



# Redux middleware



Rafał Jagielski

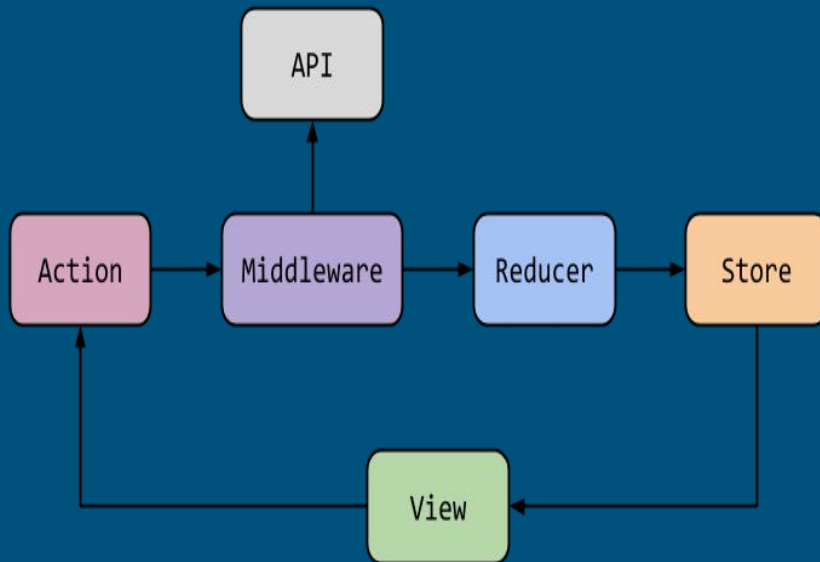


# O co chodziło w tym reduxie?

**Store** - w aplikacji jest tylko jeden Store i służy do przechowywania stanu aplikacji.

**Reducer** - decyduję jakich zmian dokonać w storze na podstawie akcji, którą użytkownik dispatchuje.

**Action** - informacja wysyłana z aplikacji do stora

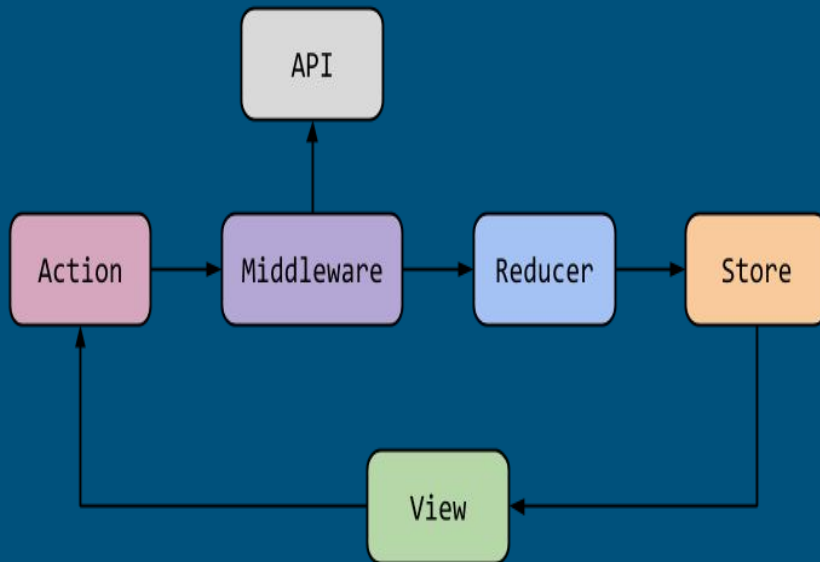


# Middleware

**Middleware** - w dosłownym tłumaczeniu oprogramowanie pośrednie.

**Schemat działania** - Przechwytuje zdispatchowaną akcję => robi to co mu zaprogramujemy => przekazuje akcję do reducera(albo nie)

**Przykłady użycia** - obsługa błędów, zapytania asynchroniczne, logowanie



# Middleware implementacja

---

```
1  const exampleMiddleware = (store) => (next) => (action) => {
2    //Tutaj wykonujemy sobie jakies operacje np. zmieniamy dane w akcji i logujemy ją
3    action.color = 'blue'
4    console.log(action)
5    //Aby przesłać akcje dalej do wykonania musimy użyć next(action) jeżeli tego nie zrobimy akcja nie będzie przekazana do reducera
6    next(action)
7  }
```

```
1  import { createStore, applyMiddleware } from "redux";
2  import exampleMiddleware from './exampleMiddleware.js'
3
4  const store = createStore(rootReducer, applyMiddleware(exampleMiddleware))
```