Google地图百度地图GPS经纬度偏移转换函数[JAVA]

三尺寒冰

2015-07-23 21:59 [来自QQ空间日志](http://m.qzone.com/infocenter#990688/list/blog)

有这个就不用去调用API了,还每秒不能超过50次限制....  
不多说，直接代码，Java版  
主要就是Google地图，百度地图，GPS经纬度偏移转换  
GCJ-02转换BD-09,Google地图经纬度转百度地图经纬度  
BD-09转换GCJ-02,百度转google  
WGS-84 到 GCJ-02 的转换（即 GPS 加偏）  
  
/\*\*  
\* 用于构造地图中的坐标点  
\* @author lw  
\* \*\*/  
public class Point {  
  
private double lat;// 纬度  
private double lng;// 经度  
  
public Point() {  
}  
  
public Point(double lng, double lat) {  
this.lng = lng;  
this.lat = lat;  
}  
@Override  
public boolean equals(Object obj) {  
if (obj instanceof Point) {  
Point bmapPoint = (Point) obj;  
return (bmapPoint.getLng() == lng && bmapPoint.getLat() == lat) ? true : false;  
} else {  
return false;  
}  
}  
  
public double getLat() {  
return lat;  
}  
public void setLat(double lat) {  
this.lat = lat;  
}  
public double getLng() {  
return lng;  
}  
public void setLng(double lng) {  
this.lng = lng;  
}  
  
@Override  
public String toString() {  
return "Point [lat=" + lat + ", lng=" + lng + "]";  
}  
  
}  
/\*\*  
\* 地图坐标转换 google,baidu,gps  
\* @author lw  
\* @Time 2015年4月16日18:19:16  
\*   
\* \*/  
public class CoordinateConversion {  
private static final double x\_pi = 3.14159265358979324 \* 3000.0 / 180.0;  
  
private static final double pi = 3.14159265358979324;  
private static final double a = 6378245.0;  
private static final double ee = 0.00669342162296594323;  
  
/\*\*  
\* gg\_lat 纬度  
\* gg\_lon 经度  
\* GCJ-02转换BD-09  
\* Google地图经纬度转百度地图经纬度  
\* \*/  
public static Point google\_bd\_encrypt(double gg\_lat, double gg\_lon){  
Point point=new Point();  
double x = gg\_lon, y = gg\_lat;  
double z = Math.sqrt(x \* x + y \* y) + 0.00002 \* Math.sin(y \* x\_pi);  
double theta = Math.atan2(y, x) + 0.000003 \* Math.cos(x \* x\_pi);   
double bd\_lon = z \* Math.cos(theta) + 0.0065;  
double bd\_lat = z \* Math.sin(theta) + 0.006;  
point.setLat(bd\_lat);  
point.setLng(bd\_lon);  
return point;  
}  
  
/\*\*

\* wgLat 纬度  
\* wgLon 经度  
\* BD-09转换GCJ-02  
\* 百度转google  
\* \*/  
public static Point bd\_google\_encrypt(double bd\_lat, double bd\_lon){  
Point point=new Point();  
double x = bd\_lon - 0.0065, y = bd\_lat - 0.006;   
double z = Math.sqrt(x \* x + y \* y) - 0.00002 \* Math.sin(y \* x\_pi);   
double theta =Math.atan2(y, x) - 0.000003 \* Math.cos(x \* x\_pi);   
double gg\_lon = z \* Math.cos(theta);   
double gg\_lat = z \* Math.sin(theta);   
point.setLat(gg\_lat);  
point.setLng(gg\_lon);  
return point;  
}  
  
  
  
/\*\*  
\* wgLat 纬度  
\* wgLon 经度  
\* WGS-84 到 GCJ-02 的转换（即 GPS 加偏）  
\* \*/  
public static Point wgs\_gcj\_encrypts(double wgLat, double wgLon) {  
Point point=new Point();  
if (outOfChina(wgLat, wgLon)) {  
point.setLat(wgLat);  
point.setLng(wgLon);  
return point;  
}  
double dLat = transformLat(wgLon - 105.0, wgLat - 35.0);  
double dLon = transformLon(wgLon - 105.0, wgLat - 35.0);  
double radLat = wgLat / 180.0 \* pi;  
double magic = Math.sin(radLat);  
magic = 1 - ee \* magic \* magic;  
double sqrtMagic = Math.sqrt(magic);  
dLat = (dLat \* 180.0) / ((a \* (1 - ee)) / (magic \* sqrtMagic) \* pi);  
dLon = (dLon \* 180.0) / (a / sqrtMagic \* Math.cos(radLat) \* pi);  
double lat = wgLat + dLat;  
double lon = wgLon + dLon;  
point.setLat(lat);  
point.setLng(lon);  
return point;  
}  
  
  
public static void transform(double wgLat, double wgLon, double[] latlng) {  
if (outOfChina(wgLat, wgLon)) {  
latlng[0] = wgLat;  
latlng[1] = wgLon;  
return;  
}  
double dLat = transformLat(wgLon - 105.0, wgLat - 35.0);  
double dLon = transformLon(wgLon - 105.0, wgLat - 35.0);  
double radLat = wgLat / 180.0 \* pi;  
double magic = Math.sin(radLat);  
magic = 1 - ee \* magic \* magic;  
double sqrtMagic = Math.

sqrt(magic);  
dLat = (dLat \* 180.0) / ((a \* (1 - ee)) / (magic \* sqrtMagic) \* pi);  
dLon = (dLon \* 180.0) / (a / sqrtMagic \* Math.cos(radLat) \* pi);  
latlng[0] = wgLat + dLat;  
latlng[1] = wgLon + dLon;  
}  
private static boolean outOfChina(double lat, double lon) {  
if (lon < 72.004 || lon > 137.8347)  
return true;  
if (lat < 0.8293 || lat > 55.8271)  
return true;  
return false;  
}  
private static double transformLat(double x, double y) {  
double ret = -100.0 + 2.0 \* x + 3.0 \* y + 0.2 \* y \* y + 0.1 \* x \* y + 0.2 \* Math.sqrt(Math.abs(x));  
ret += (20.0 \* Math.sin(6.0 \* x \* pi) + 20.0 \* Math.sin(2.0 \* x \* pi)) \* 2.0 / 3.0;  
ret += (20.0 \* Math.sin(y \* pi) + 40.0 \* Math.sin(y / 3.0 \* pi)) \* 2.0 / 3.0;  
ret += (160.0 \* Math.sin(y / 12.0 \* pi) + 320 \* Math.sin(y \* pi / 30.0)) \* 2.0 / 3.0;  
return ret;  
}  
private static double transformLon(double x, double y) {  
double ret = 300.0 + x + 2.0 \* y + 0.1 \* x \* x + 0.1 \* x \* y + 0.1 \* Math.sqrt(Math.abs(x));  
ret += (20.0 \* Math.sin(6.0 \* x \* pi) + 20.0 \* Math.sin(2.0 \* x \* pi)) \* 2.0 / 3.0;  
ret += (20.0 \* Math.sin(x \* pi) + 40.0 \* Math.sin(x / 3.0 \* pi)) \* 2.0 / 3.0;  
ret += (150.0 \* Math.sin(x / 12.0 \* pi) + 300.0 \* Math.sin(x / 30.0 \* pi)) \* 2.0 / 3.0;  
return ret;  
}  
}  
  
网上转的,感谢作者提供.