

# GQSERV - A CENTRALISED MECHANISM FOR QUERYING GOOGLE

AIDAN FINN

## 1. INTRODUCTION

gqserv allows the centralised pooling of google queries and caching of retrieved results. Queries are submitted to gqserv and queued. They are retrieved from google(in xml format) during off-peak hours and cached. Cached queries are then available to users. Interaction with gqserv is through a socket interface. There are also command line tools for accessing gqserv: gqclient, gqsubmit and gqretrieve.

## 2. THE SOCKET INTERFACE

gqserv operates as a socket server. To use the server directly, the client program must open a socket connection to the appropriate port. The client then sends a command to the server. The server executes the command and returns a status message. The first character of the returned message indicated success(1) or failure(0) of the command.

Commands sent to the server are required to be in netstring format

The format of a netstring is:

`Length:<STRING> ,`

Length is the decimal representation in ascii of the number of characters in the string. A colon follows the length, then comes the actual string and then a terminating comma for simple error check. So the string

`"hello world!"`

would actually be transmitted as

`"12:hello world!,"`

Strings sent to gqserv should be encoded in netstring format before transmission and strings received should be decoded from netstring format.

## 3. AVAILABLE COMMANDS

Commands sent to the server are of the format

`command [keyword1 keyword2 keyword3 ...] [param: parameter1 param: parameter2 ...]`

command is the actual command to be performed by the server. A sequence of keywords makes up the query. Optional parameters can be listed after the query. The parameters must come after the query.

The following commands are recognised by the server:

- add\_query
- add\_query\_force
- retrieve\_query

- `queue_stat`
- `queue_contents`

*add\_query.* Adds a query to the gqserv queue. E.g.

```
add_query hello world param:num=50
```

This adds the query 'hello world' to the queue with the google parameter num set to 50 to return 50 results. When the query is retrieved it is stored in the cache.

*retrieve\_query.* Retrieves a query from the cache if it exists. E.g.

```
retrieve_query hello world param: num=50
```

retrieves the above query from the cache. Do not try to access the cache directly.

*add\_query\_force.* Same as `add_query` except that it adds a query to the gqserv queue even if it is already in the cache.

*queue\_stat.* Returns some gqserv queue statistics

*queue\_contents.* Returns the currents of the gqserv queue

#### 4. GQCLIENT

gqclient is an tool for interactively sending commands to gqserv. It has two optional arguments.

```
gqclient [-h host] [-p port]
```

Here is a sample gqclient session:

```
[aidan@smi gqserv]$ ./gqclient.py
Host: 127.0.0.1
port: 8081
Send: asdas asda
Returned: 0 UNKNOWN COMMAND:asdas
Send: queue_stat
Returned: 0 items in queue
Send: add_query hello world
Returned: 1 ADDED: hello_world_
Send: add_query the hellacopters param: num=50
Returned: 1 ADDED: the_hellacopters_param:_num=50_
Send: retrieve_query hello world
Returned: 0 QUERY NOT IN CACHE hello_world_
Send: queue_contents
Returned: ['hello_world_', 'the_hellacopters_param:_num=50_']
```

#### 5. GQSUBMIT AND GQRETRIEVE

These are command line tools for submitting and retrieving multiple queries.

```
gqsubmit [-h hostname] [-p port] filename
```

```
gqretrieve [-h hostname] [-p port] filename
```

The input file should contain one query per line. Run `gqsubmit` to submit all the queries in the file to gqserv. Running `gqretrieve` the next day will retrieve all the queries and save each one in an individual file in the current directory. The filename will be the query with spaces replaced by underscores. Here is an example query file:

```
the hellcopters
  monster magnet
  macos vs linux param: num=50
  hello world :param num=20 param: start=10
  liverpool f.c.
  i hate man united
```

## 6. NOTES

- Software has not been tested much
- Server is multithreaded, but has not been load tested
- File config.py contains gqserv configuration information
- If in doubt, UTSL (use the source luke...)
- bug-reports, feature requests to [aidan.finn@ucd.ie](mailto:aidan.finn@ucd.ie)

*URL:* (<http://cr.yp.to/proto/netstrings.txt>).