

REACT JS

Redux Middleware

Sreekanth M. E.

Freelance Trainer & Consultant

<http://www.SreekanthME.com>

Redux middleware provides a third-party extension point between **dispatching an action**, and the moment it **reaches the reducer**.

It is nothing but a JS function that helps to extend functionality of the redux application.

Middleware sits in between the **dispatch** and **reducers**, which means it can be used to do things like:

- Alter dispatched actions before they get to the reducers.
- Execute some code during the dispatch.
- Cancel the dispatched action.

The best feature of middleware is that it is composable in a chain.

Order of chaining is important when assembling middleware functions because they are executed in the order of appearance from left to right.

Redux middleware is used for use cases like logging, crash reporting, talking to an asynchronous API, routing, etc.

Middleware can be Third party or Custom.

CUSTOM MIDDLEWARE FUNCTION FORMAT

```
function logger(store) {  
  return function (next) {  
    return function (action) {  
      ...  
    }  
  }  
}
```

```
const logger = store => next => action => {  
  ...  
}
```

>> RHS middleware syntax is ES6 equivalent of LHS syntax.

The middleware function gets the Redux store injected as a parameter by the `applyMiddleware()` function.

It returns another function which gets the `next()` function as a parameter. Next() allows dispatch of the next action.

This second function returns another function which gets `action object` as a parameter.

A middleware function can be applied to Redux store during the store creation using `applyMiddleware()` method of `redux` library.

```
const store = createStore(reducers, applyMiddleware(logger));
```

or

```
const store = createStore(reducers,  
  applyMiddleware(middleware1, middleware2,...));
```

The `next(action)` function is called inside the middleware function to signal that this middleware is done executing and that the action must be passed to the next middleware in the middleware chain.

If `next(action)` is not called inside a middleware function, the action is canceled.

EXAMPLE OF THIRD-PARTY MIDDLEWARE

```
npm install --save redux-logger
```

<https://www.npmjs.com/package/redux-logger>

END OF CHAPTER

APPENDIX