**Exercise02\_04\_01 – Step 1**

terminal

npm install mongodb --save

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

index.js

var async = require('async');

var MongoClient = require('mongodb').MongoClient;

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

index.js

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

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

**Exercise02\_04\_01 – Step 2**

storage.js

var MongoClient = require('mongodb').MongoClient;

var url = 'mongodb://localhost:27017';

var dbName = 'twitter\_notes';

var database;

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

storage.js

module.exports = {

connect: function() {

MongoClient.connect(url, function(err, client) {

if (err) {

return console.log('Error: ' + err);

}

database = client.db(dbName);

console.log('Connected to database: ' + dbName);

});

}

}

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

index.js

var app = express();

storage.connect();

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

storage.js

},

connected: function () {

return typeof database != 'undefined';

}

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

index.js

if (!storage.connected()) {

console.log('Loading friends from Twitter');

renderMainPageFromTwitter(req, res);

}

**Exercise02\_04\_01 – Step 3**

storage.js

},

insertFriends: function(friends) {

database.collection('friends').insert(friends,   
 function(err) {

if (err) {

console.log('Cannot insert friends into   
 database.');

}

});

}

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

index.js

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

var credentials = authenticator.getCredentials();

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

return res.redirect('/login');

}

if (!storage.connected()) {

console.log('Loading friends from Twitter');

renderMainPageFromTwitter(req, res);

}

});

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

index.js

res.render('index', { friends: friends });

if (storage.connected) {

storage.insertFriends(friends);

}

**Exercise02\_04\_01 – Step 4**

storage.js

},

getFriends: function(userId, callback) {

var cursor = database.collection('friends').  
 find({ for\_user: userId});

cursor.toArray(callback);

}

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

index.js

console.log('Loading friends from MongoDB');

storage.getFriends(credentials.twitter\_id, function(err,   
 friends) {

if (err) {

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

}

if (friends.length > 0) {

console.log('Friends successfully loaded from   
 MongoDB');

friends.sort(function(a, b) {

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

});

res.render('index', { friends: friends});

}

else {

console.log('Loading friends from Twitter');

renderMainPageFromTwitter(req, res);

}

});

**Exercise02\_04\_01 – Step 5**

storage.js

},

deleteFriends: function() {

database.collection('friends').remove(( {} ),   
 function(err) {

if (err) {

console.log('Cannot remove friends from   
 database.');

}

});

}

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

index.js

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

authenticator.clearCredentials();

if (storage.connected()) {

console.log('Deleting friends collection on logout');

storage.deleteFriends();

}

res.redirect('/login');

});

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

index.js

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

authenticator.clearCredentials();

if (storage.connected()) {

console.log('Deleting friends collection on login');

storage.deleteFriends();

}

res.render('login');

});

index.js

app.set('view engine', 'ejs');  
  
setInterval(function() {  
 if (storage.connected()) {  
 console.log('Clearing MongoDB cache');  
 storage.deleteFriends();  
 }  
}, 1000 \* 10);

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

index.js

}, 1000 \* 60 \* 5);