Node.js Documentation

Overview

The platform of Node.js was chosen to implement the HTML parser due the the support provided in the JavaScript programming language, specifically in terms of plugins, such as jQuery, of manipulating and interacting with the HTML Document Object Model (DOM). In addition, it was foreseen that this parsing may require several simultaneous HTTP requests to be made, such as requesting multiple successive pages of Amazon reviews, and that the asynchronous nature of JavaScript would lend itself nicely to making such requests, without having to implement anything unnecessarily complicated, such as multithreading.

Installation

To set up the Node.js system

- install node.js
- cd to the .../node directory
- run *npm install*

Testing and Development

In order to run the unit tests for the system, run *npm test* (this simply runs .../test/test-parser.js). The result of the tests, in terms of passes/fails should be printed to the console.

For development, it is often easier to pass in a JSON object via a command line argument, and receive the result printed to the console, such as via running
node frame-parser.js test
"{\"site\":\"amazon\",\"url\":\"http://www.amazon.com/reviews/iframe?akid=AKIAI4LLUAWZMGNUW5NA^&alinkCode=xm2^&asin=1853260002^&atag=drupal0a-20^&exp=2015-02-21T21%3A56%3A05Z^&v=2^&sig=mgKHH%2BL51tvsbjN3Xg9YB3YOcXMiiZNJrkVkEp5SFHM%3D\",\"maxReviews\":3,\"category\":\"book\"}"

Interfacing

In order to interface with the frame parser, you must determine the values of the necessary parameters. All of these parameters must be provided and of the correct form in order for valid results to be returned. These parameters are:

- site
  - the website name on which to search for reviews
  - 'amazon' is the supported value (however the system has been designed to enable easy integration of alternatives)
- url
  - the url of the reviews iframe
- maxReviews
  - the maximum number of reviews to return
  - if there are fewer than this number of reviews available for the specified product, then all these reviews will be returned
- category
  - the category of the product to search for
  - 'book' is the supported value (however the system has been designed to enable easy integration of alternatives)

There are three possible means of interfacing:

- via a command line argument (described in 'Testing and Development' above)
- via standard in and standard out
  - simply create a node process running frame-parser.js
  - send a JSON object indicating your request via standard in
  - the result will be sent, as a JSON object, via standard out
- via the RESTful HTTP web API
  - create a node process running frame-parser.js passing 'web' as the first command line argument (pass the port as the second argument if you wish, but this will default to 3000)
  - send your request as a HTTP GET reqeust to the web address
    - http://[server address]:3000/frame-parser?[url-encoded parameters]
  - the body of the result will be a JSON object representing the reviews requested

Files and Directories

Each file comes with a short description at the top which indicates what that script does.

frame-parser.js

This file is the main script which represents the frame parser application. Run this script, and interact with it using one of the interfacing approaches described above, in order to parse reviews held in iframes.

tools/

Files in this directory are used by the main application, but are intended to be distinct from it. Some of these scripts should be capable of being adapted to run in completely separate applications, achieving similar goals.

parse-frame/

Files in this directory are specific to this project and make up the main application which is used for parsing iframes. These files will often make use of scripts which are found in the 'tools' directory, which should be capable of being replace by scripts which do the same thing via a completely different implementation (if one exists).

parse-frame/site-parsers/

The files in this directory are used to parse the reviews which are found on specific websites. This is specifically Amazon, however it has been written so that it could easily be adapted to work with any other website, if this was found to be necessary.

test/

This directory holds all of the scripts relating to the unit testing of the application, including the tools and the parser-specific code.

test/test-parser.js

This is the main unit test runner. Run this script in order to run all of the unit tests and see the results of their success.

errors/

This directory holds scripts which define the errors which may be generated by the application (both the tools, and the main application). The errors would likely need to be updated if any of the tools were to be integrated into a separate project.