CSE3040 Java Language

Lecture 22: Networking with Java (2)

Dept. of Computer Engineering,

Sogang University

서강대학교
SOGANG UNIVERSITY

Reading contents from an HTML file

- We can read data from a URL, similar to how we read data from a file.
  - Get an InputStream object from URL using openStream method.
  - Use InputStreamReader and BufferedReader to read contents from the stream.

```
URL url = null;
BufferedReader input = null;
String address = "https://icslsogang.github.io/courses/cse3040/hello.html";
String line = "";

try {
     url = new URL(address);
     input = new BufferedReader(new InputStreamReader(url.openStream()));

     while((line=input.readLine()) != null) {
           System.out.println(line);
     }
     input.close();
} catch(Exception e) {
     e.printStackTrace();
}
```

서강대학교
SOGANG UNIVERSITY

Downloading a file from a URL

- A URL can be downloaded as a file.

```
URL url = null;
InputStream in = null;
FileOutputStream out = null;
String address = "https://icslsogang.github.io/courses/cse3040/sogang_campus.jpg";

int ch = 0;
try {
     url = new URL(address);
     in = url.openStream();
     out = new FileOutputStream("sogang_campus.jpg");
     while((ch=in.read()) != -1) {
           out.write(ch);
     }
     in.close();
     out.close();
} catch(Exception e) {
     e.printStackTrace();
}
System.out.println("File download complete.");
```

Parsing an HTML file

- Useful information is gathered by processing the HTML file.

```
public class Lecture {
    static ArrayList<String> lines = new ArrayList<String>();

    public static void main(String[] args) {
        URL url = null;
        BufferedReader input = null;
        String address = "http://www.kyobobook.co.kr/bestSellerNew/bestseller.laf";
        String line = "";

        try {
            url = new URL(address);
            input = new BufferedReader(new InputStreamReader(url.openStream()));
            while((line=input.readLine()) != null) {
                if(line.trim().length() > 0) lines.add(line);
            }
            input.close();
        } catch(Exception e) {
            e.printStackTrace();
        }
```

Parsing an HTML file

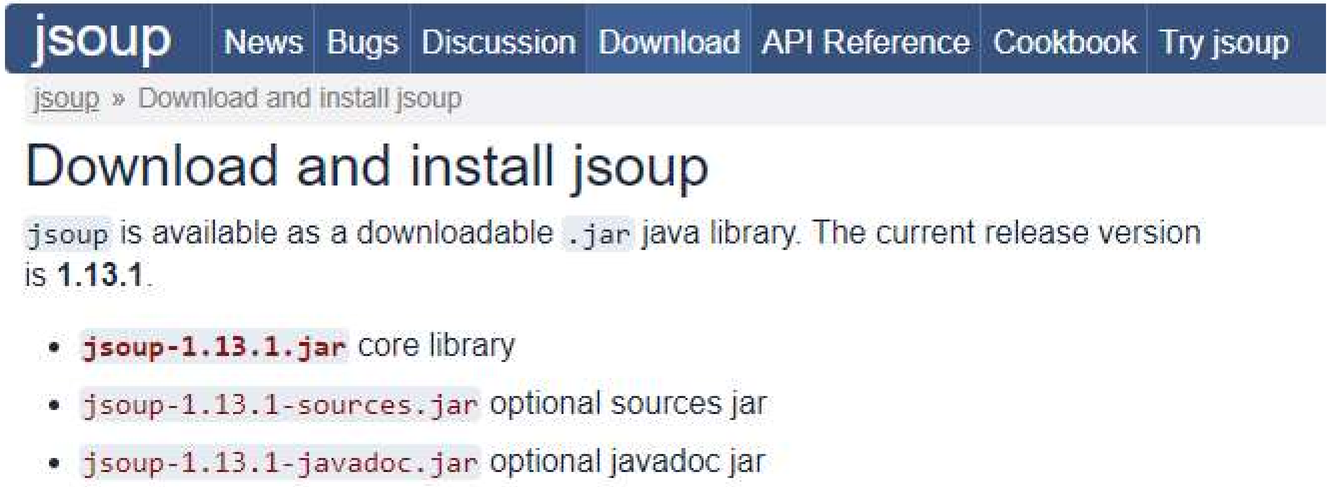
- Useful information is gathered by processing the HTML file. (cont.)

```
        int rank = 1;
        int status = 0;
        for(int i=0; i<lines.size(); i++) {
            String l = lines.get(i);
            if(status == 0) {
                if(l.contains("div class=\"detail\"")) status = 1;
            } else if(status == 1) {
                if(l.contains("div class=\"title\"")) status = 2;
            } else if(status == 2) {
                if(l.contains("a href")) {
                    int begin = l.indexOf("<strong>") + "<strong>".length();
                    int end = l.indexOf("</strong>");
                    System.out.println(rank + "위: " + l.substring(begin, end));
                    status = 0;
                    rank++;
                }
            }
        }
    }
}
```

Parsing an HTML file using jsoup

- jsoup: an external Java library that provides a convenient API for extracting and manipulating data from HTML files.

- Download jsoup-1.13.1.jar
  - https://jsoup.org/download



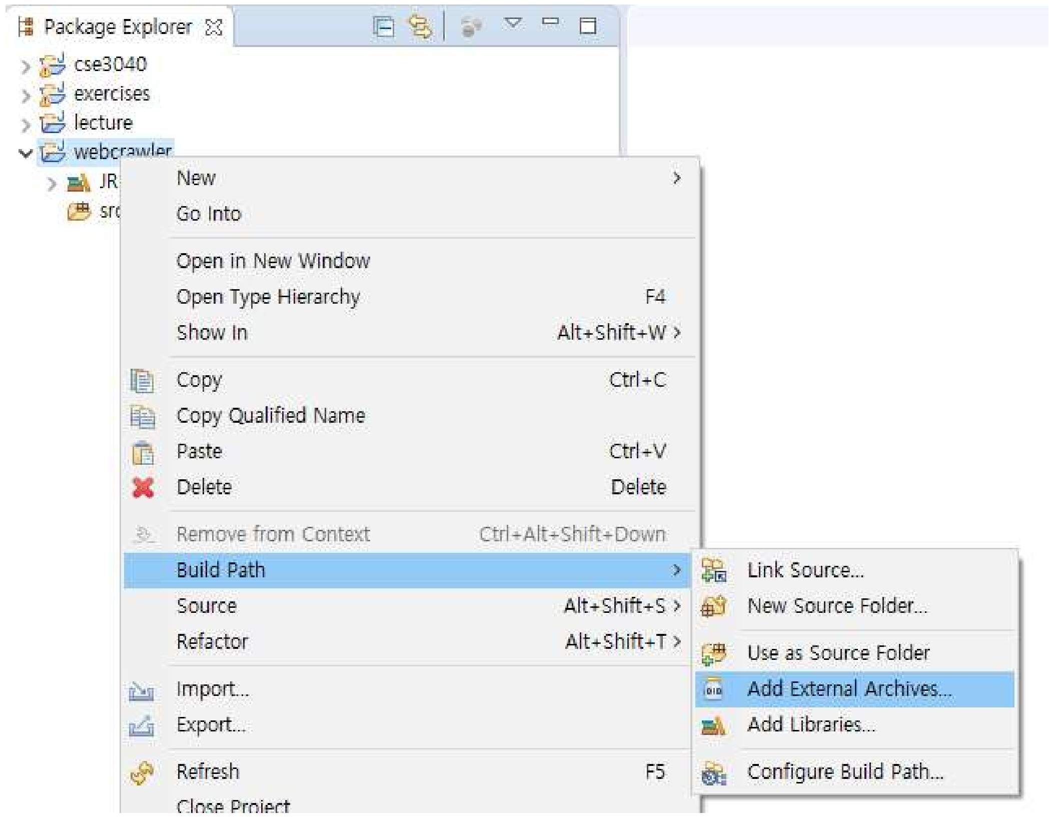
Download and install jsoup

`jsoup` is available as a downloadable `.jar` java library. The current release version is **1.13.1**.

- **`jsoup-1.13.1.jar`** core library
- `jsoup-1.13.1-sources.jar` optional sources jar
- `jsoup-1.13.1-javadoc.jar` optional javadoc jar

Parsing an HTML file using jsoup

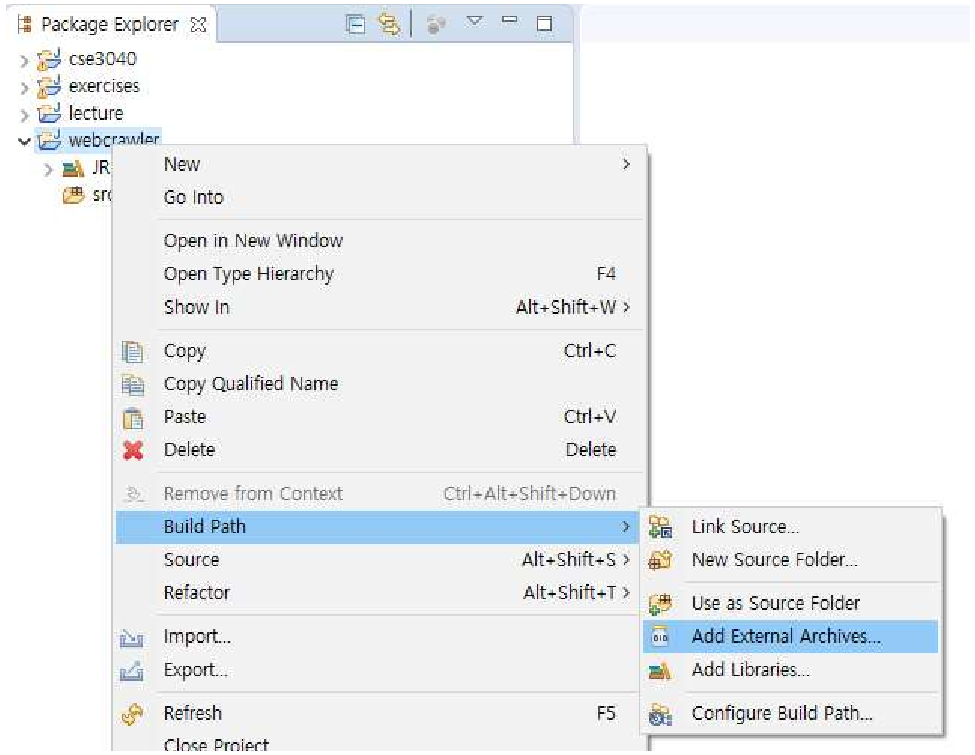
- In eclipse, right-click on the project and choose "Build Path".
  - Then, choose "Add External Archives..."

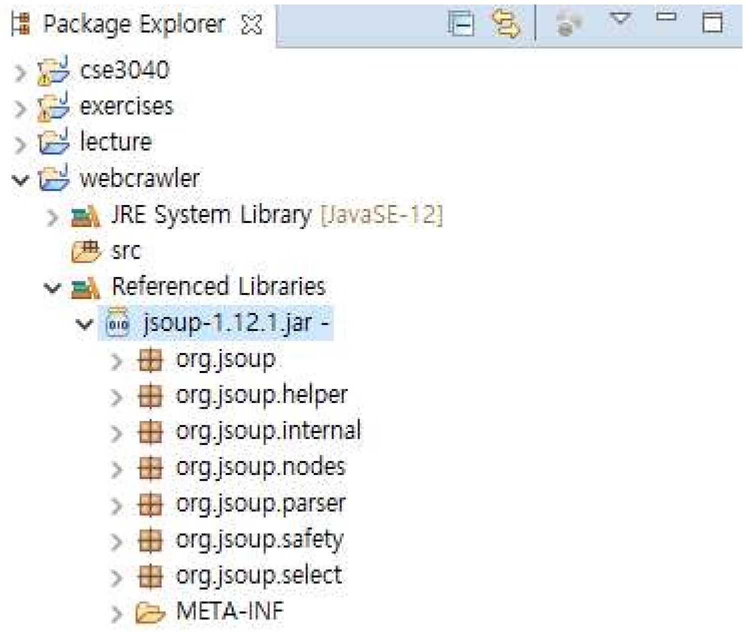


Parsing an HTML file using jsoup

- In eclipse, right-click on the project and choose "Build Path".
  - Then, choose "Add External Archives..."
  - Choose file jsoup-1.13.1.jar
  - Now we can import classes contained in jsoup-1.13.1.jar.

  ※ If you have "module-info.java" in the project, delete the file.





Parsing an HTML file using jsoup

- Extracting data is simpler with jsoup.

```
import java.io.IOException;
import org.jsoup.Jsoup;
import org.jsoup.nodes.Document;
import org.jsoup.select.Elements;

public class Lecture {
    public static void main(String[] args) throws Exception {
        String url = "http://www.kyobobook.co.kr/bestSellerNew/bestseller.laf";
        Document doc = null;
        try {
            doc = Jsoup.connect(url).get();
        } catch(IOException e) {
            System.out.println(e.getMessage());
        }

        Elements bestsellers = doc.select("div.detail");
        Elements titles = bestsellers.select("div.title");
        Elements booktitles = titles.select("a[href]");

        for(int i=0; i<booktitles.size(); i++) {
           System.out.println(i+1 + "위: " + booktitles.eq(i).text());
        }
    }
}
```

Parsing an HTML file using jsoup

- Connect to a URL using Jsoup.connect method.
  - It returns a Connection object.
  - The Connection classes has get method, which returns a Document object.
    - class Document is a subclass of class Element.
    - The get method throws the following exceptions; We should use a try-catch block.
      - MalformedURLException, HttpStatusException, UnsupportedMimeTypeException, SocketTimeoutException, IOException

```
try {
    doc = Jsoup.connect(url).get();
} catch(IOException e) {
    System.out.println(e.getMessage());
}
```

Parsing an HTML file using jsoup

- The select method is the core method for extracting data from an HTML file.
- The method is defined in class Element.

Method	Description
Elements select(String cssQuery)	Find elements that match the Selector CSS query, with this element as the starting context.

```
Elements bestsellers = doc.select("div.detail");
Elements titles = bestsellers.select("div.title");
Elements booktitles = titles.select("a[href]");
```

- The first line finds the part which starts with <div class="detail">
- The second line finds the part which starts with <div class="title">
- The third line finds the part which starts with <a href=...>

Parsing an HTML file using jsoup

- Selector (CSS query) syntax
  - tagname: find elements by tag, e.g. a
  - ns|tag: find elements by tag in a namespace, e.g. fb|name finds <fb:name> elements.
  - #id: find elements by ID, e.g. #logo
  - .class: find elements by class name, e.g. .masthead
  - [attribute]: elements with attribute, e.g. [href]
  - [^attr]: elements with an attribute name prefix, e.g. [^data-] finds elements with HTML5 dataset attributes
  - [attr=value]: elements with attribute value, e.g. [width=500]
  - [attr^=value], [attr$=value], [attr*=value]: elements with attributes that start with, end with, or contain the value, e.g. [href*=/path/]
  - [attr!=regex]: elements with attribute values that match the regular expression, e.g. img[src!=(?i)\.(png|jpe?g)]
  - *: all elements, e.g. *

Programming Lab #22

22-01. Reading Contents from an HTML File

- Execute the following code and understand the results.

```
public class Ex22_01 {
	public static void main(String[] args) {
		URL url = null;
		BufferedReader input = null;
		String address = "https://icslsogang.github.io/courses/cse3040/hello.html";
		String line = "";

		try {
			url = new URL(address);
			input = new BufferedReader(new InputStreamReader(url.openStream()));

			while((line=input.readLine()) != null) {
				System.out.println(line);
			}
			input.close();
		} catch(Exception e) {
			e.printStackTrace();
		}
	}
}
```

22-02. Downloading a File from a URL

- Execute the following code and understand the results.

```
public class Ex22_02 {
      public static void main(String[] args) {
             URL url = null;
             InputStream in = null;
             FileOutputStream out = null;
             String address = "https://icslsogang.github.io/courses/cse3040/sogang_campus.jpg";

             int ch = 0;
             try {
                          url = new URL(address);
                          in = url.openStream();
                          out = new FileOutputStream("sogang_campus.jpg");
                          while((ch=in.read()) != -1) {
                                       out.write(ch);
                          }
                          in.close();
                          out.close();
             } catch(Exception e) {
                          e.printStackTrace();
             }
             System.out.println("File download complete.");
      }
}
```

22-03. Parsing an HTML File

- Execute the following code and understand the results.

```
public class Ex22_03 {
    static ArrayList<String> lines = new ArrayList<String>();

    public static void main(String[] args) {
        URL url = null;
        BufferedReader input = null;
        String address = "http://www.kyobobook.co.kr/bestSellerNew/bestseller.laf";
        String line = "";

        try {
            url = new URL(address);
            input = new BufferedReader(new InputStreamReader(url.openStream()));
            while((line=input.readLine()) != null) {
                if(line.trim().length() > 0) lines.add(line);
            }
            input.close();
        } catch(Exception e) {
            e.printStackTrace();
        }
```

22-03. Parsing an HTML File

- Execute the following code and understand the results. (cont.)

```
 int rank = 1;
        int status = 0;
        for(int i=0; i<lines.size(); i++) {
            String l = lines.get(i);
            if(status == 0) {
                if(l.contains("div class=\"detail\"")) status = 1;
            } else if(status == 1) {
                if(l.contains("div class=\"title\"")) status = 2;
            } else if(status == 2) {
                if(l.contains("a href")) {
                    int begin = l.indexOf("<strong>") + "<strong>".length();
                    int end = l.indexOf("</strong>");
                    System.out.println(rank + "위: " + l.substring(begin, end));
                    status = 0;
                    rank++;
                }
            }
        }
    }
}
```

22-04. Parsing an HTML File using jsoup

- Execute the following code and understand the results.

```
import java.io.IOException;
import org.jsoup.Jsoup;
import org.jsoup.nodes.Document;
import org.jsoup.select.Elements;

public class Ex22_04 {
    public static void main(String[] args) throws Exception {
        String url = "http://www.kyobobook.co.kr/bestSellerNew/bestseller.laf";
        Document doc = null;
        try {
            doc = Jsoup.connect(url).get();
        } catch(IOException e) {
            System.out.println(e.getMessage());
        }

        Elements bestsellers = doc.select("div.detail");
        Elements titles = bestsellers.select("div.title");
        Elements booktitles = titles.select("a[href]");

        for(int i=0; i<booktitles.size(); i++) {
            System.out.println(i+1 + "위: " + booktitles.eq(i).text());
        }
    }
}
```

End of Class

Instructor office: AS818A

Email: jso1@sogang.ac.kr