

## Redux-Saga Burger Application:

setAuthRedirectPath("/")

Authentication: Auth.js: dispatch auth(email, password, isSignUp)

In Action: Auth.js: • auth(...)  $\Rightarrow$  AUTH-USER, email, password, isSignUp

- authStart()  $\Rightarrow$  AUTH-START
- authSuccess(...)  $\Rightarrow$  AUTH-SUCCESS, idToken=token, userID=userId
- authFail(error)  $\Rightarrow$  AUTH-FAIL, error=error
- logout()  $\Rightarrow$  AUTH-INITIATE-LOGOUT  $\Leftarrow$  triggers for Saga
- logoutSuccess()  $\Rightarrow$  AUTH-LOGOUT
- setAuthRedirectPath(path)  $\Rightarrow$  SET-AUTH-REDIRECT-PATH, path=path
- checkAuthTimeout(...)  $\Rightarrow$  AUTH-CHECKTIMEOUT, expirationTime
- authCheckState()  $\Rightarrow$  AUTH-CHECK-STATE

Auth Reducer {state = <sup>(null)</sup> token, <sup>(null)</sup> userID, <sup>(null)</sup> error, <sup>(false)</sup> loading, <sup>"/"</sup> authRedirectPath}

Now:

- AUTH-START: authStart  $\Rightarrow$  error: null, loading: true
- AUTH-SUCCESS: authSuccess  $\Rightarrow$  token: action.idToken, userID: action.userID, error = null, loading = false
- AUTH-FAIL: authfail  $\Rightarrow$  error: action.error, loading: false
- AUTH-LOGOUT: authlogout  $\Rightarrow$  token: null, userID: null
- SET-AUTH-REDIRECT-PATH: setAuth-  $\Rightarrow$  authRedirectPath = action.path

### Auth Sagas:

Index (RootSaga): AUTH-USER  $\Rightarrow$  authUserSaga

AUTH-INITIATE-LOGOUT  $\Rightarrow$  logoutSaga

AUTH-CHECK-TIMEOUT  $\Rightarrow$  authCheckTimeoutSaga

AUTH-CHECKSTATE  $\Rightarrow$  authCheckStateSaga

Note: In actions: Only creators are defined with parameters  
In Reducers: Simple logic like state changes are done  
In Sagas: Real logics and fetchings are done with the help of actions and Reducers to change states.  
\* Please see code example \*



## BurgerBuilder

BurgerBuilder.js: dispatched actions: `addIngredient(ingName)`, `removeIngredient(ingName)`, `initIngredients()`, `purchaseInit()`, `setAuthRedirectPath(path)`

Actions:

- `addIngredient=(name) ⇒ ADD-INGREDIENT, ingredientName: name`
- `removeIngredient=(name) ⇒ REMOVE-INGREDIENT, ingredientName: name`
- `initIngredients = () ⇒ INIT-INGREDIENTS`
- `setIngredients = (ing) ⇒ SET-INGREDIENTS, ingredients: ingredients`
- `fetchIngredientsFailed = () ⇒ FETCH-INGREDIENTS-FAILED`

Reducer:  $\{state \Rightarrow \overset{(null)}{ingredients}, \overset{(4)}{totalPrice}, \overset{(false)}{error}, \overset{(false)}{building}\}$

ADD-INGREDIENT: `addIngredient ⇒ Updated  $\overset{(4)}{ingredients}, \overset{(false)}{totalPrice}, \overset{(false)}{building}: true$`

REMOVE-INGREDIENT: `removeIngredient ⇒ " " " " "`

SET-INGREDIENTS: `setIngredients ⇒ Initial  $\overset{(4)}{ingredients}, \overset{(false)}{totalPrice}=4, \overset{(false)}{building}: false$`

FETCH-INGREDIENTS-FAILED: `fetchIngredientsFailed ⇒ error: true`

Saga (rootSaga): `INITINGREDIENTS, initIngredientsSaga`

Here happens:  $\Rightarrow$  axios get request, `setIngredients(response.data)`  
& if error  $\Rightarrow$  `fetchIngredientsFailed()`



## Orders

Orders.js:

dispatch  $\Rightarrow$  fetchOrders(token, userID)  $\leftarrow$  how they were received?

Actions:

fetchOrders(...)  $\Rightarrow$  FETCH-ORDERS, token: token, userID: userID  
fetchOrdersStart()  $\Rightarrow$  FETCH-ORDERS-START  
fetchOrdersSuccess()  $\Rightarrow$  FETCH-ORDERS-SUCCESS, orders: orders  
fetchOrdersFail(error)  $\Rightarrow$  FETCH-ORDERS-FAIL, error: error  
purchaseInit()  $\Rightarrow$  PURCHASE-INIT  
purchaseBurger(...)  $\Rightarrow$  PURCHASE-BURGER, orderData, token  
purchaseBurgerStart()  $\Rightarrow$  PURCHASE-BURGER-START  
purchaseBurgerSuccess()  $\Rightarrow$  " " SUCCESS, orderData, orderID  
purchaseBurgerFail()  $\Rightarrow$  " " FAIL

Reducers: state { orders: [], loading: false, purchased: false }

FETCH-ORDERS-START: fetchOrdersStart()  $\Rightarrow$  loading: true (true)  
FETCH-ORDERS-SUCCESS: fetchOrdersSuccess()  $\Rightarrow$  orders: action.orders, loading: false (true)  
" " FAIL: " " fail()  $\Rightarrow$  loading: false  
PURCHASE-BURGER-START: purchaseBurgerStart()  $\Rightarrow$  loading: true, false (true)  
" " SUCCESS: " " success  $\Rightarrow$  updated orders, loading, purchased  
" " FAIL: " " fail  $\Rightarrow$  loading: false  
PURCHASE-INIT: purchaseInit  $\Rightarrow$  purchased: false

Sagas (rootSaga): FETCH-ORDERS, fetchOrdersSaga  
PURCHASE-BURGER, purchaseBurgerSaga

fetchOrdersSaga()  $\Rightarrow$  fetchOrdersStart, Success, Fail

purchaseBurgerSaga  $\Rightarrow$  purchaseBurgerStart, Success, Fail