Hands-on Lab - Promise Callback (10 mins)

Objective for Exercise:

After completing this lab, you will be able to:

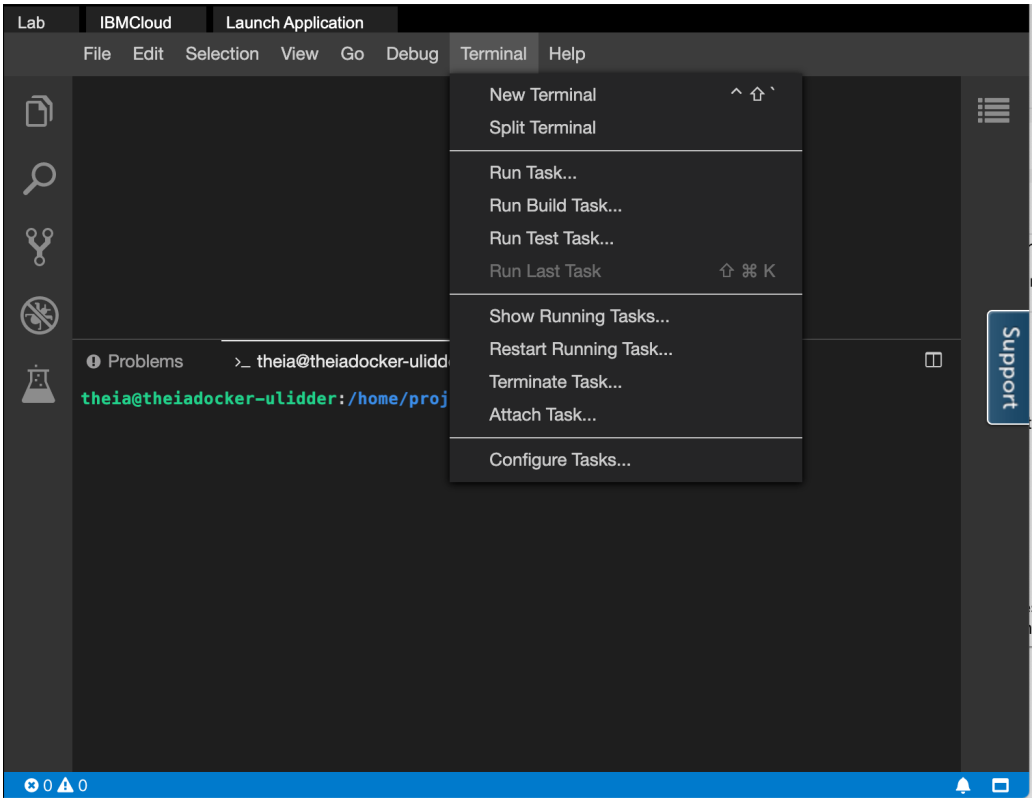
- Describe Promise callbacks
- Create a Node.js application with promises

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.

```
cd /home/project
```

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

```
touch promisescript.js
```

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

```
ls
```

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?

```
//Creating a promise method. The promise will get resolved when timer times out after 6 seconds.
let myPromise = new Promise((resolve,reject) => {
    setTimeout(() => {
      resolve("Promise resolved")
    },6000)})
//Console log before calling the promise
console.log("Before calling promise");
//Call the promise and wait for it to be resolved and then print a message.
myPromise.then((successMessage) => {
    console.log("From Callback " + successMessage)
  })
//Console log after calling the promise
  console.log("After calling promise");
```

- The above code creates a promise (`myPromise`) that resolves after 6 seconds with the message `Promise resolved`.
- The script first logs `Before calling promise` to the console.
- Then, it sets up the `then` method to handle the promise resolution, but the callback does not execute yet.
- Immediately after this, the script logs `After calling promise` to the console.
- After 6 seconds, the promise resolves, and the callback logs `From Callback Promise resolved` to the console.

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

```
node promisescript.js
```

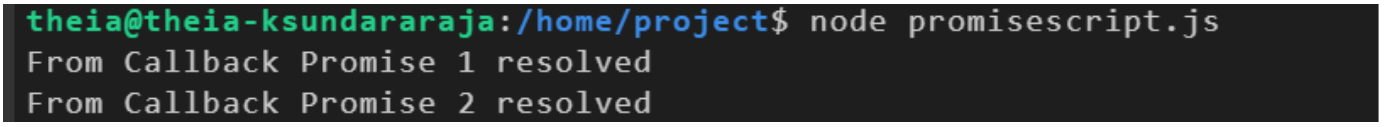
9. The order in which the console logs are executed tell you that the statements after the promise call are executed one after the other and the promise is simultaneouly getting resolved or rejected.

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 reolved 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

```
let myPromise1 = new Promise((resolve,reject) => {
    setTimeout(() => {
      resolve("Promise 1 resolved")
    },6000)})
let myPromise2 = new Promise((resolve,reject) => {
    setTimeout(() => {
      resolve("Promise 2 resolved")
    },3000)})
  myPromise1.then((successMessage) => {
    console.log("From Callback " + successMessage)
    myPromise2.then((successMessage) => {
      console.log("From Callback " + successMessage)
    })
  })
```



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

▾ Click here to view the code

```
let myPromise1 = new Promise((resolve,reject) => {
    setTimeout(() => {
      resolve("Promise 1 resolved")
    },6000)})
let myPromise2 = new Promise((resolve,reject) => {
    setTimeout(() => {
      resolve("Promise 2 resolved")
    },3000)})
  myPromise1.then((successMessage) => {
    console.log("From Callback " + successMessage)
  })
  myPromise2.then((successMessage) => {
  console.log("From Callback " + successMessage)
})
```

```
theia@theia-ksundararaja:/home/project$ node promisescript.js
From Callback Promise 2 resolved
From Callback Promise 1 resolved
```

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