OPTIONS request method

- Syntax
- Examples
  - Identifying allowed request methods
  - Preflighted requests in CORS
- Specifications
- Browser compatibility
- See also

The `OPTIONS` HTTP method requests permitted communication options for a given URL or server. This can be used to test the allowed HTTP methods for a request, or to determine whether a request would succeed when making a CORS preflighted request. A client can specify a URL with this method, or an asterisk (`*`) to refer to the entire server.

Request has body	May*
Successful response has body	May
{{Glossary("Safe/HTTP", "Safe")}}	Yes
{{Glossary("Idempotent")}}	Yes
{{Glossary("Cacheable")}}	No
Allowed in HTML forms	No

* Although an `OPTIONS` message with a request body is technically allowed, it has no defined semantics. You may include a body in an `OPTIONS` message as long as you provide a valid {{HTTPHeader("Content-Type")}} header, and when you know the server expects it, as behavior is implementation-specific.

Syntax

```
OPTIONS *|<request-target>["?"<query>] HTTP/1.1
```

The request target may be either in 'asterisk form' `*` indicating the whole server, or a request target as is common with other methods:

- `*`
  - : Indicates that the client wishes to request `OPTIONS` for the server as a whole, as opposed to a specific named resource of that server.
- `<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

Identifying allowed request methods

To find out which request methods a server supports, one can use the `curl` command-line program to issue an `OPTIONS` request:

```
curl -X OPTIONS https://httpbin.org/ -i
```

This creates the following HTTP request:

```
OPTIONS / HTTP/2
Host: httpbin.org
User-Agent: curl/8.7.1
Accept: */*
```

An example response contains an {{HTTPHeader("Allow")}} header that lists the allowed methods (the exact response headers may vary):

```
HTTP/2 200
content-type: text/html; charset=utf-8
content-length: 0
allow: HEAD, OPTIONS, GET
```

Preflighted requests in CORS

In CORS, a preflight request is sent with the `OPTIONS` method so that the server can respond if it is acceptable to send the request. In this example, we will request permission for these parameters:

- The {{HTTPHeader("Access-Control-Request-Method")}} header sent in the preflight request tells the server that when the actual request is sent, it will have a {{HTTPMethod("POST")}} request method.
- The {{HTTPHeader("Access-Control-Request-Headers")}} header tells the server that when the actual request is sent, it will have the `X-PINGOTHER` and `Content-Type` headers.

```
OPTIONS /resources/post-here/ HTTP/1.1
Host: bar.example
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8
Accept-Language: en-us,en;q=0.5
Accept-Encoding: gzip,deflate
Connection: keep-alive
Origin: https://foo.example
Access-Control-Request-Method: POST
Access-Control-Request-Headers: content-type,x-pingother
```

The server now can respond if it will accept a request under these circumstances. In this example, the server response says that:

- {{HTTPHeader("Access-Control-Allow-Origin")}}
  - : The `https://foo.example` origin is permitted to request the `bar.example/resources/post-here/` URL via the following:
- {{HTTPHeader("Access-Control-Allow-Methods")}}
  - : {{HTTPMethod("POST")}}, {{HTTPMethod("GET")}}, and `OPTIONS` are permitted methods for the URL. (This header is similar to the {{HTTPHeader("Allow")}} response header, but used only for CORS.)
- {{HTTPHeader("Access-Control-Allow-Headers")}}
  - : `X-PINGOTHER` and `Content-Type` are permitted request headers for the URL.
- {{HTTPHeader("Access-Control-Max-Age")}}
  - : The above permissions may be cached for 86,400 seconds (1 day).

```
HTTP/1.1 200 OK
Date: Mon, 01 Dec 2008 01:15:39 GMT
Server: Apache/2.0.61 (Unix)
Access-Control-Allow-Origin: https://foo.example
Access-Control-Allow-Methods: POST, GET, OPTIONS
Access-Control-Allow-Headers: X-PINGOTHER, Content-Type
```

```
Access-Control-Max-Age: 86400
Vary: Accept-Encoding, Origin
Keep-Alive: timeout=2, max=100
Connection: Keep-Alive
```

> [!NOTE] Both {{HTTPStatus("200", "200 OK")}} and {{HTTPStatus("204", "204 No Content")}} are permitted status codes, but some browsers incorrectly believe `204 No Content` applies to the resource and do not send a subsequent request to fetch it.

Specifications

{{Specifications}}

Browser compatibility

{{Compat}}

See also

- HTTP request methods
- HTTP response status codes
- HTTP headers
- {{HTTPHeader("Allow")}} header
- CORS

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