



PERSISTING DATA

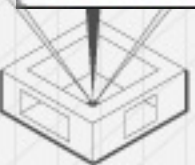
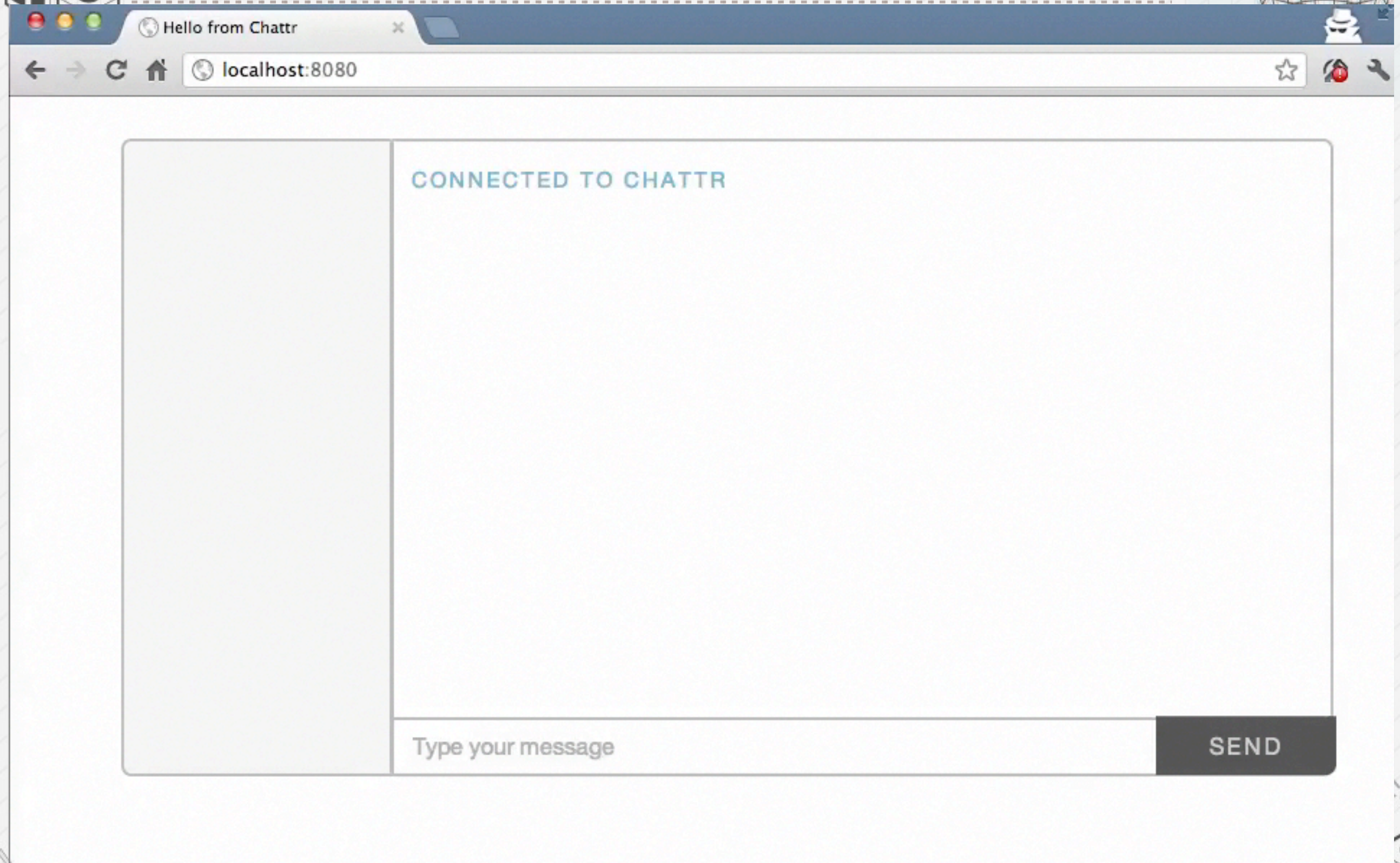
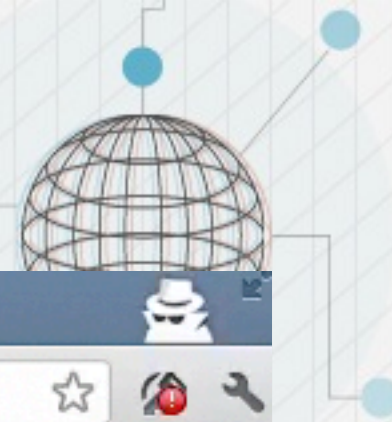
- LEVEL SEVEN -

.....

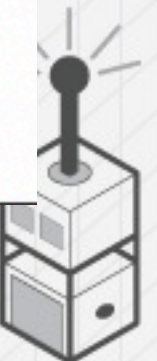




RECENT MESSAGES



PERSISTING DATA



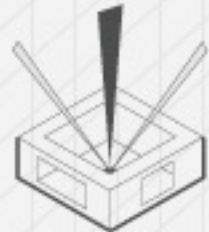


RECENT MESSAGES



app.js

```
io.sockets.on('connection', function(client) {  
  client.on('join', function(name) {  
    client.nickname = name;  
    client.broadcast.emit("chat", name + " joined the chat");  
  });  
  client.on("messages", function(message){  
    client.broadcast.emit("messages", client.nickname +  
      ": " + message);  
    client.emit("messages", client.nickname +  
      ": " + message);  
  });  
});
```



PERSISTING DATA





STORING MESSAGES



app.js

```
var messages = []; store messages in array
```

```
var storeMessage = function(name, data){  
  messages.push({name: name, data: data}); add message to end of array  
  if (messages.length > 10) {  
    messages.shift(); if more than 10 messages long,  
                      remove the first one  
  }  
}
```

```
io.sockets.on('connection', function(client) {  
  client.on("messages", function(message){  
    client.broadcast.emit("messages", client.nickname +  
      ": " + message);  
    client.emit("messages", client.nickname + ": " + message);  
    storeMessage(client.nickname, message);  
  });  
});
```

*when client sends a message
call storeMessage*





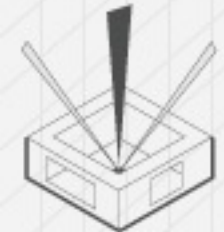
EMITTING MESSAGES



app.js

```
io.sockets.on('connection', function(client) {  
  ...  
  client.on('join', function(name) {  
    messages.forEach(function(message) {  
      client.emit("messages", message.name + ": " + message.data);  
    });  
  });  
});
```

*iterate through messages array
and emit a message on the connecting
client for each one*

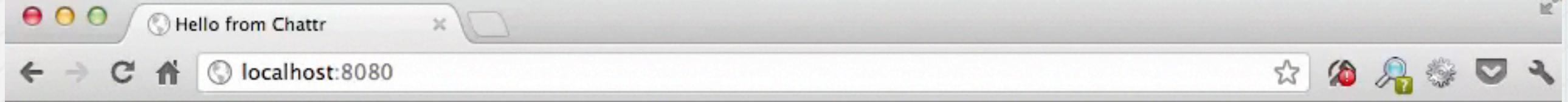


PERSISTING DATA





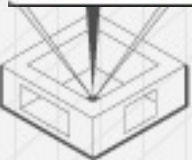
RECENT MESSAGES



CONNECTED TO CHATTR

Type your message

SEND



PERSISTING DATA





PERSISTING STORES



- MongoDB
- CouchDB
- PostgreSQL
- Memcached
- Riak

All non-blocking!



redis

Redis is a key-value store



PERSISTING DATA





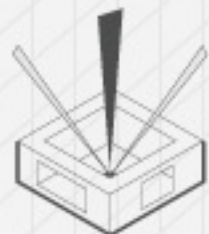
REDIS DATA STRUCTURES



data structure

commands

Strings	<i>SET, GET, APPEND, DECR, INCR...</i>
Hashes	<i>HSET, HGET, HDEL, HGETALL...</i>
Lists	<i>LPUSH, LREM, LTRIM, RPOP, LINSERT...</i>
Sets	<i>SADD, SREM, SMOVE, SMEMBERS...</i>
Sorted Sets	<i>ZADD, ZREM, ZSCORE, ZRANK...</i>

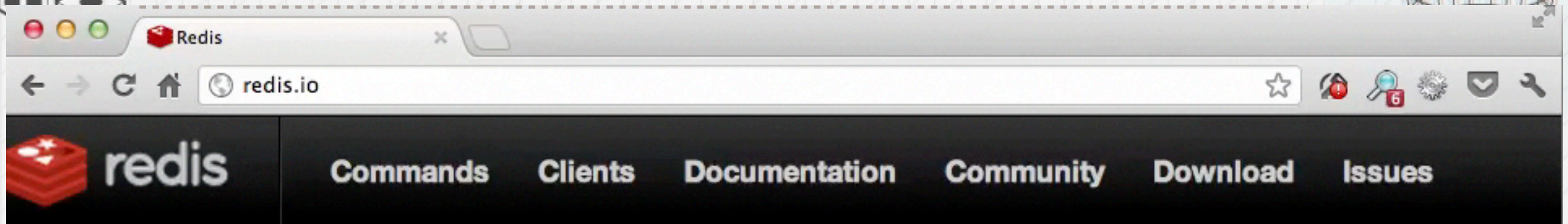


PERSISTING DATA





REDIS COMMAND DOCUMENTATION



Redis is an open source, advanced **key-value store**. It is often referred to as a **data structure server** since keys can contain **strings**, **hashes**, **lists**, **sets** and **sorted sets**.

[Learn more](#) →

Try it

Ready for a test drive? Check this [interactive tutorial](#) that will walk you through the most important features of Redis.

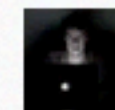
Download it

[Redis 2.4.13](#) is the latest **stable version**. Interested in release candidates or unstable versions? [Check the downloads page](#).

What people are saying



Facebook Sets I.P.O.
Price Range
<http://t.co/7qTOhWMx>



@tinkertim No more spaces screwing my Redis commands. Pretty major to me ;-)



#RedMango #coupon? Get a \$2 OFF one @coupons.com. Just enter your ZIP code in the upper left! US only.
<http://t.co/P0i9nvUh>



redis (@DIRTYBIITCH)

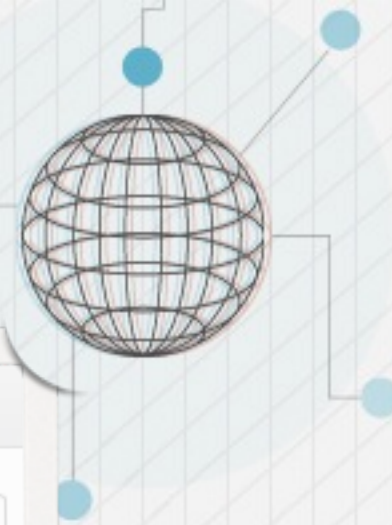


PERSISTING DATA





NODE REDIS



This repository ▾

Search or type a command



Explore

Gist

Blog

Help



TJkrusinski



mranney / **node_redis**



Watch ▾

176



Star

3,102



Fork

422

redis client for node

509 commits

3 branches

36 releases

63 contributors



branch: master ▾

node_redis / +

If there's an error in SELECT and there's no callback, emit the error.



bengl authored 12 days ago

latest commit 4672479b91

benches	rename tests to benches for clarity	2 years ago
examples	EVAL: allow parameters as an array. Close #368.	a year ago
lib	Regenerate commands.js to match redis 2.8.0. Adds support for 'config...	3 months ago
.gitignore	Issue #439 (and others): Stop assuming all "message" or "pmessage" re...	5 months ago
README.md	README for unix domain socket connection.	5 months ago
changelog.md	Update changelog for 0.10.1	13 days ago
diff_multi_bench_output.js	Adding percentage outputs to diff_multi_bench_output.js	a year ago
generate_commands.js	node 0.6 fixes	2 years ago
index.js	If there's an error in SELECT and there's no callback, emit the error.	12 days ago

<> Code

Issues 136

Pull Requests 35

Wiki

Pulse

Graphs

Network

SSH clone URL

git@github.com:mrar

You can clone with [HTTPS](#), [SSH](#), or [Subversion](#).



Clone in Desktop



Download ZIP



PERSISTING DATA



REDIS



```
$ npm install redis --save
```

```
var redis = require('redis');  
var client = redis.createClient();  
  
client.set("message1", "hello, yes this is dog");  
client.set("message2", "hello, no this is spider");
```

key

value

```
client.get("message1", function(err, reply){  
  console.log(reply); -----> "hello, yes this is dog"  
});
```

commands are non-blocking



PERSISTING DATA





REDIS LISTS: PUSHING



Add a string to the “messages” list

```
var message = "Hello, this is dog";  
client.lpush("messages", message, function(err, reply){  
  console.log(reply);  
});
```

→ "1" replies with list length

Add another string to “messages”

```
var message = "Hello, no this is spider";  
client.lpush("messages", message, function(err, reply){  
  console.log(reply);  
});
```

→ "2"



PERSISTING DATA





REDIS LISTS: RETRIEVING

Using LPUSH & LTRIM



```
var message = "Hello, this is dog";  
client.lpush("messages", message, function(err, reply){  
  client.ltrim("messages", 0, 1);  
});
```

*trim keeps first two strings
and removes the rest*

Retrieving from list

```
client.lrange("messages", 0, -1, function(err, messages){  
  console.log(messages);  
})
```

replies with all strings in list

`["Hello, no this is spider", "Oh sorry, wrong number"]`



PERSISTING DATA





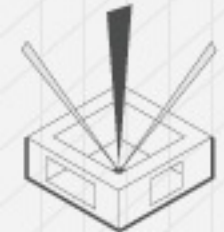
CONVERTING MESSAGES TO REDIS



```
var storeMessage = function(name, data){  
  messages.push({name: name, data: data});  
  
  if (messages.length > 10) {  
    messages.shift();  
  }  
}
```

app.js

Let's use the List data-structure



PERSISTING DATA





CONVERTING STOREMESSAGE



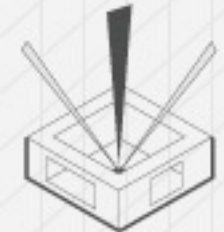
app.js

```
var redisClient = redis.createClient();

var storeMessage = function(name, data){
  var message = JSON.stringify({name: name, data: data});
  need to turn object into string to store in redis

  redisClient.lpush("messages", message, function(err, response) {
    redisClient.ltrim("messages", 0, 9);
  });
}
```

keeps newest 10 items



PERSISTING DATA



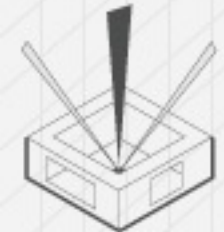


OUTPUT FROM LIST



```
client.on('join', function(name) {  
  messages.forEach(function(message) {  
    client.emit("messages", message.name + ": " + message.data);  
  });  
});
```

app.js

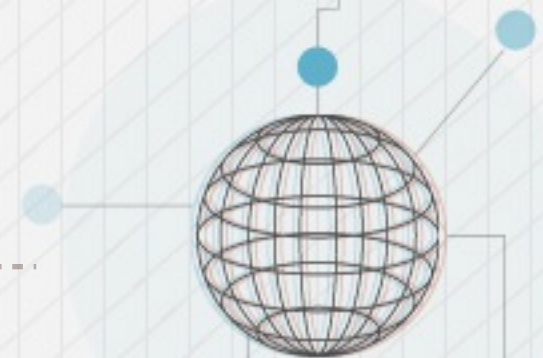


PERSISTING DATA





OUTPUT FROM LIST

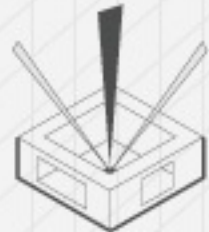


app.js

```
client.on('join', function(name) {  
  redisClient.lrange("messages", 0, -1, function(err, messages){  
    messages = messages.reverse();  
  
    messages.forEach(function(message) {  
      message = JSON.parse(message);  
      client.emit("messages", message.name + ": " + message.data);  
    });  
  });  
});
```

reverse so they are emitted in correct order

parse into JSON object

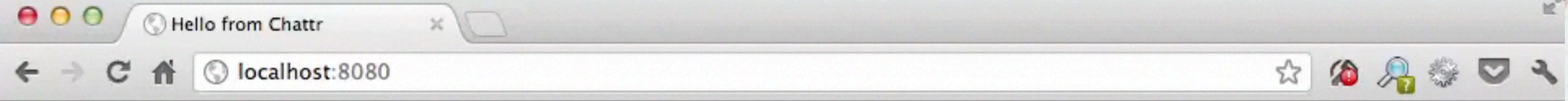


PERSISTING DATA





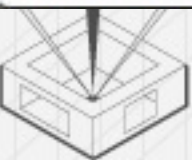
IN ACTION



CONNECTED TO CHATTR

Type your message

SEND

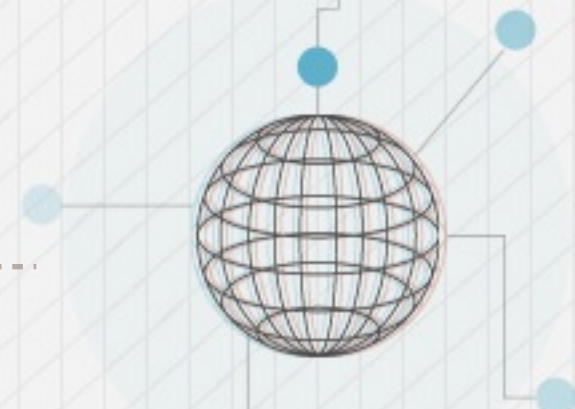


PERSISTING DATA





CURRENT CHATTER LIST



Sets are lists of unique data

add & remove members of the names set

```
client.sadd("names", "Dog");  
client.sadd("names", "Spider");  
client.sadd("names", "Gregg");
```

```
client.srem("names", "Spider");
```

reply with all members of set

```
client.smembers("names", function(err, names){  
  console.log(names);  
});
```

["Dog", "Gregg"]

DOG
SPIDER
GREGG



PERSISTING DATA





ADDING CHATTERS



```
client.on('join', function(name){
```

app.js

notify other clients a chatter has joined

```
client.broadcast.emit("add chatter", name);
```

```
redisClient.sadd("chatters", name);
```

```
});
```

add name to chatters set

```
socket.on('add chatter', function(name) {
```

index.html

```
var chatter = $('<li>'+name+'</li>').data('name', name);
```

```
$('#chatters').append(chatter);
```

```
});
```

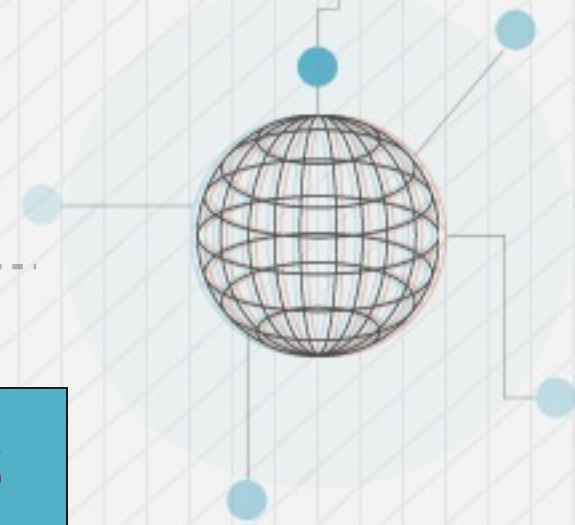


PERSISTING DATA





ADDING CHATTERS (CONT)

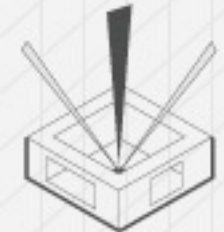


```
client.on('join', function(name){
  client.broadcast.emit("add chatter", name);
  redisClient.smembers('names', function(err, names) {
    names.forEach(function(name){
      client.emit('add chatter', name);
    });
  });
});
```

*emit all the currently logged in chatters
to the newly connected client*

```
redisClient.sadd("chatters", name);
});
```

app.js



PERSISTING DATA





REMOVING CHATTERS



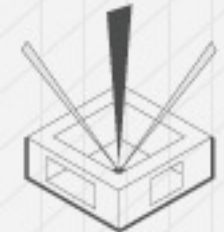
remove chatter when they disconnect from server

```
client.on('disconnect', function(name){
  client.broadcast.emit("remove chatter", client.nickname);
  redisClient.srem("chatters", client.nickname);
});
```

app.js

```
server.on('remove chatter', function(name) {
  $('#chatters li[data-name=' + name + ']').remove();
});
```

index.html

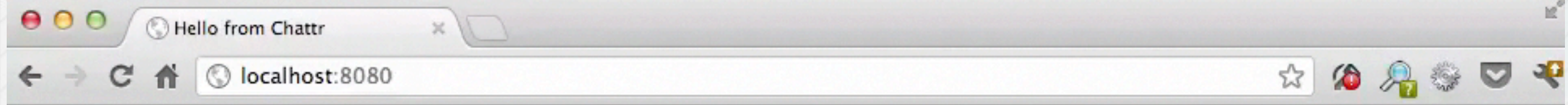


PERSISTING DATA





WELCOME TO CHATTR



Hello from Chattr

DOG

CONNECTED TO CHATTR

PERSISTING DATA

