Golang Programming

Methods. Interfaces. Goroutines and Channels

Homework 3: Web Site Technologies Fingerprints

Implement a concurrent web site crawling engine using goroutines and channels that is capable of following functionality:

1. Starting from provided web page URL/s (given as command line arguments), crawl web pages recursively, extracting all connected web resource links, and crawling that linked resources in separate goroutines. A command argument flag should be given whether to follow external links (different from cdns serving online versions of JS and CSS libraries).
2. The communication between goroutines should be implemented using channels.
3. * The maximum number of goroutines created any given moment should not exceed the predefined constant in the program.
4. For each web-site resource crawled the program should extract information about used technologies using specific patterns provided in following JSON file borrowed from project wappalyzer: https://github.com/aliasio/wappalyzer/blob/master/src/technologies.json You can choose to implement matching for only a subset of web technologies fingerprint patterns that are most wide-spread according to your opinion, but not less than for 30 technologies.
5. The final results from crawling and extracting technologies fingerprints (patterns) from the web-site resources should be aggregated (reduced) to a single fingerprint for each web site root URL provided from command line, printed to the console, and saved to an external file named fingerprints.json.
6. You should implement crawling engine yourself and NOT use available implementations such as: https://github.com/gocolly/colly, and https://github.com/altsab/gowap