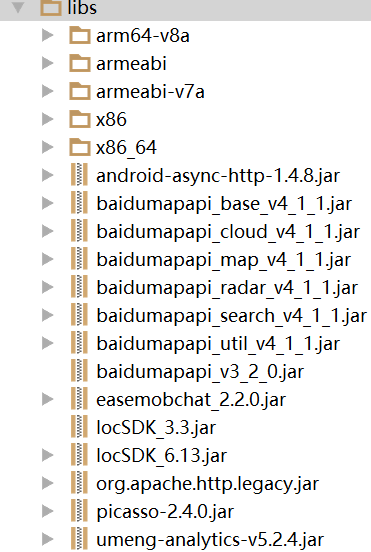
百度地图接口说明

### 若想调用百度地图的接口：

**申请秘钥-创建应用（略）**

**将需要的文件导入到libs文件中，方便方法的调用。**



**配置环境**

在Manifest中添加使用权限

添加Key:

<**meta-data  
 android:name="com.baidu.lbsapi.API\_KEY"  
 android:value="3ecea51f560650b1ed8a4b99808f52e8"** />

**Gradle.gradle里添加**

sourceSets{  
 main{  
 jniLibs.srcDirs=[**'libs'**]  
 }

**功能使用**

## 1.定位功能

1、因为项目是建在fragment中的，所以要要想实现地图的某些点击功能要implement OnclickListener**public class** MapFragment **extends** Fragment **implements** View.OnClickListener{  
 **private** Context **context**; *//上下文环境* **private** View **view**;  
 **public** Context getContext() {  
 **return context**;  
 }  
 **public void** setContext(Context context) {  
 **this**.**context** = context;  
 }  
 @Override  
 **public** View onCreateView(LayoutInflater inflater, ViewGroup container, Bundle savedInstanceState) {

获取百度地图  
 SDKInitializer.*initialize*(**context**.getApplicationContext());  
 **view** = inflater.inflate(R.layout.***activity\_location***, container, **false**);  
 **imbtn\_add** = (ImageButton) **view**.findViewById(R.id.***id\_add2***);  
 **imbtn\_add**.setOnClickListener(**this**);  
 **return view**;  
 }  
 **2、设置定位相关信息**LocationClient **mLocClient**;  
 **public** MyLocationListenner **myListener** = **new** MyLocationListenner();  
 **private** MyLocationConfiguration.LocationMode **mCurrentMode**;  
 BitmapDescriptor **mCurrentMarker**;  
 **private static final int *accuracyCircleFillColor*** = 0xAAFFFF88;  
 **private static final int *accuracyCircleStrokeColor*** = 0xAA00FF00;  
  
 MapView **mMapView**;  
 BaiduMap **mBaiduMap**;  
  
 设置UI相关信息

RadioGroup.OnCheckedChangeListener **radioButtonListener**;  
 Button **requestLocButton**;  
 **boolean isFirstLoc** = **true**; *// 是否首次定位* @Override  
 **public void** onActivityCreated(Bundle savedInstanceState){  
 **super**.onActivityCreated(savedInstanceState);  
  
 **requestLocButton** = (Button)**view**.findViewById(R.id.***button1***);  
 **mCurrentMode** = MyLocationConfiguration.LocationMode.***NORMAL***;  
 **requestLocButton**.setText(**"普通"**);  
 View.OnClickListener btnClickListener = **new** View.OnClickListener() {  
 **public void** onClick(View v) {  
 **switch** (**mCurrentMode**) {  
 **case *NORMAL***:  
 **requestLocButton**.setText(**"跟随"**);  
 **mCurrentMode** = MyLocationConfiguration.LocationMode.***FOLLOWING***;  
 **mBaiduMap** .setMyLocationConfigeration(**new** MyLocationConfiguration(  
 **mCurrentMode**, **true**, **mCurrentMarker**));  
 **break**;  
 **case *COMPASS***:  
 **requestLocButton**.setText(**"普通"**);  
 **mCurrentMode** = MyLocationConfiguration.LocationMode.***NORMAL***;  
 **mBaiduMap** .setMyLocationConfigeration(**new** MyLocationConfiguration(  
 **mCurrentMode**, **true**, **mCurrentMarker**));  
 **break**;  
 **case *FOLLOWING***:  
 **requestLocButton**.setText(**"罗盘"**);  
 **mCurrentMode** = MyLocationConfiguration.LocationMode.***COMPASS***;  
 **mBaiduMap** .setMyLocationConfigeration(**new** MyLocationConfiguration(  
 **mCurrentMode**, **true**, **mCurrentMarker**));  
 **break**;  
 **default**:  
 **break**;  
 }  
 }  
 };  
 **requestLocButton**.setOnClickListener(btnClickListener);  
  
 RadioGroup group = (RadioGroup) **view**.findViewById(R.id.***radioGroup***);  
 **radioButtonListener** = **new** RadioGroup.OnCheckedChangeListener() {  
 @Override  
 **public void** onCheckedChanged(RadioGroup group, **int** checkedId) {  
 **if** (checkedId == R.id.***defaulticon***) {  
 *// 传入null则，恢复默认图标* **mCurrentMarker** = **null**;  
 **mBaiduMap** .setMyLocationConfigeration(**new** MyLocationConfiguration(  
 **mCurrentMode**, **true**, **null**));  
 }  
 **if** (checkedId == R.id.***customicon***) {  
 *// 修改为自定义marker* **mCurrentMarker** = BitmapDescriptorFactory  
 .*fromResource*(R.drawable.***feiji***);  
 **mBaiduMap** .setMyLocationConfigeration(**new** MyLocationConfiguration(  
 **mCurrentMode**, **true**, **mCurrentMarker**,  
 ***accuracyCircleFillColor***, ***accuracyCircleStrokeColor***));  
 }  
 }  
 };  
 group.setOnCheckedChangeListener(**radioButtonListener**);  
 *// 地图初始化* **mMapView** = (MapView)**view**.findViewById(R.id.***bmapView***);  
 **mBaiduMap** = **mMapView**.getMap();  
 *// 开启定位图层* **mBaiduMap**.setMyLocationEnabled(**true**);  
 *// 定位初始化* **mLocClient** = **new** LocationClient(**context**);  
 **mLocClient**.registerLocationListener(**myListener**);  
 LocationClientOption option = **new** LocationClientOption();  
 option.setOpenGps(**true**); *// 打开gps* option.setCoorType(**"bd09ll"**); *// 设置坐标类型* option.setScanSpan(1000);  
 **mLocClient**.setLocOption(option);  
 **mLocClient**.start();  
 }  
  
  
 **public void** onClick(View v) {  
 **switch** (v.getId()){  
 **case** R.id.***id\_add2***:  
 Intent intent = **new** Intent(getActivity(),Pop\_menu.**class**);  
 startActivity(intent);  
 }  
 }  
**3、设定定位SDK监听函数****public class** MyLocationListenner **implements** BDLocationListener{  
  
 @Override  
 **public void** onReceiveLocation(BDLocation location) {  
 *// map view 销毁后不在处理新接收的位置* **if** (location == **null** || **mMapView** == **null**) {  
 **return**;  
 }  
 MyLocationData locData = **new** MyLocationData.Builder()  
 .accuracy(location.getRadius())  
 *// 此处设置开发者获取到的方向信息，顺时针0-360* .direction(100).latitude(location.getLatitude())  
 .longitude(location.getLongitude()).build();  
 **mBaiduMap**.setMyLocationData(locData);  
 **if** (**isFirstLoc**) {  
 **isFirstLoc** = **false**;  
 LatLng ll = **new** LatLng(location.getLatitude(),  
 location.getLongitude());  
 MapStatus.Builder builder = **new** MapStatus.Builder();  
 builder.target(ll).zoom(18.0f);

**4、布局文件activity\_location中要添加**  
<**com.baidu.mapapi.map.MapView  
 android:layout\_marginTop="50dp"  
 android:id="@+id/bmapView"  
 android:layout\_width="fill\_parent"  
 android:layout\_height="match\_parent"  
 android:clickable="true"** />

**5、因为在MapFragment中没有办法实现**SDKInitializer.*initialize*(**context**.getApplicationContext());

所以在MainActivity中还要添加

**case** R.id.***btn\_map***:  
 **index** = 3;  
 **mapFragment**.setContext(getApplicationContext());  
 **break**;

## 2.POI检索功能 1、在java文件中添加 PoiOverlay OverlayManager类public class PoiSearchDemo extends FragmentActivity implements OnGetPoiSearchResultListener, OnGetSuggestionResultListener { private PoiSearch mPoiSearch = null; private SuggestionSearch mSuggestionSearch = null; private BaiduMap mBaiduMap = null; private List<String> suggest; private ImageButton turn; 2、搜索关键字输入窗口private AutoCompleteTextView editCity = null; private AutoCompleteTextView keyWorldsView = null; private ArrayAdapter<String> sugAdapter = null; private int loadIndex = 0; LatLng center = new LatLng(39.92235, 116.380338); int radius = 100; LatLng southwest = new LatLng( 39.92235, 116.380338 ); LatLng northeast = new LatLng( 39.947246, 116.414977); LatLngBounds searchbound = new LatLngBounds.Builder().include(southwest).include(northeast).build(); int searchType = 0; *// 搜索的类型，在显示时区分* @Override public void onCreate(Bundle savedInstanceState) { super.onCreate(savedInstanceState); SDKInitializer.*initialize*(getApplicationContext()); setContentView(R.layout.*activity\_poisearch*); turn=(ImageButton) findViewById(R.id.*turn*); turn.setOnClickListener(new View.OnClickListener() { @Override public void onClick(View v) { Utils.*flag*=4; Intent i=new Intent(); i.setClass(PoiSearchDemo.this,MainActivity.class); i.addFlags(Intent.*FLAG\_ACTIVITY\_NEW\_TASK*); startActivity(i); } }); *// 初始化搜索模块，注册搜索事件监听* mPoiSearch = PoiSearch.*newInstance*(); mPoiSearch.setOnGetPoiSearchResultListener(this); *// 初始化建议搜索模块，注册建议搜索事件监听* mSuggestionSearch = SuggestionSearch.*newInstance*(); mSuggestionSearch.setOnGetSuggestionResultListener(this); editCity = (AutoCompleteTextView) findViewById(R.id.*city*); keyWorldsView = (AutoCompleteTextView) findViewById(R.id.*searchkey*); sugAdapter = new ArrayAdapter<String>(this, android.R.layout.*simple\_dropdown\_item\_1line*); keyWorldsView.setAdapter(sugAdapter); keyWorldsView.setThreshold(1); mBaiduMap = ((SupportMapFragment) (getSupportFragmentManager() .findFragmentById(R.id.*map*))).getBaiduMap(); 3、 当输入关键字变化时，动态更新建议列表 keyWorldsView.addTextChangedListener(new TextWatcher() { @Override public void afterTextChanged(Editable arg0) { } @Override public void beforeTextChanged(CharSequence arg0, int arg1, int arg2, int arg3) { } @Override public void onTextChanged(CharSequence cs, int arg1, int arg2, int arg3) { if (cs.length() <= 0) { return; } */\*\* \* 使用建议搜索服务获取建议列表，结果在onSuggestionResult()中更新 \*/* mSuggestionSearch.requestSuggestion((new SuggestionSearchOption()) .keyword(cs.toString()).city(editCity.getText().toString())); } }); } @Override protected void onSaveInstanceState(Bundle outState) { super.onSaveInstanceState(outState); } @Override protected void onRestoreInstanceState(Bundle savedInstanceState) { super.onRestoreInstanceState(savedInstanceState); }响应城市内搜索按钮点击事件 public void searchButtonProcess(View v) { searchType = 1; String citystr = editCity.getText().toString(); String keystr = keyWorldsView.getText().toString(); mPoiSearch.searchInCity((new PoiCitySearchOption()) .city(citystr).keyword(keystr).pageNum(loadIndex)); } 4、响应周边搜索按钮点击事件 public void searchNearbyProcess(View v) { searchType = 2; PoiNearbySearchOption nearbySearchOption = new PoiNearbySearchOption().keyword(keyWorldsView.getText() .toString()).sortType(PoiSortType.*distance\_from\_near\_to\_far*).location(center) .radius(radius).pageNum(loadIndex); mPoiSearch.searchNearby(nearbySearchOption); } public void goToNextPage(View v) { loadIndex++; searchButtonProcess(null); }

## 3.公交线路搜索功能

1、发起检索功能：

**public void** searchButtonProcess(View v) {  
 **busLineIDList**.clear();  
 **busLineIndex** = 0;  
 **mBtnPre**.setVisibility(View.***INVISIBLE***);  
 **mBtnNext**.setVisibility(View.***INVISIBLE***);  
 **editCity** = (AutoCompleteTextView) findViewById(R.id.***city***);  
 **editSearchKey** = (AutoCompleteTextView) findViewById(R.id.***searchkey***);  
 *// 发起poi检索，从得到所有poi中找到公交线路类型的poi，再使用该poi的uid进行公交详情搜索* **mSearch**.searchInCity((**new** PoiCitySearchOption()).city(  
 **editCity**.getText().toString())  
 .keyword(**editSearchKey**.getText().toString()));  
}

**2、使用RouteOverlay在地图上绘制 同时展示如何浏览路线节点并弹出泡泡**

**public void** nodeClick(View v) {  
  
 **if** (**nodeIndex** < -1 || **route** == **null** || **nodeIndex** >= **route**.getStations().size()) {  
 **return**;  
 }  
 TextView popupText = **new** TextView(**this**);  
 popupText.setBackgroundResource(R.drawable.***popup***);  
 popupText.setTextColor(0xff000000);  
  
 *// 上一个节点* **if** (**mBtnPre**.equals(v) && **nodeIndex** > 0) {  
 *// 索引减* **nodeIndex**--;  
 }  
 *// 下一个节点* **if** (**mBtnNext**.equals(v) && **nodeIndex** < (**route**.getStations().size() - 1)) {  
 *// 索引加* **nodeIndex**++;  
 }  
 **if** (**nodeIndex** >= 0) {  
 *// 移动到指定索引的坐标* **mBaiduMap**.setMapStatus(MapStatusUpdateFactory.*newLatLng*(**route** .getStations().get(**nodeIndex**).getLocation()));  
 *// 弹出泡泡* popupText.setText(**route**.getStations().get(**nodeIndex**).getTitle());  
 **mBaiduMap**.showInfoWindow(**new** InfoWindow(popupText, **route**.getStations()  
 .get(**nodeIndex**).getLocation(), 10));  
 }  
}

## 4.线路检索

1. **发起路线规划搜索示例**
2. **public void** searchButtonProcess(View v) {  
    *// 重置浏览节点的路线数据* **route** = **null**;  
    **mBtnPre**.setVisibility(View.***INVISIBLE***);  
    **mBtnNext**.setVisibility(View.***INVISIBLE***);  
    **mBaidumap**.clear();  
    *// 处理搜索按钮响应  
    // 设置起终点信息，对于tranist search 来说，城市名无意义* PlanNode stNode = PlanNode.*withCityNameAndPlaceName*(**"北京"**, **startNodeStr**);  
    PlanNode enNode = PlanNode.*withCityNameAndPlaceName*(**"北京"**, **endNodeStr**);  
     
    *// 实际使用中请对起点终点城市进行正确的设定* **if** (v.getId() == R.id.***mass***) {  
    AutoCompleteTextView s = (AutoCompleteTextView)findViewById(R.id.***edits***);  
    AutoCompleteTextView e = (AutoCompleteTextView)findViewById(R.id.***edite***);  
    String strS = s.getText().toString();  
    String strE = e.getText().toString();  
    PlanNode stNewNode = PlanNode.*withCityNameAndPlaceName*(**"北京"**, strS);  
    PlanNode enNewNode = PlanNode.*withCityNameAndPlaceName*(**"北京"**, strE);  
    **mSearch**.drivingSearch((**new** DrivingRoutePlanOption())  
    .from(stNewNode).to(enNewNode));  
    **nowSearchType** = 1;

**}**

**}**

2、**节点浏览示例**

**public void** nodeClick(View v) {  
 LatLng nodeLocation = **null**;  
 String nodeTitle = **null**;  
 Object step = **null**;  
  
 **if** ( **nowSearchType** != 0 && **nowSearchType** != -1) {  
 *// 非跨城综合交通* **if** (**route** == **null** || **route**.getAllStep() == **null**) {  
 **return**;  
 }  
 **if** (**nodeIndex** == -1 && v.getId() == R.id.***pre***) {  
 **return**;  
 }  
 *// 设置节点索引* **if** (v.getId() == R.id.***next***) {  
 **if** (**nodeIndex** < **route**.getAllStep().size() - 1) {  
 **nodeIndex**++;  
 } **else** {  
 **return**;  
 }  
 } **else if** (v.getId() == R.id.***pre***) {  
 **if** (**nodeIndex** > 0) {  
 **nodeIndex**--;  
 } **else** {  
 **return**;  
 }  
 }