

[文章作者: 张宴 本文版本: v1.0 最后修改: 2010.09.08 转载请注明原文链接: <http://blog.zyan.cc/mysql-udf-http/>]

Mysql-udf-http 是一款简单的MySQL用户自定义函数 (UDF, User-Defined Functions) , 具有http_get()、http_post()、http_put()、http_delete()四个函数, 可以在MySQL数据库中利用HTTP协议进行REST相关操作。

项目网址: <http://code.google.com/p/mysql-udf-http/>

中文说明: <http://blog.zyan.cc/mysql-udf-http/>

使用环境: Linux操作系统, 支持的MySQL版本: 5.1.x 和 5.5.x。5.0.x未经测试。

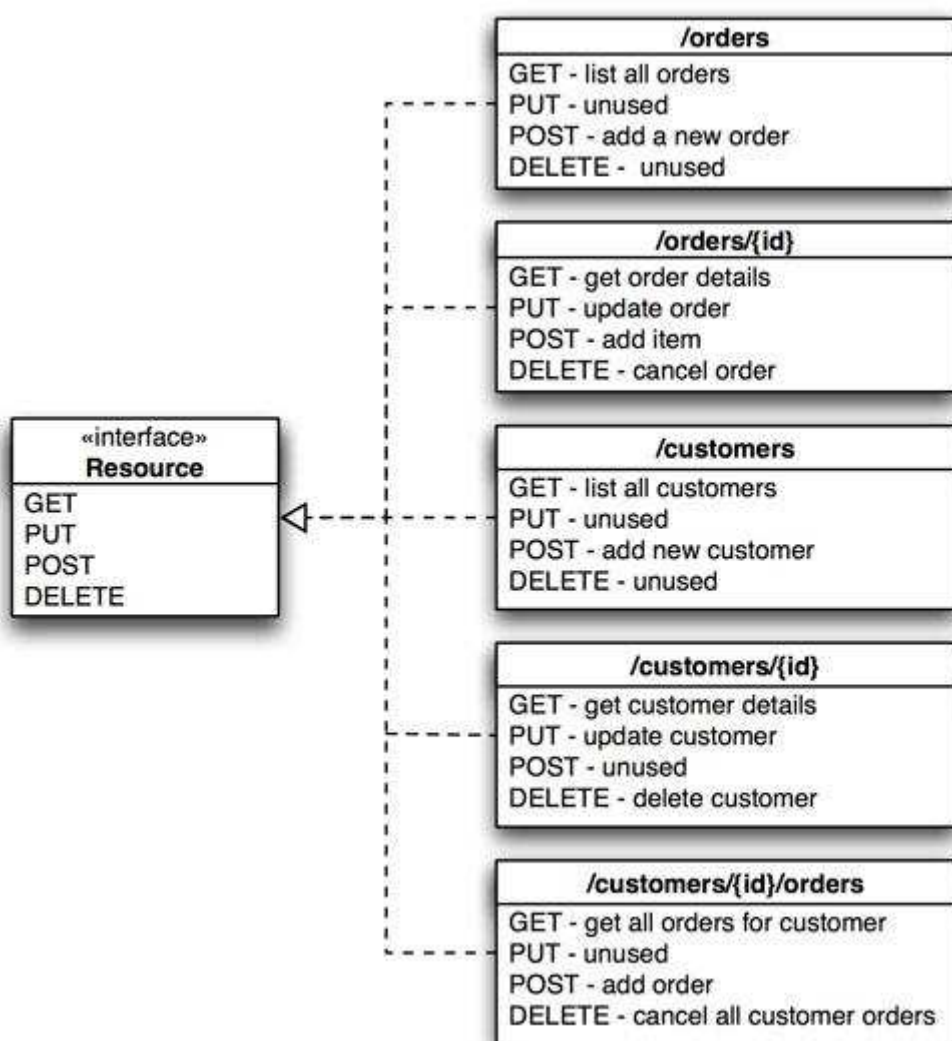
软件作者: 张宴

一、REST架构风格:

REST (Representational State Transfer) 是一种轻量级的Web Service架构风格, 其实现和操作明显比SOAP和XML-RPC更为简洁, 可以完全通过HTTP协议实现, 还可以利用缓存Cache来提高响应速度, 性能、效率和易用性上都优于SOAP协议。REST最早是由 Roy Thomas Fielding 博士2000年在论文《Architectural Styles and the Design of Network-based Software Architectures》中提出的, 中文译文全文PDF点此下载。另外, 有篇译文对REST做了一个简化说明。

目前, REST架构风格的常见实现是基于HTTP协议及其四种基本方法 (如POST、GET、PUT和DELETE) 的。有人将HTTP协议的四种方法与CRUD原则相对应, CRUD原则对于资源只需要四种行为: Create (创建)、Read (读取)、Update (更新) 和Delete (删除) 就可以完成对其操作和处理。

HTTP 方法	CRUD 方法	描述
POST	CREATE	创建一个资源
GET	READ	读取一个资源
PUT	UPDATE	更新一个资源
DELETE	DELETE	删除一个资源



在Mysql-udf-http中，四个函数http_post()、http_get()、http_put()、http_delete()分别对应HTTP协议的POST、GET、PUT、DELETE四种方法。

REST是一种架构风格，而不是协议或标准。HTTP协议“POST、GET、PUT、DELETE”四种方法与CRUD原则“Create、Read、Update、Delete”四种行为的——对应关系只是一种架构设计习惯，而不是规范。因此，POST方法也可以用来更新资源，PUT方法也可以用来创建资源，这就要看具体应用程序作者的定义了。例如Tokyo Tyrant除了支持Memcached协议外，还支持REST方式存取，PUT代表创建和更新，GET代表读取，DELETE代表删除（关于Tokyo Tyrant的安装使用请点击这儿）。

目前国内外流行的Web 2.0应用API接口中，很多都支持REST架构风格。例如：[新浪微博开放平台](#)、[人人网API](#)、[Google OpenID](#)、[Flickr](#)、[Twitter](#)、[eBay](#)、[Facebook](#)、[Last.fm](#)、[del.icio.us](#)、[Yahoo Search](#)、[Amazon S3](#)、[Amazon EC2](#)、[Digg](#)、[Microsoft Bing](#)、[FriendFeed](#)、[PayPal](#)、[Foursquare](#)，[更多...](#)

当记录数成百上千万条时，通常采用MySQL分表减低数据库压力。但是，全部数据按点击数、精华、积分排序显示等功能，在MySQL分表中则无法实现。编写Mysql-udf-http的最初目的，是为了在项目开发中，将MySQL各分表的数据自动同步到我们的TCSQL高速列表数据库，用来做列表查询、显示，内容页则根据ID直接查询各MySQL分表的内容。由于HTTP协议的通用性，通过Mysql-udf-http可以做更多的事情。

通过Mysql-udf-http，你可以在MySQL中利用触发器，将MySQL的数据同步到支持REST的应用上。例如你有一个独立博客，你可以在文章表创建MySQL触发器，这样，在发表文章时，就可以将文章

二、Mysql-udf-http的安装与使用：

1. 在Linux系统上安装Mysql-udf-http

注意：“/usr/local/webserver/mysql/”是你的MySQL安装路径，如果你的MySQL安装路径不同，请自行修改。

```
ulimit -SHn 65535
wget http://curl.haxx.se/download/curl-7.21.1.tar.gz
tar zxvf curl-7.21.1.tar.gz
cd curl-7.21.1/
./configure --prefix=/usr
make && make install
cd ../

echo "/usr/local/webserver/mysql/lib/mysql/" > /etc/ld.so.conf.d/mysql.conf
/sbin/ldconfig
wget http://mysql-udf-http.googlecode.com/files/mysql-udf-http-1.0.tar.gz
tar zxvf mysql-udf-http-1.0.tar.gz
cd mysql-udf-http-1.0/
./configure --prefix=/usr/local/webserver/mysql --with-mysql=/usr/local/webserver/mysql/bin/mysql_config
make && make install
cd ../
```

2. 通过命令行登陆进入MySQL

```
/usr/local/webserver/mysql/bin/mysql -S /tmp/mysql.sock
```

3. 创建MySQL自定义函数

mysql>

```
view plain print ?
1. create function http_get returns string soname 'mysql-udf-http.so';
2. create function http_post returns string soname 'mysql-udf-http.so';
3. create function http_put returns string soname 'mysql-udf-http.so';
4. create function http_delete returns string soname 'mysql-udf-http.so';
```

4. 使用方法

I. 函数描述:

mysql>

view plain print ?

```
1. SELECT http_get('<url>');
2. SELECT http_post('<url>', '<data>');
3. SELECT http_put('<url>', '<data>');
4. SELECT http_delete('<url>');
```

II. 示例 A:

mysql>

view plain print ?

```
1. /* HTTP GET、POST方式提交关键词“xoyo”到百度移动搜索 */
2. SELECT http_get('http://m.baidu.com/s?word=xoyo&pn=0');
3. SELECT http_post('http://m.baidu.com/s', 'word=xoyo&pn=0');
4.
5. /* 新浪微博开放平台: 获取新浪用户ID为103500的最近一条微博内容 */
6. SELECT http_get('http://api.t.sina.com.cn/statuses/user_timeline/103500.json?count=1&source=1561596835') AS data;
7. /* 新浪微博开放平台: 发表一条微博 */
8. SELECT http_post('http://your_sina_uid:your_password@api.t.sina.com.cn/statuses/update.xml?
   source=1561596835', 'status=Thins is sina weibo test information');
9.
10. /* Tokyo Tyrant 写入、读取、删除操作 */
11. SELECT http_put('http://192.168.8.34:1978/key', 'This is value');
12. SELECT http_get('http://192.168.8.34:1978/key');
13. SELECT http_delete('http://192.168.8.34:1978/key');
```

III. 示例

通过MySQL触发器，利用mysql-udf-http和第三方UDF函数lib_mysqludf_json，自动同步数据到 Tokyo Tyrant。

(1). 下载安装 lib_mysqludf_json 修改版:

以下安装包适合32位Linux操作系统:

```
wget http://mysql-udf-http.googlecode.com/files/lib_mysqludf_json-i386.tar.gz
tar zxvf lib_mysqludf_json-i386.tar.gz
cd lib_mysqludf_json-i386/
# 如果你的MySQL安装路径不是/usr/local/webserver/mysql/, 请修改以下路径。
cp -f lib_mysqludf_json.so /usr/local/webserver/mysql/lib/mysql/plugin/lib_mysqludf_json.so
cd ../
```

以下安装包适合64位Linux操作系统:

```
wget http://mysql-udf-http.googlecode.com/files/lib_mysqludf_json-x86_64.tar.gz
tar zxvf lib_mysqludf_json-x86_64.tar.gz
cd lib_mysqludf_json-x86_64/
```

```
# 如果你的MySQL安装路径不是/usr/local/webserver/mysql/, 请修改以下路径。
cp -f lib_mysqludf_json.so /usr/local/webserver/mysql/lib/mysql/plugin/lib_mysqludf_json.so
cd ../
```

通过命令行登陆进入MySQL:

```
/usr/local/webserver/mysql/bin/mysql -S /tmp/mysql.sock
```

mysql>

view plain print ?

```
1. create function lib_mysqludf_json_info returns string soname 'lib_mysqludf_json.so';
2. create function json_array returns string soname 'lib_mysqludf_json.so';
3. create function json_members returns string soname 'lib_mysqludf_json.so';
4. create function json_object returns string soname 'lib_mysqludf_json.so';
5. create function json_values returns string soname 'lib_mysqludf_json.so';
```

lib_mysqludf_json的详细用法请访问: http://www.mysqludf.org/lib_mysqludf_json/

(2). 创建测试表

mysql>

view plain print ?

```
1. SET NAMES UTF8;
2. USE test;
3. CREATE TABLE IF NOT EXISTS `mytable` (
4.   `id` int(10) NOT NULL AUTO_INCREMENT,
5.   `addtime` int(10) NOT NULL,
6.   `title` varchar(255) CHARACTER SET utf8 NOT NULL,
7.   PRIMARY KEY (`id`)
8. ) ENGINE=MyISAM DEFAULT CHARSET=utf8 AUTO_INCREMENT=1;
```

(3). 为测试表创建触发器:

mysql>

view plain print ?

```
1. /* INSERT插入操作的触发器 */
2. DELIMITER |
3. DROP TRIGGER IF EXISTS mytable_insert;
4. CREATE TRIGGER mytable_insert
5. AFTER INSERT ON mytable
6. FOR EACH ROW BEGIN
7.   SET @tt_json = (SELECT json_object(id,addtime,title) FROM mytable WHERE id = NEW.id LIMIT 1);
8.   SET @tt_resu = (SELECT http_put(CONCAT('http://192.168.8.34:1978/', NEW.id), @tt_json));
9. END |
10. DELIMITER ;
11.
```

```

12.  /* UPDATE更新操作的触发器 */
13.  DELIMITER |
14.  DROP TRIGGER IF EXISTS mytable_update;
15.  CREATE TRIGGER mytable_update
16.  AFTER UPDATE ON mytable
17.  FOR EACH ROW BEGIN
18.      SET @tt_json = (SELECT json_object(id,addtime,title) FROM mytable WHERE id = OLD.id LIMIT 1);
19.      SET @tt_resu = (SELECT http_put(CONCAT('http://192.168.8.34:1978/', OLD.id), @tt_json));
20.  END |
21.  DELIMITER ;
22.
23.  /* DELETE删除操作的触发器 */
24.  DELIMITER |
25.  DROP TRIGGER IF EXISTS mytable_delete;
26.  CREATE TRIGGER mytable_delete
27.  AFTER DELETE ON mytable
28.  FOR EACH ROW BEGIN
29.      SET @tt_resu = (SELECT http_delete(CONCAT('http://192.168.8.34:1978/', OLD.id)));
30.  END |
31.  DELIMITER ;

```

(4). 将 MySQL 表和 Tokyo Tyrant 关联进行查询:

mysql>

view plain print ?

```

1.  SELECT id,addtime,title,http_get(CONCAT('http://192.168.8.34:1978/',id)) AS tt FROM mytable ORDER BY id DESC LIMIT 0,5;

```

5. 如何删除mysql-udf-http UDF函数:

mysql>

view plain print ?

```

1.  drop function http_get;
2.  drop function http_post;
3.  drop function http_put;
4.  drop function http_delete;

```