

jsoup News Bugs Discussion Download API Reference Cookbook Try jsoup

[jsoup](#) » jsoup: Java HTML Parser

jsoup: Java HTML Parser

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.

jsoup implements the **WHATWG HTML5** specification, and parses HTML to the same DOM as modern browsers do.

- scrape and **parse** HTML from a URL, file, or string
- **find** and extract data, using DOM traversal or CSS selectors
- **manipulate** the HTML elements, attributes, and text
- **clean** user-submitted content against a safe white-list, to prevent XSS attacks
- **output** tidy HTML

jsoup is designed to deal with all varieties of HTML found in the wild; from pristine and validating, to invalid tag-soup; jsoup will create a sensible parse tree.

Example

Fetch the **Wikipedia** homepage, parse it to a DOM, and select the headlines from the *In the news* section into a list of **Elements** ([online sample](#)):

```
Document doc = Jsoup.connect("http://en.wikipedia.org/").get();  
Elements newsHeadlines = doc.select("#mp-itn b a");
```

Open source

jsoup is an open source project distributed under the liberal **MIT license**. The source code is available at [GitHub](#).

Getting started

1. **Download** the jsoup jar (version **1.10.2**)
2. Read the cookbook **introduction**
3. Enjoy!

Development and support

If you have any questions on how to use jsoup, or have ideas for future development, please get in touch via the [mailing list](#).

If you find any issues, please [file a bug](#) after checking for duplicates.

The [colophon](#) talks about the history of and tools used to build jsoup.

Status

jsoup is in general release.

Cookbook contents

Introduction

1. [Parsing and traversing a Document](#)

Input

2. [Parse a document from a String](#)
3. [Parsing a body fragment](#)
4. [Load a Document from a URL](#)
5. [Load a Document from a File](#)

Extracting data

6. [Use DOM methods to navigate a document](#)
7. [Use selector-syntax to find elements](#)
8. [Extract attributes, text, and HTML from elements](#)
9. [Working with URLs](#)
10. [Example program: list links](#)

Modifying data

11. [Set attribute values](#)
12. [Set the HTML of an element](#)
13. [Setting the text content of](#)

elements

Cleaning HTML

14. Sanitize untrusted HTML (to prevent XSS)

jsoup HTML parser © 2009 - 2016 Jonathan Hedley