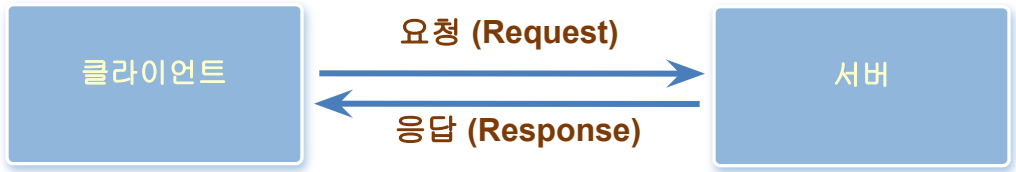
1. 네트워킹이란?

네트워킹이란 ?

■ 원격지의 서버를 연결하는 방식

• **2-tier C/S 모델**

- 클라이언트와 서버가 일대일로 연결하는 방식



[2-tier C/S 모델]

• **3-tier 모델**

- 서버를 좀 더 유연하게 구성
- 응용 서버와 데이터 서버로 구성하는 경우, 데이터베이스를 분리시킴

[3-tier 모델]

소켓 사용하기

■ 네트워킹에 대한 이해

- TCP/IP 수준의 통신 방식을 제공하는 소켓을 이용해 서버에 연결해 보면 이해하기 쉬움
- 일반적인 프로그래밍에서는 대부분 TCP 연결 사용
- 비연결성(stateless) 특성으로 인해 실시간으로 데이터를 처리하는 애플리케이션의 경우,

  응답 속도를 높이기 위해 HTTP보다 소켓 연결 선호

■ 소켓 연결 방식

- 안드로이드에서는 표준 자바의 소켓을 그대로 사용할 수 있음
- 서버쪽에는 서버소켓을 만들어 실행함 (포트 지정)
- 클라이언트쪽에서는 소켓을 만들어 서버소켓으로 연결함 (IP와 포트 지정)
- Stream 객체를 이용해 데이터를 보내거나 받을 수 있음

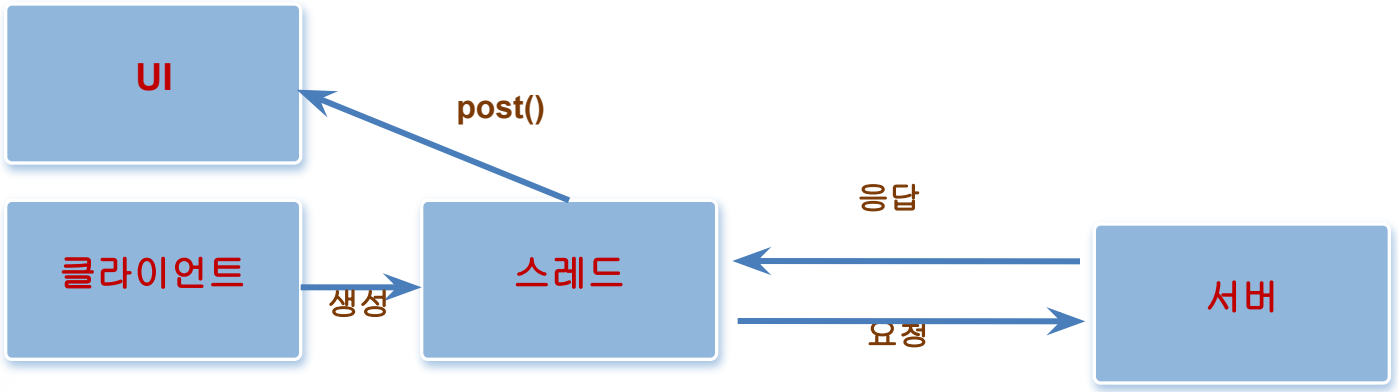
네트워킹 사용 시 주의할 점

■ 네트워킹을 사용할 때는 반드시 스레드 사용

- 최신 버전의 안드로이드에서는 네트워킹을 사용할 때는 반드시 스레드를 사용하도록 변경되었음 (이전에는 스레드 없이도 가능했음)

■ 스레드를 사용하므로 UI 업데이트를 위해서는 반드시 핸들러 사용

- 네트워킹을 위해 새로 만든 스레드 안에서 그 결과를 보여주기 위해 UI 업데이트를 하는 경우 스레드 부분에서 공부한 바와 같이 핸들러를 사용해야 함
- 가장 간단한 방법으로 post() 메소드 사용 권장



2. 웹으로 요청하기

■ HTTP 연결 방식

- 예전 휴대 단말은 데이터 통신의 송수신 속도가 느려서 소켓으로 연결하거나 웹 페이지를 보기 위해서는 많이 기다려야 함
- 비연결성(stateless)인 HTTP 프로토콜은 새로 연결을 만드는 데 따른 지연 시간이 길게 발생
- 최근 스마트폰 및 무선 네트워크 환경이 좋아져서 HTTP 프로토콜을 이용한 웹의 사용이 자연스러울 뿐만 아니라 일반 웹 사이트를 보는 풀 브라우징(full browsing)도 가능함
- 자바에서 사용하던 HTTP 관련 클래스를 그대로 사용할 수 있음

URLConnection 클래스

■ URLConnection 객체

[API]

public URLConnection openConnection()

* HttpURLConnection 객체로 형변환(casting)하여 사용

■ 요청 관련 메소드

[API]

public void setRequestMethod(String method)

public void setRequestProperty(String field, String newValue)

웹으로 요청하기 예제

웹으로 요청하기 예제

-HTTP 프로토콜을 이용해 웹 페이지 요청

메인 액티비티의
XML 레이아웃 정의

-메인 액티비티 레이아웃 정의

메인 액티비티 코드 작성

-버튼을 누르면 **HTTP**로 웹페이지 요청

XML 레이아웃 만들기

```
<EditText
  android:id="@+id/input01"
  android:layout_width="wrap_content"
  android:layout_height="wrap_content"
  android:hint="사이트 주소 입력 ... "
  android:textSize="18dp"
  >
</EditText>

<Button
  android:id="@+id/requestBtn"
  android:layout_width="wrap_content"
  android:layout_height="wrap_content"
  android:text="HTTP 요청"
  android:textSize="20dp"
  android:textStyle="bold"
  >
</Button>
```

```
<ScrollView
  android:layout_width="wrap_content"
  android:layout_height="wrap_content"
  >

  <TextView
    android:id="@+id/txtMsg"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:background="#ff99ccee"
    android:textColor="#ff0000ff"
    android:textSize="12dp"
    >
  </TextView>
</ScrollView>
```

로그를 화면에 보여주기 위한 구성

메인 액티비티 코드 만들기

웹으로 요청하기 위해 새로 정의한 **request()** 메소드 호출

```
requestBtn.setOnClickListener(new OnClickListener() {
    public void onClick(View v) {
        String urlStr = input01.getText().toString();

        ConnectThread thread = new ConnectThread(urlStr);
        thread.start();
    }
});
```

Continued..

메인 액티비티 코드 만들기 (계속)

```
public void run() {
  try {

    final String output = request(urlStr);
    handler.post(new Runnable() {
      public void run() {
        txtMsg.setText(output);
      }
    });
  } catch(Exception ex) {
    ex.printStackTrace();
  }
}

private String request(String urlStr) {
  StringBuilder output = new StringBuilder();
  try {
    URL url = new URL(urlStr);
```

결과물을 텍스트뷰에 표시

URL 문자열을 이용해 URL 객체 생성

Continued..

메인 액티비티 코드 만들기 (계속)

```
 HttpURLConnection conn = (HttpURLConnection)url.openConnection();
if (conn != null) {
  conn.setConnectTimeout(10000);
  conn.setRequestMethod("GET");
  conn.setDoInput(true);
  conn.setDoOutput(true);

  int resCode = conn.getResponseCode();
  if (resCode == HttpURLConnection.HTTP_OK) {
    BufferedReader reader = new BufferedReader(new InputStreamReader(conn.getInputStream())) ;
    String line = null;
    while(true) {
      line = reader.readLine();
      if (line == null) {
        break;
      }
      output.append(line + "\n");
    }
    reader.close();
    conn.disconnect();
…
```

URL 객체를 이용해 HttpURLConnection 객체 생성

서버 접속하여 요청

응답 결과를 읽기 위한 스트림 객체 생성

반복문 안에서 한 줄씩 읽어 결과 문자열에 추가

실행 화면

HttpURLConnection으로 웹페이지 요청 화면

3. 뉴스정보 가져오기

뉴스정보 가져오기

■ 뉴스 정보

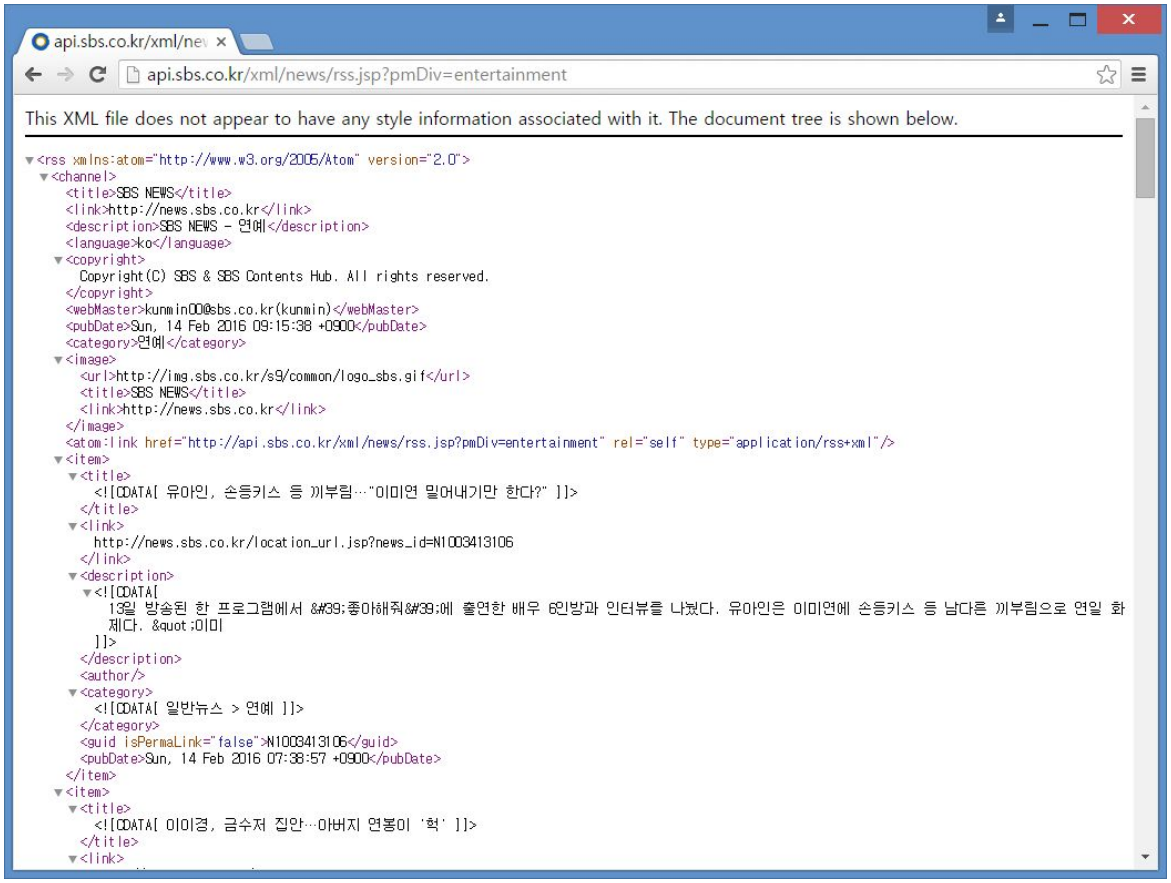
- 최신 뉴스를 다루는 정보는 **RSS**를 통해 제공하는 경우가 많음
- **RSS** 문서에는 뉴스의 제목과 내용 그리고 만든 날짜 등과 함께 **RSS** 채널 자체의 정보가 메타데이터로 들어감

RSS 문서 페이지 예시

웹 브라우저에서 RSS 사이트 조회 (참조 : SBS 뉴스-연예)

http://news.sbs.co.kr/news/newsSection.do?sectionType=14

RSS 문서 소스를 웹 브라우저로 본 경우



웹 브라우저에서 RSS 사이트 조회 (참조 : SBS 뉴스-연예 RSS)

http://api.sbs.co.kr/xml/news/rss.jsp?pmDiv=entertainment

뉴스정보 가져오기 예제

뉴스정보 가져오기 예제

-RSS로 된 뉴스정보 가져오기 기능

리스트뷰 수정

-이전에 사용했던 리스트뷰 수정

메인 액티비티 코드 작성

-스레드로 RSS 요청 및 파싱

-버튼 눌렀을 때 이벤트 처리

메인 액티비티 코드 만들기

```
list = new RSSListView(this);
adapter = new RSSListAdapter(this);
list.setAdapter(adapter);


list.setOnDataSelectionListener(new OnDataSelectionListener() {
   public void onDataSelected(AdapterView parent,View v,int position, long id) {
      RSSNewsItem curItem = (RSSNewsItem) adapter.getItem(position);
      String curTitle = curItem.getTitle();
      Toast.makeText(getApplicationContext(),"Selected : " + curTitle, 1000).show();
   }
});
```

Continued..

메인 액티비티 코드 만들기 (계속)

```
private void showRSS(String urlStr) {
    try {
        progressDialog = ProgressDialog.show(this, "RSS Refresh", "RSS 정보 업데이트 중...", true, true);
        RefreshThread thread = new RefreshThread(urlStr);
        thread.start();
    } catch (Exception e) {
       Log.e(TAG, "Error", e);
    }
…
```

Continued..

메인 액티비티 코드 만들기 (계속)

```
private int processDocument(Document doc) {
    newsItemList.clear();

    Element docEle = doc.getDocumentElement();


    NodeList nodelist = docEle.getElementsByTagName("item");
    int count = 0;
    if ((nodelist != null) && (nodelist.getLength() > 0)) {
      for (int i = 0; i < nodelist.getLength(); i++) {

        RSSNewsItem newsItem = dissectNode(nodelist, i);
        if (newsItem != null) {

          newsItemList.add(newsItem);
          count++;
        }
      }
    }

    return count;
}
```

"item" 태그를 가진 노드 리스트 확인

"item" 태그를 가진 노드를 해석해서 RSSNewsItem 객체로 생성

뉴스 아이템들을 담는 리스트 객체에 RSSNewsItem 객체 추가

Continued..

메인 액티비티 코드 만들기 (계속)

```
private RSSNewsItem dissectNode(NodeList nodelist, int index) {
  RSSNewsItem newsItem = null;

  try {

    Element entry = (Element) nodelist.item(index);

    Element title = (Element) entry.getElementsByTagName("title").item(0);
    Element link = (Element) entry.getElementsByTagName("link").item(0);
    Element description = (Element) entry.getElementsByTagName("description").i
    NodeList pubDataNode = entry.getElementsByTagName("pubDate");
    if (pubDataNode == null) {
      pubDataNode = entry.getElementsByTagName("dc:date");
    }
    Element pubDate = (Element) pubDataNode.item(0);
    Element author = (Element) entry.getElementsByTagName("author").item(0);
    Element category = (Element) entry.getElementsByTagName("category").item(0);

    String titleValue = null;
    if (title != null) {
      titleValue = title.getFirstChild().getNodeValue();
    }
    String linkValue = null;
    if (link != null) {
      linkValue = link.getFirstChild().getNodeValue();
    }
```

현재 Element 객체 참조

"title" 태그를 가진 Element 객체 참조

Continued..

메인 액티비티 코드 만들기 (계속)

```
    String descriptionValue = null;
    if (description != null) {
      descriptionValue = description.getFirstChild().getNodeValue();
    }
    String pubDateValue = null;
    if (pubDate != null) {
      pubDateValue = pubDate.getFirstChild().getNodeValue();
    }
    String authorValue = null;
    if (author != null) {
      authorValue = author.getFirstChild().getNodeValue();
    }
    String categoryValue = null;
    if (category != null) {
      categoryValue = category.getFirstChild().getNodeValue();
    }

    newsItem = new RSSNewsItem(titleValue, linkValue, descriptionValue,
                    pubDateValue, authorValue, categoryValue);
  } catch (DOMException e) {
    e.printStackTrace();
  }

  return newsItem;
}
```

필요한 태그의 정보들을 이용해 **RSSNewsItem** 객체 생성

Continued..

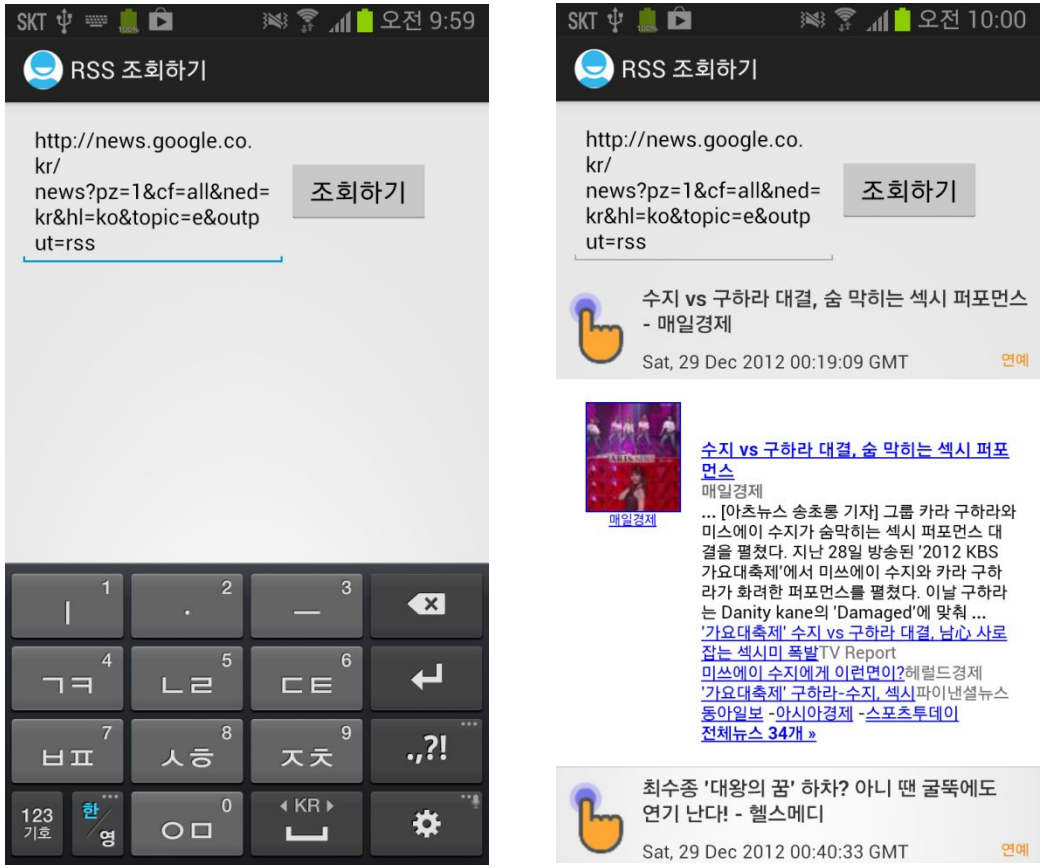
메인 액티비티 코드 만들기 (계속)

```
Runnable updateRSSRunnable = new Runnable() {
  public void run() {
   try {
    Resources res = getResources();
    Drawable rssIcon = res.getDrawable(R.drawable.rss_icon);
    for (int i = 0; i < newsItemList.size(); i++) {
     RSSNewsItem newsItem = (RSSNewsItem) newsItemList.get(i);
     newsItem.setIcon(rssIcon);


     adapter.addItem(newsItem);
    }


    adapter.notifyDataSetChanged();
    progressDialog.dismiss();
   } catch(Exception ex) {
    ex.printStackTrace();
   }
  }
 };
}
```

RSSNewsItem 객체를 어댑터에 추가

화면에 보여지는 리스트뷰 업데이트

실행 화면



화면애플리케이션에서 **RSS** 사이트 조회 화면

참고 문헌

[References]

• 기본 서적

2016, 정재곤, “Do it! 안드로이드 앱 프로그래밍(개정3판)”, 이지스퍼블리싱(주)

• Android Website

http://www.android.com/

• Google Developer’s Conference

http://code.google.com/events/io/

• Android SDK Documentation