

CloudFrontResponse

August 17, 2026

CloudFrontResponse

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

`CloudFrontResponse` represents the response object returned by a `Lambda@Edge` handler. It carries CloudFront-formatted headers and string status details that CloudFront reads when sending a response.

Properties

Property	Type
<code>headers</code>	<code>CloudFrontHeaders</code>
<code>status</code>	<code>string</code>
<code>statusDescription</code>	<code>string</code>

Diagram

```
graph LR
  Handler[Lambda Edge Handler] --> Response[CloudFrontResponse]
  Response --> Headers[CloudFrontHeaders]
  Response --> Status[status]
  Response --> Description[statusDescription]
  Response --> CloudFront[CloudFront]
```

Usage

```
import type { CloudFrontHeaders, CloudFrontResponse } from './handler';

function createResponse(
```

```

headers: CloudFrontHeaders,
status: string,
statusDescription: string,
): CloudFrontResponse {
  return {
    headers,
    status,
    statusDescription,
  };
}

const response = createResponse(
  requestHeaders,
  responseStatus,
  responseDescription,
);

```

AI Coding Instructions

- Return `CloudFrontResponse` from `Lambda@Edge` handlers when constructing a CloudFront response.
- Keep `headers` in the `CloudFrontHeaders` format expected by CloudFront.
- Set both `status` and `statusDescription` as strings.
- Preserve header names and values required by the response path.

How it works

`CloudFrontResponse` is an exported TypeScript interface for the optional `response` member of a CloudFront event record's `cf` object. [src/adapters/lambd-edge/handler.ts:56-60](#) [src/adapters/lambd-edge/handler.ts:69-75](#)

- `headers` is required and maps header names to arrays of `{ key, value }` entries, allowing multiple values for one header name. [src/adapters/lambd-edge/handler.ts:10-17](#) [src/adapters/lambd-edge/handler.ts:56-58](#)
- `status` is required and typed as a string. `statusDescription` is optional and typed as a string. [src/adapters/lambd-edge/handler.ts:56-60](#)
- It declares no body or body-encoding fields. [src/adapters/lambd-edge/handler.ts:56-60](#)

When `handle()` invokes the Hono application, it places the incoming event's optional `cf.response` value in the fetch context as `response`. [src/adapters/lambd-edge/handler.ts:128-141](#) The adapter does not otherwise read, validate, transform, or return that value in this file; response conversion instead operates on the `Response`

returned by the application and creates a separate internal `CloudFrontResult` shape.
[src/adapters/lambda-edge/handler.ts:149-162](#)