Jsoup HTML Parser

Onepage Tutorial

Small Codes

Programming Simplified

For more tutorials& articles please visit

SmlCodes.com

1.INTRODUCTION

jsoup is a Java library for working with **real-world HTML**. **It provides** a very convenient **API for extracting and manipulating data, using the best of DOM, CSS**, and jquery-like methods.

Some time in our projects we need to parse the HTML file data and use that data in our business logic like parse Students web page, get the details from HTML table of the student page and save it in to database. For doing so we need some HTML parser, JSOUP is the best way to do parsing HTML file in java.

A.JSOUP EXAMPLE : TO PARSE HTML FROM HOSTED WEBSITE (EX.SMLCODES.COM)

```
public class JsoupFromWebsite {
	public static void main(String[] args) throws IOException {

			//1.Open connection with website
			Connection connection = Jsoup.connect("http://www.smlcodes.com");

			//2.parse the connected HTML page and save into Document
			Document document = connection.get();

			//3.write methods to get Elements
			String tit = document.title();

	}
}
```

Output

```
WEBSITE TITLE IS : Small Codes – Programming Simplified
```

B.JSOUP EXAMPLE : TO PARSE HTML FROM LOCAL FILE SYSTEM (EX.D:// DRIVE)

```
public class JsoupFromLocal {
	public static void main(String[] args) throws IOException {
Document doc = Jsoup.parse(new File("D:\\Small Codes\\temp\\sml.html"), "UTF-8");
	  String title = doc.title();
	  System.out.println("TITILE => " + title);
	}
}
```

Output

```
TITILE => Small Codes – Programming Simplified
```

2.INSTALLING JSOUP

To work with JSoup in our projects we need to add **jsoup-1.9.2.jar** .we can get this jar file in two ways.

A.BY USING MAVEN

place the following into your POM's <dependencies> section.

```
<dependency>
  <!-- jsoup HTML parser library @ http://jsoup.org/ -->
  <groupId>org.jsoup</groupId>
  <artifactId>jsoup</artifactId>
  <version>1.9.2</version>
</dependency>
```

B.DOWNLOAD JSOUP-1.9.2.JAR

jsoup is available as a downloadable .jar .For latest version: https://jsoup.org/download

After Downloading we have add this Jar to classpath.

set classpath=jsoup-1.9.2.jar;.;%classpath%

3.JSOUP API

A.PACKAGES

Jsoup provides 7 packages.

Package	Description
org.jsoup	Contains the main **Jsoup** class, provides static access to the jsoup functionality.
org.jsoup.examples	Contains **example programs** and use of jsoup.
org.jsoup.helper	Helper classes
org.jsoup.nodes	**HTML document structure** nodes.

org.jsoup.parser	**Contains the HTML parser**, tag specifications, and HTML tokeniser.
org.jsoup.safety	Contains the jsoup HTML cleaner, and whitelist definitions.
org.jsoup.select	Packages to support **the CSS-style** element selector.

B.CLASSES AND METHODS

Jsoup has many classes. But in real time we use only 3 classes every time.

They are ***Jsoup, Document, and Element***.

1.Jsoup (*org.jsoup.Jsoup*)

The **core public access** point to the jsoup functionality.

Methods:

Modifier and Type	Method and Description
`static Connection`	`connect(String url)` Creates a new `Connection` to a URL.
`static Document`	`parse(File in, String charsetName)` Parse the contents of a file as HTML.
`static Document`	`parse(File in, String charsetName, String baseUri)` Parse the contents of a file as HTML.
`static Document`	`parse(InputStream in, String charsetName, String baseUri)` Read an input stream, and parse it to a Document.
`static Document`	`parse(InputStream in, String charsetName, String baseUri, Parser parser)` Read an input stream, and parse it to a Document.
`static Document`	`parse(String html)` Parse HTML into a Document.
`static Document`	`parse(String html, String baseUri)` Parse HTML into a Document.
`static Document`	`parse(String html, String baseUri, Parser parser)` Parse HTML into a Document, using the provided Parser.
`static Document`	`parse(URL url, int timeoutMillis)` Fetch a URL, and parse it as HTML.
`static Document`	`parseBodyFragment(String bodyHtml)` Parse a fragment of HTML, with the assumption that it forms the body of the HTML.

2.Document (*org.jsoup.nodes.Document*)

It loads the HTML file. Contains no.of methods to do operations on HTML page

Modifier and Type	Method and Description
`Element`	`body()` Accessor to the document's `body` element.
`Charset`	`charset()` Returns the charset used in this document.
`void`	`charset(Charset charset)` Sets the charset used in this document.
`Document`	`clone()` Create a stand-alone, deep copy of this node, and all of its children.
`Element`	`createElement(String tagName)` Create a new Element, with this document's base uri.
`static Document`	`createShell(String baseUri)` Create a valid, empty shell of a document, suitable for adding more elements to.
`Element`	`head()` Accessor to the document's `head` element.
`String`	`location()` Get the URL this Document was parsed from.
`Element`	`text(String text)` Set the text of the `body` of this document.
`String`	`title()` Get the string contents of the document's `title` element.
`void`	`title(String title)` Set the document's `title` element.
`boolean`	`updateMetaCharsetElement()` Returns whether the element with charset information in this document is updated on changes through `Document.charset(Charset)` or not.
`void`	`updateMetaCharsetElement(boolean update)` Sets whether the element with charset information in this document is updated on changes through `Document.charset(Charset)` or not.

3.Element (*org.jsoup.nodes.Element*)

A HTML element consists of a tag name, attributes, and child nodes (including text nodes and other elements). From an Element, you can extract data, traverse the node graph, and manipulate the HTML.

Modifier and Type	Method and Description
Element	append(String html) Add inner HTML to this element.
Element	appendChild(Node child) Add a node child node to this element.
Element	appendElement(String tagName) Create a new element by tag name, and add it as the last child.
Elements	getAllElements() Find all elements under this element (including self, and children of children).
Element	getElementById(String id) Find an element by ID, including or under this element.
String	html() Retrieves the element's inner HTML.
Element	html(String html) Set this element's inner HTML.

If want know about more classes and methods, please refer official document

4.JSOUP EXAMPLES

4.1 LOAD DOCUMENT FROM URL, FILE AND STRING

Example: **JsoupLoadTypes.java**

```
package jsoup;

import java.io.IOException;

import org.jsoup.Connection;
import org.jsoup.Jsoup;
import org.jsoup.nodes.Document;

public class JsoupFromWebsite {
	public static void main(String[] args) throws IOException {

			//1.Open connection with website
			Connection connection = Jsoup.connect("http://smlcodes.com");

			//2.parse the connected HTML page and save into Document
			Document document = connection.get();

			//3.write methods to get Elements
			String tit = document.title();

	}
}
```

4.2 GET ALL LINKS FROM HTML PAGE USING JSOUP

Example: **JsoupGetAllLinks.java**

```
package jsoup;
import org.jsoup.Jsoup;
import org.jsoup.nodes.Document;
import org.jsoup.nodes.Element;
import org.jsoup.select.Elements;

public class JsoupGetAllLinks {
	public static void main(String[] args) throws IOException {
		Document document = Jsoup.parse(new File("D:/Small
Codes/temp/sml.html"), "utf-8");
		Elements links = document.select("a[href]");
		for (Element link : links) {
			System.out.println("Link : " + link.attr("href"));
			System.out.println("Link Text : " + link.text());
		}
	}
}
```

4.3 GET FAV ICON FROM HTML PAGE USING JSOUP

Example: **JsoupFavIcon.java**

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

public class JsoupFavIcon {
	public static void main(String[] args) throws IOException {
		String favicon = "Image Not Found";
		Document document = Jsoup.connect("http://www.smlcodes.com").get();
		Element element =
document.head().select("link[href~=.*\\.(ico|png)]").first();
		if (element == null) {
			element = document.head().select("meta[itemprop=image]").first();
			if (element != null) {
				favicon = element.attr("content");
			}
		} else {
			favicon = element.attr("href");
		}

		System.out.println("FAV ICON Image URL Is ==>  " + favicon);
	}
}
```

4.4 GET ALL IMAGES FROM HTML PAGE USING JSOUP

Example: **JsoupGetAllImages.java**

```
import org.jsoup.nodes.Element;
import org.jsoup.select.Elements;

public class JsoupGetAllImages {
	public static void main(String[] args) throws IOException {
		Document document = Jsoup.connect("http://www.smlcodes.com").get();
		Elements images = document.select("img[src~=(?i)\\.(png|jpe?g|gif)]");
		int i = 1;
		for (Element image : images) {
			System.out.println("\n\n==================\nIMAGE : " + i);
			System.out.println("Image Source : " + image.attr("src"));
			System.out.println("HEIGHT : " + image.attr("height"));
			System.out.println("WIDTH : " + image.attr("width"));
			System.out.println("ALT-Text : " + image.attr("alt"));
			i++;
		}
	}
}
```

4.5 GET META INFORMATION OF AN URL USING JSOUP

Meta information we will found on between <head> tags. It is useful for Indexing & SEO. Some common Meta tags are Keywords, content, viewport etc.

Example: **JsoupGetMeta.java**

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

public class JsoupFavIcon {
	public static void main(String[] args) throws IOException {
		String favicon = "Image Not Found";
		Document document = Jsoup.connect("http://www.smlcodes.com").get();
		Element element =
document.head().select("link[href~=.*\\.(ico|png)]").first();
		if (element == null) {
			element = document.head().select("meta[itemprop=image]").first();
			if (element != null) {
				favicon = element.attr("content");
			}
		} else {
			favicon = element.attr("href");
		}

		System.out.println("FAV ICON Image URL Is ==>  " + favicon);
	}
}
```

4.6 GET FORM PARAMETER'S FROM HTML PAGE USING JSOUP

Example: **JsoupForm.java**

```
import org.jsoup.nodes.Document;
import org.jsoup.nodes.Element;
import org.jsoup.select.Elements;

public class JsoupForm {
	public static void main(String[] args) throws IOException {
		Document document = Jsoup.parse(new File("D:/Small
Codes/temp/form.html"), "utf-8");
		Element formElement = document.getElementById("signup");
		Elements inputElements = formElement.getElementsByTag("input");
		for (Element inputElement : inputElements) {
			String key = inputElement.attr("name");
			String value = inputElement.attr("value");
System.out.println("Element Name ==>  " + key + " \t Element value ==> " + value);
		}
	}
}
```

REF

https://jsoup.org/cookbook/

http://howtodoinjava.com/jsoup/complete-jsoup-tutorial/

http://www.mkyong.com/java/jsoup-html-parser-hello-world-examples/