

Тестовое задание:

Требуется создать real-time приложение с помощью socket.io, чтобы выполнялись следующие условия:

1. Клиент может отправлять и принимать сообщения;
2. Приложение ведет учет активных клиентов;
3. При потере соединения с каким-либо клиентом приложение отправляет всем активным сообщение с текущим количеством активных клиентов в виде: "Active clients: N", где N - это количество активных клиентов;
4. При подключении клиента к приложению, приложение должно отправлять текущему клиенту сообщение "Welcome!", а всем остальным клиентам - сообщение с текущим количеством активных клиентов в виде: "Active clients: N", где N - это количество активных клиентов;
5. Все клиенты должны записываться в MongoDB. А именно: нужно записать идентификатор соединения, время создания соединения (подключение клиента), а также при отключении клиента записать время этого события.

Результат необходимо предоставить в zip-архиве. Решение нельзя выкладывать в открытых репозиториях.

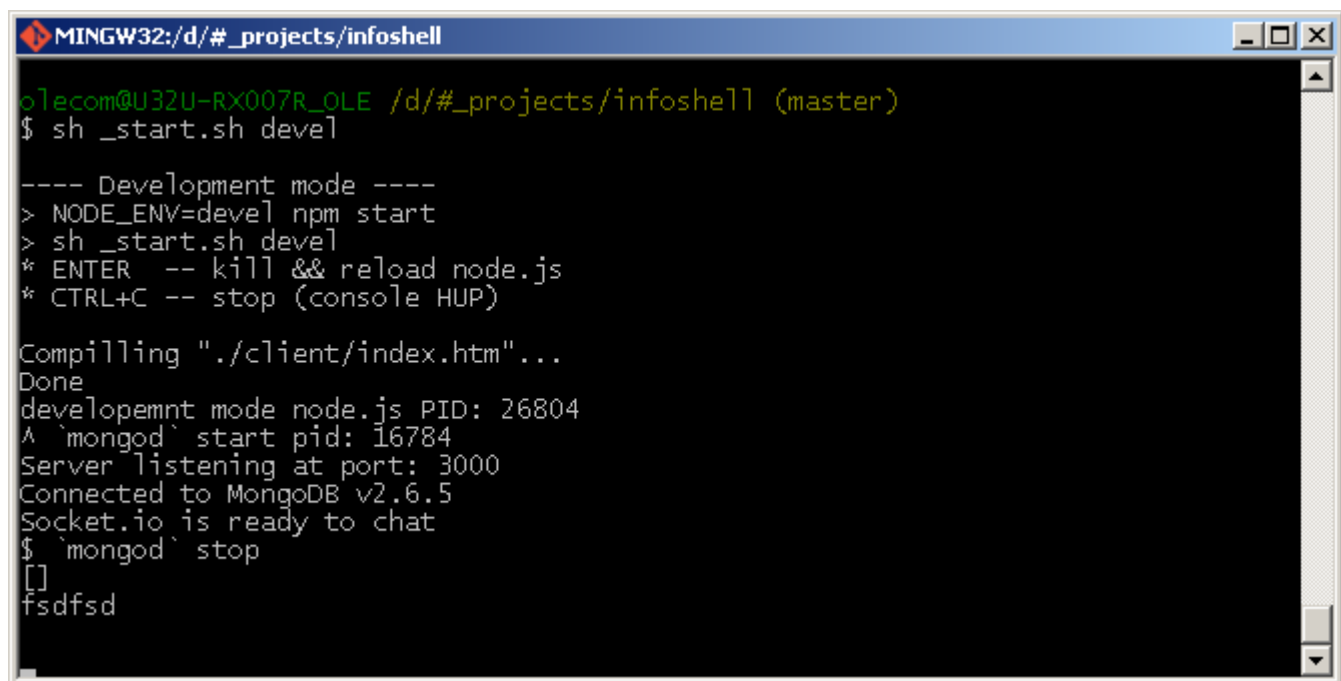
18:00 Started processing. Git, boilerplate, deps, npm,

2. Приложение ведет учет активных клиентов;

4. При подключении клиента к приложению, приложение должно отправлять текущему клиенту сообщение "Welcome!", а всем остальным клиентам - сообщение с текущим количеством активных клиентов в виде: "Active clients: N", где N - это количество активных клиентов;

3. При потере соединения с каким-либо клиентом приложение отправляет всем активным сообщение с текущим количеством активных клиентов в виде: "Active clients: N", где N - это количество активных клиентов;

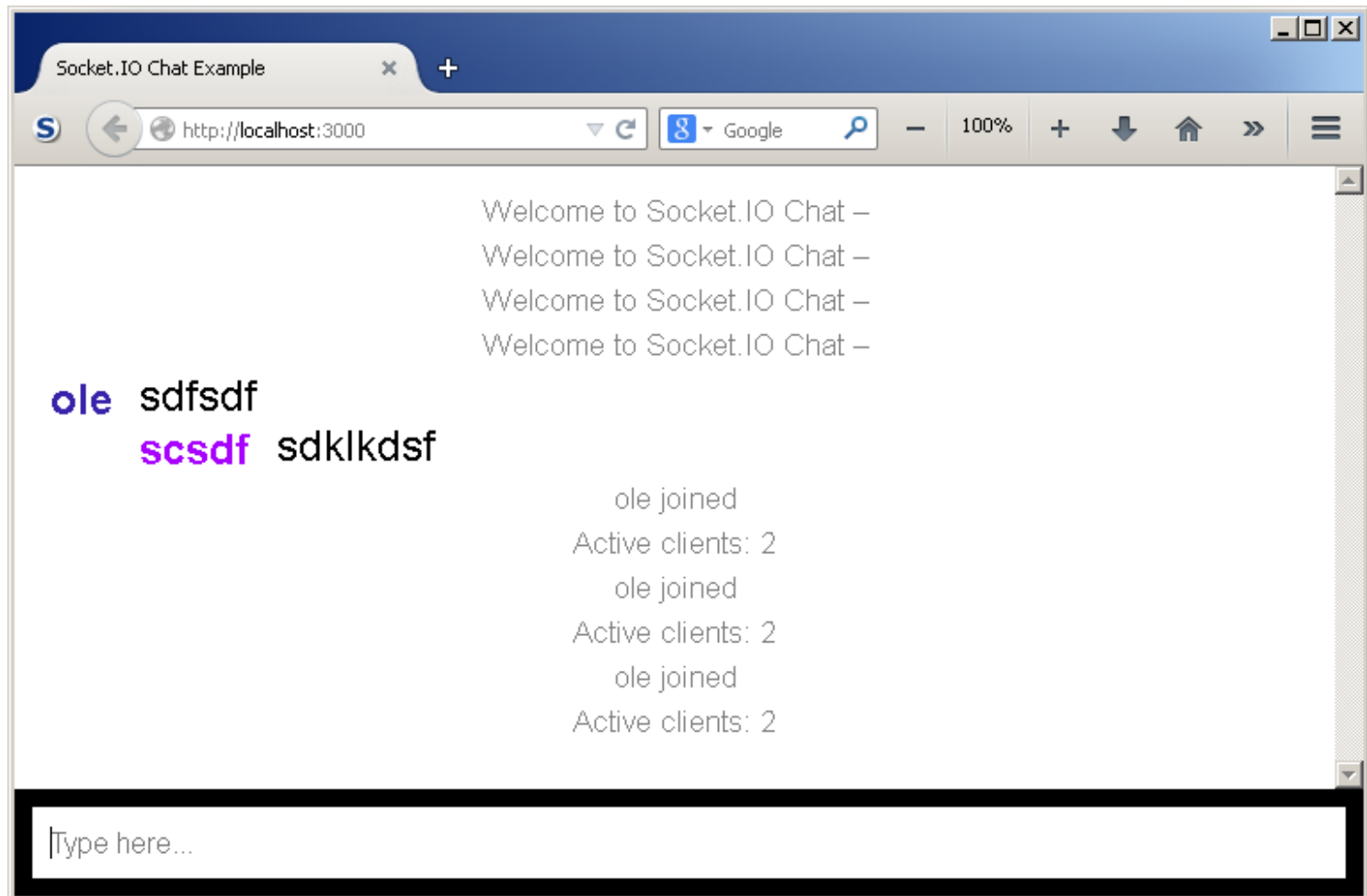
1. Клиент может отправлять и принимать сообщения;



```
MINGW32:/d/#_projects/infoshell
olecom@U32U-RX007R-OLE /d/#_projects/infoshell (master)
$ sh _start.sh devel

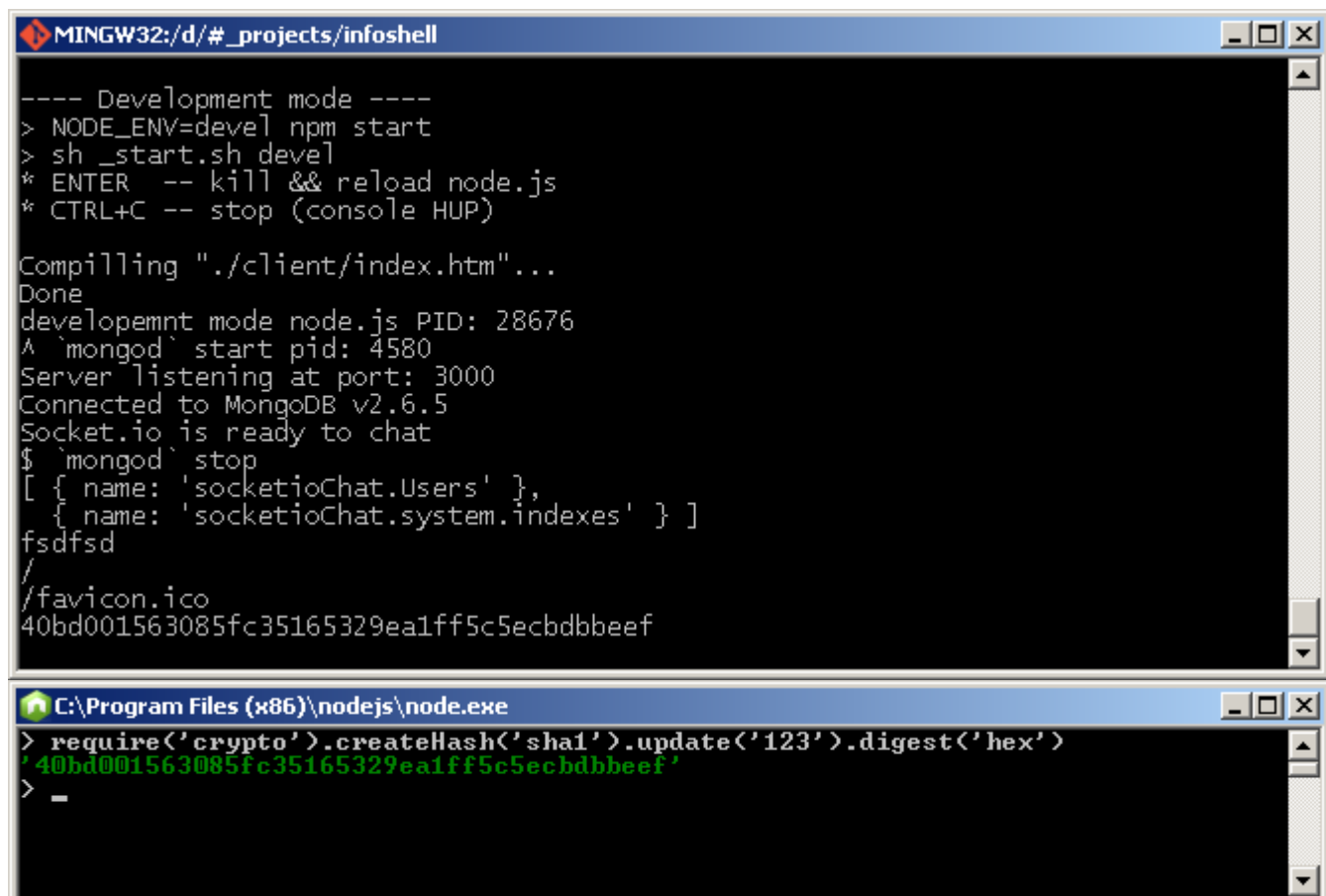
---- Development mode ----
> NODE_ENV=devel npm start
> sh _start.sh devel
* ENTER -- kill && reload node.js
* CTRL+C -- stop (console HUP)

Compiling "./client/index.htm"...
Done
development mode node.js PID: 26804
^ `mongod` start pid: 16784
Server listening at port: 3000
Connected to MongoDB v2.6.5
Socket.io is ready to chat
$ `mongod` stop
[]
fsdfsd
```



user pass date.valueOf() + date.toUTFString()

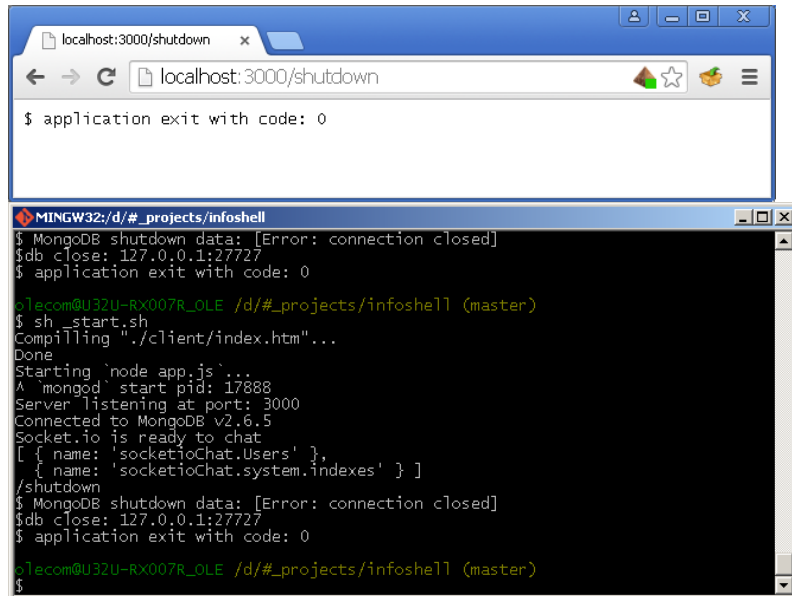
SHA1 pass:



5. Все клиенты должны записываться в MongoDB. А именно: нужно записать идентификатор соединения, время создания соединения (подключение клиента), а также при отключении клиента записать время этого события.

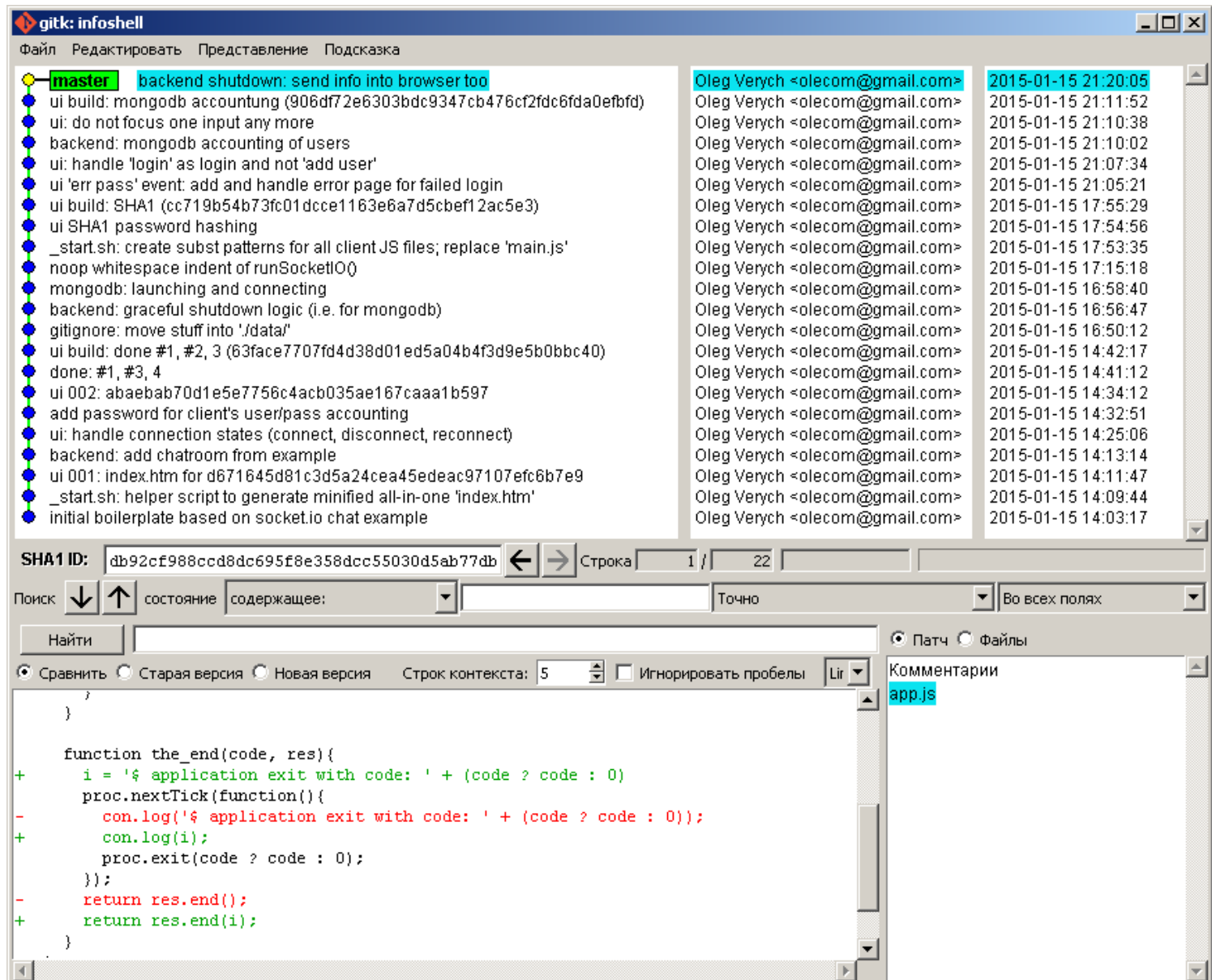
Finally mongodb and some bugfixes.

Propper Shutdown:



The screenshot shows two windows. The top window is a web browser at localhost:3000/shutdown, displaying '\$ application exit with code: 0'. The bottom window is a terminal running 'MINGW32/d/#_projects/infoshell'. It shows the MongoDB shutdown process: '\$ MongoDB shutdown data: [Error: connection closed]', '\$db close: 127.0.0.1:27727', '\$ application exit with code: 0'. Then it shows the restart process: '\$ sh_start.sh', 'Compiling "/>

Git:



The screenshot shows the Git GUI interface. The top bar indicates 'gitk: infoshell'. The main area displays a commit history list with columns for commit type, message, author, and date. The selected commit is 'master backend shutdown: send info into browser too' by 'Oleg Verych <olecom@gmail.com>' dated '2015-01-15 21:20:05'. Below the list, the 'SHA1 ID' is shown as 'db92cf988ccd8dc695f8e358dcc55030d5ab77db'. The 'Search' section shows 'состояние' and 'содержащее:' filters. The 'Diff' section shows the changes in 'app.js', with a diff view showing the 'the_end' function.

Commit Type	Message	Author	Date
master	backend shutdown: send info into browser too	Oleg Verych <olecom@gmail.com>	2015-01-15 21:20:05
ui	build: mongodb accounting (906df72e6303bdc9347cb476cf2fdc6fda0effbd)	Oleg Verych <olecom@gmail.com>	2015-01-15 21:11:52
ui	do not focus one input any more	Oleg Verych <olecom@gmail.com>	2015-01-15 21:10:38
backend	mongodb accounting of users	Oleg Verych <olecom@gmail.com>	2015-01-15 21:10:02
ui	handle 'login' as login and not 'add user'	Oleg Verych <olecom@gmail.com>	2015-01-15 21:07:34
ui	'err pass' event: add and handle error page for failed login	Oleg Verych <olecom@gmail.com>	2015-01-15 21:05:21
ui	build: SHA1 (cc719b54b73fc01dcce1163e6a7d5cbef12ac5e3)	Oleg Verych <olecom@gmail.com>	2015-01-15 17:55:29
ui	SHA1 password hashing	Oleg Verych <olecom@gmail.com>	2015-01-15 17:54:56
_start.sh	create subst patterns for all client JS files; replace 'main.js'	Oleg Verych <olecom@gmail.com>	2015-01-15 17:53:35
noop	whitespace indent of runSocketIO()	Oleg Verych <olecom@gmail.com>	2015-01-15 17:15:18
mongodb	launching and connecting	Oleg Verych <olecom@gmail.com>	2015-01-15 16:58:40
backend	graceful shutdown logic (i.e. for mongodb)	Oleg Verych <olecom@gmail.com>	2015-01-15 16:56:47
gitignore	move stuff into '/data'	Oleg Verych <olecom@gmail.com>	2015-01-15 16:50:12
ui	build: done #1, #2, 3 (63face7707fd4d38d01ed5a04b4f3d9e5b0bbc40)	Oleg Verych <olecom@gmail.com>	2015-01-15 14:42:17
done	#1, #3, 4	Oleg Verych <olecom@gmail.com>	2015-01-15 14:41:12
ui	002: abaebab70d1e5e7756c4ac035ae167caaa1b597	Oleg Verych <olecom@gmail.com>	2015-01-15 14:34:12
add	password for client's user/pass accounting	Oleg Verych <olecom@gmail.com>	2015-01-15 14:32:51
ui	handle connection states (connect, disconnect, reconnect)	Oleg Verych <olecom@gmail.com>	2015-01-15 14:25:06
backend	add chatroom from example	Oleg Verych <olecom@gmail.com>	2015-01-15 14:13:14
ui	001: index.htm for d671645d81c3d5a24cea45edeac97107efc6b7e9	Oleg Verych <olecom@gmail.com>	2015-01-15 14:11:47
_start.sh	helper script to generate minified all-in-one 'index.htm'	Oleg Verych <olecom@gmail.com>	2015-01-15 14:09:44
initial	boilerplate based on socket.io chat example	Oleg Verych <olecom@gmail.com>	2015-01-15 14:03:17

SHA1 ID: db92cf988ccd8dc695f8e358dcc55030d5ab77db

Search: состояние, содержащее: Точно, Во всех полях

Diff: app.js

```
function the_end(code, res){
+   i = '$ application exit with code: ' + (code ? code : 0)
+   proc.nextTick(function(){
-     con.log('$ application exit with code: ' + (code ? code : 0));
+     con.log(i);
      proc.exit(code ? code : 0);
    });
-   return res.end();
+   return res.end(i);
}
```