LatticeV2Processor

August 17, 2026

LatticeV2Processor

Kind: Class

Source: `src/adapter/aws-lambda/handler.ts`

Part of: Adapter

`LatticeV2Processor` adapts an AWS VPC Lattice event into request data consumed by the Lambda handler. It reads the request path, method, query string, headers, and cookies, then writes response cookies back to the Lambda result.

Extends: `EventProcessor`

Methods

Method	Signature	Returns
`getPath`	`getPath(event: LatticeProxyEventV2)`	`string`
`getMethod`	`getMethod(event: LatticeProxyEventV2)`	`string`
`getQueryString`	`getQueryString()`	`string`
`getHeaders`	`getHeaders(event: LatticeProxyEventV2)`	`Headers`
`getCookies`	`getCookies()`	`void`
`setCookiesToResult`	`setCookiesToResult(result: APIGatewayProxyResult, cookies: string[])`	`void`

Diagram

```
graph LR
  Event[AWS VPC Lattice event] --> Processor[LatticeV2Processor]
  Processor --> Path[getPath()]
  Processor --> Method[getMethod()]
  Processor --> Query[getQueryString()]
  Processor --> Headers[getHeaders()]
```

```
  Processor --> Cookies[getCookies()]
  Processor --> Result[Lambda response]
  Cookies --> SetCookies[setCookiesToResult()]
  SetCookies --> Result
```

Usage

```ts
import { LatticeV2Processor } from './handler'

function inspectRequest(processor: LatticeV2Processor) {
  const path = processor.getPath()
  const method = processor.getMethod()
  const queryString = processor.getQueryString()
  const headers = processor.getHeaders()

  processor.getCookies()

  console.log({
    path,
    method,
    queryString,
    contentType: headers.get('content-type'),
  })

  processor.setCookiesToResult()
}
```

AI Coding Instructions

- Keep request access behind `LatticeV2Processor` methods instead of reading the VPC Lattice event directly in handler logic.
- Treat `getHeaders()` as the source for request header lookups and use the `Headers` API for case-insensitive access.
- Call `getCookies()` before code that depends on request cookies.
- Call `setCookiesToResult()` after response cookies are prepared so they are included in the Lambda result.
- Keep VPC Lattice-specific event handling in `src/adapter/aws-lambda/handler.ts`.

How it works

`LatticeV2Processor` is the exported `EventProcessor<LatticeProxyEventV2>` specialization for Lambda events whose request context has a `serviceArn` property. `getProcessor()` selects its shared instance after excluding ALB and API Gateway v2 event shapes.

src/adapter/aws-lambda/handler.ts:584-623 src/adapter/aws-lambda/handler.ts:625-657

Its event type requires a path, HTTP method, array-valued headers and query-string parameters, nullable body, base64 flag, and a Lattice v2 request context. That context includes `serviceArn`, `serviceNetworkArn`, `targetGroupArn`, region, timestamp, and identity fields. src/adapter/aws-lambda/handler.ts:29-38 src/adapter/aws-lambda/types.ts:155-173