create a Spider Service (aka crawler).

The system should have the following components:

1. Front-end

a. A simple form for entering scrape requests which will include at least the basic params for a scrape job:

i. startUrl - the url to start scraping from

ii. maxDepth - the maximum depth to crawl down to from the start url

iii. maxTotalPages - the max number of pages for the entire scrape job

b. A display of the actual jobs with as much relevant information as you think would be helpful.

2. Back-end (Node.js. ES6 syntax required)

a. An api service for the front end to communicate with. It should implement either REST over http or a well structured api over websockets.

b. A crawler service that handles the actual crawling, with the following specs:

i. Use the BFS algorithm. BFS here means that all items of depth N finish processing before any items from depth N+1 start.

ii. Stop crawling a job when it reaches maxDepth or maxPages whichever comes first.

iii. For each page, save at least:

1. title - The document.title of the page

2. depth - Current depth being scraped

3. url - The URL that was scraped

4. links - All hrefs in the anchor tags in the page

- **Build for horizontal scale**. Pure in-memory solutions will not be accepted.
- **Make the UI update in real time**.