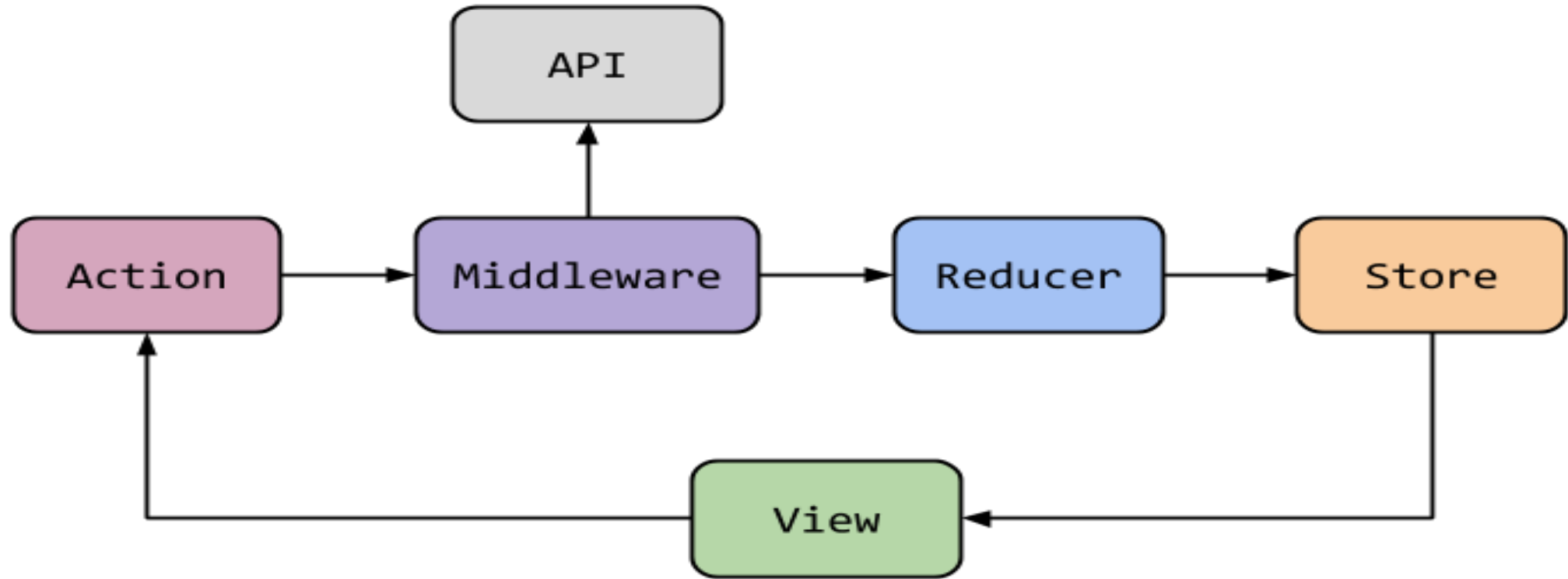


**Redux 속을 까봅시다.**

# Redux란?



출처: [https://miro.medium.com/max/760/1\\*BHUKvOmqPjJHpRY1wp1YFw.png](https://miro.medium.com/max/760/1*BHUKvOmqPjJHpRY1wp1YFw.png)

## Redux 구현 - createStore

```
const store = createStore(reducer, initialState);  
  
store.getState();  
store.dispatch(action);  
store.subscribe(listener);
```

## createStore 함수 - 초기화

```
function createStore(reducer, initialState) {  
  let currentReducer = reducer;  
  let currentState = initialState;  
  const listeners = [];  
}
```

## createStore 함수 - getState()

```
function createStore(reducer, initialState) {  
  let currentReducer = reducer;  
  //...  
  
  function getState() {  
    return currentState;  
  }  
}
```

## createStore 함수 - subscribe(listener)

```
function createStore(reducer, initialState) {  
  //...  
  const listeners = [];  
  
  function subscribe(listener) {  
    listeners.push(listener);  
  
    return function unsubscribe() {  
      const index = listeners.indexOf(listener);  
      listeners.splice(index, 1);  
    };  
  }  
}
```

## createStore 함수 - dispatch(action)

```
function createStore(reducer, initialState) {  
  let currentReducer = reducer;  
  let currentState = initialState;  
  const listeners = [];  
  
  //...  
  
  function dispatch(action) {  
    // currentReducer: (state, action) => newState  
    currentState = currentReducer(currentState, action);  
    listeners.forEach((listener) => listener());  
    return action;  
  }  
}
```

## createStore 함수 - dispatch(action)

```
function createStore(reducer, initialState) {  
  let currentReducer = reducer;  
  let currentState = initialState;  
  const listeners = [];  
  
  function getState() {  
    //...  
  }  
  
  function subscribe(listener) {  
    //...  
  }  
  
  function dispatch(action) {  
    //...  
  }  
  
  return { getState, subscribe, dispatch };  
}
```



## Redux 구현 - applyMiddleware

```
const middleware1 = (store) => (next) => (action) => {  
  // 전처리  
  const result = next(action);  
  // 후처리  
  return result;  
};  
  
const middleware2 = (store) => (next) => (action) => next(action);  
  
const enhancer = applyMiddleware(middleware1, middleware2);  
const store = createStore(reducer, initialState, enhancer);
```

## 예제1

<https://codesandbox.io/s/simple-redux-1ltzy>

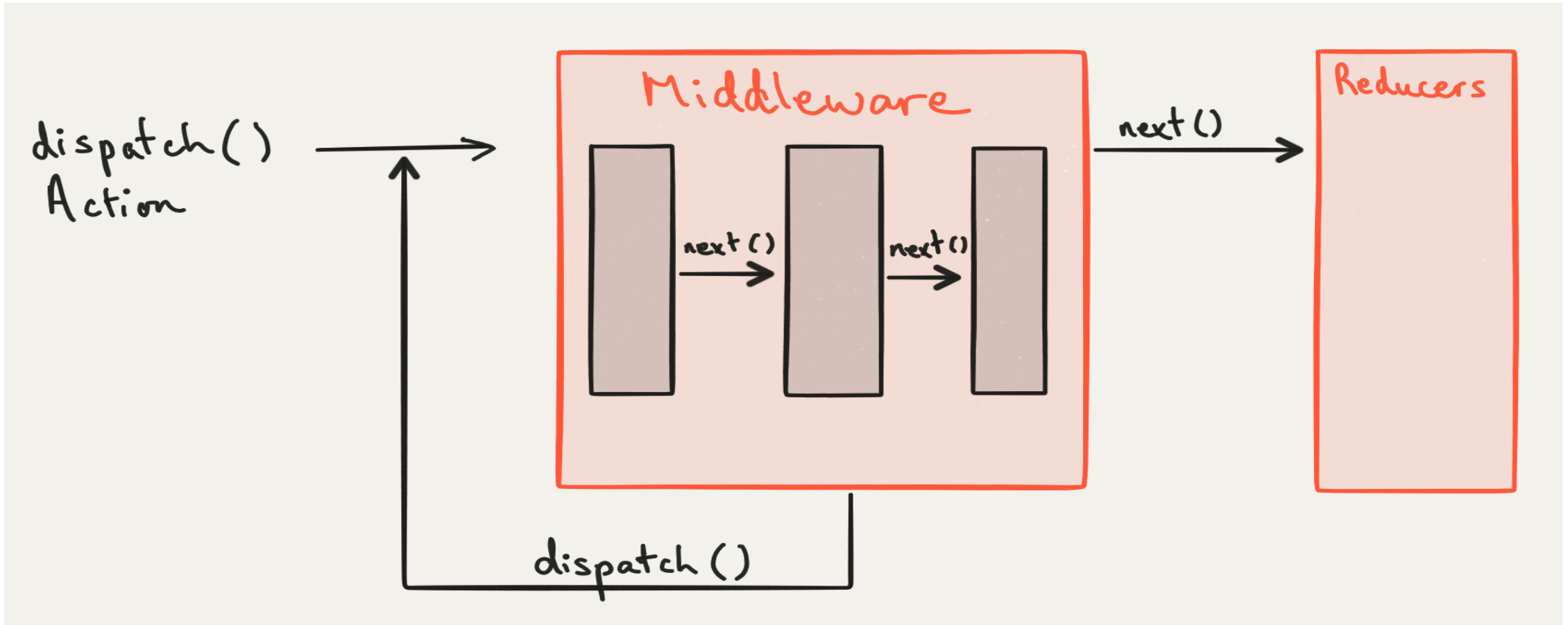
## applyMiddleware 함수 - 초기화

```
const applyMiddleware = (...middlewares) => (createStore) => (...storeArgs) => {  
  const store = createStore(...storeArgs);  
  // ...  
};
```

## applyMiddleware 함수 - store 할당

```
const applyMiddleware = (...middlewares) => (createStore) => (...storeArgs) => {
  const store = createStore(...storeArgs);
  /*
   middleware: (store) => (next) => (action) => next(action)
   middlewaresWithStore: [(next) => (action) => next(action)]
  */
  const middlewaresWithStore = middlewares.map((middleware) => middleware(store));
};
```

# applyMiddleware 함수 - middleware 체이닝



출처: <https://media.vlpt.us/images/cyongchoi/post/a9ba8061-23a1-4f42-870e-4e52e4880fd5/reduxMiddleware.png>

## applyMiddleware 함수 - middleware 체이닝

```
const applyMiddleware = (...middlewares) => (createStore) => (...storeArgs) => {  
  // ...  
  // middlewaresWithStore: [(next) => (action) => next(action)]  
  
  // chainedMiddlewares: next => action => m1(m2(m3(...(next))))(action);  
  const chainedMiddlewares = middlewaresWithStore.reduce((a, b) => (next) => a(b(next)));  
};
```

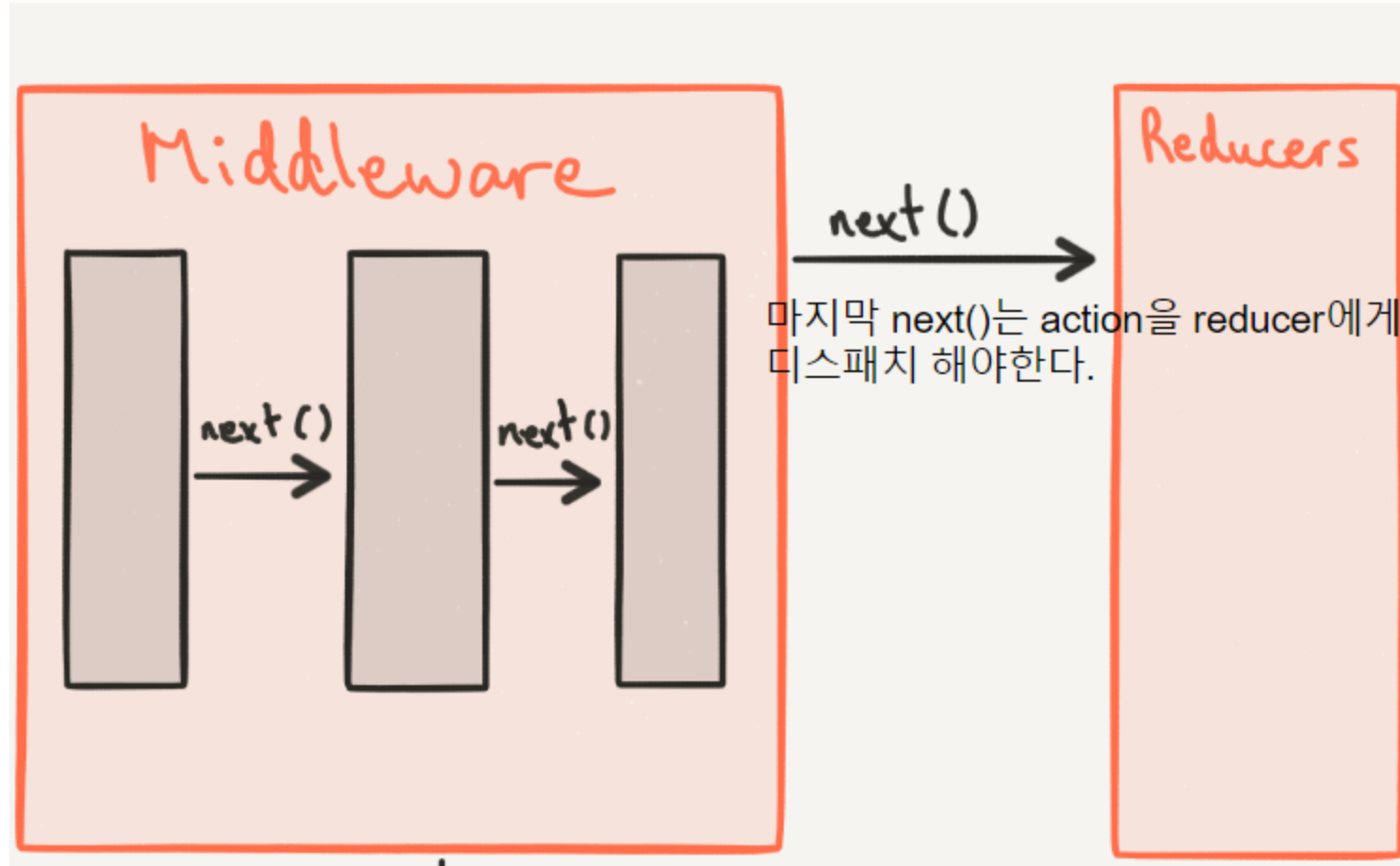
## applyMiddleware 함수 - middleware 체이닝

```
[m1, m2, m3].reduce((a, b) => (next) => a(b(next)));
```

여기서 m1,m2,m3는 (next) => (action) => next(action) 형태의 함수.

1. 초기값이 없으므로, a는 m1이 되고, b는 m2가 된다.  
(m1, m2) => (next) => m1(m2(next))
2. 1의 결과인 (next) => m1(m2(next)) 를 c로 치환한다면,  
(c, m3) => (next) => c(m3(next))  
c를 다시 치환하면, (next) => m1(m2(m3(next)))

## applyMiddleware 함수 - 마지막 next





## applyMiddleware 함수 - 마지막 next

```
const applyMiddleware = (...middlewares) => (createStore) => (...storeArgs) => {
  // ...
  // chainedMiddlewares: next => action => m1(m2(m3(...(next))))(action);
  const chainedMiddlewares = middlewaresWithStore.reduce((a, b) => (next) => a(b(next)));
  // dispatch: action => a(b(c(...(store.dispatch))))(action)
  const dispatch = chainedMiddlewares(store.dispatch);
  return {
    ...store,
    dispatch,
  };
};
```

## applyMiddleware 함수 - createStore 함수 리팩토링

```
function createStore(reducer, initialState, enhancer) {  
  let currentReducer = reducer;  
  let currentState = initialState;  
  const listeners = [];  
  
  function getState() { ... }  
  function subscribe(listener) { ... }  
  function dispatch(action) { ... }  
  
  if (enhancer) {  
    return enhancer(createStore)(reducer, initialState);  
  }  
  
  return { getState, subscribe, dispatch };  
}
```

## 예제2

<https://codesandbox.io/s/simple-redux-with-applymiddleware-sgowj>

## 참고

- <https://github.com/reduxjs/redux>
- <https://gist.github.com/gaearon/ffd88b0e4f00b22c3159>
- [책] 실전 리액트 프로그래밍, 이재승 지음, 프로그래밍 인사이트