NodeJS & Mongodb CRUD 2

Step 1. update Express and Express scaffolding

```
$ npm update -g express
$ npm update -g generator-express
```

Step 2. create Express project

```
$ express nodetest2
```

Step 3. edit package.json

add `mongodb` and `mongoskin` package, the code like this:

```
{
    "name": "application-name",
    "version": "0.0.1",
    "private": true,
    "scripts": {
        "start": "node ./bin/www"
    },
    "dependencies": {
        "express": "~4.0.0",
        "static-favicon": "~1.0.0",
        "morgan": "~1.0.0",
        "cookie-parser": "~1.0.1",
        "body-parser": "~1.0.0",
        "debug": "~0.7.4",
        "jade": "*",
        "mongodb": "*",
        "mongoskin": "*"
    }
}
```

Step 4. install

and then install

```
$ npm install
```

Step 5. create data folder

create a data folder, this folder will store mongo database

```
$ mkdir data
```

Step 6. edit layout.jade

include `jquery` and `global` javascript.

and you can download this css file [click]

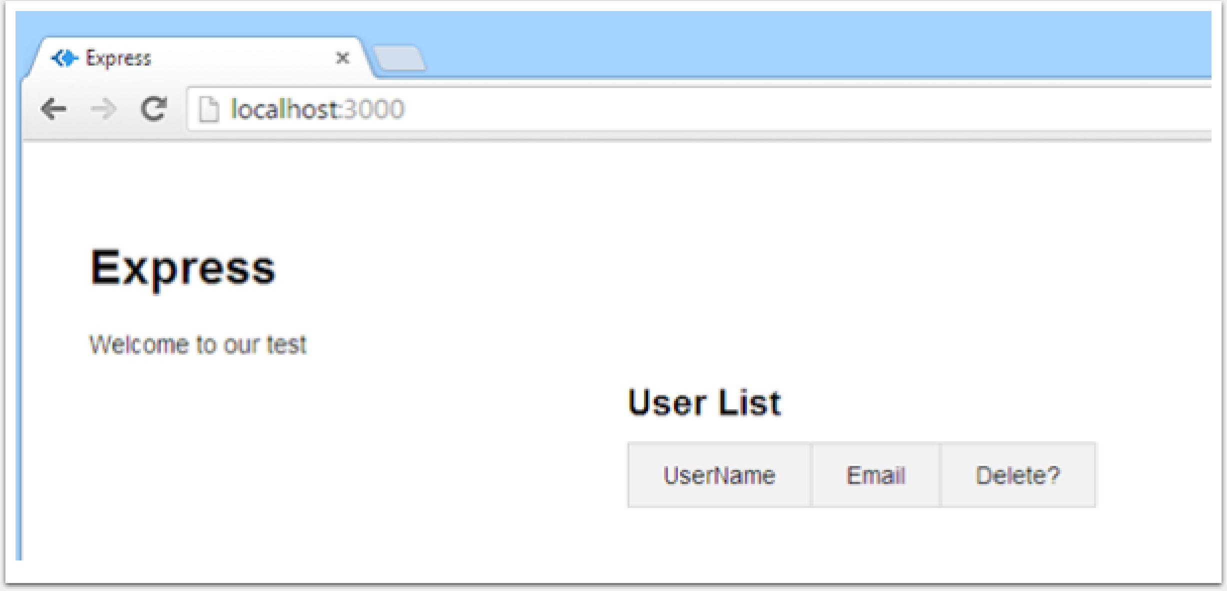
```
doctype html
html
    head
        title= title
        link(rel='stylesheet', href='/stylesheets/style.css')
    body
        block content
        script(src='http://ajax.googleapis.com/ajax/libs/jquery/2.0.3/jquery.min.js')
        script(src='/javascripts/global.js')
```

Step 7. edit index.jade

```
extends layout
block content
    h1= title
    p Welcome to our test
    // Wrapper
    #wrapper
        // USER LIST
        h2 User List
        #userList
            table
                thead
                    th UserName
                    th Email
                    th Delete?
                tbody
        // /USER LIST
    // /WRAPPER
```

and then start application

```
$ npm start
```



Express

Welcome to our test

User List

UserName	Email	Delete?

Step 8. start up database

```
$ mongod --dbpath c:\node\nodetest2\data
```

connect to mongodb, open new terminal and input this command:

```
$ mongo
```

create new database

```
$ use nodedatabase2
```

insert data

```
$ db.userlist.insert({'username' : 'test1','email' : 'test1@test.com','fullname' :
'Bob Smith','age' : 27,'location' : 'San Francisco','gender' : 'Male'})
```

Step 9. App.js database

create database

```
var express = require('express');
var path = require('path');
var favicon = require('static-favicon');
var logger = require('morgan');
var cookieParser = require('cookie-parser');
var bodyParser = require('body-parser');
// Database
var mongo = require('mongoskin');
var db = mongo.db("mongodb://localhost:27017/nodetest2", {native_parser:true});
```

Step 10. App.js accessible to our router

```
// Make our db accessible to our router
app.use(function(req,res,next){
    req.db = db;
    next();
});
app.use('/', routes);
app.use('/users', users);
```

Step 11. edit user.js, list user

```
var express = require('express');
var router = express.Router();
/*
 * GET userlist.
 */
router.get('/userlist', function(req, res) {
    var db = req.db;
    db.collection('userlist').find().toArray(function (err, items) {
        res.json(items);
    });
});
module.exports = router;
```

Step 12. edit global.js

create user table

```
// Userlist data array for filling in info box
var userListData = [];
// DOM Ready =============================================================
$(document).ready(function() {
    // Populate the user table on initial page load
    populateTable();
});
// Functions =============================================================
// Fill table with data
function populateTable() {
    // Empty content string
    var tableContent = '';
    // jQuery AJAX call for JSON
    $.getJSON( '/users/userlist', function( data ) {
        // For each item in our JSON, add a table row and cells to the content string
        $.each(data, function(){
            tableContent += '<tr>';
            tableContent += '<td><a href="#" class="linkshowuser" rel="' + this.username + '" title="Show Details">' + this.username + '</a></td>';
            tableContent += '<td>' + this.email + '</td>';
            tableContent += '<td><a href="#" class="linkdeleteuser" rel="' + this._id + '">delete</a></td>';
            tableContent += '</tr>';
        });
        // Inject the whole content string into our existing HTML table
        $('#userList table tbody').html(tableContent);
    });
};
```

Step 13. show user info

edit `global.js`

```
// Show User Info
function showUserInfo(event) {
    // Prevent Link from Firing
    event.preventDefault();
    // Retrieve username from link rel attribute
    var thisUserName = $(this).attr('rel');
    // Get Index of object based on id value
    var arrayPosition = userListData.map(function(arrayItem) { return arrayItem.username; }).indexOf(thisUserName);
    // Get our User Object
    var thisUserObject = userListData[arrayPosition];
    //Populate Info Box
    $('#userInfoName').text(thisUserObject.fullname);
    $('#userInfoAge').text(thisUserObject.age);
    $('#userInfoGender').text(thisUserObject.gender);
    $('#userInfoLocation').text(thisUserObject.location);
};
```

Step 14. edit index.jade

add user info

```
      // USER INFO
      #userInfo
          h2 User Info
          p
              strong Name:
              |  <span id='userInfoName'></span>
              br
              strong Age:
              |  <span id='userInfoAge'></span>
              br
              strong Gender:
              |  <span id='userInfoGender'></span>
              br
              strong Location:
              |  <span id='userInfoLocation'></span>
      // /USER INFO
```

Step 15. add user

```
        // ADD USER
        h2 Add User
        #addUser
            fieldset
                input#inputUserName(type='text', placeholder='Username')
                input#inputUserEmail(type='text', placeholder='Email')
                br
                input#inputUserFullname(type='text', placeholder='Full Name')
                input#inputUserAge(type='text', placeholder='Age')
                br
                input#inputUserLocation(type='text', placeholder='Location')
                input#inputUserGender(type='text', placeholder='gender')
                br
                button#btnAddUser Add User
        // /ADD USER
```

the index.jade complete code

```
extends layout
block content
  h1= title
  p Welcome to #{title}
  // Wrapper
  #wrapper
      // USER INFO
      #userInfo
          h2 User Info
          p
              strong Name:
              |  <span id='userInfoName'></span>
              br
              strong Age:
              |  <span id='userInfoAge'></span>
              br
              strong Gender:
              |  <span id='userInfoGender'></span>
              br
              strong Location:
              |  <span id='userInfoLocation'></span>
      // /USER INFO
      // USER LIST
      h2 User List
      #userList
          table
              thead
                  th UserName
                  th Email
                  th Delete?
              tbody
      // /USER LIST
      // ADD USER
      h2 Add User
      #addUser
          fieldset
              input#inputUserName(type='text', placeholder='Username')
              input#inputUserEmail(type='text', placeholder='Email')
              br
              input#inputUserFullname(type='text', placeholder='Full Name')
              input#inputUserAge(type='text', placeholder='Age')
              br
              input#inputUserLocation(type='text', placeholder='Location')
              input#inputUserGender(type='text', placeholder='gender')
              br
              button#btnAddUser Add User
      // /ADD USER
  // /WRAPPER
```

Step 16. add adduser router

edit `users.js`

```
/*
 * POST to adduser.
 */
router.post('/adduser', function(req, res) {
    var db = req.db;
    db.collection('userlist').insert(req.body, function(err, result){
        res.send(
            (err === null) ? { msg: '' } : { msg: err }
        );
    });
});
```

Step 17. add user button clicke event

edit `global.js`

```
// Add User button click
    $('#btnAddUser').on('click', addUser);
```

the `addUser` function

```
// Add User
function addUser(event) {
    event.preventDefault();
    // Super basic validation - increase errorCount variable if any fields are blank
    var errorCount = 0;
    $('#addUser input').each(function(index, val) {
        if($(this).val() === '') { errorCount++; }
    });
    // Check and make sure errorCount's still at zero
    if(errorCount === 0) {
        // If it is, compile all user info into one object
        var newUser = {
            'username': $('#addUser fieldset input#inputUserName').val(),
            'email': $('#addUser fieldset input#inputUserEmail').val(),
            'fullname': $('#addUser fieldset input#inputUserFullname').val(),
            'age': $('#addUser fieldset input#inputUserAge').val(),
            'location': $('#addUser fieldset input#inputUserLocation').val(),
            'gender': $('#addUser fieldset input#inputUserGender').val()
        }
        // Use AJAX to post the object to our adduser service
        $.ajax({
            type: 'POST',
            data: newUser,
            url: '/users/adduser',
            dataType: 'JSON'
        }).done(function( response ) {
            // Check for successful (blank) response
            if (response.msg === '') {
                // Clear the form inputs
                $('#addUser fieldset input').val('');
                // Update the table
                populateTable();
            }
            else {
                // If something goes wrong, alert the error message that our service
returned
                alert('Error: ' + response.msg);
            }
        });
    }
    else {
        // If errorCount is more than 0, error out
        alert('Please fill in all fields');
        return false;
    }
};
```

Step 18. delete use router

edit `user.js`

```
/*
 * DELETE to deleteuser.
 */
router.delete('/deleteuser/:id', function(req, res) {
    var db = req.db;
    var userToDelete = req.params.id;
    db.collection('userlist').removeById(userToDelete, function(err, result) {
        res.send((result === 1) ? { msg: '' } : { msg:'error: ' + err });
    });
});
```

open global.js, modify table's delete button

```
tableContent += '<td><a href="#"" class="linkdeleteuser" rel="' + this._id +
'">delete</a></td>';
```

Step 19. edit global.js, add delete click event

```
// Delete User link click
    $('#userList table tbody').on('click', 'td a.linkdeleteuser', deleteUser);
```

`deleteUser` function

```
// Delete User
function deleteUser(event) {
    event.preventDefault();
    // Pop up a confirmation dialog
    var confirmation = confirm('Are you sure you want to delete this user?');
    // Check and make sure the user confirmed
    if (confirmation === true) {
        // If they did, do our delete
        $.ajax({
            type: 'DELETE',
            url: '/users/deleteuser/' + $(this).attr('rel')
        }).done(function( response ) {
            // Check for a successful (blank) response
            if (response.msg === '') {
            }
            else {
                alert('Error: ' + response.msg);
            }
            // Update the table
            populateTable();
        });
    }
    else {
        // If they said no to the confirm, do nothing
        return false;
    }
};
```