

9장. Android Network

1. Http 기반의 네트워크 통신

* 비동기 처리 => 멀티태스킹

Thread --> AsyncTask

* AsyncTask 사용이유

앱은 메인 스레드와 작업 스레드의 협력으로 원활히 돌아 갈 수 있다.

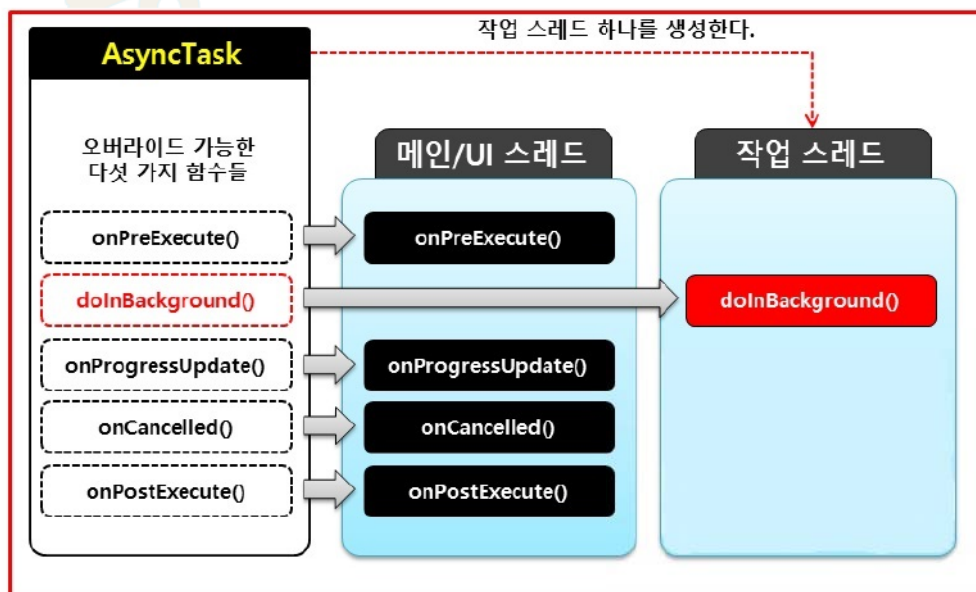
하지만 이렇게 스레드마다 처리해야 하는 일이 엄격히 구분되는 것은 개발에 있어 부담감이 크다.

안드로이드에서는 이러한 복잡성을 줄이기 위해 다양한 도우미 클래스를 제공한다.

도우미 클래스는 개발 시간을 단축시키고 소스를 간결하게 정리해줄 뿐만 아니라,

가독성을 높여 유지 보수하기 편하다.

AsyncTask는 안드로이드에서 요구하는 메인 스레드와 작업 스레드의 분리 구조를 보다 쉽게 구현하도록 도와주는 추상 클래스다.



* Http 상태 코드

404 : Page Not Found

200 : OK

500 : 웹 프로그램 에러

* Loopj Asynchronous Http Client

- <http://loopj.com/android-async-http/>

- android-async-http-1.4.4.jar

- 크기가 25KB로 비교적 작으며, 비동기를 기본으로 지원하는 라이브러리이다.

Instagram, Pinterest 등 유명한 애플리케이션에 사용되었다.

용량을 적게 차지하면서도 Request Cancellation, back-off Policy 지원, Binary File Converter, SharedPreferences를 이용한 Persistent Cookie Store 등 유용한 기능을 많이 제공한다.

단점은 AsyncHttpClient.javaApache HttpClient에 대해 강한 의존성이 있다는 것이다.

요청 전후의 작업을 하는 Interceptor나 backoff policy 등에서 Apache HttpClient의 기능을 직접 사용하고 있다.

(1) MAMP (My Apach-MySQL-PHP) 설치

- Apache 및 Nginx 웹서버와 PHP 모듈, MySQL 서버를 묶어서 설치 및 관리할 수 있는 도구
- 이 도구는 Mac OS X 및 Windows 모두 지원하며, 일반 버전은 무료로 제공됩니다.

1) 설치

- 사이트 : <http://www.mamp.info>
- MAMP를 다운로드한다. => setup_MAMP_3_0_1.exe
- 따로 변경없이 기본 설정으로 설치한다.

2) 사용방법

- 바탕화면의 MAMP 아이콘을 실행한다.
- Winkey + R -> cmd 엔터 -> ipconfig 엔터

(실행 결과 중에서)

무선 LAN 어댑터 Wi-Fi:

연결별 DNS 접미사. :

링크-로컬 IPv6 주소 : fe80::25ca:dc57:70af:8452%3

IPv4 주소 : 192.168.0.10

서브넷 마스크 : 255.255.255.0

기본 게이트웨이 : 192.168.0.1

- 내 컴퓨터의 ip 주소를 복사해 놓는다.
- 안드로이드 앱에서 서버 접근 방법
=> 인터넷 주소 : <http://www.naver.com>
=> ip 주소 : <http://192.168.0.10>

3) 서버 폴더 만들기

- D:\Android\server_data
- MAMP 설정 변경
Preferences 클릭 -> "Web Server" 탭 클릭 -> "Select" 버튼 클릭
-> "D:\Android\server_data" 선택 -> OK 클릭
- 작업 파일들을 저장한다.
- 파일은 반드시 "utf-8"로 저장한다.

(2) 안드로이드 스튜디오에 라이브러리 추가하기

- 해당 프로젝트 폴더/app/libs 폴더 안에 라이브러리 파일 (.jar) 복사하기
- 라이브러리 파일 (.jar) 선택 후, 우클릭 - Add as Library
- build.gradle 파일에 추가되어있는지 확인

```
dependencies {  
    compile fileTree(include: ['*.jar'], dir: 'libs')  
    compile 'com.android.support:appcompat-v7:21.0.3'  
    compile files('libs/android-async-http-1.4.4.jar')  
}
```

예제1. Http 기반의 네트워크 통신

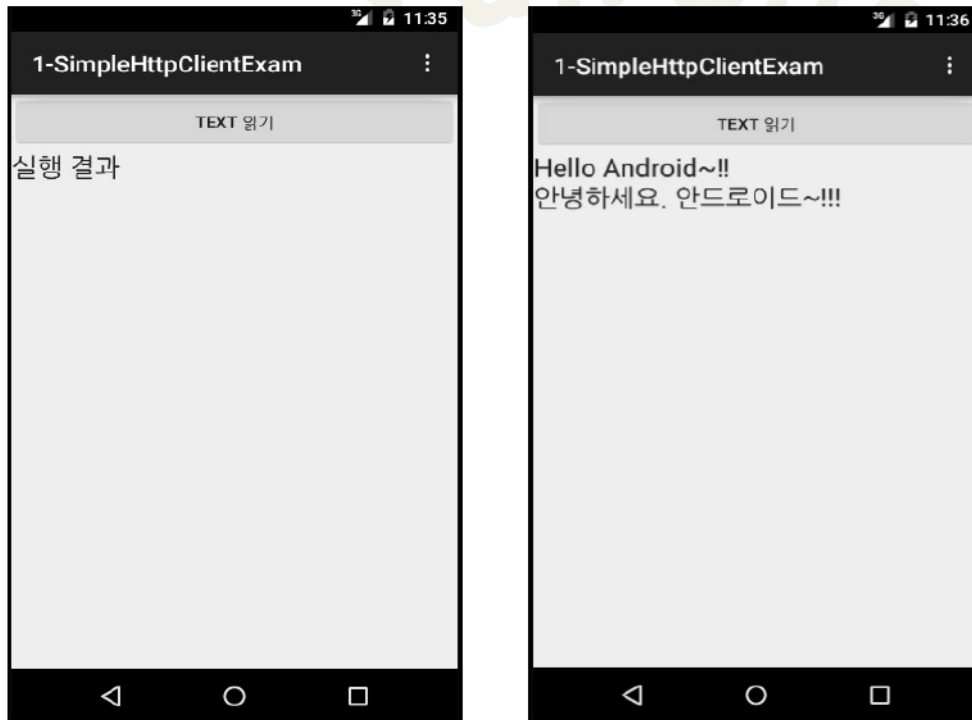
Application Name : 1-SimpleHttpClientExam

Company Domain : .example.com

Package Name : com.example.simplehttpclientexam

Project location : D:\Android\workspace\ch09\1-SimpleHttpClientExam

<실행 결과>



<작업 순서>

- MAMP를 실행하여, 아파치 서버를 동작시킨다.
- 서버에 저장할 파일을 만들고, 저장한다.
- "android-async-http-1.4.4.jar"를 라이브러리에 추가한다.
- activity_main.xml 파일 기본 설정
 - => Design에서 Button을 1개 추가한다.
 - => Design에서 Large Text를 1개 추가한다.
 - => 실행 결과처럼 설정을 수정한다.
- MainActivity.java에서 이벤트 처리를 한다.
- AndroidManifest.xml에 퍼미션을 추가한다.
 - => <uses-permission android:name="android.permission.INTERNET"/>

① 리소스 파일

1) D:/android/server_data/simple_text.txt ("utf-8"로 저장)

① app/res/layout/activity_main.xml

② app/java/com.example.simplehttpclientexam/MainActivity.java

③ app/manifests/AndroidManifest.xml

2. GET/POST 방식의 데이터 전송

* GET 방식

- URL 주소뒤에 파라미터를 붙여서 전달하는 방식
- 글자수가 최대 2048글자로 제한되어 있다.
- 보안에 취약하다.
- 북마크가 가능하고, “뒤로가기”가 보장된다.

* POST 방식

- URL 주소뒤에 파라미터를 붙이지 않고, HTTP Request 헤더에 포함시켜 전달하는 방식
- Data 길이 제한이 없다.
- 보안이 유지된다.
- POST 요청은 캐시되지 않으며 브라우저 히스토리에도 남아있지 않다.
- 북마크가 안된다.
- POST 요청을 한 후에 “뒤로가기” 버튼을 누르면, 데이터를 다시 보내야한다는 브라우저의 경고가 나온다

* HTTP Request 헤더

```
POST /test/input.jsp HTTP/1.1
Host: www.naver.com
name1=value1&name2=value2
...
```

예제2. GET/POST 방식의 데이터 전송

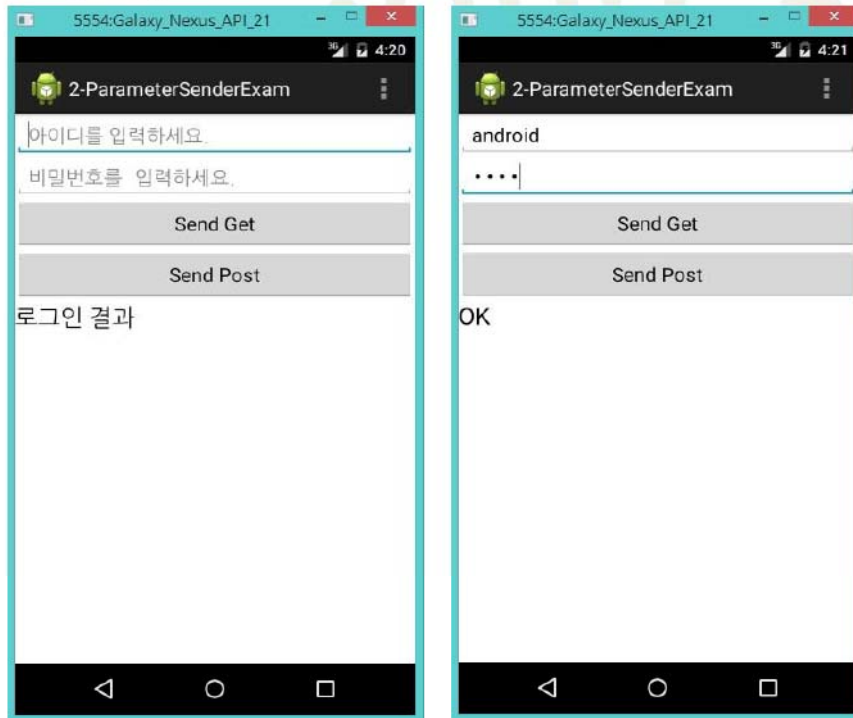
Application Name : 2-ParameterSenderExam

Company Domain : .example.com

Package Name : com.example.parametersenderexam

Project location : D:\Android\workspace\ch09\2-ParameterSenderExam

<실행 결과>



<작업 순서>

- MAMP를 실행하여, 아파치 서버를 동작시킨다.
- 서버에 저장할 파일을 만들고, 저장한다.
- "android-async-http-1.4.4.jar"를 라이브러리에 추가한다.
- activity_main.xml 파일 기본 설정
 - => Design에서 EditText를 2개 추가한다.
 - => Design에서 Button을 2개 추가한다.
 - => Design에서 Large Text를 1개 추가한다.
 - => 실행 결과처럼 설정을 수정한다.
- MainActivity.java에서 이벤트 처리를 한다.
- AndroidManifest.xml에 퍼미션을 추가한다.

① 리소스 파일

1) D:/android/server_data/send_get.php

2) D:/android/server_data/send_post.php

① app/res/layout/activity_main.xml

② app/java/com.example.parametersenderexam/MainActivity.java

③ app/manifests/AndroidManifest.xml

3. JSON 데이터의 활용

(1) 파일 포맷

1) CSV (comma separated value)

- 쉼표를 기준으로 항목을 구분하여 저장한 데이터를 말한다.
- 데이터베이스나 표 계산 소프트웨어 데이터를 보존하기 위해 이런 형식을 사용한다.
- 각 항목이나 판매 내용마다 쉼표(comma)로 구분하여 기록한다.
- CSV 형식의 파일은 텍스트 파일로 보존하여 문서 처리기나 편집기에서 열람·편집할 수 있다.
- 수많은 애플리케이션에서 취급하는 범용 형식이기 때문에 PDA(personal digital assistant)와 PC 사이에 주소록이나 표의 데이터를 주고 받을 때에도 데이터 파일을 CSV 형식으로 변환해서 송수신하는 경우가 많다.

2) XML (extensible markup language)

- 인터넷 웹페이지를 만드는 HTML을 획기적으로 개선하여 만든 언어이다.
- 홈페이지 구축기능, 검색기능 등이 향상되었고, 웹 페이지의 추가와 작성이 편리해졌다.
- HTML보다 홈페이지 구축 기능, 검색 기능 등이 향상되었고 클라이언트 시스템의 복잡한 데이터 처리를 쉽게 한다. 또한 인터넷 사용자가 웹에 추가할 내용을 작성, 관리하기에 쉽게 되어 있다.
- HTML은 웹 페이지에서 데이터베이스처럼 구조화된 데이터를 지원할 수 없지만, XML은 사용자가 구조화된 데이터베이스를 뜻대로 조작할 수 있다.

3) JSON (JavaScript Object Notation)

- 자바스크립트 표준 객체 표기법
- 자바스크립트 객체를 문자로 표현 가능
- 순수 문자열만을 사용함으로 오버헤드를 최대한 줄일 수 있다.
- 객체 방식이므로 파싱할 필요없이 손쉽게 접근 가능
- 인터넷에서 자료를 주고 받을 때 그 자료를 표현하는 방법이다.
- 그 형식은 자바스크립트의 구문 형식을 따르지만, 프로그래밍 언어나 플랫폼에 독립적이다.
- 입체적인 데이터를 표현하기 위한 경량화된 표현 기법
- 현재는 거의 모든 언어에서 표준처럼 사용하고 있다.
- 표현 방법
{ "이름" : "값", "이름" : "값" }

ex) 01.json

```
{  
  "message": "Hello JSON Object",  
  "age" : 20  
}
```

* JSON을 쓰는 이유

- Java Beans와 일치시켜서 데이터를 처리할 수 있기 때문이다.

(2) XML과 JSON의 비교

20150001, 홍길동, 010-1234-5678

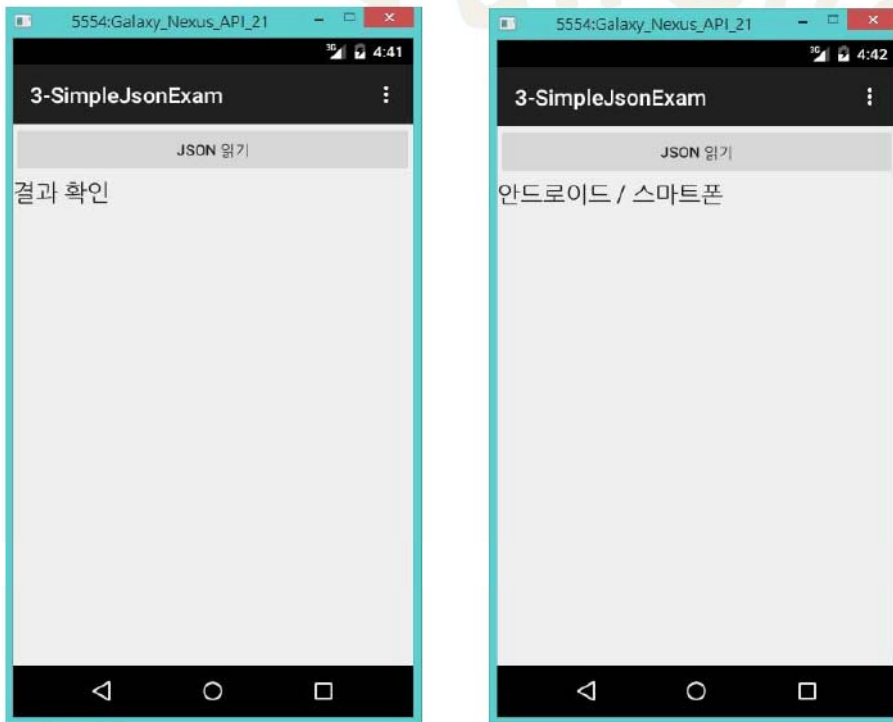
20150002, 고길동, 010-2345-6789

XML	<pre><?xml version="1.0" encoding="utf-8"?> <students> <student st_id="20150001"> <name>홍길동</name> <tel>010-1234-5678</tel> </student> <student st_id="20150002"> <name>고길동</name> <tel>010-2345-6789</tel> </student> </students></pre>
JSON	<pre>{ "students" : [{"st_id" : "20150001", "name" : "홍길동", "tel" : "010-1234-5678" }, {"st_id" : "20150002", "name" : "고길동", "tel" : "010-2345-6789" }] }</pre>

예제3. JSON 데이터의 활용

Application Name : 3-SimpleJsonExam
Company Domain : .example.com
Package Name : com.example.simplejsonexam
Project location : D:\Android\workspace\ch09\3-SimpleJsonExam

<실행 결과>



<작업 순서>

- MAMP를 실행하여, 아파치 서버를 동작시킨다.
- 서버에 저장할 파일을 만들고, 저장한다.
- "android-async-http-1.4.4.jar"를 라이브러리에 추가한다.
- activity_main.xml 파일 기본 설정
 - => Design에서 Button을 1개 추가한다.
 - => Design에서 Large Text를 1개 추가한다.
 - => 실행 결과처럼 설정을 수정한다.
- MainActivity.java에서 이벤트 처리를 한다.
- AndroidManifest.xml에 퍼미션을 추가한다.
 - => `<uses-permission android:name="android.permission.INTERNET"/>`

⑩ 리소스 파일

1) D:/android/server_data/simple.json ("utf-8"로 저장)

```
{  
  "device": {  
    "name": "안드로이드",  
    "type": "스마트폰"  
  }  
}
```

① app/res/layout/activity_main.xml

② app/java/com.example.simplejsonexam/MainActivity.java

③ app/manifests/AndroidManifest.xml

4. JSON 데이터를 활용한 리스트 뷰의 사용

예제4. JSON 데이터를 활용한 리스트 뷰의 사용

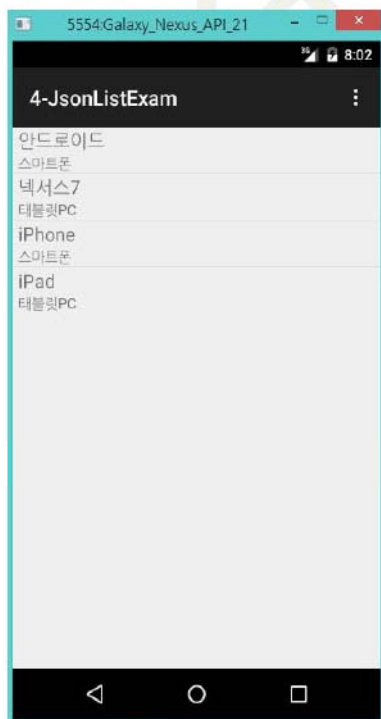
Application Name : 4-JsonListExam

Company Domain : .example.com

Package Name : com.example.jsonlistexam

Project location : D:\Android\workspace\ch09\4-JsonListExam

<실행 결과>



<작업 순서>

- MAMP를 실행하여, 아파치 서버를 동작시킨다.
- 서버에 저장할 파일을 만들고, 저장한다.
- "android-async-http-1.4.4.jar"를 라이브러리에 추가한다.
- activity_main.xml 파일 기본 설정
 - > Design에서 ListView를 1개 추가한다.
 - => 실행 결과처럼 설정을 수정한다.
- ListView에 데이터를 출력하기
 - => 아래의 <ListView 데이터 출력 준비> 참조
- MainActivity.java에서 이벤트 처리를 한다.
- AndroidManifest.xml에 퍼미션을 추가한다.
 - => <uses-permission android:name="android.permission.INTERNET"/>

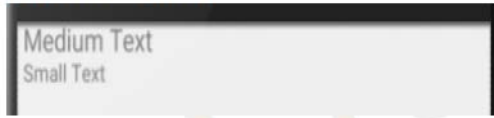
<ListView 데이터 출력 준비>

1) JavaBeans 구성

- > model 패키지 안에 작성
- > Device : name, type

2) /res/layout/list_item.xml이 필요

- > TextView 두 개
- name용, type용



3) ArrayAdapter를 상속받는 클래스 필요

- > adapter 패키지 안에 작성
- > 클래스 이름은 JavaBeans이름+Adapter
- => DeviceAdapter
- > getView() 메서드 내용 작성

4) 통신 처리를 위한 ~~~Response 클래스 준비

- > response 패키지 안에 작성
 - > 클래스 이름은 JavaBeans이름+Response
 - => DeviceResponse
 - > AsyncHttpResponseHandler를 상속
 - > json 데이터를 추출하여 List 클래스에 저장시킨다.
 - > 다음의 멤버변수 추가
- ```
Activity activity;
DeviceAdapter adapter;
ProgressDialog dialog;
```
- > 생성자를 통해서 activity와 adapter를 할당
- ```
public DeviceResponse(Activity activity, DeviceAdapter adapter) {
    this.activity = activity;
    this.adapter = adapter;
}
```
- > 기본 메서드 재정의
- ```
public void onStart() { }
public void onSuccess(String content) {}
public void onFailure(int statusCode, Throwable error, String content) {}
public void onFinish() { }
```
- > 통신 시작: 로딩창 띄우기
- ```
public void onStart() { }
```
- > 통신 종료: 로딩창 닫기
- ```
public void onFinish() { }
```
- > 통신 성공: json 데이터 처리
- ```
public void onSuccess(...) { }
```
- > 통신 실패: 에러메시지 표시
- ```
public void onFailure(...) { }
```

## ⑩ 리소스 파일

1) D:/android/server\_data/list.json ("utf-8"로 저장)

```
{
 "device": [{
 "name": "안드로이드",
 "type": "스마트폰"
 }, {
 "name": "넥서스7",
 "type": "태블릿PC"
 }, {
 "name": "iPhone",
 "type": "스마트폰"
 }, {
 "name": "iPad",
 "type": "태블릿PC"
 }]
}
```

① app/res/layout/activity\_main.xml

② app/res/layout/list\_item.xml

③ app/java/com.example.jsonlistexam/model/Device.java

④ app/java/com.example.jsonlistexam/adapter/DeviceAdapter.java

⑤ app/java/com.example.jsonlistexam/response/DeviceResponse.java

⑥ app/java/com.example.jsonlistexam/MainActivity.java

⑦ app/manifests/AndroidManifest.xml

## 문제1. JSON 데이터를 활용한 리스트 뷰의 사용 2

Application Name : GradeListExam

Company Domain : .example.com

Package Name : com.example.gradelistexam

Project location : D:\Android\workspace\ch09\GradeListExam

### <실행 결과>



### <작업 순서>

- MAMP를 실행하여, 아파치 서버를 동작시킨다.
- 서버에 저장할 파일을 만들고, 저장한다.
- "android-async-http-1.4.4.jar"를 라이브러리에 추가한다.
- activity\_main.xml 파일 기본 설정
  - => Design에서 ListView를 1개 추가한다.
  - => 실행 결과처럼 설정을 수정한다.
- ListView에 데이터를 출력하기
  - => <ListView 데이터 출력 준비> 참조
- MainActivity.java에서 이벤트 처리를 한다.
- AndroidManifest.xml에 퍼미션을 추가한다.
  - => <uses-permission android:name="android.permission.INTERNET"/>



## ⑩ 리소스 파일

1) D:/android/server\_data/grade.json ("utf-8"로 저장)

```
{
 "grade": {
 "member": [
 {"name": "홍길동", "kor": 80, "eng": 70, "math": 85},
 {"name": "고길동", "kor": 90, "eng": 85, "math": 94},
 {"name": "김길동", "kor": 100, "eng": 72, "math": 71}
]
 }
}
```

① app/res/layout/activity\_main.xml

② app/res/layout/list\_item.xml

③ app/java/com.example.gradelistexam/model/Grade.java

④ app/java/com.example.gradelistexam/adapter/GradeAdapter.java

⑤ app/java/com.example.gradelistexam/response/GradeResponse.java

⑥ app/java/com.example.gradelistexam/MainActivity.java

⑦ app/manifests/AndroidManifest.xml

## 10장. Android Network 응용

### 1. 원격지 이미지 주소를 포함한 JSON 데이터의 처리

#### \* Android Universal Image Loader 사용하기

- <https://github.com/nostra13/Android-Universal-Image-Loader>
- Android Universal Image Loader(AUIL)는 많은 앱에 적용되어 있고, 화면 크기를 기준으로 캐시 용량을 제한하는 등 다양한 캐시 정책을 지원한다.  
Executor, 스레드 풀 크기, Bitmap Options 등 변경할 수 있는 옵션이 많다.

#### \* Android Universal Image Loader 특징

- 다중 스레드 이미지 로딩
- 이미지 디코드
- 메모리 캐시
- 디스크 캐시
- 이미지 디스플레이 옵션 설정 등 분리된 옵션이 있는 모든 디스플레이 이미지 호출을 사용자 정의 할 수 있는 가능성
- 위젯 지원

## 예제1. 원격지 이미지 주소를 포함한 JSON 데이터의 처리

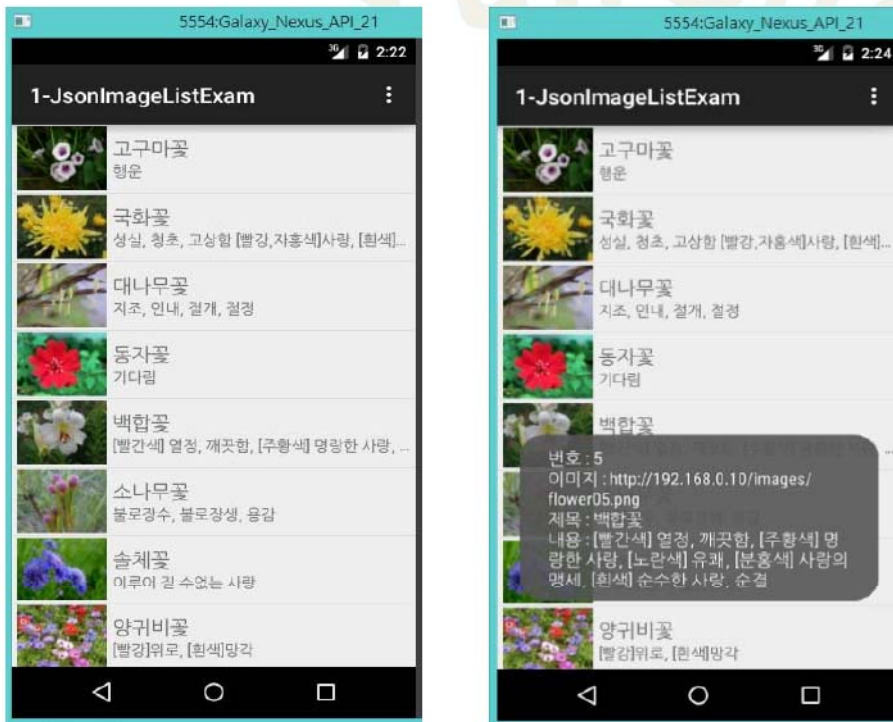
Application Name : 1-JsonImageListExam

Company Domain : .example.com

Package Name : com.example.jsonimagelistexam

Project location : D:\Android\workspace\ch10\1-JsonImageListExam

### <실행 결과>



### <작업 순서>

- MAMP를 실행하여, 아파치 서버를 동작시킨다.
- 서버에 저장할 파일을 만들고, 저장한다.
- "android-async-http-1.4.4.jar"를 라이브러리에 추가한다.
- "universal-image-loader-1.9.1.jar"를 라이브러리에 추가한다.
- activity\_main.xml 파일 기본 설정
  - => Design에서 ListView를 1개 추가한다.
  - => 실행 결과처럼 설정을 수정한다.
- ListView에 데이터를 출력하기
  - => <ListView 데이터 출력 준비> 참조
  - => java beans 만들기 : model/Column
  - => adapter 만들기 : adapter/ColumnAdapter
  - => response 만들기 : response/ColumnResponse
- MainActivity.java에서 이벤트 처리를 한다.
- AndroidManifest.xml에 퍼미션을 추가한다.
  - => <uses-permission android:name="android.permission.INTERNET"/>

## ⑩ 리소스 파일

- 1) app/libs/android-async-http-1.4.4.jar
- 2) app/libs/universal-image-loader-1.9.1.jar
- 3) app/res/drawable/  
ic\_empty.png, ic\_error.png, ic\_stub.png
- 4) D:/android/server\_data/images/  
flower01.png ~ flower13.png
- 5) D:/android/server\_data/column.json ("utf-8"로 저장)

```
{
 "column" : {
 "item": [
 {
 "num": 1,
 "img": "http://192.168.0.10/images/flower01.png",
 "subject": "고구마꽃",
 "content": "행운"
 },
 {
 "num": 2,
 "img": "http://192.168.0.10/images/flower02.png",
 "subject": "국화꽃",
 "content": "성실, 청초, 고상함 [빨강,자홍색]사랑, [흰색]순결 [황색]질투"
 },
 {
 "num": 3,
 "img": "http://192.168.0.10/images/flower03.png",
 "subject": "대나무꽃",
 "content": "지조, 인내, 절개, 절정"
 },
 {
 "num": 4,
 "img": "http://192.168.0.10/images/flower04.png",
 "subject": "동자꽃",
 "content": "기다림"
 },
 {
 "num": 5,
 "img": "http://192.168.0.10/images/flower05.png",
 "subject": "백합꽃",
 "content": "[빨간색] 열정, 깨끗함, [주황색] 명량한 사랑, [노란색] 유쾌, [분홍색] 사랑의 맹세, [흰색] 순수한 사랑, 순결"
 },
 {
 "num": 6,
 "img": "http://192.168.0.10/images/flower06.png",
 "subject": "소나무꽃",
 "content": "불로장수, 불로장생, 용감"
 },
 {
 "num": 7,
 "img": "http://192.168.0.10/images/flower07.png",
 "subject": "솔채꽃",
 "content": "이루어 질 수없는 사랑"
 },
 {
 "num": 8,
```



```

 "img": "http://192.168.0.10/images/flower08.png",
 "subject": "양귀비꽃",
 "content": "[빨강]위로, [흰색]망각"
 },
 {
 "num": 9,
 "img": "http://192.168.0.10/images/flower09.png",
 "subject": "은방울꽃",
 "content": "섬세함"
 },
 {
 "num": 10,
 "img": "http://192.168.0.10/images/flower10.png",
 "subject": "장미",
 "content": "[노랑색]질투, 완벽한 성취, [흰색] 순결, 순진, 매력, [빨강색] 욕망, 열정, 기쁨, [파랑색] 기적, [핑크색] 맹세, 행복한 사랑"
 },
 {
 "num": 11,
 "img": "http://192.168.0.10/images/flower11.png",
 "subject": "질레꽃",
 "content": "고독, 신중한 사랑, 가족에 대한 그리움"
 },
 {
 "num": 12,
 "img": "http://192.168.0.10/images/flower12.png",
 "subject": "투구꽃",
 "content": "밤의 열림"
 },
 {
 "num": 13,
 "img": "http://192.168.0.10/images/flower13.png",
 "subject": "해바라기",
 "content": "애모, 아름다운 빛, 그리움, 기다림, 숭배"
 }
]
}

```

- ① app/res/layout/activity\_main.xml
- ② app/res/layout/list\_item.xml
- ③ app/java/com.example.jsonimagelistexam/model/Column.java
- ④ app/java/com.example.jsonimagelistexam/adapter/ColumnAdapter.java
- ⑤ app/java/com.example.jsonimagelistexam/response/ColumnResponse.java
- ⑥ app/java/com.example.jsonimagelistexam/MainActivity.java
- ⑦ app/manifests/AndroidManifest.xml



## 문제1. 원격지 이미지 주소를 포함한 JSON 데이터의 처리

Application Name : JsonImageListTest

Company Domain : .example.com

Package Name : com.example.jsonimagelisttest

Project location : D:\Android\workspace\ch10\JsonImageListTest

### <실행 결과>



### <작업 순서>

- MAMP를 실행하여, 아파치 서버를 동작시킨다.
- 서버에 저장할 파일을 만들고, 저장한다.
- "android-async-http-1.4.4.jar"를 라이브러리에 추가한다.
- "universal-image-loader-1.9.1.jar"를 라이브러리에 추가한다.
- activity\_main.xml 파일 기본 설정
  - => Design에서 ListView를 1개 추가한다.
  - => 실행 결과처럼 설정을 수정한다.
- ListView에 데이터를 출력하기
  - => <ListView 데이터 출력 준비> 참조
  - => java beans 만들기 : model/Job
  - => adapter 만들기 : adapter/JobAdapter
  - => response 만들기 : response/JobResponse
- MainActivity.java에서 이벤트 처리를 한다.
- AndroidManifest.xml에 퍼미션을 추가한다.
  - => <uses-permission android:name="android.permission.INTERNET"/>

## ⑩ 리소스 파일

1) app/libs/android-async-http-1.4.4.jar

2) app/libs/universal-image-loader-1.9.1.jar

3) app/res/drawable/

ic\_empty.png, ic\_error.png, ic\_stub.png

4) D:/android/server\_data/images/

job01.png ~ job14.png

5) D:/android/server\_data/job.json ("utf-8"로 저장)

```
{
 "job" : {
 "item": [
 {
 "num": 1,
 "img": "http://192.168.0.10/images/job01.png",
 "subject": "가수",
 "content": "노래 부르는 것이 직업인 사람"
 },
 {
 "num": 2,
 "img": "http://192.168.0.10/images/job02.png",
 "subject": "경찰공무원",
 "content": "경찰 업무에 종사하는 공무원. 치안총감, 치안정감, 치안감, 경무관, 총경, 경정, 경감, 경위, 경사, 경장, 순경의 계급이 있다."
 },
 {
 "num": 3,
 "img": "http://192.168.0.10/images/job03.png",
 "subject": "과학자",
 "content": "과학을 전문으로 연구하는 사람. 주로 자연 과학을 연구하는 사람을 이른다."
 },
 {
 "num": 4,
 "img": "http://192.168.0.10/images/job04.png",
 "subject": "군인",
 "content": "군대에서 복무하는 사람. 육해공군의 장교, 부사관, 병사를 통틀어 이르는 말이다."
 },
 {
 "num": 5,
 "img": "http://192.168.0.10/images/job05.png",
 "subject": "미용사",
 "content": "일정한 자격을 가지고 사람의 머리카락 피부 따위를 아름답게 매만지는 일을 직업으로 하는 사람"
 },
 {
 "num": 6,
 "img": "http://192.168.0.10/images/job06.png",
 "subject": "비행기 조종사",
 "content": "항공기를 일정한 방향과 속도로 움직이도록 다루는 기능과 자격을 갖춘 사람"
 },
 {
 "num": 7,
 "img": "http://192.168.0.10/images/job07.png",
 "subject": "선생님",
 "content": "학생을 가르치는 사람"
 }
]
 }
}
```

```

{
 "num": 8,
 "img": "http://192.168.0.10/images/job08.png",
 "subject": "소방공무원",
 "content": "화재를 예방·경계·진압하는 데 종사하는 공무원. 국민의 생명, 신체, 재산을 보호하는 것을 직무로 하며 국가 소방 공무원과 지방 소방 공무원이 있다."
},
{
 "num": 9,
 "img": "http://192.168.0.10/images/job09.png",
 "subject": "스튜어디스",
 "content": "여객기나 여객선 따위에서 승객을 돌보는 여자 승무원"
},
{
 "num": 10,
 "img": "http://192.168.0.10/images/job10.png",
 "subject": "운동선수",
 "content": "운동 경기에 뛰어난 재주가 있거나 전문적으로 운동을 하는 사람"
},
{
 "num": 11,
 "img": "http://192.168.0.10/images/job11.png",
 "subject": "요리사",
 "content": "요리를 전문으로 하는 사람"
},
{
 "num": 12,
 "img": "http://192.168.0.10/images/job12.png",
 "subject": "의사",
 "content": "의술과 약으로 병을 치료·진찰하는 것을 직업으로 삼는 사람. 국가시험에 합격하여 보건 복지 가족부 장관의 면허를 취득하여야 한다."
},
{
 "num": 13,
 "img": "http://192.168.0.10/images/job13.png",
 "subject": "화가",
 "content": "그림 그리는 것을 직업으로 하는 사람"
},
{
 "num": 14,
 "img": "http://192.168.0.10/images/job14.png",
 "subject": "회사원",
 "content": "회사에서 근무하는 사람"
}
}
}

```

- ① app/java/res/layout/activity\_main.xml
- ② app/res/layout/list\_item.xml
- ③ app/java/com.example.jsonimagelisttest/model/Job.java
- ④ app/java/com.example.jsonimagelisttest/adapter/JobAdapter.java
- ⑤ app/java/com.example.jsonimagelisttest/response/JobResponse.java
- ⑥ app/java/com.example.jsonimagelisttest/MainActivity.java
- ⑦ app/manifests/AndroidManifest.xml

## 2. 카카오 OpenAPI를 활용한 이미지 검색

### \* Daum 개발자 네트워크

- <https://developers.kakao.com/>

### \* 준비 작업

- 카카오 개발자 네트워크 사이트를 방문한다.
- 로그인 한다.
- “앱만들기”를 한다.
  - => 이름을 입력하고 “앱만들기” 버튼 클릭하기
- “계속 진행” 버튼 클릭
- api key가 발급된다.
- API Key를 복사해서, 메모장에 저장해 놓는다.
- “개발가이드” 클릭
  - => 검색 -> “REST API 개발가이드” 클릭
- “이미지 검색” 클릭
- 이미지 검색 항목 확인
- “REST API 도구”에서 결과 미리보기
- 검색결과 보기
- 요청변수들을 바꿔가면서 내용들을 확인해본다.
- 검색내용 중에서 사용할 항목을 정한다.
  - => total\_count, pageable\_count, is\_end
  - => collection, thumbnail\_url, image\_url, width, height, display\_sitename, doc\_url, datetime



## 예제2. 카카오 OpenAPI를 활용한 이미지 검색

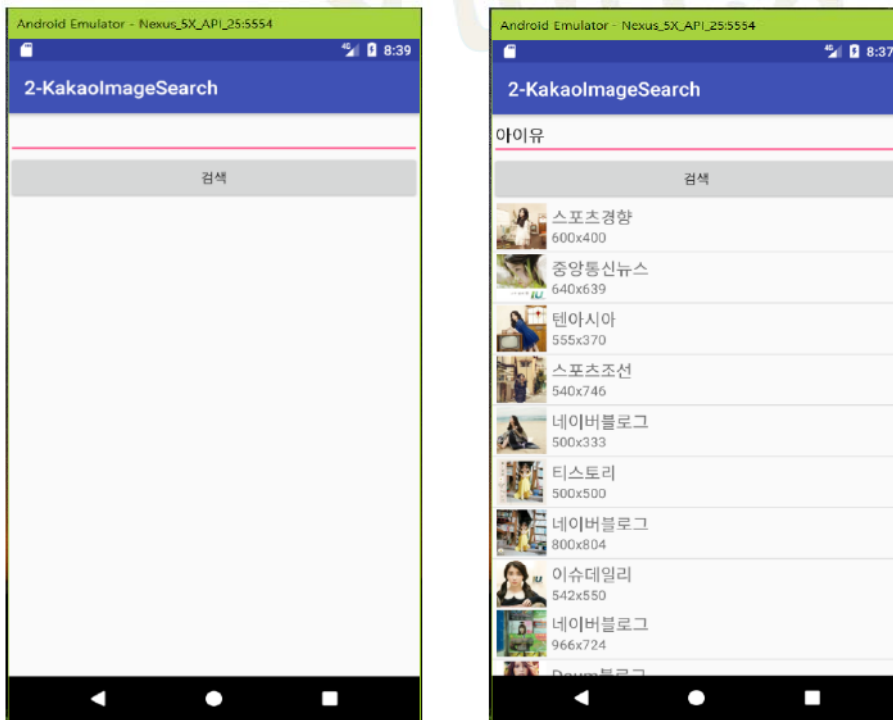
Application Name : 2-KakaoImageSearch

Company Domain : .example.com

Package Name : com.example.kakaoimagesearch

Project location : D:\Android\workspace\ch10\2-KakaoImageSearch

### <실행 결과>



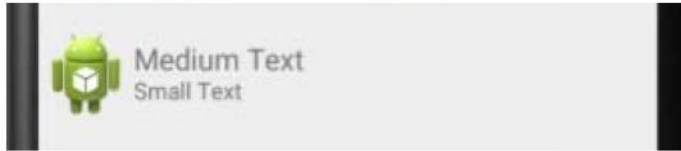
### <작업 순서>

- "android-async-http-1.4.4.jar"를 라이브러리에 추가한다.
  - "universal-image-loader-1.9.1.jar"를 라이브러리에 추가한다.
  - activity\_main.xml 파일 기본 설정
    - => Design에서 EditText를 1개 추가한다.
    - => Design에서 Button을 1개 추가한다.
    - => Design에서 ListView를 1개 추가한다.
    - => 실행 결과처럼 설정을 수정한다.
  - ListView에 데이터를 출력하기
    - => <ListView 데이터 출력 준비> 참조
- 
- 1) "다음"에서 검색된 내용 중에서 사용할 항목을 선택한다.
    - => total\_count, pageable\_count, is\_end
    - => collection, thumbnail\_url, image\_url, width, height, display\_sitename, doc\_url, datetime
  - 2) 저장할 JavaBeans를 만든다.
    - => class name : Image
    - => total\_count, pageable\_count, is\_end
    - => collection, thumbnail\_url, image\_url, width, height, display\_sitename, doc\_url, datetime

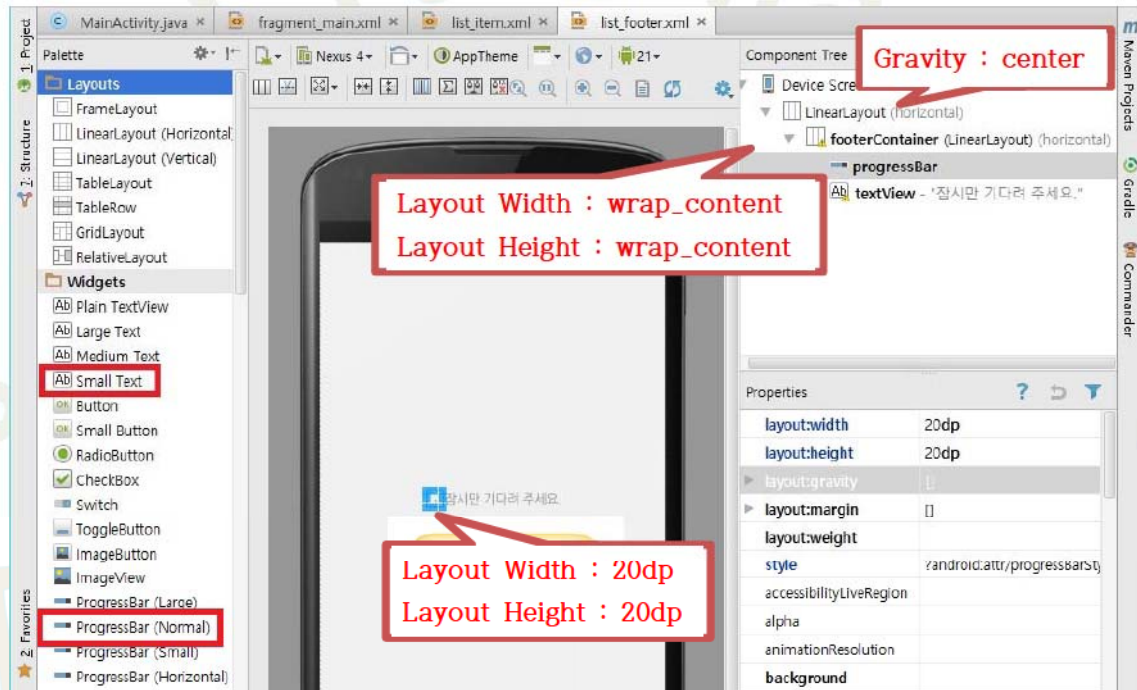


3) ListView에 출력할 화면을 layout XML 파일로 구성한다.

=> item XML name : list\_item.xml



=> footer XML name : list\_footer.xml



4) ListView에 사용할 adapter 클래스를 만든다.

=> adapter name : ImageAdapter

5) 통신용 response 클래스를 만든다.

=> response name : ImageResponse

- MainActivity.java에서 이벤트 처리를 한다.

- AndroidManifest.xml에 퍼미션을 추가한다.

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

## ⑥ 리소스 파일

1) app/libs/android-async-http-1.4.4.jar

2) app/libs/universal-image-loader-1.9.1.jar

3) app/res/drawable/

-> ic\_empty.png, ic\_error.png, ic\_stub.png

① app/res/layout/activity\_main.xml

② app/res/layout/list\_item.xml

③ app/res/layout/list\_footer.xml

④ app/java/com.example.kakaoimagesearch/model/Image.java

⑤ app/java/com.example.kakaoimagesearch/adapters/ImageAdapter.java

⑥ app/java/com.example.kakaoimagesearch/response/ImageResponse.java

⑦ app/java/com.example.kakaoimagesearch/MainActivity.java

⑧ app/manifests/AndroidManifest.xml