

Express-Status Codes

1. Informational Responses (100-199)

- These indicate a provisional response, typically sent before the final response is available.
- Example: 101 Switching Protocols

2. Successful Responses (200-299)

- This group includes codes for successful actions like GET, POST, PUT, and DELETE operations that were successfully executed on the server.
- Example:
 - **200 OK:** Request was successful.
 - **201 Created:** Resource created successfully.
 - **202 Accepted:** The request has been accepted for processing, but the processing has not been completed.
 - **204 No Content:** Server successfully processed the request, but is not returning any content.

3. Redirection Messages (300-399)

- Codes in this range are used when a different URI should be used to access the same resource. This can happen after a resource has been moved or if the server requires additional information before fulfilling the request.
- Example:
 - **301 Moved Permanently:** Resource is available at another URI and will not change again (the client should update its links).
 - **302 Found:** Temporarily redirect to the new location. Can be repeated indefinitely.
 - **303 See Other:** The response should be accessed using a different HTTP method (GET for instance) or in a different location.

4. Client Error Responses (400-499)

- These status codes are used when there is a client-side error that prevents the request from being fulfilled as requested by the client.
- Example:
 - **401 Unauthorized:** Authentication credentials were missing or incorrect.
 - **403 Forbidden:** Request was not accepted because of some issue on server side but it does not relate to the authentication.
 - **404 Not Found:** The requested resource could not be found.

5. Server Error Responses (500-599)

- Used when there is an error on the server-side that prevented the request from being fulfilled.
- Example:
 - **500 Internal Server Error:** A generic error code, indicating something went wrong at the server but doesn't give more details about what's happening.

Here's how you might implement some of these status codes in Express.js using Node.js:

```
const express = require('express');
const app = express();

// For a simple GET request that returns the 200 OK status code:
app.get('/', (req, res) => {
  res.status(200).send({ message: "Server is working!" });
});

// For a POST request where something was created, so it should return 201 Created:
app.post('/users', (req, res) => {
  // Assuming 'createUser' function creates a new user and returns an ID.
  const id = createUser(req.body);
  res.status(201).send({ message: "New User has been created", userId: id });
});

// For redirection to another URI:
app.get('/old-url', (req, res) => {
  res.redirect(301, '/new-url'); // 301 is a permanent redirect.
});
```

This summary provides an overview of the main status codes used in HTTP for expressing the outcome of requests and gives examples of how they might be utilized in Express.js.

Curated by Brajesh Kumar