Umsetzung Crawler

Erster Ansatz:

Crawler ruft mittels http request die gewünschte Seite auf und speichert den zurückgelieferten Inhalt.

Technische Umsetzung

JavaScript wurde zur Umsetzung benutz und NodeJS um es lokal auszuführen und zu testen.

Benutzte Module: fs, request, request-promise, cheerio

- fs: internes Modul und stellt das File System da. Wird benötigt, um Dateien zu schreiben und zu lesen.
- request: ehemals internes Modul und wird benötigt um Anfragen, sog. Requests an das Internet bzw. an die betreffende Website zu stellen
- request-promise: Wrapper Modul für request.
- Cheerio: externes Modul. Wird benutzt, um direkt das zurückgelieferte Ergebnis zu filtern.

Ergebnis:

Nur die Artikel Hauptseite kann auf diese Art zurück geliefert werden.

Nicht über den Crawler abrufbar:

- WikiData Seiten
- Linkliste was auf den Artikel verlinkt

Linkliste auf was der Artikel verlinkt ist nur mit schwer aus dem zurückgelieferten html extrahierbar. Noch nicht geprüft: Seiteninformation (extra Seite in Wikipedia. Aufrufbar durch den Reiter auf der linken Seite)

Zweiter Ansatz:

Wikipedia stellt eine komplexe API zu Verfügung. Schwierigkeit hierbei liegt es die dementsprechende API Aufrufe zu modellieren. Ein tatsächliches Crawlen findet so nicht mehr großflächig statt.

Technische Umsetzung

JavaScript wurde zur Umsetzung benutz und NodeJS um es lokal auszuführen und zu testen.

Benutzte Module: fs, request, request-promise

- fs: internes Modul und stellt das File System da. Wird benötigt, um Dateien zu schreiben und zu lesen.
- request: ehemals internes Modul und wird benötigt um Anfragen, sog. Requests an das Internet bzw. an die betreffende Website zu stellen
- request-promise: Wrapper Modul für request.

Preprocessor.js normalisiert die User Eingabe über die enhanceName() Funktion. Diese wird dann an apiCalls.js weiter gegeben woraus diese dann die entsprechenden API URLS generiert. Und zum

Aufruf an crawler.js weitergegeben wird. Die API verfügt über Schnittstellen zu JSON also auch zu XML. Die Antwort auf die request mit den dementsprechenden API aufrufen. Die zurückgelieferten daten werden an ./../toRefinery abgelegt wo sie dann zur Abholung für die DataRefinery bereit stehen. Das aufbereiten und der Datenbank zu Verfügung zu stellen übernimmt die DataRefinery.

Benutzte API Calls:

- API Call Aufrufs zahlen der Wikipedia Seite
- API Call WikiData ID
- API Call Was verlinkt der Artikel
- API Call Was wurde auf den Artikel verlinkt

Beispielhafter API Aufruf anhand von Bill Gates Wikipedia Seite

```
https://wikimedia.org/api/rest_v1/metrics/pageviews/per-
article/en.wikipedia/all-access/all-agents/Bill_Gates/daily/20151010/20181012
```

API Call liefert die täglichen Seitenaufrufe in einem Zeitraum von 10.10.2015-12.10.2018 im JSON Format

Um eine Flut an Daten einzudämmen wurden die zurückgelieferten Ergebnisse nur auf eigene Artikel eingeschränkt.