

1. The following request and response header log was generated by running socat on the Index or '/' page:

```
Request:
> 2015/03/26 23:50:28.424462 length=323 from=0 to=322
GET / HTTP/1.1\r
Host: localhost:3001\r
User-Agent: Mozilla/5.0 (X11; Ubuntu; Linux i686; rv:34.0) Gecko/20100101 Firefox/34.0\r
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8\r
Accept-Language: en-US,en;q=0.5\r
Accept-Encoding: gzip, deflate\r
Connection: keep-alive\r
If-None-Match: W/"24c-32554b76"\r
\r
Response:
< 2015/03/26 23:50:28.667242 length=261 from=0 to=260
HTTP/1.1 304 Not Modified\r
X-Powered-By: Express\r
ETag: W/"24c-32554b76"\r
set-cookie: connect.sid=s%3AsRQBCWwT6sT_-R71aoxvB0h1C1h40GgM.NphZwYA33FSppMnIJs9LogOdmTgksYkKHl4OjyqZru8; Path=/; HttpOnly\r
Date: Fri, 27 Mar 2015 03:50:28 GMT\r
Connection: keep-alive\r
```

Following response and request headers were generated by typing **localhost:3001** in the address bar. The http requests were generated because the browser sent a GET request to the express server.

The server ran the following code to generate these request and response headers:

```
router.get('/', function(req, res) {
  if (req.session.username) {
    res.redirect("/notes");
  } else {
    res.render('index', { title: 'COMP 2406 AJAX Notes Demo',
                        error: req.query.error });
  }
});
```

There was a 404 error generated on the **"/"** page, for the missing favicon.ico. This error was fixed by placing a random ico file in the **ajax-notes/public** folder.

2. The following are the request/response headers for /GetNotes

```
Request:
> 2015/03/26 22:47:42.444235 length=402 from=0 to=401
GET /getnotes HTTP/1.1\r
Host: localhost:3001\r
User-Agent: Mozilla/5.0 (X11; Ubuntu; Linux i686; rv:34.0) Gecko/20100101 Firefox/34.0\r
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8\r
Accept-Language: en-US,en;q=0.5\r
Accept-Encoding: gzip, deflate\r
Cookie: connect.sid=s%3AfRDr-oOdTtv7asRY37pmQqvFTGaEL9mK.Vwx%2BbRMJScOAtur4MvbQvBrnlCKVWHJ0EWaF30nanM\r
Connection: keep-alive\r
\r
Response:
< 2015/03/26 22:47:42.449643 length=283 from=0 to=282
HTTP/1.1 200 OK\r
X-Powered-By: Express\r
Content-Type: application/json; charset=utf-8\r
Content-Length: 91\r
ETag: W/"5b-1391554"\r
Date: Fri, 27 Mar 2015 02:47:42 GMT\r
Connection: keep-alive\r
\r
[{"title":"Untitled","owner":"as","content":"No content","_id":"5514c1d45ac921880d952d07"}]
```

Following socat log was generated by typing *localhost:3001/getNotes* in the address bar.

The server ran the following code in index.js:

```
router.get('/getNotes', function(req, res) {
  var username = req.session.username;

  var renderNotes = function(err, notes) {
    if (err) {
      notes = [{"title": "Couldn't get notes",
        "owner": username,
        "content": "Error fetching notes!"}];
    }
    res.send(notes);
  }

  if (username) {
    notesCollection.find({owner: username}).toArray(renderNotes);
  } else {
    res.send([{"title": "Not Logged In",
      "owner": "None",
      "content": "Nobody seems to be logged in!"}]);
  }
});
```

The server checks if there is a user logged in with the current username, if it's true, a *“find”* is run on the *notesCollection* database and an array of all available notes for that user are sent to be displayed by *getAndListNotes in notes.js*.

3. The following are the request/response headers for /newnote

```
Request:
> 2015/03/26 22:52:21.507950 length=476 from=0 to=475
POST /newNote HTTP/1.1\r
Host: localhost:3001\r
User-Agent: Mozilla/5.0 (X11; Ubuntu; Linux i686; rv:34.0) Gecko/20100101 Firefox/34.0\r
Accept: */*\r
Accept-Language: en-US,en;q=0.5\r
Accept-Encoding: gzip, deflate\r
X-Requested-With: XMLHttpRequest\r
Referer: http://localhost:3001/notes\r
Cookie: connect.sid=s%3AfRDr-
oOdTtv7asRY37pmQqvFTGaEL9mK.Vwx%2BbRMJScOAtur4MvbQvBrnlCKVWHJ0EWaF30nanM\r
Connection: keep-alive\r
Pragma: no-cache\r
Cache-Control: no-cache\r
Content-Length: 0\r
\r
Response:
< 2015/03/26 22:52:21.512333 length=170 from=0 to=169
HTTP/1.1 200 OK\r
X-Powered-By: Express\r
Content-Type: application/json; charset=utf-8\r
Content-Length: 26\r
Date: Fri, 27 Mar 2015 02:52:21 GMT\r
Connection: keep-alive\r
\r
< 2015/03/26 22:52:21.514102 length=26 from=170 to=195
"5514c5e5889dc41a0e41543f"> 2015/03/26 22:52:24.994640 length=488 from=476 to=963
GET /getNotes HTTP/1.1\r
Host: localhost:3001\r
User-Agent: Mozilla/5.0 (X11; Ubuntu; Linux i686; rv:34.0) Gecko/20100101 Firefox/34.0\r
Accept: application/json, text/javascript, */*; q=0.01\r
Accept-Language: en-US,en;q=0.5\r
Accept-Encoding: gzip, deflate\r
X-Requested-With: XMLHttpRequest\r
Referer: http://localhost:3001/notes\r
Cookie: connect.sid=s%3AfRDr-
oOdTtv7asRY37pmQqvFTGaEL9mK.Vwx%2BbRMJScOAtur4MvbQvBrnlCKVWHJ0EWaF30nanM\r
Connection: keep-alive\r
If-None-Match: W/"5b-1391554"\r
\r
< 2015/03/26 22:52:24.999157 length=465 from=196 to=660
HTTP/1.1 200 OK\r
X-Powered-By: Express\r
Content-Type: application/json; charset=utf-8\r
Content-Length: 271\r
ETag: W/"10f-c8ee2252"\r
Date: Fri, 27 Mar 2015 02:52:24 GMT\r
Connection: keep-alive\r
\r
[{"title":"Untitled","owner":"as","content":"No
content","_id":"5514c1d45ac921880d952d07"},{"title":"Untitled","owner":"as","content":"No
content","_id":"5514c580889dc41a0e41543e"},{"title":"Untitled","owner":"as","content":"No
content","_id":"5514c5e5889dc41a0e41543f"}] 2015/03/26 22:52:25.000397 length=1 from=661 to=661
]
```

Interaction with the page was made by logging in and then clicking on the button **“New Note”** button, which generated the following get and post requests. Once the new note button is pressed, it calls the ***newNote function*** in ***public/javascript/notes.js***. The function makes a post request to the ***/newNote*** in ***index.js***.

The server ran the following code in index.js:

```
router.post('/newNote', function(req, res) {
  var username = req.session.username
  var newNote;

  var reportInserted = function(err, notesInserted) {
    if (err) {
      res.send("ERROR: Could not create a new note");
    } else {
      res.send(notesInserted[0]._id);
    }
  }

  if (username) {
    newNote = {title: "Untitled",
      owner: username,
      content: "No content"};

    notesCollection.insert(newNote, reportInserted);
  } else {
    res.send("ERROR: Not Logged In");
  }
});
```

The **/newNote** first checks if there is a username logged in, if that is true a note is created with a title of **“Untitled”** and content of **“No content”** and inserted in the **notesCollection** which later calls a callback function called **reportInsterted**. This callback function checks if there are errors, if so sends the error string. If not the note inserted along with its id is sent to be either listed or edited.

4. Post request header from /updateNote

Clicking on the update button generates the following socat code:

```
Request:
POST /updateNote HTTP/1.1\r
Host: localhost:3001\r
User-Agent: Mozilla/5.0 (X11; Ubuntu; Linux i686; rv:34.0) Gecko/20100101 Firefox/34.0\r
Accept: */*\r
Accept-Language: en-US,en;q=0.5\r
Accept-Encoding: gzip, deflate\r
Content-Type: application/x-www-form-urlencoded; charset=UTF-8\r
X-Requested-With: XMLHttpRequest\r
Referer: http://localhost:3001/notes\r
Content-Length: 47\r
Cookie: connect.sid=s%3AfRDr-
oOdTtv7asRY37pmQqvftGaEL9mK.Vwxb%2BbRMJScOAtur4MvbQvBrnlCKVWHJ0EWaF30nanM\r
Connection: keep-alive\r
Pragma: no-cache\r
Cache-Control: no-cache\r
\r
Response:
id=5514c5e5889dc41a0e41543f&title=as&content=as< 2015/03/26 22:52:25.007324 length=163 from=662 to=824
HTTP/1.1 200 OK\r
X-Powered-By: Express\r
Content-Type: text/html; charset=utf-8\r
Content-Length: 16\r
Date: Fri, 27 Mar 2015 02:52:25 GMT\r
Connection: keep-alive\r
```

```

\r
< 2015/03/26 22:52:25.045601 length=16 from=825 to=840
update succeeded> 2015/03/26 22:52:25.047979 length=490 from=1555 to=2044
GET /getNotes HTTP/1.1\r
Host: localhost:3001\r
User-Agent: Mozilla/5.0 (X11; Ubuntu; Linux i686; rv:34.0) Gecko/20100101 Firefox/34.0\r
Accept: application/json, text/javascript, */*; q=0.01\r
Accept-Language: en-US,en;q=0.5\r
Accept-Encoding: gzip, deflate\r
X-Requested-With: XMLHttpRequest\r
Referer: http://localhost:3001/notes\r
Cookie: connect.sid=s%3AfRDr-
oOdTtv7asRY37pmQqvFTGaEL9mK.Vwx%2BbRMJScOAtur4MvbQvBrnlCKVWHJ0EWaF30nanM\r
Connection: keep-alive\r
If-None-Match: W/"10f-c8ee2252"\r
\r
< 2015/03/26 22:52:25.052401 length=195 from=841 to=1035
HTTP/1.1 200 OK\r
X-Powered-By: Express\r
Content-Type: application/json; charset=utf-8\r
Content-Length: 257\r
ETag: W/"101-b83452d2"\r
Date: Fri, 27 Mar 2015 02:52:25 GMT\r
Connection: keep-alive\r
\r
< 2015/03/26 22:52:25.088544 length=257 from=1036 to=1292
[{"title":"Untitled","owner":"as","content":"No
content","_id":"5514c1d45ac921880d952d07"},{"title":"Untitled","owner":"as","content":"No
content","_id":"5514c580889dc41a0e41543e"},{"_id":"5514c5e5889dc41a0e41543f","content":"as","owner":"as","title":
"as"}]]

```

editNote function in ***notes.js*** calls the ***updateNoteOnSever*** when the user clicks on the ***update note*** button. This function makes a post request to ***/updateNote*** function in ***index.js***. Server then runs the following code:

```

router.post('/updateNote', function(req, res) {
  var username = req.session.username;
  var id = req.body.id;
  var title = req.body.title;
  var content = req.body.content;

  var checkUpdate = function(err, result) {
    if (err) {
      res.send("ERROR: update failed");
    } else {
      res.send("update succeeded");
    }
  }

  if (username) {

    if (id && title && content) {

      notesCollection.update({_id: ObjectID(id)},

        {$set: {title: title,
          content: content}},
        checkUpdate);

    } else {
      res.send("ERROR: bad parameters");
    }
  } else {
    res.send("ERROR: not logged in");
  }
});

```

This post request first checks if the user is logged in with the correct username, then later it also checks if the id, title and content belongs to the current user with the session username. Later an **“.update”** on the **notesCollection** using the notes id is made, and the title and content of the notes is replaced or **“\$set”** with the title and content the user types in. The update function calls the callback function – **checkUpdate**. This callback prints a success or failure string regarding if there were errors or not.

5. The following are the request/response headers for `/javascripts/notes.js`

The following socat code is generated by typing
`localhost:3001/javascripts/notes.js`

```
Resquest:
> 2015/03/26 23:36:48.560060 length=497 from=4629 to=5125
GET /javascripts/notes.js HTTP/1.1\r
Host: localhost:3001\r
User-Agent: Mozilla/5.0 (X11; Ubuntu; Linux i686; rv:34.0) Gecko/20100101 Firefox/34.0\r
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8\r
Accept-Language: en-US,en;q=0.5\r
Accept-Encoding: gzip, deflate\r
Cookie:
connect.sid=s%3AuLmAA1oredF6IOdR25XugNs71D1mnSHB.Rzig%2FAWQ0VCmqCdHOLEcesHqaHRXkpPpOceaZC1pmSU\r
Connection: keep-alive\r
If-Modified-Since: Mon, 09 Mar 2015 01:05:38 GMT\r
If-None-Match: W/"d90-56881568"\r
\r
Response:
< 2015/03/26 23:36:48.568692 length=239 from=4138 to=4376
HTTP/1.1 304 Not Modified\r
X-Powered-By: Express\r
Accept-Ranges: bytes\r
Date: Fri, 27 Mar 2015 03:36:48 GMT\r
Cache-Control: public, max-age=0\r
Last-Modified: Mon, 09 Mar 2015 01:05:38 GMT\r
ETag: W/"d90-56881568"\r
Connection: keep-alive\r
\r
```

The server just makes a GET request to the **`javascripts/notes.js`** which retrieves the code from **`notes.js`** from **`public/javascripts/notes.js`** directory and displays it on the screen.