's next?

Hello World

Redux Dev Tools

Undo/Redo

Syncing states

Hello Real World

What's next?

Undo/Redo

Reducer enhancers!

A Reducer Enhancer is simply a

Higher Order Reducer

redux undo/redo

npm v0.6.1 | build passing | dependencies up to date | code style standard | tips $0.00/week

simple undo/redo functionality for redux state containers

Protip: You can use the **redux-undo-boilerplate** to quickly get started with `redux-undo`.

Note: Make sure to update your programs to the **latest History API**.

Installation

```
npm install --save redux-undo
```

Undo/Redo

Reducer enhancers!

Hello World

Redux Dev Tools

Undo/Redo

Syncing states

Hello Real World

What's next?

Hello World

Redux Dev Tools

Undo/Redo

Syncing states

Hello Real World

What's next?

Syncing state

Redux Middleware!

MIDDLEWARE IS AN **EXTENSION POINT** BETWEEN THE ACTION DISPATCH AND THE REDUCER

redux-localstorage-simple public

Save and load Redux state to and from LocalStorage. Supports Immutable.js data structures.

Installation

```
npm install --save redux-localstorage-simple
```

Usage Example (ES6 code)

```
import { applyMiddleware, createStore } from "redux"
import reducer from "./reducer"

// Import the necessary methods for saving and loading
import { save, load } from "redux-localstorage-simple"
```

Syncing state

Redux Middleware!

Hello World

Redux Dev Tools

Undo/Redo

Syncing states

Hello Real World

What's next?

Hello World

Redux Dev Tools

Undo/Redo

Syncing states

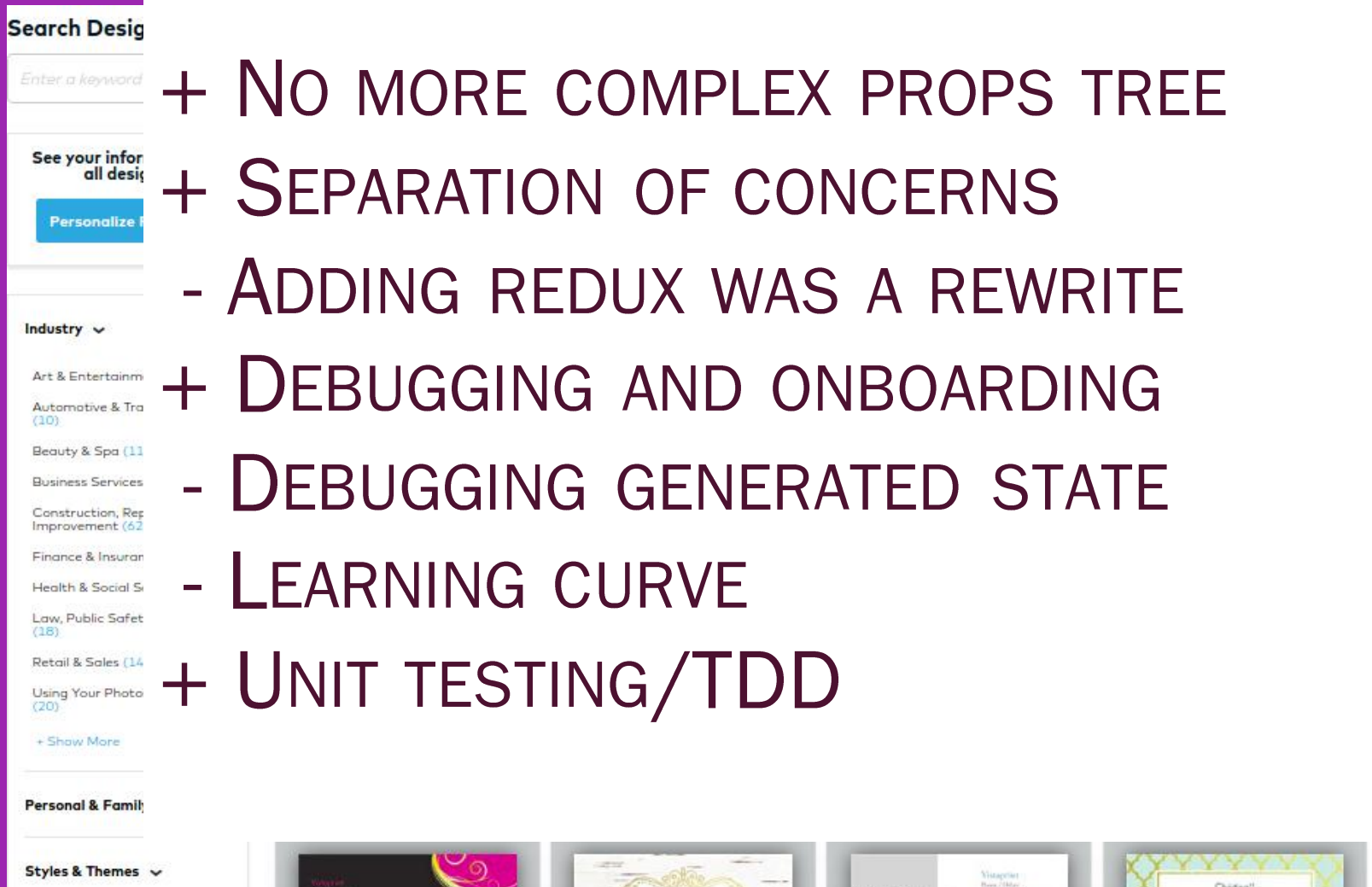
Hello Real World

What's next?

Hello Real World

Lessons Learned

Redux (with react) at Vistaprint



+ No more complex props tree
+ Separation of concerns
- Adding redux was a rewrite
+ Debugging and onboarding
- Debugging generated state
- Learning curve
+ Unit testing/TDD

Hello Real World

Lessons Learned

Hello World

Redux Dev Tools

Undo/Redo

Syncing states

Hello Real World

What's next?

Hello World

Redux Dev Tools

Undo/Redo

Syncing states

Hello Real World

What's next?

To use Redux, just remember the 3 parts:

1) **Store**: the entire state of your app
2) **Action**: description of what happened
3) **Reducer**: a function that creates a new modified store based on the action

WHAT is Angular Redux?

Angular Redux is an npm package that provides a series of Redux bindings for Angular

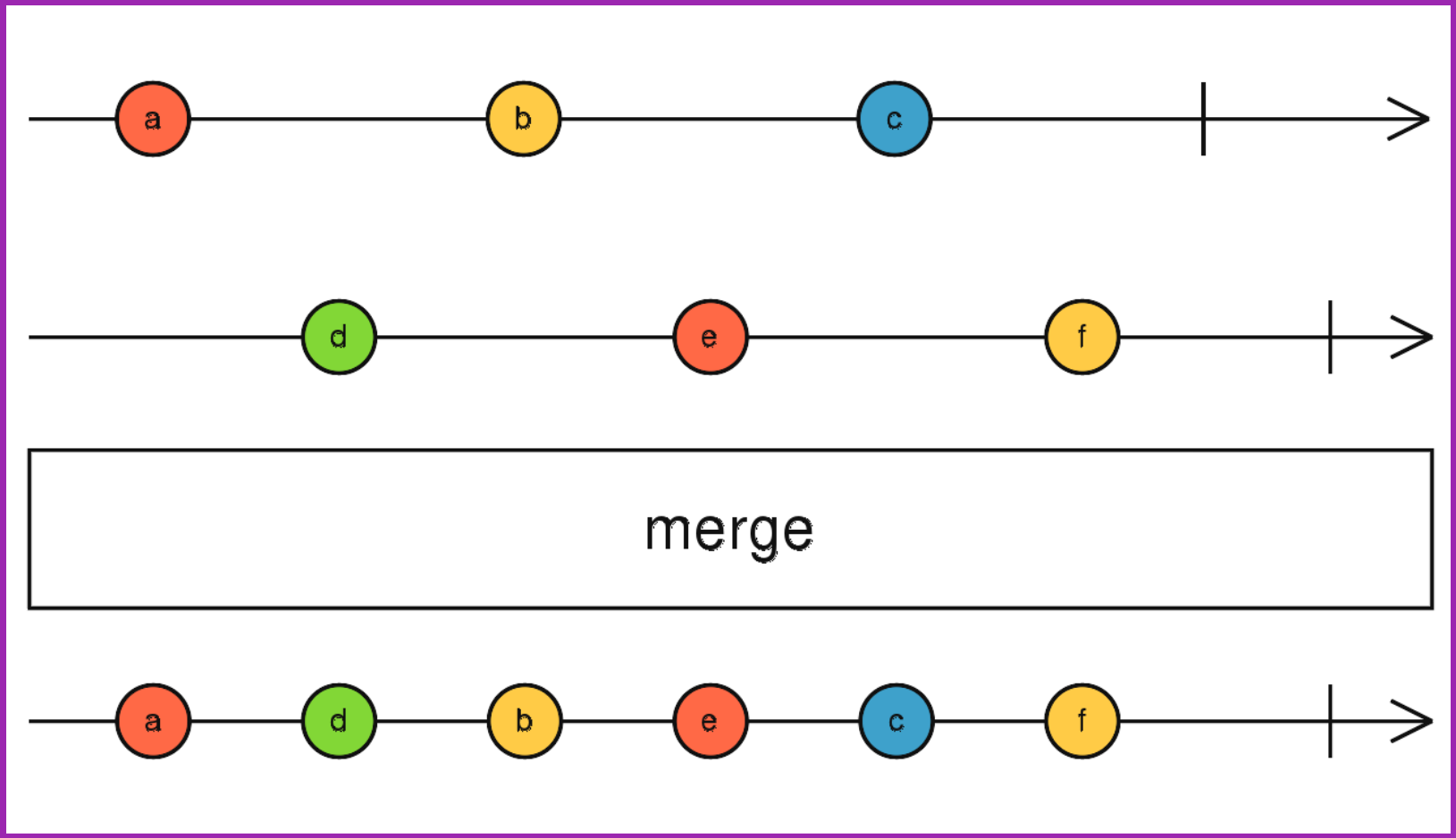
HOW do I use Angular Redux?

QUICK REVIEW

Observables

The Observable object represents a push based collection

The Observer and Observable interfaces provide a generalized mechanism for push-based notification, also known as the observer design pattern. The Observable object represents the object that sends notifications (the provider); the Observer object represents the class that receives them (the observer).


a
b
c
d
e
f
merge
a
d
b
e
c
f

```
1  const sourceOne = Rx.Observable.create(observer => {
2    observer.onNext(1);
3    observer.onNext(2);
4    observer.onNext(3);
5  })
6  sourceOne.subscribe(val => console.log('SourceOne:', val));
7
```

```
9  const sourceTwo = Rx.Observable.interval(2000);
10 sourceTwo.subscribe(val => console.log('SourceTwo:', val));
```

```
const {Component} = ng.core;
const {bootstrap} = ng.platform.browser;

@Component({
    selector: 'my-app',
    template: `
        <section>
          <h1>{{number$ | async}}</h1>
        </section>
    `
})
class AppComponent {
  constructor(){
    this.number$ = Rx.Observable.interval(1000);
  }
}

bootstrap(AppComponent);

```

HOW do I use Angular Redux?

Hello World

Redux Dev Tools

Undo/Redo

Syncing states

Hello Real World

What's next?

Hello World

Redux Dev Tools

Undo/Redo

Syncing states

Hello Real World

What's next?

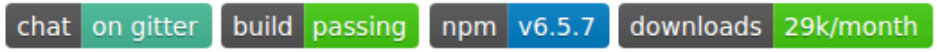
Hello World+

Let's get started!

@angular-redux/store

Angular bindings for Redux.

For Angular 1 see ng-redux



What is Redux?

Redux is a popular approach to managing state in applications. It emphasises:

- A single, immutable data store.
- One-way data flow.
- An approach to change based on pure functions and a stream of actions.

You can find lots of excellent documentation here: Redux.

What is @angular-redux?

We provide a set of npm packages that help you integrate your redux store into your Angular 2+ a
helps you by bridging the gap with some of Angular's advanced features, including:

Hello World

Redux Dev Tools

Undo/Redo

Syncing states

Hello Real World

What's next?

Hello World

Redux Dev Tools

Undo/Redo

Syncing states

Hello Real World

What's next?

Redux DevTools

Your best friend!

Hello World

Redux Dev Tools

Undo/Redo

Syncing states

Hello Real World

What's next?

Hello World

Redux Dev Tools

Undo/Redo

Syncing states

Hello Real World

What's next?

Syncing state

Redux Middleware!

Hello World

Redux Dev Tools

Undo/Redo

Syncing states

Hello Real World

What's next?

Hello World

Redux Dev Tools

Undo/Redo

Syncing states

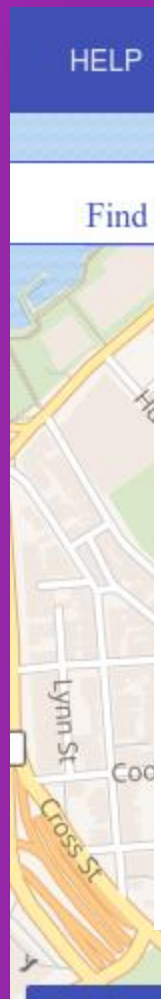
Hello Real World

What's next?

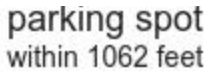
Hello Real World

Lessons Learned

Redux (with angular) at Parkabler



+ Logic is consolidated
 - Adding redux was a rewrite
+ Separation of concerns
+ Rapid iteration
+ Unit testing/TDD
+ Debugging
+ Data sync



Hello Real World

Lessons Learned

Hello World

Redux Dev Tools

Undo/Redo

Syncing states

Hello Real World

What's next?

Hello World

Redux Dev Tools

Undo/Redo

Syncing states

Hello Real World

What's next?

To use Angular Redux:

1) npm install @angular-redux/store
2) Import/Setup your app
3) Store, Action, Reducer

WHY should I use Angular Redux?

Angular Redux is awesome because:

- Redux to manage your state
- Redux tools work out of the box
- Works with RxJS Observable
- ...

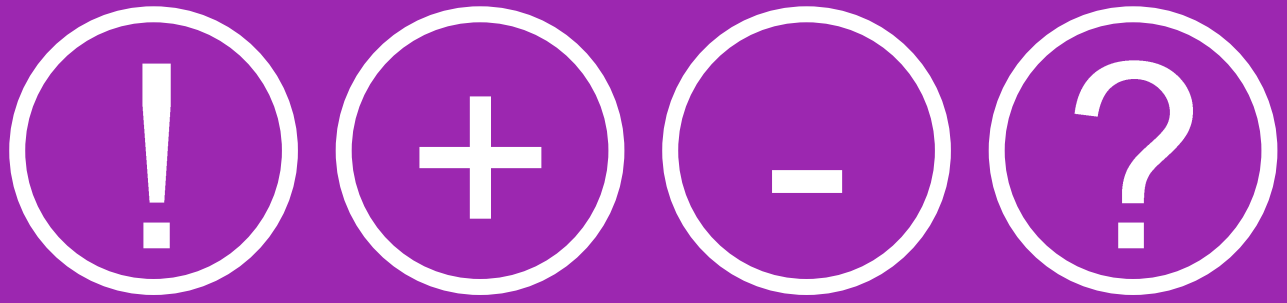
Angular Redux
WHAT is it ?
HOW can I use it ?
WHY should I care ?

Is Angular Redux READY to use in Prod?

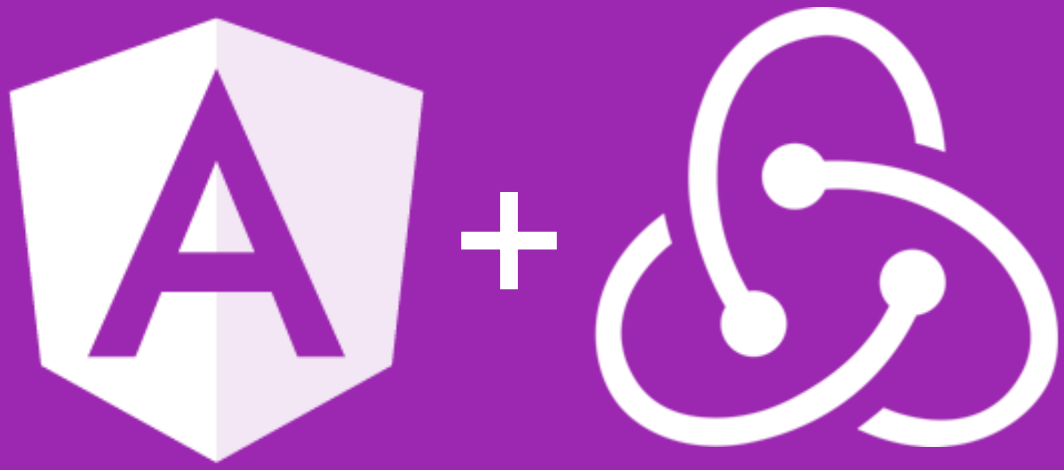
Angular Redux is in 6.2.1

- Not an officially supported package by Google, open-source and maintained by Rangle.IO
- https://github.com/angular-redux/store

THANK YOU!



@christikaes



MANAGE STATE

Angular Redux

Christina Kayastha
Senior Software Engineer
Vistaprint, Cimpress
@christikaes