HTTP status codes let us know whether our request is a success, a failure or something in between.

**Responses are grouped in five classes:**

1. Informational responses (100–199)
2. Successful responses (200–299)
3. Redirects (300–399)
4. Client errors (400–499)
5. Server errors (500–599)

**1XX** Informational responses: Server wants to send some information to the client.

**100 Continue:** When server send 100 it says I am receiving your request please continue sending your request.

for ex: If we want to send a request of 500mb in such situation after every 5mb or 7mb or 10 mb server send request to client that yes you should continue and I am receiving your request]

**101 Switching Protocols**: In this protocol Server says to the client that you are using a certain protocol but we may use a better version.

[for Ex: if client using http 1.0 and server and client both are capable of http 1.1 then server may send back a response with 101 saying that we can switch protocols for better connections].

**102 Processing:** Server says to the client, I have received your request and I am processing it, I need more time.

**103 Early hints:** It is similar to 102 but in that server not only tells the client that the request are under processing but also sends back some useful information.

**122 Request URI too long**:This status code is specific to internet explorer, If we sending a request having length more than 2032 characters then  **IE** says request URI is too long.

**2XX:** It means the action was successfully received, understood, and accepted.

**200 OK**: Request you made is a good request, it completed correctly, it did what it suppose to did.

**201 Created**: It says that client was trying to create a new resource to server and it was successful.

**202 Accepted**: It says I have accepted your request and I will process it later

[The request has been accepted for processing, but the processing has not been completed yet. ]

**203 Non-Authoritative Information :** it says that the request was successful but the enclosed information has been modified by a transforming proxy

it says that the request was successful but the enclosed information has been modified by a transforming proxy

**204 No Content**: everything went well as expected but there is nothing to return i.e api server is not going to give you information back.

**206 Partial Content**: When Range header send by the client to request only the part of the resource.

Basically when client send Range header it request just some part of the resource then the server will send this status code.

Content-Range: bytes 200-1000/67589

**207 Multi Status**: It is used when there are multiple response codes.

[For ex: Client sends request to the server and to prepare response server needs to connect to three different machines one is Database system, One is caching system and one is file system lets assume that the database system is down and it send back 500, but other two are working fine in such case server wont say it was successful or not successful it sends back 207 multisatus]

**208 Already reported** : It is similar to 207 but it is used only for WebDAV

WebDAV(Web Distributed Authoring and Versioning ) is an extension of HTTP that allows system to read and write document on the web

**3XX Redirection**: It means further action must be taken in order to complete the request.

When someone goes to some website and that web page URL is redirecting into another webpage URL ( url says!! This page is use to be here but now it’s over there)

**300 Multiple Choices: It** indicates that the request has more than one possible responses.

**301 Moved Permanently**: Permanently replacing one URL to other.

<https://www.poonawallafinance.com/contact>

It indicates that the resource requested has been definitively moved to the URL given by the [Location](https://developer.mozilla.org/en-US/docs/Web/HTTP/Headers/Location) headers.

**302 Found** : Indicates that the resource requested has been temporarily moved to the URL given by the [Location](https://developer.mozilla.org/en-US/docs/Web/HTTP/Headers/Location) header.

**304 Not Modified: It indicates that a file is unchanged since it was last accessed on the server**

**Browser make a request with the “If-Modified-Since” header, If the file hasn’t modified then, request won’t be fulfilled.**

**307 Temporary Redirect**: Temporarilyreplacing one URL to other

The only difference between 307 and [302](https://developer.mozilla.org/en-US/docs/Web/HTTP/Status/302) is that 307 guarantees that the method and the body of response will not be changed when the redirected request is made.

302 is then unpredictable on the Web, whereas the behavior with 307 is predictable.

**308 Permanent Redirect** : Indicates that the resource requested has been definitively moved to the URL given by the [Location](https://developer.mozilla.org/en-US/docs/Web/HTTP/Headers/Location) headers.

The difference between 301 and 308 is that 308 redirect is a MUST do request on the target location. If the request was a POST and had a body, then the client must do a POST request with a body on the new location.

In the case of 301 a client may do this. In practice, most clients don’t do this and convert the POST request to a GET request.

**4XX** **Client Errors** : This category of error status codes points the finger at clients.

**400 Bad Request**: it’s an error from the client side, this error says ‘something went wrong’, your information is bad’ and I am telling you it’s bad by sending 400

For ex: if we want to create a new user and we need to pass name and email and the client that using this api sends name only, so it’s a bad request, so we send him back 400 error saying you don’t give me all information I need

**401 Unauthorized**: you are trying to access something which require some type of authentication like you must be logged in or you want some API key

You didn’t pass api key/login details or paas it wrong (server send 401 and say I don’t know who you are you have to give me API key)

**403 Forbidden**: It happens when a client send a API key, but the service they’re accessing require different permission.

(for ex: there is a basic user and they want to access a admin feature, then the server give 403 and say I know who you are but you don’t have permission to access this)

**404 Not Found**: you’re trying to access that part of application which don’t exist.

(for ex: if a user search for blog page in a application but there is no blog page in that application so it gives us 404)

**405 Method Not Allowed : It** indicates that the request method is known by the server but is not supported by the target resource.

GetPIVCPolicyDetails

**408 Request Timeout** : When the request of the client takes too long, the server closes the connection and we get 408 error code.

The reason behind that is the server didn’t receive a complete request from the client within the time frame it was prepared to wait.

**409 Conflict :** indicates a request conflict with current state of the server.

Conflicts are most likely to occur in response to a [PUT](https://developer.mozilla.org/en-US/docs/Web/HTTP/Methods/PUT) request. For example, you may get a 409 response when uploading a file which is older than the one already on the server resulting in a version control conflict.

**410 Gone :** indicates that access to the target resource is no longer available at the origin server and that this condition is likely to be permanent.

A 404 status code does not indicate whether the resource is temporarily or permanently missing. But if a resource is permanently removed, a [410](https://developer.mozilla.org/en-US/docs/Web/HTTP/Status/410) (Gone) should be used instead of a 404 status.

**412 Precondition Failed :** It indicates that access to the target resource has been denied.

This happens with conditional requests, when the condition defined by the [If-Unmodified-Since](https://developer.mozilla.org/en-US/docs/Web/HTTP/Headers/If-Unmodified-Since) or [If-None-Match](https://developer.mozilla.org/en-US/docs/Web/HTTP/Headers/If-None-Match) headers is not fulfilled.

If-None-Match: "<etag\_value>"

If-None-Match: "<etag\_value>", "<etag\_value>", …

If-None-Match: \*

**415 Unsupported Media Type : It** indicates that the server refuses to accept the request because the payload format is in an unsupported format.

Type in UpdatePIVCFinalStatus

**417 Expectation Failed**: indicates that expectation given in the request's [Expect](https://developer.mozilla.org/en-US/docs/Web/HTTP/Headers/Expect) header could not be met.

PUT /somewhere/fun HTTP/1.1

Host: origin.example.com

Content-Type: video/h264

Content-Length: 1234567890987

Expect: 100-continue

**418 I'm a teapot :** This error is a reference to Hyper Text Coffee Pot Control Protocol defined in April Fools' jokes in 1998 and 2014.

<https://www.google.com/teapot>

**429 Too Many Requests :** indicates the user has sent too many requests in a given amount of time ("rate limiting").

A [Retry-After](https://developer.mozilla.org/en-US/docs/Web/HTTP/Headers/Retry-After) header might be included to this response indicating how long to wait before making a new request.

**451 Unavailable For Legal Reasons : It** indicates that the user requested a resource that is not available due to legal reasons, such as a web page for which a legal action has been issued.

**5XX** **Server errors**: Client made a good request but server didn’t complete it.

**500 Internal server error**: indicates that the server encountered an unexpected condition that prevented it from fulfilling the request.

(for ex: your database is down for some reason )

**501 Not Implemented: It indicates** that **the server does not support the functionality required to fulfil the request.**

**502 Bad Gateway:** indicates that the server, while acting as a gateway or proxy, received an invalid response from the upstream server.

When a server you are connected with tries to access another server in the chain the 502 error occur when next server in the chain cannot fulfil the request or gives an invalid response

**503 Service Unavailable:** indicates that the server is not ready to handle the request, Common causes are a server that is down for maintenance or that is overloaded.

It is similar to 500 but 503 is usually an expected error, whereas 500 is a general unexpected error.

**504 Gateway Timeout: It** indicates that the server, while acting as a gateway or proxy, did not get a response in time from the upstream server that it needed in order to complete the request.

when chained server not sending the response in time

**505 HTTP Version Not Supported : It** indicates that the HTTP version used in the request is not supported by the server.