

EventV1Processor

August 19, 2026

EventV1Processor

Kind: Class**Source:** `src/adapters/aws-lambda/handler.ts`**Part of:** [Adapter](#)

`EventV1Processor` adapts an AWS Lambda event in the V1 format into request data used by the handler layer. It reads the path, method, query string, cookies, and headers, then writes response cookies back to the Lambda result.

Extends: `EventProcessor`

Methods

Method	Signature	Returns
<code>getPath</code>	<code>getPath(event: APIGatewayProxyEvent)</code>	<code>string</code>
<code>getMethod</code>	<code>getMethod(event: APIGatewayProxyEvent)</code>	<code>string</code>
<code>getQueryString</code>	<code>getQueryString(event: APIGatewayProxyEvent)</code>	<code>string</code>
<code>getCookies</code>	<code>getCookies(_event: APIGatewayProxyEvent, _headers: Headers)</code>	<code>void</code>
<code>getHeaders</code>	<code>getHeaders(event: APIGatewayProxyEvent)</code>	<code>Headers</code>
<code>setCookiesToResult</code>	<code>setCookiesToResult(result: APIGatewayProxyResult, cookies: string[])</code>	<code>void</code>

Diagram

```
graph LR
    Event[Event[AWS Lambda V1 Event]] --> Processor[Processor[EventV1Processor]]
    Processor --> Path[Path[getPath]]
    Processor --> Method[Method[getMethod]]
    Processor --> Query[Query[getQueryString]]
```

```
Processor --> Headers[getHeaders]
Processor --> Cookies[getCookies]
Processor --> Result[Lambda Result]
Cookies --> SetCookies[setCookiesToResult]
SetCookies --> Result
```

Usage

```
import { EventV1Processor } from './adapter/aws-lambda/handler'

async function handleLambdaEvent(event: any, result: any) {
  const processor = new EventV1Processor(event, result)

  processor.getCookies()

  const request = new Request(
    `https://lambda.local${processor.getPath()}${processor.getQueryString()}`,
    {
      method: processor.getMethod(),
      headers: processor.getHeaders(),
    },
  )

  const response = await app.fetch(request)

  result.statusCode = response.status
  result.body = await response.text()

  processor.setCookiesToResult()

  return result
}
```

AI Coding Instructions

- Keep event parsing inside `EventV1Processor` ; downstream handlers should work with normalized request values.
- Call `getCookies()` before request handling when cookie data must be available through request headers.
- Call `setCookiesToResult()` after response processing so response cookies are written to the Lambda result.
- Preserve the `Headers` return type from `getHeaders()` rather than converting headers into a plain object unless required by an integration.

- Treat query strings and cookies as transport data; avoid rebuilding them in application handlers.