Chapter 10.
URL 관련클래스

Mingyu Lim

Collaborative Computing Systems Lab,
School of Internet & Multimedia Engineering
Konkuk University, Seoul, Korea

■ 학습 목표

- ❑ URL이란
- ❑ URL클래스
- ❑ URL클래스를 이용해서 웹사이트 읽어오기
- ❑ URLConnection 클래스로 웹 페이지 읽기
- ❑ GET 방식으로 URL 주소 호출하기
- ❑ POST 방식으로 URL 주소 호출하기
- ❑ URLEncoder 클래스를 이용한 문자열의 변환
- ❑ URLDecoder 클래스를 이용한 디코딩

URL 이란

- Uniform Resource Locator
  - 인터넷에서 접근 가능한 자원의 주소를 일관되게 표현할 수 있는 형식
  - 자원: HTML, 이미지, CGI, 자바애플릿 등..
- URL 사용예
  - http://sunny.sarang.net/sunny
  - http://sunny.sarang.net/sunny/sunny.gif
  - ftp://sunny.sarang.net/java.txt

URL 클래스

❑ URL 클래스 목적

- URL 주소 형식 검사
- URL 주소로부터 사용되는 프로토콜, 서버명, 포트, 파일명 구하기
- URL 주소로 지정된 파일 읽어들일때
  - TCP 사용할때와의 차이?

❑ URLInfo (예제10-1)

- 사용자입력의 URL 적합성 검사
  - Java.net.MalformedURLException
- URL클래스의 메소드 이용한 여러 정보 구하기
  - getProtocol(): 프로토콜
  - getHost(): 호스트명
  - getPort(): 포트번호
  - getPath(): 파일명
  - getQuery(): 사용자쿼리

URL클래스로 웹사이트 읽어오기

- URL::openStream()
  - URL클래스에 지정된 URL주소가 가리키는 문서를 읽을 수 있는 InputStream반환
  - GET방식의 웹페이지 호출만 가능 (POST방식은 URLConnection 이용)
- WebSpider (예제10-2)
  - oepnStream()으로 inputStream획득
  - InputStream으로 읽어들인 내용은 FileOutputStream으로 저장

```
InputStream in = url.openStream();
fos = new FileOutputStream(args[1]);
byte[] buffer = new byte[512];
int readcount = 0;
System.out.println("읽어오기 시작합니다.");
while((readcount = in.read(buffer)) != -1){
  fos.write(buffer,0, readcount);
}
```

URL클래스로 웹사이트 읽어오기

- WebSpider 실행
  - Java WebSpider http://kr.yahoo.com yahoo.html

URLConnection 클래스

- 역할
  - URL주소의 내용 읽기
  - URL주소가 가리키는 웹애플리케이션에 GET,POST 방식으로 정보 전달
- URL주소의 내용 읽기
  - URL 객체 생성
  - URL객체의 openConnection() 호출하여 URLConnection 객체 획득
  - URLConnection객체의 getInputStream() 호출하여 InputStream객체 획득
  - InputStream 이용해서 URL주소의 내용 읽기

URLConnection 클래스

❑ WebSpiderWithURLConnection (예제 10-3)

- URL 객체 생성
  - ◆ URL url = null; url = new URL(args[0]);
- URL객체의 openConnection() 호출하여 URLConnection 객체 획득
  - ◆ URLConnection urlcon = url.openConnection();
- URLConnection객체의 getInputStream() 호출하여 InputStream객체 획득
  - ◆ InputStream in = urlcon.getInputStream();
- 읽을 문서 형식, 작성시간 등 정보

```
URLConnection urlcon = url.openConnection();
String contentType = urlcon.getContentType();
long d1 = urlcon.getDate();
java.util.Date d = new java.util.Date(d1);
java.text.SimpleDateFormat format = new java.text.SimpleDateFormat("yyyy-MM-dd HH:mm:ss a");
String sdate = format.format(d);
System.out.println("Content Type : " + contentType);
System.out.println("읽어온 시간 : " + sdate);
```

GET 방식 쿼리 전달

❑ Daum 검색 예제

- http://search.daum.net/cgi-bin/nsp/search.cgi?w=tot&q=%BB%E7%B0%FA
  - ◆ ?이하는 사용자 쿼리
  - ◆ %BB%E7%B0%FA : 사과

❑ DaumSearch (예제 10-4)

- URLConnection객체로 주소 읽기와 주소에 쓰기모두를 true로 설정

```
URL url = new URL(u);
URLConnection connection = url.openConnection();
HttpURLConnection hurlc = (HttpURLConnection)connection;
hurlc.setRequestMethod("GET");
hurlc.setDoOutput(true);
hurlc.setDoInput(true);
hurlc.setUseCaches(false);
hurlc.setDefaultUseCaches(false);
```

GET 방식 쿼리 전달

❑ DaumSearch (예제 10-4)

- URLConnection객체에서 OutputStream을 얻어 PrintWriter객체로 변환후, println() 이용해서 쿼리 전달

```
PrintWriter out = new PrintWriter(hurlc.getOutputStream());
out.println(query);
```

- URLConnection객체에서 InputStream을 얻어 PrintWriter객체의 println()이용하여 파일에 읽은 내용 저장

```
BufferedReader in = new BufferedReader(new
InputStreamReader(hurlc.getInputStream()));
PrintWriter pw = new PrintWriter(new FileWriter(args[1]));
String inputLine = null;
while ((inputLine = in.readLine()) != null){
        pw.println(inputLine);
}
```

POST 방식 쿼리 전달

❑ POST방식의 특징

- 사용자 쿼리 길이 제한 없음
- URL주소에 쿼리내용 표시되지 않음
- HTML폼이나 POST방식 쿼리 전송 프로그램 이용해서 쿼리 전달

❑ PostConnection (예제 10-6)

- POST방식 설정

```
URL url = new URL(u);
URLConnection connection = url.openConnection();
HttpURLConnection hurlc = (HttpURLConnection)connection;
hurlc.setRequestMethod("POST");
```

- 사용자 쿼리

```
String query = "id=" + id + "&passwd=" + passwd;
PrintWriter out = new
PrintWriter(hurlc.getOutputStream());
out.println(query);
```

URLEncoder/URLDecoder

❑ URLEncoder

- 시스템마다 문자 인식 방법이 틀리기 때문에, 아스키 문자가 아닌 문자열을 특수 문자로 변환할 필요

```
String encodeStr = URLEncoder.encode(args[0]);
String encodeStr = URLEncoder.encode(args[0], enc);
```

- Enc: UTF-8, UTF-16BE, UTF-16LE, UTF-16, ISO-8859-1, US-ASCII

❑URLDecoder

- Encoder와 반대 역할

```
String decodeStr = URLEncoder.decode(args[0]);
String decodeStr = URLEncoder.decode(args[0], enc);
```