watcher – следит за Redux действиями и определяет worker’a

worker – определяет как реагировать на событие

export function\* incrementAsync() {  
 yield ***call***(delay, 1000)  
 yield ***put***({type: 'INCREMENT'})  
}  
  
export function\* watchIncrementAsync() {  
 yield takeEvery('INCREMENT\_ASYNC', incrementAsync)  
}  
  
export default function\* rootSaga() {  
 yield all([helloSaga(), watchIncrementAsync()])  
}

const sagaMiddleware = createSagaMiddleware()  
const store = ***createStore***(reducer, applyMiddleware(sagaMiddleware))

sagaMiddleware.run(rootSaga)

export default function counter(state = 0, action) {  
 switch (action.type) {  
 case 'INCREMENT':  
 return state + 1  
 case 'INCREMENT\_IF\_ODD':  
 return (state % 2 !== 0) ? state + 1 : state  
 case 'DECREMENT':  
 return state - 1  
 default:  
 return state  
 }  
}