

百度地图与react-native

1.环境搭建与安装

1.安装

```
react-native init BaiDuMapDemo (初始化一个项目)
```

//在项目的根目录下:

```
npm install react-native-baidu-map --save
```

2.react-native启动

遇到adb报错cannot bind listener: Operation not permitted的解决方案:

[android - adb reverse failing with Operation Not Permitted error - Stack Overflow](#)

```
sudo sdb kill server
```

```
sudo adb root
```

启动步骤:

```
react-native start
```

```
react-native run-android
```

2.百度ak码的获取

1.登陆百度官网，注册账号

2.进入控制台创建应用

3.开发版sh1（是第二个）密码是android

```
cd .android
```

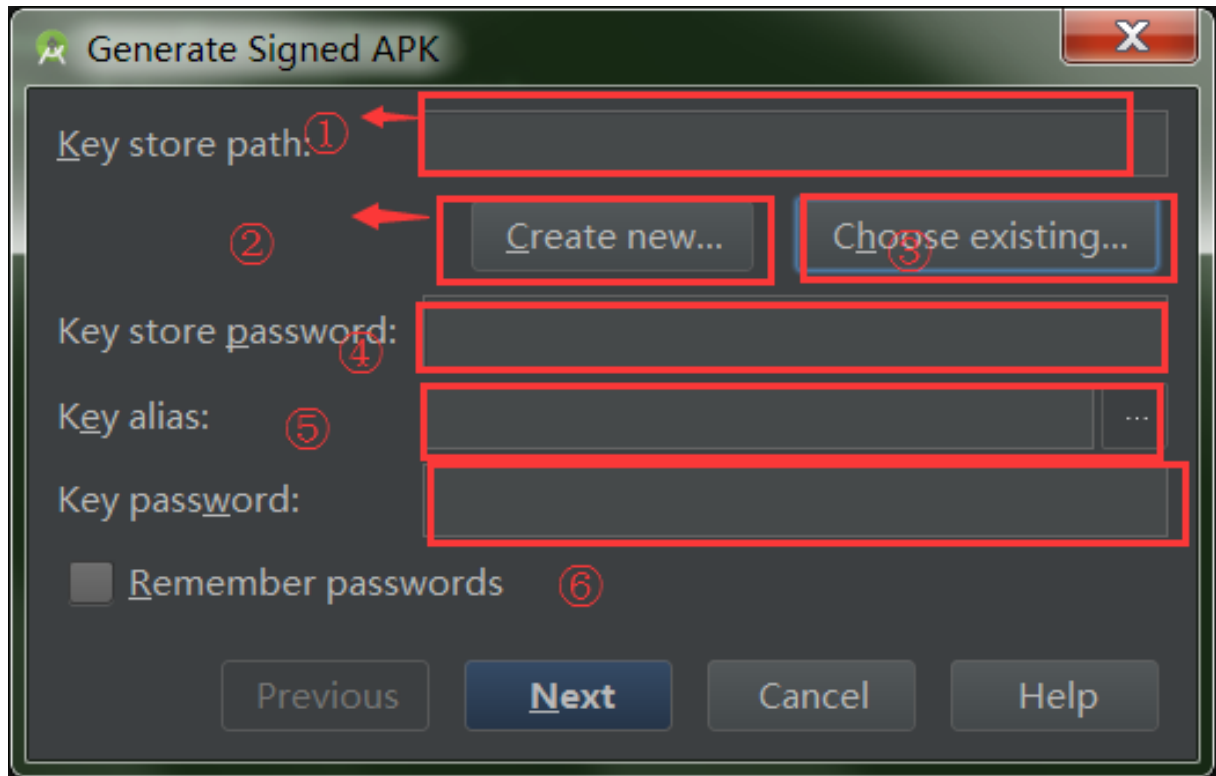
```
keytool -list -v -keystore debug.keystore;
```

显示的sh1就是开发版

4.发布版sh1（是第一个）

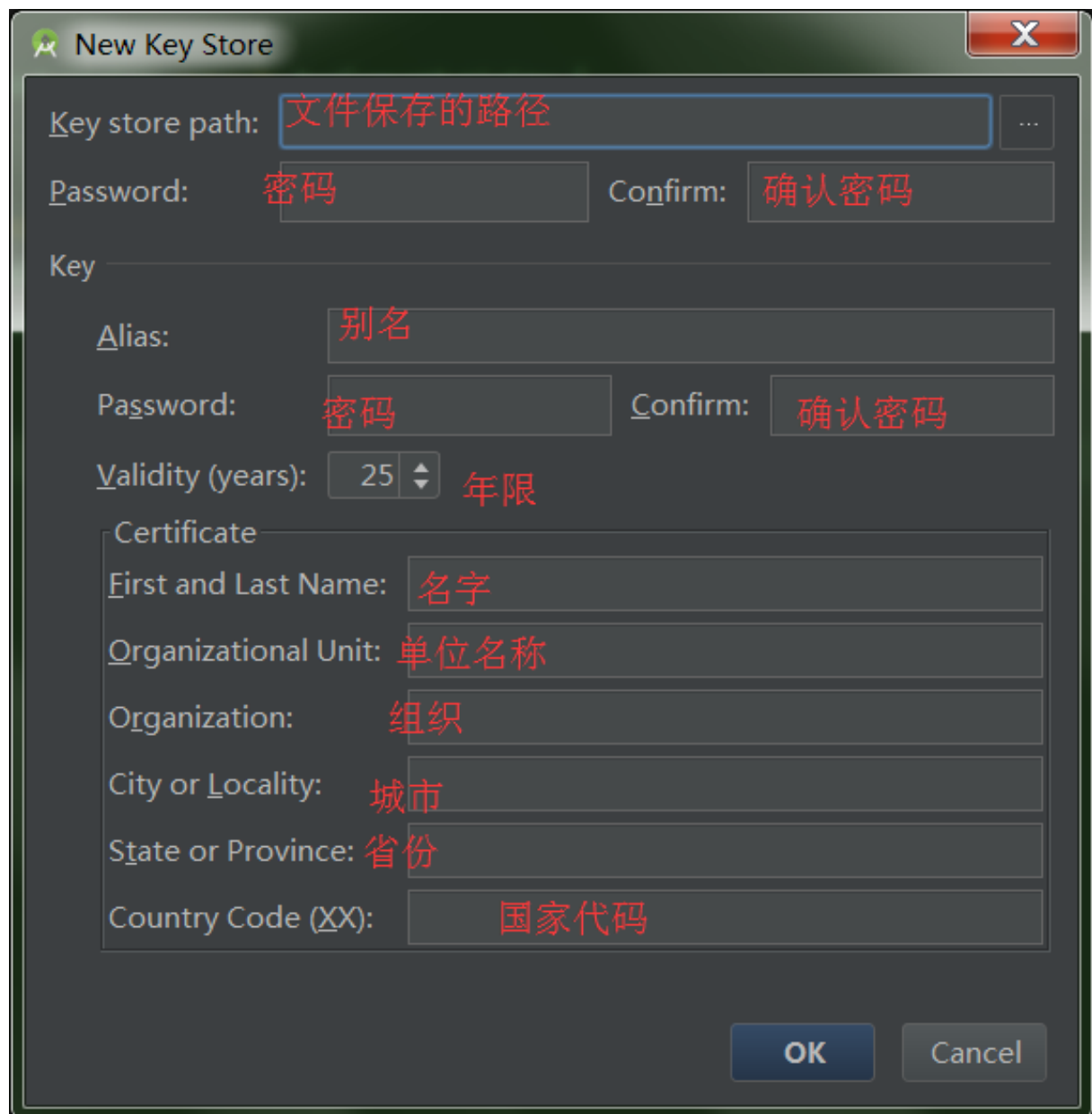
在android-studio的build-generate signed apk

打开后



点create new，就是生成新jks

另一种生成jks的方式：[Android jks 签名文件 生成 - lipeil - 博客园](#)

A screenshot of the 'New Key Store' dialog box. The dialog has a title bar with a green icon and a red 'X' button. The main area contains several input fields with red Chinese labels. The 'Key store path' field is at the top, followed by 'Password' and 'Confirm' fields. Below these is a 'Key' section with an 'Alias' field, another 'Password' and 'Confirm' pair, and a 'Validity (years)' field with a spinner set to 25. The bottom section is titled 'Certificate' and contains fields for 'First and Last Name', 'Organizational Unit', 'Organization', 'City or Locality', 'State or Province', and 'Country Code (XX)'. At the bottom right are 'OK' and 'Cancel' buttons.

New Key Store

Key store path: 文件保存的路径

Password: 密码 Confirm: 确认密码

Key

Alias: 别名

Password: 密码 Confirm: 确认密码

Validity (years): 25 年限

Certificate

First and Last Name: 名字

Organizational Unit: 单位名称

Organization: 组织

City or Locality: 城市

State or Province: 省份

Country Code (XX): 国家代码

OK Cancel

Alias要记住等下要填写， keystorepath找一个自己能找到的位置， 举例：
userwang/keystore1.jks （keystore1.jks是你想新建的jks的名称）
密码全部设置成一样的很简单的

New Key Store

Key store path: ministrator.WIN-UUEL82AF03O\android\BaiDuDT.jks ...

Key Password: Confirm: 确认密码

Alias: 别名 Password: Confirm: 确认密码

Validity (years): 25

Certificate

First and Last Name: http://blog.csdn.net/jack_dream_li

Organizational Unit:

Organization:

City or Locality: beijing

State or Province:

Country Code (XX):

可自己任意填写

OK Cancel

创建成功后

进入该文件夹，终端输入keytool -list -v -keystore 文件名.jks，即可获取

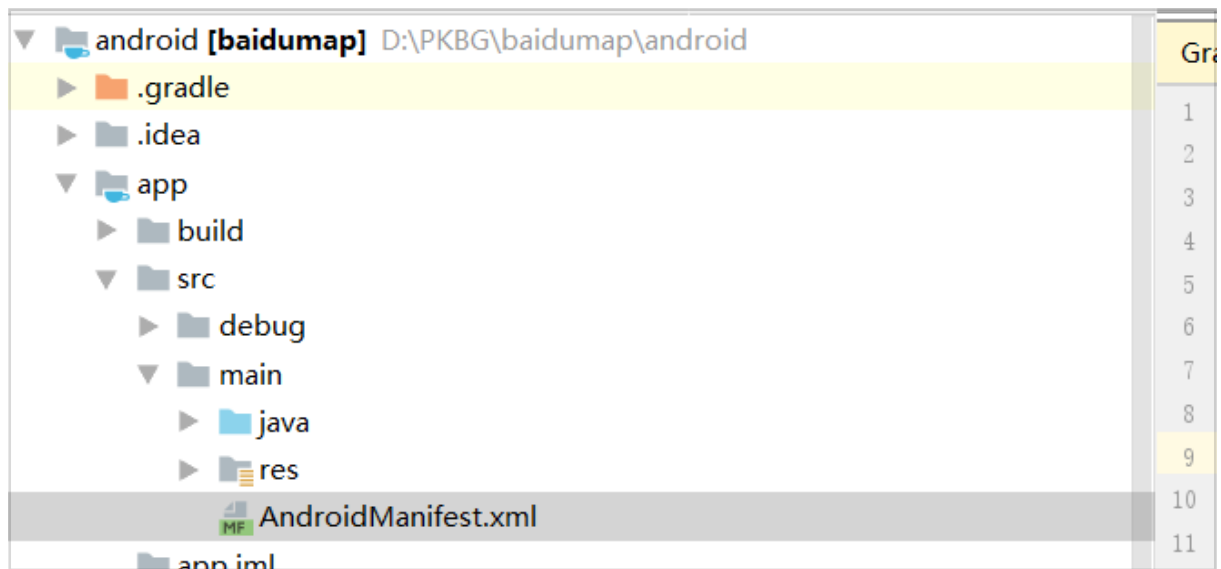
参考网站：https://blog.csdn.net/jack_dream_li/article/details/53606799

5.包名：见官网文档即可 在gradle里的applicationid复制即可

3.项目中需要修改的内容

安卓权限获取

在android-app-src-main-res-androidmanifest.xml中



加入：

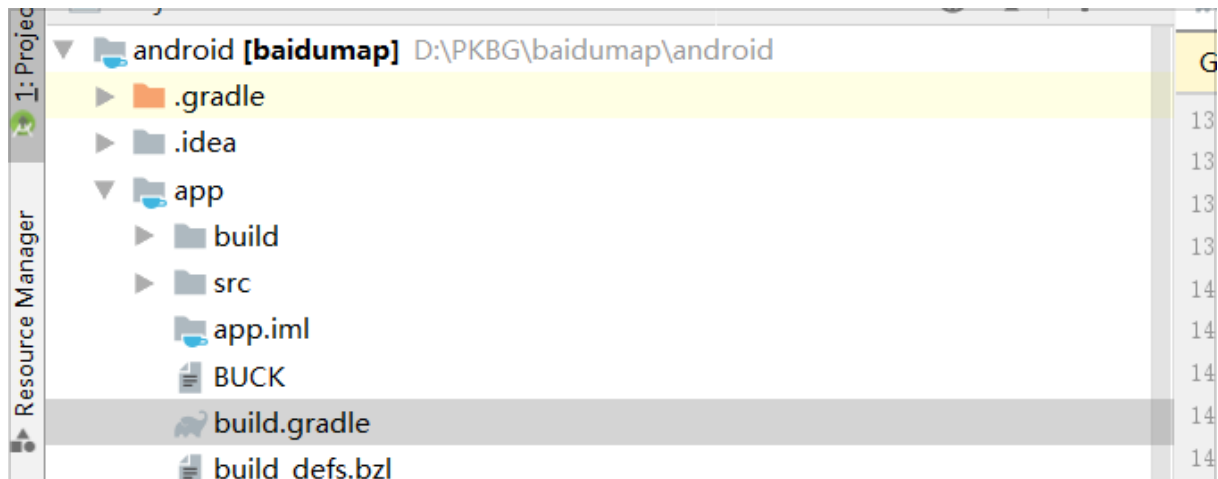
```
<uses-permission android:name="android.permission.INTERNET" />
<uses-permission android:name="android.permission.SYSTEM_ALERT_WINDOW"/>
<!-- 这个权限用于进行网络定位-->
<uses-permission android:name="android.permission.ACCESS_COARSE_LOCATION"></uses-permission>
<!-- 这个权限用于访问GPS定位-->
<uses-permission android:name="android.permission.ACCESS_FINE_LOCATION"></uses-permission>
<!-- 用于访问wifi网络信息，wifi信息会用于进行网络定位-->
<uses-permission android:name="android.permission.ACCESS_WIFI_STATE"></uses-permission>
<!-- 获取运营商信息，用于支持提供运营商信息相关的接口-->
<uses-permission android:name="android.permission.ACCESS_NETWORK_STATE"></uses-permission>
<!-- 这个权限用于获取wifi的获取权限，wifi信息会用来进行网络定位-->
<uses-permission android:name="android.permission.CHANGE_WIFI_STATE"></uses-permission>
<!-- 用于读取手机当前的状态-->
<uses-permission android:name="android.permission.READ_PHONE_STATE"></uses-permission>
<!-- 写入扩展存储，向扩展卡写入数据，用于写入离线定位数据-->
<uses-permission android:name="android.permission.WRITE_EXTERNAL_STORAGE"></uses-permission>
<!-- 访问网络，网络定位需要上网-->
<!-- SD卡读取权限，用户写入离线定位数据-->
```

```
<uses-permission android:name="android.permission.MOUNT_UNMOUNT_FILESYSTEMS"></uses-permission>
```

```
<meta-data  
    android:name="com.baidu.lbsapi.API_KEY"  
    android:value="G4gjYknjdGw01TiU682GQbK9kdPduXgV"/>
```

替换jks信息

在build.gradle中

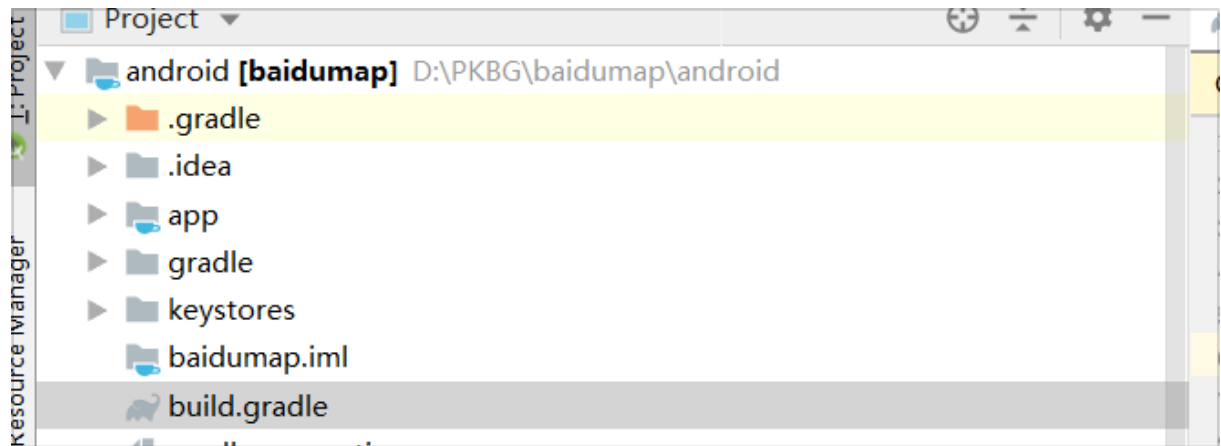


找到signingconfigs, 把里面的storefile换成刚才申请百度ak时候自己生成的jks的信息, 密码和alias

```
signingConfigs {  
    debug {  
        storeFile file('shy_debug.jks')  
        storePassword 'android'  
        keyAlias 'shy_debug.jks'  
        keyPassword 'android'  
    }  
    /*debug {  
        storeFile file('../keystores/debug.keystore')  
        storePassword "android"  
        keyAlias "androiddebugkey"  
        keyPassword "android"  
    }*/  
}
```

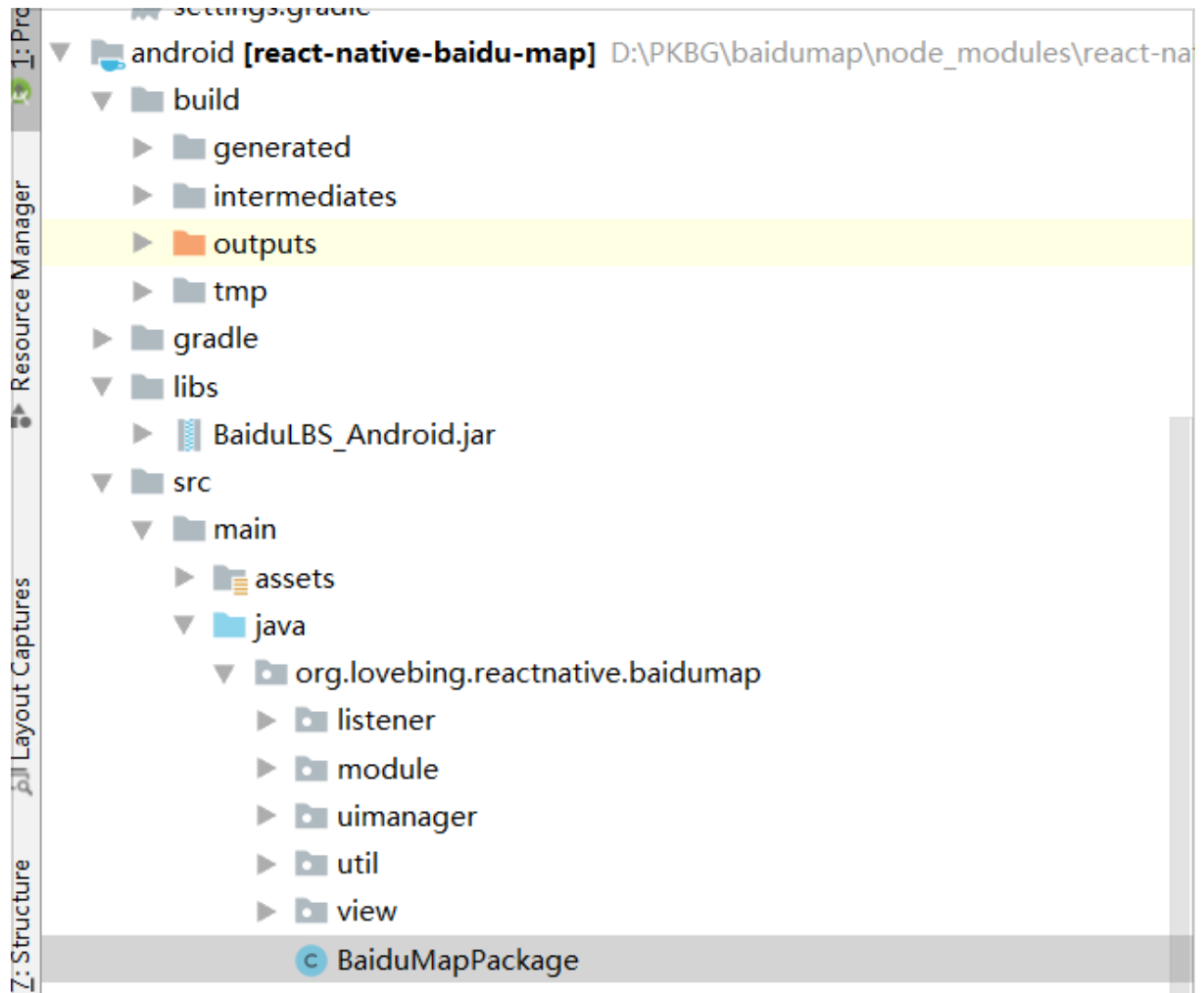
修改最低sdk版本

在build.gradle中



```
buildscript {  
    ext {  
        buildToolsVersion = "28.0.3"  
        minSdkVersion = 19  
        compileSdkVersion = 28  
        targetSdkVersion = 28  
        supportLibVersion = "28.0.0"  
    }  
}
```

baidumappackage 代码修改, node-modules中baidumap文件



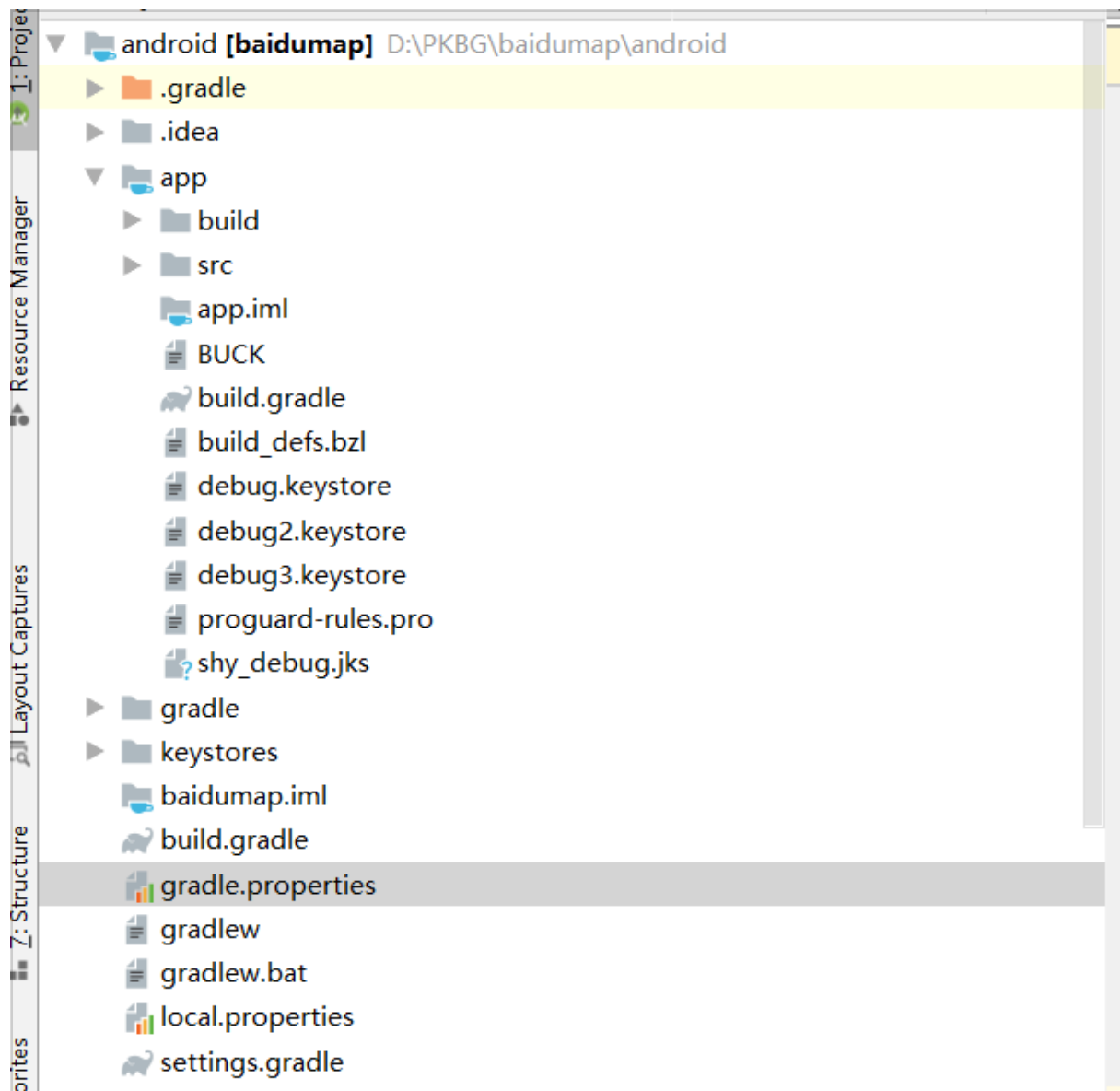
在图示文件中，加入

```
@MainThread
protected void init(ReactApplicationContext reactContext) {
    if (Looper.myLooper()==null){

        Looper.prepare();

    }
    SDKInitializer.initialize(reactContext.getApplicationContext());
}
```

grade.properties



```
TEST_RELEASE_STORE_FILE=debug1.keystore
TEST_RELEASE_KEY_ALIAS=my-key-alias
TEST_RELEASE_STORE_PASSWORD=android
TEST_RELEASE_KEY_PASSWORD=android
```

具体功能详见app.js

```
import React, { Component } from 'react';
import {
  StyleSheet,
  Text,
  View,
  ScrollView,
  Platform,
```

```

    FlatList,
  } from 'react-native';

import { MapView, MapTypes, Geolocation, Overlay } from 'react-native-baidu-map';
const { Marker } = Overlay;

import Dimensions from 'Dimensions';
const { width,height } = Dimensions.get('window');

export default class BaiduMapDemo extends Component {
  constructor() {
    super();
    this.state = {
      zoomControlsVisible: true,
      trafficEnabled: false,
      baiduHeatMapEnabled: false,
      mapType: MapTypes.NORMAL,
      zoom: 15,
      center: {
        longitude: 113.896198,
        latitude: 22.959144,
      },
      clickMessage: '', //空白区域信息
      poiMessage: '', //已有点信息
      clickMessageData: [], //空白区域附近数据
      poiMessageData: [], //已有点附近数据
    };
  }

  componentDidMount() {
    Geolocation.getCurrentPosition()
      .then(data => {
        // console.log(data)
        // this.setState({
        //   center: {
        //     longitude: data.longitude,
        //     latitude: data.latitude
        //   },

```

[illegible]

```

        })
        if(Platform.OS === 'ios') {
            this.setState({
                clickMessage: res.district +
res.streetName,
            })
        }else {
            this.setState({
                clickMessage: e.name,
            })
        }
    })
    .catch(err => {
        console.log(err)
    })
}

onMapPoiClick={(e) => { //地图已有点点击
    Geolocation.reverseGeoCode(e.latitude,e.longitude)
        .then(res => {
            this.setState({
                center: {
                    longitude: e.longitude,
                    latitude: e.latitude,
                },
                poiMessage: e.name,
                poiMessageData: res.poiList,
            })
        })
        .catch(err => {
            console.log(err)
        })
    }}
    style={styles.map}
>
    <Marker
        title={'title'}
        location={this.state.center}
    >

```

```

        </Marker>
    </MapView>

    <View style={styles.list}>
        <Text>地图缩放控件状态: </Text>

        {this.state.zoomControlsVisible ?
            <Text onPress={() =>
this.setState({zoomControlsVisible:false})}>显示</Text>
            :
            <Text onPress={() =>
this.setState({zoomControlsVisible:true})}>关闭</Text>
        }
    </View>
    <View style={styles.list}>
        <Text>交通线状态: </Text>

        {this.state.trafficEnabled ?
            <Text onPress={() =>
this.setState({trafficEnabled:false})}>显示</Text>
            :
            <Text onPress={() =>
this.setState({trafficEnabled:true})}>关闭</Text>
        }
    </View>
    <View style={styles.list}>
        <Text>热力图状态: </Text>

        {this.state.baiduHeatMapEnabled ?
            <Text onPress={() =>
this.setState({baiduHeatMapEnabled:false})}>显示</Text>
            :
            <Text onPress={() =>
this.setState({baiduHeatMapEnabled:true})}>关闭</Text>
        }
    </View>
    <View style={styles.list}>
        <Text>地图模式状态: </Text>

        {this.state.mapType == MapTypes.NORMAL ?
            <Text onPress={() =>
this.setState({mapType:MapTypes.SATELLITE})}>普通</Text>
            :

```

```

        <Text onPress={() =>
this.setState({mapType:MapTypes.NORMAL})}>卫星</Text>

      }
    </View>

    {/* 空白区域 */}
    <View style={styles.list}>
      <Text>地图空白区域附近信息： {this.state.clickMessage}</Text>
    </View>
    <View style={{paddingLeft: 10,marginBottom: 5,}}>
      <FlatList
        data={this.state.clickMessageData}
        keyExtractor={(item,index) => index.toString()}
        renderItem={this._renderItem}
        extraData={this.state}
      />
    </View>

    {/* 已有点 */}
    <View style={styles.list}>
      <Text>地图已有点点击信息： {this.state.poiMessage}</Text>
    </View>
    <View style={{paddingLeft: 10,marginBottom: 5,}}>
      <FlatList
        data={this.state.poiMessageData}
        keyExtractor={(item,index) => index.toString()}
        ListHeaderComponent={this._header}
        renderItem={this._renderItem}
        extraData={this.state}
      />
    </View>

    <View style={styles.list}>
      <Text onPress={() => {
        Geolocation.getCurrentPosition()
          .then(data => {
            this.setState({
              center: {
                longitude: data.longitude,

```

```

        latitude: data.latitude
      },
    })
  })
  .catch(e =>{
    console.warn(e, 'error');
  })
}}>当前位置</Text>
</View>
</ScrollView>
);
}
_header = () => {
  return (
    <Text>附近信息: </Text>
  )
}
_renderItem = (item) => {

  item = item.item;
  let H = Math.random() * 1000;
  H = Math.ceil(H);
  return (
    <View style={styles.item,{backgroundColor:`hsl(${H},50%,30%)`}}>
      <Text style={styles.itemText}>纬度: {item.latitude}</Text>
      <Text style={styles.itemText}>经度: {item.longitude}</Text>
      <Text style={styles.itemText}>地点: {item.name}</Text>
      <Text style={styles.itemText}>地址: {item.address}</Text>
    </View>
  )
}
}

const styles = StyleSheet.create({
  container: {
    flex: 1,
    backgroundColor: '#F5FCFF',
  },
  map: {

```

```
        width: width,
        height: height - 300,
        marginBottom: 5,
    },
    list: {
        flexDirection: 'row',
        paddingLeft: 10,
        marginBottom: 5,
    },
    item: {
        marginLeft: 10,
        marginRight: 10,
        marginBottom: 5,
        paddingHorizontal: 5,
        paddingVertical: 8,
    },
    itemText: {
        color: '#fff',
    }
  });
```

4.项目文档

1.百度地图ak获取结合项目内容见项目文档

<http://lbsyun.baidu.com>

2.github app.js函数文档

<https://github.com/lovebing/react-native-baidu-map>