

LambdaContext

August 18, 2026

LambdaContext

Kind: Interface

Source: `src/adapters/aws-lambda/types.ts`

Part of: `Adapter`

Handler context parameter.

See [://docs.aws.amazon.com/lambda/latest/dg/nodejs-prog-model-context.html](https://docs.aws.amazon.com/lambda/latest/dg/nodejs-prog-model-context.html) AWS documentation.

`LambdaContext` models the context parameter passed to an AWS Lambda handler. It contains invocation metadata, runtime settings, optional Cognito identity and client context data, and a method for checking the remaining execution time.

Properties

Property	Type
<code>callbackWaitsForEmptyEventLoop</code>	<code>boolean</code>
<code>functionName</code>	<code>string</code>
<code>functionVersion</code>	<code>string</code>
<code>invokedFunctionArn</code>	<code>string</code>
<code>memoryLimitInMB</code>	<code>string</code>
<code>awsRequestId</code>	<code>string</code>
<code>logGroupName</code>	<code>string</code>
<code>logStreamName</code>	<code>string</code>
<code>identity</code>	<code>`CognitoIdentity</code>
<code>clientContext</code>	<code>`ClientContext</code>

Diagram

```
graph LR
  Handler["Lambda handler"] --> Context["LambdaContext"]
  Context --> Invocation["Invocation metadata"]
  Context --> Runtime["Runtime settings"]
  Context --> Identity["CognitoIdentity | undefined"]
  Context --> Client["ClientContext | undefined"]
  Context --> Remaining["getRemainingTimeInMillis()"]
```

Usage

```
import type { LambdaContext } from './types';

export async function handler(
  event: { action?: string },
  context: LambdaContext,
) {
  console.log({
    requestId: context.awsRequestId,
    functionName: context.functionName,
    invokedFunctionArn: context.invokedFunctionArn,
    remainingTime: context.getRemainingTimeInMillis(),
  });

  if (context.getRemainingTimeInMillis() <= 0) {
    throw new Error('Lambda execution time has expired.');
```

AI Coding Instructions

- Accept `LambdaContext` as the handler context parameter when adapter code needs AWS Lambda invocation metadata.
- Call `getRemainingTimeInMillis()` before work that may exceed the Lambda execution limit.
- Treat `identity` and `clientContext` as optional values and check for `undefined` before reading their properties.

- Set `callbackWaitsForEmptyEventLoop` only when the handler should return before the Node.js event loop is empty.
- Keep AWS-specific context handling inside the Lambda adapter boundary.