CHAP 13. 네트워크

안드로이드에서의 네트워크

네트워킹 상태 조회

- ConnectivityManager 클래스
  - **Monitor** network connections (Wi-Fi, GPRS, UMTS, etc.)
  - Send **broadcast intents** when network connectivity changes
  - Attempt to "fail over" to another network when connectivity to a network is lost
  - Provide an API that allows applications to **query** the coarse-grained or fine-grained **state** of the available networks
  - Provide an API that allows applications to **request and select** networks for their data traffic

네트워킹 상태 조회

- ConnectivityManager 클래스는 네트워크 연결 상태를 감시하고 만약 네트워크 연결 상태가 변경되면 다른 애플리케이션에게 방송.

```
if (Build.VERSION.SDK_INT < 23) {
    final NetworkInfo activeInfo = connMgr.getActiveNetworkInfo();
    if (activeInfo != null && activeInfo.isConnected()) {
        wifiConnected = (activeInfo.getType() == ConnectivityManager.TYPE_WIFI);
        mobileConnected = (activeInfo.getType() == ConnectivityManager.TYPE_MOBILE);
        if(wifiConnected) {
            Log.i(TAG, getString(R.string.wifi_connection));
        } else if (mobileConnected){
            Log.i(TAG, getString(R.string.mobile_connection));
        }
    } else {
        Log.i(TAG, getString(R.string.no_wifi_or_mobile));
    }

} else {
```

네트워킹 상태 조회

- ConnectivityManager 클래스는 네트워크 연결 상태를 감시하고 만약 네트워크 연결 상태가 변경되면 다른 애플리케이션에게 방송.

```
} else {  // API level >= 23
    final Network n = connMgr.getActiveNetwork();
    if (n != null) {
        final NetworkCapabilities nc = connMgr.getNetworkCapabilities(n);
        wifiConnected = nc.hasTransport(NetworkCapabilities.TRANSPORT_WIFI);
        mobileConnected = nc.hasTransport(NetworkCapabilities.TRANSPORT_CELLULAR);
        if(wifiConnected) {
            Log.i(TAG, getString(R.string.wifi_connection));
        } else if (mobileConnected){
            Log.i(TAG, getString(R.string.mobile_connection));
        }
    } else {
        Log.i(TAG, getString(R.string.no_wifi_or_mobile));
    }
}
```

class NetworkCapabilities

- Constants

Constant	Description
NET_CAPABILITY_INTERNET	Indicates that this network should be able to reach the internet.
NET_CAPABILITY_MMS	Indicates this is a network that has the ability to reach the carrier's MMSC for sending and receiving MMS messages.
NET_CAPABILITY_NOT_ROAMING	Indicates that this network is not roaming.
NET_CAPABILITY_WIFI_P2P	Indicates this is a network that has the ability to reach a Wi-Fi direct peer.
NET_CAPABILITY_NOT_CONGESTED	Indicates that this network is not congested.

class NetworkCapabilities

- Constants

Constant	Description
TRANSPORT_BLUETOOTH	Indicates this network uses a Bluetooth transport.
TRANSPORT_CELLULAR	Indicates this network uses a Cellular transport.
TRANSPORT_WIFI	Indicates this network uses a Wi-Fi transport.
TRANSPORT_VPN	Indicates this network uses a VPN transport.

- Methods

Method	Description
hasCapability(int capability)	Tests for the presence of a capability on this instance.
hasTransport(int transportType)	Indicates this network uses a Cellular transport.

메니페스트 파일 수정

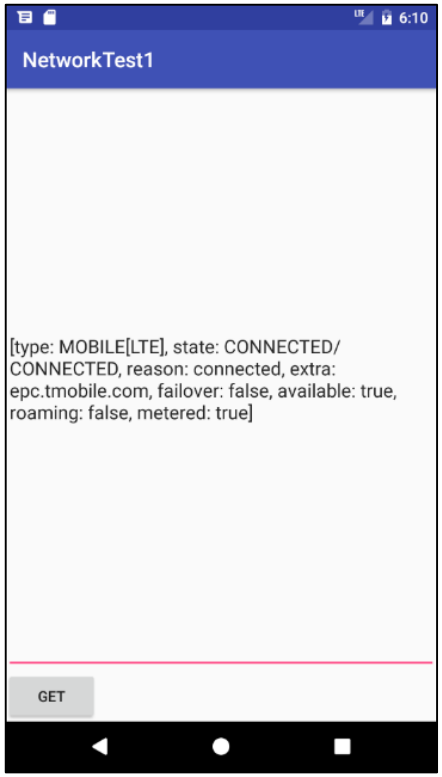
```
...
<uses-permission
        android:name="android.permission.ACCESS_NETWORK_STATE" />
...
```

예제: 현재 네트워크 상태 출력

- 버튼을 누르면 현재 네트워크 상태를 출력한다.

사용자 인터페이스 작성

- 에디트 텍스트와 버튼으로 이루어진 사용자 인터페이스를 작성한다. 자세한 내용은 소스를 참고한다.

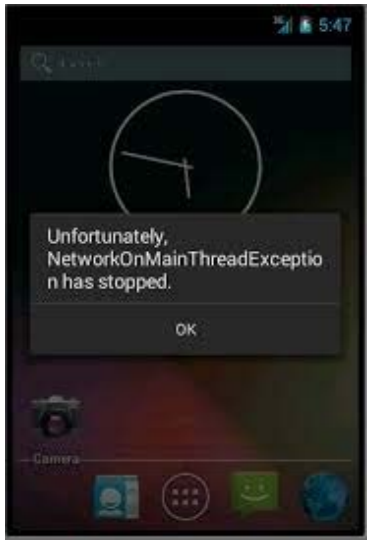


실행 결과


6:10
NetworkTest1
[type: MOBILE[LTE], state: CONNECTED/
CONNECTED, reason: connected, extra:
epc.tmobile.com, failover: false, available: true,
roaming: false, metered: true]
GET
현재 활성화된 네트워크 상태를 보여준다.

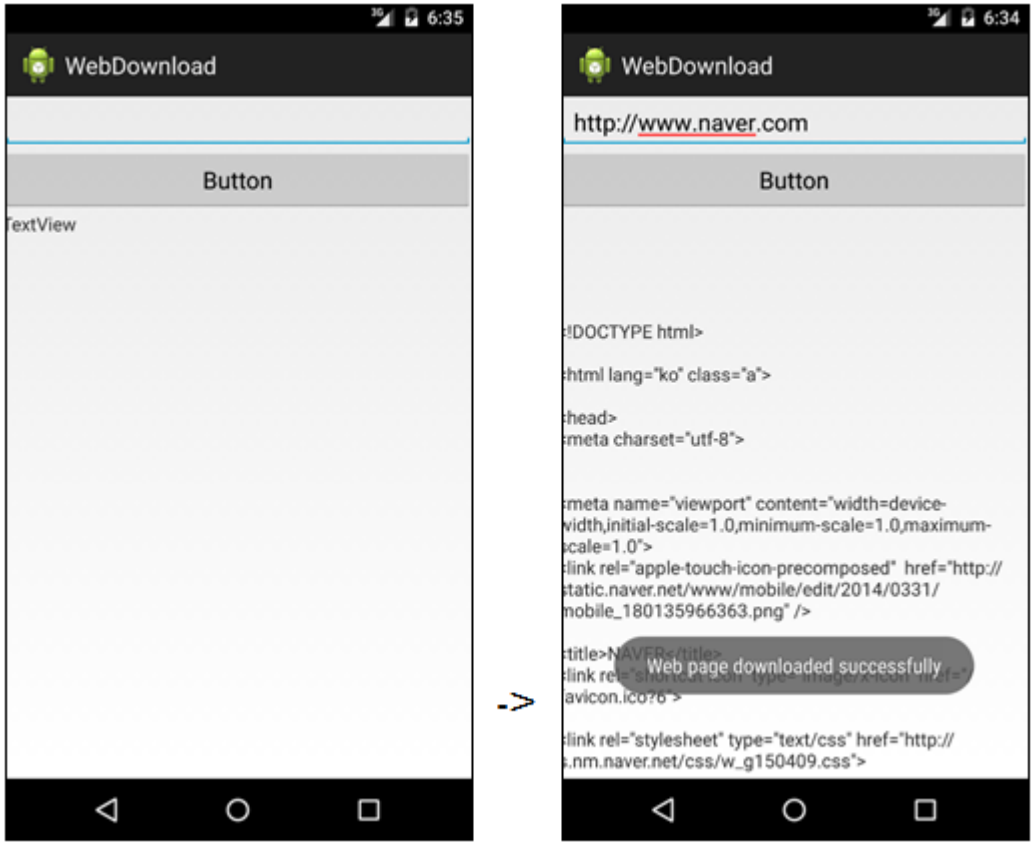
웹에서 파일 다운로드

- HTTP 프로토콜을 이용하여서 네트워크에서 파일을 다운로드 할 때는 HttpURLConnection 사용
- Main thread에서 직접 파일을 다운로드하면 NetworkOnMainThreadException 예외 발생
- 네트워크 보안을 위해 Android Pie부터 clear text(암호화되지 않은 HTTP 전송 문서) 차단.
  - 강제 허용 설정



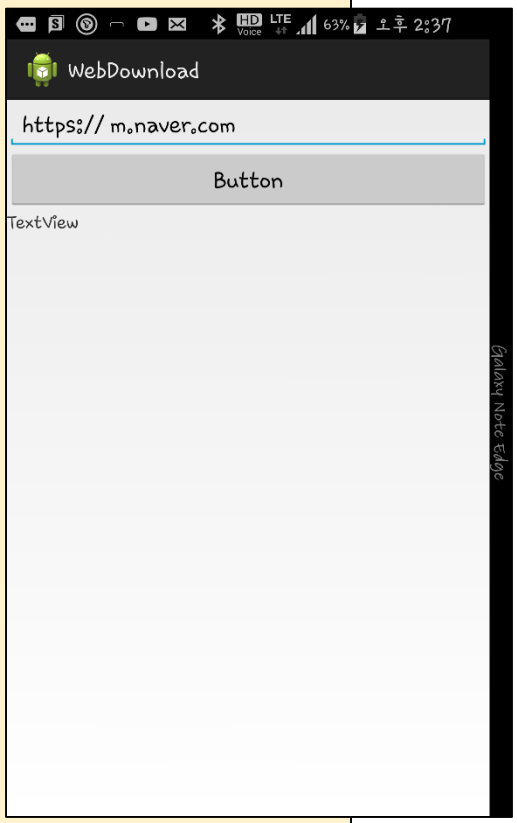

```
<application
    android:allowBackup="true"
    android:usesCleartextTraffic="true"
    android:icon="@drawable/ic_launcher"
    android:label="@string/app_name"
    android:theme="@style/AppTheme" >
    <activity
        android:name="kr.co.company.webdownload.MainActivity"
```

예제: 웹서버에서 웹페이지 다운로드



사용자 인터페이스 작성

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

    <EditText
        android:id="@+id/url"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:ems="10" >
        <requestFocus />
    </EditText>
    <Button
        android:id="@+id/download"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:text="Button" />\
    <TextView
        android:id="@+id/text"
        android:layout_width="match_parent"
        android:layout_height="match_parent"
        android:text="TextView" />
</LinearLayout>
```



코드 작성

```
public class MainActivity extends Activity {
   @Override
   public void onCreate(Bundle savedInstanceState) {
      super.onCreate(savedInstanceState);
      setContentView(R.layout.main);

      Button btnDownload = (Button) findViewById(R.id.download);
      OnClickListener downloadListener = new OnClickListener() {
         @Override
         public void onClick(View v) {
            if (isNetworkAvailable()) {
               EditText url = (EditText) findViewById(R.id.url);
               DownloadTask downloadTask = new DownloadTask();
               downloadTask.execute(url.getText().toString());

            } else {
               Toast.makeText(getBaseContext(),
                     "Network is not Available", Toast.LENGTH_SHORT)
                     .show();
            }
         }

      };
      btnDownload.setOnClickListener(downloadListener);
   }
```

코드 작성

```
private boolean isNetworkAvailable() {
   boolean available = false;
   ConnectivityManager connMgr = (ConnectivityManager) getSystemService(Context.CONNECTIVITY_SERVICE);
   final Network n = connMgr.getActiveNetwork();
   if (n!= null) {
      final NetworkCapabilities nc = connMgr.getNetworkCapabilities(n);
      if (nc != null &&
            (nc.hasTransport(NetworkCapabilities.TRANSPORT_WIFI) ||
                  nc.hasTransport(NetworkCapabilities.TRANSPORT_CELLULAR)))
         available = true;
   }
   return available;
}

private String downloadUrl(String strUrl) throws IOException {
   String s = null;
   byte[] buffer = new byte[1000];
   InputStream iStream = null;
   try {
      URL url = new URL(strUrl);
      HttpURLConnection urlConnection = (HttpURLConnection) url.openConnection();
      urlConnection.connect();
      iStream = urlConnection.getInputStream();
      iStream.read(buffer);
      s = new String(buffer);
   } catch (Exception e) {
      Log.d("Exception download", e.toString());
   } finally {
      iStream.close();
   }
   return s;
}
```

bitmap data

```
bitmap = BitmapFactory.decodeStream(iStream)
```

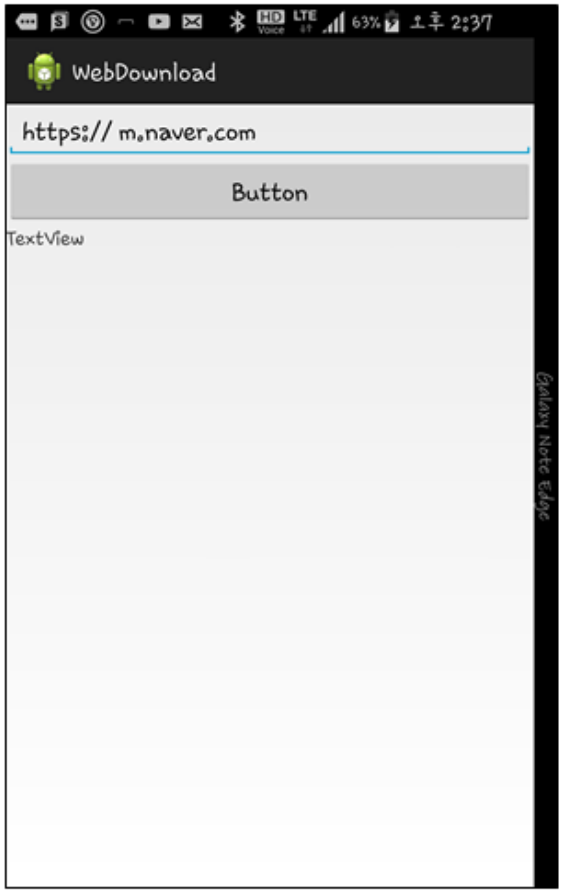
코드 작성

```
    private class DownloadTask extends AsyncTask<String, Integer, String> {
      String s = null;

      @Override
      protected String doInBackground(String... url) {
         try {
            s = downloadUrl(url[0]);
         } catch (Exception e) {
            Log.d("Background Task", e.toString());
         }
         return s;
      }

      @Override
      protected void onPostExecute(String result) {
         TextView tView = (TextView) findViewById(R.id.text);
         tView.setText(result);
         Toast.makeText(getBaseContext(),
               "Web page downloaded successfully", Toast.LENGTH_SHORT)
               .show();
      }
   }
}
```

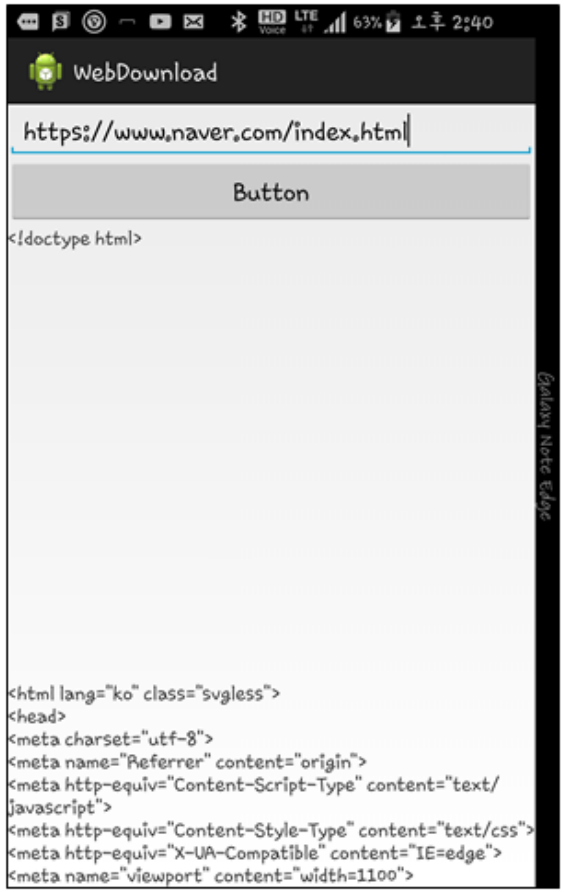
매니페스트 파일 수정

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

실행 결과


63% 오후 2:37
WebDownload
https:// m.naver.com
Button
TextView
Galaxy Note Edge

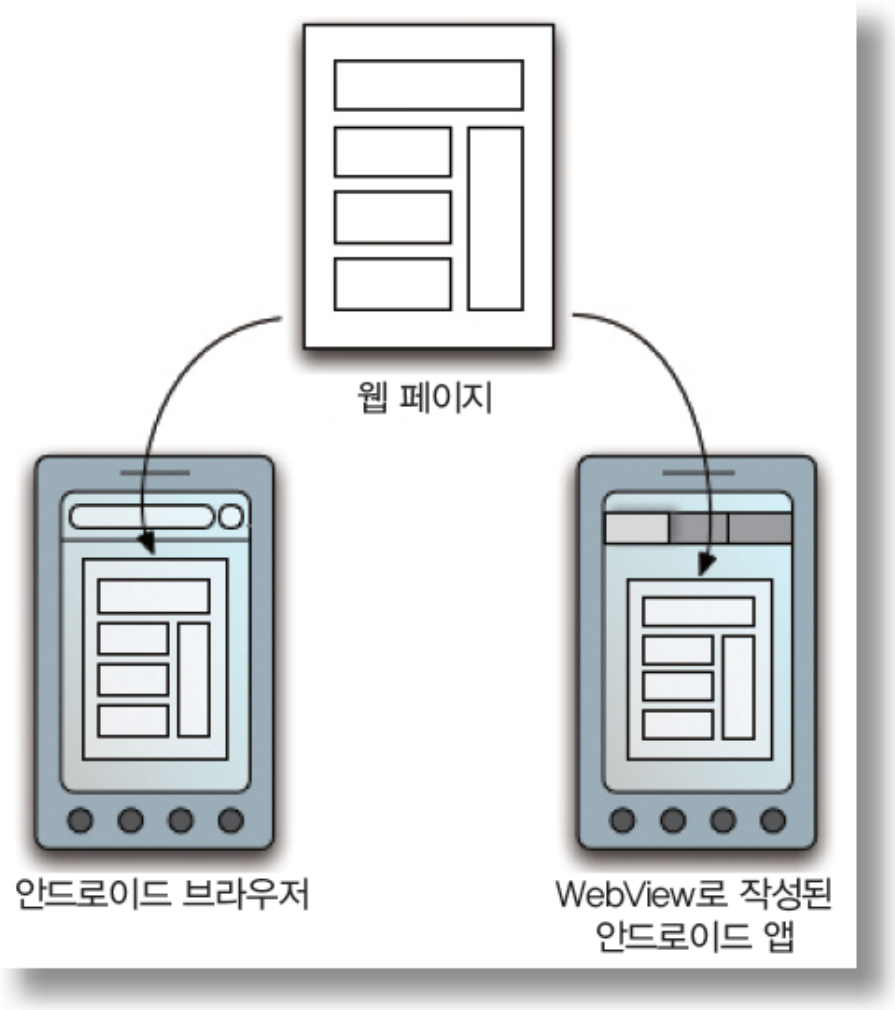
->


63% 오후 2:40
WebDownload
https://www.naver.com/index.html
Button
<!doctype html>
<html lang="ko" class="svgless">
<head>
<meta charset="utf-8">
<meta name="Referrer" content="origin">
<meta http-equiv="Content-Script-Type" content="text/
javascript">
<meta http-equiv="Content-Style-Type" content="text/css">
<meta http-equiv="X-UA-Compatible" content="IE=edge">
<meta name="viewport" content="width=1100">
Galaxy Note Edge

모바일 애플리케이션의 종류

- 안드로이드 SDK 기반
  - 안드로이드 SDK를 사용하여 개발하고 APK 형식으로 사용자 장치에 설치되는 클라이언트 쪽 애플리케이션
- 웹 표준(HTML5) 기반
  - **웹앱(Web App)**으로 웹 표준을 사용하여 개발하고 사용자는 웹 브라우저를 통해 액세스. 사용자 장치에 설치할 필요 없음.

웹앱 개발 방식



Hybrid App = Native App + Web App

Advantage

- Built on web technology (easy to build)
- Faster and less expensive to develop than a native app
- One app for all platforms
- No browser needed
- Make use of device's internal APIs

Disadvantage

- Slower than native apps
- More expensive than web apps
- Less interactive than native apps
- Not appropriate for providing customized features

WebView 위젯

- 안드로이드에서 웹 서버로부터 웹페이지를 읽어서 웹 브라우저처럼 화면에 표시하는 것이 가능할까?
  - **WebView 위젯**을 사용하면 가능하다.

- WebView 위젯은 **WebKit**이라는 엔진을 사용하여서 HTML 문서를 해석하여서 화면에 그려준다.

WebView 사용 방법

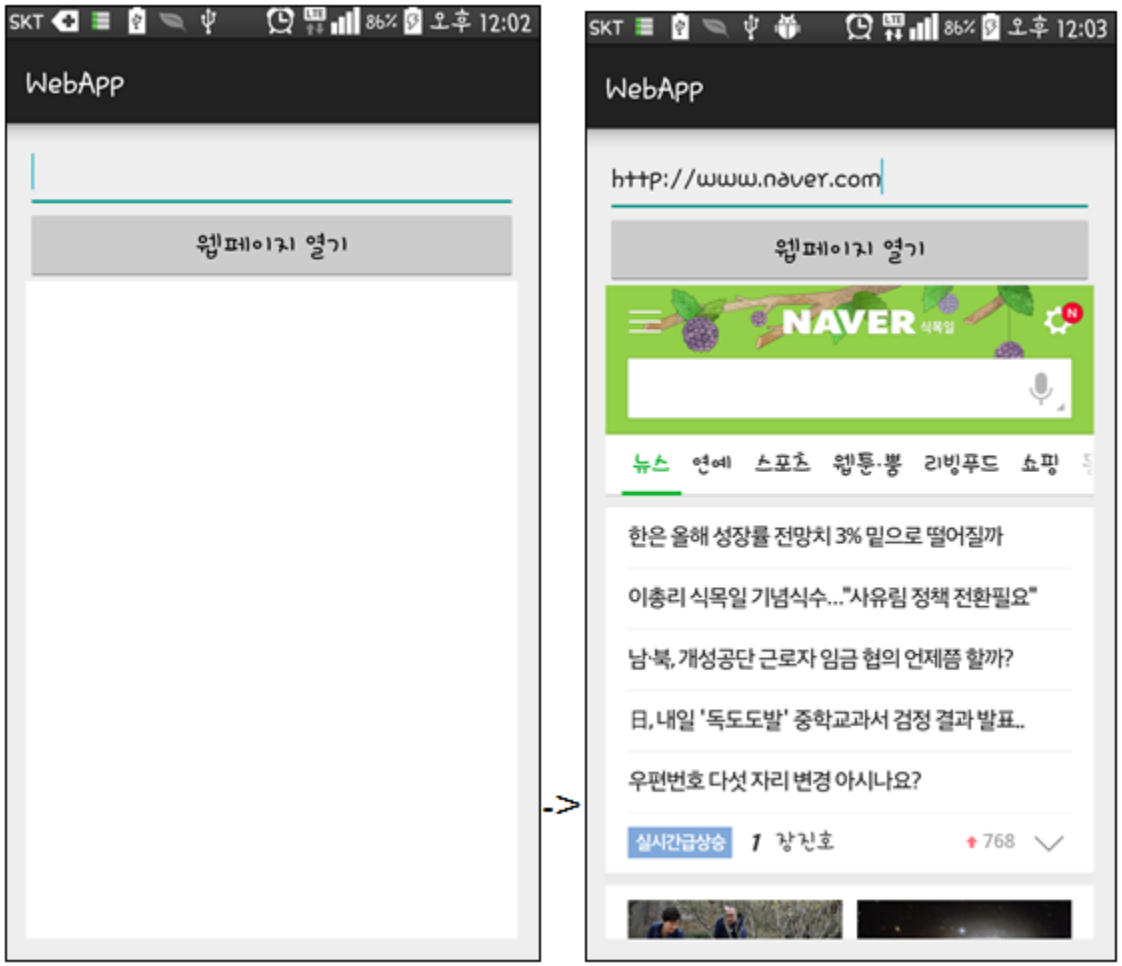
```
public void onCreate(Bundle savedInstanceState) {
    ...
    WebView webview = new WebView(this);
    setContentView(webview);
    ...
}
```

웹뷰를 액티비티의 화면으로 설정한다.

레이아웃 파일

```
<?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"
/>
```

예제: 나만의 웹브라우저 작성



사용자 인터페이스

activity_main.xml

```
<RelativeLayout xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    tools:context=".MainActivity" >

    <EditText
        android:id="@+id/url"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:layout_centerHorizontal="true"
        android:ems="10" />

    <Button
        android:id="@+id/button1"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:layout_below="@+id/url"
        android:layout_centerHorizontal="true"
        android:onClick="open"
        android:text="웹페이지 열기" />

    <WebView
        android:id="@+id/webView"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:layout_alignParentBottom="true"
        android:layout_below="@+id/button1" />

</RelativeLayout>
```

코드

```
public class MainActivity extends AppCompatActivity {

    private EditText field;
    private WebView webview;

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);
        field = (EditText)findViewById(R.id.url);
        webview = (WebView)findViewById(R.id.webView);
        webview.setWebViewClient(new TestBrowser());
    }

    public void open(View view){
        String url = field.getText().toString();
        webview.getSettings().setLoadsImagesAutomatically(true);
        webview.getSettings().setJavaScriptEnabled(true);
        webview.setScrollBarStyle(View.SCROLLBARS_INSIDE_OVERLAY);
        webview.loadUrl(url);
    }

    private class TestBrowser extends WebViewClient {
        @Override
        public boolean shouldOverrideUrlLoading(WebView view, String url) {
            view.loadUrl(url);
            return true;
        }
    }
}
```

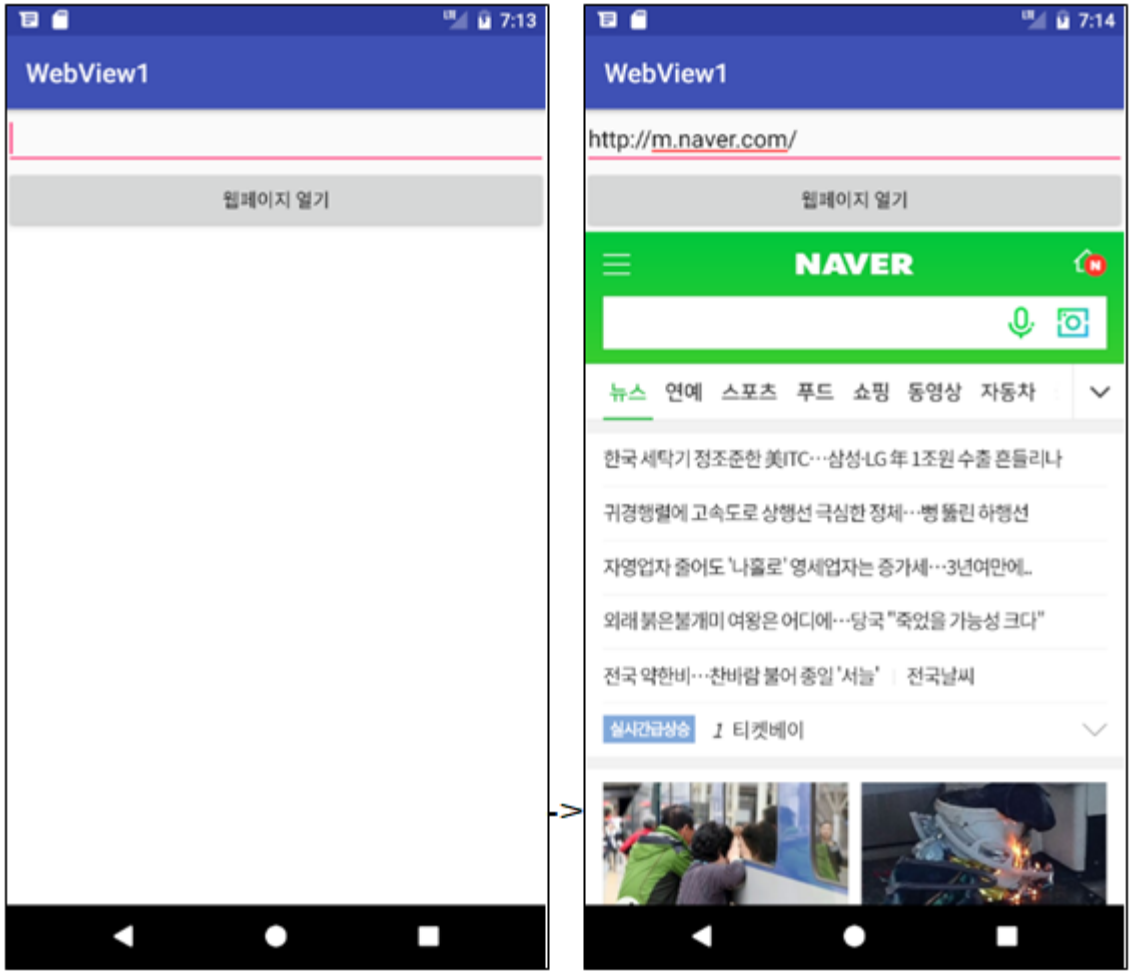
웹뷰 안에서 이벤트가 발생하면 호출

메니페스트 파일

AndroidManifest.xml

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

실행 결과


7:13
WebView1
웹페이지 열기
->
7:14
WebView1
http://m.naver.com/
웹페이지 열기
NAVER
뉴스 연예 스포츠 푸드 쇼핑 동영상 자동차
한국 세탁기 정조준한 美ITC…삼성·LG 年 1조원 수출 흔들리나
귀경행렬에 고속도로 상행선 극심한 정체…뻥 뚫린 하행선
자영업자 줄어도 '나홀로' 영세업자는 증가세…3년여만에..
외래 붉은불개미 여왕은 어디에…당국 "죽었을 가능성 크다"
전국 약한비…찬바람 불어 종일 '서늘' 전국날씨
실시간급상승 1 티켓베이

XML 처리

- 인터넷을 통하여 전달되는 데이터는 주로 XML 형식

```
<?xml version="1.0" encoding="utf-8"?>
<quiz>
   <question>
   안드로이드를 만든 회사는?
   </question>
   <answer>
   구글
   </answer>
<!-- 여기에 퀴즈를 추가한다. -->
</quiz>
```

XML 파서

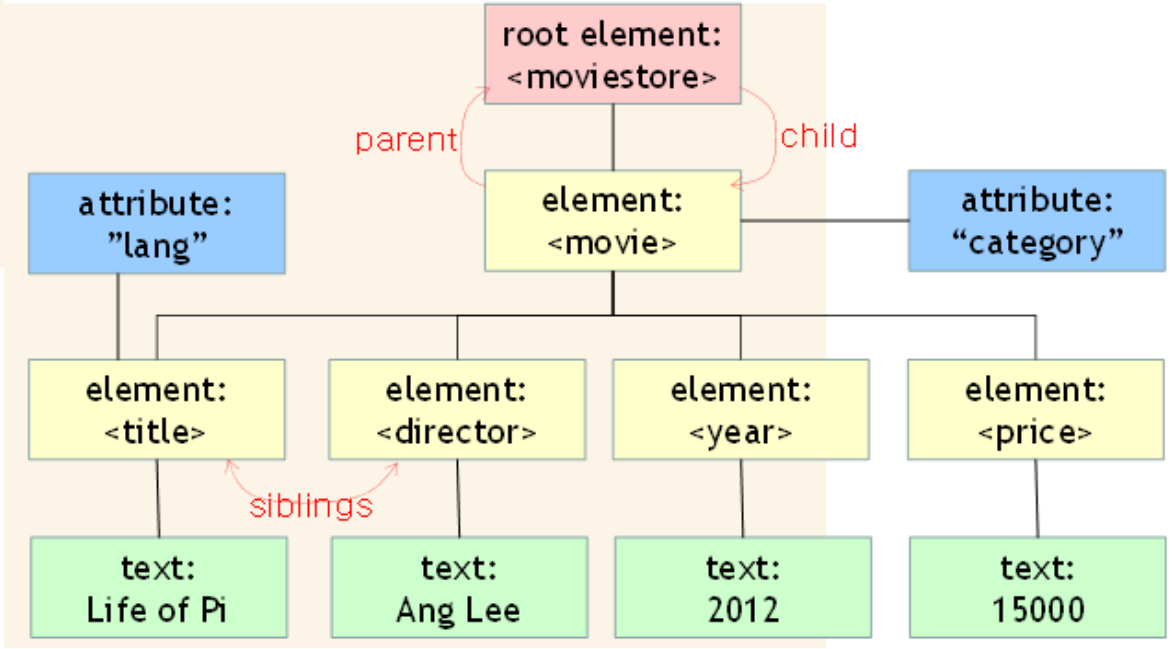
- DOM 파서
- SAX 파서
- PullParser 파서

DOM

- DOM(Document Object Model)은 W3C의 표준으로 XML 문서에 접근하고 처리하는 표준적인 방법을 정의
- DOM은 XML 문서를 트리 구조로 표현한다.
- DOM은 문서 요소의 객체(object), 특징(property), 메소드(interface)를 정의

DOM

```
<?xml version="1.0" encoding="ISO-8859-1"?>
 <moviestore>
    <movie category="drama">
      <title lang="en">Life of Pi</title>
      <director>Ang Lee</director>
      <year>2012</year>
      <price>15000</price>
    </movie>
    <movie category="fantasy">
      <title lang="en">Hobbit</title>
      <director>Peter Jackson</director>
      <year>2012</year>
      <price>20000</price>
    </movie>
 </moviestore>
```



DOM XML Parsing

- DocumentBuilder 준비

```
DocumentBuilderFactory dbf = DocumentBuilderFactory.newInstance();
DocumentBuilder builder = null;
try {
    builder = dbf.newDocumentBuilder();
} catch (ParserConfigurationException e) {
    e.printStackTrace();
}
```

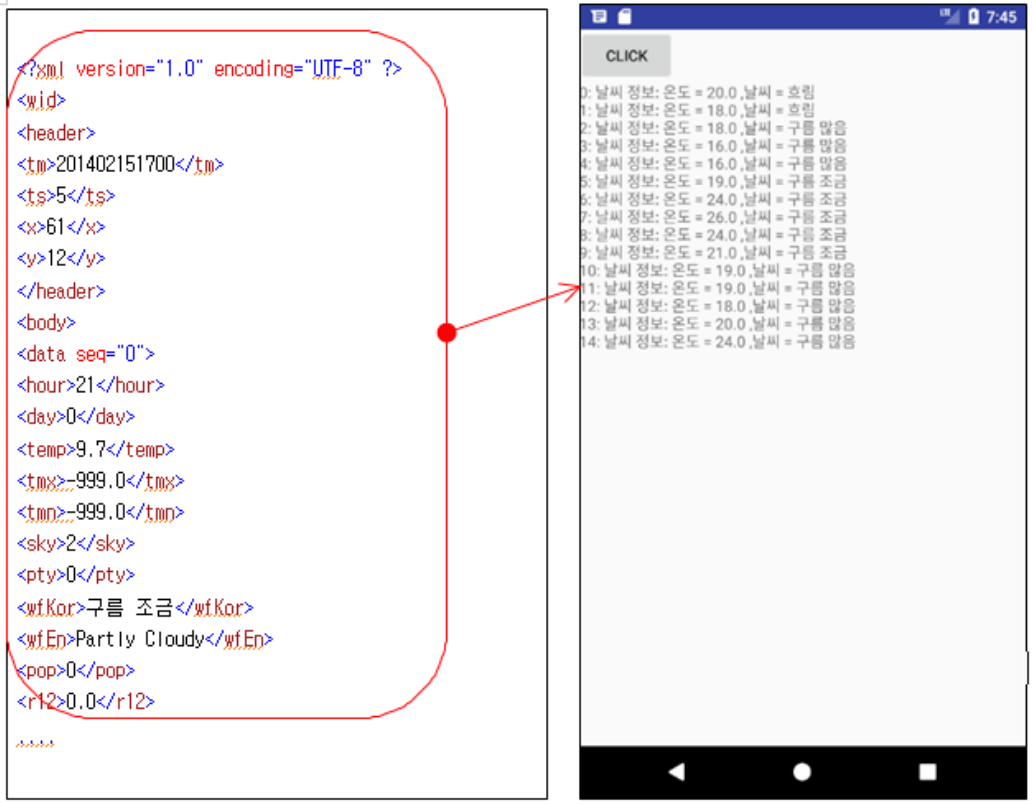
- XML 파일 파싱

```
try {
    Document doc = builder.parse(new FileInputStream("data\\text.xml"));
} catch (SAXException e) {
    e.printStackTrace();
} catch (IOException e) {
    e.printStackTrace();
}
```

- Document 객체(doc)의 methods를 이용하여 DOM 트리 사용

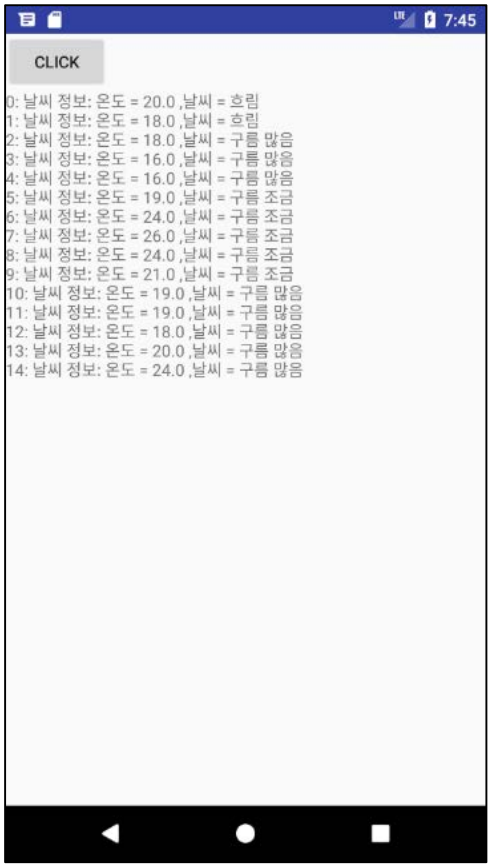
예제

- 기상청에서 제공하는 특정한 위치의 정보를 해석하여서 화면에 시간에 따른 온도와 날씨만을 표시하는 애플리케이션



사용자 인터페이스

- 사용자 인터페이스를 XML로 정의



코드 작성

```
public class MainActivity extends Activity {
    TextView textview;
    Document doc = null;
    LinearLayout layout = null;

    @Override
    public void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);
        textview = (TextView) findViewById(R.id.textView1);

    }

    public void onClick(View view) {
        GetXMLTask task = new GetXMLTask(this);
        task.execute("http://www.kma.go.kr/wid/queryDFS.jsp?gridx=61&gridy=125");
    }
```

코드 작성

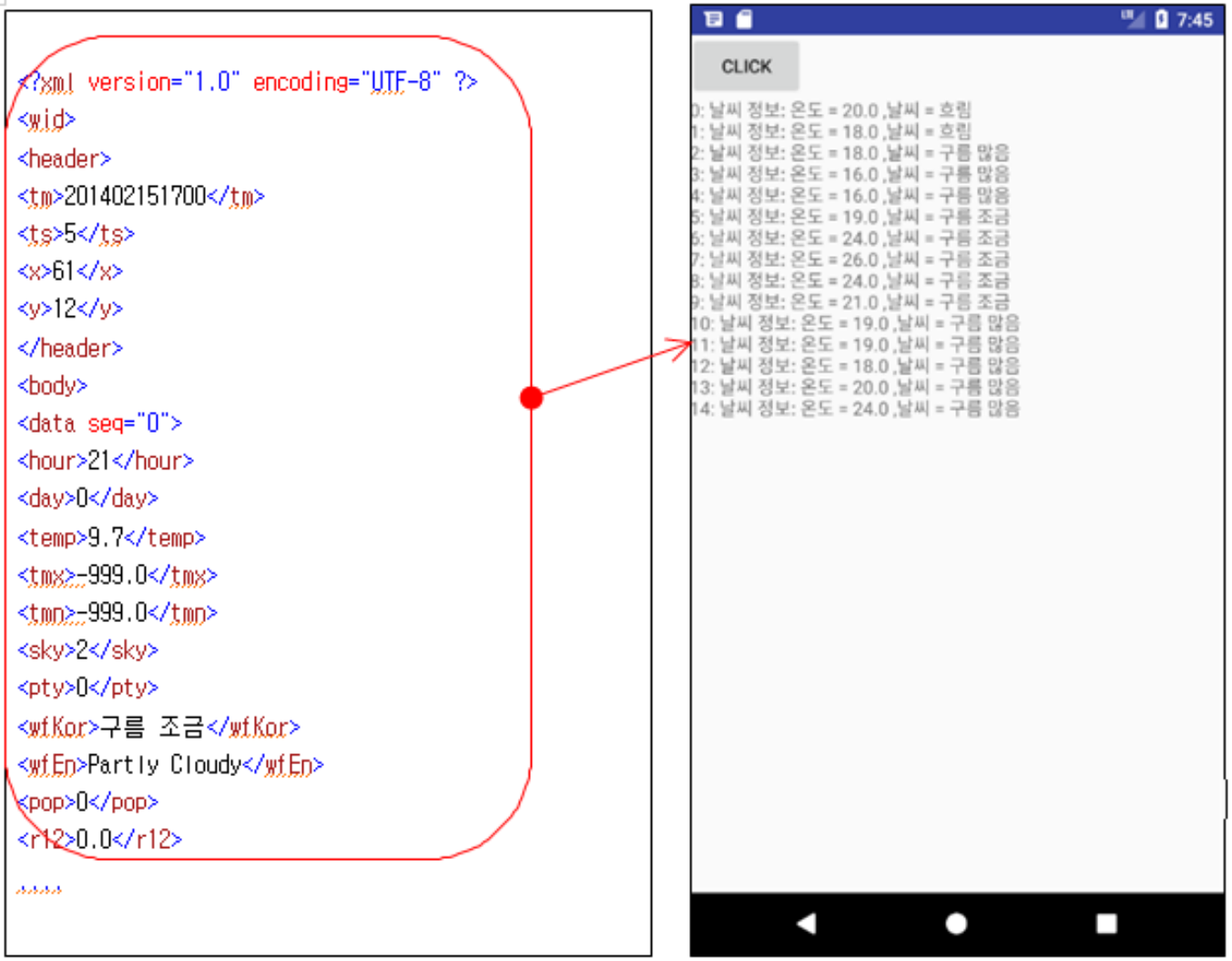
```
private class GetXMLTask extends AsyncTask<String, Void, Document> {
    private Activity context;

    public GetXMLTask(Activity context) {
        this.context = context;
    }

    @Override
    protected Document doInBackground(String... urls) {
        URL url;
        try {
            url = new URL(urls[0]);
            DocumentBuilderFactory dbf = DocumentBuilderFactory
                    .newInstance();
            DocumentBuilder db;

            db = dbf.newDocumentBuilder();
            doc = db.parse(new InputSource(url.openStream()));
            doc.getDocumentElement().normalize();

        } catch (Exception e) {
            Toast.makeText(getBaseContext(), "Parsing Error",
                    Toast.LENGTH_SHORT).show();
        }
        return doc;
    }
    ...
```

코드 작성

```
    ...
        protected void onPostExecute(Document doc) {
            String s = "";
            NodeList nodeList = doc.getElementsByTagName("data");

            for (int i = 0; i < nodeList.getLength(); i++) {
                s += "" + i + ": 날씨 정보: ";
                Node node = nodeList.item(i);
                Element fstElmnt = (Element) node;
                NodeList nameList = fstElmnt.getElementsByTagName("temp");
                Element nameElement = (Element) nameList.item(0);
                nameList = nameElement.getChildNodes();

                s += "온도 = " + ((Node) nameList.item(0)).getNodeValue() + " ,";

                NodeList websiteList = fstElmnt.getElementsByTagName("wfKor");
                Element websiteElement = (Element) websiteList.item(0);
                websiteList = websiteElement.getChildNodes();
                s += "날씨 = " + ((Node) websiteList.item(0)).getNodeValue()
                        + "\n";
            }
            textview.setText(s);
        }

    }
}
```

```
<wid>
<header>
<tm>201905221700</tm>
<ts>5</ts>
<x>61</x>
<y>125</y>
</header>
<body>
<data seq="0">
   <hour>21</hour>
   <day>0</day>
   <temp>20.0</temp>
   <tmx>-999.0</tmx>
   <tmn>-999.0</tmn>
   <sky>1</sky>
   <pty>0</pty>
   <wfKor>맑음</wfKor>
   <wfEn>Clear</wfEn>
   <pop>0</pop>
   <r12>0.0</r12>
   <s12>0.0</s12>
   <ws>2.5</ws>
   <wd>4</wd>
   <wdKor>남</wdKor>
   <wdEn>S</wdEn>
   <reh>50</reh>
   <r06>0.0</r06>
   <s06>0.0</s06>
</data>
```

실행 결과


<?xml version="1.0" encoding="UTF-8" ?>
<wid>
<header>
<tm>201402151700</tm>
<ts>5</ts>
<x>61</x>
<y>12</y>
</header>
<body>
<data seq="0">
<hour>21</hour>
<day>0</day>
<temp>9.7</temp>
<tmx>-999.0</tmx>
<tmn>-999.0</tmn>
<sky>2</sky>
<pty>0</pty>
<wfKor>구름 조금</wfKor>
<wfEn>Partly Cloudy</wfEn>
<pop>0</pop>
<r12>0.0</r12>
.....
7:45
CLICK
0: 날씨 정보: 온도 = 20.0 ,날씨 = 흐림
1: 날씨 정보: 온도 = 18.0 ,날씨 = 흐림
2: 날씨 정보: 온도 = 18.0 ,날씨 = 구름 많음
3: 날씨 정보: 온도 = 16.0 ,날씨 = 구름 많음
4: 날씨 정보: 온도 = 16.0 ,날씨 = 구름 많음
5: 날씨 정보: 온도 = 19.0 ,날씨 = 구름 조금
6: 날씨 정보: 온도 = 24.0 ,날씨 = 구름 조금
7: 날씨 정보: 온도 = 26.0 ,날씨 = 구름 조금
8: 날씨 정보: 온도 = 24.0 ,날씨 = 구름 조금
9: 날씨 정보: 온도 = 21.0 ,날씨 = 구름 조금
10: 날씨 정보: 온도 = 19.0 ,날씨 = 구름 많음
11: 날씨 정보: 온도 = 19.0 ,날씨 = 구름 많음
12: 날씨 정보: 온도 = 18.0 ,날씨 = 구름 많음
13: 날씨 정보: 온도 = 20.0 ,날씨 = 구름 많음
14: 날씨 정보: 온도 = 24.0 ,날씨 = 구름 많음