

모바일 프로그래밍

13 네트워크

2017 2학기

강승우

안드로이드의 네트워크



네트워크의 연결 상태 확인

- ConnectivityManager 클래스
 - 네트워크 연결 상태를 모니터링
 - 네트워크 연결 상태가 변경되면 브로드캐스트로 알려줌
 - 한 네트워크 연결이 끊기면 다른 네트워크로 연결 시도
 - 애플리케이션이 사용 가능한 네트워크의 상태에 대한 질의를 할 수 있는 API 제공
 - 애플리케이션이 데이터 트래픽 처리에 사용할 네트워크를 요청하고 선택할 수 있는 API 제공

ConnectivityManager 객체 참조 방법

- `ConnectivityManager manager = (ConnectivityManager) getSystemService(Context.CONNECTIVITY_SERVICE);`
- Context 클래스에 정의된 `getSystemService()` 메소드 이용

`Object getSystemService(String name)`

- Service의 이름은 Context 클래스에 정의된 String 상수가 있음
 - 예: `CONNECTIVITY_SERVICE`, `LOCATION_SERVICE`, `ALARM_SERVICE`, `POWER_SERVICE`, 등

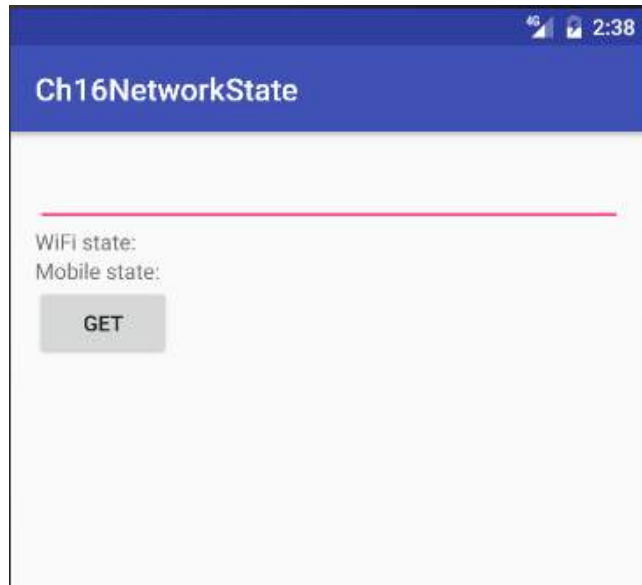
<https://developer.android.com/reference/android/net/ConnectivityManager.html?hl=ko>

연결 상태 확인 예제

- 예제 프로젝트 이름
 - 12_Nework\Ch16NetworkState
- 기능
 - 버튼을 누르면 활성화된 네트워크 상태 확인 후 결과 표시
 - 상태가 변경되면 브로드캐스트를 수신하고 Toast 메시지 표시
- 필요 permission
 - AndroidManifest 파일 permission 추가

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

연결 상태 확인 예제



연결 상태 조회

// 네트워크 상태를 조회할 수 있는 `ConnectivityManager` 객체를 얻음

```
ConnectivityManager manager = (ConnectivityManager) getSystemService(Context.CONNECTIVITY_SERVICE);
```

// 현재 활성화된(active) 네트워크의 정보를 얻음

```
NetworkInfo info = manager.getActiveNetworkInfo();
```

// `NetworkInfo` 객체의 내용을 `String`으로 변환하여 `EditText` 객체인 `display`에 텍스트 내용으로 설정

```
display.setText(info.toString());
```

```
if(info != null) {
```

```
    if(info.getType() == ConnectivityManager.TYPE_WIFI) {
```

```
        if(info.isAvailable() && info.isConnected()) {
```

```
            wifiState.setText("WiFi state: connected");
```

```
        } else if(info.isAvailable() && !info.isConnected()) {
```

```
            wifiState.setText("WiFi state: available, but not connected");
```

```
        }
```

```
    } else if(info.getType() == ConnectivityManager.TYPE_MOBILE) {
```

```
        if(info.isAvailable() && info.isConnected()) {
```

```
            mobileState.setText("Mobile state: connected");
```

```
        } else if(info.isAvailable() && !info.isConnected()) {
```

```
            mobileState.setText("Mobile state: available, but not connected");
```

```
        }
```

```
    }
```

```
}
```

브로드캐스트 수신

@Override

```
protected void onCreate(Bundle savedInstanceState) {
```

```
...  
...
```

IntentFilter를 생성하고
BroadcastReceiver를 등록한다

```
    IntentFilter filter = new IntentFilter(ConnectivityManager.CONNECTIVITY_ACTION);  
    registerReceiver(receiver, filter);  
}
```

네트워크 상태 변경 시 발생하는 브로드캐스트의
action이름

```
...  
...
```

@Override

```
protected void onDestroy() {
```

```
    super.onDestroy();
```

```
    unregisterReceiver(receiver);  
}
```

BroadcastReceiver를 등록 해제한다

브로드캐스트 수신

BroadcastReceiver 객체 생성

```
BroadcastReceiver receiver = new BroadcastReceiver() {
```

```
    @Override
```

```
    public void onReceive(Context context, Intent intent) {
```

```
        String action = intent.getAction();
```

```
        if(action.equals(ConnectivityManager.CONNECTIVITY_ACTION)) {
```

```
            // 브로드캐스트가 disconnect 이벤트에 의해 발생되고, 연결된 네트워크가 전혀 없는 경우
```

```
            // EXTRA_NO_CONNECTIVITY가 true로 설정됨
```

```
            if(intent.getBooleanExtra(ConnectivityManager.EXTRA_NO_CONNECTIVITY, false))
```

```
                Toast.makeText(context, "Network: not connected", Toast.LENGTH_LONG).show();
```

```
            ConnectivityManager manager = (ConnectivityManager) getSystemService(Context.CONNECTIVITY_SERVICE);
```

```
            NetworkInfo info = manager.getActiveNetworkInfo();
```

```
            if(info != null && info.isConnected()) {
```

```
                if(info.getType() == ConnectivityManager.TYPE_WIFI)
```

```
                    Toast.makeText(context, "WiFi Network: connected", Toast.LENGTH_LONG).show();
```

```
                else if(info.getType() == ConnectivityManager.TYPE_MOBILE)
```

```
                    Toast.makeText(context, "Mobile Network: connected", Toast.LENGTH_LONG).show();
```

```
            }
```

```
        }
```

```
    }
```

```
};
```

네트워크 상태 변경 시 발생하는 브로드캐스트의 action 이름

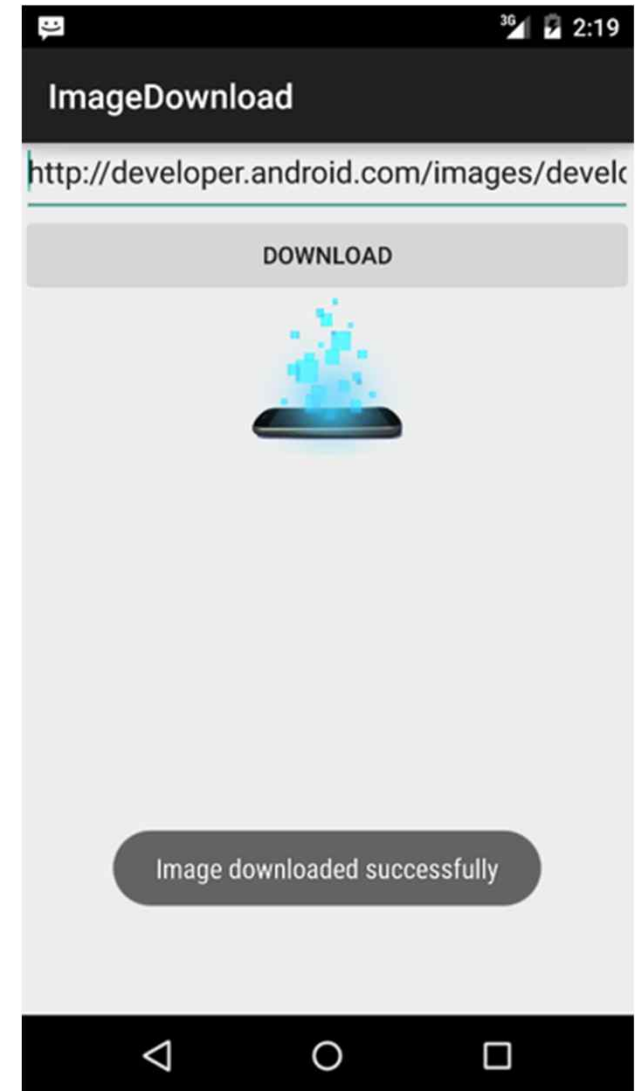
웹에서 파일 다운로드

- 웹 서버에 접속하여 텍스트나 이미지를 다운로드 받을 경우
 - 웹 프로토콜, HTTP 이용
- HttpURLConnection 클래스
 - HTTP 프로토콜을 이용하여 네트워크에서 파일 또는 웹 페이지를 다운로드할 때 사용

<https://developer.android.com/reference/java/net/HttpURLConnection.html?hl=ko>

이전 수업 예제

- 입력된 URL로부터 이미지를 다운로드 하여 화면 상의 이미지 뷰에 표시
- 예제 프로젝트
 - 10_Process_Thread\Ch13ImageAsyncTask
 - Download 버튼을 누르면 EditText 상에 입력된 http 주소에 연결하여 이미지 파일을 다운로드 받고 완료가 되면 화면 상의 ImageView에 표시

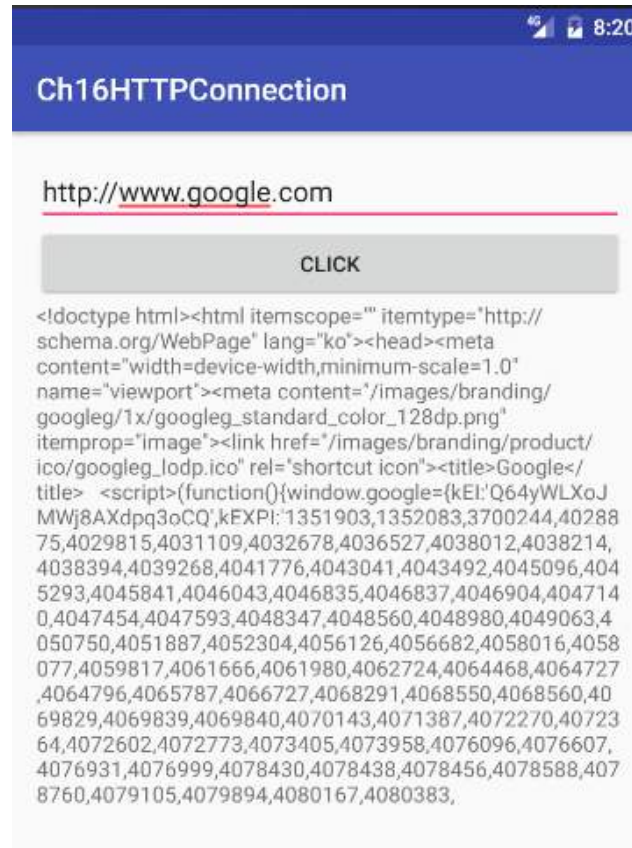
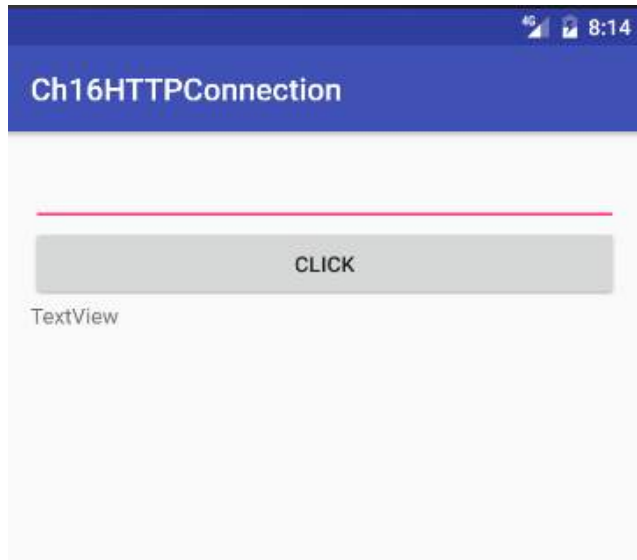


웹 페이지 텍스트 다운로드 예제

- 예제 프로젝트 이름
 - 12_Nework\Ch16URLConnection
- 앞의 예제와 동일하게 HttpURLConnection 을 사용
 - 연결을 맺기 전에 네트워크 상태를 확인하여 현재 네트워크를 사용 가능한지 확인
 - AsyncTask를 활용하여 연결을 맺고 읽어오는 작업 수행
 - 작업이 완료되면 TextView에 내용 표시
- 필요 permission (AndroidManifest 파일)

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

웹 페이지 텍스트 다운로드 예제



- 버튼을 클릭하면 네트워크 상태를 확인
- 연결 상태인 경우 입력된 URL로 HTTP 연결을 생성하고 데이터를 읽음
- 연결을 종료
- 읽은 데이터를 String 객체로 반환하여 TextView에 표시

웹 앱

- 네이티브 앱
 - 안드로이드 SDK를 사용하여 개발하고 APK 형식으로 디바이스에 설치되는 앱
- 웹 앱
 - 웹 기반(HTML, CSS, Javascript)으로 앱을 개발하고 클라이언트에서는 웹 브라우저를 통해 액세스
- 하이브리드 앱
 - 겉은 네이티브 앱으로 되어 있고, 내용을 보여주는 것은 웹 기반으로
 - 내용으로 보여줄 html 파일 혹은 이미지, 자바스크립트 등을 앱 내부에 다 포함하는 방식으로 만들 수도 있고, 인터넷으로 그때 그때 웹 페이지를 다운로드 해서 보여주는 방식으로 만들 수도 있음

WebView

<http://developer.android.com/intl/ko/reference/android/webkit/WebView.html>

- 웹 서버에서 웹 페이지를 읽어서 브라우저처럼 표시할 때 사용
 - View 클래스의 하위 클래스
 - 액티비티 레이아웃 안에 웹 페이지를 표시할 수 있음
 - WebKit 엔진을 이용하여 HTML 문서를 해석해서 화면에 보여줌
 - 상용 웹 브라우저의 모든 기능을 지원하는 것은 아님
- XML 레이아웃 안에 WebView element가 들어간 예제

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

WebView 예제

- 예제 프로젝트 이름
 - 12_Network\Ch16WebView
- 기능
 - HTTP URL을 입력하고 click 버튼을 누르면, Network이 이용 가능한 상태인지 검사하고 가능하면 해당 URL의 웹 페이지를 하단의 WebView에서 보여줌
 - 웹 브라우저에서 보는 것과 비슷하게 동작
- 필요 permission
 - 앞의 예제와 동일

• 추가적인 내용은 교재의 설명과 예제를 참고할 것

