

HTTP 400 Series Status Codes - Explained

400 - Bad Request

Meaning:

The server cannot process the request due to malformed syntax, invalid input, or other client-side issues.

Common Causes:

- Invalid JSON or malformed body
- Missing required fields or parameters
- Invalid query string format
- Incorrect headers (e.g., wrong Content-Type)
- Too long URL
- Corrupted cookies
- Improper path variables

Example:

POST /api/login

Content-Type: application/json

```
{ "username": "john", "password": 1234
```

Fixes:

- Ensure JSON is valid
- Check required query/path/body fields
- Use proper headers

- Validate and encode all input

401 - Unauthorized

Meaning:

Authentication is required or has failed. The client must provide valid credentials.

Common Causes:

- Missing Authorization header
- Expired/invalid token
- Malformed Basic Auth

Example:

GET /user/profile -> 401 Unauthorized

Fixes:

- Provide a valid token (e.g., Bearer Token)
- Handle login and token refresh properly

402 - Payment Required

Meaning:

Reserved for future use; some APIs use it for subscription or quota issues.

Common Uses (Custom APIs):

- Subscription expired
- Free-tier usage limit exceeded
- Premium feature access denied

Example:

GET /api/premium-report -> 402 Payment Required

Fixes:

- Prompt user to upgrade or pay
- Provide clear error messaging

403 - Forbidden

Meaning:

Server understood the request and credentials are valid, but access is denied.

Common Causes:

- Insufficient permissions
- Blocked IP or region
- Resource is private

Example:

GET /admin-dashboard -> 403 Forbidden

Fixes:

- Check user role or scope
- Ensure API key or token has proper access

404 - Not Found

Meaning:

The server can't find the requested resource. URL might be incorrect or deleted.

Common Causes:

- Typo in URL
- Resource no longer exists
- Route not defined in backend

Example:

GET /api/products/9999 -> 404 Not Found

Fixes:

- Validate resource existence
- Check endpoint spelling and structure

405 - Method Not Allowed

Meaning:

HTTP method (GET, POST, etc.) is not allowed on the specified resource.

Common Causes:

- Sending POST where only GET is allowed
- Calling DELETE on a read-only endpoint

Example:

POST /readonly-data -> 405 Method Not Allowed

Allow: GET

Fixes:

- Use the correct HTTP method
- Refer to the Allow response header