A proper Error handling makes an application most helpful for users from the user experience aspect. Error handlers report a user correctly what causing the problem vise verse. It also notifies developers in the development phase about the error reason, which assists them to fix it before it goes live.

There are two types of errors which might occur in an application, inside errors and outside errors.

**Outside Errors:** These errors often occur from the server-side and usually starts with (5xxx) status code. These errors return the error message with the proper reason so identifying these errors is easy.

**Inside Error:**When something unexpected happens, then these errors are thrown. These errors occur with the proper message, which contains the reason for the error occurrence.

Next, we practically understand how to handle client-side errors in Angular. A Client-side error handling is one of the features among other various features for developing a user friendly application.

A user gets frustrated when suddenly an app stopped working, and the user doesn’t find the reason for it. It downgrades the overall user experience of the application. To enhance the user experience, we must properly handle errors.

We will use the regular error handling approaches such as handling errors with RxJS operators catchError and throwError and managing errors with HttpClient and HttpInterceptor.

## Handle Errors in Angular with HttpClient and RxJS

The simplest way to handle errors in Angular is to use [Angular’s HttpClient service](https://www.positronx.io/angular-8-httpclient-http-tutorial-build-consume-restful-api/) along with RxJS operators throwError and catchError. Http request is made, and it returns the data with a response if anything wrong happens then it returns an error object with error status code.

4xx Client errors

For the Mr. Robot episodes named after several 4xx errors, see Mr. Robot (season 4).

This class of status code is intended for situations in which the error seems to have been caused by the client. Except when responding to a HEAD request, the server should include an entity containing an explanation of the error situation, and whether it is a temporary or permanent condition. These status codes are applicable to any request method. User agents should display any included entity to the user.[30]

**400 Bad Request**

The server cannot or will not process the request due to an apparent client error (e.g., malformed request syntax, size too large, invalid request message framing, or deceptive request routing).[31]

**401 Unauthorized**

Similar to 403 Forbidden, but specifically for use when authentication is required and has failed or has not yet been provided. The response must include a WWW-Authenticate header field containing a challenge applicable to the requested resource. See Basic access authentication and Digest access authentication.[32] 401 semantically means "unauthorised",[33] the user does not have valid authentication credentials for the target resource.

Note: Some sites incorrectly issue HTTP 401 when an IP address is banned from the website (usually the website domain) and that specific address is refused permission to access a website.[citation needed]

**402 Payment Required**

Reserved for future use. The original intention was that this code might be used as part of some form of digital cash or micropayment scheme, as proposed, for example, by GNU Taler,[34] but that has not yet happened, and this code is not usually used. Google Developers API uses this status if a particular developer has exceeded the daily limit on requests.[35] Sipgate uses this code if an account does not have sufficient funds to start a call.[36] Shopify uses this code when the store has not paid their fees and is temporarily disabled.[37] Stripe uses this code for failed payments where parameters were correct, for example blocked fraudulent payments.[38]

**403 Forbidden**

The request contained valid data and was understood by the server, but the server is refusing action. This may be due to the user not having the necessary permissions for a resource or needing an account of some sort, or attempting a prohibited action (e.g. creating a duplicate record where only one is allowed). This code is also typically used if the request provided authentication via the WWW-Authenticate header field, but the server did not accept that authentication. The request should not be repeated.

**404 Not Found**

The requested resource could not be found but may be available in the future. Subsequent requests by the client are permissible.

**405 Method Not Allowed**

A request method is not supported for the requested resource; for example, a GET request on a form that requires data to be presented via POST, or a PUT request on a read-only resource.

**406 Not Acceptable**

The requested resource is capable of generating only content not acceptable according to the Accept headers sent in the request.[39] See Content negotiation.

**407 Proxy Authentication Required (RFC 7235)**

The client must first authenticate itself with the proxy.[40]

**408 Request Timeout**

The server timed out waiting for the request. According to HTTP specifications: "The client did not produce a request within the time that the server was prepared to wait. The client MAY repeat the request without modifications at any later time."[41]

**5xx Server errors[[edit](https://en.wikipedia.org/w/index.php?title=List_of_HTTP_status_codes&action=edit&section=5" \o "Edit section: 5xx Server errors)]**

The [server](https://en.wikipedia.org/wiki/Server_(computing)) failed to fulfill a request.[[61]](https://en.wikipedia.org/wiki/List_of_HTTP_status_codes#cite_note-62)

Response status codes beginning with the digit "5" indicate cases in which the server is aware that it has encountered an error or is otherwise incapable of performing the request. Except when responding to a HEAD request, the server *should* include an entity containing an explanation of the error situation, and indicate whether it is a temporary or permanent condition. Likewise, user agents *should* display any included entity to the user. These response codes are applicable to any request method.[[62]](https://en.wikipedia.org/wiki/List_of_HTTP_status_codes#cite_note-63)

**500 Internal Server Error**

A generic error message, given when an unexpected condition was encountered and no more specific message is suitable.[[63]](https://en.wikipedia.org/wiki/List_of_HTTP_status_codes#cite_note-64)

**501 Not Implemented**

The server either does not recognize the request method, or it lacks the ability to fulfil the request. Usually this implies future availability (e.g., a new feature of a web-service API).[[64]](https://en.wikipedia.org/wiki/List_of_HTTP_status_codes#cite_note-65)

**502 Bad Gateway**

The server was acting as a [gateway](https://en.wikipedia.org/wiki/Gateway_(telecommunications)) or proxy and received an invalid response from the upstream server.[[65]](https://en.wikipedia.org/wiki/List_of_HTTP_status_codes#cite_note-66)

**503 Service Unavailable**

The server cannot handle the request (because it is overloaded or down for maintenance). Generally, this is a temporary state.[[66]](https://en.wikipedia.org/wiki/List_of_HTTP_status_codes#cite_note-67)

**504 Gateway Timeout**

The server was acting as a gateway or proxy and did not receive a timely response from the upstream server.[[67]](https://en.wikipedia.org/wiki/List_of_HTTP_status_codes#cite_note-68)

**505 HTTP Version Not Supported**

The server does not support the HTTP protocol version used in the request.[[68]](https://en.wikipedia.org/wiki/List_of_HTTP_status_codes#cite_note-69)

**506 Variant Also Negotiates (**[**RFC 2295**](https://tools.ietf.org/html/rfc2295)**)**

Transparent [content negotiation](https://en.wikipedia.org/wiki/Content_negotiation) for the request results in a [circular reference](https://en.wikipedia.org/wiki/Circular_reference).[[69]](https://en.wikipedia.org/wiki/List_of_HTTP_status_codes#cite_note-RFC_2295-70)

**507 Insufficient Storage (WebDAV;**[**RFC 4918**](https://tools.ietf.org/html/rfc4918)**)**

The server is unable to store the representation needed to complete the request.[[16]](https://en.wikipedia.org/wiki/List_of_HTTP_status_codes#cite_note-RFC_4918-17)

**508 Loop Detected (WebDAV;**[**RFC 5842**](https://tools.ietf.org/html/rfc5842)**)**

The server detected an infinite loop while processing the request (sent instead of [208 Already Reported](https://en.wikipedia.org/wiki/List_of_HTTP_status_codes#208)).

**510 Not Extended (**[**RFC 2774**](https://tools.ietf.org/html/rfc2774)**)**

Further extensions to the request are required for the server to fulfil it.[[70]](https://en.wikipedia.org/wiki/List_of_HTTP_status_codes#cite_note-RFC_2774-71)

**511 Network Authentication Required (**[**RFC 6585**](https://tools.ietf.org/html/rfc6585)**)**

The client needs to authenticate to gain network access. Intended for use by intercepting proxies used to control access to the network (e.g., "[captive portals](https://en.wikipedia.org/wiki/Captive_portal)" used to require agreement to Terms of Service before granting full Internet access via a [Wi-Fi hotspot](https://en.wikipedia.org/wiki/Hotspot_(Wi-Fi))).[[59]](https://en.wikipedia.org/wiki/List_of_HTTP_status_codes#cite_note-rfc6585-60)