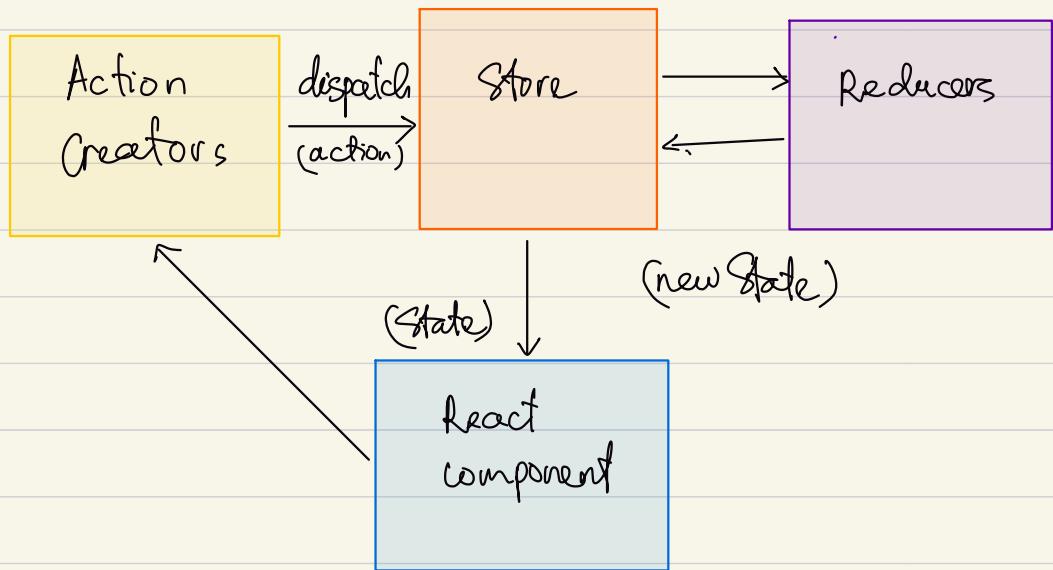


# React + Redux

(previousState, action)



# To do List

React

Add a task + display + save localStorage

1. Create an input + a button
  - input : onChange value
  - button: onClick handle submit

2. State:
  - list of jobs: [ ] an array
  - job: ' ' string

3.2. handleSubmit,

- list of jobs: [ ] an array  
prev + job  
+ )  
←
- job: ' ' string

- 3.1. Render a list with map((job, index) → { })

4. Lấy data vào browser : stringify

Có [ ] → chuyển thành JSON → lưu vào storage

5. Render data ra giao diện : parse.getItem → initial state

6. Optimise initial state

---

local storage

JSON

parse getItem

get data

type of data

[ ] {} " " number

←

stringify, setItem

lưu dữ liệu

# To do App

## use Reducer

### ADD + DELETE a task

1. Nhập tên công việc phải làm  
    Click Add → Render list  
    [ Enter keyboard ]
2. Bấm dấu X → xóa bỏ công việc đó
3. Sau khi thêm công việc thành công  
    → Bàn phím chuột và input  
    → Tự động focus lại input

4 bđc đe' n' dung use Reducer

1. Initial State
2. Actions
3. Reducer
4. Dispatch

## ① Initial State

job: '' string: dữ liệu của input nhập vào  
jobs, [ ] State update lai danh sách công việc  
gõ ↗ ^add

const initial State = {

job: ''

jobs, [ ]

}

## ② Actions

1. định nghĩa actions

input: SPT - JOB : update lại job

add: ADD - JOB :

x: DPL - JOB

### ③ Reducer

const reducer = (state, action) →  $\ddot{\delta} \ddot{\delta}$ .

### ④ Dispatch

const [state, dispatch] = useReducer  
(reducer, initial state)

const [jobs, setJobs] = state

### ⑤ Render

jobs.map: → list jobs

⑥ Go wā input → giv' they do':

input, onchange (<sup>x</sup>e. target. value)

dispatch

→ Take 3 actions thru damp functions

Worst set Job = payload → !  
return !

type : SET\_JOB, // set-job  
payload // wash dishes

} {

input: onChange ( dispatch ( setJob ( e. target. value ) ) )

⑦ Update lại job → input hiện chí?

switch case rỗng bù 3 : reducer  
switch (action. type)

case SET\_JOB :

return !

... state,

job: action. payload

}

default:

throw error

in step 3: log : action, Prev state, New state

⑧ **Add** → add vào job[]

btn : onclick : { handle submit }

input lives  
↑ fair

+ handle submit () : dispatch (addJob (job))

+ switch case ADD\_JOB :

new state = { ... state,

jobs: [... state.jobs ,

action . payload ]

}

break

⑨ Clear input sau khi Add

handle submit : dispatch (setJob (""))  
trả lại đầu vào

## ⑩ Focus input

useRef = inputRef

input: ref =  $\{ \text{inputRef} \}$

handle submit:  $\text{inputRef.current.focus()}$

## ⑪ Delete a task / job

Bt: Click to X  $\rightarrow$  from to 1 index

Help tip: xoá 1 thanh phím trong mang splice

+ onClick : dispatch ( deleteJob ( index ) )

+ switch case:  $\text{state.jobs} \rightarrow \text{job cei}$

const newJobs = [ ...state.jobs ]

newJobs.splice (action.payload, 1)

index  $\rightarrow$  element

newState = { ...state,

jobs: newJobs }