



**FPT POLYTECHNIC**

LẬP TRÌNH ANDROID CƠ BẢN

**Bài 8: HTTP, WebView**

---

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

---

## Nội dung bài học

- WebView
- HTTP



## WebView

- View hiển thị trang web
- Hiển thị nội dung online của trang Web trong Activity
- Sử dụng WebKit
- WebView hỗ trợ các phương thức điều hướng trang như sang trang mới, quay trở lại trang cũ, xem lịch sử trang web, phóng to, thu nhỏ, tìm kiếm text
- Để Activity có thể truy cập Internet và tải trang web trong WebView, bạn phải thêm quyền INTERNET trong file Android Manifest

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

## WebView

- Mặc định, WebView không cung cấp widget giống như trình duyệt, không hỗ trợ JavaScript và các lỗi của trang web bị bỏ qua.
- Nếu chỉ muốn hiển thị một số nội dung HTML đơn giản, WebView đáp ứng tốt
- Nếu muốn một trình duyệt web đầy đủ chức năng, bạn nên gọi một ứng dụng Browser đầy đủ với một URL thay vì dùng WebView

```
Uri uri = Uri.parse("http://www.example.com");  
Intent intent = new Intent(Intent.ACTION_VIEW, uri);  
startActivity(intent);
```

# WebView

- Muốn WebView chiếm toàn bộ màn hình Activity sử dụng lệnh sau:

```
WebView webview = new WebView(this);  
setContentView(webview);
```

```
String url = field.getText().toString();  
browser.getSettings().setLoadsImagesAutomatically(true);  
browser.getSettings().setJavaScriptEnabled(true);  
browser.setScrollBarStyle(View.SCROLLBARS_INSIDE_OVERLAY);  
browser.loadUrl(url);
```

## Tùy biến WebView

- Tạo và thiết lập lớp con WebChromeClient. Lớp này được gọi khi có thay đổi trên UI trình duyệt, ví dụ như cập nhật tiến độ hoặc thông báo JavaScript
- Tạo và thiết lập lớp con WebViewClient. Lớp này được gọi khi có vấn đề gì đó xảy ra khi sinh ra nội dung, hoặc lỗi hoặc khi submit form. Bạn cũng có thể chặn tải URL thông qua shouldOverrideUrlLoading()
- Thay đổi WebSettings, như kích hoạt JavaScript sử dụng setJavaScriptEnabled()

# Tùy biến WebView

```
// Let's display the progress in the activity title bar, like the  
// browser app does.
```

```
getWindow().requestFeature(Window.FEATURE_PROGRESS);
```

```
webView.getSettings().setJavaScriptEnabled(true);
```

```
final Activity activity = this;
```

```
webView.setWebChromeClient(new WebChromeClient() {
```

```
    public void onProgressChanged(WebView view, int progress) {
```

```
        // Activities and WebViews measure progress with different scales.
```

```
        // The progress meter will automatically disappear when we reach 100%
```

```
        activity.setProgress(progress * 1000);
```

```
    }
```

```
});
```

```
webView.setWebViewClient(new WebViewClient() {
```

```
    public void onReceivedError(WebView view, int errorCode, String description, String failingUrl) {
```

```
        Toast.makeText(activity, "Oh no! " + description, Toast.LENGTH_SHORT).show();
```

```
    }
```

```
});
```

```
webView.loadUrl("http://developer.android.com/");
```



# DEMO

- Sử dụng webView





## Quản lý cửa sổ và cookie

- Ví lý do bảo mật, mỗi ứng dụng có kho lưu trữ cache và cookie riêng và không chia sẻ dữ liệu trình duyệt với các ứng dụng khác
- Mặc định, yêu cầu mở HTML trên cửa sổ mới bị bỏ qua



## Zoom trên WebView

- Để kích hoạt zoom trên WebView, thiết lập WebSettings.setBuiltInZoomControls(boolean)
- Chú ý: sử dụng zoom khi height hoặc weight có giá trị WRAP\_CONTENT có thể dẫn tới các kết quả không mong muốn và nên tránh



## Hỗ trợ video HTML5

- Để hỗ trợ HTML5 video trên ứng dụng của bạn, bạn cần thiết lập hardware acceleration và thiết lập một WebChromeClient. Đối với chế độ toàn màn hình, miêu tả onShowCustomView(View, WebChromeClient.CustomViewCallback) và onHideCustomView()



## Sử dụng JavaScript trên WebView

- Nếu trang web của bạn có sử dụng JavaScript, bạn phải kích hoạt JavaScript trên WebView
- Mặc định, JavaScript không được kích hoạt. Bạn có thể kích hoạt thông qua WebSettings

```
WebView myWebView = (WebView) findViewById(R.id.webView1);  
WebSettings webSettings = myWebView.getSettings();  
webSettings.setJavaScriptEnabled(true);
```



## Điều hướng trang trên WebView

- Khi người dùng click vào một link trên trang web trong WebView, mặc định Android sẽ tìm ứng dụng có thể xử lý URL này
- Tuy nhiên, bạn có thể nạp chồng hành động này, do đó link sẽ được mở ngay trong WebView hiện tại

```
WebView myWebView = (WebView) findViewById(R.id.webView1);  
myWebView.setWebViewClient(new WebViewClient());
```



## **HTTP access**

- Android có gói java.net của java có thể sử dụng để truy cập tài nguyên mạng. Android cũng có thêm thư viện HttpClient
- Lớp cơ bản để truy cập mạng HTTP trong gói java.net là lớp HttpURLConnection

## HTTP access

- Để truy cập Internet, ứng dụng của bạn phải có quyền `android.permission.INTERNET`
- Để kiểm tra trạng thái mạng trên ứng dụng yêu cầu quyền `android.permission.ACCESS_NETWORK_STATE`



## **Android StrictMode**

- Trong ứng dụng Android, bạn nên tránh thực hiện thao tác tốn nhiều thời gian trên thread giao diện
- StrictMode cho phép thiết lập chính sách trong ứng dụng để tránh làm các công việc không cần thiết





## **Android StrictMode**

- Từ Android 3.0 (Honeycomb), StrictMode được cấu hình sẽ ném ra ngoại lệ `NetworkOnMainThreadException` nếu mạng được truy cập trên thread giao diện



# HttpURLConnection

- Lớp có sẵn trong standard java
- Phù hợp là HTTP Client cho hầu hết ứng dụng



# URLConnection

```
public void SendRequest(View v)
{
    try {
        URL url = new URL("http://www.google.com");
        HttpURLConnection con = (HttpURLConnection) url
            .openConnection();
        readStream(con.getInputStream());
    } catch (Exception e) {
        e.printStackTrace();
    }
}

private void readStream(InputStream in) {
    BufferedReader reader = null;
    try {
        reader = new BufferedReader(new InputStreamReader(in));
        String line = "";
        while ((line = reader.readLine()) != null) {
            System.out.println(line);
        }
    } catch (IOException e) {
        e.printStackTrace();
    } finally {
        if (reader != null) {
            try {
                reader.close();
            } catch (IOException e) {}
        }
    }
}
```

# Kiểm tra tính sẵn sàng của mạng

```
public boolean isNetworkAvailable() {  
    ConnectivityManager cm = (ConnectivityManager)  
        getSystemService(Context.CONNECTIVITY_SERVICE);  
    NetworkInfo networkInfo = cm.getActiveNetworkInfo();  
    // if no network is available networkInfo will be null  
    // otherwise check if we are connected  
    if (networkInfo != null && networkInfo.isConnected()) {  
        return true;  
    }  
    return false;  
}
```



# DEMO

- Sử dụng HttpURLConnection



# Tổng kết nội dung bài học

- WebView
- HTTP



**Kết thúc!**

