



URL 통신 실습

배 희호 교수
경북대학교
소프트웨어융합과



URL 통신



- Android에서 URL 통신은 Network를 통해 Web Server와 Data를 주고받기 위해 사용
- 이는 보통 HTTP(S) Protocol을 사용하여 Web API와 통신하거나 Web Resource를 가져오는 작업을 포함
- 이 과정은 RESTful API나 GraphQL API와 상호 작용할 때 주로 사용



URL 통신



■ 기본 구성요소

■ URL(Uniform Resource Locator)

- Resource의 위치를 지정

- 예) `https://api.example.com/data`

■ HTTP 메소드

- GET : Server로부터 Data 가져오기

- POST : Server에 Data 보내기

- PUT : Data를 생성하거나 Update하기

- DELETE : Data를 삭제하기

■ Headers

- Request/Response과 관련된 Metadata를 포함

- 예) Content-Type, Authorization.

■ Payload (실질적인 내용)

- Request 본문에 JSON, XML 등의 Data를 담아 보낼 수 있음



URL 통신



- Android에서 URL 통신 구현 방식
 - HttpURLConnection (Java 기반 기본 클래스)
 - Android 기본 API에서 제공하는 HTTP 연결 클래스
 - 가볍고 직접적인 HTTP 통신을 구현할 수 있지만, Code가 길어 질 수 있음
 - OkHttp (권장 Library)
 - Square에서 개발한 Library로, Android에서 URL 통신을 간편하게 수행하도록 설계됨
 - 비동기 요청 및 간단한 API 제공
 - Retrofit (고급 Library)
 - OkHttp 기반으로 동작하며 RESTful API 호출을 간편화
 - Code 간결성 및 JSON 직렬화/역직렬화에 유리



URI 클래스

- URL을 일반화(Generalization) 시킨 클래스
- Internet상의 Resource를 경로로 지정하여 필요한 정보를 추출할 수 있게 해주는 클래스
- URI 클래스와 URL 클래스 차이

URI	<ul style="list-style-type: none">✓ URI 클래스는 순수하게 Resource를 식별하고 URI를 분석하는 기능만 제공하고, URI가 참조하는 Resource를 가져오는 메소드는 제공하지 않음✓ URI 클래스는 URL 클래스보다 관련된 Spec 및 표준 사항을 더 잘 준수함✓ URI 객체는 상대적인 URI를 표현할 수 있음✓ URI 객체는 순수하게 문자열 분석과 조작을 위한 객체
URL	<ul style="list-style-type: none">✓ URL 클래스는 모든 URI를 저장하기 전에 URL 형태로 변경✓ URL 객체는 Network 전송을 위한 Application Layer Protocol을 표현하기 위한 객체



URL 클래스



- URL 클래스는 Java의 java.net 패키지에서 제공되며, Network를 통해 Data를 주고받기 위한 간단한 방법을 제공
- 이 클래스는 URL(Uniform Resource Locator)을 표현하며, URL을 통해 Resource에 접근하고 Data를 송수신할 수 있음
- URL을 추상화하여 만든 클래스로 원격 Server 자원의 결과를 가져옴
 - URL 클래스만 사용하면 원하는 Data를 쉽게 얻을 수 있음
- URL 클래스는 JAVA Program이 특정 위치로부터 Data를 가져오는 가장 쉬운 방법을 제공
- 이 클래스는 내부의 Protocol이나 Server와 통신하는 방법에 대한 자세한 부분은 신경 쓰지 않아도 됨



URL 클래스

- URL이란 Internet에서 접근 가능한 자원의 주소를 표현하는 형식으로 URL을 이용하여 Internet 상의 다양한 Protocol과 자원을 표기할 수 있음
- 웹(World Wide Area)상에 존재하는 자원(Source)에 접근할 때 사용
 - Internet에서 자원의 유일한 주소를 나타내기 위한 기능을 제공해 주는 클래스
- JAVA나 Android내의 URL을 처리하고 싶을 때 사용
- URL 클래스는 Web 상의 주소를 나타내는 클래스로써 InetAddress 클래스보다 확장된 개념으로 Network 연결까지 가능. 따라서 URL 객체 생성시 잘못된 URL 형태를 주었을 때 예외처리를 반드시 해주어야 함



URL 클래스



- 주요 특징

- 간단한 URL 처리

- URL 클래스를 통해 쉽게 URL을 Parsing하고 구성할 수 있음

- HTTP Request/Response 처리

- GET, POST 등 다양한 HTTP Request 방식 지원

- 직접 통신

- 간단한 Network 작업에 적합하며, Library를 추가로 설치하지 않아도 됨



URL 클래스



■ 생성자

- URL을 추상화 하여 만든 클래스
- final 클래스로 되어 있기 때문에 상속하여 사용할 수 없음
- 모든 생성자는 MalformedURLException 예외를 발생하기 때문에 반드시 예외 처리를 해야 함



URL 클래스



■ 생성자

생성자	설명
URL(String spec)	<ul style="list-style-type: none">✓ spec은 URL로 해석할 수 있는 문자열이어야 함✓ 매개변수 spec으로 자원에 대한 URL객체를 생성
URL(String protocol, String host, int port, String file)	<ul style="list-style-type: none">✓ protocol과 host, port와 file이름이 지정하는 자원에 대한 URL객체를 생성✓ File은 path와 query를 의미함
URL(String protocol, String host, String file)	protocol과 host, file로 해당 자원에 대한 URL객체를 생성
URL(URL context, String spec)	URL 객체 context에 대한 상대 경로가 지정하는 자원에 대한 URL 객체 생성



URL 클래스

■ 절대 경로로 URL 객체 생성

```
URL homePage = new URL("http://news.hankooki.com");
```



■ 상대 경로로 URL 객체 생성

```
URL opinion = new URL(homePage, "opinion/editorial.htm");
```

opinion이라는 URL은 결국 다음 주소 의미
"http://news.hankooki.com/opinion/editorial.htm"

■ 잘못된 주소의 URL을 입력하면 MalformedURLException 예외 발생



URL 클래스



■ 메소드

메소드	설명
Object getContent()	URL의 Content를 반환
String getAuthority()	URL의 Host명과 Port를 결합한 문자열을 반환
int getPort()	<ul style="list-style-type: none">✓ URL에 명시된 Port를 반환✓ 만약에 없으면 -1을 반환✓ 일반적으로 HTTP Port는 명시하지 않더라도 80번 Port를 인식
int getLocalPort()	Socket이 연결된 Local Port 번호 반환
String getDefaultPort()	<ul style="list-style-type: none">✓ URL에 상관없이 Protocol의 default Port 번호를 반환✓ http(80), ftp(21)
String getFile()	<ul style="list-style-type: none">✓ URL 주소의 File 이름 반환✓ URL의 path와 query를 결합한 문자열을 반환



URL 클래스

■ 메소드

메소드	설명
String getFile()	URL 주소의 File 이름 반환 URL의 path와 query를 결합한 문자열을 반환
String getHost()	URL의 host를 문자열로 반환
String getPath()	URL 주소의 경로 부분 반환 URL의 query를 문자열로 반환
String getProtocol()	URL의 protocol를 문자열로 반환
String getQuery()	URL의 query를 문자열로 반환
String getRef()	URL의 reference를 문자열로 반환
URLConnection openConnection()	URL 주소의 원격 객체에 접속한 뒤 통신할 수 있는 URLConnection 객체를 생성 URLConnection 클래스는 추상 클래스이기 때문에 객체를 생성할 수 없고, URL 클래스의 openConnection() 메소드를 이용해서 객체를 생성 할 수 있음



URL 클래스

■ 메소드

메소드	설명
InputStream openStream()	InputStream의 객체를 생성 이 메소드로 해당 URL의 자원(resource)를 가져올 수 있음 URL에 대해 연결을 설정하고 이 연결로 부터 입력을 받을 수 있는 InputStream 객체 반환
String toExternalForm()	URL을 문자열로 반환



URL 클래스

■ 메소드

메소드	설 명
getOutputStream()	새로운 OutputStream에 Request 할 OutputStream을 받아옴 <code>OutputStream os = http.getOutputStream();</code>
int getResponseCode()	Web Server에 Request 과정 수행 <code>int resCode = http.getResponseCode();</code> ✓ 요청이 무사히 전달되었으면 HTTP_OK(200), ✓ URL이 발견되지 않으면 HTTP_NOT_FOUND(404), ✓ 인증에 실패하면 HTTP_UNAUTHORIZED(401) 에러 코드가 반환



URL 클래스



■ URL 클래스 사용 예

```
URL url = new URL("http://codedragon.tistory.com");
```

메소드 호출	반환 값
String getProtocol()	http
String getHost()	codedragon.tistory.com
int getPort()	80
String getFile()	index.html
String getRef()	null



URL 클래스

■ URL 객체

```
URL url = new URL("http://www.chat.com/chatroom?no=3");
```

■ 위 url 객체로 부터 호출할 수 있는 주요 내용

- String getProtocol() : http

- int getDefaultPort() : 80

- int getPort() : -1

(URL 자체에 명시가 안되어 있으면 -1)

- String getHost() : www.chat.com

- String getFile() : /chatroom?no=3

- String getPath() : /chatroom

- String getQuery() : no=3



URL 클래스

■ 메소드

메소드	설명
setConnectionTimeout(int)	연결 제한 시간을 1/1000초 단위로 지정 0이면 무한 대기
setRequestMethod()	Request 방식 지정(기본값은 GET)
setDoOutput(true)	OutputStream으로 POST Data를 넘겨 주겠다는 Option을 정의
setDoInput(true)	InputStream으로 Server로 부터 Response Header와 Message를 읽어 들이겠다는 Option을 정의
setRequestProperty()	Request Header를 정의 setRequestProperty("Content-Type", "application/x-www-form-urlencoded");
String getContentEncoding()	Content 객체의 Encoding을 반환
int getContentLength()	content의 크기를 반환
int getContentType()	content의 type을 반환



URL 클래스



■ 메소드

메소드	설명
setReadTimeout(int timeout)	✓ 읽기 제한 시간을 지정 ✓ 0이면 무한 대기
setUseCaches(boolean newValue)	캐시 사용 여부를 지정
setDoInput(boolean newValue)	입력을 받을 것인지를 지정
setDoOutput(boolean newValue)	출력을 할 것인지를 지정
setRequestProperty(String field, String newValue)	Request Header에 값을 설정
addRequestProperty(String field, String newValue)	✓ Request Header에 값을 추가 ✓ 속성의 이름이 같더라도 덮어 쓰지는 않음



URL 클래스

■ 메소드

메소드	설명
String getHost()	URL의 host를 문자열로 반환
String getPath()	URL 주소의 경로 부분 반환 URL의 query를 문자열로 반환
String getProtocol()	URL의 protocol를 문자열로 반환
String getQuery()	URL의 query를 문자열로 반환
String getRef()	URL의 reference를 문자열로 반환
URLConnection openConnection()	✓ URL 주소의 원격 객체에 접속한 뒤 통신할 수 있는 URLConnection 객체를 생성 ✓ URLConnection 클래스는 추상 클래스이기 때문에 객체를 생성할 수 없고, URL 클래스의 openConnection() 메소드를 이용해서 객체를 생성



URL 클래스



■ 메소드

메소드	설명
InputStream openStream()	<ul style="list-style-type: none">✓ InputStream의 객체를 생성✓ 해당 URL의 Resource를 가져올 수 있음✓ URL에 대해 연결을 설정하고 이 연결로부터 입력을 받을 수 있는 InputStream 객체 반환
String toExternalForm()	URL을 문자열로 반환



URL 클래스



■ HTTP Request

- 기본 속성을 설정한 후 각 연결 Type 별로 속성을 추가로 설정
- Http 연결의 경우 Request Method를 지정해야 하고 다음 Method로 “GET”, “POST” 등의 방법을 지정하되, 별 지정이 없으면 Default인 “GET” 방식이 적용
- 모든 속성 설정이 완료되었으면 다음 메소드로 Server에 Request를 보내는데 이 메소드는 원격지의 Server로 명령을 보내고 Response 결과를 받음

```
void HttpURLConnection.setRequestMethod(  
    String method)
```

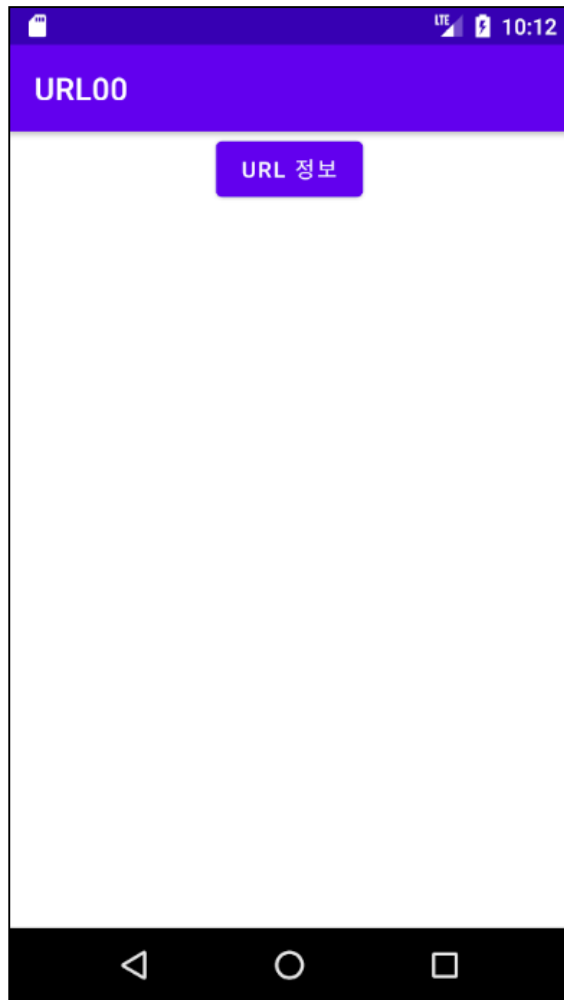


URL 클래스

- URLConnection 클래스는 Application과 URL간의 통신 Link를 위한 작업을 수행
- Network의 일반적인 작업들은 먼저 Connection 연결을 한 후, 그 Connection에 연결되어 있는 Stream을 생성
- URL 클래스는 URLConnection 클래스의 기능을 내부적으로 수행하고 있음



URL 통신 예제 0





URL 통신 예제 0



■ 사용자 인터페이스

```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout 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"
    android:orientation="vertical"
    tools:context=".MainActivity">
    <Button
        android:id="@+id/button"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_gravity="center"
        android:text="URL 정보" />
    <TextView
        android:id="@+id/textView"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:text="" />
</LinearLayout>
```



URL 통신 예제 0

■ MainActivity.JAVA

```
public class MainActivity extends AppCompatActivity {
```

```
    @Override
```

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

```
        super.onCreate(savedInstanceState);
```

```
        setContentView(R.layout.activity_main);
```

```
        TextView textView = findViewById(R.id.textView);
```



URL 통신 예제 0

■ MainActivity.JAVA

```
Button button = findViewById(R.id.button);
button.setOnClickListener(new View.OnClickListener() {
    @Override
    public void onClick(View v) {
        URL opinion = null;
        try {
            URL homepage = new URL("http://news.hankooki.com");
            opinion = new URL(homepage, "opinion/deitorial.htm");
            opinion = new URL("http", "news.hankooki.com",
                             "opinion/deitorial.htm");
        } catch (MalformedURLException e) {
            Toast.makeText(getApplicationContext(), "잘못된 URL입니다.",
                             Toast.LENGTH_SHORT).show();
        }
    }
});
```



URL 통신 예제 0

■ MainActivity.JAVA

```
textView.append("Protocol = " + home.getProtocol()); // 프로토콜 출력
textView.append("\nhost = " + home.getHost()); // 호스트 이름 출력
textView.append("\ndefault port = " + home.getDefaultPort());
textView.append("\nport = " + home.getPort()); // 포트 번호 출력
textView.append("\nfilename = " + home.getFile());
textView.append("\npath = " + home.getPath());
textView.append("\nquery = " + home.getQuery());
    }
});
}
```



URL 통신 예제 0

- URL 클래스는 http, https, file, ftp 그리고 jar Protocol들만 지원
- URL 객체를 생성하는 2가지 방법
 - 절대 경로를 이용하여 URL 생성하기

```
try {  
    homepage = new URL  
        ("http://news.hankooki.com/opinion/deitorial.htm");  
} catch (MalformedURLException e) {  
}
```

- 상대 경로를 이용하여 URL 생성하기

```
try {  
    homepage = new URL("http://news.hankooki.com");  
    opinion = new URL(homepage, "opinion/deitorial.htm");  
} catch (MalformedURLException e) {  
}
```



URL 통신 예제 0



- URL의 구성 요소를 이용하여 URL 생성하기

```
try {  
    opinion = new URL("http", "news.hankooki.com",  
                    "opinion/deitorial.htm");  
} catch (MalformedURLException e) {  
}
```



URL 통신 예제 0

- 규칙에 맞지 않은 무효한 URL을 설정하면 `MalformedURLException` 예외가 발생
- 접속할 주소가 결정되면 `URL` 클래스의 다음 메소드로 접속
`URLConnection URL.openConnection([Proxy proxy])`
- 접속에 성공했으면 지정한 주소로 양방향 통신이 가능한 연결 객체가 반환



URL 통신 예제 0



■ 다음 메소드는 URL의 각 부분을 분리

메소드	반환값	설명
String getProtocol()	http	통신 방법을 정의하는 Protocol
int getDefaultPort()	80	Protocol이 정의하는 default Port
int getPort()	-1	URL에 정의된 Port 없을 경우 -1이 반환
String getHost()	www.chatting.com	Server 주소
String getFile()	/showroom?girl=3	Path와 Query
String getPath()	/showroom	Server내의 경로
String getQuery()	girl=3	Server로 전달되는 Query 변수값



URL 통신 예제 1



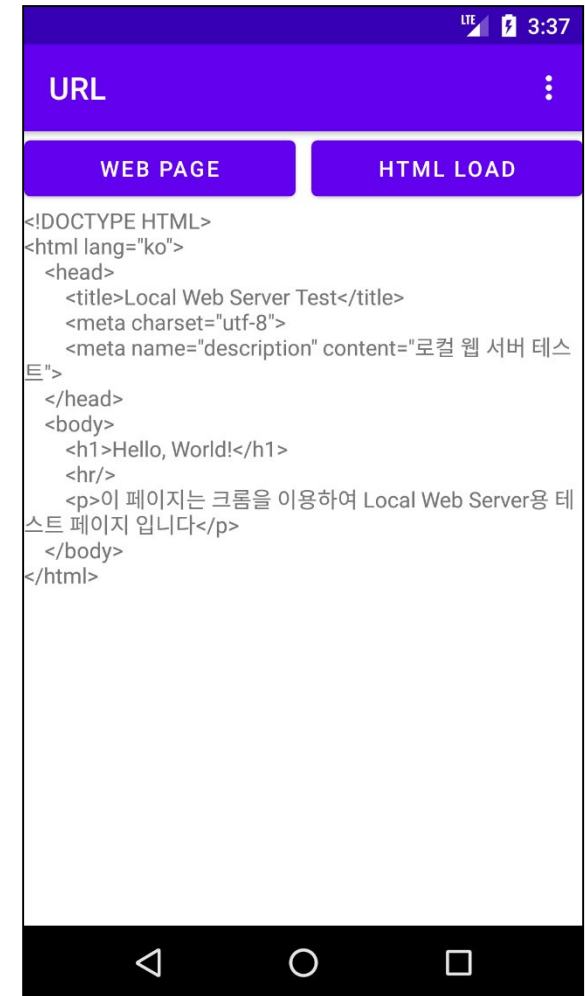
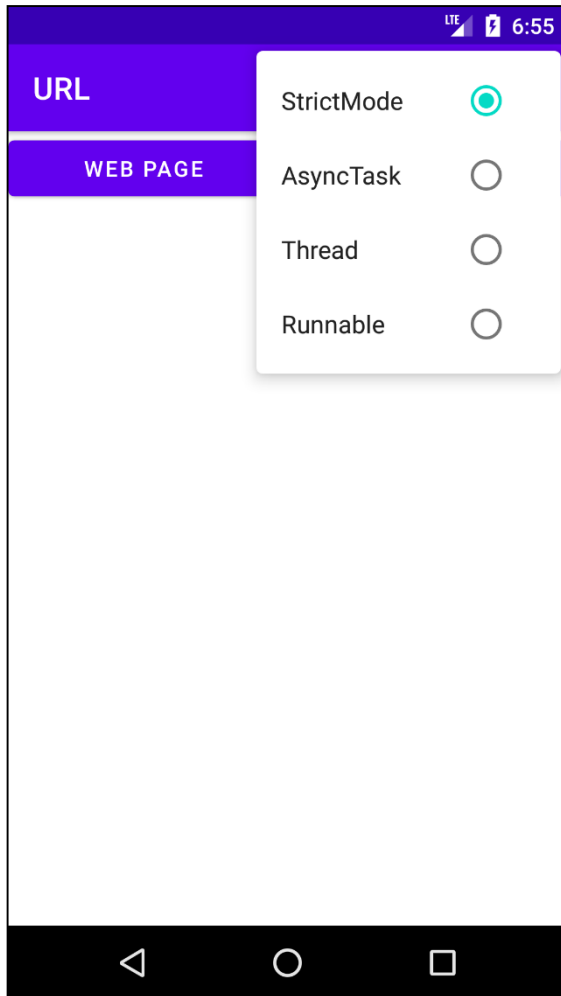
- Local Web Site index.html의 HTML 문서를 다운 받아보자

```
<!DOCTYPE HTML>
<html lang="ko">
  <head>
    <title>Local Web Server Test</title>
    <meta charset="utf-8">
    <meta name="description" content="로컬 웹 서버 테스트">
  </head>
  <body>
    <h1>Hello, World!</h1>
    <hr/>
    <p>이 페이지는 크롬을 이용하여 Local Web Server용 테스트
페이지 입니다</p>
  </body>
</html>
```



URL 통신 예제 1

■ 실행 결과





URL 통신 예제 1

■ Local Web Server 구동



URL 통신 예제 1

```
<?xml version="1.0" encoding="utf-8"?>
<menu xmlns:android="http://schemas.android.com/apk/res/android">
  <group
    android:checkableBehavior="single"
    android:enabled="true">
    <item
      android:id="@+id/item1"
      android:checked="true"
      android:title="StrictMode" />
    <item
      android:id="@+id/item2"
      android:title="AsyncTask" />
    <item
      android:id="@+id/item3"
      android:title="Thread" />
    <item
      android:id="@+id/item4"
      android:title="Runnable" />
  </group>
</menu>
```



URL 통신 예제 1

■ 사용자 인터페이스

```
<?xml version="1.0" encoding="utf-8" ?>
<LinearLayout 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"
    android:orientation="vertical"
    tools:context=".MainActivity">

    <LinearLayout
        android:layout_width="match_parent"
        android:layout_height="wrap_content">

        <Button
            android:id="@+id/button1"
            android:layout_width="0dp"
            android:layout_height="wrap_content"
            android:layout_gravity="center"
            android:layout_weight="1"
            android:text="Web Page" />
    </LinearLayout>
</LinearLayout>
```



URL 통신 예제 1



■ 사용자 인터페이스

```
<Button
    android:id="@+id/button2"
    android:layout_width="0dp"
    android:layout_height="wrap_content"
    android:layout_gravity="center"
    android:layout_marginStart="10dp"
    android:layout_weight="1"
    android:text="Html Load" />
```

```
</LinearLayout>
```

```
<FrameLayout
    android:layout_width="match_parent"
    android:layout_height="match_parent">
```

```
<WebView
    android:id="@+id/webWiew"
    android:layout_width="match_parent"
    android:layout_height="match_parent" />
```



URL 통신 예제 1

■ 사용자 인터페이스

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

    <TextView
        android:id="@+id/textViewt"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:text="" />

</ScrollView>
</FrameLayout>
</LinearLayout>
```



URL 통신 예제 1

■ MainActivity.JAVA

```
public class MainActivity2 extends AppCompatActivity {  
    final String page = "http://10.50.80.19:8887/index.html";  
    private TextView textView;  
    private WebView webView;  
    private int type = 1;
```

@Override

```
public void onCreate(Bundle savedInstanceState) {  
    super.onCreate(savedInstanceState);  
    setContentView(R.layout.activity_main2);  
  
    textView = findViewById(R.id.textView);  
    webView = findViewById(R.id.webView);  
    webView.getSettings().setJavaScriptEnabled(true);
```




URL 통신 예제 1

■ MainActivity.JAVA

```
Button button1 = findViewById(R.id.button1);
button1.setOnClickListener(new View.OnClickListener() {
    @Override
    public void onClick(View v) {
        textView.setText("");
        webView.loadUrl(page);
    }
});
```

```
Button button2 = findViewById(R.id.button2);
button2.setOnClickListener(new View.OnClickListener() {
    @Override
    public void onClick(View v) {
        webView.clearView();
        textView.setText("");
        if (type == 1) {
```



URL 통신 예제 1

■ MainActivity.JAVA

```
StrictMode.ThreadPolicy policy =  
    new StrictMode.ThreadPolicy.Builder().permitAll().build();  
StrictMode.setThreadPolicy(policy);  
HtmlDown html = new HtmlDown(getBaseContext());  
textView.setText(html.download(page));  
} else if (type == 2) {  
    HtmlDownTask task = new HtmlDownTask(getBaseContext());  
    try {  
        String temp = task.execute(page).get();  
        textView.setText(temp);  
    } catch (ExecutionException | InterruptedException e) {  
        Toast.makeText(getBaseContext(), e.getMessage(),  
                        Toast.LENGTH_SHORT).show();  
    }  
} else if (type == 3) {
```



URL 통신 예제 1

■ MainActivity.JAVA

```
HtmlDownloadThread thread =  
    new HtmlDownloadThread(getBaseContext(), page);  
thread.start();  
try {  
    thread.join();  
    textView.setText(thread.getResult());  
} catch (InterruptedException e) {  
    Toast.makeText(getBaseContext(), e.getMessage(),  
                   Toast.LENGTH_SHORT).show();  
}  
} else {
```



URL 통신 예제 1

■ MainActivity.JAVA

```
HtmlDownloadRunnable runnable =
    new HtmlDownloadRunnable(getBaseContext(), page);
Thread thread = new Thread(runnable);
thread.start();
try {
    thread.join();
    textView.setText(runnable.getResult());
} catch (InterruptedException e) {
    Toast.makeText(getBaseContext(), e.getMessage(),
        Toast.LENGTH_SHORT).show();
}
}
```



URL 통신 예제 1

■ MainActivity.JAVA

@Override

```
public boolean onCreateOptionsMenu(Menu menu) {  
    MenuInflater inflater = getMenuInflater();  
    inflater.inflate(R.menu.menu, menu);  
    return true;  
}
```

@Override

```
public boolean onOptionsItemSelected(MenuItem item) {  
    switch (item.getItemId()) {  
        case R.id.item1:  
            type = 1;  
            break;  
        case R.id.item2:  
            type = 2;  
            break;  
    }
```



URL 통신 예제 1

■ MainActivity.JAVA

```
        case R.id.item3:
            type = 3;
            break;
        case R.id.item4:
            type = 4;
        }
        webView.clearView();
        textView.setText("");
        item.setChecked(true);
        return true;
    }
}
```



URL 통신 예제 1

■ HtmlDown.JAVA

```
public class HtmlDown {  
    private Context context;  
  
    public HtmlDown(Context context) {  
        this.context = context;  
    }  
  
    public String download(String page) {  
        StringBuilder buffer = new StringBuilder();  
        try {  
            URL url = new URL(page);  
            InputStream inputStream = url.openStream();  
            InputStreamReader reader = new InputStreamReader(inputStream);  
            Scanner scanner = new Scanner(reader);  
            while (scanner.hasNext()) {  
                String line = scanner.nextLine();  
                buffer.append(line).append("\n");  
            }  
        }  
    }  
}
```



URL 통신 예제 1

■ HtmlDown.JAVA

```
        inputStream.close();
        reader.close();
        scanner.close();
    } catch (Exception e) {
        Toast.makeText(context, e.getMessage(),
                        Toast.LENGTH_SHORT).show();
    }

    return buffer.toString();
}
```




URL 통신 예제 1



■ HtmlDownloadTask.JAVA

```
public class HtmlDownloadTask extends AsyncTask<String, String, String> {  
    private Context context;
```

```
    public HtmlDownloadTask(Context context) {  
        this.context = context;  
    }
```

@Override

```
    protected String doInBackground(String... strings) {  
        StringBuilder buffer = new StringBuilder();  
        try {  
            URL url = new URL(strings[0]);  
            InputStream inputStream = url.openStream();  
            InputStreamReader streamReader = new InputStreamReader(inputStream);  
            BufferedReader reader = new BufferedReader(streamReader);  
            String line;
```



URL 통신 예제 1

■ HtmlDownloadTask.JAVA

```
while ((line = reader.readLine()) != null) {  
    buffer.append(line).append("\n");  
}  
inputStream.close();  
streamReader.close();  
reader.close();  
} catch (Exception e) {  
    publishProgress(e.getMessage());  
}
```

```
return buffer.toString();  
}
```

@Override

```
protected void onProgressUpdate(String... values) {  
    Toast.makeText(context, values[0], Toast.LENGTH_SHORT).show();  
}  
}
```



URL 통신 예제 1



■ HtmlDownloadThread.JAVA

```
public class HtmlDownloadThread extends Thread{
    private Context context;
    private String page;
    private Handler handler = new Handler();
    private StringBuilder builder = new StringBuilder();

    public HtmlDownloadThread(Context context, String page) {
        this.context = context;
        this.page = page;
    }

    @Override
    public void run() {
        try {
            URL url = new URL(page);
            InputStream inputStream = url.openStream();
            InputStreamReader streamReader = new InputStreamReader(inputStream);
            BufferedReader reader = new BufferedReader(streamReader);
```



URL 통신 예제 1

```
String line;
while ((line = reader.readLine()) != null) {
    builder.append(line + "\n");
}
inputStream.close();
streamReader.close();
reader.close();
} catch (Exception e) {
    handler.post(new Runnable() {
        @Override
        public void run() {
            Toast.makeText(context, e.getMessage(),
                           Toast.LENGTH_SHORT).show();
        }
    });
}
}
public String getResult() {
    return builder.toString();
}
}
```



URL 통신 예제 1



■ HtmlDownRunnable.JAVA

```
public class HtmlDownRunnable implements Runnable{
    private Context context;
    private String page;
    private Handler handler = new Handler();
    private StringBuilder builder = new StringBuilder();

    public HtmlDownRunnable(Context context, String page) {
        this.context = context;
        this.page = page;
    }

    @Override
    public void run() {
        try {
            URL url = new URL(page);
            InputStream inputStream = url.openStream();
            InputStreamReader streamReader = new InputStreamReader(inputStream);
            BufferedReader reader = new BufferedReader(streamReader);
```



URL 통신 예제 1

■ HtmlDownloadRunnable.JAVA

```
String line;
while ((line = reader.readLine()) != null) {
    builder.append(line + "\n");
}
inputStream.close();
streamReader.close();
reader.close();
} catch (Exception e) {
    handler.post(new Runnable() {
        @Override
        public void run() {
            Toast.makeText(context, e.getMessage(),
                           Toast.LENGTH_SHORT).show();
        }
    });
}
}
```



URL 통신 예제 1

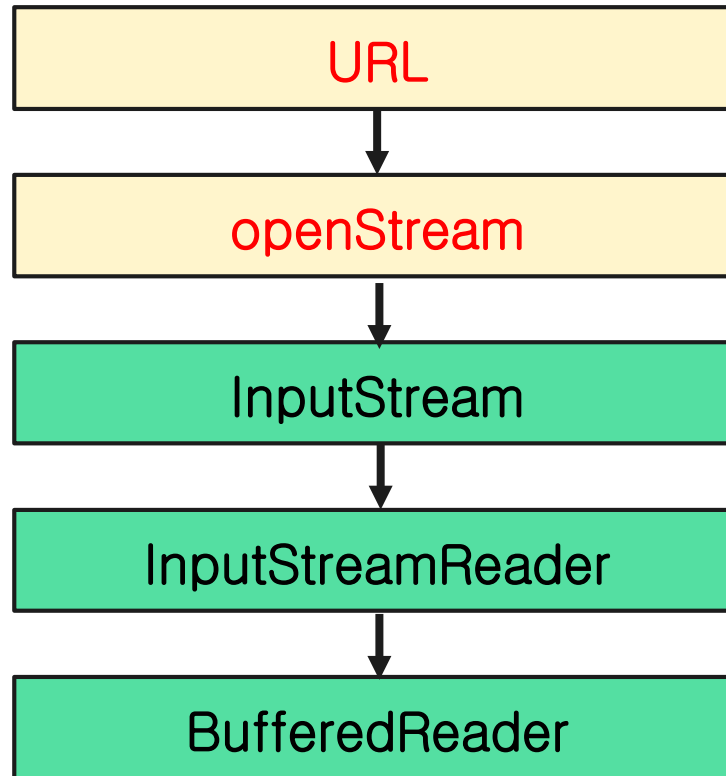
■ HtmlDownRunnable.JAVA

```
public String getResult() {  
    return builder.toString();  
}  
}
```



URL 통신 예제 1

- Web Server에서 HTML 문서 받아오기



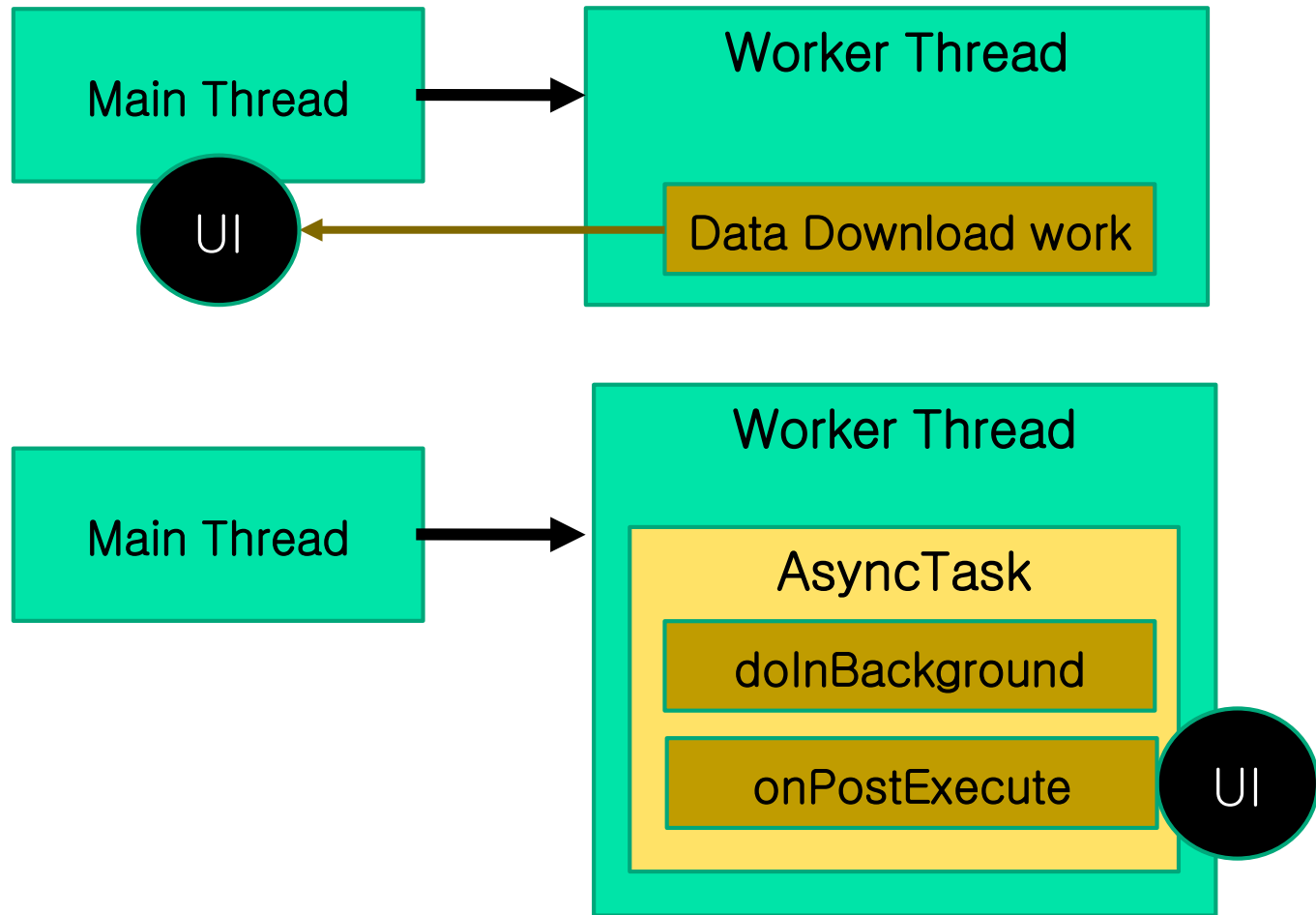
- openStream() 메소드

- url 주소를 InputStream에 넣어주는 것. stream에 넣어주었으니 이제 다양한 입출력 기능을 할 수가 있음



URL 통신 예제 1

- Web Server에서 HTML 문서 받아오기





URL 통신 예제 1



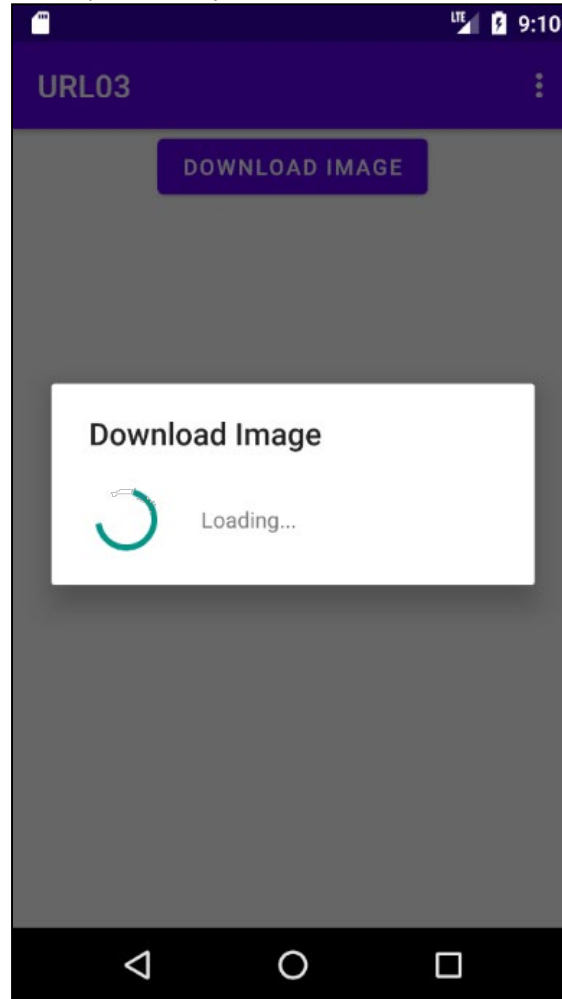
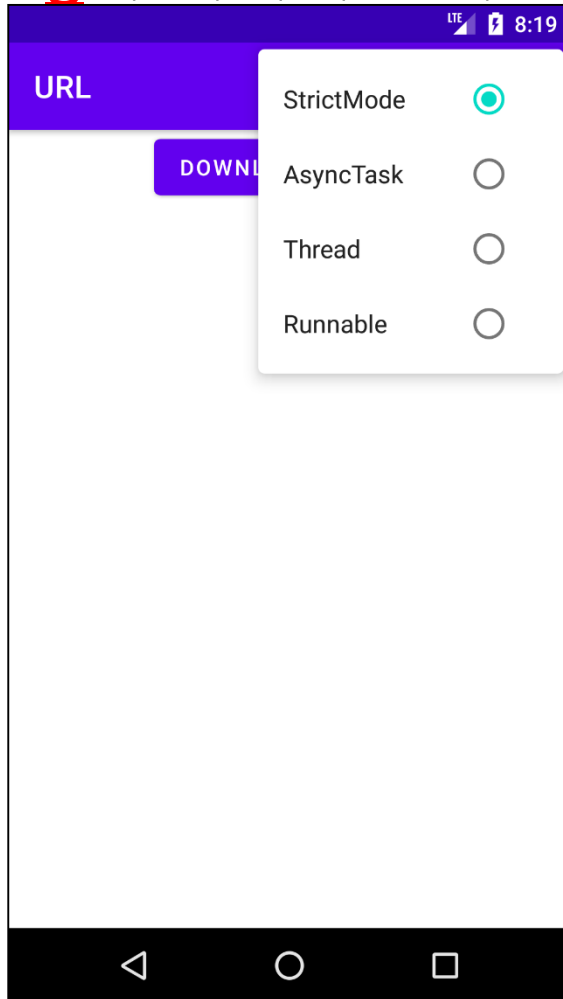
- 접속하려는 Site https://가 아니면 아래 Code를 AndroidManifest.xml 파일의 application tag에 추가 함

```
android:usesCleartextTraffic="true"
```



URL 통신 예제 2

- http://news.kyungbok.ac.kr/images/0087877437_041/2.jpg의 이미지를 다운받아보자





URL 통신 예제 2



■ 사용자 인터페이스

```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout 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"
    android:orientation="vertical"
    tools:context=".MainActivity">

    <Button
        android:id="@+id/button"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_gravity="center"
        android:text="Download Image" />

    <ImageView
        android:id="@+id/image"
        android:layout_width="match_parent"
        android:layout_height="wrap_content" />

</LinearLayout>
```



URL 통신 예제 2

■ MainActivity.JAVA

```
public class MainActivity3 extends AppCompatActivity {
    String imageURL = "http://news.kyungbok.ac.kr/images/0087877437_041/2.jpg";
    ImageView imageView;
    int type = 1;

    Handler handler = new Handler(Looper.getMainLooper()) {
        public void handleMessage(Message msg) {
            Bitmap bitmap = (Bitmap) msg.obj;
            if (bitmap == null) {
                Toast.makeText(getApplicationContext(), "bitmap is null",
                    Toast.LENGTH_SHORT).show();
            } else {
                imageView.setImageBitmap(bitmap);
            }
        }
    };
};
```



URL 통신 예제 2

■ MainActivity.JAVA

```
public void onCreate(Bundle savedInstanceState) {  
    super.onCreate(savedInstanceState);  
    setContentView(R.layout.activity_main3);  
  
    imageView = findViewById(R.id.image);  
    Button button = findViewById(R.id.button);  
    button.setOnClickListener(new View.OnClickListener() {  
        @Override  
        public void onClick(View v) {  
            if (type == 1) {  
                StrictMode.ThreadPolicy policy =  
                    new StrictMode.ThreadPolicy.Builder().permitAll().build();  
                StrictMode.setThreadPolicy(policy);  
                ImageDown imageDown = new ImageDown(getBaseContext());  
                Bitmap bitmap = imageDown.download(imageURL);  
                imageView.setImageBitmap(bitmap);  
            } else if (type == 2) {
```



URL 통신 예제 2



■ MainActivity.JAVA

```
    } else if (type == 2) {  
        ImageDownTask task = new ImageDownTask(  
                                getBaseContext(), imageView);  
        task.execute(imageURL);  
    } else {  
        ImageDownThread thread = new ImageDownThread(  
                                getBaseContext(), imageURL, imageView, handler);  
        thread.start();  
    }  
} else {
```



URL 통신 예제 2

■ MainActivity.JAVA

```
ImageDownRunnable runnable = new ImageDownRunnable(  
                                getBaseContext(), image, handler);  
    Thread thread = new Thread(runnable);  
    thread.start();  
}  
}  
});  
}  
  
@Override  
public boolean onCreateOptionsMenu(Menu menu) {  
    MenuInflater inflater = getMenuInflater();  
    inflater.inflate(R.menu.menu, menu);  
    return true;  
}
```




URL 통신 예제 2



@Override

```
public boolean onOptionsItemSelected(MenuItem item) {  
    switch (item.getItemId()) {  
        case R.id.item1:  
            type = 1;  
            break;  
        case R.id.item2:  
            type = 2;  
            break;  
        case R.id.item3:  
            type = 3;  
            break;  
        case R.id.item4:  
            type = 4;  
    }  
    imageView.setImageBitmap(null);  
    item.setChecked(true);  
    return true;  
}  
}
```



URL 통신 예제 2

■ ImageDown.JAVA

```
public class ImageDown {  
    private Context context;  
    private ProgressDialog dialog;  
  
    public ImageDown(Context context) {  
        this.context = context;  
        dialog = new ProgressDialog(context);  
        dialog.setTitle("Download Image");  
        dialog.setMessage("Loading...");  
        dialog.setIndeterminate(false);  
        dialog.show();  
    }  
}
```



URL 통신 예제 2



■ ImageDown.JAVA

```
public Bitmap download(String page) {  
    Bitmap bitmap = null;  
    try {  
        URL url = new URL(page);  
        InputStream input = url.openStream();  
        bitmap = BitmapFactory.decodeStream(input);  
    } catch (Exception e) {  
        Toast.makeText(context, e.getMessage(), Toast.LENGTH_SHORT).show();  
    }  
    dialog.dismiss();  
    return bitmap;  
}
```



URL 통신 예제 2

■ ImageDownTask.JAVA

```
public class ImageDownTask extends AsyncTask<String, String, Bitmap> {  
    private Context context;  
    private ImageView imageView;  
    private ProgressDialog dialog;  
  
    public ImageDownTask(Context context, ImageView imageView) {  
        this.context = context;  
        this.imageView = imageView;  
    }  
    @Override  
    protected void onPreExecute() {  
        dialog = new ProgressDialog(context);  
        dialog.setTitle("Download Image");  
        dialog.setMessage("Loading...");  
        dialog.setIndeterminate(false);  
        dialog.show();  
    }  
}
```



URL 통신 예제 2

■ ImageDownTask.JAVA

@Override

```
protected Bitmap doInBackground(String... strings) {  
    Bitmap bitmap = null;  
    try {  
        URL url = new URL(strings[0]);  
        InputStream input = url.openStream();  
        bitmap = BitmapFactory.decodeStream(input);  
    } catch (Exception e) {  
        publishProgress(e.getMessage());  
    }  
  
    return bitmap;  
}
```



URL 통신 예제 2

■ ImageDownTask.JAVA

@Override

```
protected void onProgressUpdate(String... values) {  
    Toast.makeText(context, values[0], Toast.LENGTH_SHORT).show();  
    dialog.dismiss();  
}
```

@Override

```
protected void onPostExecute(Bitmap result) {  
    imageView.setImageBitmap(result);  
    dialog.dismiss();  
}  
}
```



URL 통신 예제 2

■ ImageDownThread.JAVA

```
public class ImageDownThread extends Thread{
    private Context context;
    private String imageUrl;
    private ProgressDialog dialog;
    private Handler handler;

    public ImageDownThread(Context context, String imageUrl,
                                                                    Handler handler) {

        this.context = context;
        this.imageUrl = imageUrl;
        this.handler = handler;
        dialog = new ProgressDialog(context);
        dialog.setTitle("Download Image");
        dialog.setMessage("Loading...");
        dialog.setIndeterminate(false);
        dialog.show();
    }
}
```



URI 통신 예제 2

```
@Override
public void run() {
    try {
        URL url = new URL(imageUrl);
        InputStream inputStream = url.openStream();
        Bitmap bitmap = BitmapFactory.decodeStream(inputStream);
        inputStream.close();
        Message message = handler.obtainMessage();
        message.obj = bitmap;
        dialog.dismiss();
        handler.sendMessage(message);
    } catch (Exception e) {
        handler.post(new Runnable() {
            @Override
            public void run() {
                Toast.makeText(context, e.getMessage(),
                                Toast.LENGTH_SHORT).show();
                dialog.dismiss();
            }
        });
    }
}
```




URL 통신 예제 2



■ ImageDownRunnable.JAVA

```
public class ImageDownRunnable implements Runnable{
    private Context context;
    private String imageUrl;
    private ProgressDialog dialog;
    private Handler handler;

    public ImageDownRunnable(Context context, String imageUrl, Handler handler) {
        this.context = context;
        this.imageUrl = imageUrl;
        this.handler = handler;
        dialog = new ProgressDialog(context);
        dialog.setTitle("Download Image");
        dialog.setMessage("Loading...");
        dialog.setIndeterminate(false);
        dialog.show();
    }
}
```



URI 통신 예제 2

@Override

```
public void run() {
```

```
    try {
```

```
        URL url = new URL(imageUrl);
```

```
        InputStream inputStream = url.openStream();
```

```
        Bitmap bitmap = BitmapFactory.decodeStream(inputStream);
```

```
        inputStream.close();
```

```
        Message message = handler.obtainMessage();
```

```
        message.obj = bitmap;
```

```
        dialog.dismiss();
```

```
        handler.sendMessage(message);
```

```
    } catch (Exception e) {
```

```
        handler.post(new Runnable() {
```

```
            @Override
```

```
            public void run() {
```

```
                Toast.makeText(context, e.getMessage(),
```

```
                    Toast.LENGTH_SHORT).show();
```

```
                dialog.dismiss();
```

```
            }
```

```
        });
```

```
    }
```

```
}
```

```
}
```