Link: <http://code.google.com/android/add-ons/google-apis/reference/index.html>

Book: Pro Android (Page: 240)

MapView & MapActivity

Nhiều công nghệ bản đồ trong Android dựa trên control MapView và phần mở rộng android.app.Activity được gọi là MapActivity. Lớp MapView và MapActivity là thành phần cốt lõi của bản đồ trên Android nó cho phép hiển thị và thao tác một bản đồ trên Android. Một trong những điều phải nhớ về 2 lớp này là chúng làm việc cùng nhau. Đặc biệt để sử dụng MapView, bạn cần khai báo bên trong một MapActivity. Thêm vào đó để sử dụng được MapView bạn cần cung cấp api-key. Ngoài ra ứng dụng của bạn cần truy cập được internet.



Hình trên cho thấy một ứng dụng hiển thị một bản đồ ở chế độ street-view. Ứng dụng cũng cho thấy rõ phóng to, thu nhỏ và thay đổi chế độ nhìn của bản đồ như thế nào. Layout của ứng dụng được thiết kế như bên dưới:

<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"

android:orientation="vertical" android:layout\_width="fill\_parent"

android:layout\_height="fill\_parent">

<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"

android:orientation="horizontal" android:layout\_width="fill\_parent"

android:layout\_height="wrap\_content">

<Button android:id="@+id/zoomin" android:layout\_width="wrap\_content"

android:layout\_height="wrap\_content" android:text="+"/>

<Button android:id="@+id/zoomout" android:layout\_width="wrap\_content"

android:layout\_height="wrap\_content" android:text="-"/>

<Button android:id="@+id/sat" android:layout\_width="wrap\_content"

android:layout\_height="wrap\_content" android:text="Satellite"/>

<Button android:id="@+id/street" android:layout\_width="wrap\_content"

android:layout\_height="wrap\_content" android:text="Street"/>

<Button android:id="@+id/traffic" android:layout\_width="wrap\_content"

android:layout\_height="wrap\_content" android:text="Traffic"/>

</LinearLayout>

<com.google.android.maps.MapView android:id="@+id/mapview"

android:layout\_width="fill\_parent"

android:layout\_height="wrap\_content"

android:apiKey="07vhL0usFXryRakmo2A4t8aKViWwKyGJGEDqpdg"

/>

</LinearLayout>

Dưới đây là một đoạn code mẫu cho ứng dụng bản đồ trên (Listing 7-13)

import android.os.Bundle;

import android.view.View;

import android.view.View.OnClickListener;

import android.widget.Button;

import com.google.android.maps.MapActivity;

import com.google.android.maps.MapView;

public class MapViewDemoActivity extends MapActivity

{

private MapView mapView;

@Override

protected void onCreate(Bundle savedInstanceState) {

super.onCreate(savedInstanceState);

setContentView(R.layout.mapview);

mapView = (MapView)findViewById(R.id.mapview);

Button zoominBtn = (Button)findViewById(R.id.zoomin);

Button zoomoutBtn = (Button)findViewById(R.id.zoomout);

Button satBtn = (Button)findViewById(R.id.sat);

Button streetBtn = (Button)findViewById(R.id.street);

Button trafficBtn = (Button)findViewById(R.id.traffic);

// zoomin

zoominBtn.setOnClickListener(new OnClickListener(){

@Override

public void onClick(View view)

{

mapView.getController().zoomIn();

}});

// zoom out

zoomoutBtn.setOnClickListener(new OnClickListener(){

@Override

public void onClick(View view)

{

mapView.getController().zoomOut();

}});

// satellite

satBtn.setOnClickListener(new OnClickListener(){

@Override

public void onClick(View view)

{

mapView.setStreetView(false);

mapView.setTraffic(false);

mapView.setSatellite(true);

}});

// street

streetBtn.setOnClickListener(new OnClickListener(){

@Override

public void onClick(View view)

{

mapView.setTraffic(false);

mapView.setSatellite(false);

mapView.setStreetView(true);

}});

// traffic

trafficBtn.setOnClickListener(new OnClickListener(){

@Override

public void onClick(View view)

{

mapView.setSatellite(false);

mapView.setStreetView(false);

mapView.setTraffic(true);

}});

}

@Override

protected boolean isRouteDisplayed() {

return false;

}

}

Như hiển thị trong Listing 7-13, hiển thị MapView cũng sử dụng onCreate() không khác so với hiển thị các control khác. Để phóng to, thu nhỏ bạn sử dụng lớp MapController của MapView. Để làm được điều này bằng cách mapView.getController và sau đó gọi phương thức thích hợp zoomIn() hay zoomOut().

MapView hỗ trợ một số chế độ hiển thị: map, street, satellite và traffic. Map là chế độc mặc định. Chết độ hiển thị Street chứa thông tin đường đi như tên đường. Chế độ Satelite hiển thị bản đồ ở chế độ vệ tinh. Chế độ Traffic hiển thị thông tin giao thông trên bản đồ. Lưu ý rằng chế độ Traffic chỉ được hỗ trợ trên một số tuyến đường cao tốc chính. Để thay đổi chế độ hiển thị, bạn phải gọi các phương thức thích hợp được thiết lập là true và các chế độ khác là false.

**Lớp MapView**

Một View hiển thị một bản đồ (dữ liệu được lấy về từ dịch vụ Google Maps). Khi được focus, nó sẽ bắt lấy các hành động nhấn phím và chạm vào màn hình để dịch chuyển và thay đổi kích thướt bản đồ. Nó có thể được điều khiển một cách theo cách lập trình (getController()) và có thể vẽ một các Overlay trên bản đồ (getOverlays()).

MapView cũng là một ViewGroup với LayoutParameters cho phép gán các chế độ hiển thị, tại các offset pixel xác định hay các tọa độ xác định.

Một MapView có thể được khởi tạo bằng một MapActivity do nó phụ thuộc vào các tiến trình truy cập vào mạng và các file hệ thống bên dưới; những tiến trình này phải tuân theo lifecycle trong MapActivity.

Dưới đây là thông tin chi tiết về lớp MapView:

Constructor

|  |
| --- |
| [**MapView**](http://code.google.com/android/add-ons/google-apis/reference/com/google/android/maps/MapView.html#MapView(android.content.Context, android.util.AttributeSet))(android.content.Context context,android.util.AttributeSet attrs)           Khởi tạo một đối tượng MapView |
| [**MapView**](http://code.google.com/android/add-ons/google-apis/reference/com/google/android/maps/MapView.html#MapView(android.content.Context, android.util.AttributeSet, int))(android.content.Context context, android.util.AttributeSet attrs, int defStyle) |
| [**MapView**](http://code.google.com/android/add-ons/google-apis/reference/com/google/android/maps/MapView.html#MapView(android.content.Context, java.lang.String))(android.content.Context context, java.lang.String apiKey) |

Các phương thức:

|  |  |
| --- | --- |
| Phương thức | Mô tả |
| protected void **onSizeChanged**(int w, int h,  int oldw,int oldh) | Điều chỉnh lại kích thướt của đối tượng map và chuyển lại vị trí các đường kẻ |
| public void **computeScroll**() | Bắt lấy các sự kiện cuộn màn hình và điều chỉnh lại camera bản đồ |
| protected final void **onDraw**(android.graphics.Canvas canvas) | Đây là phương thức vẽ chính. Vẽ nền của bản đồ, các vật phủ trên bản đồ (overlay), các đường lưới và logo của Google |
| public boolean **onKeyDown**(int keyCode,  android.view.KeyEvent event) | Truyến các phím nhấn đến các đối tượng overlay trước. Nếu các overlay này không xử lý thì sẽ được chuyển qua sự kiện dịch chuyển bản đồ |
| public boolean **onKeyUp**(int keyCode,  android.view.KeyEvent event) | Giống như onKetDown |
| public void **displayZoomControls**(boolean takeFocus) | Hiển thị các control phóng to/thu nhỏ |
| public int **getZoomLevel**() | Trả về mức phóng to/thu nhỏ của bản đồ |
| public void **setSatellite**(boolean on) | Thiết lập bản đồ ờ chế độ vệ tinh |
| public boolean **isSatellite**() | Kiểm tra xem bản đồ có ở chế độ hiển thị vệ tinh hay không |
| public void **setStreetView**(boolean on) |  |
| public boolean **isStreetView**() |  |
| public [GeoPoint](http://code.google.com/android/add-ons/google-apis/reference/com/google/android/maps/GeoPoint.html) **getMapCenter**() | Trả về vị trí trung tâm hiện tại của bản đồ |
| public [MapController](http://code.google.com/android/add-ons/google-apis/reference/com/google/android/maps/MapController.html) **getController**() | Trả về đối tượng MapController cho bản đồ, có thể sử dụng đối tượng này để điều khiển bản đồ |
| public final java.util.List<[Overlay](http://code.google.com/android/add-ons/google-apis/reference/com/google/android/maps/Overlay.html)> **getOverlays**() | Truy cập danh sách các overlay |
| public int **getLatitudeSpan**() | Lấy vĩ độ hiện tại của span (từ cạnh trên cùng đến cạnh cuối cùng của bản đồ) ở độ thập phân được nhân lên một triệu lần |
| public int **getLongitudeSpan**() | Lấy kinh độ hiện tại của span (từ cạnh trên cùng đến cạnh cuối cùng của bản đồ) ở độ thập phân được nhân lên một triệu lần |
| public void **setReticleDrawMode**  ([MapView.ReticleDrawMode](http://code.google.com/android/add-ons/google-apis/reference/com/google/android/maps/MapView.ReticleDrawMode.html" \o "enum in com.google.android.maps) mode) | Specify how the reticle should be drawn when the device is not in touch mode.  Parameter  mode - The reticle draw mode. Valid values are: [MapView.ReticleDrawMode.DRAW\_RETICLE\_OVER](http://code.google.com/android/add-ons/google-apis/reference/com/google/android/maps/MapView.ReticleDrawMode.html" \l "DRAW_RETICLE_OVER) - this is the default mode and in this mode the reticle will be drawn over all the overlays so it's always visible. [MapView.ReticleDrawMode.DRAW\_RETICLE\_UNDER](http://code.google.com/android/add-ons/google-apis/reference/com/google/android/maps/MapView.ReticleDrawMode.html" \l "DRAW_RETICLE_UNDER) - the reticle will be drawn before all the overlays are drawn meaning the reticle is there but if any object is drawn at the same position it will cover the reticle. [MapView.ReticleDrawMode.DRAW\_RETICLE\_NEVER](http://code.google.com/android/add-ons/google-apis/reference/com/google/android/maps/MapView.ReticleDrawMode.html" \l "DRAW_RETICLE_NEVER)- the reticle is not drawn |
| public int **getMaxZoomLevel**() | Returns the maximum zoom level for the point currently at the center. (Not all areas of the map offer the same high-res, close-up map tiles.) Takes into account whether satellite mode is enabled. |
| public void **onSaveInstanceState**(android.os.Bundle state) | Saves the state of the MapView to a Bundle. Currently, this persists the center, zoomlevel, and state of the Zoom dialog.  Paramater  state - The Bundle to maintain the current state. |
| public void **onRestoreInstanceState**(android.os.Bundle state) | Restores the state of the MapView to a Bundle. Currently, this restores the center, zoomlevel, and state of the Zoom dialog.  Parameter  state - The Bundle with the prior instance state. |
| public android.view.View **getZoomControls**() | **Deprecated.** *The built-in zoom mechanism is preferred, see [setBuiltInZoomControls(boolean)](http://code.google.com/android/add-ons/google-apis/reference/com/google/android/maps/MapView.html" \l "setBuiltInZoomControls(boolean)).*  Returns a View containing a ZoomControls widget (i.e. +/- buttons). This view will become visible when the user starts panning the Map via touch and fade away if the user does not interact with it. The MapView handles this show and hide as appropriate, but the Activity is responsible for placing the controls on the screen. Using ViewGroup.addView(View) is the common way to add this View to the screen.  API version 3 introduces a built-in zoom mechanism that is shown automatically by the MapView. This is the preferred approach for showing the zoom UI. |
| public void **setBuiltInZoomControls**(boolean on) | Sets whether to enable the built-in zoom controls. If enabled, the MapView will show these zoom controls automatically.  **Parameters:**  on - Whether the built-in zoom controls are enabled. If false, the client handles showing the UI for zoom controls. |
| public [Projection](http://code.google.com/android/add-ons/google-apis/reference/com/google/android/maps/Projection.html) **getProjection**() | Get a projection for converting between screen-pixel coordinates and latitude/longitude coordinates.  **Returns:**  The Projection of the map in its current state. You should not hold on to this object for more than one draw, since the projection of the map could change. |