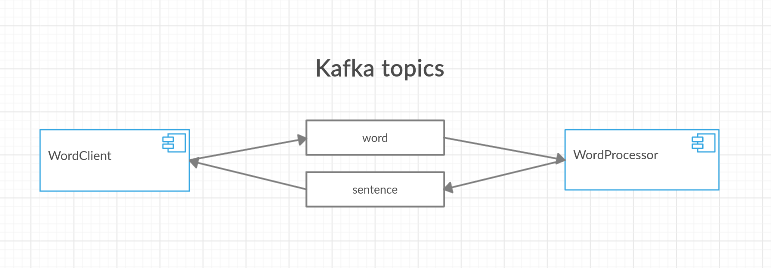
**Services and Kafka topics**



**WordClient** sends messages to **word** topic and consumes messages from **sentence** topic.

**WordProcessor** consumes messages from **word** topic and sends messages to **sentence** topic.

**Word client**

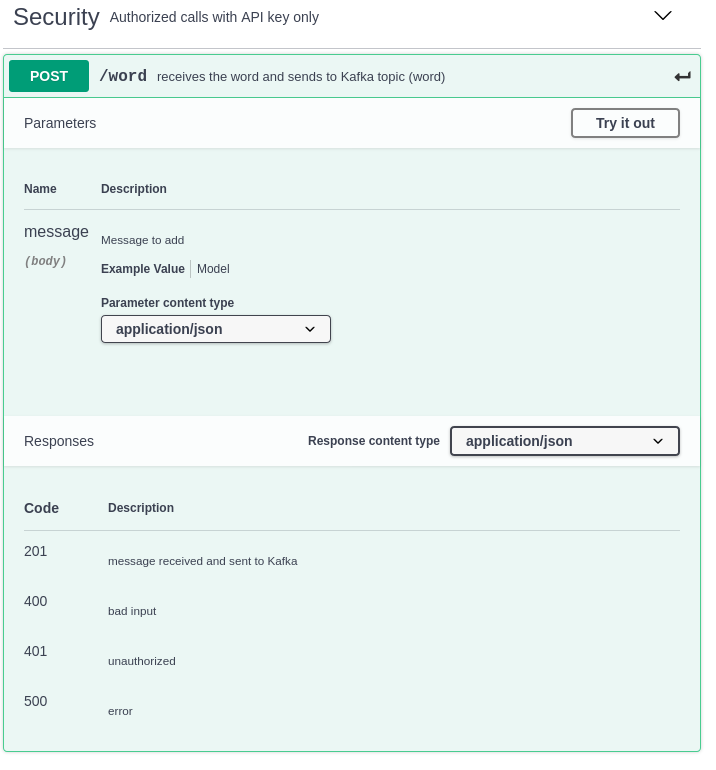
Word client receives the words through the REST API calls and sends them to Kafka topic (word).

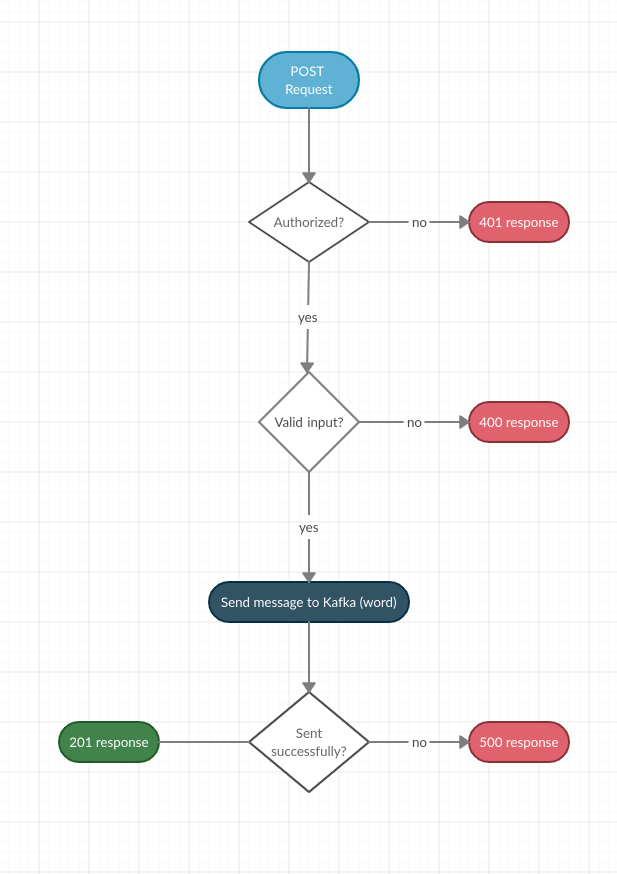
It concumes the messages from Kafka topic (sentence) and stores them in the cassandra database.

Calls to word client should be secured with API key.

Word client exposes to following endpoints:

**/word**

****

****

Example curl request:

curl --location --request POST 'http://localhost:8080/word' \

--header 'api-key: 72A91A6CECF5EFA948118EF2F0EE297F' \

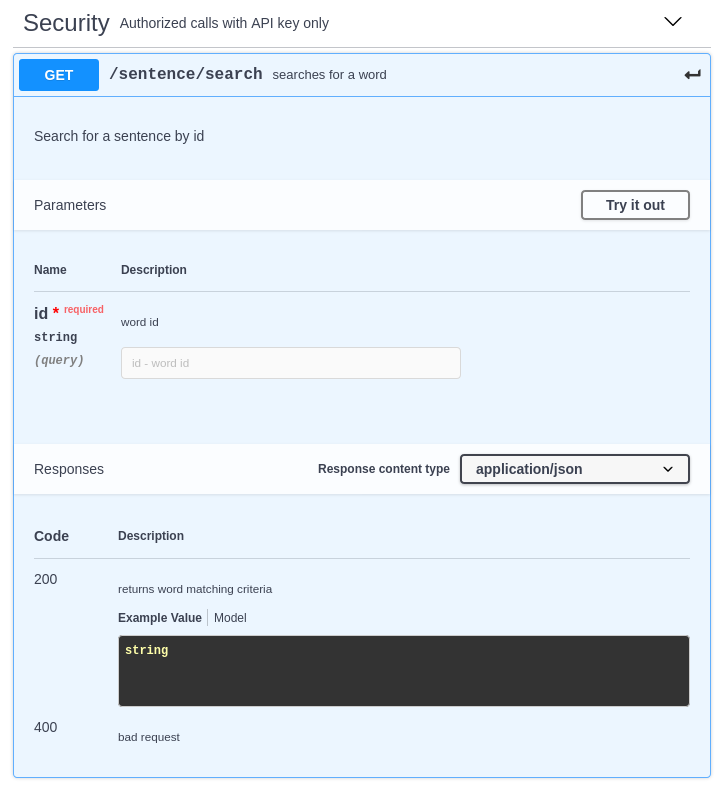
--header 'Content-Type: application/json' \

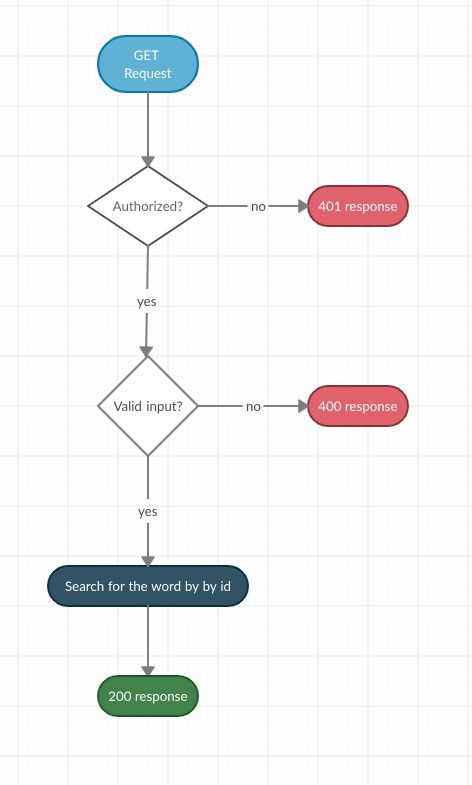
--data-raw '{

"body": "word1"

}'

**/sentence/search**

****

****

Example curl request:

curl --location --request GET 'http://localhost:8080/sentence/search?id=0930fcce-068f-40f3-88b6-e06fdb163bdb' \

--header 'api-key: 72A91A6CECF5EFA948118EF2F0EE297F' \

--data-raw ''

**Word processor**

Word processor consumes the words from Kafka topic (word), concatenates them into sentences (all words for every minute should be joined together) and sends the sentences to Kafka topic (sentence).

Storage should append the words received in the same minute to a sentence (use current minute as an id).

Scheduler (runs every minute) should send sentences from the storage to Kafka topic (sentence) and clear from from the storage.

**Cron job schedule**

0 0/1 \* \* \* ?

**Database**

**Cassandra table creation**

CREATE TABLE sentence(id varchar PRIMARY KEY, value text );

**Kafka**

Topics:

* word
* sentence

**TODO/improve**

* Add unit tests
* Improve the security – use more complex authentication instead of API key
* Store received words in a permanent storage instead of memory to avoid possible data loss if the server goes down
* More Kafka configs to a config server