



# **A Solid alternative to React**

## **SolidJS Overview**

**Luca Di Gianventura**

# About me

*Front End Developer @Accenture*

 @lutaok

 /in/lucadigianventura

 @lutaokdev



# Solid

A declarative, lightweight and fine-grained Javascript framework

Granularity



Signals, Reactive System

Easy to use



JSX, Familiar Apis, Components

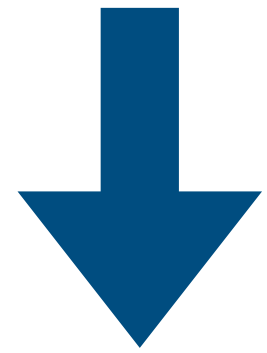
One stop Framework



Control Flow, Data Fetching, Stores

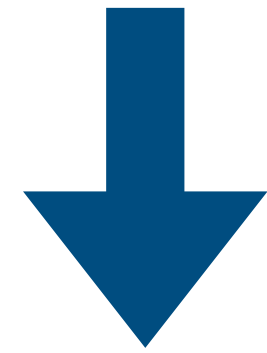
# React's Problems

Library Size



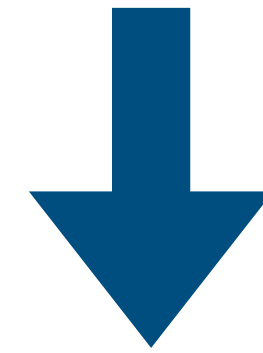
Preact

Data Fetching



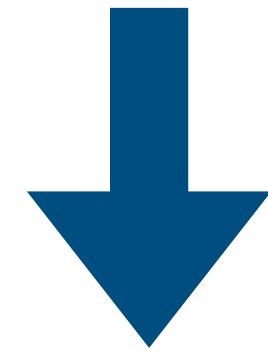
RTK Query  
Tanstack Query  
SWR

Global State Management



Redux Toolkit  
Zustand  
Jotai

Performance



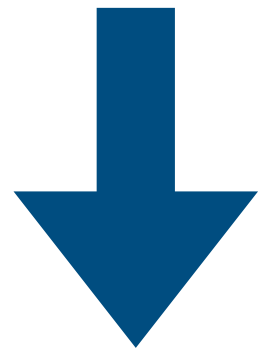
MillionJS/Preact

---

Common Solutions

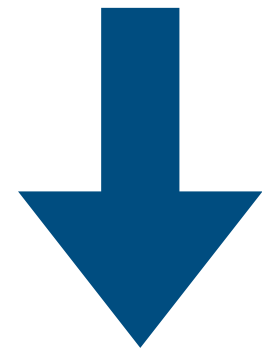
# Solid Solutions

Library Size



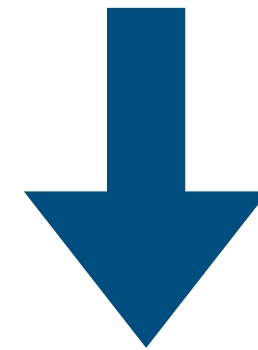
7kb gzipped

Data Fetching



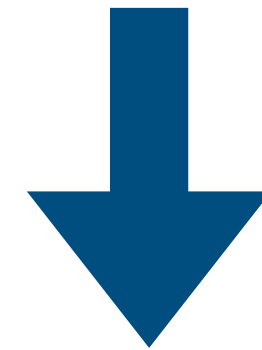
*createResource*

Global State Management



*createStore*

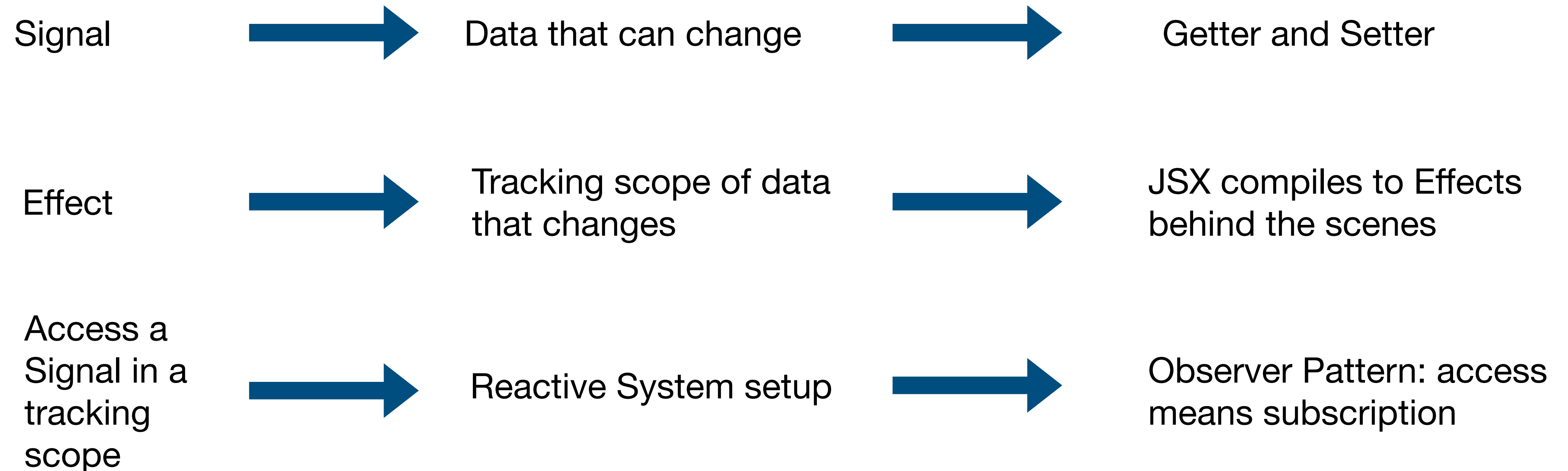
Performance



Signals

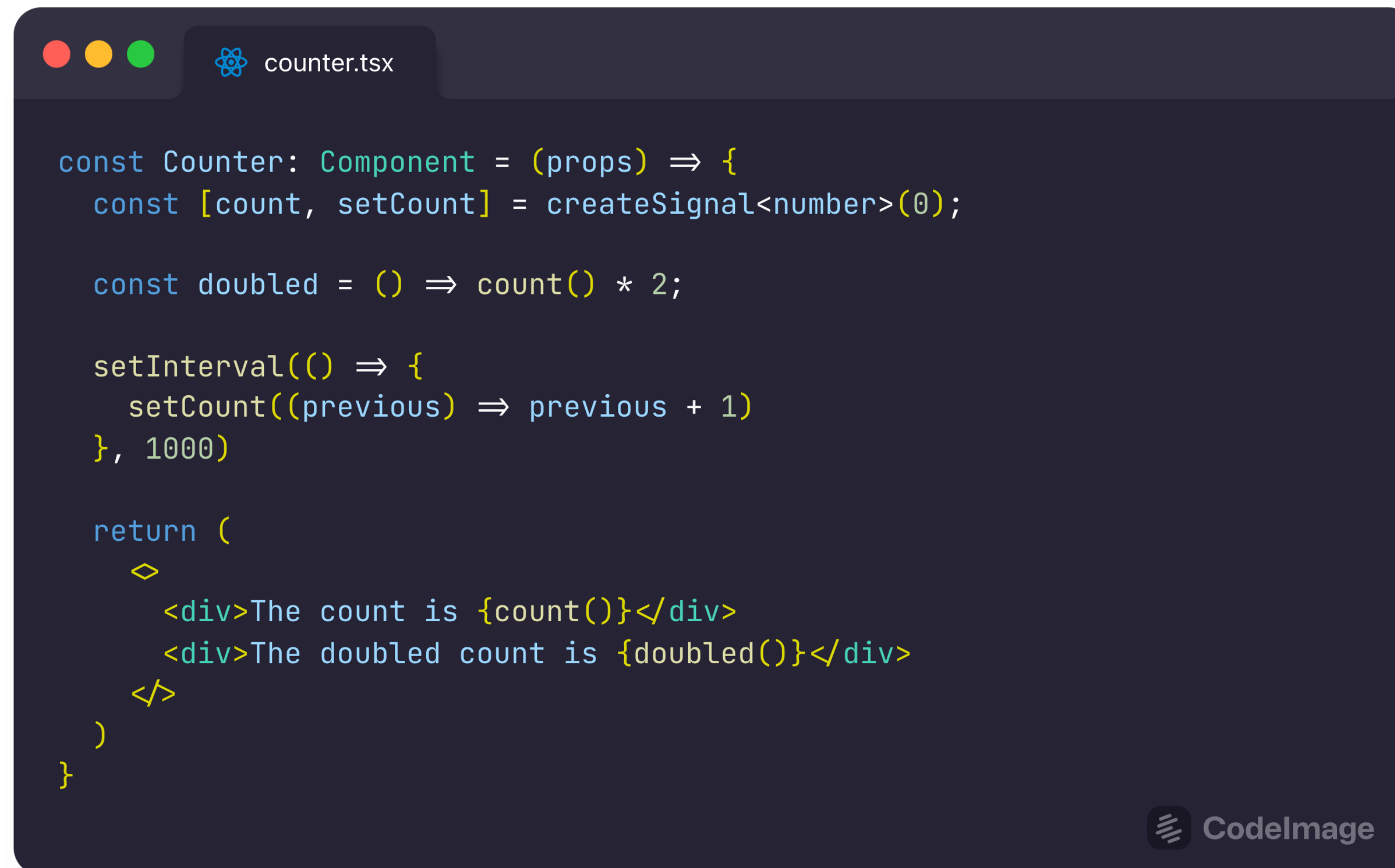
# Solid Signals

Reactive system based on Signals



# Why Solid?

*createSignal* - [Playground](#)



```
const Counter: Component = (props) => {
  const [count, setCount] = createSignal<number>(0);

  const doubled = () => count() * 2;

  setInterval(() => {
    setCount((previous) => previous + 1)
  }, 1000)

  return (
    <div>
      <div>The count is {count()}</div>
      <div>The doubled count is {doubled()}</div>
    </div>
  )
}
```

Fine-grained reactivity

- ➔ Similar API to React
- ➔ Component body runs only when mounted
- ➔ DOM updates directly where Signal changes
- ➔ Derived state must be a function to track changes



# Why Solid?

**createEffect** - [Playground](#)

Side Effects Tracking System

```
const Calculator: Component = (props) => {
  const [operand1, setOperand1] = createSignal<number>(0);
  const [operand2, setOperand2] = createSignal<number>(0);
  const [sum, setSum] = createSignal<number>(0);

  createEffect(() => {
    setSum(operand1() + operand2());
  });

  return (
    <div>{operand1()} + {operand2()} = {sum()}</div>
    <button onClick={() => setOperand1((prev) => prev + 1)}>Operand1 + 1</button>
    <button onClick={() => setOperand2((prev) => prev + 1)}>Operand2 + 1</button>
  )
};
```

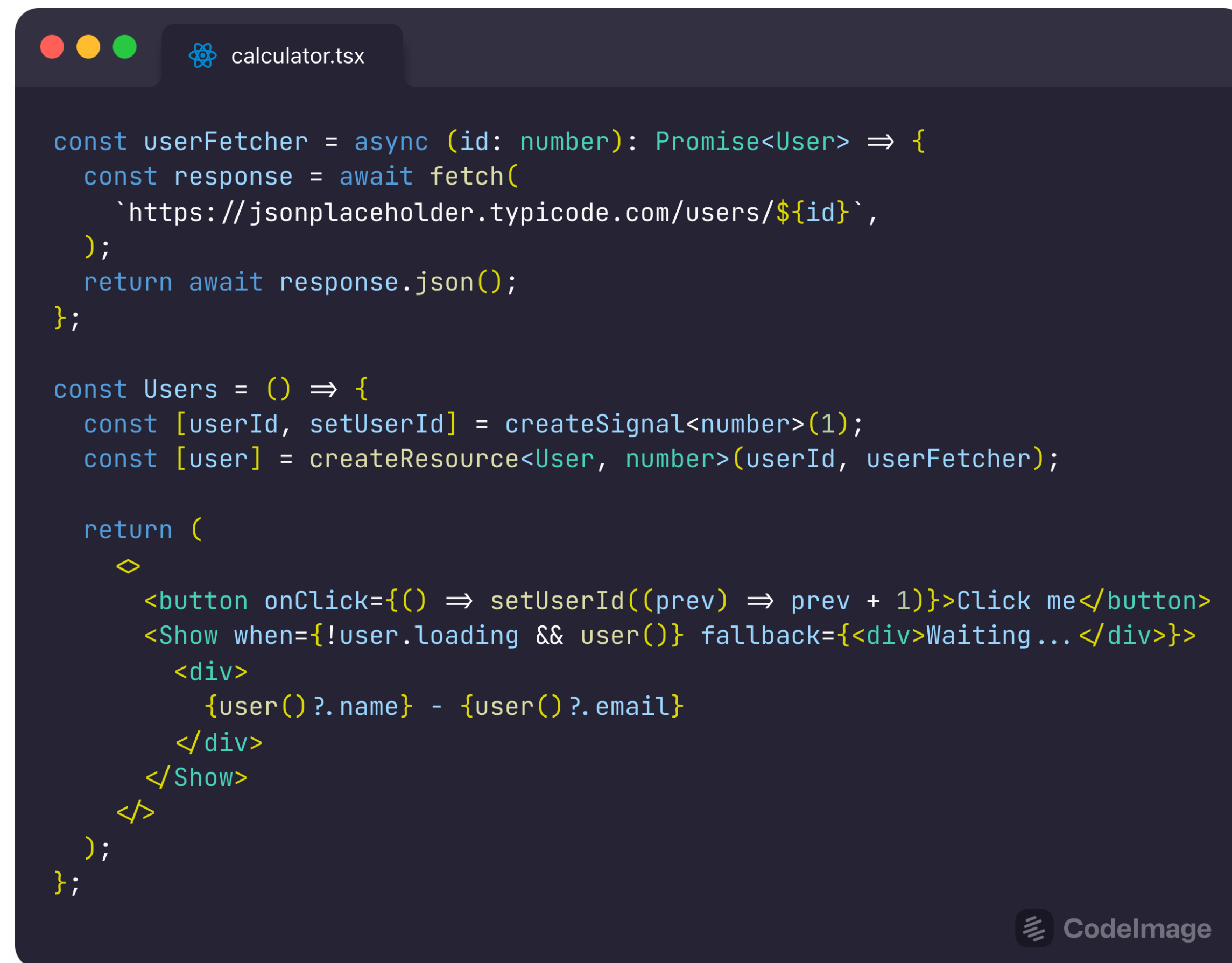
 CodeImage

- ➔ No manual dependency tracking
- ➔ Should be used to perform side effects
- ➔ Runs first when component is mounted
- ➔ Similar to `useLayoutEffect` in React



# Why Solid?

*createResource* - [Playground](#)



```
calculator.tsx

const userFetcher = async (id: number): Promise<User> => {
  const response = await fetch(
    `https://jsonplaceholder.typicode.com/users/${id}`,
  );
  return await response.json();
};

const Users = () => {
  const [userId, setUserId] = createSignal<number>(1);
  const [user] = createResource<User, number>(userId, userFetcher);

  return (
    <div>
      <button onClick={() => setUserId((prev) => prev + 1)}>Click me</button>
      <Show when={!user.loading && user()} fallback=<div>Waiting...</div>>
        <div>
          {user()?.name} - {user()?.email}
        </div>
      </Show>
    </div>
  );
};
```

Data Fetching

- ➔ Builtin API
- ➔ Augments “users” with request state properties
- ➔ Configurable and ssr-friendly
- ➔ Fetched data can be put into external storage

# Why Solid?

## Utilities

### Other Apis

- ✦ *createMemo*
- ✦ *createStore*
- ✦ *createContext/useContext*
- ✦ *mergeProps/splitProps*
- ✦ *onMount/onCleanup*
- ✦ *createSelector*

### Component Apis

- ✦ *createContext*
- ✦ *useContext*
- ✦ *lazy*
- ✦ *createUniqueld*

### Control Flow

- ✦ *Show*
- ✦ *For*
- ✦ *Index*
- ✦ *Suspense*
- ✦ *Portal*
- ✦ *Switch/Match*

# Development Status

## ***Solid - 1.8.x***

- ◆ Continuous improvements on primitives
- ◆ Hydration improvements
- ◆ Streaming improvements

## ***SolidStart (Metaframework)***

- ◆ Currently in Beta
- ◆ Actively developed
- ◆ Already used in production somewhere

## ***Integrations***

- ◆ Astro
- ◆ NativeScript (mobile)
- ◆ Tauri (desktop)

# Resources

- ✦ [Solid Original Docs](#)
- ✦ [Solid New Docs \(Incomplete\)](#)
- ✦ [Solid Playground](#)
- ✦ [SolidStart Docs](#)
- ✦ [Solid Github Organization](#)
- ✦ [Solid Discord](#)
- ✦ [CodeImage](#)

*Slide Repo*



# Thanks

 @lutaok

 /in/lucadigianventura

 @lutaokdev

