



Передовые
инженерные
школы



МИНОБРНАУКИ
РОССИИ



УНИВЕРСИТЕТ
ИННОПОЛИС

Занятие 22

Immutable | Action | ActionCreator |
PureFunctions | Reducer | combineReducers |
getState | dispatch | subscribe

План занятия

1. Immutable
2. Action
3. ActionCreator
4. PureFunction
5. Reducer
6. combineReducers
7. getState
8. dispatch
9. subscribe

Неизменяемость может повысить производительность приложения и упростить программирование и отладку, поскольку данные, которые никогда не меняются, легче анализировать, чем данные, которые можно произвольно изменять в приложении.

В частности, неизменяемость в контексте веб-приложения позволяет просто и дешево реализовать сложные методы обнаружения изменений, гарантируя, что дорогостоящий с точки зрения вычислений процесс обновления DOM происходит только тогда, когда это абсолютно необходимо



Action

Объект, который передаем для выполнения действия

```
{type: 'ADD_EVENT', payload: {text: 'Событие 2'}}
```



ActionCreator

```
// actionCreator
const actionCreatorAddEvent = (eventInfo) => {
  // const eventInfoFull = fetch('запрос на получение данных')
  // prepare
  const eventInfoFull = eventInfo

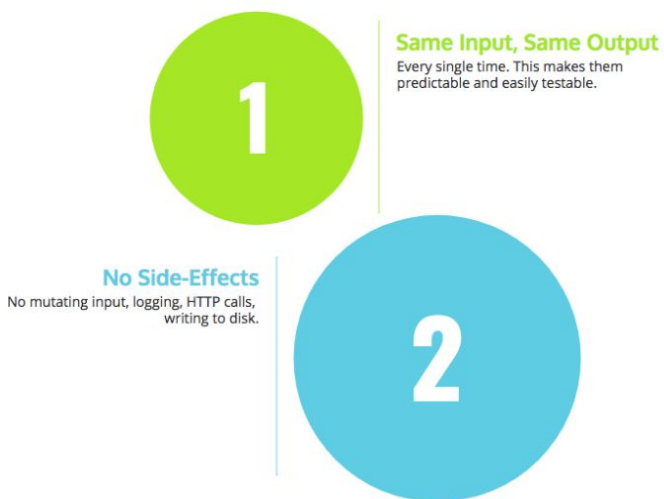
  return (
    {
      type: ACTIONS.ADD_EVENT,
      payload: eventInfoFull
    }
  )
}
```

Чистые функции - это атомарные строительные блоки в функциональном программировании. Их обожают за их простоту и тестируемость.

Функция должна пройти два теста, чтобы считаться “чистой”:

1. Одни и те же входные данные *всегда* возвращают одни и те же выходные данные
2. Никаких побочных эффектов

Pure Functions



Reducer

```
const initialStateUsers = {  
  users: [  
    'Пользователь 1'  
  ]  
}  
  
const reducerLogin = (state = initialStateUsers, action) => {  
  switch (action.type) {  
    case ACTIONS.SUCCESS_LOGIN:  
      return {  
        ...state,  
        users: action.payload  
      }  
  
    default:  
      return {  
        ...state,  
      }  
  }  
}
```

reducer — чистая функция которая будет отвечать за обновление состояния.

Функция принимает значение текущего состояния и объект события (action). Объект события содержит два свойства — это тип события (action.type) и значение события (action.value).

combineReducer

combineReducers() позволяет объединить несколько редьюсеров в один.

```
const combineReducers = (reducersMap) => {  
  return (state, action) => {  
    const nextState = {}  
  
    Object.entries(reducersMap).forEach(([key, reducer]) => {  
      nextState[key] = reducer(state ? state[key] : state, action)  
    })  
  
    return nextState  
  }  
}  
  
const rootReducer = combineReducers({  
  reducerEventsState: reducerEvents,  
  reducerUsersState: reducerLogin  
})
```


getState

// Реализация стора

```
const createStore = (reducer) => {  
  let state = reducer(undefined, {type: '__INIT'})  
  let subscribers = []  
  
  return {  
    getState: () => state,  
    dispatch: action => {  
      state = reducer(state, action)  
      subscribers.forEach((callback) => callback())  
    },  
    subscribe: (callback) => subscribers.push(callback)  
  }  
}
```

```
const store = createStore(rootReducer)  
  
console.log('store после', store.getState())
```

dispatch

// Реализация стора

```
const createStore = (reducer) => {  
  let state = reducer(undefined, {type: '__INIT'})  
  let subscribers = []  
  
  return {  
    getState: () => state,  
    dispatch: action => {  
      state = reducer(state, action)  
      subscribers.forEach((callback) => callback())  
    },  
    subscribe: (callback) => subscribers.push(callback)  
  }  
}
```

```
const store = createStore(rootReducer)
```

```
store.dispatch({type: 'ADD_EVENT', payload: {text: 'Событие 2'}})
```

subscribe

// Реализация стора

```
const createStore = (reducer) => {  
  let state = reducer(undefined, {type: '__INIT'})  
  let subscribers = []  
  
  return {  
    getState: () => state,  
    dispatch: action => {  
      state = reducer(state, action)  
      subscribers.forEach((callback) => callback())  
    },  
    subscribe: (callback) => subscribers.push(callback)  
  }  
}
```

```
const store = createStore(rootReducer)
```

```
store.subscribe(() => console.log('Изменились события...'))
```



Передовые
инженерные
школы



МИНОБРНАУКИ
РОССИИ



УНИВЕРСИТЕТ
ИННОПОЛИС

Спасибо за внимание

