

SQL OVER HTTP



A GENERIC REST API SERVICE FOR CONTROL DATABASES

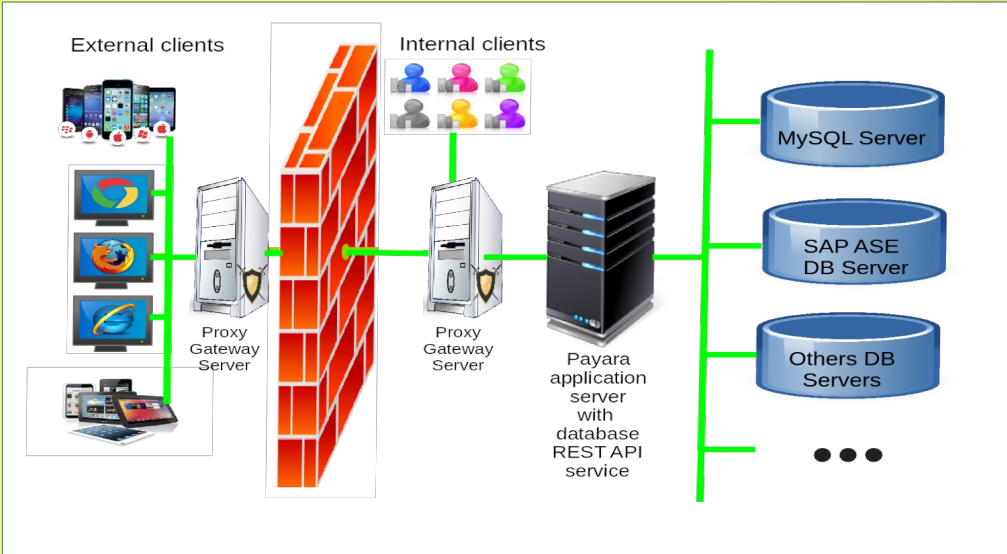
Wenge Fu, Ted D'Ottavio, Seth Nemesure
Brookhaven National Laboratory, Upton, NY 11793, USA

Background

Accessing database resources from accelerator controls servers or applications with JDBC/ODBC and other dedicated programming interfaces have been common for many years. However, availability and performance limitations of these technologies were obvious as rich web and mobile communication technologies became more mainstream. The HTTP Representational State Transfer (REST) services have become a more reliable and common way for easy accessibility for most types of data resources include databases. Commercial products for quickly setup database REST API services have become available in recent years, each with their own pros and cons. This paper presents a simple way for setting up a generic database HTTP REST API service with technology that combines the advantages of application servers (such as Glassfish/Payara), JDBC drivers, and REST API technology to make major RDBMS systems easy to access and handle data in a secure way. This allows database clients to retrieve data in standard formats such as XML or JSON.

The idea is simple

- REST API bridges the gaps between data resources and destinations over the internet with http protocol.
- SQL is a standard and powerful language for database management and CRUD operations.
- Sending SQL statements through REST APIs and get the result back. This way we combines the power of SQL and power of REST API service.



Data Security:

Data protection is a crucial part of REST API service. We do it in this way:

- Use a network firewall. Users have to have a legal system ID and password to access the REST API service.
- Database data change history logging on server side.
- Client IP filtering to ensure only the legal client IP ranges can access the API service;
- REST API layer user credential data verification which includes: User data double encoding and decoding along with secret API key and data parsing procedures shared by clients and server;
- Database server user ID and password verification; SQL statement data verification (prevent databases from SQL injection attacks); SQL access working load control (to prevent from DOS attacks etc).

The REST APIs :

This API sends request data in two ways:

- Sending DB server name, server type, database name, and SQL command directly through REST API and execute the SQL statement on server side with a proper JDBC connection;

For example:

```
http://domain:port/DBServer/api/sql?  
server=<server>&db=<db>&table=  
<table>&sql=<sql>
```

- Sending DB server name, server type, database name, table name, operation type (such as create, drop, alter, select, insert, update, delete etc.) and conditions through the REST API and build SQL statement on server side and execute it with a JDBC connection;

For example:

```
http://domain:port/DBServer/api/insert?  
Server=<server>&db=<db>&table=  
<table>&datapairs=<column-value JSON  
data pairs>
```

User credential data are sent through the HTTP Request Header.

API Data Format

REST API calls return following information:

- Results of SQL statements in XML or JSON, includes: meta data, page info, and result data set.
- HTTP standard error codes for various error situations.

Summary

REST API services provide a convenient way to make data resources of many kinds available through the HTTP protocol over the network with standard data exchange format such as XML and JSON. This bridges the gaps between different OS systems and platforms, and makes data operations especially easy through mobile and web applications. This API design takes advantage of the REST API capability and combines the power of other technology such as JDBC. With the Database REST API service, we are able to communicate with different database systems using a common method, eliminating the need to know how to connect to each database server. This makes database data retrieval and CRUD operations much easier from systems like mobile and web applications, Matlab applications, and other accelerator control applications using different programming languages. With a little more effort, this REST API can be applied to all JDBC supported database systems.

