



**FPT POLYTECHNIC**

THIẾT KẾ GIAO DIỆN TRÊN  
ANDROID

## **Bài 7: Menu- WebKit**

[www.poly.edu.vn](http://www.poly.edu.vn)

## Nội dung bài học

1. Menu
2. Context menu
3. WebKit Browser

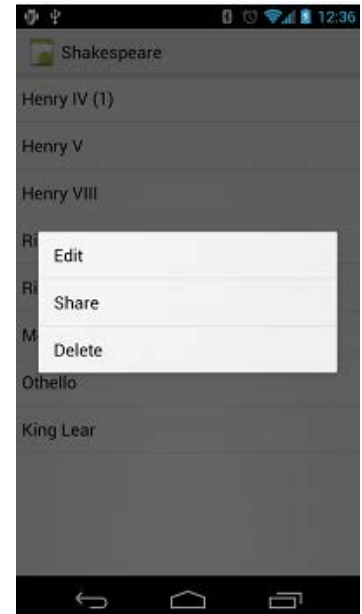


# 1. Menu

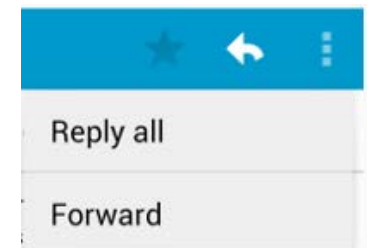
- Menu thường có trong hầu hết ứng dụng.
- Trong android có 3 loại menu phổ biến:
  - Option menu (Hình 1): thường được kích hoạt bằng cách nhấn vào nút Menu
  - Context menu (Hình 2): hiện lên khi chạm và giữ vào một ứng dụng widget.
  - Popup menu (Hình 3)



**Hình 1.** Options menu in the Browser, on Android 2.3.



**Hình 2.** Screenshots of a floating context menu



**Hình 3:** A popup menu in the Gmail app,

# 1. Menu

- Cả option menu và context menu đều có thể chứa:
  - Text
  - Icons
  - Radio Buttons
  - Check Boxes
  - Sub- menu
  - Short-cut keys

# 1. Menu

- Tạo Menu bằng XML resource, tải và viết sự kiện cho các item:
  - Tạo một tệp mymenu.xml trong thư mục **/res/menu/**

```
<?xml version="1.0" encoding="utf-8"?>
<menu
    xmlns:android="http://schemas.android.com/apk/res/android" >
    <item
        android:id="@+id/item_xemdssv"
        android:title="Xem danh sách sinh viên"
        android:showAsAction=["ifRoom" | "never"
| "withText" | "always" ]>
    </item>
    <item
        android:id="@+id/item_xemdslop"
        android:title="Xem danh sách lớp học">
```

```
<menu >
    <item
        android:id="@+id/item_lop1"
        android:title="Lớp 1">
    </item>
    <item
        android:id="@+id/item_lop2"
        android:title="Lớp 2">
    </item>
    <item
        android:id="@+id/item_lop3"
        android:title="Lớp 3">
    </item>
</menu>
</item>
</menu>
```

# 1. Menu

- Tiếp theo mở **MainActivity.java** lên:

Sử dụng hàm **getMenuInflater().inflate(R.menu.mymenu, menu);** dùng để gắn Menu XML Resource vào ứng dụng.

- **Viết Coding để xử lý sự kiện cho các Menu Item:**

Để viết sự kiện cho các Menu Item cần Override phương thức **onOptionsItemSelected**, Trong phương thức này ta dựa vào **Id** của các **Menu Item** để xử lý.

Ở đây ta dùng **switch case** để xử lý theo đúng **Id** mà người sử dụng chọn lựa. Tùy vào yêu cầu của bài toán mà chúng ta xử lý trong này. Nên viết từng hàm riêng theo nghiệp vụ rồi cứ thế mà gọi hàm theo đúng Menu Item.

# 1. Menu

## MainActivity.java

```
public class MainActivity extends Activity {  
    @Override  
    protected void onCreate(Bundle savedInstanceState)  
    {  
        super.onCreate(savedInstanceState);  
        setContentView(R.layout.activity_main);  
    }  
    @Override  
    public boolean onCreateOptionsMenu(Menu menu)  
    {  
        getMenuInflater()  
        .inflate(R.menu.mymenu, menu);  
        return true;  
    }  
}
```

```
public boolean onOptionsItemSelected(MenuItem  
item) {  
    switch(item.getItemId()) {  
        case R.id.item_xemdssv:  
            //Xử lý xem danh sách sinh viên  
            break;  
        case R.id.item_lopdhth1a:  
            //xử lý xem thông tin lớp DHTH1A  
            break;  
        case R.id.item_lopdhth1b:  
            break;  
        case R.id.item_lopdhth1c:  
            break; }  
    return super.onOptionsItemSelected(item);  
}
```

# 1. Menu

- Tạo Menu bằng Coding (Runtime):

- Chúng ta có thể tạo Menu lúc Runtime (không cần dùng XML Resource):

```
public boolean onCreateOptionsMenu(Menu menu) {  
    menu.add("Menu 1");  
    menu.add("Menu 2");  
    SubMenu sub3= menu.addSubMenu("Menu 3");  
    sub3.add("File 1 Menu 3");  
    sub3.add("File 2 Menu 3");  
    sub3.add("File 3 Menu 3");  
    return true;  
}
```

- Mỗi lần tạo Menu Item bạn nên gán Id cho nó và xử lý theo Id này ở trong hàm **onOptionsItemSelected**.



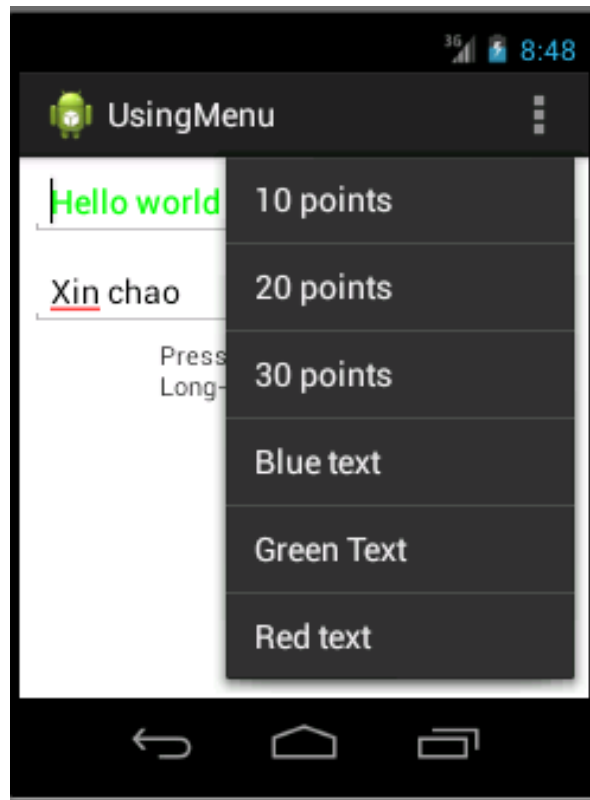
# 1. Menu

```
public class MainActivity extends Activity {
    @Override
    protected void onCreate(Bundle
savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);
    }
    @Override
    public boolean onCreateOptionsMenu(Menu
menu) {
        int itemId=113;
        menu.add(0, itemId, 0, "Menu 1");
        itemId=114;
        menu.add(0,itemId,1,"Menu 2");
        SubMenu sub3=
menu.addSubMenu("Menu 3");
        itemId=115;
        sub3.add(0,itemId,0,"File 1 Menu 3");
        itemId=116;
        sub3.add(0,itemId,1,"File 2 Menu 3");
        itemId=117;
        sub3.add(0,itemId,2,"File 3 Menu 3");
        return true;
    }
}
```

```
    @Override
    public boolean onOptionsItemSelected(MenuItem
item) {
        switch(item.getItemId()) {
            case 113:
                //Xử lý Menu 1
                break;
            case 114:
                //Xử lý Menu 2
                break;
            case 115:
                //Xử lý File 1 của Menu 3
                break;
            case 116:
                //Xử lý File 2 của Menu 3
                break;
            case 117:
                //Xử lý File 3 của Menu 3
                break; }
        return super.onOptionsItemSelected(item);
    }
} //end class
```

# 1. Menu

- Demo → Menu



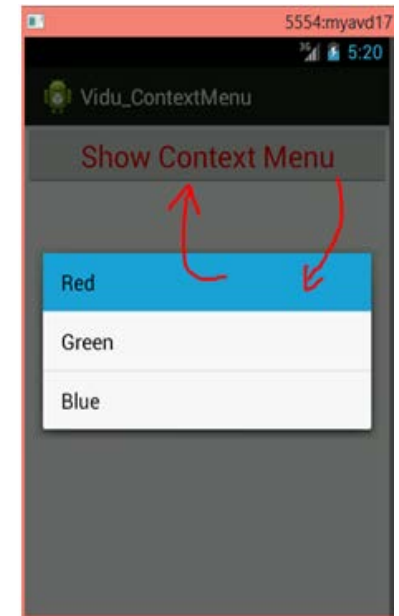
## 2. Context Menu

- Với C# , VB, hay Java để hiển thị Context Menu : Sau khi thiết lập Context Menu ta chỉ cần bấm chuột phải vào đối tượng thì sẽ có Menu Popup ra theo đúng yêu cầu.
- Trong Android: Ta phải đăng ký Context Menu cho đối tượng sau đó muốn hiển thị lên thì ta nhấn thật lâu vào đối tượng (long click ).
- Tương tự như với Menu, Ta có thể tạo Context menu trong XML hoặc trong Coding.
- Cách tạo Context Menu bằng XML tương tự như Menu.
- Ta sẽ đề cập chủ yếu đến cách đăng ký Context Menu cho đối tượng cũng như cách xử lý sự kiện cho từng Menu Item trong Context Menu.

## 2. Context Menu

### Ví dụ

- Ví dụ đơn giản về Context Menu. Giao diện có 1 Button, khi nhấn thật lâu vào nó thì sẽ hiển thị Context Menu cho phép đổi màu chữ : Màu đỏ, màu xanh lá cây và xanh dương:
- Ở hình bên là khi Context Menu hiển thị ra, nếu chọn Red → Màu của Button sẽ chuyển thành màu đỏ.

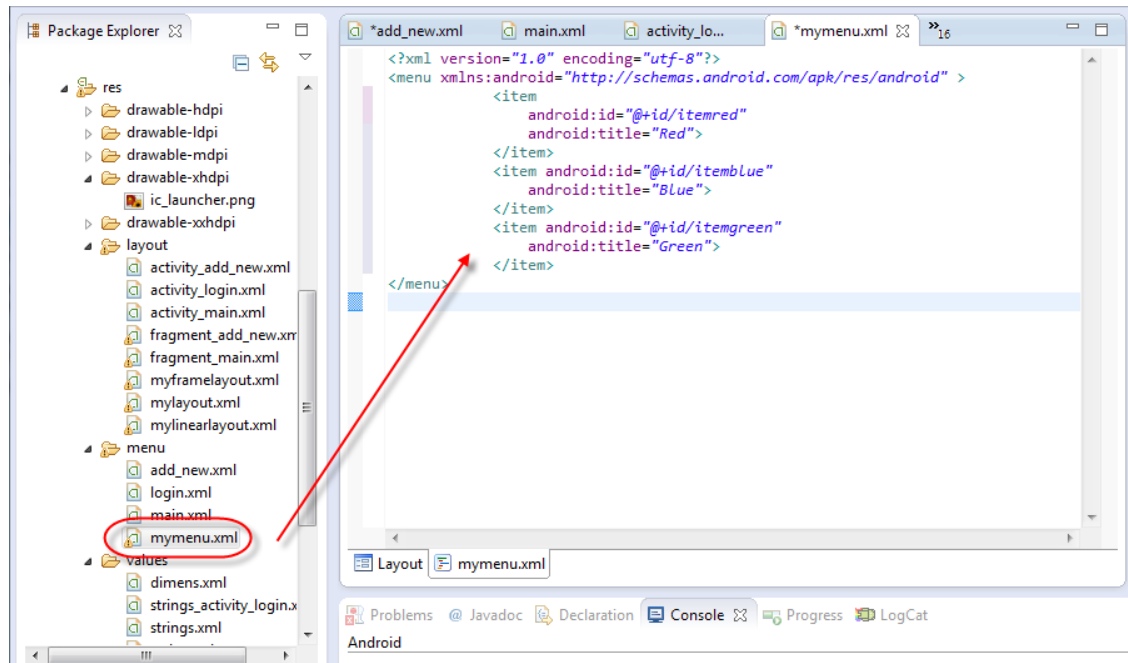


## 2. Context Menu

### Ví dụ

- **BƯỚC 1.** Tạo tệp my\_context\_menu.xml trong thư mục /res/menu

```
<?xml version="1.0" encoding="utf-8"?>
<menu xmlns:android="http://schemas.android.com/apk/res/android" >
    <item
        android:id="@+id/itemred"
        android:title="Red">
    </item>
    <item android:id="@+id/itemblue"
        android:title="Blue">
    </item>
    <item android:id="@+id/itemgreen"
        android:title="Green">
    </item>
</menu>
```



## 2. Context Menu

### Ví dụ

- **BƯỚC 2.** Mở tệp **strings.xml** trong thư mục `/res/value` , và định nghĩa một số color trong này.
- Ở trong `strings.xml` , tạo 3 tag color : Red, Green, Blue; Nội dung phải để dạng Hex Color. 3 màu này sẽ được gọi trong hàm xử lý sự kiện khi người sử dụng chọn từng Menu Item trong Context Menu.

```
<color name= "clrred">#FF0000</color>  
<color name= "clrblue">#000080</color>  
<color name= "clrgreen">#008000</color>
```

## 2. Context Menu

### BƯỚC 3. MỞ tệp MainActivity.java

```
public class MainActivity extends Activity {  
    Button btnCtx;  
    @Override  
    protected void onCreate(Bundle  
savedInstanceState) {  
        super.onCreate(savedInstanceState);  
        setContentView(R.layout.activity_main);  
        btnCtx=(Button)  
findViewById(R.id.btnshowcontext);  
        registerForContextMenu(btnCtx);  
    }  
    @Override  
    public void onCreateContextMenu(ContextMenu  
        menu, View v,  
        ContextMenuInfo menuInfo) {  
        super.onCreateContextMenu(menu, v,  
menuInfo);  
        getMenuInflater()  
        .inflate(R.menu.my_context_menu, menu);  
    }  
}
```

```
@Override  
public boolean onContextItemSelected(MenuItem  
        item) {  
    switch(item.getItemId()) {  
        case R.id.itemRed:  
            btnCtx.setTextColor(  
                getResources().getColor(R.color.clrred));  
            break;  
        case R.id.itemGreen:  
            btnCtx.setTextColor(  
                getResources().getColor(R.color.clrgreen));  
            break;  
        case R.id.itemBlue:  
            btnCtx.setTextColor(  
                getResources().getColor(R.color.clrblue));  
            break; }  
    return super.onContextItemSelected(item);  
    }  
}
```

## 2. Context Menu

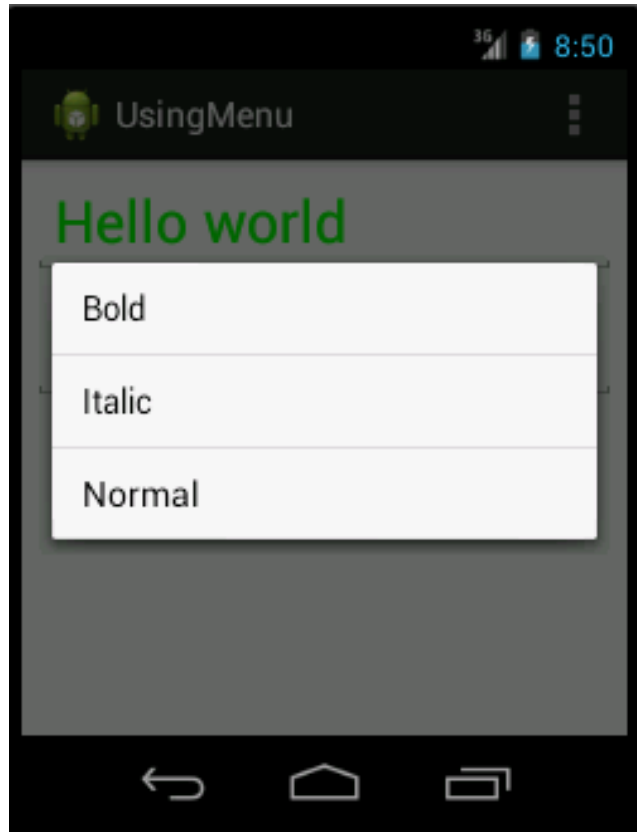
### Giải thích một số dòng lệnh:

- Để đăng ký Context Menu cho đối tượng nào sẽ sử dụng, ta dùng hàm sau : **registerForContextMenu(btnCtx);**
- Thực hiện hàm **onCreateContextMenu()** dùng để nạp Context Menu XML vào ứng dụng. Hàm sẽ được gọi mỗi khi context menu được yêu cầu.
- **getResources().getColor(R.color.clrred)** dùng để lấy màu từ XML Resource
- **onContextItemSelected()** dùng để xử lý sự kiện, để tìm ra xem menu nào được lựa chọn.



## 2. Context Menu

- Demo → ContextMenu



### 3. WebKit Brower

- Với Android, ta có thể nhúng trình duyệt web có sẵn dưới dạng một widget vào trong các activity để hiển thị các nội dung HTML hoặc để duyệt Internet.
- Android browser dựa trên **WebKit**, engine được dùng cho trình duyệt *Safari Web* của Apple.
- Android dùng widget **WebView** để làm chỗ trú cho các trang của trình duyệt
- Ứng dụng dùng **WebView** phải yêu cầu **INTERNET permission**.
- **Permission:** Để Activity truy nhập được Internet và tải các trang web vào một *WebView*, ta phải bổ sung **INTERNET permission** và file Android Manifest:

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

### 3. WebKit Brower

#### Browsing Power

Browser sẽ truy nhập Internet bằng bất cứ cách nào mà từng thiết bị cụ thể cho phép (WiFi, mạng điện thoại, ...).

**WebKit** bao gồm các phương thức để

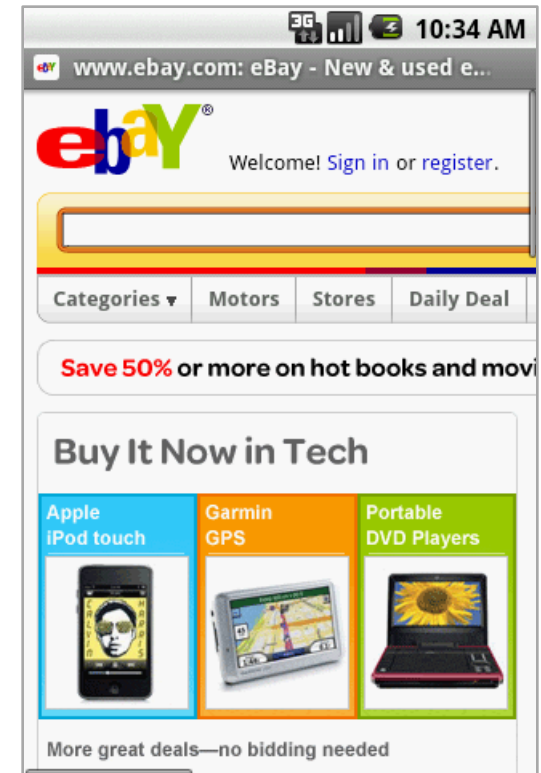
1. navigate forward and backward through a history,
2. zoom in and out,
3. perform text searches,
4. load data
5. stop loading and
6. more.

### 3. WebKit Brower

**Ví dụ: Dùng android truy cập đến trang ebay.**

**Bước 1: khai báo tệp .xml**

```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout
xmlns:android="http://schemas.android.com/ap
k/res/android"
    android:orientation="vertical"
    android:layout_width="fill_parent"
    android:layout_height="fill_parent"
    >
    <WebView
        android:id="@+id/webkit"
        android:layout_width="fill_parent"
        android:layout_height="fill_parent"
    />
</LinearLayout>
```



### 3. WebKit Brower

Ví dụ: Dùng android truy cập đến trang ebay.

Bước 2: viết hàm truy cập trực tiếp tới trang <http://eBay.com>

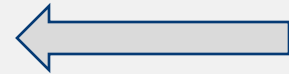
```
package exam.demoui;

import android.os.Bundle;
import android.app.Activity;
import android.webkit.WebView;

public class AndDemoUI extends Activity {
    WebView browser;

    @Override
    public void onCreate(Bundle icle) {
        super.onCreate(icle);
        setContentView(R.layout.main);
        browser=(WebView)findViewById(R.id.webkit);
        browser.loadUrl("http://eBay.com");
        browser.getSettings().setJavaScriptEnabled(true);
    }
}
```

Ứng dụng này  
nổi cồ  
địnhh với eBay



### 3. WebKit Brower

**Ví dụ: Dùng android truy cập đến trang ebay.**

**Bước 3: khai báo permission trong tệp manifest**

```
<?xml version="1.0" encoding="utf-8"?>
<manifest xmlns:android="http://schemas.android.com/apk/res/android"
    package="cis493.demoui" android:versionCode="1" android:versionName="1.0">

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

    <application android:icon="@drawable/icon" android:label="@string/app_name">
        <activity android:name=".AndDemoUI" 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>

    <uses-sdk android:minSdkVersion="3" />

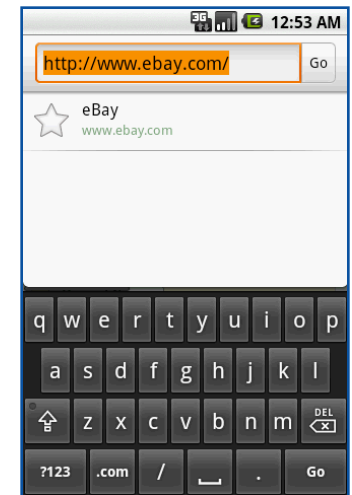
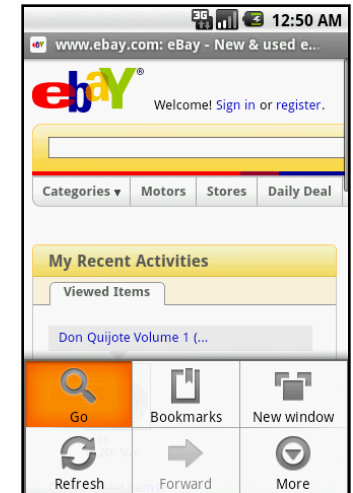
</manifest>
```



### 3. WebKit Brower

#### Chú ý:

- Nếu vào URL của một trang dùng *Javascript*, ta có thể thấy trang trắng trống không.
- Trong các widget WebView, **Javascript** mặc định ở chế độ **off**.
- Để bật Javascript, gọi :  
`myWebView.setSettings().setJavaScriptEnabled(true);`  
đối với đối tượng WebView.
- Với SDK 1.5, WebView có một **Option Menu** dựng sẵn  
Ví dụ : sử dụng lựa chọn Go (tìm kiếm) như hình bên



### 3. WebKit Brower

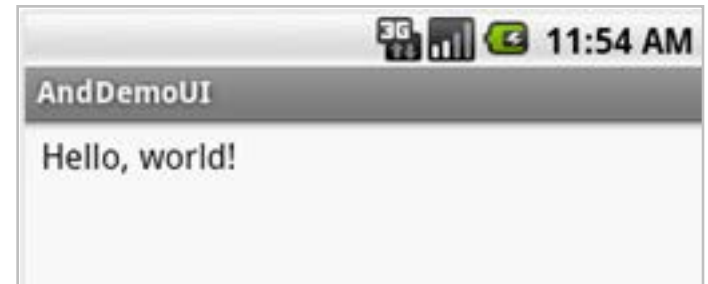
#### Loading Data *.loadData(...)*

Ta có thể cung cấp mã HTML để trình duyệt hiển thị (*chẳng hạn một hướng dẫn sử dụng, hoặc giao diện của ứng dụng thực chất được viết bằng HTML thay vì dùng giao diện native Android*).

```
package exam.demoui;
import android.os.Bundle;
import android.app.Activity;
import android.webkit.WebView;
public class AndDemoUI extends Activity {
    WebView browser;

    @Override
    public void onCreate(Bundle icle) {
        super.onCreate(icle);
        setContentView(R.layout.main);
        browser=(WebView)findViewById(R.id.webkit);

        browser.loadData( "<html><body>Hello, world!</body></html>",
                           "text/html",
                           "UTF-8" );
    }
}
```





### 3. WebKit Brower

#### Browser Commands

Không có navigation toolbar đi kèm mỗi WebView widget (*tiết kiệm không gian*). Ta có thể cung cấp UI – chẳng hạn tại menu – để phục vụ các thao tác sau:

- **reload()** to refresh the currently-viewed Web page
- **goBack()** to go back one step in the browser history, and **canGoBack()** to determine if there is any history to trace back
- **goForward()** to go forward one step in the browser history, and **canGoForward()** to determine if there is any history to go forward to
- **goBackOrForward()** to go backwards or forwards in the browser history, where *negative/positive* numbers represent a count of steps to go
- **canGoBackOrForward()** to see if the browser can go backwards or forwards the stated number of steps (following the same positive/negative convention as **goBackOrForward()**)
- **clearCache()** to clear the browser resource cache and **clearHistory()** to clear the browsing history

### 3. WebKit Brower

## Kết hợp HTML + JAVASCRIPT + ANDROID

### Ưu điểm mà Android Development mang lại

1. Truy nhập các dịch vụ native trên điện thoại, trong đó có các dịch vụ địa điểm (location services)
2. Vị trí tại Android Market
3. Phát triển ứng dụng nhanh bằng Android SDK và Eclipse.

### Lợi thế của Google Maps API

1. Ứng dụng tồn tại ở một server chứ không phải bên trong một thiết bị.
2. Nhanh chóng nâng cấp phiên bản, không cần yêu cầu người dùng tải về và cài đặt phiên bản mới.
3. Google thường xuyên có bổ sung tính năng mới và sửa lỗi.
4. Tương thích giữa các platform compatibility: Dùng Maps API cho phép ta tạo một map duy nhất nhưng chạy trên nhiều platform.
5. Được thiết kế để nạp nhanh tại các thiết bị Android và iPhone.

### 3. WebKit Brower

#### Android & Google Map V3 App (xác định vị trí thực)



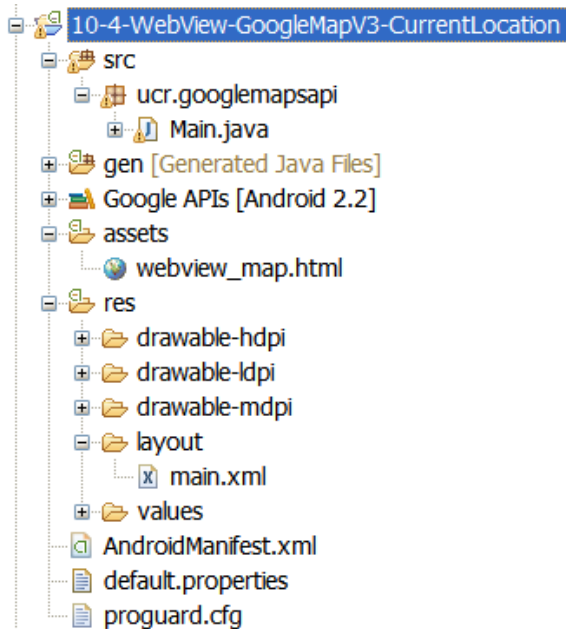
Ví dụ này gồm hai phần:

- Mục tiêu là dùng một đối tượng Android để truyền dữ liệu '*vị trí thực*' cho một trang html.
- Trang html chứa một đoạn mã JavaScript vẽ bản đồ có tâm là tọa độ đã cho.

Vĩ độ (Latitude) và kinh độ (longitude) do thiết bị xác định.  
Ảnh chụp từ điện thoại Android.

**Warning:** Make sure your target is: **Google APIs (API Level 8)** or higher.

### 3. WebKit Brower



Thực hiện các phần sau:

1. Tạo một **WebView** trong tệp main.xml file
2. Tạo một trang html trong thư mục **assets**
3. Thêm yêu cầu **permission** trong manifest
4. Kết nối tới Java code

```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout
xmlns:android="http://schemas.android.com/apk/res
/android"
    android:orientation="horizontal"
    android:layout_width="fill_parent"
    android:layout_height="fill_parent">
    <WebView
        android:id="@+id/webview"
        android:layout_width="fill_parent"
        android:layout_height="fill_parent"/>
</LinearLayout>
```

Add the following permission requests to the AndroidManifest.xml file

```
<uses-permission android:name="android.permission.INTERNET"/>
<uses-permission android:name="android.permission.ACCESS_COARSE_LOCATION" />
<uses-permission android:name="android.permission.ACCESS_FINE_LOCATION" />
```

### 3. WebKit Brower

```
<!DOCTYPE html>
<html>
<head>
<meta name="viewport" content="initial-scale=1.0, user-scalable=no" />
<meta http-equiv="content-type" content="text/html; charset=UTF-8"/>
<title>Google Maps JavaScript API v3 Example: Marker Simple</title>
<style type="text/css">
  html { height: 100% }
  body { height: 100%; margin: 0px; padding: 0px }
  #map_canvas { height: 100% }
</style>
<script type="text/javascript" src="http://maps.google.com/maps/api/js?sensor=false"></script>
<script type="text/javascript">
  function initialize() {
    //var myLatLng = new google.maps.LatLng(41.5020952, -81.6789717);
    var myLatLng = new google.maps.LatLng(locater.getLatitude(), locater.getLongitude());
    var myOptions = {
      zoom: 17,
      center: myLatLng,
      mapTypeId: google.maps.MapTypeId.ROADMAP
    }
    var map = new google.maps.Map(document.getElementById("map_canvas"), myOptions);

    var marker = new google.maps.Marker({
      position: myLatLng,
      map: map
    });
  }
</script>
</head>
<body onload="initialize()">
  <div id="map_canvas"></div>
</body>
</html>
```

Trang webview\_map.html  
tạo một bản đồ sử dụng  
tọa độ mà android cung



### 3. WebKit Brower

## Viết code java trong tệp main.java

```
public class Main extends Activity implements LocationListener {
    private static final String MAP_URL = "http://gmaps-
    samples.googlecode.com/svn/trunk/articles-android-webmap/simple-android-map.html";
    private WebView browser;
    //Location mostRecentLocation;
    LocationManager locationManager;
    MyLocater locater = new MyLocater();

    @Override
    protected void onDestroy() {
        super.onDestroy();
        // cut location service requests
        locationManager.removeUpdates(this);
    }

    private void getLocation() {
        locationManager = (LocationManager) getSystemService(Context.LOCATION_SERVICE);
        Criteria criteria = new Criteria();
        criteria.setAccuracy(Criteria.ACCURACY_FINE); // use GPS (you must be outside)
        //criteria.setAccuracy(Criteria.ACCURACY_COARSE); // towers, wifi
        String provider = locationManager.getBestProvider(criteria, true);
        // In order to make sure the device is getting the location, request
        // updates [wakeup after changes of: 1 sec. or 0 meter]
        locationManager.requestLocationUpdates(provider, 1, 0, this);
        locater.setNewLocation(locationManager.getLastKnownLocation(provider));
    }
}
```

## 3. WebKit Brower

### Viết code java trong tệp main.java (tiếp)

@Override

```
public void onCreate(Bundle savedInstanceState) {  
    super.onCreate(savedInstanceState);  
    setContentView(R.layout.main);  
    getLocation();  
    setupbrowser();  
    this.setRequestedOrientation(ActivityInfo.SCREEN_ORIENTATION_PORTRAIT);  
}
```

/\*\* Sets up the browser object and loads the URL of the page \*/

```
private void setupbrowser() {  
    final String centerURL = "javascript:centerAt("  
        + locator.getLatitude() + ","  
        + locator.getLongitude() + ")";  
  
    // set up the browser to show location results  
    browser = (WebView) findViewById(R.id.webview);  
    browser.getSettings().setJavaScriptEnabled(true);  
    browser.addJavascriptInterface(locater, "locater");  
  
    browser.loadUrl("file:///android_asset/webview_map.html");
```

### 3. WebKit Brower

## Viết code java trong tệp main.java (tiếp)

```
// Wait for the page to load then send the location information
browser.setWebViewClient(new WebViewClient() {
    @Override
    public void onPageFinished(WebView view, String url) {
        browser.loadUrl(centerURL);
    }
});

@Override
public void onLocationChanged(Location location) {
    String lat = String.valueOf(location.getLatitude());
    String lon = String.valueOf(location.getLongitude());
    Toast.makeText(getApplicationContext(), lat + "\n" + lon, 1).show();
    locater.setNewLocation(location);
}

@Override
public void onProviderDisabled(String provider) {
    // needed by Interface. Not used
}

@Override
public void onProviderEnabled(String provider) {
    // needed by Interface. Not used
}

@Override
public void onStatusChanged(String provider, int status, Bundle extras) {
    // needed by Interface. Not used
}
```



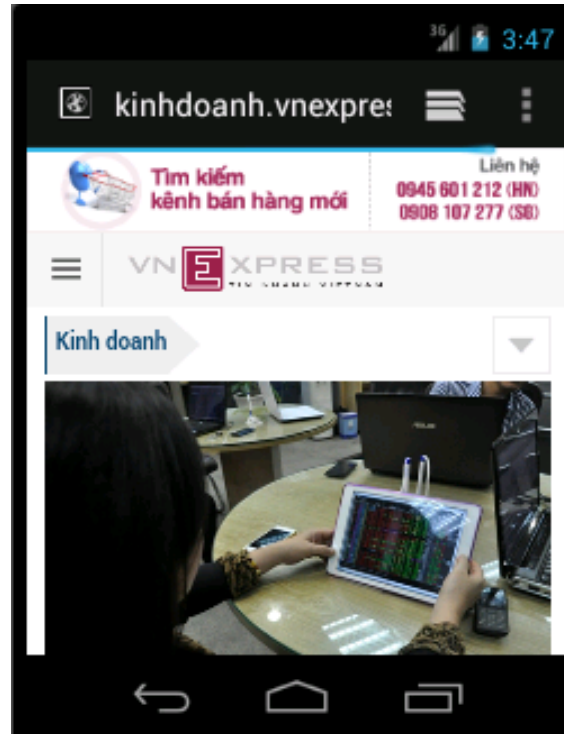
### 3. WebKit Brower

#### Viết code java trong tệp main.java (tiếp)

```
// ////////////////////////////////////////  
// An object of type "MyLocater" will be used to pass data back and  
// forth between the Android app and the JS code behind the html page.  
// ////////////////////////////////////////  
public class MyLocater {  
    private Location mostRecentLocation;  
  
    public void setNewLocation(Location newCoordinates){  
        mostRecentLocation = newCoordinates;  
    }  
  
    public double getLatitude() {  
        if (mostRecentLocation == null) return (0);  
        else return mostRecentLocation.getLatitude();  
    }  
  
    public double getLongitude() {  
        if (mostRecentLocation == null) return (0);  
        else return mostRecentLocation.getLongitude();  
    }  
} // MyLocater  
  
} //class
```

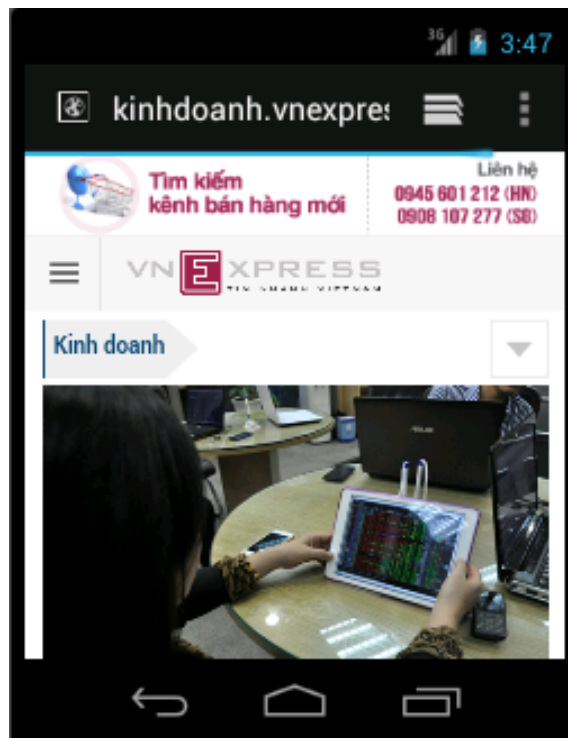
### 3. WebKit Brower

- Demo → WebKit Brower



# Tổng kết bài học

- Demo → Tổng kết bài học



**Kết thúc!**

