



UNIVERSITA' degli STUDI di ROMA
T O R V E R G A T A



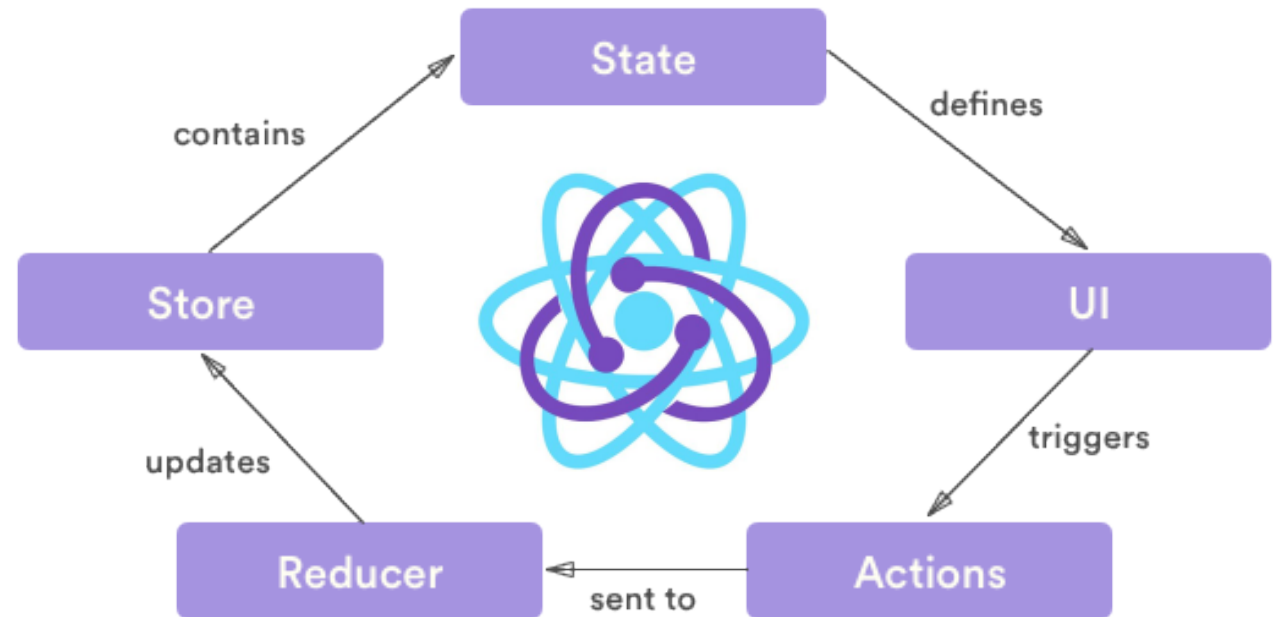
Presentazione finale ISSSR – Redux

Pelella, Tranzocchi, Trotta, Perrone, Pusceddu, Scarpitta, Scarlino

AGENDA

Redux

- Introduzione Redux
 - Benefici e limiti
- Reducers
 - Cambio dello stato dell'applicazione in risposta alle azioni
- Actions
 - Payload di informazioni inviate allo store
- Chiamate sincrone vs asincrone



Presentazione

Introduzione Redux

1

Singola fonte di verità

Lo stato dell'intera applicazione è memorizzato in un unico oggetto

2

Lo stato è in sola lettura

Non è possibile modificare direttamente le informazioni memorizzate nello stato dell'applicazione; l'unico modo per farlo è tramite azioni esplicite

3

Funzioni pure

Lo stato viene sostituito da un nuovo stato generato da funzioni esclusivamente in base ad una azione ed allo stato precedente



*Redux is a
predictable state
container for
JavaScript apps.*

Redux

Principali benefici



1

La predicibilità
delle transizioni
di stato

2

L'organizzazione
del codice
obbligatoria

3

Il Server
rendering

4

Il tracciamento
dello stato

5

La testabilità del
codice

ACTION

- Action creator

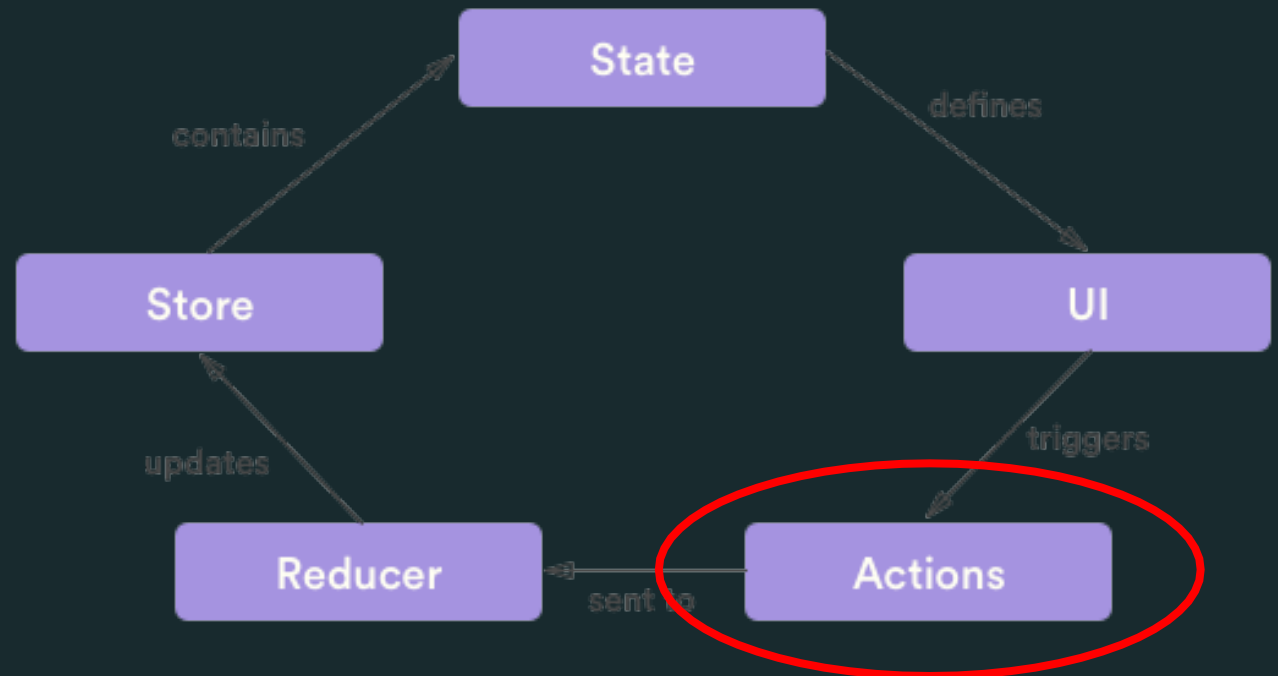
```
function toggleTodo(index) {  
  return {  
    type: TOGGLE_TODO,  
    index  
  }  
}
```

- Payload representation

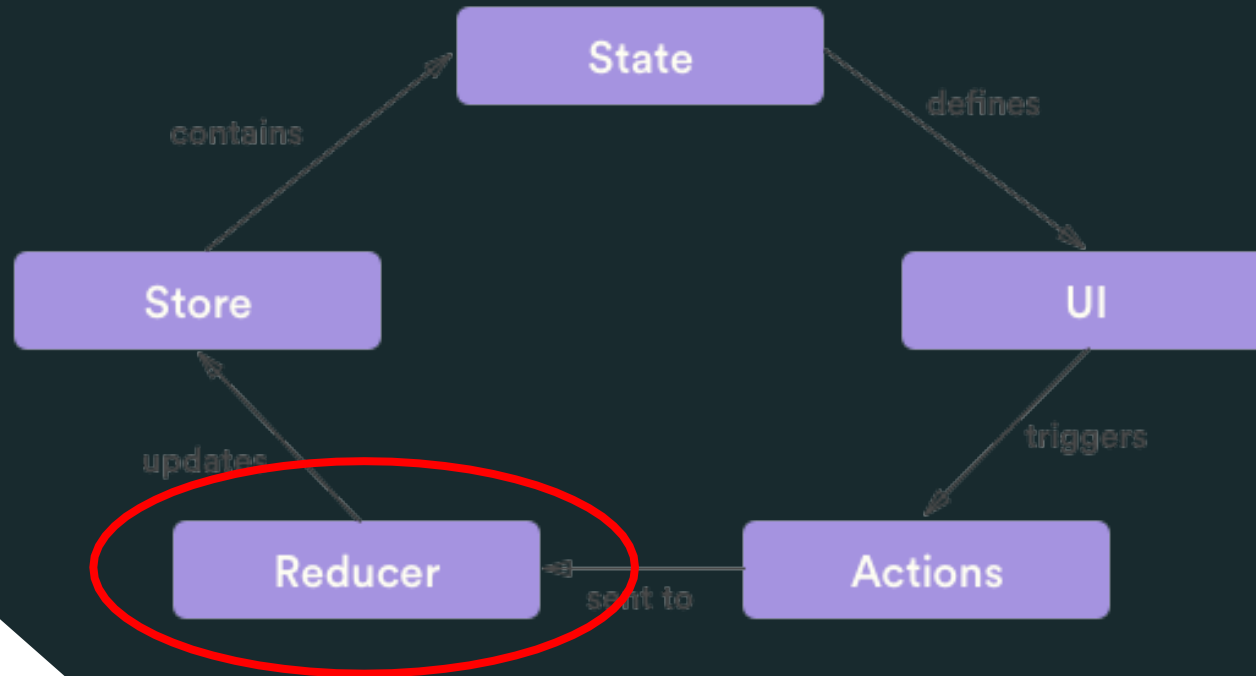
```
{  
  type: TOGGLE_TODO,  
  index: 5  
}
```

- Push in store

```
store.dispatch(toggleTodo(index))
```



REDUCER

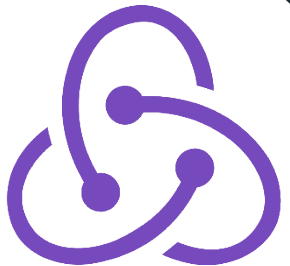


- Lo stato è un oggetto JS

```
{  
  visibilityFilter: SHOW_ALL,  
  todos: [  
    {  
      text: 'Consider using Redux',  
      completed: true  
    },  
    {  
      text: 'Keep all state in a single  
tree',  
      completed: false  
    }  
  ]  
}
```

- Semplice comportamento

`(previousState, action) => newState`





Don't be fooled by all the
fancy talk about reducers,
middleware, store enhancers
—Redux is incredibly simple

REDUCER

Funzione che calcola lo stato

```
function todoApp(prev_state, action)
{
  switch (action.type) {
    case SET_VISIBILITY_FILTER:
      return Object.assign({}, prev_state, {
        visibilityFilter: action.filter })
    case ADD_TODO:
      return Object.assign({}, prev_state, {
        todos: [
          ...prev_state.todos,
          {
            text: action.text,
            completed: false
          }
        ]
      })
    default:
      return prev_stat
```

```
if (typeof state==='undefined') {
  return startState
}
```

STORE



getState()

Permette l'accesso allo stato



dispatch(action)

Permette di aggiornare lo stato



subscribe(listener)

Registra un listener



UNIVERSITA' degli STUDI di ROMA
TOR VERGATA



Grazie per l'attenzione

Pelella, Perrone, Pusceddu, Scarpitta, Scarlino, Tranzocchi, Trotta