02

OPEN ORIENTED

凹凸实验室

聊一聊Redux-Saga

Redux-Saga

redux-saga 是一个用于管理 Redux 应用异步操作的中间件（又称异步 action）

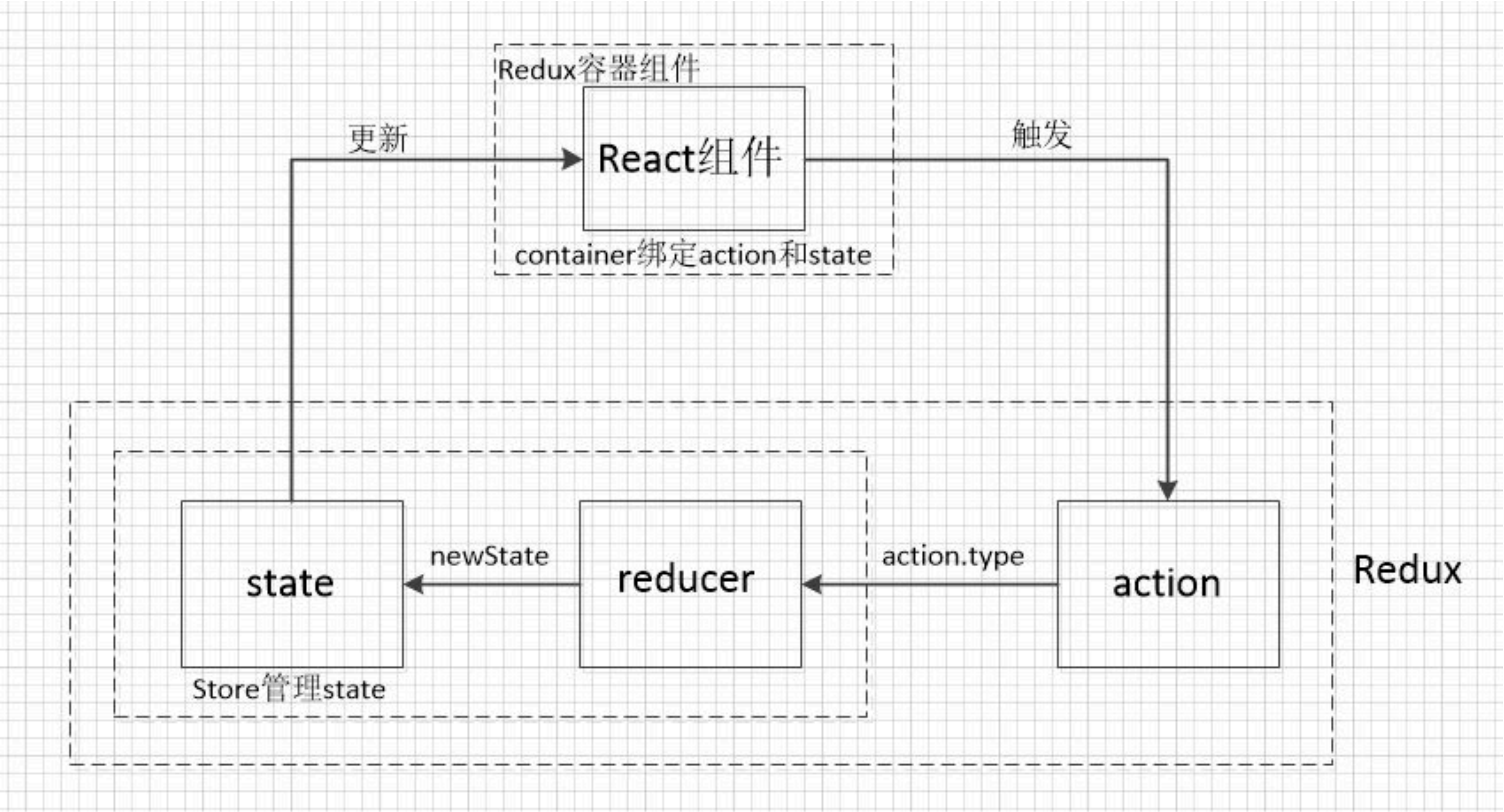
1 为什么选择Redux-Saga?

为什么?

Contrary to redux thunk, you don't end up in callback hell, you can test your asynchronous flows easily and your actions stay pure.

为什么?

Redux-Saga比Redux-Thunk更加的纯



为什么?

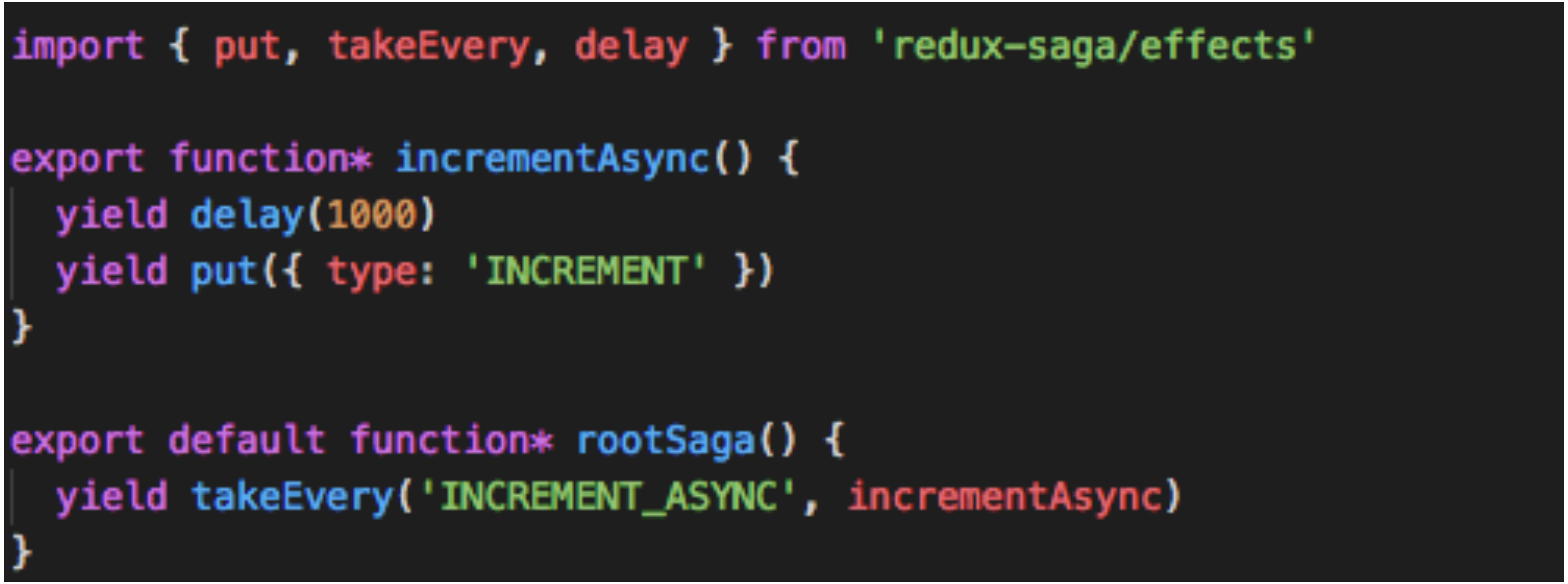
提供了一些实用、强大的API

对于Action的执行流程具有强大的控制

2 如何使用?

如何使用?

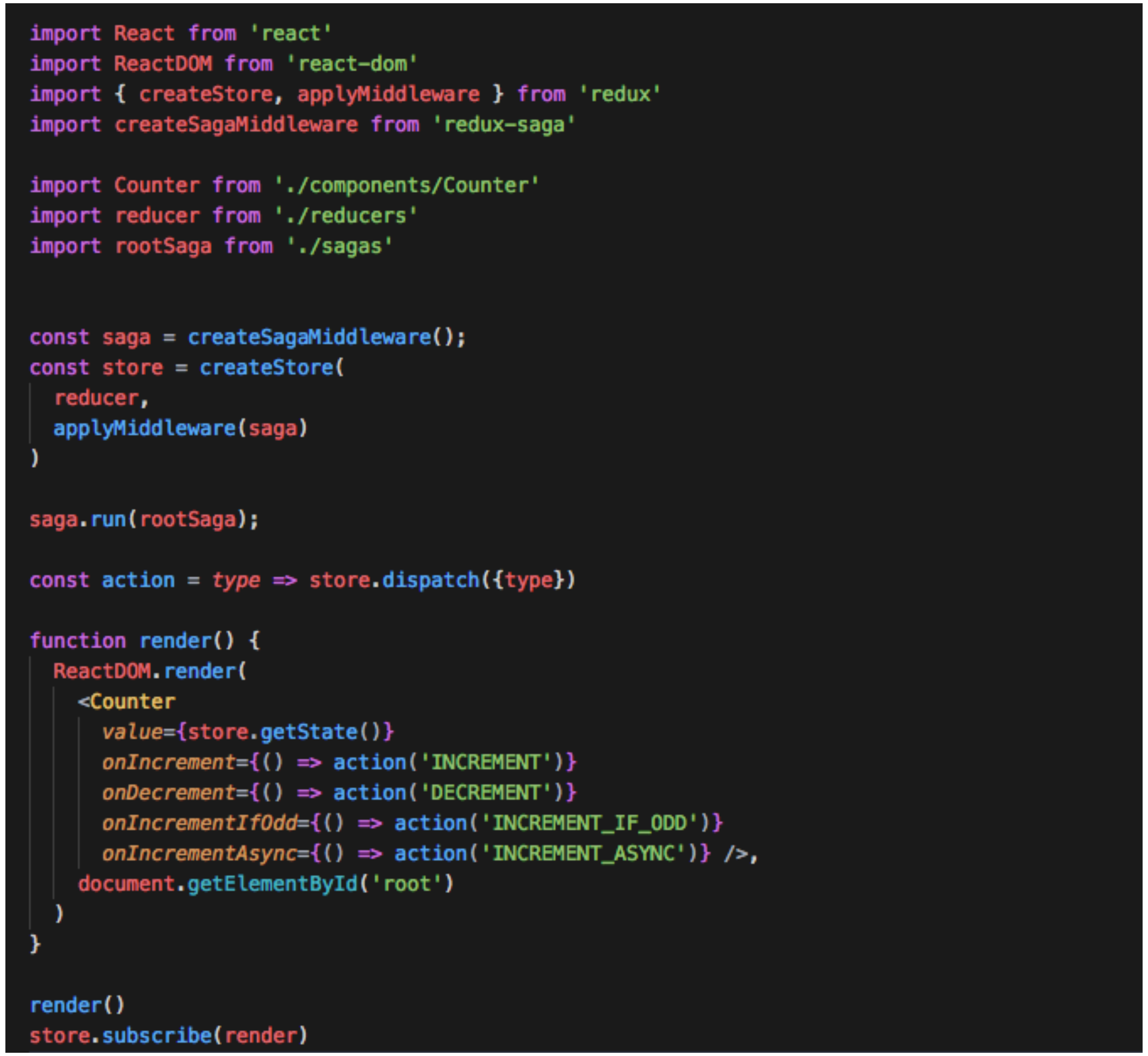
1.建立Saga文件



```
import { put, takeEvery, delay } from 'redux-saga/effects'

export function* incrementAsync() {
  yield delay(1000)
  yield put({ type: 'INCREMENT' })
}

export default function* rootSaga() {
  yield takeEvery('INCREMENT_ASYNC', incrementAsync)
}
```

如何使用?

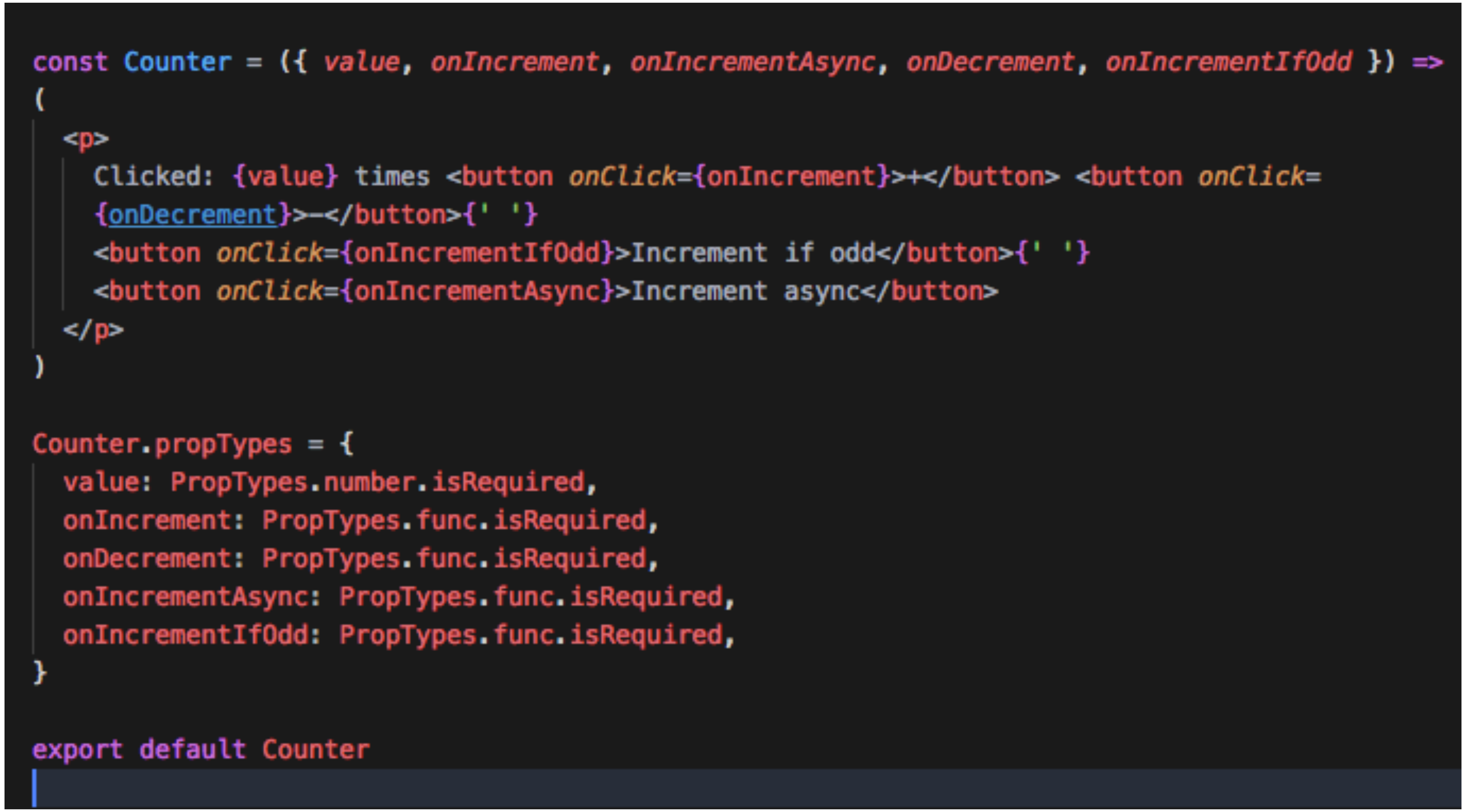
2.启动Saga



```
import React from 'react'
import ReactDOM from 'react-dom'
import { createStore, applyMiddleware } from 'redux'
import createSagaMiddleware from 'redux-saga'

import Counter from './components/Counter'
import reducer from './reducers'
import rootSaga from './sagas'


const saga = createSagaMiddleware();
const store = createStore(
  reducer,
  applyMiddleware(saga)
)

saga.run(rootSaga);

const action = type => store.dispatch({type})

function render() {
  ReactDOM.render(
    <Counter
      value={store.getState()}
      onIncrement={() => action('INCREMENT')}
      onDecrement={() => action('DECREMENT')}
      onIncrementIfOdd={() => action('INCREMENT_IF_ODD')}
      onIncrementAsync={() => action('INCREMENT_ASYNC')} />,
    document.getElementById('root')
  )
}

render()
store.subscribe(render)
```

如何使用?

3.触发Saga



```
const Counter = ({ value, onIncrement, onIncrementAsync, onDecrement, onIncrementIfOdd }) =>
(
  <p>
    Clicked: {value} times <button onClick={onIncrement}>+</button> <button onClick=
    {onDecrement}>-</button>{' '}
    <button onClick={onIncrementIfOdd}>Increment if odd</button>{' '}
    <button onClick={onIncrementAsync}>Increment async</button>
  </p>
)

Counter.propTypes = {
  value: PropTypes.number.isRequired,
  onIncrement: PropTypes.func.isRequired,
  onDecrement: PropTypes.func.isRequired,
  onIncrementAsync: PropTypes.func.isRequired,
  onIncrementIfOdd: PropTypes.func.isRequired,
}

export default Counter
```

3

一些API介绍

API介绍

02

四个概念

阻塞　　非阻塞

Task　　Channel

API介绍

Action的获取

take

takeEvery

throttle

takeLeading

takeLatest

API介绍

指示Middle调用函数

call

cancle

put

fork

join

select

OPEN ORIENTED

凹凸实验室