EventV2Processor

August 17, 2026

EventV2Processor

Kind: Class

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

Part of: Adapter

This class adapts AWS Lambda HTTP request data into the request values consumed by the handler layer. It exposes path, method, query string, headers, and cookie handling while writing response cookies back to the Lambda result.

Extends: `EventProcessor`

Methods

Method	Signature	Returns
`getPath`	`getPath(event: APIGatewayProxyEventV2)`	`string`
`getMethod`	`getMethod(event: APIGatewayProxyEventV2)`	`string`
`getQueryString`	`getQueryString(event: APIGatewayProxyEventV2)`	`string`
`getCookies`	`getCookies(event: APIGatewayProxyEventV2, headers: Headers)`	`void`
`setCookiesToResult`	`setCookiesToResult(result: APIGatewayProxyResult, cookies: string[])`	`void`
`getHeaders`	`getHeaders(event: APIGatewayProxyEventV2)`	`Headers`

Diagram

```
graph LR
  Event["EventV2Processor [census]"] --> Request["Request accessors"]
  Event --> Cookies["Cookie handling"]
  Request --> Path["getPath"]
  Request --> Method["getMethod"]
```

```
  Request --> Query["getQueryString"]
  Request --> Headers["getHeaders"]
  Cookies --> ReadCookies["getCookies"]
  Cookies --> WriteCookies["setCookiesToResult"]
```

Usage

```
declare const processor: EventV2Processor; // [census]

const requestDetails = {
  path: processor.getPath(),
  method: processor.getMethod(),
  queryString: processor.getQueryString(),
  contentType: processor.getHeaders().get("content-type"),
};

processor.getCookies();

// Handle the request and prepare response cookies.

processor.setCookiesToResult();
```

AI Coding Instructions

- Keep AWS Lambda event-specific parsing inside this adapter layer.
- Read cookies before handler code depends on them, then write response cookies before returning the Lambda result.
- Treat `getHeaders()` as the source for request header access rather than reading the Lambda event directly.
- Preserve the existing path, method, and query string mapping behavior when changing request handling.

How it works

`EventV2Processor` is an exported concrete `EventProcessor` subclass for `APIGatewayProxyEventV2` events. It supplies the v2-specific path, method, query, header, and cookie mappings used by the inherited request/result conversion methods. src/adapter/aws-lambda/handler.ts:404-439

- `getProcessor()` selects the module-level `EventV2Processor` instance when an event has both `rawPath` and `requestContext.http`, after checking whether it is an ALB

event. src/adapter/aws-lambda/handler.ts:625-637 src/adapter/aws-lambda/handler.ts:646-651

- For a v2 event, it takes the request path from `rawPath`, the HTTP method from `requestContext.http.method`, and the query string directly from `rawQueryString`. src/adapter/aws-lambda/handler.ts:405-415
- Its header conversion starts with a new `Headers` object. If `event.cookies` is an array, it writes one `Cookie` header whose value joins the entries with `; `. It then copies each truthy entry from `event.headers` with `Headers.set()`. src/adapter/aws-lambda/handler.ts:417-438
- Through `EventProcessor.createRequest()`, the processor constructs a URL as `https://${domainName}${rawPath}`, appending `?${rawQueryString}` only when that string is non-empty. The domain comes first from `event.requestContext.domainName`; otherwise the shared logic checks `host` in `headers`, then `multiValueHeaders`. src/adapter/aws-lambda/handler.ts:301-323
- When `event.body` is truthy, the inherited request conversion decodes it from base64 if `isBase64Encoded` is true; otherwise it UTF-8 encodes the string. It assigns that byte sequence as the `Request` body and sets `content-length` to its byte length. src/adapter/aws-lambda/handler.ts:327-341
- The inherited result conversion marks a response body as base64-encoded when the configured binary-content predicate, or the default predicate, classifies its `content-type` as binary. If it is not already marked binary, a non-`identity` `content-encoding` also marks it binary. Binary bodies are read as an `ArrayBuffer` and base64 encoded; other bodies are read as text. src/adapter/aws-lambda/handler.ts:344-360 src/adapter/aws-lambda/handler.ts:666-674
- For response cookies, this subclass writes extracted `Set-Cookie` values to the result's `cookies` array. The shared logic removes `set-cookie` from the response headers before copying remaining headers into the Lambda result. src/adapter/aws-lambda/handler.ts:388-400 src/adapter/aws-lambda/handler.ts:423-425
- In the standard `handle()` path, exceptions while creating the request or reading its request context are logged. A `TypeError` becomes a `400` result with body `Invalid request`; other caught errors become a `500` result with body `Internal Server Error`. src/adapter/aws-lambda/handler.ts:252-266 An invalid v2 header name is covered by a test that expects this `400` result. src/adapter/aws-lambda/handler.test.ts:441-467