



# AWS Lambda Context Object in Node.js Function

- When Lambda runs our function, it passes a context object to the handler. This context object provides methods and properties that provide information about the invocation, function, and execution environment.
- So now we are going to develop an example lambda function to see context objects and parameters in the lambda function.
- To do that first create a new folder in VS Code and create the commands text file.
- Then we need to create an index.js file and convert it into a zip file.

The screenshot shows the VS Code interface with the following details:

- Folders:** SERVE... (with icons for file operations), 8 Updating some more Additional Dependencies, response.json, commands.txt, Lab 9 AWS Lambda Context Object in Node.js Function.
- File Explorer:** Shows a tree view of files and folders related to the lab, including Lab 1 through Lab 8, and Lab 9 AWS Lambda Context Object in Node.js Function.
- Code Editor:** Displays the content of index.js:

```
Lab 9 AWS Lambda Context Object in Node.js Function > O: index.js > handler > handler
1 exports.handler = async (event, context) => {
2   console.log('Remaining time: ', context.getRemainingTimeInMillis())
3   console.log('Function name: ', context.functionName)
4
5   const body =
6     `Function name: ${context.functionName}
7     LogStream name: ${context.logStreamName}
8
9   return body;
10 }
```
- Terminal:** Shows the command history for creating the lambda function:

```
PS C:\Serverless\Lab 8 Updating some more Additional Dependencies> cd ..
● PS C:\Serverless> cd '.\Lab 9 AWS Lambda Context Object in Node.js Function\'
```

- After that we run this command to get the ARN of our lambda execution role which we created in the earlier labs. Copy the ARN and we will use this in the create lambda function command.

```
PS C:\Serverless\Lab 9 AWS Lambda Context Object in Node.js Function> aws iam get-role --role-name lambda-exec
>>
Role:
Arn: arn:aws:iam::878893308172:role/lambda-exec
AssumeRolePolicyDocument:
Statement:
- Action: sts:AssumeRole
Effect: Allow
Principal:
```

- Below we ran the create function command and you can see that our function has been created.

```

PS C:\Serverless\Lab 9 AWS Lambda Context Object in Node.js Function> aws lambda create-function \
--function-name my-function3 \
--runtime nodejs18.x \
--zip-file fileb://function.zip \
--handler index.handler \
--role arn:aws:iam::878893308172:role/lambda-exec
Architectures:
- x86_64
CodeSha256: 1xM5OYMr8cI3rZl57f4PySYm3WrUbUefmGmQs2FsGvQ=
CodeSize: 286
Description: ''
EphemeralStorage:
  Size: 512
FunctionArn: arn:aws:lambda:us-east-1:878893308172:function:my-function3
FunctionName: my-function3
Handler: index.handler
LastModified: 2024-11-12T12:21:39.741+0000
LoggingConfig:
  LogFormat: Text
  LogGroup: /aws/lambda/my-function3

```

- From the lambda console too, you can see that your function has been created.

The screenshot shows the AWS Lambda Functions list page. At the top, there is a breadcrumb navigation: Lambda > Functions. Below the header, there is a search bar with the placeholder "Filter by tags and attributes or search by keyword". The main area displays a table with three rows, each representing a function. The columns are: a checkbox column, Function name, Description, Package type, Runtime, and Last modified. The data is as follows:

<input type="checkbox"/>	Function name	Description	Package type	Runtime	Last modified
<input type="checkbox"/>	<a href="#">myfirst-function</a>	-	Zip	Node.js 20.x	1 day ago
<input type="checkbox"/>	<a href="#">my-function3</a>	-	Zip	Node.js 18.x	35 seconds ago
<input type="checkbox"/>	<a href="#">my-function2</a>	-	Zip	Node.js 18.x	17 minutes ago

- Now we just need to invoke our function. For that create the invoke.json file put the necessary code in it and save the file.

FOLDERS: SERVE... Folders Open Save Sync New Edit Search Help

Lab 9 AWS Lambda Context Object in Node.js Function > response.json con

- ▶ Lab 1 Using AWS Lambda with ...
- ▶ Lab 2 Invoke Lambda using CLI
- ▶ Lab 3 Invoke Lambda functions ...
- ▶ Lab 4 Create Execution Role
- ▶ Lab 5 Creating Lambda Funcito...
- ▶ Lab 6 Creating Cloud Watch Lo...
- ▶ Lab 7 Update Function Code usi...
- ▶ Lab 8 Updating some more Add...
- ▼ Lab 9 AWS Lambda Context Obj...
- commands.txt
- event.json
- function.zip
- index.js

```

1  {
2      "key": "value"
3  }

```

- Run the invoke function command. Here you can see that we got the response as expected.

```

PS C:\Serverless\Lab 9 AWS Lambda Context Object in Node.js Function> aws lambda invoke \
--function-name my-function3 \
--cli-binary-format raw-in-base64-out \
--payload file://event.json \
response.json
ExecutedVersion: $LATEST
StatusCode: 200
●
○ PS C:\Serverless\Lab 9 AWS Lambda Context Object in Node.js Function>

```

- If you open the response file you will see that you have the function name and the log stream name which you can match by going to the CloudWatch logs.

Lab 9 AWS Lambda Context Object in Node.js Function > response.json

```

1  "Function name: my-function3\n          LogStream name: 2024/11/12[$LATEST]\n          3ca2bdc45de942a1a9b4cb79047d30c0\n"

```

Log streams Tags Anomaly detection Metric filters Subscription filters Contributor Insights Data protection

Log streams (1)		Create log stream	Search all log streams
<input type="text" value="Filter log streams or try prefix search"/> <input type="checkbox"/> Exact match <input type="checkbox"/> Show expired		<span>&lt;</span> <span>1</span> <span>&gt;</span>	
<input type="checkbox"/>	Log stream	Last event time	
<input type="checkbox"/>	<a href="#">2024/11/12[\$LATEST]3ca2bdc45de942a1a9b4cb79047d30c0</a>	2024-11-12 17:58:27 (UTC+05:30)	