

CPSC 473 -01
Assignment 6 Solution
Sarah Alfayez

1. Screenshots of TryRedis: (finishing TryRedis, and the result of keys *)



* TRY REDIS *

> NEXT

That wraps up the *Try Redis* tutorial. Please feel free to goof around with this console as much as you'd like.

Check out the following links to continue learning about Redis.

- [Redis Documentation](#)
- [Command Reference](#)
- [Implement a Twitter Clone in Redis](#)
- [Introduction to Redis Data Types](#)

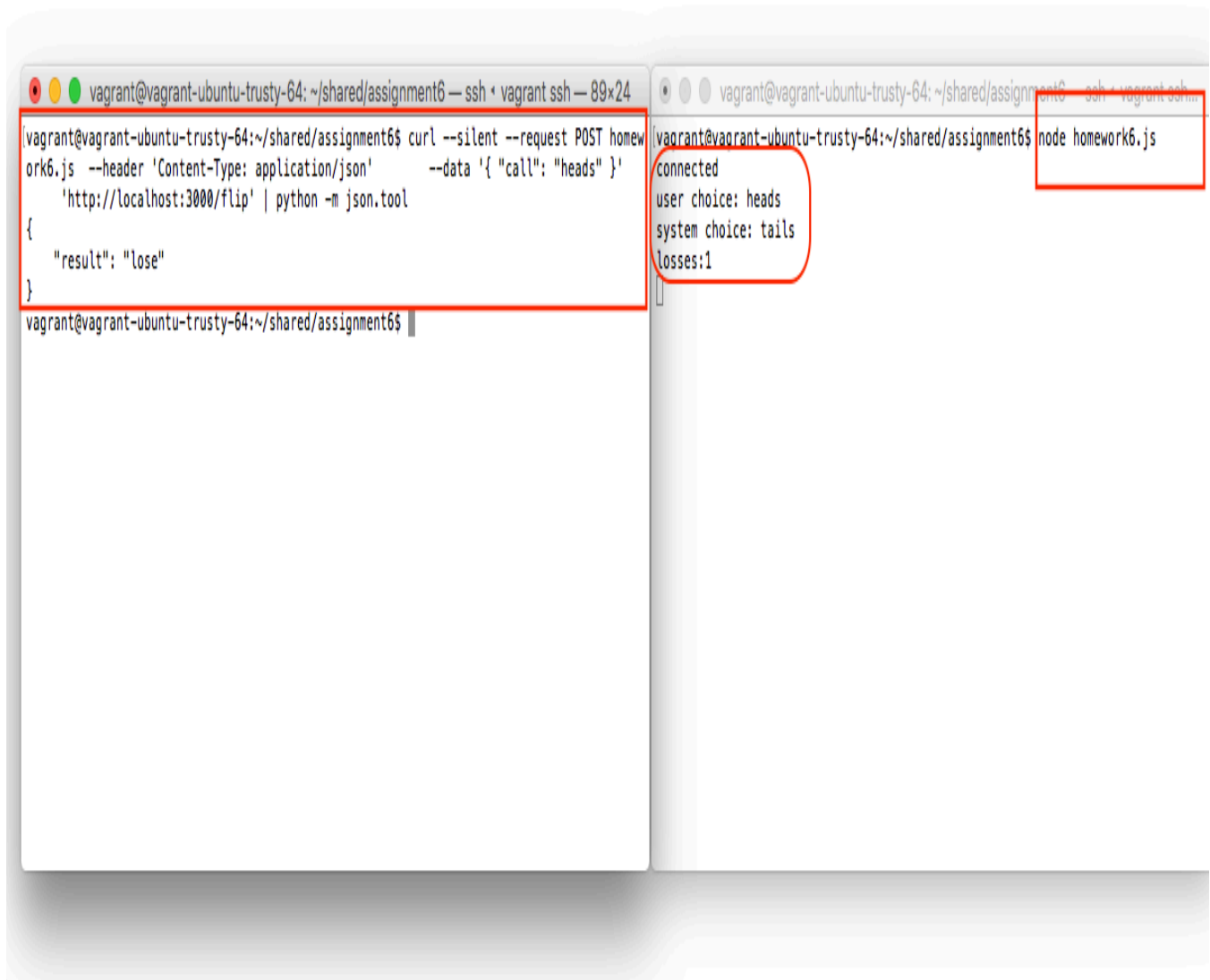
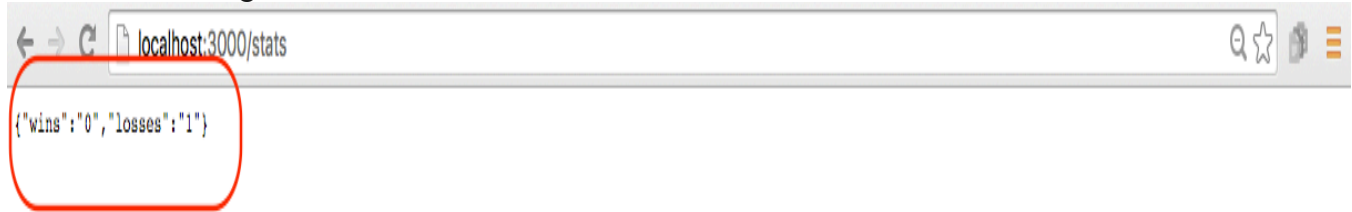
> keys *

- 1) "friends"
- 2) "superpowers"
- 3) "connections"
- 4) "hackers"
- 5) "birdpowers"
- 6) "user:1001"
- 7) "resource:lock"
- 8) "server:name"
- 9) "user:1000"

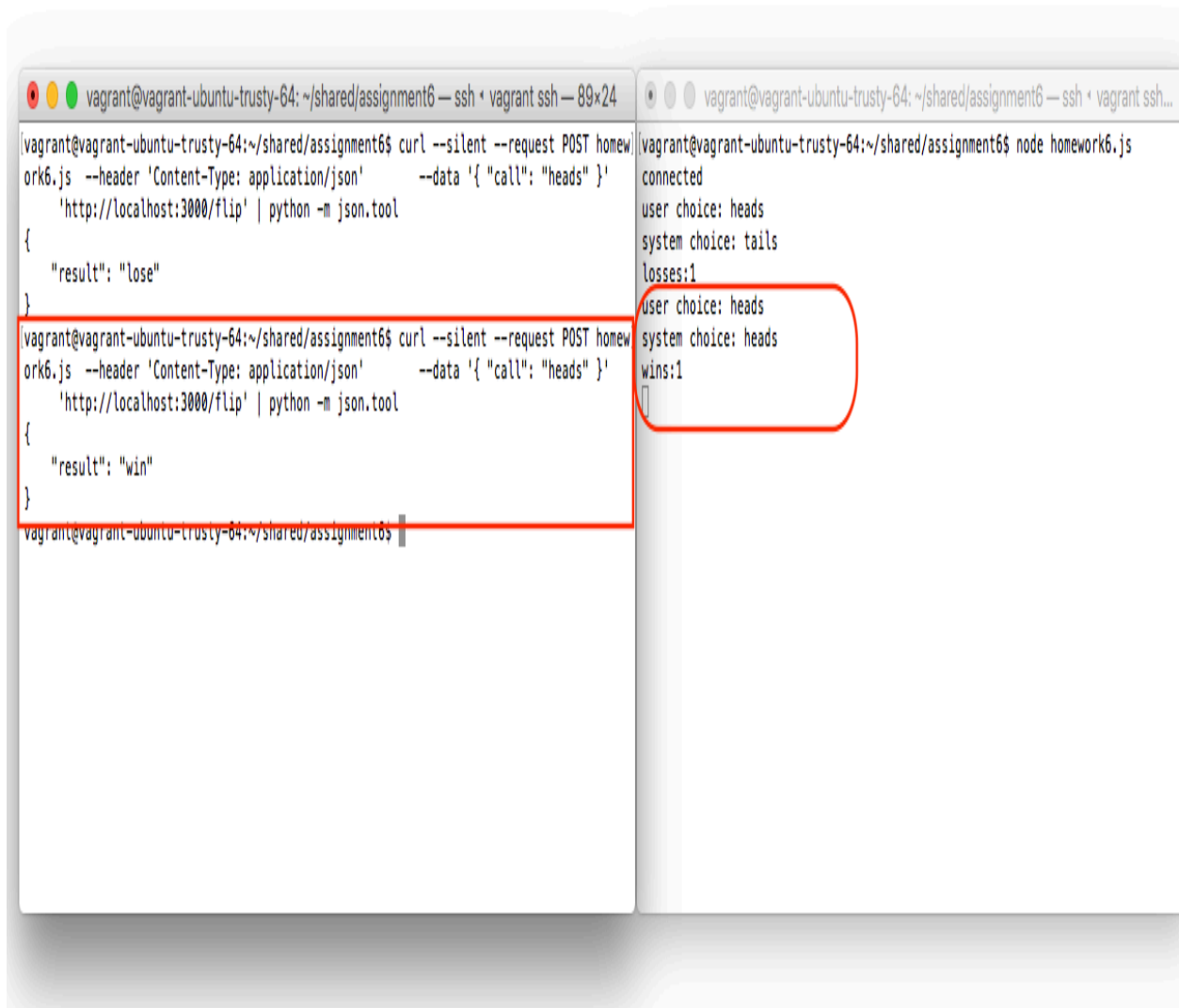


2. Screenshots of server in action:

The first figure shows the initial test with result of a lose



The second and third figure shows the result of a series of tests:



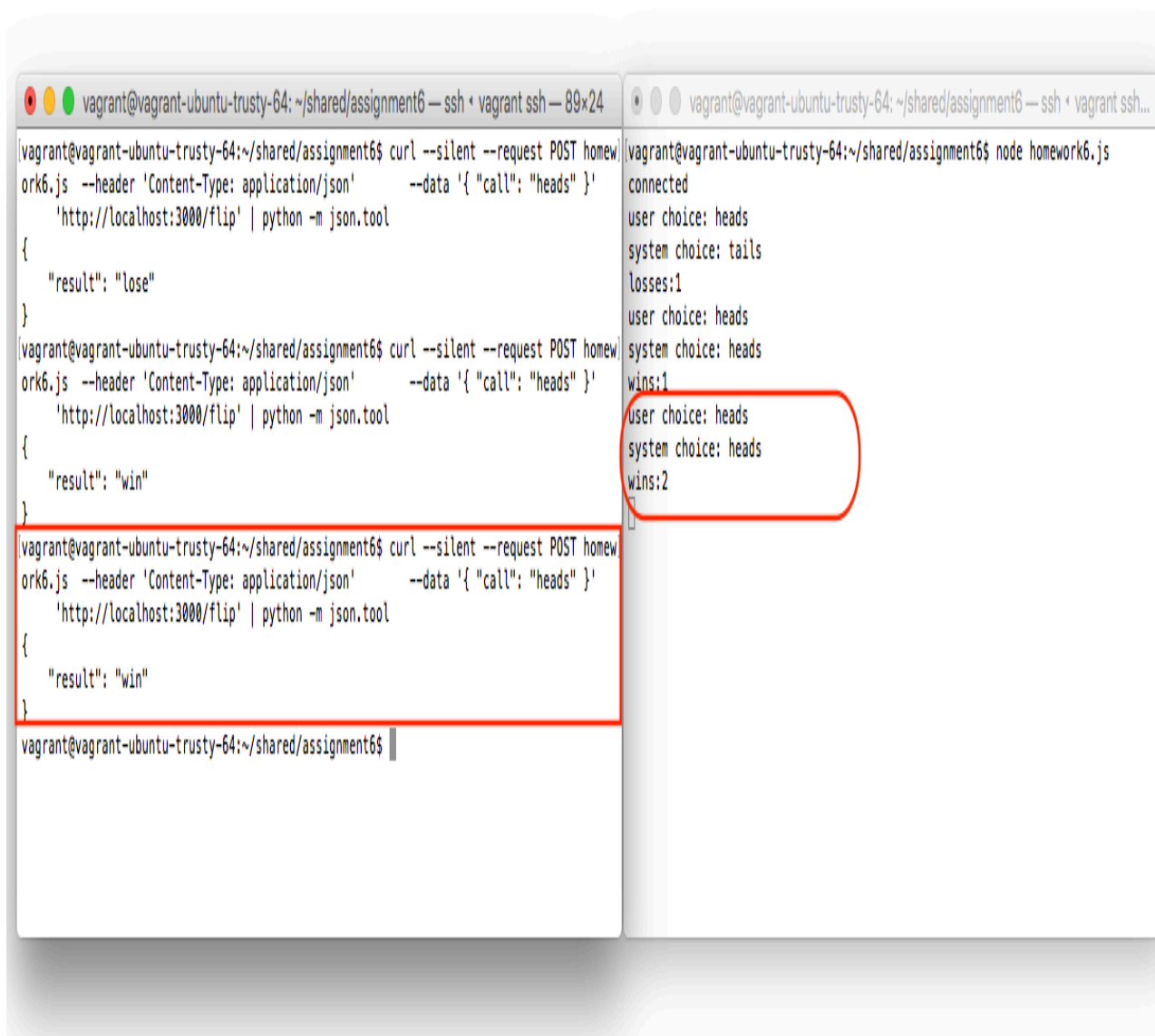
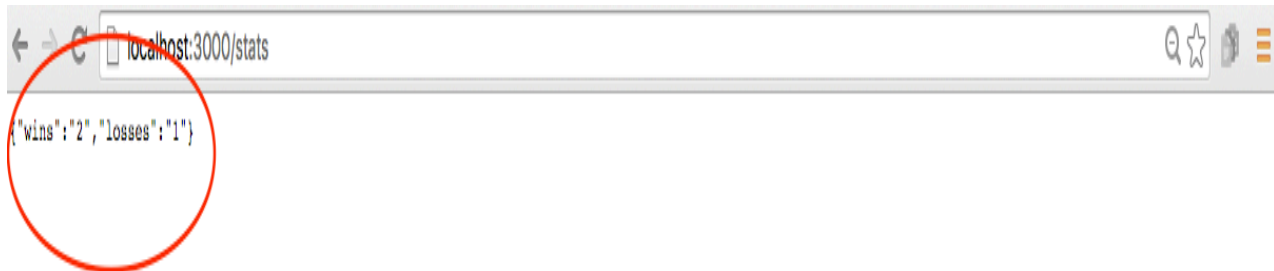


Figure 4 shows that the results will persist even if we restart node.js

The image shows a web browser window at the top and two terminal windows below it. The browser window displays the URL `localhost:3000/stats` and the JSON response `{"wins": "3", "losses": "4"}`, which is circled in red. The left terminal window shows a series of `curl` requests to `localhost:3000/flip` using `ork6.js` with a `Content-Type: application/json` header. The requests alternate between `"heads"` and `"tails"` data. The right terminal window shows the output of the application, which tracks the number of wins and losses. The output shows that the state is persistent across restarts, with wins increasing from 1 to 3 and losses increasing from 1 to 4. The right terminal window is also circled in red.

```
localhost:3000/stats
{"wins": "3", "losses": "4"}
```

```
vagrant@vagrant-ubuntu-trusty-64: ~/shared/assignment6 — ssh • vagrant ssh — 89x24
ork6.js --header 'Content-Type: application/json' --data '{"call": "heads"}'
'http://localhost:3000/flip' | python -m json.tool
{
  "result": "win"
}
[vagrant@vagrant-ubuntu-trusty-64: ~/shared/assignment6$ curl --silent --request POST http://localhost:3000/flip | python -m json.tool
ork6.js --header 'Content-Type: application/json' --data '{"call": "heads"}'
'http://localhost:3000/flip' | python -m json.tool
{
  "result": "lose"
}
[vagrant@vagrant-ubuntu-trusty-64: ~/shared/assignment6$ curl --silent --request POST http://localhost:3000/flip | python -m json.tool
ork6.js --header 'Content-Type: application/json' --data '{"call": "heads"}'
'http://localhost:3000/flip' | python -m json.tool
{
  "result": "lose"
}
[vagrant@vagrant-ubuntu-trusty-64: ~/shared/assignment6$ curl --silent --request POST http://localhost:3000/flip | python -m json.tool
ork6.js --header 'Content-Type: application/json' --data '{"call": "heads"}'
'http://localhost:3000/flip' | python -m json.tool
{
  "result": "lose"
}
[vagrant@vagrant-ubuntu-trusty-64: ~/shared/assignment6$
```

```
vagrant@vagrant-ubuntu-trusty-64: ~/shared/assignment6 — ssh • vagrant ssh...
system choice: tails
losses:1
user choice: heads
system choice: heads
wins:1
user choice: heads
system choice: heads
wins:2
user choice: heads
system choice: heads
wins:3
user choice: heads
system choice: tails
losses:2
user choice: heads
system choice: tails
losses:3
user choice: heads
system choice: tails
losses:4
^C
vagrant@vagrant-ubuntu-trusty-64: ~/shared/assignment6$ node homework6.js
connected
```

Figure 5 shows the results after restarting node.js

The image shows a web browser window at the top and two terminal windows below it. The browser window displays the URL `localhost:3000/stats` and the JSON response `{"wins": "4", "losses": "6"}`, which is circled in red. The left terminal window shows a series of curl commands and their JSON outputs, with the final output `{"result": "lose"}` circled in red. The right terminal window shows the output of the `node homework6.js` command, with the final output `losses:6` circled in red.

```
localhost:3000/stats
{"wins": "4", "losses": "6"}
```

```
vagrant@vagrant-ubuntu-trusty-64: ~/shared/assignment6 — ssh * vagrant ssh — 89x24
ork6.js --header 'Content-Type: application/json' --data '{ "call": "heads" }'
'http://localhost:3000/flip' | python -m json.tool
{
  "result": "lose"
}
vagrant@vagrant-ubuntu-trusty-64: ~/shared/assignment6$ curl --silent --request POST homew
ork6.js --header 'Content-Type: application/json' --data '{ "call": "heads" }'
'http://localhost:3000/flip' | python -m json.tool
{
  "result": "lose"
}
vagrant@vagrant-ubuntu-trusty-64: ~/shared/assignment6$ curl --silent --request POST homew
ork6.js --header 'Content-Type: application/json' --data '{ "call": "heads" }'
'http://localhost:3000/flip' | python -m json.tool
{
  "result": "win"
}
vagrant@vagrant-ubuntu-trusty-64: ~/shared/assignment6$ curl --silent --request POST homew
ork6.js --header 'Content-Type: application/json' --data '{ "call": "heads" }'
'http://localhost:3000/flip' | python -m json.tool
{
  "result": "lose"
}
vagrant@vagrant-ubuntu-trusty-64: ~/shared/assignment6$
```

```
vagrant@vagrant-ubuntu-trusty-64: ~/shared/assignment6 — ssh * vagrant ssh...
system choice: heads
wins:3
user choice: heads
system choice: tails
losses:2
user choice: heads
system choice: tails
losses:3
user choice: heads
system choice: tails
losses:4
^C
vagrant@vagrant-ubuntu-trusty-64: ~/shared/assignment6$ node homework6.js
connected
user choice: heads
system choice: tails
losses:5
user choice: heads
system choice: heads
wins:4
user choice: heads
system choice: tails
losses:6
```

Figure 6 show the result of running the delete on /stats using the following curl statement
`curl -i -X DELETE -H "Content-Type: application/json" --user root:root http://localhost:3000/stats`

The image shows a web browser window at the top and two terminal windows below it. The browser window displays the URL `localhost:3000/stats` and the JSON response `{"wins": "0", "losses": "0"}`, which is circled in red. The left terminal window shows a series of curl commands and their outputs, with the final delete command and its response circled in red. The right terminal window shows the output of a Node.js application, with the final message `stats has been reset to 0` circled in red.

```
localhost:3000/stats
{"wins": "0", "losses": "0"}
```

```
vagrant@vagrant-ubuntu-trusty-64: ~/shared/assignment6 — ssh • vagrant ssh — 89x24
}
vagrant@vagrant-ubuntu-trusty-64:~/shared/assignment6$ curl --silent --request POST homework6.js --header 'Content-Type: application/json' --data '{"call": "heads"}' 'http://localhost:3000/flip' | python -m json.tool
{
  "result": "win"
}
vagrant@vagrant-ubuntu-trusty-64:~/shared/assignment6$ curl --silent --request POST homework6.js --header 'Content-Type: application/json' --data '{"call": "heads"}' 'http://localhost:3000/flip' | python -m json.tool
{
  "result": "lose"
}
vagrant@vagrant-ubuntu-trusty-64:~/shared/assignment6$ curl -i -X DELETE -H "Content-Type: application/json" --user root:root http://localhost:3000/stats
HTTP/1.1 200 OK
X-Powered-By: Express
Content-Type: application/json; charset=utf-8
Content-Length: 4
ETag: W/"4-Wzq/nBqnVWw6Nv6k5pXF0g"
Date: Tue, 12 Apr 2016 23:17:18 GMT
Connection: keep-alive

"OK"vagrant@vagrant-ubuntu-trusty-64:~/shared/assignment6$
```

```
vagrant@vagrant-ubuntu-trusty-64: ~/shared/assignment6 — ssh • vagrant ssh...
wins:3
user choice: heads
system choice: tails
losses:2
user choice: heads
system choice: tails
losses:3
user choice: heads
system choice: tails
losses:4
^C
vagrant@vagrant-ubuntu-trusty-64:~/shared/assignment6$ node homework6.js
connected
user choice: heads
system choice: tails
losses:5
user choice: heads
system choice: heads
wins:4
user choice: heads
system choice: tails
losses:6
stats has been reset to 0
```


Figure 7 shows the results after resetting the counters to 0

The image shows a web browser window at the top and two terminal windows below it. The browser window displays the URL `localhost:3000/stats` and the JSON response `{"wins": "0", "losses": "2"}`, which is circled in red. The left terminal window shows a `curl` command being used to delete the `stats` endpoint, followed by a `node` command to run `homework6.js` with a `POST` request to `http://localhost:3000/flip`. The right terminal window shows the output of the `node` command, which includes a series of coin flip results (heads/tails) and a message indicating that the stats have been reset to 0. The output is circled in red.

```
localhost:3000/stats
{"wins": "0", "losses": "2"}
```

```
vagrant@vagrant-ubuntu-trusty-64: ~/shared/assignment6$ curl -i -X DELETE -H "Content-Type: application/json" --user root:root http://localhost:3000/stats
HTTP/1.1 200 OK
X-Powered-By: Express
Content-Type: application/json; charset=utf-8
Content-Length: 4
ETag: W/"4-Wzq/nBqnVWw6Nv6k5pXF0g"
Date: Tue, 12 Apr 2016 23:17:18 GMT
Connection: keep-alive

{"OK"}
vagrant@vagrant-ubuntu-trusty-64:~/shared/assignment6$ node homework6.js
connected
user choice: heads
system choice: tails
losses:3
user choice: heads
system choice: tails
losses:4
user choice: heads
system choice: heads
wins:4
user choice: heads
system choice: tails
losses:5
user choice: heads
system choice: tails
losses:6
stats has been reset to 0
user choice: heads
system choice: tails
losses:1
user choice: heads
system choice: tails
losses:2
```