码龄6年 暂无认证
GitCode
6 原创 | 106万+ 周排名 | 24万+ 总排名 | 4939 访问 | 等级
66 积分 | 0 粉丝 | 1 获赞 | 0 评论 | 2 收藏
私信 关注

搜博主文章

热门文章
React Native三端同构 2058
React Native包体积优化之图片优化 1535
基于React Navigation的拦截器封装 463
Retrofit的初步使用 373
SpringBoot起服务报错 255

分类专栏
React Native 1篇
Java 1篇
安卓学习笔记 1篇

您愿意向朋友推荐“博客详情页”吗?
强烈不推荐 不推荐 一般般 推荐 强烈推荐

最新文章
SpringBoot起服务报错
React Native三端同构
React Native包体积优化之图片优化
2022年 4篇 2016年 2篇

目录

原创 土豆爱吃西瓜 已于 2022-02-14 21:27:34 修改 463 收藏 版权
分类专栏: React Native 文章标签: react.js react native javascript

运行环境

```
React Navigation 6.x
React Native >= 0.63.0
Node > 12
```

代码参考

npm地址
github地址：lib文件夹为库项目、demo文件夹为示例

React Navigation介绍

React Navigation为React Native提供的路由管理库，利用该库可以很好的管理页面跳转，及导航栏的配置。利用该库可以实现普通的页面跳转，但是在涉及拦截器使用的时候，实现起来就比较麻烦。我们举例说明一下，假如我们程序有如下几个页面

```
|-- pages
    |-- login.js 登录页面
    |-- record.js 我的订单页面
    |-- buy.js 购买页面
    |-- home.js 首页
```

我的订单跟购买页面需要依赖用户登录成功才能进入，那么我们常规的操作是通过标识来实现的，如下代码所示
pages/home.js

```
import React from 'react';
import { Button } from 'react-native';
class Home extends React.Component{
    render(){
      return (
      <View style={styles.container}>
        <Button title="我的订单" onPress={() => this.props.navigation.navigate('login'), {toPage: "/record"}} />
        <Button title="购买" onPress={() => this.props.navigation.navigate('login'), {toPage: "/buy"}} />
      </View>
    );
    }
}
```

如上所示，通过传递toPage来标识登录成功后，跳转的页面，如下修改登录页面，pages/login.js

```
import React from 'react';
import { Button } from 'react-native';
class Login extends React.Component{
    login(){
        ...扒拉扒拉一顿登录操作
        const { toPage } = this.props.route.params;
        this.props.navigation.navigate(toPage);
    }

    render(){
      return (
          <View style={styles.container}>
            <Button title="我的订单" onPress={this.login} />
          </View>
      );
    }
}
```

从以上可以看出，非常繁琐，而且代码非常耦合，有没有更好的解决方案，答案是有的，也就是拦截器的概念，通过配置拦截器让页面只需关系自己业务功能就行了。

如何设计拦截器

先看下React Navigation是怎么配置路由表的，app.js如下所示：

```
import { NavigationContainer } from '@react-navigation/native';
import { createStackNavigator } from '@react-navigation/stack';
import { Home } from './pages/home';
import { Login } from './pages/login';
import { Record } from './pages/record';
import { Buy } from './pages/buy';
const Stack = createStackNavigator();

function App() {
```

```
  return (
    <NavigationContainer>
      <Stack.Navigator>
        <Stack.Screen name="/home" component={Home} />
        <Stack.Screen name="/login" component={Login} />
        <Stack.Screen name="/record" component={Record} />
        <Stack.Screen name="/buy" component={Buy} />
      </Stack.Navigator>
    </NavigationContainer>
  );
}

export default App;
```

如何设置路由表

首先，我们需要有个路由表，然后根据路由表解析成React Navigation认识的形式，也就是上面的代码。新增路由表route-config.js，代码如下

```
import { Home } from './pages/home';
import { Login } from './pages/login';
export default {
    "/home":{
   component: Home
    },
    "/login":{
   component: Login
    }
}
```

从上面的路由可以看到路由的路径是以 '/' 开头的，做过Web应该比较熟悉，对以后做三端(Android、iOS、Web)重构也是适配的。

然后解析成React Navigation认识的格式：

```
import { NavigationContainer } from '@react-navigation/native';
import { createStackNavigator } from '@react-navigation/stack';
const Stack = createStackNavigator();
import routeConfig from './route-config'

function App() {
  return (
    <NavigationContainer>
      <Stack.Navigator>
            {
              Object.keys(routeConfig).map((key) => {
                  const item = routes[key];
                  return <Stack.Screen name={key} component={item.component} key={key}/>
            }
      </Stack.Navigator>
    </NavigationContainer>
  );
}

export default App;
```

如何配置拦截器

路由表已经有了，接着思考如何加上拦截器，考虑到一个页面有多个拦截器的可能性，所以拦截器应该是一个数组，拦截器在一开始就能确定，所以修改route-config.js，如下所示

```
import { Home } from './pages/home';
import { Login } from './pages/login';
import { Record } from './pages/record';//我的订单
import { LoginInterceptor } from './interceptors/login-interceptor';//登录拦截器

export default {
    "/home":{
   component: Home
    },
    "/record":{
   interceptors:[
           {
         clazz: LoginInterceptor
           }
       ]
   component: Record
    }
}
```

拦截器写好，以上逻辑为进入我的订单路由 /record 会先检验登录拦截器，也就是判断用户有没有登录，没有登录的话，会先在登录拦截器里面跳转到登录页面。登录拦截器代码如下：

```
class LoginInterceptor {
  intercept() {
    if(已登录){
      跳转到下一个拦截器或到目标页
```

```
    }else {
      跳转到登录页面
    }
  }
}
```

intercept为拦截函数

如何才能执行拦截器

这一步，首先我们得先用个管理类，来管理包裹React Navigation的跳转方法，等拦截器执行结束在执行跳转到目标页面。所以新增lib/router.js。

```
export class Router {
    // name为路由如/record，params为参数
    push(name, params){}
}
```

Router 需要拿到React Navigation的ref，才能跳转，修改app.js。

```
import { NavigationContainer } from '@react-navigation/native';
import { createStackNavigator } from '@react-navigation/stack';
import routeConfig from './route-config'
import { navigationRef } from './router';

const Stack = createStackNavigator();

function App() {
  return (
    <NavigationContainer ref={navigationRef}>
      <Stack.Navigator>
            {
              Object.keys(routeConfig).map((key) => {
                  const item = routes[key];
                  return <Stack.Screen name={key} component={item.component} key={key}/>
            }
      </Stack.Navigator>
    </NavigationContainer>
  );
}
```

修改lib/router.js，涉及拦截过滤器模式

```
import { createNavigationContainerRef } from '@react-navigation/native';
import { StackActions, CommonActions } from '@react-navigation/native';
import { FilterManager } from './filter-manager';
import routeResolve from './route-resolve';
import routeConfig from './route-config'

export class Router {
  constructor() {
    this.filterManager = {};
    this.startLen = 0;
  }

  //跳转函数
  push(name, params){
      // 声明目标函数
      function targetFun() {
        const pushAction = StackActions.push(name, params);
        navigationRef.dispatch(pushAction);
      }
      this.execute(name, targetFun);
  }

  execute(name, targetFun) {

    let route = routeConfig[name];

    if (route) {
      const interceptors = route.interceptors;

      if (interceptors) {
        let interceptorClazzs = [];
        interceptors.forEach(element => {
          interceptorClazzs.push(element["clazz"]);
        });

        const state = navigationRef.getState();
        this.startLen = state ? state.routes.length : 1;

        // 过滤管理器
        this.filterManager = new FilterManager(interceptorClazzs, targetFun);
        this.filterManager.execute();

      } else {
        targetFun();
```

```
      }
    }
  }

  /**
   * 执行下一个拦截器
   */
  interceptNext() {
    this.clearStack();
    this.filterManager.execute();
  }

  /**
   * 清栈
   */
  clearStack() {
    navigationRef.dispatch(state => {
      const routes = state.routes.filter((r, index) => {
        return index < this.startLen;
      });

      return CommonActions.reset({
        ...state,
        routes,
        index: routes.length - 1,
      });
    });
  }
}

export const navigationRef = createNavigationContainerRef();
export let router = new Router();
```

新增过滤管理器 lib/filter-manager.js

```
/**
 * 拦截过滤管理器
 */
export class FilterManager {
  constructor(interceptorClazzs, targetFun) {
    this.index = 0;
    this.targetFun = targetFun;
    this.interceptorClazzs = interceptorClazzs;
  }

  /**
   * 执行拦截器
   */
  execute(){
    if(this.index == this.interceptorClazzs.length){
      // 执行目标函数
      this.targetFun();
    }else{
      // 获取下一个拦截器
      let interceptor = new this.interceptorClazzs[this.index]();
      this.index++;
      interceptor.intercept();
    }
  }
}
```

至此，拦截器的设计就完成了，可以愉快的调用了。

使用router跳转

修改Home页面的跳转，对应的就会调起登录拦截器

```
import React from 'react';
import { Button } from 'react-native';
import { router } from './lib/router';
class Home extends React.Component{
    render(){
      return (
      <View style={styles.container}>
        <Button title="我的订单" onPress={() => router.push('/record')} />
        <Button title="购买" onPress={() => router.push('/buy')} />
      </View>
    );
    }
}
```

react项目：react拦截器和token问题 初郁 1915
这次的交互遇到token问题真的毫无思路，之前想的是将登录的user保存到memoryUtils，通过判断是否有值来确定用户有没有登录，可当请求其他接口时…

react（axios）拦截器的简单配置 lxw0518的博客 1495
import React from 'react'; import axios from 'axios'; import {BrowserRouter as Router} from 'react-router-dom' import {baseLocalUrl} from '…/server' imp…

React Axios拦截器配置 皮皮很开心 545
1. 安装 Axios： npm i axios 2. 创建 utils文件夹： 3.创建request.js： import axios from 'axios' // 第一步，创建实例 const service = axios.create({ baseUR…

React-Native配置@react-navigation/stack@6.x——使用自定义导航栏/通用路由模块/统一切换效果 没沙发 555
本篇文章所讲的内容均可以在这个库中查看hao-react-navigation，下载安装依赖即可运行 相关依赖版本 "@react-native-masked-view/masked-view": "^0.…

React拦截器 大宇的博客，欢迎访问 506
axios拦截器之axios.interceptors.request.use和axios.interceptors.response.use（附示例代码）_阿小绿的博客-CSDN博客_interceptors.request axios.int…

react-navigation 路由级登录拦截 zww学习笔记 6072
// 路由栈 const Navigator = StackNavigator({ Setting: {screen: SettingSCreenView}, ... }); 1、设置路由状态改变拦截 function getCurrentRouteName(na…

vue3.x+vite+element-plus+vuex+typescript weixin_50923296的博客 484
vue3.x+vite+element+vuex 环境准备：nodejs>12.0 npm>6.0 1.下载模板vue-ts npm init vite@latest my-vue-typescript --template vue-ts cd my-vue-type…

React Native非内容页跳转问题（axios拦截器跳转登录页） Liangingword的博客 343
React的开发在网络访问的时候，后台返回token过期的地方，跳转登录页，是在axios中用拦截器实现的。在React Native使用同样的功能时候，react-navi…

如何更好的在 react 中使用 axios 的拦截器 qq_42881675的博客 504
title: 如何更好的在 react 中使用 axios 的拦截器 createAt: 2021-09-28 作者：玄晓乌屋 备注：转载与借鉴请注明出处。 前言 axios 算是当下最热门的前端…

react-router v6 简单的路由拦截 喵 2587
非常简单的路由拦截，使用localstorage储存登录状态，写一个拦截组件，由他根据登录状态来判断是否跳转到登录。 import React from 'react' import { N…

react请求数据统一处理（axios） wangweiruning的博客 8249
开发中为了方便开发和便于维护，我们将所有的请求统一处理。这样可以提高我们的开发效率，而且便于后期的维护。 在src目录下新建文件夹api，创建…

【Android Jetpack】NavigationMenuView的简单封装 最新发布 梦否 101
文章目录1. 前言2. NavigationView学习2.1 结构3. 思路4. 代码： 1. 前言 现需要做一个目录的页面，虽然可以使用另起一个Activity来解决，但是却做不出…

React Native一些自己封装的react navigation导航器常用的工具函数 n e w x c 75
使用方法 ， 创建一个NavigationUtil.js的文件，把代码复制过去 import React, {Component} from 'react'; class NavigationUtil extends Component { /** *跳…

React native View 事件拦截与处理 dabusidede的博客 1430
ReactNative事件处理流程 在事件处理中，每个事件只能由一个组件处理，如组件C处理了事件，那么AB就不会处理这个事件了 示例：如果发送一个触摸…

react中使用axios拦截器 北鼻娃的博客 839
写在开头：项目中每次切换页面时都在全局的dva里面监听路由切换，调用一个checkToken方法，token失效就跳转登录页。但项目中的接口大部分都需要…

react navigation 5.x 如何在UI组件之外使用进行跳转 qq_38116108的博客 626
有时，需要从无法访问navigation的地方（例如Redux中间件, request等）触发导航操作。在这种情况下，可以从导航容器中存储导航ref。 如果你是在子…

React Navigation源代码阅读：createNavigationContainer.js Details Inside Spring 1243
import React from 'react'; import {Linking} from 'react-native'; import {BackHandler} from './PlatformHelpers'; import NavigationActions from './Navigation…

react通过react-router-dom拦截实现登录验证 热门推荐 przlc的博客 1万+
在使用react开发项目中，有些页面需要登录之后才能访问，所以需要进行拦截，此处分享采用react-router-dom v4+redux+redux-saga…-mobile+axios…