PATCH request method

- Syntax
- Examples
  - Successfully modifying a resource
- Specifications
- Browser compatibility
- See also

The `PATCH` HTTP method applies partial modifications to a resource.

`PATCH` is somewhat analogous to the "update" concept found in {{Glossary("CRUD")}} (in general, HTTP is different than {{Glossary("CRUD")}}, and the two should not be confused).

In comparison with {{HTTPMethod("PUT")}}, a `PATCH` serves as a set of instructions for modifying a resource, whereas `PUT` represents a complete replacement of the resource. A `PUT` request is always {{Glossary("idempotent")}} (repeating the same request multiple times results in the resource remaining in the same state), whereas a `PATCH` request may not always be idempotent. For instance, if a resource includes an auto-incrementing counter, a `PUT` request will overwrite the counter (since it replaces the entire resource), but a `PATCH` request may not.

Like {{HTTPMethod("POST")}}, a `PATCH` request can potentially have side effects on other resources.

A server can advertise support for `PATCH` by adding it to the list in the {{HTTPHeader("Allow")}} or {{HTTPHeader("Access-Control-Allow-Methods")}} (for CORS) response headers. Another implicit indication that `PATCH` is supported is the {{HTTPHeader("Accept-Patch")}} header (usually after an {{HTTPMethod("OPTIONS")}} request on a resource), which lists the media-types the server is able to understand in a `PATCH` request for a resource.

Request has body	Yes
Successful response has body	May
{{Glossary("Safe/HTTP", "Safe")}}	No
{{Glossary("Idempotent")}}	No
{{Glossary("Cacheable")}}	Only if freshness information is included

Allowed in HTML forms	No

Syntax

```
PATCH <request-target>["?"<query>] HTTP/1.1
```

- `<request-target>`
  - : Identifies the target resource of the request when combined with the information provided in the {{HTTPHeader("Host")}} header. This is an absolute path (e.g., `/path/to/file.html`) in requests to an origin server, and an absolute URL in requests to proxies (e.g., `http://www.example.com/path/to/file.html`).
- `<query>` {{optional_inline}}
  - : An optional query component preceded by a question-mark `?`. Often used to carry identifying information in the form of `key=value` pairs.

Examples

Successfully modifying a resource

Assume there is a resource on the server representing a user with a numeric ID of `123` in the following format:

```
{
  "firstName": "Example",
  "LastName": "User",
  "userId": 123,
  "signupDate": "2024-09-09T21:48:58Z",
  "status": "active",
  "registeredDevice": {
    "id": 1,
    "name": "personal",
    "manufacturer": {
      "name": "Hardware corp"
    }
  }
}
```

Instead of sending a JSON object to fully overwrite a resource, a `PATCH` modifies only specific parts of the resource. This request updates the `status` field:

```
PATCH /users/123 HTTP/1.1
Host: example.com
Content-Type: application/json
Content-Length: 27
Authorization: Bearer ABC123

{
  "status": "suspended"
}
```

The interpretation and authentication of the `PATCH` request depend on the implementation. Success can be indicated by any of the successful response status codes. In this example, a {{HTTPStatus("204", "204 No Content")}} is used as there's no need to transmit a body with additional context about the operation. An {{HTTPHeader("ETag")}} is provided so the caller can perform a conditional request in future:

```
HTTP/1.1 204 No Content
Content-Location: /users/123
ETag: "e0023aa4f"
```

Specifications

{{Specifications}}

Browser compatibility

The browser doesn't use the `PATCH` method for user-initiated actions, so "browser compatibility" doesn't apply. Developers can set this request method using `fetch()`.

See also

- HTTP request methods
- HTTP response status codes

- HTTP headers
- {{HTTPStatus("204")}}
- {{HTTPHeader("Allow")}}, {{HTTPHeader("Access-Control-Allow-Methods")}} headers
- {{HTTPHeader("Accept-Patch")}} – specifies the patch document formats accepted by the server
- JSON Patch Generator

□□□ #2
□ □□□ □□□ 2026-10-31 03:27:53 UTC
□ □□□ □□□ 2026-10-31 03:27:53 UTC