## **Module 2 Lab 1, Javascript**

Saurabh Kale

IFT 458/554: Middleware Programming & Database Security

Dinesh Sthapit

## **Module 2 Lab 1, Javascript**

**Part A**

* npm init –

A screenshot of a computer

Description automatically generated

* Server Started-

A screenshot of a computer

Description automatically generated

Code:

// Student Name: Saurabh Kale

// Student ID: 1223450319

// Date: 09/02/2023

const httpServer = require('http');

//Create server

const server = httpServer.createServer(function(req, res){ //callback function

res.end(` We received our first request from the client`)

});

// Start listening to requests

server.listen(8000, 'localhost', function(){

console.log('Listening to requests on port 8000')

});

* Output-

A screenshot of a computer

Description automatically generated

* Ipaddress-

A screenshot of a computer screen

Description automatically generated

* Url parameter-

A computer screen shot of a program

Description automatically generated

Script:

// Student Name: Saurabh Kale

// Student ID: 1223450319

// Date: 09/02/2023

const httpServer = require('http');

const url = require('url');

//Create server

const server = httpServer.createServer(function(req, res){ //callback function

const urlparameter = url.parse(req.url, true);

console.log(urlparameter.query);

console.log(urlparameter.path);

res.end(` We received our first request from the client`)

});

// Start listening to requests

server.listen(8000, 'localhost', function(){

console.log('Listening to requests on port 8000')

});

* Output Screen with url-

A screenshot of a computer

Description automatically generated

* Debugging req.url-

A screenshot of a computer

Description automatically generated

* Screenshot of pathname-

A screenshot of a computer program

Description automatically generated

// Student Name: Saurabh Kale

// Student ID: 1223450319

// Date: 09/02/2023

const httpServer = require('http');

const url = require('url');

//Create server

const server = httpServer.createServer(function(req, res){ //callback

const urlparameter = url.parse(req.url, true);

console.log(urlparameter.query);

console.log(urlparameter.pathname);

res.end(` We received our first request from the client`)

});

// Start listening to requests

server.listen(8000, 'localhost', function(){

console.log('Listening to requests on port 8000')

});

**Part B**

* Urlparameter for query-

**A screenshot of a computer

Description automatically generated**

Script:

// Student Name: Saurabh Kale

// Student ID: 1223450319

// Date: 09/02/2023

const httpServer = require('http');

const url = require('url');

//Create server

const server = httpServer.createServer(function(req, res){ //callback

const urlparameter = url.parse(req.url, true);

console.log(urlparameter.query);

console.log(urlparameter.pathname);

if(urlparameter.query.id){ //if there is query parameter named id

// course page

if(urlparameter.pathname === '/' || urlparameter.pathname.toLowerCase() === '/course'){

res.writeHead(200, {

'Content-type':'text/html'

});

}

else{

res.writeHead(404, {

'Content-type':'text/html'

});

res.end('resource not found');

}

}

res.end(` We received our first request from the client at resource ${urlparameter.pathname.toLowerCase()} with query parameter ${urlparameter.query.id}`)

});

// Start listening to requests

server.listen(8000, 'localhost', function(){

console.log('Listening to requests on port 8000')

});

* Resourse not Found-

**A screenshot of a computer

Description automatically generated**

* 200 OK response-

**A screenshot of a computer

Description automatically generated**

**Step 6: Reading Data From File and Sending the Data to the Client**

* Reading data-

**A screenshot of a computer

Description automatically generated**

Code-  
// Student Name: Saurabh Kale

// Student ID: 1223450319

// Date: 09/02/2023

const httpServer = require('http');

const url = require('url');

const fs = require('fs');

//Read data from file

const tempCourse = fs.readFileSync(

    `${\_\_dirname}/data.txt`,

    'utf-8'

)

//Create Server

const server = httpServer.createServer(function(req, res) { //callback function

    const urlparameter = url.parse(req.url, true);

    console.log(urlparameter.query);

    console.log(urlparameter.pathname);

    if(urlparameter.query.id){ //if there is query parameter named id

        //Course page

        if(urlparameter.pathname === '/' || urlparameter.pathname.toLowerCase() === '/courses'){

            res.writeHead(200, { //Everything ran successfully

                'Content-type':'text/html'

            });

            res.end(`We received our first request from the client at resourse ${urlparameter.pathname.toLowerCase()} with query parameter ${urlparameter.query.id}

            ${tempCourse}`);

        }

        else{

            res.writeHead(404, {    //Server did not find what you were looking

                'Content-type':'text/html'

            });

            res.end(`resource not found`);

        }

    }

});

//Start Listening to requests

server.listen(8000, 'localhost', function() {

    console.log('Listening to requests on port 8000');

});

**Step 7: Reading JSON data from the file**

* JSON Object-

A screenshot of a computer

Description automatically generated

* JSON Object file-

A computer screen shot of a program

Description automatically generated

Code-  
// Student Name: Saurabh Kale

// Student ID: 1223450319

// Date: 09/02/2023

const httpServer = require('http');

const url = require('url');

const fs = require('fs');

//Read data from file

const tempCourse = fs.readFileSync(

`${\_\_dirname}/data.json`,

'utf-8'

)

const dataObj = JSON.parse(tempCourse);

//Create Server

const server = httpServer.createServer(function(req, res) { //callback function

const urlparameter = url.parse(req.url, true);

console.log(urlparameter.query);

console.log(urlparameter.pathname);

if(urlparameter.query.id){ //if there is query parameter named id

//Course page

if(urlparameter.pathname === '/' || urlparameter.pathname.toLowerCase() === '/courses'){

res.writeHead(200, { //Everything ran successfully

'Content-type':'text/html'

});

const course = dataObj[urlparameter.query.id]

res.end(`We received our first request from the client at resourse ${urlparameter.pathname.toLowerCase()} with query parameter ${urlparameter.query.id}

${course}`);

}

else{

res.writeHead(404, { //Server did not find what you were looking

'Content-type':'text/html'

});

res.end(`resource not found`);

}

}

});

//Start Listening to requests

server.listen(8000, 'localhost', function() {

console.log('Listening to requests on port 8000');

});

* Json object stringified-

A screenshot of a computer

Description automatically generated

Code-

// Student Name: Saurabh Kale

// Student ID: 1223450319

// Date: 09/02/2023

const httpServer = require('http');

const url = require('url');

const fs = require('fs');

//Read data from file

const tempCourse = fs.readFileSync(

`${\_\_dirname}/data.json`,

'utf-8'

)

const dataObj = JSON.parse(tempCourse);

//Create Server

const server = httpServer.createServer(function(req, res) { //callback function

const urlparameter = url.parse(req.url, true);

console.log(urlparameter.query);

console.log(urlparameter.pathname);

if(urlparameter.query.id){ //if there is query parameter named id

//Course page

if(urlparameter.pathname === '/' || urlparameter.pathname.toLowerCase() === '/courses'){

res.writeHead(200, { //Everything ran successfully

'Content-type':'text/html'

});

const course = dataObj[Number(urlparameter.query.id)]

res.end(`We received our first request from the client at resourse ${urlparameter.pathname.toLowerCase()} with query parameter ${urlparameter.query.id}

${JSON.stringify(course)}`);

}

else{

res.writeHead(404, { //Server did not find what you were looking

'Content-type':'text/html'

});

res.end(`resource not found`);

}

}

});

//Start Listening to requests

server.listen(8000, 'localhost', function() {

console.log('Listening to requests on port 8000');

});

* Third Parameter added-

A screenshot of a computer

Description automatically generated

* Third JSON object-

A computer screen shot of a program

Description automatically generated

**Step 8: Adding a User Interface in HTML: Download the Sample template file**

**A computer screen shot of a program

Description automatically generated**

Script-

// Student Name: Saurabh Kale

// Student ID: 1223450319

// Date: 09/02/2023

const httpServer = require('http');

const url = require('url');

const fs = require('fs');

//Read data from file

const tempCourse = fs.readFileSync(

    `${\_\_dirname}/data.json`,

    'utf-8'

)

const replaceTemplate = (htmlStr, course) =>{   //fat arrow function or lambda

    let output= htmlStr.replace(/${NAME%}/g, course.courseName);

    output= output.replace(/{%IMAGE%}/g, course.image);

    output= output.replace(/{%FROM%}/g, course.from);

    output= output.replace(/{%INSTRUCTOR%}/g, course.instructor);

    output= output.replace(/{%CREDITS%}/g, course.credits);

    output= output.replace(/{%DESCRIPTION%}/g, course.description);

    output= output.replace(/{%ID%}/g, course.id);

    return output;

}

const dataObj = JSON.parse(tempCourse); //converts String to Javascript object JSON

//Create Server

const server = httpServer.createServer( (req, res)=> { //callback function

    const urlparameter = url.parse(req.url, true);

    console.log(urlparameter.query);

    console.log(urlparameter.pathname);

    if(urlparameter.query.id){ //if there is query parameter named id read as String

        //Course page

        if(urlparameter.pathname === '/' || urlparameter.pathname.toLowerCase() === '/courses'){

            res.writeHead(200, { //Everything ran successfully

                'Content-type':'text/html'

            });

            const course = dataObj[Number(urlparameter.query.id)]   //Convert string to numeric value

            res.end(`We received our first request from the client at resourse ${urlparameter.pathname.toLowerCase()} with query parameter ${urlparameter.query.id}

            ${JSON.stringify(course)}`);    //Convert object back to string

        }

        else{

            res.writeHead(404, {    //Server did not find what you were looking

                'Content-type':'text/html'

            });

            res.end(`resource not found`);

        }

    }

});

//Start Listening to requests

server.listen(8000, 'localhost',()=> {

    console.log('Listening to requests on port 8000');

});

**Step 9: Module Export (reusable functions)**

A computer screen shot of a course list

Description automatically generated

* replaceTemplate.js file-

A computer screen shot of a computer program

Description automatically generated

Script-

// Student Name: Saurabh Kale

// Student ID: 1223450319

// Date: 09/02/2023

const httpServer = require('http');

const url = require('url');

const fs = require('fs');

const replaceTemplate = require('./replaceTemplate');

//Read data from file

const tempCourse = fs.readFileSync(

    `${\_\_dirname}/data.json`,

    'utf-8'

)

//Read data from file

const templateHTMLCourse = fs.readFileSync(

    `${\_\_dirname}/templateCourse.html`,

    'utf-8'

)

// const replaceTemplate = (htmlStr, course) =>{   //fat arrow function or lambda

//     let output= htmlStr.replace(/${NAME%}/g, course.courseName);

//     output= output.replace(/{%IMAGE%}/g, course.image);

//     output= output.replace(/{%FROM%}/g, course.from);

//     output= output.replace(/{%INSTRUCTOR%}/g, course.instructor);

//     output= output.replace(/{%CREDITS%}/g, course.credits);

//     output= output.replace(/{%DESCRIPTION%}/g, course.description);

//     output= output.replace(/{%ID%}/g, course.id);

//     return output;

// }

const dataObj = JSON.parse(tempCourse); //converts String to Javascript object JSON

//Create Server

const server = httpServer.createServer( (req, res)=> { //callback function

    const urlparameter = url.parse(req.url, true);

    console.log(urlparameter.query);

    console.log(urlparameter.pathname);

    if(urlparameter.query.id){ //if there is query parameter named id read as String

        //Course page

        if(urlparameter.pathname === '/' || urlparameter.pathname.toLowerCase() === '/courses'){

            res.writeHead(200, { //Everything ran successfully

                'Content-type':'text/html'

            });

            const course = dataObj[Number(urlparameter.query.id)]   //Convert string to numeric value

            // res.end(`We received our first request from the client at resourse ${urlparameter.pathname.toLowerCase()} with query parameter ${urlparameter.query.id}

            // ${JSON.stringify(course)}`);    //Convert object back to string

            const strCourseName = JSON.stringify(course);

            const courseHTML = replaceTemplate(templateHTMLCourse,course);

            res.end(courseHTML);

        }

        else{

            res.writeHead(404, {    //Server did not find what you were looking

                'Content-type':'text/html'

            });

            res.end(`resource not found`);

        }

    }

});

//Start Listening to requests

server.listen(8000, 'localhost',()=> {

    console.log('Listening to requests on port 8000');

});

* ReplaceTemplate.js File-

module.exports = (htmlStr, course) =>{ //fat arrow function or lambda

let output= htmlStr.replace(/{%NAME%}/g, course.courseName);

output= output.replace(/{%IMAGE%}/g, course.image);

output= output.replace(/{%FROM%}/g, course.from);

output= output.replace(/{%INSTRUCTOR%}/g, course.instructor);

output= output.replace(/{%CREDITS%}/g, course.credits);

output= output.replace(/{%DESCRIPTION%}/g, course.description);

output= output.replace(/{%ID%}/g, course.id);

return output;

}

**Step 10: Object Destruction**

A computer screen shot of a program

Description automatically generated

Converted 3 lines of code to 1 line.

Script:

// Student Name: Saurabh Kale

// Student ID: 1223450319

// Date: 09/02/2023

const httpServer = require('http');

const url = require('url');

const fs = require('fs');

const replaceTemplate = require('./modules/replaceTemplate');

// Read data from file

// Template

const tempCourse = fs.readFileSync(

`${\_\_dirname}/data/data.json`,

'utf-8'

);

// Template

const templateHTMLCourse = fs.readFileSync(

`${\_\_dirname}/template/templateCourse.html`,

'utf-8'

);

const dataObj = JSON.parse(tempCourse); //convert string to javascript object JSON

// create server

const server = httpServer.createServer((req, res) => { //callback

//const urlparameter = url.parse(req.url, true):

//console.log(urlparameter.query);

//console.log(urlparameter.pathname);

const{query, pathname} = url.parse(req.url, true);

if(query.id){ //if there is query parameter named id read as string

// Course page

if(pathname === '/' || pathname.toLowerCase() === '/courses'){

res.writeHead(200, { //Everything ran successfully

'Content-type':'text/html'

});

const course = dataObj[Number(query.id)]; //convert string to numeric value

//res.end(` We received our first request from the client at resource ${urlparameter.pathname.toLowerCase()} with query parameter ${urlparameter.query.id}

//${JSON.stringify(course)}`); //Convert object back to string

const strCourseName = JSON.stringify(course);

const courseHTML = replaceTemplate(templateHTMLCourse, course);

res.end(courseHTML);

}

else{

res.writeHead(404, { //server didn't find what you were looking for

'Content-type':'text/html'

});

res.end('resource not found');

}

}

});

// Start listening to requests

server.listen(8000, 'localhost', () => {

console.log('Listening to requests on port 8000')

});

**Step 11: Refactoring and re-arranging the folder structure.**

**A screenshot of a computer

Description automatically generated**