

Workshop 3 – Express and jQuery

This workshop is worth 2.5%

You must demonstrate your completed workshop to your tutor in your lab to receive the marks.

In this workshop you'll set up a new express project, create a simple login page and use jQuery to send and receive data to and from the express server.

Task 1 – Setting up a basic express project

Follow the lecture slides to set up a new server side express project.

Task 2 – Routing

Create routes for the login page(form) and an endpoint that can be used to test a users credentials. Modularizing your code into multiple files is a good practice. Module.exports is a special object included in every Javascript file. Functions, objects or variables assigned to this object are exposed as a module in node. To access the properties of the module.exports object it must be imported to the code using the require() method. The full path to the modules location must be used when requiring the file.

.Eg

```
module.exports = function(app,path){  
    // app passes in the express object needed for the route.  
    // path passes in a path object needed to find the file. The  
    // path module is part of node and needs to be required in the  
    // server.js file  
    app.get('/mypage',function(req,res){  
        let filepath = path.resolve('./www/mypage.html');  
        res.sendFile(filepath);  
    });  
}
```

In your server.js file import the route with:

```
require('./routes/accountroute.js')(app,path);
```

The “routes” subdirectory is used here purely as a way to organise our code and keep “like” files in the same directory. It is your choice on where to locate the files.

Add an index route to express that returns a HTML login page. This page should have a login form that has an email and password field.

Add an account route to express that returns a HTML page that contains an image, a h2 title and some dummy account data.

Task 3 – Submitting Data

Add a submit button to the form that sends a post request to a login/attempt route via an ajax post request.

The function that this route uses should look at the post data and check it against a set of hardcoded values (3 users). If the inputs match one these values return the json {"ok":true}. If the inputs don't match then return {"ok":false,errors:{}}.

If the value of the return value(ok) is “false” then change the display property of a div from none to block to show a pre-configured error message.

```
<div id="errmsg" class="hidemessage">User credentials do not match</div>
```

CSS

```
hidemessage{
    display:none;
}

showmessage{
    display:block;
}
```