CREATE LOGIN SCREEN USING EXPRESS

The Express framework provides a router() method to create HTTP endpoints. Let's check out how to handle GET and POST requests using Express.

GET request

Handling GET request in Express is pretty straightforward. You have to create instances of express and router. Here is a small snippet to achieve the same.

```
const express = require("express");
const router = express.Router();
const app = express();

router.get('/handle',(request,response) => {
//code to perform particular action.
//To access GET variable use req.query() and req.params() methods.
});

// add router in the Express app.
app.use("/", router);
```

GET requests can be cached and remains in the browser history. This is why the GET method is not recommended to use for sensitive data (passwords, ATM pins, etc). You should GET requests to retrieve data from the server only.

POST Request:

Express requires an additional middleware module to extract incoming data of a POST request. This middleware is called **'body-parser.** We need to install it and configure it with Express instance.
You can install **body-parser** using the following command.

```
sudo npm install --save body-parser
```

You need to import this package into your project and tell Express to use this as middleware. Refer to this code for a reference.

```
const express = require("express");
const bodyParser = require("body-parser");
const router = express.Router();
const app = express();

//Here we are configuring express to use body-parser as middle-ware.
app.use(bodyParser.urlencoded({ extended: false }));
app.use(bodyParser.json());

router.post('/handle',(request,response) => {
//code to perform particular action.
//To access POST variable use req.body()methods.
console.log(request.body);
});

// add router in the Express app.
app.use("/", router);
```

In this way, you can handle the GET and POST request in the Express framework.

et's create a new Node.js project.

```
npm init --y
```

Let's install the dependencies.

```
npm i --S express body-parser
```

Next, let's create an Express web server.

```
const express = require("express");
const bodyParser = require("body-parser");
const router = express.Router();
const app = express();

app.listen(3000,() => {
console.log("Started on PORT 3000");
})
```

Run Server using the command: ***node server.js***
We will use Express Router to handle the routes of the app. So when users request the app from a web browser, we will serve the HTML file.

```
router.get('/',function(req,res){
res.sendfile("index.html");
});
```

When the user clicks on the log-in button on the HTML page we will POST the request to Server and get the response.

```
router.post('/login',function(req,res){
var user_name = req.body.user;
var password = req.body.password;
console.log("User name = "+user_name+", password is "+password);
res.end("yes");
});
```

Here is the complete server.js file code.

```
const express = require("express");
const bodyParser = require("body-parser");
const router = express.Router();
const app = express();
// add router in express app
app.use("/",router);

app.use(bodyParser.urlencoded({ extended: false }));
app.use(bodyParser.json());

router.get('/',(req, res) => {
res.sendfile("index.html");
});

router.post('/login',(req, res) => {
var user_name = req.body.user;
var password = req.body.password;
console.log("User name = "+user_name+", password is "+password);
res.end("yes");
});

app.listen(3000,() => {
console.log("Started on PORT 3000");
})
```

On the front-end, we will create a simple HTML file and call the POST request using Ajax. Here is the code of the index.html page.

```
<html>
<head>
<script src="//ajax.googleapis.com/ajax/libs/jquery/2.1.1/jquery.min.js">  </script>
<script>
```

```
    $(document).ready(function(){
      var user,pass;
      $("#submit").click(function(){
        user=$("#user").val();
        pass=$("#password").val();
        $.post("http://localhost:3000/login",{user: user,password: pass}, function(data){
          if(data === 'yes') {
              alert("login success");
            }
        });
      });
    });
  </script>
</head>
<body>
<h1>Hello people !</h1>

<input id="user" size="40" type="TEXT" />
<input id="password" size="40" type="password" />
<input id="submit" type="button" value="Submit" />

</body>
</html>
```