

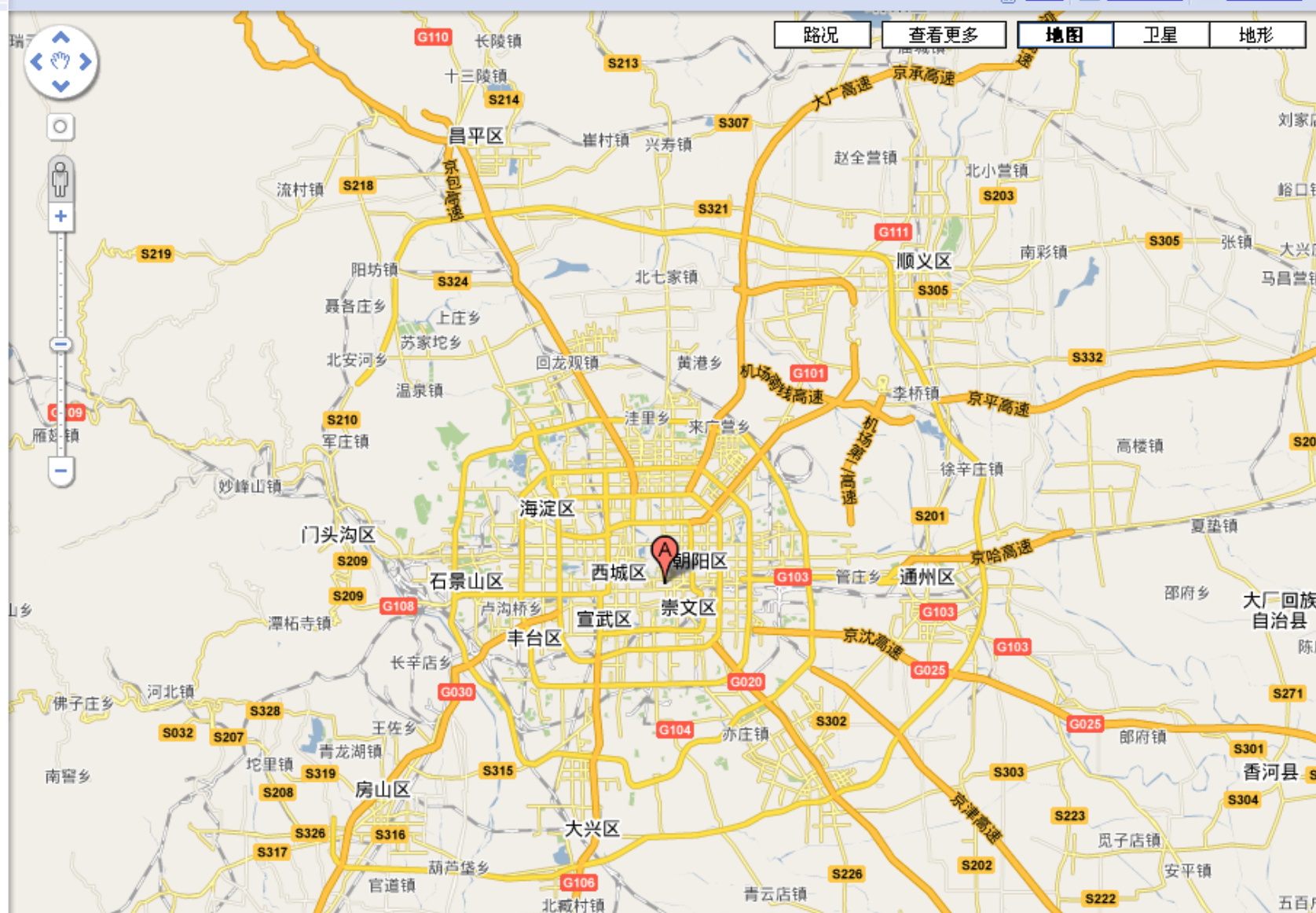


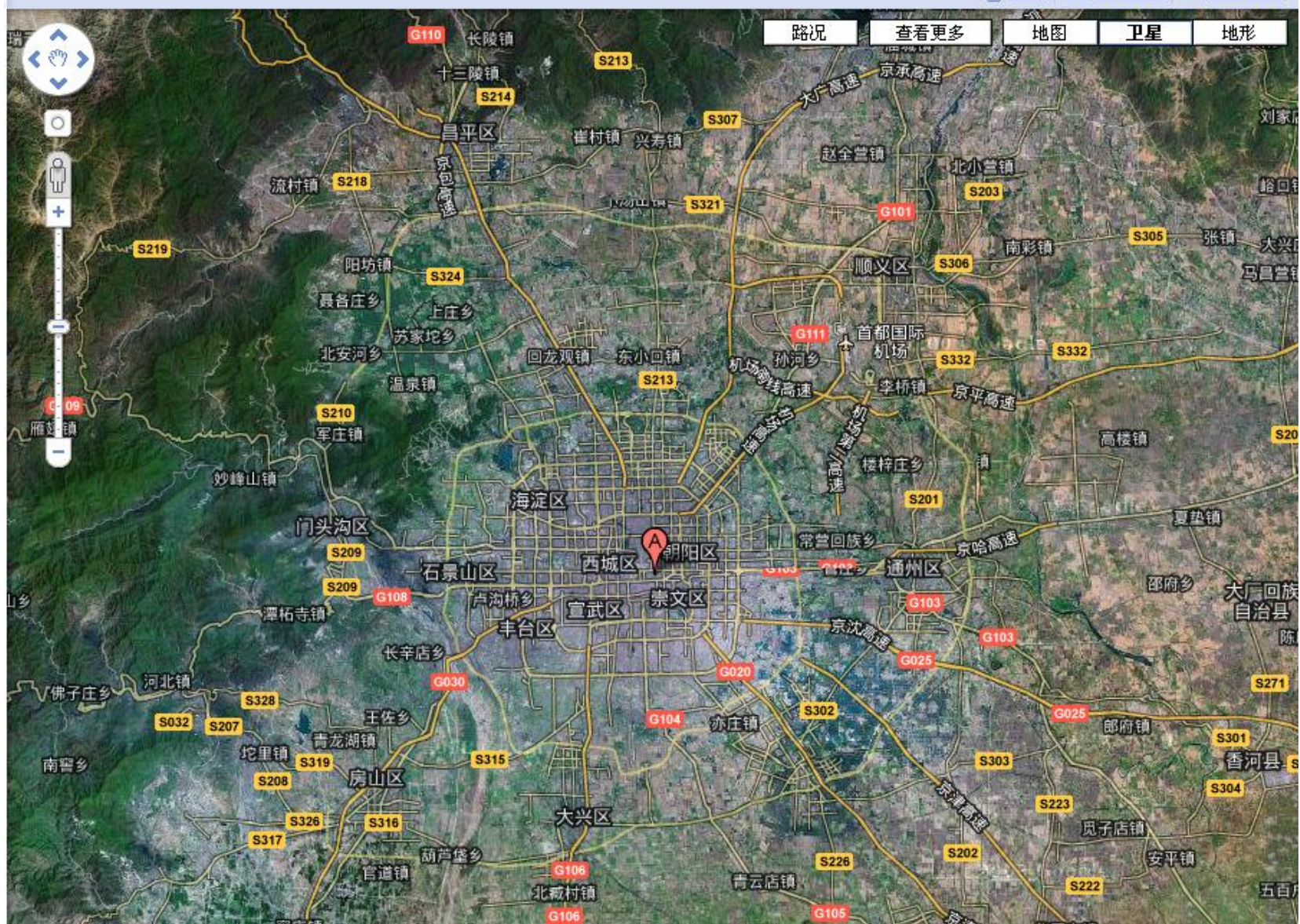
Google Map应用

3G移动开发

Google Map介绍

Google地图（英语：**Google Maps**）是 **Google** 公司提供的电子地图服务，包括局部详细的卫星照片。能提供三种视图：一是矢量地图（传统地图），可提供政区和交通以及商业信息；二是不同分辨率的卫星照片（俯视图，跟 **Google Earth** 上的卫星照片基本一样）；三是后来加上的地形视图，可以用以显示地形和等高线。它的姊妹产品是**Google Earth**，一个桌面应用程序，在三维模型上提供街景和更多的卫星视图及**GPS**定位（付费版本）的功能，但没有前述的矢量视图和地形视图功能。





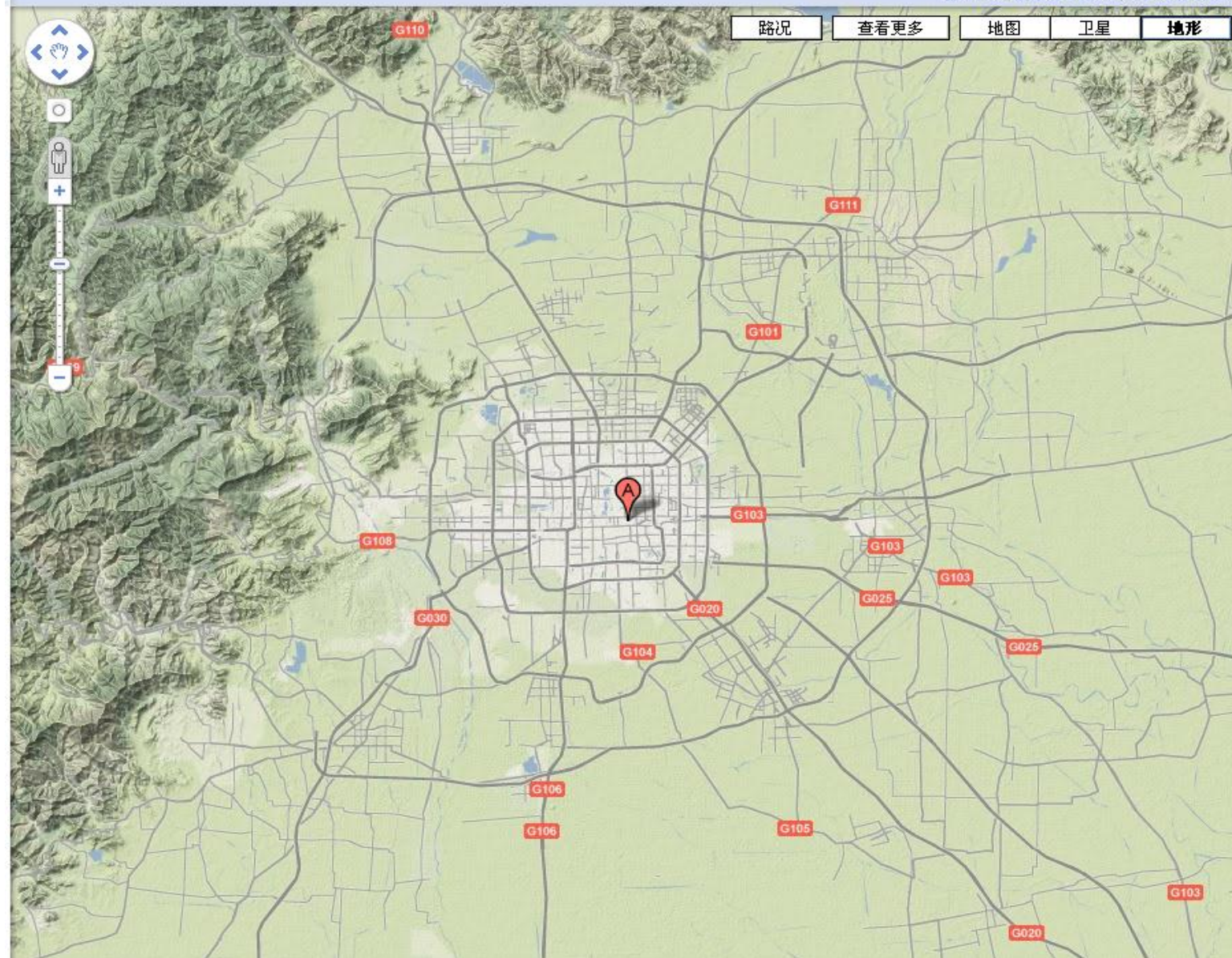
路况

查看更多

地图

卫星

地形



Google Map介绍

Google提供地图，有了地图我们的很多应用软件都更加丰富。我们可以搜索我们附近饭店、影院和交通路线等等。



获取 Google Map API Key

在模拟器中正常使用**Google Map**之前需要提供一个 **APIKey**（密钥），
否则在**MapView**中只能看到网格，没有任何地图信息。步骤：

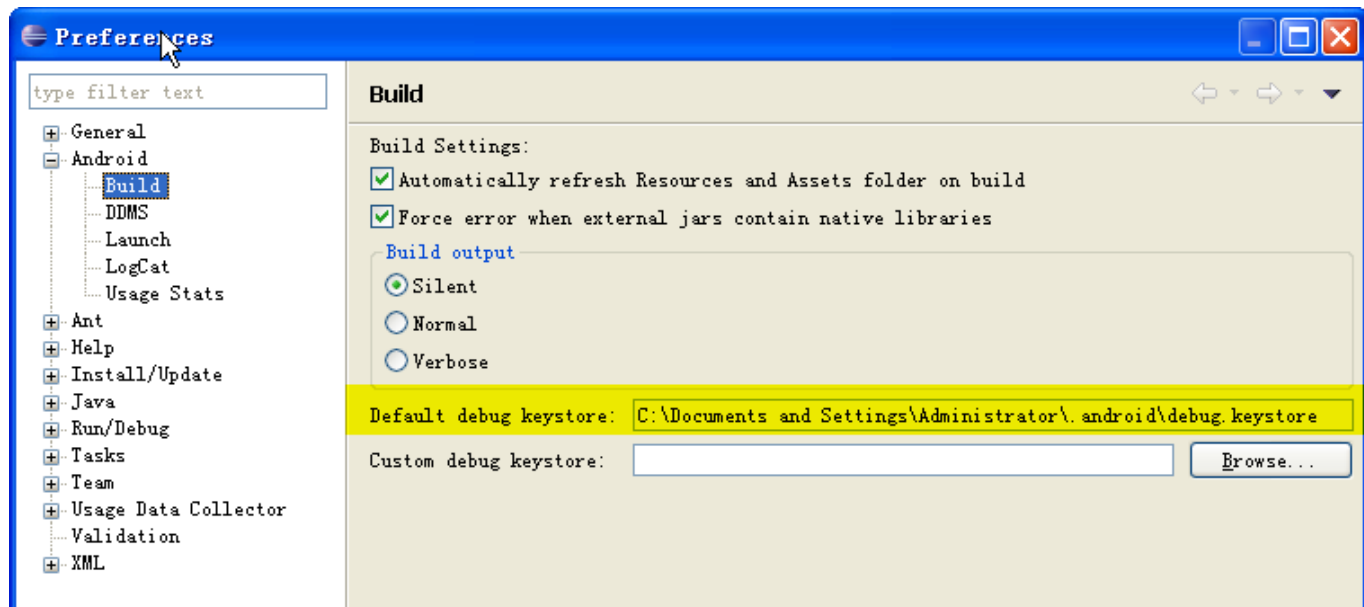
- 找到keystore证书文件

- 生成MD5认证指纹

- 生成Google Map API Key

keystore

keystore是一个密码保护的文件，用来存储密钥和证书。**Eclipse**提供了一个临时**debug.keystore**文件。而实际发布要自己生成。



生成MD5认证指纹

启动**Command Prompt**. 使用**Keytool**命令（**keytool.exe**存放在**Java/bin**目录下）。完整的命令行：

C:\>keytool -list -keystore (回车输入密码： android)



```
C:\WINDOWS\system32\cmd.exe

0 个目录 30,388,895,744 可用字节

C:\Program Files\Java\jdk1.6.0_06\bin>keytool -list -keystore "C:\Documents and
Settings\Administrator\.android\debug.keystore"
输入keystore密码:

Keystore 类型: JKS
Keystore 提供者: SUN

您的 keystore 包含 1 输入

androiddebugkey, 2010-3-14, PrivateKeyEntry,
认证指纹 (MD5): 63:EF:1E:4F:8A:1D:A1:AD:C9:7E:FF:FC:D7:13:55:F3
```

生成Google Map API Key

进入<http://code.google.com/intl/zh-CN/android/maps-api-signup.html>

这是Google Map API Key在线生成网站。

Sign Up for the Android Maps API

The Android Maps API lets you embed [Google Maps](#) in your own Android applications. A single Maps API key for more information about application signing. To get a Maps API key for your certificate, you will need Linux or Mac OSX, you would examine your debug keystore like this:

```
$ keytool -list -keystore ~/.android/debug.keystore
...
Certificate fingerprint (MD5): 94:1E:43:49:87:73:BB:E6:A6:88:D7:20:F1:8E:B5:98
```

If you use different keys for signing development builds and release builds, you will need to obtain a separate certificate.

You also need a [Google Account](#) to get a Maps API key, and your API key will be connected to your G

Android Maps APIs Terms of Service

Last Updated: October 13, 2008

Thanks for your interest in the Android Maps APIs. The Android Maps APIs are a collection of services (including, but not limited to, the "com.google.android.maps.MapView" and "android.location.Geocoder" classes) that allow you to include maps, geocoding, and other content from Google and its content providers in your Android applications. The Android Maps APIs explicitly do not include any driving directions data or local search data that may be owned or licensed by Google.

1. Your relationship with Google.
 - 1.1. Your use of any of the Android Maps APIs (referred to in this



I have read and agree with the terms and conditions ([printable version](#))

My certificate's MD5 fingerprint: 94:1E:43:49:87:73:BB:E6:A6:88:D7:20:F1:8E:B5:98

Generate API Key

输入自己的MD5认证
指纹

确保具有Google账户并且登录！！

Google 地图 API

[Google 代码主页](#) > [Google 地图 API](#) > Google 地图 API 注册

感谢您注册 Android 地图 API 密钥！

您的密钥是：

Oobloo9wXdp0Jx7DfJQcM-PJQNobDLO7gWrAMeg

此密钥适用于所有使用以下指纹所对应证书进行验证的应用程序：

94:1E:43:49:87:73:BB:E6:A6:88:D7:20:F1:8E:B5:98

下面是一个 xml 格式的示例，帮助您了解地图功能：

```
<com.google.android.maps.MapView
    android:layout_width="fill_parent"
    android:layout_height="fill_parent"
    android:apiKey="Oobloo9wXdp0Jx7DfJQcM-PJQNobDLO7gWrAMeg"
/>
```

有关详细信息，请查看 [API 文档](#)。

Google 帐户

电子邮件:

密码:

☒ 保持登录状态

[无法访问您的帐户？](#)

还没有 Google 帐户？
[现在就创建一个帐户](#)

Copy Google Map API Key

```
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
    android:id="@+id/main"
    android:layout_width="fill_parent"
    android:layout_height="fill_parent">
    <com.google.android.maps.MapView
        android:layout_width="fill_parent"
        android:layout_height="fill_parent"
        android:enabled="true"
        android:clickable="true"
        android:apiKey="0obloo9wXdp1XPk8Y8SpXDxDMMZwYmYyOo8OZ0Q"
    />
</LinearLayout>
```

创建Google Map程序

选择Google API

继承MapActivity

在AndroidManifest.xml中增加internet访问权限

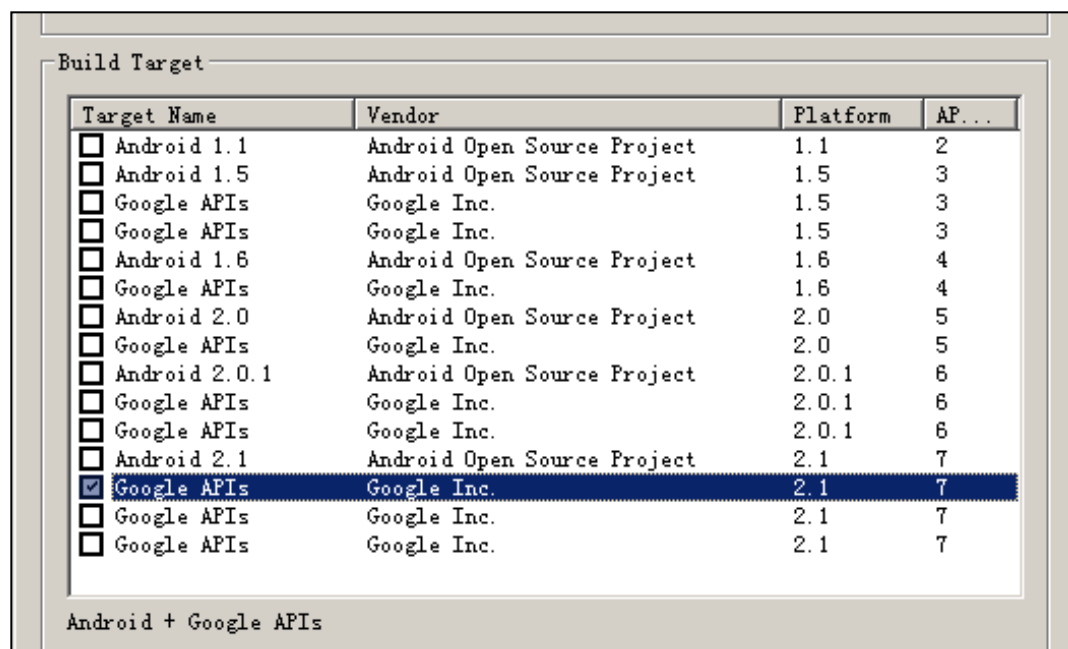
在AndroidManifest.xml中增加Google地图函数库

选择Google API

Google Map API是Google API一部分，它不属于Android，是Android的附件（add-ons）。

我们需要选择，

Google API



继承MapActivity

我们需要继承MapActivity 而不是Activity

```
public class MapDemo1 extends MapActivity {  
  
    @Override  
    public void onCreate(Bundle savedInstanceState) {  
        super.onCreate(savedInstanceState);  
        setContentView(R.layout.main);  
    }  
  
    @Override  
    protected boolean isRouteDisplayed() {  
        return false;  
    }  
}
```

AndroidManifest.xml

在XML文件中我们修改访问Internet权限和Google API 用户函数库声明。

```
<uses-permission android:name=" android.permission.INTERNET" >
</uses-permission>
<application android:icon="@drawable/icon"
    android:label="@string/app_name" >
    <uses-library android:name=" com.google.android.maps" />
    <activity android:name=".MapDemo1"
        android:label="@string/app_name" >
        <intent-filter>
            <action android:name=" android.intent.action.MAIN" />
            <category android:name=" android.intent.category.LAUNCHER" />
        </intent-filter>
    </activity>
</application>
```


internet权限

**<uses-permission
android:name="android.permission.INTERNET">**

为我们的应用程序提供可以访问internet权限。

<uses-permission>标签要放到**<application>**之外。



保持网络连通

Google API 用户函数库声明

<uses-library android:name="com.google.android.maps" />

因为Google地图函数库是选择性的API，所以在使用它之前，必须在**<application>**标签中，需额外添加一条**<uses-library>**（函数库），表示程序会使用到Google地图的函数库。

运行结果



控制地图

地图的控制主要是通过**MapController**实现，可以实现如下控制：

- 设定地图中心点

- 设定地图的缩放尺寸

- 设定地图的显示模式

通过**MapView**对象可以获得**MapController**，如下代码：

```
MapController mc = mapView.getController()
```


设定地图中心点

设定地图中心点**GeoPoint**类结合**MapController**的**animateTo**方法。

```
private MapView mapView;  
private MapController mc;  
  
@Override  
public void onCreate(Bundle savedInstanceState) {  
    super.onCreate(savedInstanceState);  
    setContentView(R.layout.main);  
  
    mapView = (MapView) findViewById(R.id.map);  
    mc = mapView.getController();  
  
    GeoPoint gp = new GeoPoint((int) (39.904667 * 1000000),  
        (int) (116.408198 * 1000000)); // 地理坐标  
    mc.animateTo(gp);  
}
```

以北京为中心的运行结果



如何知道北京的坐标？

我们可以通过<http://www.mygeoposition.com/>

The screenshot shows the MyGeoPosition.com website interface. At the top, there are flags for various countries and a language selection dropdown. Below this is a navigation bar with links: "Support us & translate:", "Geocoding", "Geotags", "Geo-Metatags", "KML (Google Earth™)", and the site logo "MyGeoPosition.com". A search bar contains the text "柏林勃兰登堡 / 39.9,116.3 / filzhut.de" and a "确切" (Exact) button. To the right of the search bar is a button labeled "计算地理数据" (Calculate geographic data). Below the search bar is a horizontal menu with tabs: "信息" (Info), "地图" (Map), "地理文件" (Geographic files), "地理元数据标记" (Geographic metadata tags), "地理标记" (Geographic tags), "KML", "地图上链接" (Links on the map), "语言" (Language), and "声明" (Disclaimer). The main content area displays a Google Map of Beijing. A red location pin is placed in the central part of the city. A tooltip box is open over the pin, displaying the following coordinates: "地平纬度: 39.904667 (39° 54' 16.80\" N)" and "地平经度: 116.408198 (116° 24' 29.51\" O)". Below the coordinates, the tooltip text says "移动标记或者点击地图来改变位置!" (Move the marker or click the map to change the position!). The map shows various landmarks and streets in Beijing, including the Forbidden City, Tiananmen Square, and the Great Wall. At the bottom of the map, there is a scale bar indicating "2 英里" (2 miles) and "2 公里" (2 kilometers). Below the map is a row of social media sharing icons, including "Add to iGoogle", "Share" (Twitter), and "Share" (Facebook). At the very bottom, the site logo "MyGeoPosition.com" is displayed.

小结

北京的地平纬度:**39.904667**地平经度:**116.408198**

在构造**GeoPoint**对象时候需要将经纬度乘 **1000000**

```
new GeoPoint((int) (39.904667 * 1000000), (int) (116.408198 *  
1000000));
```

也可以使用科学计数法**1E6**

```
new GeoPoint((int) (39.904667 * 1E6), (int) (116.408198 * 1E6));  
animateTo指定坐标的中心点。
```


设定地图的缩放尺寸

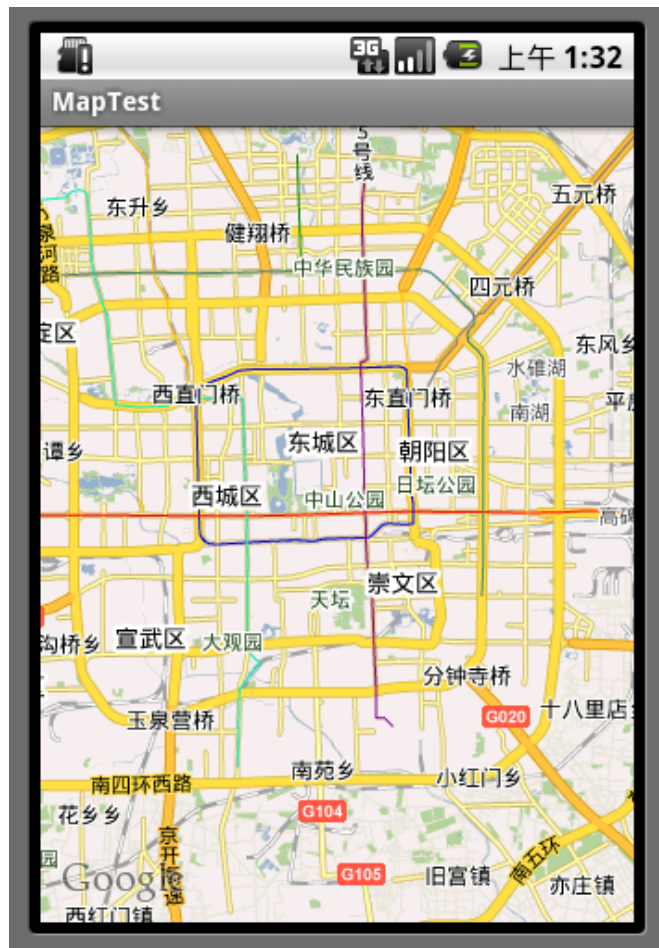
在上面的例子中我们能看见整个亚洲，但是看不到北京，我们要通过 **setZoom** 设定地图的缩放尺寸，它的取值访问时1-21，1是全球地图，21是街道地图。

```
@Override
public void onCreate(Bundle savedInstanceState) {
    super.onCreate(savedInstanceState);
    setContentView(R.layout.main);
    mapView = (MapView) findViewById(R.id.map);
    mc = mapView.getController();

    GeoPoint gp = new GeoPoint((int) (39.904667 * 1E6),
    (int) (116.408198 * 1000000)); // 地理坐标

    mc.animateTo(gp);
    mc.setZoom(12);
}
```

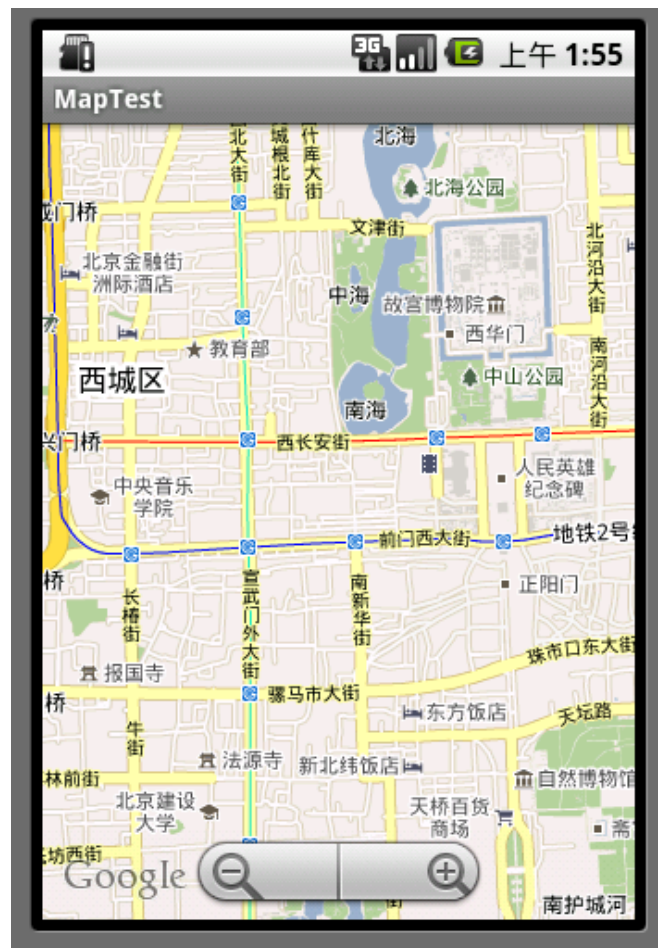
设置setZoom(12)运行结果



增加缩放按钮

在上面的案例中无法控制地图的缩放，我们需要添加缩放按钮。SDK1.5后MapView增加了**setBuiltInZoomControls**实现是否显示这个按钮组。

```
mapView.setBuiltInZoomControls(true);
```



设定地图的显示模式

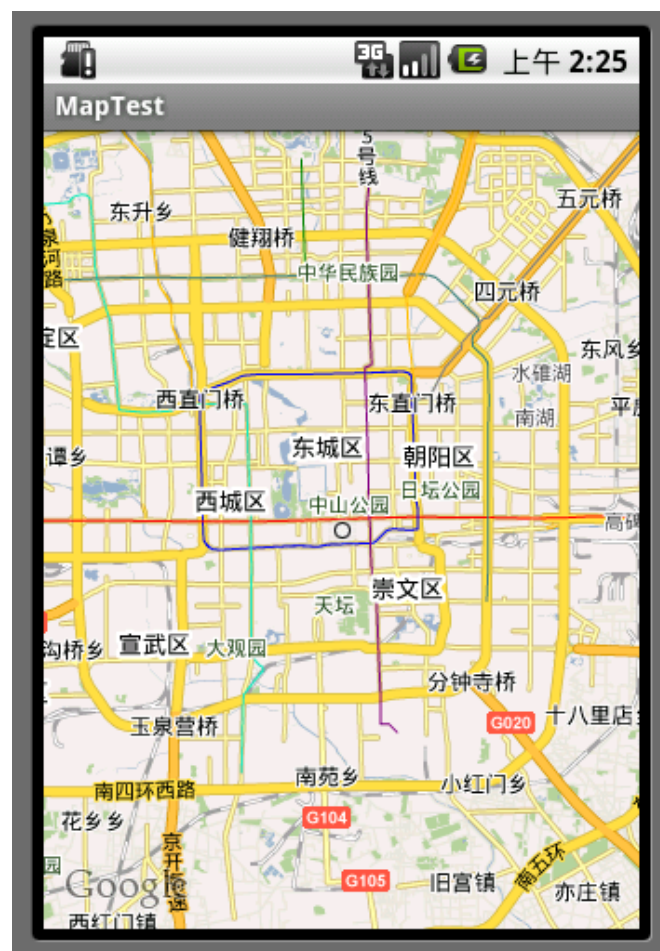
Google Map有三种显示模式：普通模式、卫星模式、和街道模式。
分别通过**MapView**的如下方式实现：

普通模式`setTraffic(true)`

卫星模式`setSatellite(true)`

街道模式`setStreetView(true)`

卫星模式和普通模式



卫星-普通模式切换实例



增加菜单

```
public class MapModelSample extends MapActivity {  
  
    // 菜单项  
    final private int menuModel1 = Menu.FIRST;  
    final private int menuMode2 = Menu.FIRST + 1;  
  
    private MapView mapView;  
    private MapController mc;  
  
    @Override  
  
    @Override  
    public boolean onCreateOptionsMenu(Menu menu) {  
        // 建立菜单  
        menu.add(0, menuModel1, 0, "地图模式");  
        menu.add(0, menuMode2, 1, "卫星模式");  
        return super.onCreateOptionsMenu(menu);  
    }  
}
```

增加菜单项事件

```
@Override
public boolean onOptionsItemSelected(MenuItem item) {
    switch (item.getItemId()) {
        case MENU_MODE1:
            mapView.setTraffic(true);
            mapView.setSatellite(false);
            mapView.setStreetView(false);
            break;
        case MENU_MODE2:
            mapView.setSatellite(true);
            mapView.setTraffic(false);
            mapView.setStreetView(false);
    }
    return super.onOptionsItemSelected(item);
}
```

Google Map 图层

有的时候我们需要在地图上添加一些标志来提供一些信息，如在一个旅游区附件标志出来旅游点的位置。这样我们需要使用图层（**Layers**）在**Android Google Map** 图层类是**Overlay**。我们需要根据自己的需要继承**Overlay**并重写其方法。

图层实例

在北京市标志处四个旅游景点。

中关村

中南海

天坛

玉渊潭公园



ItemizedOverlay

ItemizedOverlay是Google Map提供的**Overlay**子类。

它可以提供一组坐标点的图层。该类又是一个集合内部容纳的是**OverlayItem**实例。

OverlayItem是代表着各个图层的坐标点。

我们要编写继承**ItemizedOverlay<OverlayItem>**子类。

自定义ItemizedOverlay

```
private class SitesOverlay extends ItemizedOverlay<OverlayItem> {
    private List<OverlayItem> items =
new ArrayList<OverlayItem>();
    private Drawable marker = null;

    public SitesOverlay(Drawable marker) {
        super(marker);
        this.marker = marker;

        items.add(new OverlayItem(new GeoPoint((int)
(39.882079 * 1E6), (int) (116.411133 * 1E6)),
            "天坛", "这里是天坛皇上祭祀的地方。"));
        ... ..

        populate();
    }
    ... ..
```

```
@Override
protected OverlayItem createItem(int i) {
    return (items.get(i));
}
```

```
@Override
public void draw(Canvas canvas, MapView mapView, boolean shadow) {
    super.draw(canvas, mapView, shadow);
    boundCenterBottom(marker);
}
```

```
@Override
protected boolean onTap(int i) {
    Toast.makeText(MapDemoLayer.this,
        items.get(i).getSnippet(),
        Toast.LENGTH_LONG).show();
    return (true);
}
```

```
@Override
public int size() {
    return (items.size());
}
```

```
}
```

小结

onTap方法是在屏幕上触摸某个标志是触发的事件。

size方法返回图层中地标的个数。

createItem方法返回图层地标对象。

MapActivity的onCreate方法

```
public void onCreate(Bundle savedInstanceState) {  
    super.onCreate(savedInstanceState);  
    setContentView(R.layout.main);  
  
    map = (MapView) findViewById(R.id.map);  
  
    map.getController().setZoom(12);  
    map.setBuiltInZoomControls(true);  
    GeoPoint gp = new GeoPoint((int) (39.904667 * 1E6),  
                                (int) (116.408198 * 1E6)); // 地理坐标  
    map.getController().animateTo(gp);  
    Drawable marker = getResources().getDrawable(R.drawable.marker);  
    map.getOverlays().add(new SitesOverlay(marker));  
  
}
```

运行结果

