RASQL - REST API for SQL

Viverae
November 29, 2016
Dallas, Texas, USA

John Scott
CTO of RJ2 Technologies

REST API for SQL (rasql)

Transform SQL Queries in Files to REST Queries via HTTP Protocol

curl --user id:passwd http://localhost:8080/<schema>/<query-file>?a1=abc&a2=xyz

```
$ curl --user jmscott:lose20pd \
        http://localhost:8080/pg_catalog/pg_stat_activity-state-count?state=active
[
    "duration,columns,rows",
    0.001444535,
    ["activity_count"],

    [
      [1]
    ]
]
```

Find Available REST Queries

```
$ curl --user jmscott:lose20pd https://localhost:8080/pg_catalog
[
    "duration,colums,rows",
    0.0,
    ["path","synopsis","description"],

    [
["pg_class-by-nspname","Fetch all classes in a particular name space",""],
["pg_stat_activity","Select all tuples from pg_stat_activity",""],
["pg_roles","Fetch all roles in the pg_roles view",""],
["pg_stat_activity-slow-count","Select count of queries slower than a certain duration.",""],
["pg_stat_activity-count","Select all queries in table pg_stat_activity.",""],
["pg_stat_activity-state-count","Select count of queries in a certain state.",""],
["pg_class","Fetch all classes in a pg_catalog.pg_class",""]
    ]
]
```

Specification

- Build REST Queries from SQL Queries in Files
- Simple Configuration File
- Validate URL Query Arguments
- Encryption via Secure Sockets.
- Basic Authentification
- Self Documenting from SQL File
- Export Results in JSON, CSV and TSV
- Log Slow Queries
- Pool SQL Connections for SSL

Only 1500 Lines of Go Code!

Parse Preamble in SQL File

```
/*
 *  Synopsis:
 *    Select count of queries slower than a certain duration.
 *
 *  Command Line Variables:
 *    duration text
 *
 *  Usage:
 *    psql --file pg_stat_activity-slow-count.sql --set duration="'-1min'"
 */

SELECT
    count(*) as slow_count
  FROM
      pg_catalog.pg_stat_activity
  WHERE
      query_start <= (now() + :duration)
;
```

https://github.com/jmscott/work/blob/master/rasql/pg_stat_activity-slow-count.sql

Configuration File in JSON (abbreviated)

```
{
    "synopsis":        "PostgreSQL 9.6 System Catalog Schema",
    "http-listen":        "localhost:8080",
    "basic-auth-path":    "password-rasql.example",
    "tls-http-listen":    "localhost:443",
    "tls-cert-path":    "self-signed.cert",
    "tls-key-path":        "self-signed.key",
    "rest-path-prefix":    "/pg_catalog",
    "sql-query-set": {
        "pg_stat_activity-slow-count":    {
            "source-path":    "pg_stat_activity-slow-count.sql"
        }
    },
    "http-query-arg-set": {
        "dur" : {
            "matches": "^.{1,63}$",
            "sql-alias": "duration"
        }
    },
    "warn-slow-sql-query-duration": 5
}
```

https://github.com/jmscott/work/blob/master/rasql/pg_catalog.rasql.example

All SQL Queries in pg_catalog.rasql.example

```
"sql-query-set": {
    "pg_class": {
        "source-path":    "pg_class.sql"
    },
    "pg_class-by-nspname":    {
        "source-path":    "pg_class-by-nspname.sql"
    },
    "pg_stat_activity":    {
        "source-path":    "pg_stat_activity.sql"
    },
    "pg_roles":    {
        "source-path":    "pg_roles.sql"
    },
    "pg_stat_activity-slow-count":    {
        "source-path":    "pg_stat_activity-slow-count.sql"
    },
    "pg_stat_activity-count":    {
        "source-path":    "pg_stat_activity-count.sql"
    },
    "pg_stat_activity-state-count":    {
        "source-path":    "pg_stat_activity-state-count.sql"
    }
},
```

All HTTP Query Arguments in pg_catalog.rasql.example

```
"http-query-arg-set": {
    "name" : {
        "matches": "^.{1,63}$",
        "sql-alias": "nspname"
    },
    "dur" : {
        "matches": "^.{1,63}$",
        "sql-alias": "duration"
    },
    "state" : {
        "matches": "^[a-z]{1,32}$"
    }
},
```

https://github.com/jmscott/work/blob/master/rasql/pg_catalog.rasql.example

Regular Expressions Validate Query Arguments.
SQL Layer Also Validates.

Easy to Bind JSON Keys to Go Variables

```
type Config struct {
    source_path string

    Synopsis        string `json:"synopsis"`
    HTTPListen      string `json:"http-listen"`
    RESTPathPrefix  string `json:"rest-path-prefix"`
    SQLQuerySet     `json:"sql-query-set"`
    HTTPQueryArgSet `json:"http-query-arg-set"`

    BasicAuthPath   string `json:"basic-auth-path"`

    basic_auth map[string]string

    //  Note:  also want to log slow http requests!
    //         consider moving into WARN section.

    WarnSlowSQLQueryDuration float64 `json:"warn-slow-sql-query-duration"`

    //  https paramters

    TLSHTTPListen   string `json:"tls-http-listen"`
    TLSCertPath              string  `json:"tls-cert-path"`
    TLSKeyPath               string  `json:"tls-key-path"`
}
```

Extract Row as JSON, CSV or Tab Separated

```
$ curl --user jmscott:lose20pd curl --user jmscott:lose20pd \
        http://localhost:8080/pg_catalog/csv/pg_roles

rolname,rolsuper,rolinherit,rolcreaterole,rolcreatedb,rolcanlogin,rolreplication,rolconnlimit,rolpasswor
blobio,false,true,false,false,true,false,-1,********,,false,,16384
jmscott,false,false,false,false,true,false,-1,********,,false,,16386
omega,false,true,false,false,true,false,-1,********,,false,,16387
pg_signal_backend,false,true,false,false,false,false,-1,********,,false,,4200
pgbackup,false,true,false,false,true,true,-1,********,,false,,39214049
postgres,true,true,true,true,true,true,-1,********,,false,,10
root,false,true,false,false,false,false,-1,********,,false,,39311573
setspace,false,true,false,false,true,false,-1,********,,false,,16388
wwwuser,false,true,false,false,false,false,-1,********,,false,,39311574
```

Content-Type: text/csv

Basic Authentification is Apache Password Format

```
#
#  Synopsis:
#    Example password file for rasql basic authorization
#  Usage:
#    "basic-auth-path": "etc/passwd-rasql"
#

#  user:clear-password
#  user must be alphanumeric.  password follows colon, exactly
#  comments and empty strings are ignored

jmscott:ch0mski4told2
cassie:lose10pd
```

https://github.com/jmscott/work/blob/master/rasql/password-rasql.example

Need to authenticate with SQL query!

Adding https/SSL to Server is Trivial

```
    if cf.TLSHTTPListen != "" {
        if cf.TLSCertPath == "" {
            die("http listen tls: missing tls-cert-path")
        }
        if cf.TLSKeyPath == "" {
            die("http listen tls: missing tls-key-path")
        }
        log("tls listening: %s%s", cf.TLSHTTPListen, cf.RESTPathPrefix)
        go func() {
            err := http.ListenAndServeTLS(
                cf.TLSHTTPListen,
                cf.TLSCertPath,
                cf.TLSKeyPath,
                nil,
            )
            die("http listen tls: %s", err)
        }()
    }
```

https://github.com/jmscott/work/blob/master/rasql/rasqld.go

SSL Key Generation and Self Signing is Easy

Generate private key (.key)

```
# Key considerations for algorithm "RSA" ≥ 2048-bit

openssl genrsa -out server.key 2048

# Key considerations for algorithm "ECDSA" ≥ secp384r1
# List ECDSA the supported curves (openssl ecparam -list_curves)

openssl ecparam -genkey -name secp384r1 -out server.key
```

Generation of self-signed(x509) public key (PEM-encodings .pem|.crt) based on the private (.key)

```
openssl req -new -x509 -sha256 -key server.key -out server.pem -days 3650
```

May need insecure mode in curl during testing

```
curl --insecure http://localhost:8080...
```

Enhancements

- Use GoLang Text Templates to Conditionally Rewrite SQL queries
- Authenticate with an SQL Query or PostgreSQL Auth
- Generate HTML Documentation of SQL
- Parse SQL to Extract Target List and Arguments.
- Monitor Changes in SQL Files
- View SQL Source Code from REST Query
- Authenticate with Trusted Keys
- Syslog
- SSL SQL Connections
- HTTP Redirect of http:// to https://
- Scan Directory for SQL Query Files
- Trusted Network

Thank you

Viverae
November 29, 2016
Dallas, Texas, USA

John Scott
CTO of RJ2 Technologies

jmscott@setspace.com (mailto:jmscott@setspace.com)
jmscott@rj2tech.com (mailto:jmscott@rj2tech.com)
https://github.com/jmscott/talk (https://github.com/jmscott/talk)