# Context Object

When Lambda runs your function, it passes a context object to the [handler](https://docs.aws.amazon.com/lambda/latest/dg/nodejs-handler.html). This object provides methods and properties that provide information about the invocation, function, and execution environment.

## Context methods

* getRemainingTimeInMillis() – Returns the number of milliseconds left before the execution times out.

## Context properties

* functionName – The name of the Lambda function.
* functionVersion – The [version](https://docs.aws.amazon.com/lambda/latest/dg/configuration-versions.html) of the function.
* invokedFunctionArn – The Amazon Resource Name (ARN) that's used to invoke the function. Indicates if the invoker specified a version number or alias.
* memoryLimitInMB – The amount of memory that's allocated for the function.
* awsRequestId – The identifier of the invocation request.
* logGroupName – The log group for the function.
* logStreamName – The log stream for the function instance.
* identity – (mobile apps) Information about the Amazon Cognito identity that authorized the request.
  + cognitoIdentityId – The authenticated Amazon Cognito identity.
  + cognitoIdentityPoolId – The Amazon Cognito identity pool that authorized the invocation.
* clientContext – (mobile apps) Client context that's provided to Lambda by the client application.
  + client.installation\_id
  + client.app\_title
  + client.app\_version\_name
  + client.app\_version\_code
  + client.app\_package\_name
  + env.platform\_version
  + env.platform
  + env.make
  + env.model
  + env.locale
  + Custom – Custom values that are set by the mobile application.
* callbackWaitsForEmptyEventLoop – Set to false to send the response right away when the [callback](https://docs.aws.amazon.com/lambda/latest/dg/nodejs-handler.html#nodejs-handler-sync) executes, instead of waiting for the Node.js event loop to be empty. If this is false, any outstanding events continue to run during the next invocation.

var greets = [

"Hello NFJS!",

"Hello Awesome Serverless!",

"Today is a great day!",

"Go Lambda function!",

"I am expecting miracles",

"Go get it!",

"Twinkle Twinkle Little star!",

"How I wonder what you are",

"Love the serverless!",

"Cool apps using serverless!"

];

exports.handler = (event, context, callback) => {

let greet = greets[Math.floor(Math.random()\*10)];

// Context object

console.log('Remaining time: ', context.getRemainingTimeInMillis())

console.log('Function name: ', context.functionName)

console.log('Memory Limit In MB : ', context.memoryLimitInMB)

callback(null, greet);

};

<https://docs.aws.amazon.com/lambda/latest/dg/nodejs-context.html>