

Noah  
JDOCS  
CIT  
Fall '18  
CCAC

ACCESS:     GitHub:     <https://github.com/nbaertsch/JDOCS>

Explanation:   This project, dubbed the “Java Document Object Complexity Scanner”, aims to quantify the complexity of a website by traversing the DOM and calculating it’s breadth and depth as well as enumerating link tags. It is designed with MVC principles in mind, so as to be extensible by other students or myself in the future. This project leverages the Jsoup library to do the bulk of the heavy lifting, including DOM retrieval over HTTP and basic DOM traversal.

Future Improvements:     In the future, I would like to see this project expand to include other metrics for website efficiency. These could be, for example: mean time-to-load, file-size (including images), and possibly, the program could build a DOM from links and create a model of the entire site. Another idea I had was to ask the user how many URL’s to scan, and then the result could be a sorted list of the websites by complexity.

The image displays two screenshots of a Java application window titled "JDOCS". The window has a standard Mac OS X title bar with a dropdown arrow, minimize, maximize, and close buttons. The interface is divided into two sections. The top section is the input form, and the bottom section shows the results of a scan.

**Top Screenshot (Input Form):**

- URL:** A text field containing "https://github.com/nbaertsch/JDOCS".
- Total Nodes:** An empty text field.
- Max Node Depth:** An empty text field.
- Links:** An empty text field.
- Scan ...** A blue button with white text.

**Bottom Screenshot (Results):**

- URL:** A text field containing "https://github.com/nbaertsch/JDOCS".
- Total Nodes:** A text field containing "1477".
- Max Node Depth:** A text field containing "19".
- Links:** A text field containing "91".
- Scan ...** A blue button with white text.

Resources:     [Jsoup API](#)