脅首页 (http://www.lcode.org/) » React Native (http://www.lcode.org/category/react-native-zong/) » React Native技术文章 (http://www.lcode.org/category/react-native-zong/react-native/) » 正文 译|React Native中的图片缓存详解(二) **2017/03/23** □React Native技术文章(http://www.lcode.org/category/react-nativezong/react-native/) ▲ Sky、清 (http://www.lcode.org/author/admin/) 尊重版权,未经授权不得转载 本项目来自wcandillon (https://github.com/wcandillon) 文章来自江清清的技术专 栏(http://www.lcode.org (http://www.lcode.org/)) 原文地址(需要翻墙):https://hackernoon.com/image-caching-in-react-native-96d8df33ca84#.uaznyr11h (https://hackernoon.com/image-caching-in-reactnative-96d8df33ca84#.uaznyr11h) 1.基本介绍 我最近正在开发一个React Native项目,同时想要给大家分享一个方法库,可以在 图片显示的时候更加的平滑,以此来提升用户体验。该使用的库就是reactnative-img-cache (https://github.com/wcandillon/react-native-img-cache) 项目 实现的效果如下: React Native交流10群:157867561,欢迎各位大牛,React Native技术爱好者加入 交流!同时博客右侧欢迎微信扫描关注订阅号,移动技术干货,精彩文章技术推送! 以上的动态效果中,会有头像,列表上面的图片等多种图片类型,在本项目中采 用 2.用户上传的图片处理 针对用户自定义生成的图片(例如:app的用户上传的图片),我们有两种缓存策 略:immmutable和mutable。immutable缓存策略用于图片地址不变的情况,反正 mutable用于图片地址经常变化的情况。但是我们为什么同时需要这两种策略 呢?让我们来看下面的用户配置项。 如上界面效果:当我们现实用于资料头像的时候,该图片通过后台服务器提供。 然后当我们需要显示列表中的图像信息的时候,只需要显示用户头像的缩略图即 可,并不需要发送额外的请求到服务器。 不可变的图片需要依靠后台服务器提供的最新图片的URL地址,在上面个人头像 的案例中,后台可能会返回如下的信息: profile: { pictures: {
 large: "http://path-to-image/uid.jpg (http://p
ath-to-image/uid.jpg)",
 medium: "http://path-to-image/uid.jpg (http://
path-to-image/uid.jpg)",
 small: "http://path-to-image/uid.jpg (http://p
ath-to-image/uid.jpg)" } } 缓存这些头像图片是非常简单便捷的,通过图片URL地址的哈希串号,我们可以 检测本地缓存文件中是否存在该图片,然后进行相关处理。下面就是采用reactnative-fetch-blob (https://www.npmjs.com/package/react-native-fetch-blob)库下 载图片,然后进行缓存处理的实例。 import RNFetchBlob from "react-native-fetch-blob"; const SHA1 = require("crypto-js/sha1"); export default class ImageCache { static function get(uri: string): Promise<string> const path = RNFetchBlob.fs.dirs.CacheDir + "\_i
mmutable\_images/" + SHA1(uri) + ".jpg"; return RNFetchBlob.fs.exists(path).then(exists => { if(exists) { return path; } else { return RNFetchBlob.config({ path }) .fetch("GET", uri,
.then(() => path); 3 }); } 针对可变的图片,我们需要实现一个observable接口。针对APP应用中的每一 URL,该监听着会观察到在本地缓存中的图片当前的路径。如果图片被下载,那 么该观察者就会收到图片的本地当前路径。 type ImageCacheObserver = (path: string) => void; interface IImageCacheObservable { on(uri: string, observer: ImageCacheObserver); dispose(observer: ImageCacheObserver); interface IImageCache extends IImageCacheObservable { \* When busting an URI, the new local path of the image will be provided to all its observers bust(uri: string); 下面的代码片段显示了如何在React Native组件实现IIMageCache的方法。 interface CachedImageProps { uri: string; style?: React.ImageStyle; } @observer export class CachedImage extends Component<CachedIm</pre> ageProps, void> { @observable private \_ \_path; private handler = path => this.path = path; @computed get path() { return this.\_path; }
set path(path: string) { return this.\_path = pa th; } observe(uri: string) {
 // We start observing the cache ImageCache.on(uri, handler): } componentWillMount() { observe(this.props.uri); } componentWillReceiveProps(nextProps: CachedImag eProps) { observe(nextProps.uri); componentWillUnmount() {
 // We stop observing the cache ImageCache.dispose(handler); } render() { const {style} = this.props; return <Image style={style}</pre> source={{ uri: this.store.pat h }}>{this.props.children}</Image>; 实现IlmageCache接口类,可以实现同时不下载相同的图片。同时在接口方法中 添加一个额外的immutable的参数来进行设置缓存策略。 interface IImageCacheObservable { // The immutable parameter indicates which caching strategy on(uri: string, observer: ImageCacheObserver, immu table: boolean); dispose(observer: ImageCacheObserver); 针对具体的ImageCache的实现效果代码可以在react-native-img-cache (https://github.com/wcandillon/react-native-img-cache)库中找到 3.静态资源图片处理 针对静态图片资源,同时这边也提供一个小小的react-native-static-images (https://github.com/wcandillon/react-native-static-images)库用来扫描应用的图 片文件夹,然后生成包含应用中所有的静态图片资源的类。可以看如下一个实 例: \* Generated by https://github.com/wcandillon/reactnative-static-images (https://github.com/wcandillon/ react-native-static-images) export default class Images { static readonly resetPassword = require("../publ ic/images/ResetPas mages/ResetPassword.jpg");
static readonly signIn = require("../public/imag es/SignIn.jpg" static readonly signUp = require("../public/imag es/SignUp.jpg"); 但是如果用户使用的是旧版本的应用,后台可能会引用到部分静态资源,但是这 些图片没有在已安装的应用中。如果是这种情况,那我们就必须下载远程图片 了,具体做法如下: export default class { static getImage(id: string): any { We check if the image is present within the app bundle const entry: any = \_.get(Images, \_.camelCas e(id)); if (entry !== undefined) { return entry; } else that image return { uri: `\${StaticDomain}/\${id}` } } } React Native交流10群:157867561,欢迎各位大牛,React Native技术爱好者加入 交流!同时博客右侧欢迎微信扫描关注订阅号,移动技术干货,精彩文章技术推送! 尊重原创,转载请注明:From Sky丶清(http://www.lcode.org (http://www.lcode.org/) ) 侵权必究! 关注我的订阅号(codedev123),每天推送分享移动开发技术 (Android/iOS),React Native技术文章,项目管理,程序猿日常点滴以及精品技术资 讯文章(欢迎关注,精彩第一时间推送)。 (http://lookcodewordpress.stor.sinaapp.com/uploads/2016/01/qrcode\_for\_gh\_c6208faa1f42\_430. 关注我的微博,可以获得更多精彩内容 (http://weibo.com/u/1855428195?s=6uyXnP) (http://www.jiathis.com/share/) 0 优秀人才不缺工作机会,只缺适合自己的好机会。但是他们往往没有 精力从海量机会中找到最适合的那个。 100offer 会对平台上的人才和 企业进行严格筛选,让「最好的人才」和「最好的公司」相遇。 注册 100offer, 谈谈你对下一份工作的期待。一周内, 收到 5-10 个满足你 要求的好机会!点击了解... (https://cn.100offer.com/how-it-works/? utm\_source=icode&utm\_medium=display&utm\_campaign=icode&utm\_co 1000ffer 在这里,只看满足你要求的好机会 (https://cn.100offer.com/how-it-works/? utm\_source=icode&utm\_medium=display&utm\_campaign=icode&utm\_content=find #react native (http://www.lcode.org/tag/react-native-2/) #react native image cache (http://www.lcode.org/tag/react-native-image-cache/) #react native redux (http://www.lcode.org/tag/react-native-redux/) #react native图片异步加载 (http://www.lcode.org/tag/reactnative%e5%9b%be%e7%89%87%e5%bc%82%e6%ad%a5%e5%8a%a0%e8%bd%bd/) #react native图片缓存 (http://www.lcode.org/tag/reactnative%e5%9b%be%e7%89%87%e7%bc%93%e5%ad%98/) #React Native开源项目 (http://www.lcode.org/tag/reactnative%e5%bc%80%e6%ba%90%e9%a1%b9%e7%9b%ae/) #react native源码分析 (http://www.lcode.org/tag/reactnative%e6%ba%90%e7%a0%81%e5%88%86%e6%9e%90/) #react native缓存库 (http://www.lcode.org/tag/reactnative%e7%bc%93%e5%ad%98%e5%ba%93/) 转载请注明来源:译|React Native中的图片缓存详解(二) (http://www.lcode.org/react-native-img-cache-yiwen/) - 江清清的技术专栏 (http://www.lcode.org) 上一篇 (http://www.lcode.org/react-native-hot-update/) 你可能也喜欢: React Native Image Cache(图片缓存库模块)详解(一) (http://www.lcode.org/react-native-image-cache/) • 可直接用于React Native开发各种加密标准的JavaScript库 (MD5,AES,DES,SHA...) (http://www.lcode.org/react-native-crypto-js/) • React Native 详细实现热部署、增量差异化热更新 (http://www.lcode.org/react-native-hot-update/) • React Native开发技术周报Issue#21-时隔半年,继续更新周报~ (http://www.lcode.org/react-native-week-issue21/) • React Native开源特效动画封装库模块(lottie-react-native) (http://www.lcode.org/lottie-react-native/) Copyright © 2016 江清清的技术专栏 | Theme By Icode.org (http://www.lcode.org) (http://www.51.la/?19137953)