# NGX\_HTTP\_QUEUE\_MODULE说明文档

## 主要功能描述

本模块对一般队列的数据结构进行了改造，使得任意一个队列内的消息既可以根据FIFO机制入队出队，也可以根据某些消息的键值在红黑树上查找并直接抽取，实现了对消息队列更为方便灵活的管理。

## 2. 模块结构介绍



图2.1 ngx\_http\_queue\_module数据结构

Shm\_zone是共享内存区域，shm\_data是一级红黑树和信道队列的入口。Shm\_data中的tree，指向一级信道红黑树的根节点。一级红黑树上的节点存储着不同id的信道的地址。Channels\_queue列作为岗哨节点，通过信道结构体的queue指针，将所有的信道结构体链接成一个循环队列，如图2.2。这样既可以通过id在红黑树上快速哈希查找到信道结构体，也可以通过信道队列逐个遍历信道，利于信道和信道内消息的统计。

信道结构体中的tree指向该消息结构体的二级红黑树的根节点。二级红黑树上的叶节点存储着不同key值的消息的地址。leafs\_queue列作为岗哨节点，通过叶子结构体的queue指针，将所有的叶子结构体链接成一个循环队列，如图2.3，叶子结构体中的msg\_poiner指针指向对应的消息结构体的地址。这样既可以通过key在红黑树上快速哈希查找到消息结构体，也可以通过叶子结构体队列逐个遍历消息，方便信道内消息的管理。



图2.2 ngx\_http\_queue\_module信道组织结构



图2.3 ngx\_http\_queue\_module信道消息组织结构

## 3. 模块操作方法

3.1. 入队命令

向id为123的信道队列输入消息“Hello World!”，key值为aaa的命令格式如下：

curl "http://127.0.0.1/add\_queue?id=123&key=aaa" -d 'Hello World!' -i

命令返回存储进信道队列的内容。以上命令的返回值即是Hello World!

如果入队命令没有key值，则默认key值为空，出队时只需输入信道id即可。

nginx.conf文件中配置命令如下：

location =/add\_queue {

channel\_id $arg\_id;

channel\_key $arg\_key;

add\_queue;

}

3. 2. 出队命令

从id为123的信道队列输出key值为aaa的消息，命令格式如下：

curl "http://127.0.0.1/get\_queue?id=123&key=aaa"

如果信道队列不存在或者key值为aaa的消息不存在则返回406，如果信道队列存在并且key值为aaa的消息存在则返回该消息.

从id为123的信道队列输出队头消息，命令格式如下：

curl "http://127.0.0.1/ get\_queue?id=123"

如果信道队列不存在则返回406，如果信道队列存在但为空，则返回空值

nginx.conf文件中配置命令如下：

location =/get\_queue {

channel\_id $arg\_id;

channel\_key $arg\_key;

get\_queue;

}

3.3. 删除信道队列

删除id为123的信道队列，命令格式如下：

curl "http://127.0.0.1/delete\_queue?id=123"

如果信道队列不存在则返回406，如果信道队列存在则返回值为空

nginx.conf文件中配置命令如下：

location =/delete\_queue{

channel\_id $arg\_id;

delete\_queue;

}

3.4. 统计整个存储空间内信道队列，以及分别存储的消息条数

命令格式如下：

curl "http://127.0.0.1/statis\_queue"

返回信道队列信息

nginx.conf文件中配置命令如下：

location =/statis\_queue{

statis\_queue;

}

3.5. 最大信道数，每个信道存储的最大消息个数，共享内存大小设置参数分别为下：

max\_number\_of\_channels

max\_messages\_stored\_per\_channel

queue\_shm\_zone

3.6. nginx配置文件具体配置举例如下。

ex：

max\_number\_of\_channels 10;

max\_messages\_stored\_per\_channel 10000;

queue\_shm\_zone 200M;

server {

listen 80;

server\_name \*.intsig.net \*.camcard.me \*.camcard.com;

lua\_check\_client\_abort on;

include \*server.conf;

location =/add\_queue {

channel\_id $arg\_id;

channel\_key $arg\_key;

add\_queue;

}

location =/get\_queue {

channel\_id $arg\_id;

channel\_key $arg\_key;

get\_queue;

}

location =/delete\_queue{

channel\_id $arg\_id;

delete\_queue;

}

location =/statis\_queue{

statis\_queue;

}

}

## 4. 模块效果展示

各种命令的具体使用效果，请参考下图的效果展示。测试脚本参考test.sh。

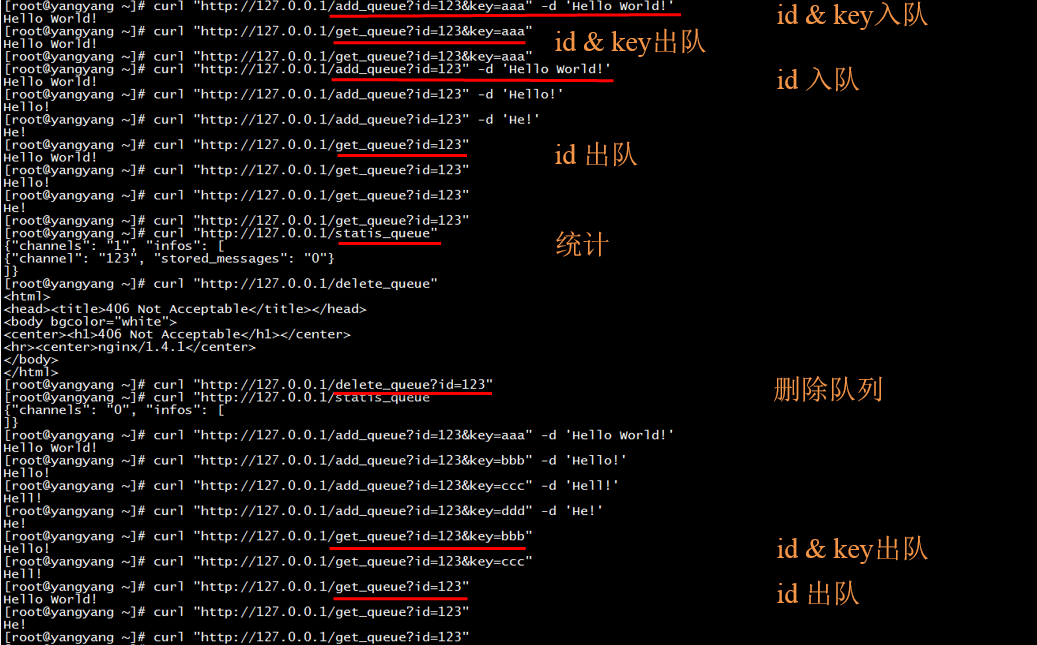


图4.1 各种命令的效果展示