

Hands-on Lab - Promise Callback (10 mins)



Objective for Exercise:

After completing this lab, you will be able to:

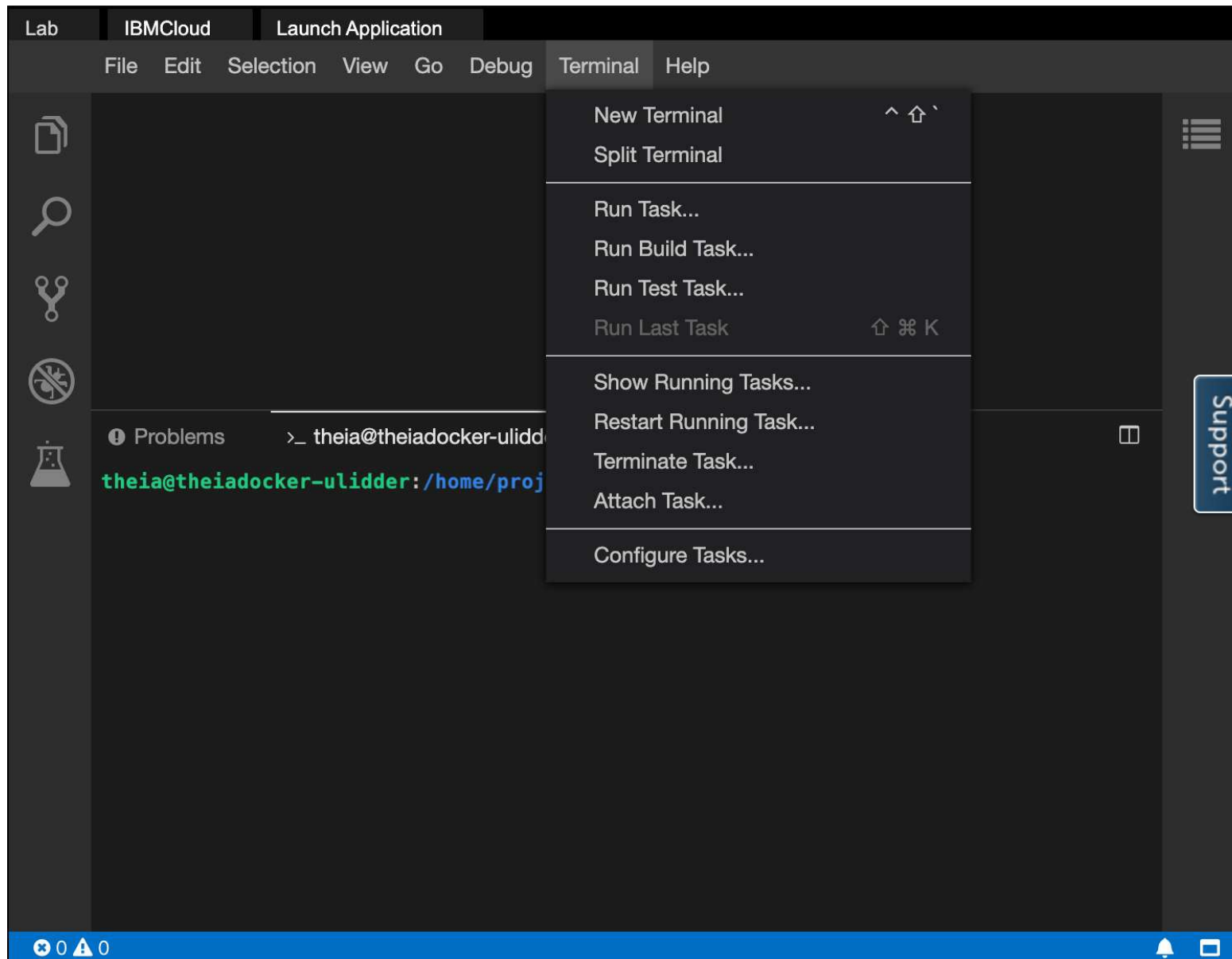
- Describe Promise callbacks
- Create a Node.js application with promises

Prerequisites

- Basic knowledge of JavaScript

Task 1: Create and execute a script with promise

1. Open a terminal window by using the menu in the editor: Terminal > New Terminal.



2. Change to your project folder.

1. 1

1. `cd /home/project`

Copied!

3. Create a new file by running the following command.

1. 1

1. `touch promiscrypt.js`

Copied!

5. List the files in the current directory to see if the file has been created.

```
1. 1
1. ls
```

Copied!

6. Open the file **promisescript.js** and edit it.

Open **promisescript.js** in IDE

7. Paste the following code in the file and save it. Are you able to look at the file and guess the order in which the console logs will be executed?

```
1. 1
2. 2
3. 3
4. 4
5. 5
6. 6
7. 7
8. 8
9. 9
10. 10
11. 11
12. 12
13. 13
14. 14
15. 15
16. 16

1. //Creating a promise method. The promise will get resolved when timer times out after 6 seconds.
2. let myPromise = new Promise((resolve,reject) => {
3.   setTimeout(() => {
4.     resolve("Promise resolved")
5.   },6000)})
6.
7. //Console log before calling the promise
8. console.log("Before calling promise");
9.
10. //Call the promise and wait for it to be resolved and then print a message.
11. myPromise.then((successMessage) => {
12.   console.log("From Callback " + successMessage)
13. })
14.
15. //Console log after calling the promise
16. console.log("After calling promise");
```

Copied!

8. Go to the terminal and run the file to verify your guess.

```
1. 1
1. node promisescript.js
```

Copied!

9. The order in which the console logs are executed tell you that the statements after the promise call are executed one after the other and the promise is simultaneously getting resolved or rejected.

Practice Exercise: Create and execute a script with two promises

1. Create a script which has two methods that return promises - One of the promises should get resolved after 6 seconds timeout and the other one after 3 seconds timeout. Call the promise in such a way that the second promise is invoked after the first promise is resolved.

▼ Click here to view the code

```
1. 1
2. 2
3. 3
4. 4
5. 5
6. 6
7. 7
8. 8
9. 9
10. 10
11. 11
12. 12
13. 13
14. 14
15. 15
16. 16

1. let myPromise1 = new Promise((resolve,reject) => {
2.   setTimeout(() => {
3.     resolve("Promise 1 resolved")
4.   },6000)})
5.
```

```
6. let myPromise2 = new Promise((resolve,reject) => {
7.   setTimeout(() => {
8.     resolve("Promise 2 resolved")
9.   },3000)})
10.
11. myPromise1.then((successMessage) => {
12.   console.log("From Callback " + successMessage)
13.   myPromise2.then((successMessage) => {
14.     console.log("From Callback " + successMessage)
15.   })
16. })
```

Copied!

2. Change the code to call the promises sequentially and see how the output changes.

▼ Click here to view the code

```
1. 1
2. 2
3. 3
4. 4
5. 5
6. 6
7. 7
8. 8
9. 9
10. 10
11. 11
12. 12
13. 13
14. 14
15. 15
16. 16
17. 17

1. let myPromise1 = new Promise((resolve,reject) => {
2.   setTimeout(() => {
3.     resolve("Promise 1 resolved")
4.   },6000)})
5.
6. let myPromise2 = new Promise((resolve,reject) => {
7.   setTimeout(() => {
8.     resolve("Promise 2 resolved")
9.   },3000)})
10.
11. myPromise1.then((successMessage) => {
12.   console.log("From Callback " + successMessage)
13. })
14.
15. myPromise2.then((successMessage) => {
16.   console.log("From Callback " + successMessage)
17. })
```

Copied!

Congratulations! You have completed the Promise-Callback lab.

Summary

In this lab, you learned how to use promises with callbacks in two ways:

- How to call a second promise after the first is resolved.
- How to call promises sequentially.

Next, you will learn how to improve server-side capabilities using these methods.

Author(s)

Lavanya

Changelog

| Date | Version | Changed by | Change Description |
|------------|---------|-----------------|-------------------------|
| 2022-10-07 | 1.0 | Lavanya | Initial version created |
| 2022-10-18 | 1.1 | Sapthashree K S | Summary Updated |