

Why does my API return a 404 Not Found status?

404 Not Found responses are well known to just about every web surfer. It simply means that the target resource of the URL is not found. However, there are different reasons and different sources for the 404 error.

Web Server Generated

One source could be your web server itself (e.g., IIS, Apache, Abyss, etc.). This means the request was never handled by the OECGI, much less your API. You'll know if this is the case rather quickly because of the lack of activity on the API side. For example, you do not see new logs being generated or you do not see output coming from the command prompt running the OEngineServer. Another sign that the 404 response was generated from a web server is to see if the content is HTML formatted. 404 responses generated by the HTTP Framework return a [RFC7807](#) standard problem+json media type response (see below for examples).

In these cases it is most likely a configuration issue. The web server might not be routing the request to the OECGI properly. Go back through the [Server Preparation](#) steps. The [URL Rewrite Rules](#) can easily fail with just one minor typo.

API Generated

Generally there are two reasons why the HTTP Framework would return a 404 response. The usual scenario is that the resource hasn't been defined in the *Resource Manager* yet. In these cases, the API will return this message:

GET <https://www.examples.org/parts>

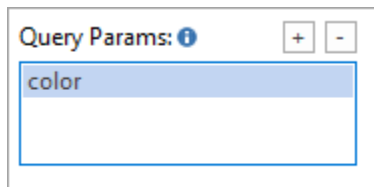
```
{
  "type": "about:blank",
  "title": "Not Found",
  "status": 404,
  "detail": "This is not a valid web API endpoint."
}
```

However, it is still possible to get a 404 response even if your resource has been defined. This can happen if the URL includes *query params* that are not defined as a part of the resource:

GET <https://www.examples.org/parts?color=red>

```
{
  "type": "about:blank",
  "title": "Not Found",
  "status": 404,
  "detail": "color is not valid for this web API endpoint.",
  "instance": "http://www.examples.org/api/parts?color=red"
}
```

By default, the HTTP Framework treats query params as a part of the URL. Thus, if an unexpected query param appears in the URL it will consider it to be an unavailable resource. If one or more query params need to be supported then these should be defined in the *Resource Manager* for the specific resource:



On the other hand, if unspecified query params need to be supported or if you never want query params to be the source of a 404 error, then you can configure the resource with a wildcard query param:

