

Error Handling

If an error occurs during the execution of an API 1 call, the error is returned to the client with a particular HTTP status code and a message. The message is in the form of a JSON Object — unless the error occurs before the call begins the execution of API functionality. For example, this would occur when a consumer calls a non-existing web service or path. In this case, most errors would have an HTTP 4xx status code.

When an error occurs once API functionality commences, the system will:

- Record the error in the api.log file on the server. (Logger level is set to at least ERROR.)
- Roll-back changes that occurred during call execution.
- Respond with a code other than code 200 (OK). The response code gives a generic description of the problem, with the response message providing a more detailed description of the cause of the error. The error message is a JSON Object that may contain up to two keys:
 - status : A string that represents the HTTP status code; and
 - message: A user-friendly string that contains the description of the error.

An example of an error object is shown below:

```
{
  "status": "Not Found",
  "message": "Work Order with ID#1273 does not exists"
}
```



Note: Response code 200 (OK) implies that there were no problems with the API call and so the response should be interpreted as valid output for an API call. Any other response code indicates an error and should be handled by the client as invalid output.

Examples of calls that generate errors are shown below:

- `curl -i -X DELETE -d '{"filter":[{"WORK_ORDER_ID","=",1622}, {"WORK_ORDER_ID","=",1621}], "OR"}' http://localhost:8080/ams_dev/rest/v1/workOrder`

Error 415 – unsupported media type. (The call expects a JSON Object and the request does not specify a request type.)



Note: The error message in this example is not a JSON Object because the error occurred before API functionality commenced.

- `curl -i -X DELETE -H "Accept:application/json" -H "Content-Type: application/json" -d '{"filter":[{"WORK_ORDER_ID","=",1622}, {"WORK_ORDER_ID","=",1621}], "OR"}' http://localhost:8080/ams_dev/rest/v1/workOrder/`

Error 400 – Bad Request. (The JSON Object is incorrectly formatted.)



Note: The error message in this example is not a JSON Object because the error occurred before API functionality commenced.

- `curl -i -X PUT -H "Content-Type: application/json" http://localhost:8080/ams_dev/rest/v1/workOrder/1643/DayCard/ -d '{"WORK_ORDERS_LABOR_DC":[{"LABOR_ID":"TRC_ID":1,"TOTAL_HOURS":0.1,"WAC_CODE_ID":null}]}'`

Error 400 – Bad Request. (This call has the correct structure, but it returns an error because not all primary key columns are in the request.)



Note: The error message in this example is not a JSON Object because the error occurred before API functionality commenced.

- `curl -i -X PUT -H "Content-Type: application/json" http://localhost:8080/ams_dev/rest/v1/workOrder/1643/DayCard/ -d '{"WORK_ORDERS_LABOR_DC":[{"LABOR_ID":2412555,"DATE_WORK":"20130314","TRC_ID":1,"TOTAL_HOURS":0.0,"WAC_CODE_ID":null}]}'`

Error 400 – Bad Request. (This call has the correct structure, but it returns an error because zero hours are not allowed.)