

COMP 9783 Front-end Development

Lab-2-1 - JS Modules Lab. (4% of the course mark)

Name:

Student Number:

In this lab, you will explore the concept of JavaScript modules, a crucial aspect of modern JavaScript development. Modules help in organizing code, promoting reusability, and reducing complexity. You will learn different module systems, such as ES6 modules and CommonJS, and how to use them effectively in your projects.

Lab objectives:

1. Understand the importance of modular code in JavaScript development.
2. Learn the syntax and usage of ES6 modules and CommonJS.

Create a Node.js app:

1. On **VSCode**, create a **folder** named **Node-Modules-App**.
2. Open the **terminal** and change the directory to **Node-Modules-App**.
3. Initialize the app by typing **npm init -y** and **press enter**.

Note: *npm init -y is a quick shortcut to initialize the package.json file. It will initialize the app with default values.*

4. Type **npm install uuid** and **press enter**.
5. Create a folder named: **lib**. This will be used later.

Using CommonJS module:

1. Create a file named: **CommonJS.js** and enter the following code:

```
const uuid = require("uuid");
```

```
console.log(`UUID Version 1 (Time-based): ${uuid.v1()}`);
```

```
console.log(`UUID Version 4 (Random): ${uuid.v4()}`);
```

2. Save the changes.
3. Open the **terminal** and type **node CommonJS.js**.
4. Take a screenshot and name it **CommonJS01.png**.
5. Navigate to the **lib folder** and create a **file** named: **UUIDUtilsCommonJS.js**.
6. Enter the following code:

```
const uuid = require("uuid");
```

7. Create a function named: **generateUUID(version)**, where version is a number.
8. Implement the following logic:
 - a. If **version** is equal to **1**, return the value of **uuid.v1()**.
 - b. If **version** is equal to **4**, return the value of **uuid.v4()**.
 - c. **Otherwise** return an **empty string**.
9. At the end of file, enter the following code:

```
module.exports = { generateUUID };
```

10. Save the changes.

11. Comment out the contents of **CommonJS.js** and enter the following code:

```
const UUIDUtilsCommonJS = require("./lib/UUIDUtilsCommonJS");
```

12. Implement the following coding tasks:

a. Display the text: "**CommonJS module demo...**".

Note: The **generateUUID function** from the **UUIDUtilsCommonJS module**, can be invoked like this: `UUIDUtilsCommonJS.generateUUID(version)`, where version is any valid number.

b. Call **generateUUID function** with **version** equal to **1** and display the results.

c. Call **generateUUID function** with **version** equal to **4** and display the results.

d. Call **generateUUID function** with **version** equal to **anything other than 1 or 4** and display the results.

13. Save the changes.

14. Open the **terminal** and type **node CommonJS.js**.

15. Take a screenshot and name it **CommonJS02.png**.

Using ES6 module:

1. Update package.json and enter the following json attribute: **"type": "module"**, right after the **description** attribute.
2. Create a file named: **ES6.js** and enter the following code:

```
import { v1, v4 } from "uuid";
```

```
console.log(`UUID Version 1 (Time-based): ${v1()}`);
```

```
console.log(`UUID Version 4 (Random): ${v4()}`);
```

3. Save the changes.
4. Open the **terminal** and type **node ES6.js**.
5. Take a screenshot and name it **ES601.png**.
6. Navigate to the **lib folder** and create a **file** named: **UUIDUtilsES6.js**.
7. Enter the following code:

```
import { v1, v4 } from "uuid";
```

8. Create a function named: **generateUUID(version)**, where version is a number.
9. Implement the following logic:
 - a. If **version** is equal to **1**, return the value of **v1()**.
 - b. If **version** is equal to **4**, return the value of **v4()**.
 - c. **Otherwise** return an **empty string**.
10. At the end of file, enter the following code:

```
export { generateUUID };
```

11. Save the changes.

12. Comment out the contents of **ES6.js** and enter the following code:

```
import { generateUUID } from "../lib/UUIDUtilsES6.js";
```

13. Implement the following coding tasks:

- a. Display the text: "**ES6 module demo...**".

Note: The **generateUUID function** from the **UUIDUtilsES6 module**, can be invoked like this:
generateUUID(version), where version is any valid number.

- b. Call **generateUUID function** with **version** equal to **1** and display the results.
- c. Call **generateUUID function** with **version** equal to **4** and display the results.
- d. Call **generateUUID function** with **version** equal to **anything other than 1 or 4** and display the results.

14. Save the changes.

15. Open the **terminal** and type **node ES6.js**.

16. Take a screenshot and name it **ES602.png**.

Submission:

1. Use the html template: **index.html** and **write HTML codes for each screenshot**:
 - a. Write a title and short description.
 - b. Display the screenshot.
2. Create a new folder named **html** and copy all the **HTML**, **CSS** and **PNG** files used in the previous step.
3. Create a new folder named **app** and copy the following from **Node-Modules-App**:
 - a. lib folder
 - b. CommonJS.js
 - c. ES6.js
 - d. package-lock.json

e. package.json

Note: Do not include the node_modules folder, this is not needed as I can just run npm install to reinstall the packages.

4. Create a new folder named **submit** and copy both the **html** and **app** folders.
5. Create a **zip file** of the **submit** folder.
6. Submit the **zip file** to **GBC - D2L**.