

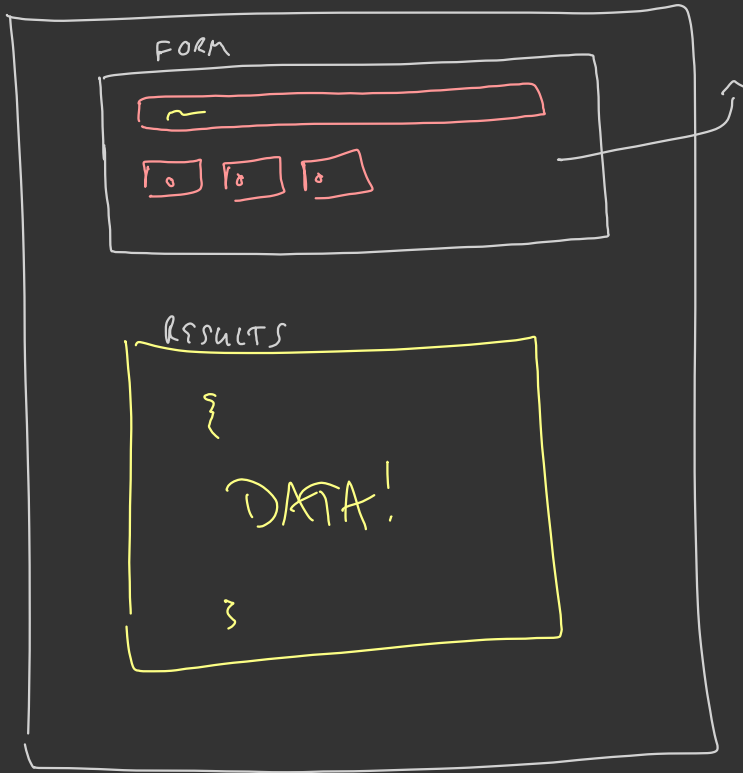
APPLICATION STATE

METHOD

URL

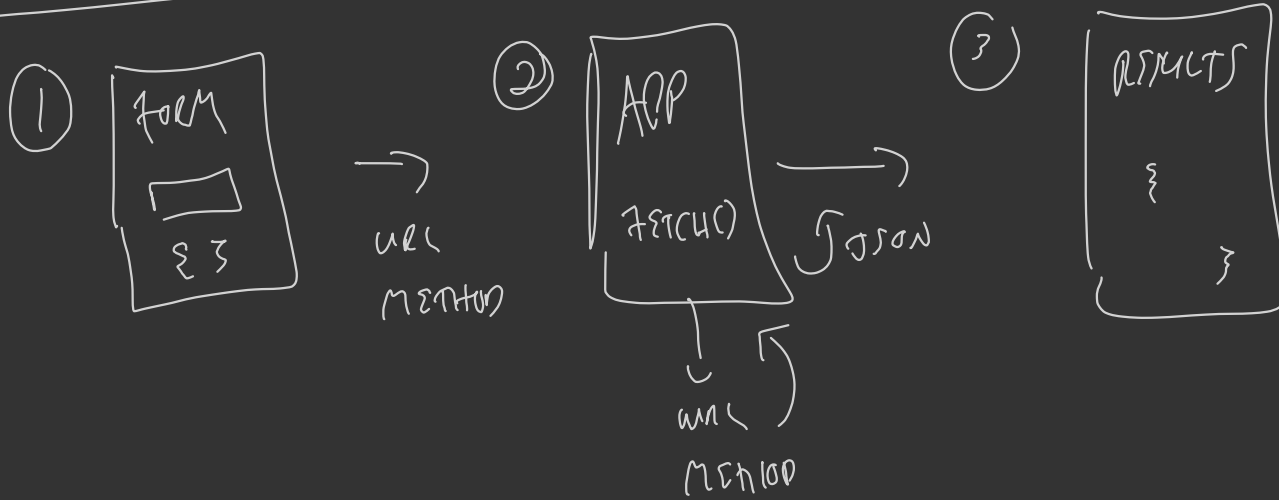
RESULTS

APP



- ① <FORM>
 - STATE : { }
 - CHANGE:
 - onChange()
 - onSubmit()
- ② <APP>
 - ↳ PROP()
 - ↳ FORM STATE
- ③ <APP>
 - FETCH w/ SA
 - UPDATE STATE
- ④ <RESULTS DATA="21">
PROP

USER/DATA flow



useEffect(cb)

useEffect () \Rightarrow {

// CODE

⋮

return () \Rightarrow { } // CLEAN-UP - OPTIONAL

} , [URL]);

CONDITIONS TO RUN

USE ANYTHING()

STATE

EFFECT

MEMO

CALCBACK

SET X()

REACT
STATE

52 MEM

LIFECYCLE -

- PRE-HOOK
- AFTER HOOK
- SET STATS
- GROW OR
- DISCOUNT

↓ FUNCTION DO STUFF (x) {

↓ ~~Let x = 0;~~

↓ while (x < 10) {

~~scribble~~

x ++;

← 10 x
the "N"

}

↓

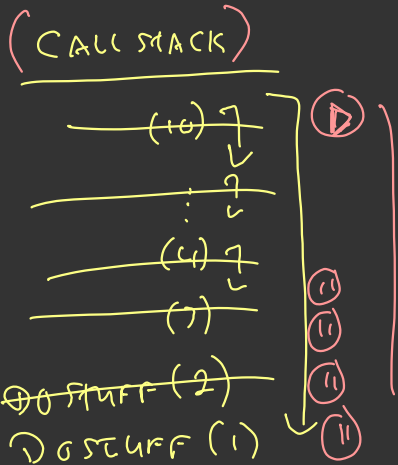
* RETURN x

}

DO STUFF(7)

FUNCTION DO STUFF(x) {
 if ($x \geq 10$) { return x }
 ~~~~~  
 return DO STUFF( $x + 1$ )  
 ~~~~~  
 }

~~~~~  
 return DO STUFF( $x + 1$ )  
 ~~~~~  
 }



TIME: Θ
 SPACE: N

DO STUFF(1) \rightarrow 10

const [NAME, SETNAME] = useState("x");

useEffect(() => {

// CODE...

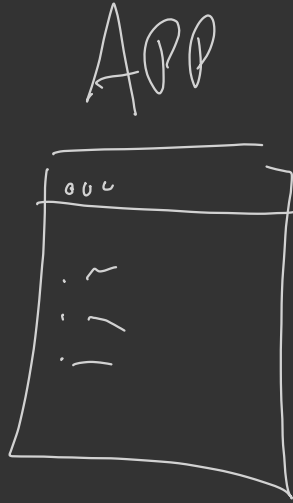
getSomeData(NAME);

}, [NAME, getSomeData]);

on('change:x', cb);

WATCHING: x, y, or z TO CHANGE

EXTERNAL DEPENDENCY



TEST



TAKE
API

API
}

→ use Reducer () ←

Let numbers = [1, 2, 3, 4]

current
"STATE"

Let squares = numbers.reduce ((acc, val, idx) => {

{ return [...acc, val * val];

} , []);

← NEXT STATE

payload

INITIAL STATE