**rExercise02\_02\_01 – Step 1**

terminal

npm install cookie-parser --save

npm install oauth --save

------------------------------------

config.json

{

"port": 8080,

"request\_token\_url":   
 "https://api.twitter.com/oauth/request\_token",

"access\_token\_url":   
 "https://api.twitter.com/oauth/access\_token",

"authorize\_url": "https://api.twitter.com/oauth/authorize",

"consumer\_key": "f2sHurvY8Asq2bbXYwQDS05vE",

"consumer\_secret":   
 "BnsqYkR8sLZahUizwldIXz3qCN6Jso11gFPlSls3XNhTr1N1j6",

"oauth\_version": "1.0",

"oauth\_signature": "HMAC-SHA1",

"oauth\_callback": "http://www.example.com/callback"

}

------------------------------------

index.js

var express = require('express');

var config = require('./config.json');

var app = express();

app.get("/", function(req, res) {

res.send("<h3>Hello, world!</h3>");

});

app.listen(config.port, function() {

console.log("Server listening on localhost:%s", config.port);

});

------------------------------------

index.js

var url = require('url');

var express = require('express');

app.use(require('cookie-parser')());

**Exercise02\_02\_01 – Step 2**

index.js

app.get("/auth/twitter", function(req, res) {

res.send("This route will get a request token and redirect   
 the user to the twitter sign in.");

});

------------------------------------

index.js

app.get("/auth/callback", function(req, res) {

res.send("This route will handle OAuth as a callback.");

});

------------------------------------

index.js

app.get("/auth/twitter", function   
 redirectToTwitterLoginPage(req, res) {

------------------------------------

authenticator.js

var OAuth = require('oauth').OAuth;

var config = require('./config.json');

------------------------------------

index.js

var express = require('express');

var authenticator = require('./authenticator.js');

------------------------------------

authenticator.js

var oauth = new OAuth(

config.request\_token\_url,

config.access\_token\_url,

config.consumer\_key,

config.consumer\_secret,

config.oauth\_version,

config.oauth\_callback,

config.oauth\_signature

);

authenticator.js

var twitterCredentials = {

oauth\_token: "",

oauth\_token\_secret: ""

}

------------------------------------

authenticator.js

module.exports = {

redirectToTwitterLoginPage: function(req, res) {

res.send("Using our authenticator module.");

}

};

------------------------------------

index.js

app.get("/auth/twitter",   
 authenticator.redirectToTwitterLoginPage);

------------------------------------

authenticator.js

redirectToTwitterLoginPage: function(req, res) {

oauth.getOAuthRequestToken(function(error, oauth\_token,   
 oauth\_token\_secret, results) {

if (error) {

console.log(error);

res.send("Authentication failed!");

}

else {

res.send("Authentication successful!");

}

});

}

------------------------------------

authenticator.js

else {

twitterCredentials.oauth\_token = oauth\_token;

twitterCredentials.oauth\_token\_secret =   
 oauth\_token\_secret;

res.send("Credentials stored");

authenticator.js

twitterCredentials.oauth\_token\_secret =   
 oauth\_token\_secret;

res.redirect(config.authorize\_url +   
 '?oauth\_token=' + oauth\_token);

**Exercise02\_02\_01 – Step 3**

terminal

ngrok http 8080

------------------------------------

config.json

"oauth\_callback": "http://15c8db02.ngrok.io/request\_token"

------------------------------------

index.js

app.listen(config.port, function() {

console.log("Server listening on localhost:%s",   
 config.port);

console.log("OAuth callback: " +

url.parse(config.oauth\_callback).hostname +

url.parse(config.oauth\_callback).path);

});

------------------------------------

index.js

app.get(url.parse(config.oauth\_callback).path, function(req,   
 res) {

res.send("This route will handle OAuth as a callback.");

});

------------------------------------

index.js

app.get(url.parse(config.oauth\_callback).path, function(req, res) {

authenticator.authenticate(req, res, function(err) {

if (err) {

console.log(err);

res.sendStatus(401);

}

else {

res.send('Authentication successful!');

}

});

});

authenticator.js

},

authenticate: function(req, res, callback) {

return callback("Testing authenticate failure.");

}

------------------------------------

authenticator.js

authenticate: function(req, res, callback) {

if (!(twitterCredentials.oauth\_token &&   
 twitterCredentials.oauth\_token\_secret &&   
 req.query.oauth\_verifier)) {

return callback("Request does not have all required   
 keys.");

}

twitterCredentials.oauth\_token = "";

twitterCredentials.oauth\_token\_secret = "";

return callback();

}

**Exercise02\_02\_01 – Step 4**

authenticator.js

twitterCredentials.oauth\_token = "";

twitterCredentials.oauth\_token\_secret = "";

oauth.getOAuthAccessToken(  
 twitterCredentials.oauth\_token,   
 twitterCredentials.oauth\_token\_secret,   
 req.query.oauth\_verifier,   
 function(error, oauth\_access\_token,   
 oauth\_access\_token\_secret, results) {

if (error) {

return callback(error);

}

});

------------------------------------

authenticator.js

if (error) {

return callback(error);

}

oauth.get('https://api.twitter.com/1.1  
 /account/verify\_credentials.json',   
 oauth\_access\_token, oauth\_access\_token\_secret,   
 function(error, data) {

if (error) {

console.log(error);

return callback(error);

}

data = JSON.parse(data);

console.log(data);

callback();

});

------------------------------------

authenticator.js

var twitterCredentials = {

oauth\_token: "",

oauth\_token\_secret: "",

access\_token: "",

access\_token\_secret: "",

twitter\_id: ""

}

authenticator.js

data = JSON.parse(data);

twitterCredentials.access\_token =   
 oauth\_access\_token;

twitterCredentials.access\_token\_secret =   
 oauth\_access\_token\_secret;

twitterCredentials.twitter\_id = data.id\_str;

console.log(twitterCredentials);

callback();

**Exercise02\_02\_01 – Step 5**

authenticator.js

module.exports = {

get: function(url, access\_token, oauth\_access\_token\_secret,   
 callback) {

oauth.get.call(oauth, url, access\_token,   
 oauth\_access\_token\_secret, callback);

},

post: function(url, access\_token, oauth\_access\_token\_secret,   
 body, callback) {

oauth.post.call(oauth, url, access\_token,   
 oauth\_access\_token\_secret, body, callback);

},

------------------------------------

authenticator.js

getCredentials: function() {

return twitterCredentials;

},

------------------------------------

index.js

app.get('/tweet', function(req, res) {

credentials = authenticator.getCredentials();

if (!credentials.access\_token ||   
 !credentials.access\_token\_secret) {

return res.sendStatus(401);

}

res.sendStatus(200);

});

index.js

var url =   
 'https://api.twitter.com/1.1/statuses/update.json';

authenticator.post(url, credentials.access\_token,   
 credentials.access\_token\_secret,

{

status: "Hello there Twitter RESTful API"

},

function (error, data) {

if (error) {

return res.status(400).send(error);

}

res.send('Tweet successful!');

});

**Exercise02\_02\_01 – Step 6**

index.js

var express = require('express');

var querystring = require('querystring');

------------------------------------

index.js

app.get('/search', function (req, res) {  
 var credentials = authenticator.getCredentials();  
 if (!credentials.access\_token || !credentials.access\_token\_secret)   
{  
 return res.sendStatus(401);  
 }  
 var url = 'https://api.twitter.com/1.1/search/tweets.json';

------------------------------------

index.js

var url = 'https://api.twitter.com/1.1/search/tweets.json';  
 authenticator.get(url, credentials.access\_token,   
 |credentials.access\_token\_secret,

function (error, data) {

if (error) {

return res.status(400).send(error);

}

res.send(data);

});

------------------------------------

index.js

var url = 'https://api.twitter.com/1.1/search/tweets.json';

var query = querystring.stringify({ q: ‘BMW' });

url += "?" + query;

**Exercise02\_02\_01 – Step 7**

index.js

app.get('/friends', function (req, res) {

var credentials = authenticator.getCredentials();

if (!credentials.access\_token ||   
 !credentials.access\_token\_secret) {

return res.sendStatus(401);

}

var url = 'https://api.twitter.com/1.1/friends/list.json';

authenticator.get(url, credentials.access\_token,   
 credentials.access\_token\_secret,

function (error, data) {

if (error) {

return res.status(400).send(error);

}

res.send(data);

});

------------------------------------

index.js

var url = 'https://api.twitter.com/1.1/friends/list.json';

if (req.query.cursor) {

url += "?" + querystring.stringify({ cursor:   
 req.query.cursor });

}

------------------------------------

Browser URL

http://localhost:8080/friends?cursor=1553263176318567747

**Exercise02\_02\_01 – Step 8**

terminal

npm install async –save

var querystring = require('querystring');

var async = require('async');

------------------------------------

index.js

app.get('/allfriends', function(req, res) {

var credentials = authenticator.getCredentials();

});

------------------------------------

index.js

app.get('/allfriends', function(req, res) {

async.waterfall([

// get friends ids

function(callback) {

},

// lookup friends data

function(ids, callback) {

}

]);

});

------------------------------------

index.js

async.waterfall([

// get friends ids

function(callback) {

var cursor = -1;

var ids = [];

console.log('ids.length: ' + ids.length);

index.js

console.log('ids.length: ' + ids.length);

async.whilst(function() {

return cursor != 0;

}, function(callback) {

var url =   
 'https://api.twitter.com/1.1/friends/ids.json';

url += "?" + querystring.stringify({ user\_id:   
 credentials.twitter\_id, cursor: cursor});

authenticator.get(url, credentials.access\_token,   
 credentials.access\_token\_secret,

function(error, data) {

if (error) {

return res.status(400).send(error);

}

data = JSON.parse(data);

cursor = data.next\_cursor\_str;

ids = ids.concat(data.ids);

callback();

});

}, function(error) {

if (error) {

return res.status(500).send(error);

}

console.log(ids);

callback(null, ids);

});

},

// lookup friends data

function(ids, callback) {

}

]);

------------------------------------

index.js

function(ids, callback) {

var getHundredIds = function(i) {

return ids.slice(100\*i, Math.min(ids.length,   
 100\*(i+1)));

}

var requestsNeeded = Math.ceil(ids.length/100);

}

index.js

var requestsNeeded = Math.ceil(ids.length/100);

async.times(requestsNeeded, function(n, next) {

var url =   
 'https://api.twitter.com/1.1/users/lookup.json';

url += "?" + querystring.stringify(  
 { user\_id: getHundredIds(n).join(',')});

authenticator.get(url, credentials.access\_token,   
 credentials.access\_token\_secret,

function(error, data) {

if (error) {

return res.status(400).send(error);

}

var friends = JSON.parse(data);

next(null, friends);

});

},

function(err, friends) {

friends = friends.reduce(function(previousValue,   
 currentValue, currentIndex, array) {

return previousValue.concat(currentValue);

}, []);

friends.sort(function(a, b) {

return   
 a.name.toLowerCase().  
 localeCompare(b.name.toLowerCase())

});

res.send(friends);

console.log('ids.length: ' + ids.length);

});

}