

title: HTTP Monitor description: HTTP monitor can monitor any HTTP endpoint like API, Web URL, etc.

The HTTP monitor sends a single HTTP request to the specified URL with configured request settings. It records the response data like headers, body, error if any and the various timings.

## Configuration

### General

- **Name** - Name of the monitor. Max length is 255 characters.
- **HTTP Method** - HTTP Method.
- **URL** - URL of the HTTP monitor of format `<http(s)>://hostname:port/path`.
- **Interval** - Monitor execution interval.
- **Locations** - List of locations to run the monitor.

### Request Settings

- **Headers** - List of HTTP headers to be sent.
- **Query Params** - List of request query parameters to be appended to the URL.
- **Cookies** - List of Cookies to be sent.
- **Body** - Request body data. Max length is 4096 characters.

Optionally, the header, query param, and cookie values can be marked as hidden, which means it will be hidden for the users. Currently, we do not encrypt the hidden values while storing it in the DB.

Max length of the header, query param, and cookie name and value are 512 and 1024 characters respectively. A maximum of 100 rows for each of the headers, query params, and cookies is allowed.

By default, the HTTP monitor adds the headers below for all requests sent from the agent:

Name	Value	Description
x-sematext-origin	synthetics	Origin of the request. Can be used to identify Synthetics requests in the back end or filter the requests in a firewall.
x-sematext-synthetics- <del>id</del> run-id		Uniquely identifies this request. Can be used for tracing and correlation in the back end.

## Conditions

Conditions can be configured on the response fields and metrics. The conditions are evaluated for every run result. All the conditions should pass for a run to be declared a passed. If any condition fails, the run will fail and the monitor will be marked as failing. Condition types supported are:

- **Error** - During each run, if there are any errors like connection, timeout, etc., it will be recorded in the error field of the response. This does not include the error returned as part of the response body.
- **Response Code** - HTTP Status Code. Used to make sure the response code should be of specific value or range.
- **Metric** - Used to make sure the metrics are within the expected range.
- **Response Header** - Can be used to compare the value of a specific header field against the expected value.
- **Response Body** - Compare the response body against the specified value using an operator.
- **Response Body JSON** - If the response body is JSON, JSONPath can be used to extract a specific value and compare against the expected value.

Various operators like Equals, Less Than, Greater Than, Contains are supported based on condition type.

By default, the UI adds the below conditions while creating an HTTP monitor. You can change them during the creation:

- Error equals empty.
- Response Code equals 200.
- Response Time metric Less Than 10000 ms.

## Run environment

Each HTTP monitor run is executed in an isolated environment. Go HTTP Client library is used to execute the HTTP requests. Each run is allocated 128MB memory.

The default values for various timeouts are:

- **Socket Connection Timeout** - 3 seconds.
- **TLS Handshake Timeout** - 5 seconds.
- **Total Timeout** - Timeout for complete request-response operation - 20 seconds.