Traefik Errors Documentation

- Configuration Examples
- Configuration Options
  - service and HostHeader
  - statusRewrites
  - query
  - errorRequestHeaders

The `errors` middleware returns a custom page in lieu of the default, according to configured ranges of HTTP Status codes.

Configuration Examples

```
# Dynamic Custom Error Page for 5XX Status Code excluding 502 and 504
http:
  middlewares:
    test-errors:
      errors:
        status:
          - "500"
          - "501"
          - "503"
          - "505-599"
        statusRewrites:
          "418": 404
          "502-504": 500
        service: error-handler-service
        query: "/{status}.html"
```

```
  services:
    # ... definition of the error-handler-service
```

```
# Dynamic Custom Error Page for 5XX Status Code excluding 502 and 504
[http.middlewares]
  [http.middlewares.test-errors.errors]
    status = ["500","501","503","505-599"]
    service = "error-handler-service"
    query = "/{status}.html"

    [http.middlewares.test-errors.errors.statusRewrites]
      "418" = 404
      "502-504" = 500

[http.services]
  # ... definition of the error-handler-service
```

```
# Dynamic Custom Error Page for 5XX Status Code
labels:
  - "traefik.http.middlewares.test-errors.errors.status=500,501,503,505-599"
  - "traefik.http.middlewares.test-errors.errors.statusRewrites.418=404"
  - "traefik.http.middlewares.test-errors.errors.statusRewrites.502-504=500"
  - "traefik.http.middlewares.test-errors.errors.service=error-handler-service"
  - "traefik.http.middlewares.test-errors.errors.query=/{status}.html"
```

```
// Dynamic Custom Error Page for 5XX Status Code excluding 502 and 504
{
  // ...
  "Tags": [
    "traefik.http.middlewares.test-errors.errors.status=500,501,503,505-599",
    "traefik.http.middlewares.test-errors.errors.statusRewrites.418=404",
    "traefik.http.middlewares.test-errors.errors.statusRewrites.502-504=500",
    "traefik.http.middlewares.test-errors.errors.service=error-handler-service",
    "traefik.http.middlewares.test-errors.errors.query=/{status}.html"
  ]

}
```

```yaml
apiVersion: traefik.io/v1alpha1
kind: Middleware
metadata:
  name: test-errors
spec:
  errors:
    status:
      - "500"
      - "501"
      - "503"
      - "505-599"
    statusRewrites:
      "418": 404
      "502-504": 500
    query: /{status}.html
    service:
      name: error-handler-service
      port: 80
```

Configuration Options

Field	Description	Default	Required
`status`	Defines which status or range of statuses should result in an error page. The status code ranges are inclusive (`505-599` will trigger with every code between `505` and `599`, `505` and `599` included). You can define either a status code as a number (`500`), as multiple comma-separated numbers (`500,502`), as ranges by separating two codes with a dash (`505-599`), or a combination of the two (`404,418,505-599`).	[]	No

Field	Description	Default	Required
`statusRewrites`	An optional mapping of status codes to be rewritten. More information here.	[]	No
`service`	The service that will serve the new requested error page. More information here.	""	Yes
`query`	The URL for the error page (hosted by `service`). More information here	""	No
`errorRequestHeaders`	Defines the list of original request headers forwarded to the error page service. More information here	[]	No

service and HostHeader

By default, the client `Host` header value is forwarded to the configured error service. To forward the `Host` value corresponding to the configured error service URL, the `passHostHeader` option must be set to `false`.

!!!info "Kubernetes" When specifying a service in Kubernetes (e.g., in an IngressRoute), you need to reference the `name`, `namespace`, and `port` of your Kubernetes Service resource. For example, `my-service.my-namespace@kubernetescrd` (or `my-service.my-namespace@kubernetescrd:80`) ensures that requests go to the correct service and port.

!!! info "ServersTransport (Kubernetes)"

```
To customize how Traefik connects to the error page service (for example, to configure TLS to the backend), set a [`serversTransport`](../../kubernetes/crd/http/serverstransport.md) on the middleware's `service`.
The `traefik.ingress.kubernetes.io/service.serverstransport` annotation on the Kubernetes Service is not applied here: it only affects a Service used as an Ingress backend, not one referenced by a middleware.

```yaml
apiVersion: traefik.io/v1alpha1
kind: Middleware
metadata:
```
```

```
  name: test-errors
spec:
  errors:
    status:
      - "500"
      - "501"
    service:
      name: error-handler-service
      port: 80
      serversTransport: mytransport
```

statusRewrites

`statusRewrites` is an optional mapping of status codes to be rewritten.

For example, if a service returns a 418, you might want to rewrite it to a 404. You can map individual status codes or even ranges to a different status code.

The syntax for ranges follows the same rules as the `status` option.

query

There are multiple variables that can be placed in the `query` option to insert values in the URL.

The table below lists all the available variables and their associated values.

Variable	Value
`{status}`	The response status code.
`{originalStatus}`	The original response status code, if it has been modified by the `statusRewrites` option.
`{url}`	The escaped request URL.

errorRequestHeaders

Defines the list of original request headers forwarded to the error page service.

By default (`errorRequestHeaders` not set), all request headers — including authentication material such as `Authorization` and `Cookie` — are forwarded. If the error page service is in a separate trust

domain, use this option to restrict which headers cross the service boundary.

Set to an explicit list to forward only those headers, or set to an empty list (`errorRequestHeaders: []`) to forward no headers.

□□□ #2
□ □□□ □□□ 2026-10-31 14:03:02 UTC
□ □□□ □□□ 2026-10-31 14:03:03 UTC