



Updating the function with Additional Dependencies

1. In this lab we are again going to update our lambda function code but with some more additional dependencies.
2. As usual create a folder in your VS Code and then create the commands text file with that you need to create the **index.js** and **event.json** files too.
3. Use the code from previous labs for both of the files or you can copy the files from the previous lab and paste them in the current lab.

The screenshot shows a VS Code interface. On the left, the file tree (Folders: SERVE...) shows several lab files. On the right, the editor pane displays the contents of **index.js**. The **commands.txt** file is highlighted with a red border at the bottom left of the editor area.

```
exports.handler = async (event) => {
  console.log("EVENT: \n" + JSON.stringify(event, null, 2));
  const total = event.num1 * event.num2;
  const response = {
    statusCode: 200,
    body: "The total of " + event.num1 + " and " + event.num2 + " is " + total
  };
  return response;
};
```

4. After that we will Install the required Dependencies into **node_modules** for that we need the command given below. The below command will install the node modules and the JSON packages as you can see in the snapshot.

npm install aws-xray-sdk

The screenshot shows a VS Code interface. On the left, the file tree shows **package.json** being edited. On the right, the terminal pane shows the command **npm install aws-xray-sdk** being run, which installs 43 packages in 3 seconds. The **package.json** file is highlighted with a red border at the bottom left of the editor area.

```
{
  "dependencies": {
    "aws-xray-sdk": "^3.10.1"
  }
}
```

TERMINAL

```
PS C:\Serverless> cd './Lab 8 Updating some more Additional Dependencies'
PS C:\Serverless\Lab 8 Updating some more Additional Dependencies> npm install aws-xray-sdk
● >>

added 43 packages in 3s
○ PS C:\Serverless\Lab 8 Updating some more Additional Dependencies>
```

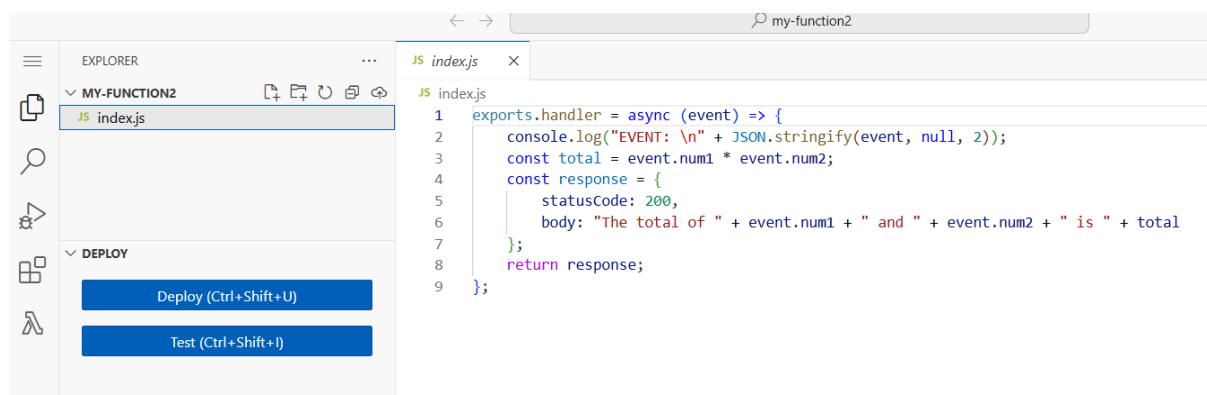
5. Once your modules and packages are created now you need to zip them together. Leave the command.txt and event.json files we don't want to zip them.
6. We need to do this manually by going directly into the folder on our laptop. Select the files right click on them and then compress them to a zip file.

Name	Date modified	Type	Size
node_modules	12-11-2024 17:26	File folder	
commands	12-11-2024 17:23	Text Document	2 KB
event	12-11-2024 17:23	JSON Source File	1 KB
index	12-11-2024 17:23	JavaScript Source File	1 KB
package	12-11-2024 17:26	JSON Source File	1 KB
package-lock	12-11-2024 17:26	JSON Source File	16 KB

7. You need to rename your zipped file to function.zip and save it.

Name	Date modified	Type	Size
node_modules	12-11-2024 17:26	File folder	
commands	12-11-2024 17:23	Text Document	2 KB
event	12-11-2024 17:23	JSON Source File	1 KB
index	12-11-2024 17:23	JavaScript Source File	1 KB
package	12-11-2024 17:26	JSON Source File	1 KB
package-lock	12-11-2024 17:26	JSON Source File	16 KB
function	12-11-2024 17:31	Compressed (zipped) F...	1,410 KB

8. If we go inside our function, we can see that we only have the index.js file.



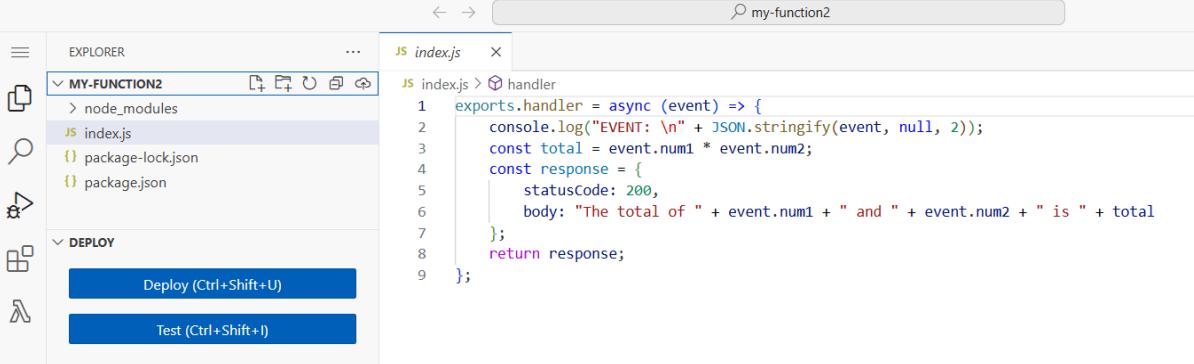
9. Now we are going to run the update command. Here you can see that our function has been updated successfully.

```

○ PS C:\Serverless\Lab 8 Updating some more Additional Dependencies> aws lambda update-function-code ` 
>> --function-name my-function2 ` 
>> --zip-file fileb://function.zip
Architectures:
- x86_64
CodeSha256: y8iFMkB7W04Lk8lAz1keepj4p2JceUnb8KLE1Rti+iU=
CodeSize: 1443783
Description: ''
EphemeralStorage:
  Size: 512
FunctionArn: arn:aws:lambda:us-east-1:878893308172:function:my-function2
FunctionName: my-function2
Handler: index.handler
LastModified: 2024-11-12T12:04:53.000+0000
LastUpdateStatus: InProgress
LastUpdateStatusReason: The function is being created.
LastUpdateStatusReasonCode: Creating

```

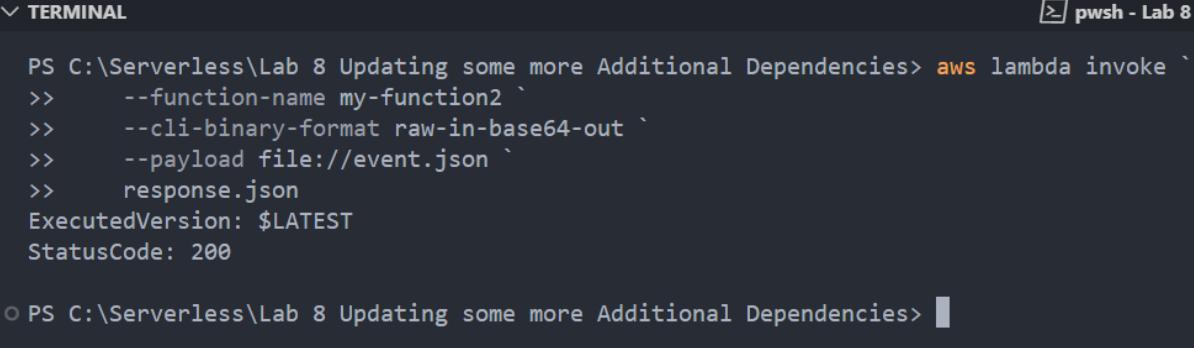
10. Once your function is updated go to the console and refresh the page you will see all the files there.



The screenshot shows the VS Code interface with the 'EXPLORER' view open. The 'MY-FUNCTION2' folder is selected. Inside, there are files: 'index.js', 'package-lock.json', and 'package.json'. The 'index.js' file is currently selected and shown in the main code editor area. The code in 'index.js' is:

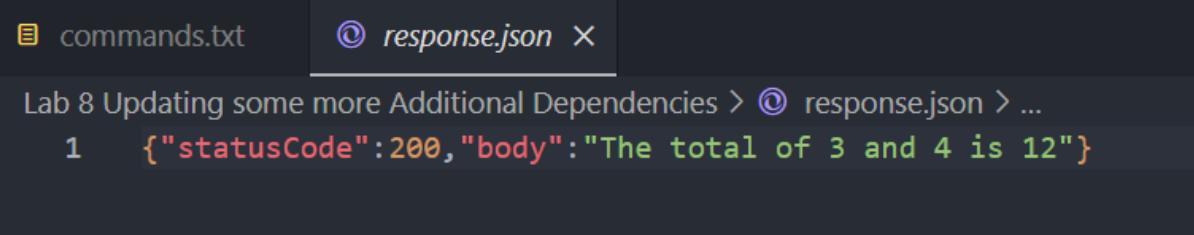
```
JS index.js > handler
1 exports.handler = async (event) => {
2   console.log("EVENT: \n" + JSON.stringify(event, null, 2));
3   const total = event.num1 * event.num2;
4   const response = {
5     statusCode: 200,
6     body: "The total of " + event.num1 + " and " + event.num2 + " is " + total
7   };
8   return response;
9 };
```

11. Now the last step for us is to invoke our function. So run the invoke command and you will get the status code 200.



```
PS C:\Serverless\Lab 8 Updating some more Additional Dependencies> aws lambda invoke \
--function-name my-function2 \
--cli-binary-format raw-in-base64-out \
--payload file://event.json \
response.json
ExecutedVersion: $LATEST
StatusCode: 200
```

12. In the end go to the response file and you will see the response as expected.



The screenshot shows the VS Code interface with the 'response.json' file open in the editor. The file contains the following JSON object:

```
1 {"statusCode":200,"body":"The total of 3 and 4 is 12"}
```

13. You can always go and check the logs in CloudWatch.