

ClientResponse

August 17, 2026

ClientResponse

Kind: Interface

Source: `src/client/types.ts`

Part of: [Client](#)

`ClientResponse` is the typed response returned by the client layer. It extends the standard response shape with status-aware `ok` values and format-aware body readers such as `json()` and `text()`.

Properties

Property	Type
<code>body</code>	<code>`ReadableStream</code>
<code>bodyUsed</code>	<code>boolean</code>
<code>ok</code>	<code>U extends SuccessStatusCode ? true : U extends Exclude<StatusCode, SuccessStatusCode> ? false : boolean</code>
<code>redirected</code>	<code>boolean</code>
<code>status</code>	<code>U</code>
<code>statusText</code>	<code>string</code>
<code>type</code>	<code>`basic'</code>
<code>headers</code>	<code>Headers</code>
<code>url</code>	<code>string</code>

Diagram

```
graph LR
  ClientResponse --> Metadata[status, statusText, url, headers]
  ClientResponse --> Body[body, bodyUsed]
```

```
ClientResponse --> Status[ok derived from status]
ClientResponse --> Readers[json, text, blob, formData, bytes, arrayBuffer]
ClientResponse --> ResponseOps[clone, redirect]
Readers --> Payload[Typed payload based on response format]
```

Usage

```
type User = {
  id: string
  name: string
}

async function handleUserResponse(
  response: ClientResponse<User, SuccessStatusCode, 'json'>
) {
  if (!response.ok) {
    throw new Error(`Request failed: ${response.status} ${response.statusText}`)
  }

  const user = await response.json()

  console.log(response.url)
  console.log(user.name)
}
```

AI Coding Instructions

- Preserve the inferred `ClientResponse<T, U, F>` type from client route calls so `status`, `ok`, and body reader return types remain specific.
- Check `response.ok` or `response.status` before reading a successful payload.
- Call `json()` only for JSON response formats; `text()` has separate type behavior for text responses.
- A body reader consumes the response body. Use `clone()` before reading when the body must be read again.
- Read response metadata from `headers`, `url`, `redirected`, and `statusText` rather than reconstructing it from request state.