使用js-sdk的步骤：

1. 引入js-sdk
2. 通过config接口注入配置信息，
3. 通过ready接口处理成功的验证。config配置信息验证后会执行ready方法。Config是异步操作，所以在页面加载时就要调用接口的话必须把接口调用放在ready函数中。
4. 所有接口通过wx对象(也可使用jWeixin对象)来调用，参数是一个对象，
5. “分享到朋友圈”：wx.onMenuShareTimeline({})
6. “分享给朋友”： wx.onMenuShareAppMessage({})
7. “分享到QQ”： wx.onMenuShareQQ({})
8. “分享到QQ空间”: wx.onMenuShareQZone({})
9. 拍照或从手机相册中选图 : wx.chooseImage({})
10. 微信内置地图查看位置接口 : wx.openLocation({})
11. clearfix 和clear？<http://www.daqianduan.com/3606.html>?
12. 网速自定义？查看当前网速？

TODO:

1. 自执行函数的七种写法：+-！~（（））()() var a = f{}()

任何能将函数变成一个函数表达式的作法，都可以使解析器正确的调用定义函数。

1. 函数作为对象方法调用：this？
2. 函数创建的三种方式？
3. 函数调用的四种方法？
4. https://www.zhihu.com/question/33629083
5. pingfang 字体？？
6. translate 布局为什么比 top left 布局好？？
7. GPU加速？

<https://www.zhihu.com/question/33629083>？

1. 评论分页：
2. React Native Router v4.x

based on [React Navigation](https://reactnavigation.org/)

motivation: 动机

intuitive：直观的

RNRF - simple API for easy navigation

navigation instruments:导航仪器

## I want to talk about latest version (v4) of RNRF based on ReactNavigation and MobX and provide best practices. New version provides not only navigation solution but also proposes a way to manage your app state.

Comparision with ReactNavigation API core (best navigation so far)

写法上的区别：

Actions[routeName](params) vs this.props.navigation.navigate({key: routeName, params})

this.props vs this.props.navigation.state.params

More navigation actions (like popTo) (additional functionality of RNRF)

Based on latest [React Navigation](https://reactnavigation.org/) API

Separate navigation logic from presentation

onEnter, onExit, success, and failure

 You may subscribe to navigationStore (Actions in v3) and observe current navigation state.

Replace：替换屏幕：当前屏幕立即替换

Pop error：弹出一个模态框

Drawer support (provided by React Navigation)

Inheritance of scene attributes allow you to avoid any code/attribute duplications

Use Actions.currentScene to get name of current scene.

Router：

This Object is the main utility is to provide navigation features to your application.

Pop： Go back to the previous scene by "popping" the current scene off the nav stack

注册组件：

import Example from './Example';

AppRegistry.registerComponent('Example', () => Example);

Example：

<Router><Overlay></Overlay></Router>

Stack 里面 可以有scene ，scene 里面又可以有 stack。

RN router flux VS react router 4.0

每添加一个页面的时候就要添加一个scene。

属性：

leftTitle="Cancel"： 左边标题

onLeft={Actions.pop}：点击左边按钮

component={Login}：设置跳转到的组件

title="Lgin"： 设置标题

hideNavBar：隐藏头部的导航栏

navBar：定制一个当前scene的navbar

组件里面的onEnter函数：

static onEnter = () => {

Actions.refresh({

title: 'Login!',

rightTitle: 'rightTitle',

onRight: () => {},

});

};

一个key就对应着一个组件，每一个scene，stack，modal，layout都对应着一个key。key的值是挂在到actions下面的，直接通过Actions.keyname({data:;title:;})跳转到对应的组件。

###

进入一个栈（默认进入stack中的第一个scene）之后：

跳转scene的写法：

1. <Button onPress={Actions.register2}>Register</Button>
2. <Button onPress={（）=> Actions.register2()}>Register</Button>

错误的写法：

<Button onPress={Actions.home()}>Replace screen</Button>

这样写会进入页面后直接跳转到scene的key为home的场景。

Actions.pop()：退出当前这个栈Stack, 而不是返回到上一个页面scene。

renderLeftButton={() => <Text>Right</Text>}

renderTitle={() => <Text>Right</Text>}

renderRightButton={() => <Text>Right</Text>}

一个文件（Example.js文件里面集中了所有的路由，所有的页面都在这里配置。）

怎么自定义NavBar？

一个stack里面的scene是从右边滑过来的，

navBar：定制一个当前scene的navbar，并且其组件可以获得scene的参数title。

Translation：平移

Actions API：

refresh

：Reloads the current scene by loading new props into the Scene

replace：

Pops the current scene from the stack and pushes the new scene to the navigation stack. \*No transition will occur.

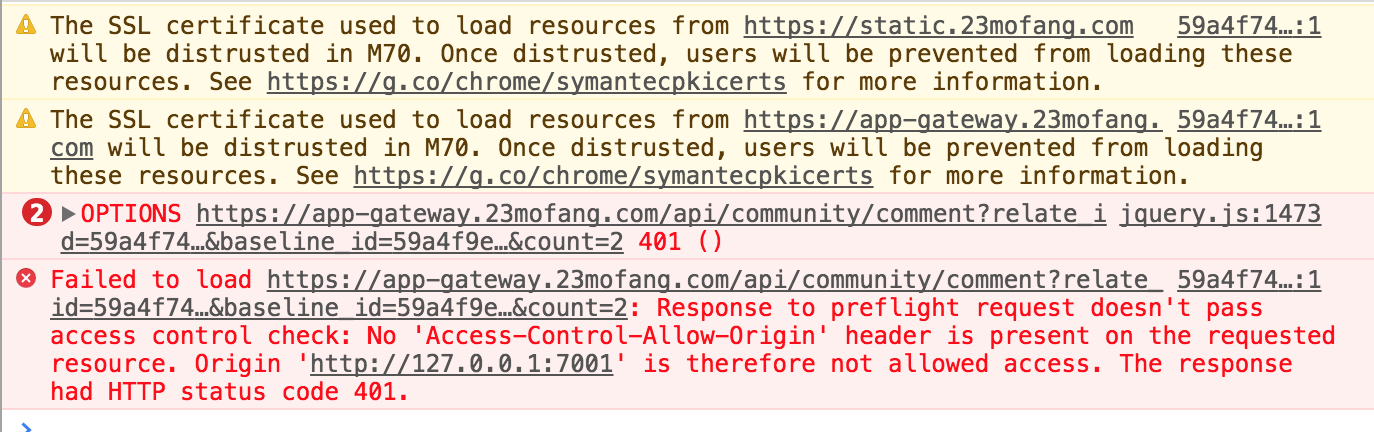
pop：Go back to the previous scene by "popping" the current scene off the nav stack

currentScene

:Returns the current scene that is active，返回的是当前scene的key

sceneKey：

TODO:



before()?

在html中可以嵌入其他文件内容或者base64编码值，可以在资源定位的基础上，给资源加 **?\_\_inline** 参数来标记资源嵌入需求。?

编译函数 **\_\_inline()** 来提供内容嵌入能力。?

**var** res = nunjucks.render('foo.html', { username: 'James' });

Template 是一个模板编译后的对象，然后进行渲染?

tmpl.render(context, [callback])

渲染模板，**context** 为数据，?

TODO:跨域？