

Error Handling

HTTP Status Codes

REST API methods return three-digit HTTP "status codes" to indicate the outcome of a request. See [RFC 2616 Section 10](#) for details about HTTP Status Codes. By convention, codes in the range 200-299 indicate success. Codes in the range 400-499 indicate an error in the request. Codes in the range 500-599 indicate a server-side error.

The HTTP status codes returned by LightWave Server™ when calling user-defined API methods may come from one of two sources:

- a response mapping in the user-defined API method
- from LightWave Server itself, in the case that the user-defined API method could not be executed

In the case of a response mapping, it's up to the API designer (developer) to [decide which HTTP status codes and message content should be returned](#) based upon the response code from the server application.

In the case of status codes returned by LightWave Server itself, the following may be returned:

Code	Name	Description
400-level errors are returned when there is a problem with the request. These are generally due to a client application programming error.		
400	Bad Request	The request is malformed. For example, required query parameters missing, body content missing or invalid JSON, etc. Also, a bad transaction id is supplied, bad Pathway dialog id, etc.
401	Unauthorized	No HTTP Authentication header was supplied and the Service has an Access Control Policy that is restricted to a particular user or group.
403	Forbidden	An HTTP Authentication header was supplied, but the Service's Access Control Policy does not permit the authenticated user (due to user, group or source IP restrictions).
404	Not Found	The supplied URI (the 'path') does not match any path belonging to any deployed Service (on the TCP/IP port).
405	Method Not Allowed	The supplied URI is valid, but the requested method (GET, POST, PUT, etc.) is not defined for the URI.
500-level codes are returned for errors that are not due to the content of the request itself -- there is nothing the client application can do to fix the problem. There are typically server configuration errors.		
500	Internal Server Error	Any error that prevents LightWave from communicating with your server application (e.g., a Pathsend or Guardian error). Other causes include license expiration.
501	Not Implemented	A method other than OPTIONS, GET, POST, PUT or DELETE was attempted.

HTTP Headers

In addition to the status code, error responses generated by LightWave Server include error details within the following HTTP headers:

HTTP Header	Description
lw-error-source	The source of the error; "lightwave", "nsk", or "pathway"
lw-error-code	The error number within the 'error source'
lw-error-subcode	Additional detail for 'error code', if applicable
lw-error-text	A textual description of the error

Error responses generated by LightWave Server have no body content, i.e. content-length = 0. Error responses returned from user-defined API methods do not include 'lw-error' headers and may include body content.

Interpreting lw-error-code

lw-error-source	lw-error-code	lw-error-subcode
-----------------	---------------	------------------

lightwave	value	name	description	n/a
	9013	deserialization error	Error converting the request JSON body content to server IPM format	
	9014	serialization error	Error converting the server response IPM content to JSON format	
	9016	API compiler error	Error in API definition	
nsk	Guardian File System Error; see the HPE Guardian Procedure Errors and Messages Manual			n/a
pathway	Pathsend Error; see "Pathsend Errors" in the HPE NonStop TS/MP 2.5 Pathsend and Server Programming Manual			Guardian File System Error, when applicable

Suppressing HTTP Status Codes

Some client frameworks will not return response content if the HTTP Status Code is not 200. To overcome this, use the [lw-suppress-status](#) parameter when invoking the request. When this parameter is present, the response status code will be 200 regardless of the outcome of the operation. The actual status information is provided in the response [lw-http-status](#) and [lw-http-reason](#) parameters. When an error occurs, additional error detail is returned in the response [lw-error](#) parameter group.

The following examples show how the [lw-suppress-status](#) parameter alters the response for an HTTP "400 Bad Request" error returned by the Process API when the target server does not exist. In this example, if [lw-suppress-status](#) were not used, the returned HTTP status code would be 400.

Request

```
POST /lightwave/api/v1/transaction/not-a-transaction-id HTTP/1.1
lw-suppress-status: 1
Content-Length: 0
```

Response

```
HTTP/1.1 200 OK
lw-http-status: 400
lw-http-reason: Bad Request
lw-error-source: nsk
lw-error-code: 78
lw-error-subcode: 0
lw-error-message: Transaction identifier is invalid or obsolete.
```