

Accept

August 17, 2026

Accept

Kind: Interface

Source: `src/utils/accept.ts`

Part of: [Utils](#)

`Accept` represents a parsed media type from an HTTP `Accept` header. It stores the media `type`, associated parameters, and the `q` quality value used when selecting a preferred response format.

Properties

Property	Type
<code>type</code>	<code>string</code>
<code>params</code>	<code>Record<string, string></code>
<code>q</code>	<code>number</code>

Diagram

```
graph LR
  Header[HTTP Accept header] --> Accept[Accept]
  Accept --> Type["Type[type: string]"]
  Accept --> Params["Params[params: Record<string, string>]"]
  Accept --> Quality["Quality[q: number]"]
```

Usage

```
import type { Accept } from "../utils/accept";

const acceptedType: Accept = {
  type: "application/json",
```

```
params: {  
  charset: "utf-8",  
},  
q: 1,  
};  
  
if (acceptedType.type === "application/json" && acceptedType.q > 0) {  
  console.log("Return a JSON response");  
}
```

AI Coding Instructions

- Treat `type` as the parsed media type, such as `application/json` or `text/html`.
- Store media-type parameters in `params` as string key-value pairs.
- Use `q` when comparing acceptable response formats; higher values indicate higher preference.
- Preserve parsed parameter values as strings rather than coercing them to other types.
- Keep this interface aligned with the `Accept` header parsing and response content-negotiation code.