Lightning Talk

Querqy/SMUI x ElasticSearch

Dominic Bestler, Product Manager Search

MICES 2021

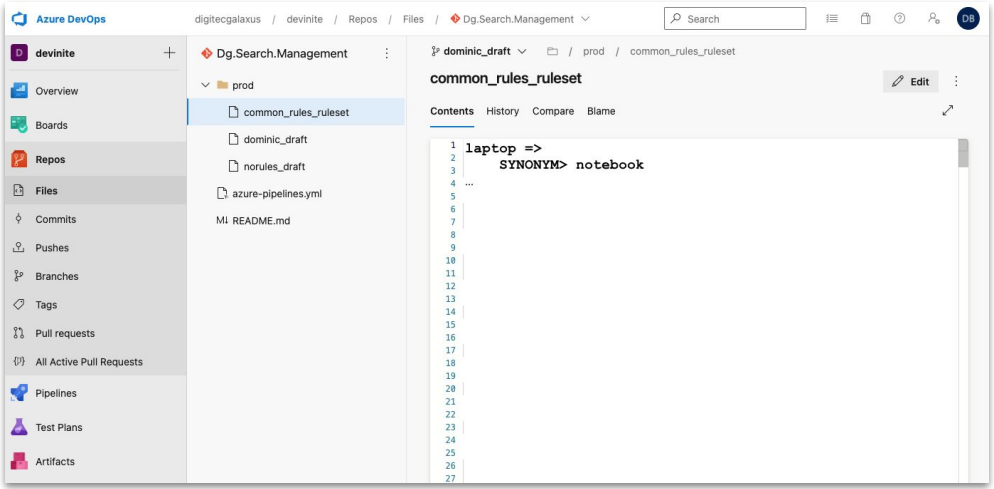
How are we using Querqy with ES?

Use case:

So far mainly for Search management using Common Rules Rewriter

Rules Management: Walking skeleton

- Rules.txt edited as text and committed to Azure Git (via Azure DevOps web interface for Git Repositories)
- Upon Commit an automatic validation & upload pipeline is triggered on commits and merges into master



Quickly testing a new rule in production

1. Create rule to be tested in a separate “draft rules.txt” and upload it to a separate common rules “draft rewriter”.

2. Test the rule directly in the productive search simply by activating the “draft rewriter”
   e.g. via URL Paramater q=dining table&**draftrules=mydraftrules**

No need for a separate Test Environment that is struggling hard to be like PROD

With this approach, we can as well easily see a SERP without any rules applied by configuring a dummy “no-rules.txt” draft.

Monitoring “rule managed” searches

Problem:

Querqy@ES can not tell the caller whether or not a rule has been applied (unlike SOLR)

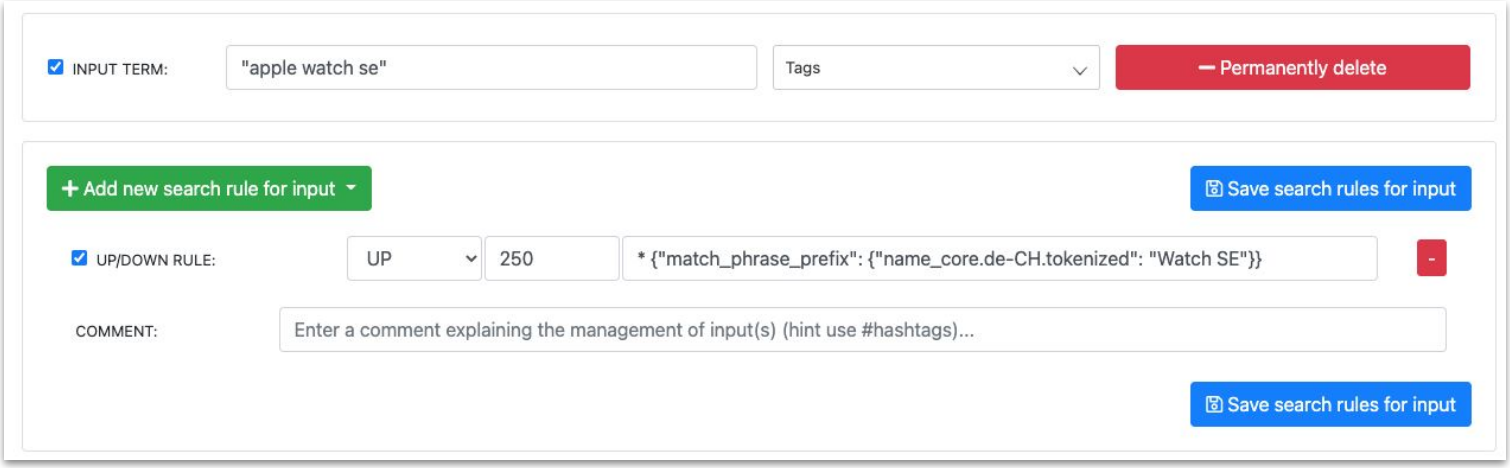
Solution:

- Using the newly supported Info Logging Feature for ES: https://docs.querqy.org/querqy/solr-plugin-configuration.html#info-logging
- We chose to log to ES System Log and use Logstash to push these logs to Google BQ. From where we generate our monitoring reports

Outlook Rule Management: Migrate to SMUI

But is ElasticSearch supported? Officially no. But, it actually does.

Goal: Make SMUI accept Raw ES Queries & use custom deploy scripts to upload to ES



Outlook Rule Management: Migrate to SMUI

How to do it?

- Make SMUI show a “raw query” field where we can put a raw ES Query
- Add a custom deployment script to upload the rules to Querqy@ES

```
toggle.ui-concept.all-rules.with-solr-fields=false
```

```
toggle.rule-deployment.custom-script=true
toggle.rule-deployment.custom-script-SMUI2SOLR-SH_PATH
="my-deployment-script.sh"
```

Tradeoff: **Higher risk of human error** when writing raw query syntax unless there is ES specific validation added to these inputs.

→ https://github.com/querqy/smui/issues/56

→ https://github.com/querqy/smui/issues/43

Testing raw query validation

Since SMUI does not validate raw ES Query Syntax. What happens if we deploy rules with invalid raw queries?

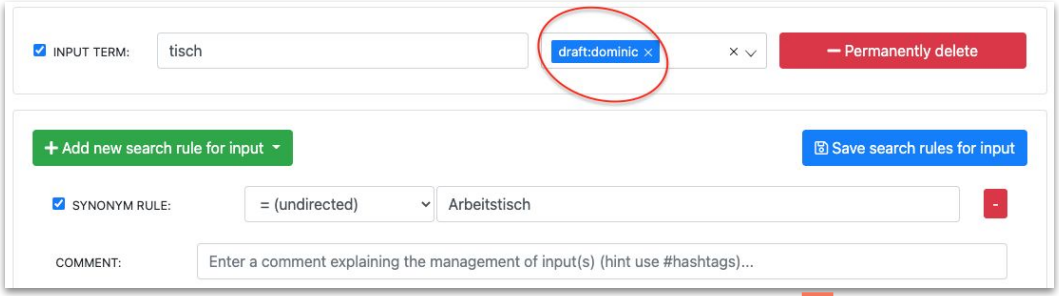
Test Raw Query Validation

* {"term":{"producttype_id": 2165}}	STATUS
JSON Format is invalid	OK only matching queryy query fails with "syntax_exception" Error parsing raw query.
ES Keywords wrong (e.g. trm instead of term)	OK only matching queryy query fails with "parsing_exception" unknown query [trm] did you mean any of [term, terms]?
Unknown fields used (e.g. PT_ID instead of producttype_id	OK only matching queryy query fails with "failed to create query: Rewrite first"
String is used for Term Query	OK only matching queryy query fails with "failed to create query: For input string: \"ab\""
Illegal character is used in Match Query	?? can't enforce a problem -- https://lucene.apache.org/core/3_0_3/queryparsersyntax.html#Escaping%20Special%20Characters

Seems to be unproblematic so far. But clearly the rules are not applied which is only detected at runtime in the elasticsearch logs

Outlook: Draft-Testing concept with SMUI

- Map the testing “drafts” concept to SMUI using Tags which resolve in @-properties in rules.txt.




```
tisch =>
        SYNONYM: Arbeitstisch
        @{
          "draft" : [ "dominic" ]
        }@

Arbeitstisch =>
        SYNONYM: tisch
        @{
          "draft" : [ "dominic" ]
        }@
```

- Upload the whole ruleset to a to a draft rewriter which is configured to only apply “@draft” annotated rules. Using JSON Path.

```
{
  "query": {
      "querqy": {
          "rewriters": [
              {
                  "name": "draft_rules",
                  "params": {
                      "criteria": {
                          "filter": "$[?( @.draft || \"dominic\" in @.draft) ]"
                      }
                  }
              }
          ]
      }
  }
}
```