https://dvajs.com/guide/

（一）介绍

dva是基于redux和redux-saga的数据流方案，内置react-router和fetch

（二）快速上手

（三）例子和脚手架

（四）Dva 概念

（1）数据流向

dispatch一个action，如果是同步通过Reducers改变State，如果是异步先触发Effects 然后通过Reducers改变State

（2）action（动），改变state的唯一途径

（3）dispatch，触发 action

（4）router（路），底层使用react-router

（5）route components（组）

（五）入门课

（1.1）connect，绑定state到view（UI 组件），返回容器组件

（1.2）connect传入的第一个参数是mapStateToProps函数，mapStateToProps 函数返回一个对象，建立state到props的映射关系。

（1.3）被connect的component会在props中拥有dispatch方法。

（2.1）dva提供app.model（模）对象，所有的应用逻辑都定义在上面，属性包括：

（2.2）namespace，当前mode名称

（2.2）state，状态

（2.3）reducers（减），纯函数

（2.4）effects（效），副作用，底层使用redux-sagas

（2.5）subscription，订阅

（六）Dva 图解

（七）使用 Dva 开发复杂 SPA

（1）动态加载model

（2）使用model共享全局信息

const { namespace1State, namespace2State } = yield select();

（3）model的复用

（4）动态扩展model

（5）长流程的业务逻辑

（6）使用take操作进行事件监听

（7）多任务调度

yield []; 并行执行

yield\* []; 顺序执行

（8）任务的竞争

（9）跨model的通信

yield call({ type: 'namespace/typeName' });

yield take('namespace/typeName');

（10）为DVA应用编写测试

基流动触路组绑传被 模名状减效订全多跨