Handling Errors

If a call to an POS API endpoint fails, error information is returned in the API JSON response.As there are some situations where an API client can respond to an error – e.g. by gathering and supplying additional parameters, or correcting invalid input – the API defines formal error codes that can be examined by the client. These error codes are supported as part of the formal API, so API clients can rely upon them across POS versions.

The JSON error response looks like the following:

```json
{
  "errors": [
    {
      "message": "Custom error message...",
      "locations": [
        {
          "line": 2,
          "column": 3
        }
      ],
      "path": [
        "giftcardAdd"
      ],
      "extensions": {
        "code": "VALIDATION",
        "data": {
          "ValidationType": "NotEmptyValidator",
          "Property": "Member"
        }
      }
    }
  ]
}
```

The error code is supplied in the `errors/extensions/code` value. Other values may be helpful while developing an API client, but typically should not be used:

- `message` indicates more information about the error; however this is rarely suitable for display to an end-user.
- `locations` indicates where in the GraphQL request that led to the error.
- `path` indicates the endpoint path.

Additional error details may be provided for some error code types in the `extensions/data` property.

Formal Error Codes

The POS API supports a list of common, formal error codes that your API client may choose to handle. This list is formally supported and is part of the public POS API.

Common Errors

There are several errors that may be returned by any POS feature:

- `FUSION_ERROR`
  - A generic error occurred, usually representing an unexpected condition or state.
  - The caller is not able to do much with this type of error, except log and/or report the error to the end user.
- `NOT_AUTHENTICATED`
  - The endpoint requires authentication, and the caller is not currently authenticated.
  - Ensure the caller is authenticated and retry the request.
- `NOT_AUTHORIZED`
  - While the caller is authenticated, the endpoint requires permission(s) that the caller does not possess.
  - Consider granting the required permission to the caller.
- `INVALID_CONFIG`
  - The endpoint operation cannot proceed as POS is not configured correctly.
- `FORMAT`
  - One or more input parameters do not have the correct data format.
  - Examples include:
    - Passing an incorrectly formatted DateTime value, including a blank string.
    - Passing an incorrectly formatted number, including a blank string.
- `VALIDATION`
  - One or more input parameters to the endpoint failed validation.
  - Additional error information is available in the error response:
    - `extensions/data/ValidationType` indicates the type of validation that failed.
    - `extensions/data/Property` indicates the name of the input property that failed validation.
- `INVALID_SINCE_FORMAT`
  - A `since` parameter was passed to the API endpoint, but is not correctly formatted for that endpoint.
  - Never generate your own `since` values; only use the values that are return in the `until` result of POS endpoints.

- `SERVICE_UNAVAILABLE`
  - The request failed because a system component is currently not available.
  - The request can be retried later.

Gift Card Errors

The Gift Card module can return several types of errors:

- `GC_CODE_AMBIGUOUS`
  - A gift card could not be identified by the supplied code, as the code matches multiple gift cards.
  - The caller must also supply the code type.
- `GC_CODE_EXISTS`
  - A supplied gift card code in an add or edit operation is already used by another gift card.
- `GC_CODE_EXPIRED`
  - The supplied card code has expired, and can no longer be used.
- `GC_CODE_NOT_FOUND`
  - The supplied code does not match an active gift card.
- `GC_ID_NOT_FOUND`
  - The supplied internal ID does not match an active gift card.
- `GC_INVALID_PIN`
  - The supplied gift card PIN is invalid, or a PIN is required but has not been supplied.
- `GC_INSUFFICIENT_FUNDS`
  - A gift card does not have sufficient funds to perform the specified transaction.
- `GC_MIN_REDEEM_AMOUNT`
  - An attempt was made to redeem points for a gift card, but the requested dollar amount was less than the minimum amount configured in POS settings.
- `GC_TRANS_CANCELLED`
  - An attempt to capture or cancel a gift card transaction failed because the transaction was already cancelled.
- `GC_TRANS_COMPLETED`
  - An attempt to capture or cancel a gift card transaction failed because the transaction was already completed.
- `GC_TRANS_NOT_FOUND`
  - The supplied internal ID did not match a gift card transaction.

Other Supported Errors

- `AC_ERROR`
  - A general error occurred with the Acumatica integration.

- AW_CONFIG_ERROR
  - AWS integration is not configured correctly.
- BP_ERROR
  - An operation involving the BridgePay payment processor failed.
- CU_CUSTOMER_NOT_FOUND
  - A supplied ID or code does not match a customer record.
- LC_STORE_NOT_FOUND
  - A supplied ID or code does not match a store record.
- SY_INVALID_INPUT
  - Data imported by the sync process is not properly formatted.

Non-Supported Errors

The API may also return other errors that are not in the formal list above. These types of errors typically correspond to incorrect program code, operation or misconfiguration.

API clients should not attempt to parse unsupported error codes – or error messages – as these are subject to change with new versions of POS.

> Handling non-supported error codes or messages may break API clients as new versions of POS are released. Only base error handling on the supported list of error codes defined above.

API clients **are** encouraged to log these errors and display friendly error messages to end users, if applicable.