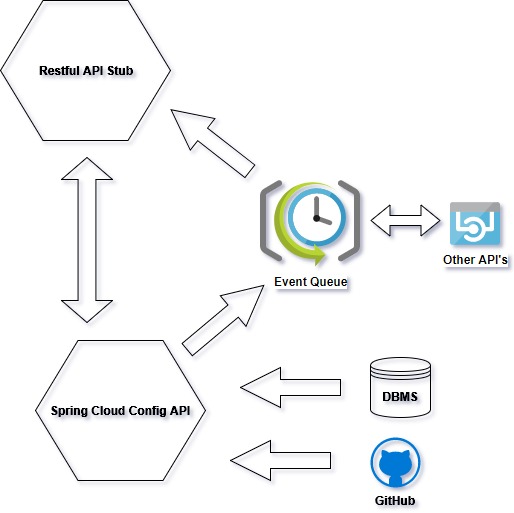
# Using Spring Cloud Config to consolidate and serve runtime configuration from different sources simultaneously.

# Scope

* Find out if it is possible to reliably serve configuration from git and DBMS system simultaneously.
* Check if it is possible to Spring Cloud Config server in combination with Event Queue.
* Implement the system with different DBMS and evaluate them.
* Check if it is possible to store and serve configuration data in JSON format, explore limitations as there might be some in relational DBMS.

# Design



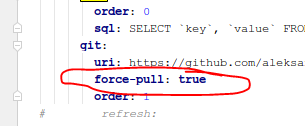
# Investigation

## GitHub with Spring Cloud Config Server

<https://github.com/aleksandrskrivickis/spring_cloud_config_research_config>

It is served in a way which is different from JDBC. For every label on GitHub user has to create separate branch.

Also, there is a feature **force-pull,** that allows to pull git repo and serve fresh config data on-demand. Theoretically there is no need to create a callback/hook from GitHub.



**It is also possible to fetch config from multiple repositories:**

spring:

cloud:

config:

server:

git:

uri: https://git/common/config-repo.git

force-pull: **true**

repos:

team-a:

pattern: team-a-\*

uri: http://git/team-a/config-repo.git

force-pull: **true**

team-b:

pattern: team-b-\*

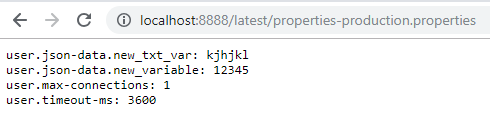
uri: http://git/team-b/config-repo.git

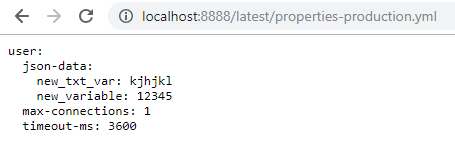
force-pull: **true**

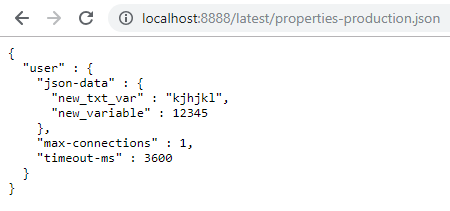
team-c:

pattern: team-c-\*

uri: http://git/team-a/config-repo.git



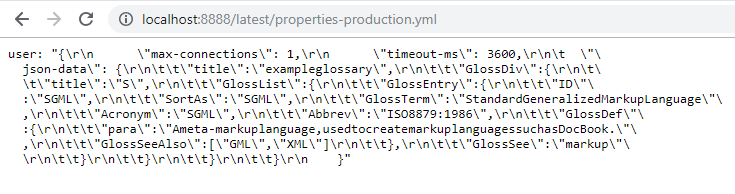


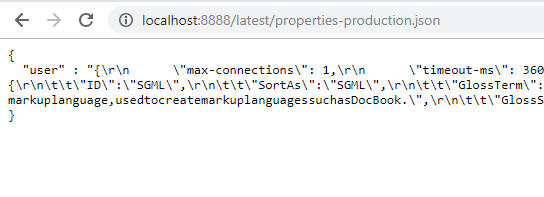


## MySQL with Spring Cloud Config Server

Same limitations and structure of response as with MariaDB. Different driver has to be used for establishing connection.







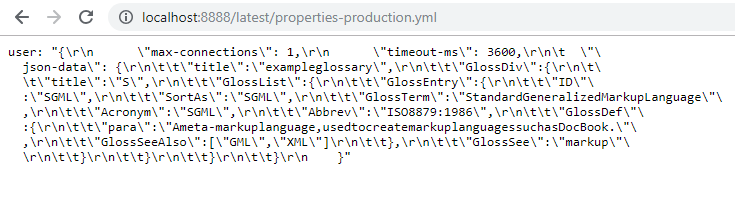
## MySQL(MariaDB) with Spring Cloud Config Server

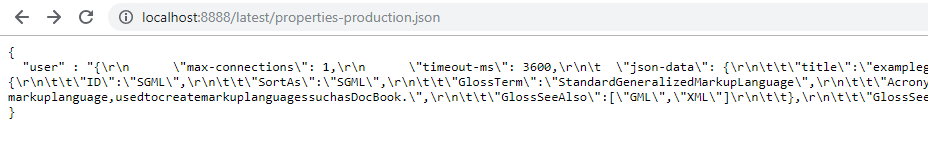
It is possible to serve MongoDB database in three different formats: properties, yml/yaml, json. It is possible to use exactly the same paths as in MongoDB. Unfortunately, I was unable to make the server display **indents** in the same way as with MongoDB but assume it does not make a difference as Client will ignore \n \t \r tags. **Must investigate further if it makes a difference in Spring Cloud Config Client.**

**Limitations**:

The effective maximum length of a **VARCHAR** is subject to the maximum row size (**65,535** bytes, which is shared among all columns). TINYTEXT is a string data type that can store up to to 255 characters. TEXT is a string data type that can store up to 65,535 characters. TEXT is commonly used for brief articles. **LONGTEXT** is a string data type with a maximum length of **4,294,967,295** **characters**.

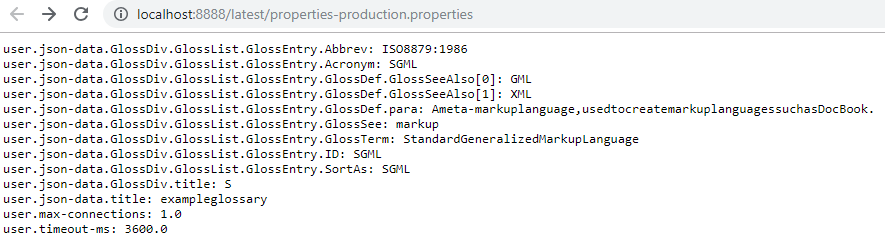


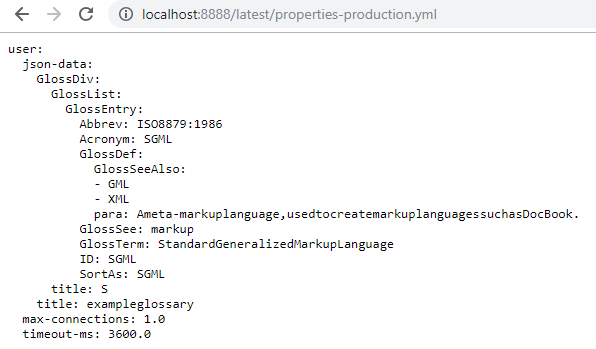




## MongoDB with Spring Cloud Config Server

It is possible to serve MongoDB database in three different formats: properties, yml/yaml, json. In order to display json with indents property “**spring.jackson.serialization.indent\_output=true**” has to be set.







## GitHub and MySQL with Spring Cloud Config Server

## GitHub and MySQL with Spring Cloud Config served through Event Queue

## Investigation summary

# Limitations

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Feature** | **MySQL** | **MySQL(MariaDB)** | **MongoDB** | **GitHub** |
| Value column character length | 4,294,967,295 | 4,294,967,295 | N/A | N/A |
| Max document size | N/A | N/A | 16 MB | 100 MB+ |
| Can use “-“ in database names | No | No | Yes | Yes |
| Serves JSON | Yes | Yes | Yes | Yes |

# Conclusions

# References

* <https://github.com/spring-cloud-incubator/spring-cloud-config-server-mongodb>
* <https://medium.com/@nani2ratna/spring-cloud-config-server-with-jdbc-55de8f7ec86d>
* <https://cloud.spring.io/spring-cloud-static/Edgware.SR2/single/spring-cloud.html#_spring_cloud_config>
* <https://dev.mysql.com/doc/refman/8.0/en/data-types.html>
* <https://start.spring.io/>