GLOBAL ERROR HANDLER

(BY CHINNA)

CHAPTERS	

A new hero is born, able to leap clouds in a single bound.

TABLE OF CONTENTS

The global-error-handler

- The global-error-handler is a utility error handler flow across projects.
- Each projects foundation needs global-error-handler.

Advantages:

- One-time development.
- Common error handling across the projects.
- Developer friendly and decreases a lot of efforts in modern API's through API LED Connectivity API layers.

How to build:

- The global error handler can handle all common errors across the projects.
- All the possible error should be identified over REST API's and should be handled through global error handler.
- The possible errors covered with this global error handler are listed below.

APIKIT: BAD_REQUEST: This error type usually generated by API based project but moved the same to global error handler to maintain it as unique.

This handles bad request over request i.e. it executes when invalid request payload sent over http request. API Kit Router can able validate the request against to RAML.

```json
{
  "code": 400,
  "message": "Bad request",
  "description": [
    "required key [empID] not found Location:    Position: Line 0,
Column 0",
    "required key [empName] not found Location:    Position: Line 0,
Column 0"
  ],
  "dateTime": "2020-05-29T05:58:02",
  "transactionId": "44b32520-61ee-47b4-907d-fa15869f3c4d"
}
```

APIKIT: METHOD_NOT_ALLOWED: This error type usually generated by API based project but moved the same to global error handler to maintain it as unique.

It executes when invalid method has sent over http request. API Kit Router can able validate the request against to generated flow method.

```json
{
  "code": 405,
  "message": "Method Not Allowed",
  "description": "The method specified in the request is not allowed
for this resource",
  "dateTime": "2020-05-29T05:58:02",
  "transactionId": "44b32520-61ee-47b4-907d-fa15869f3c4d"
}
```

APIKIT: NOT_ACCEPTABLE: This block handles when The resource identified by the request is not capable of generating response entities according to the request accept headers

```
{
  "code": 406,
  "message": " Not Acceptable",
  "description": "The resource identified by the request is not
capable of generating response entities according to the request
accept headers",
  "dateTime": "2020-05-29T05:58:02",
  "transactionId": "64b92520-61he-47b4-907d-fa15869f3c4d"
}
```

APIKIT: NOT_FOUND: This block will execute if the server not found anything that matching to the request URI.

```
{
  "code": 404,
  "message": " Resource not found",
  "description": "The server has not found anything matching the
Request-URI",
  "dateTime": "2020-05-29T05:58:02",
  "transactionId": "44b32450-61ee-47b4-907d-fa12a69f3c4d"
}
```

APIKIT: UNSUPPORTED_MEDIA_TYPE: This block will execute when the given request is not supported in any of predefined format and hence the server will refuse the request.

```
{
  "code": 415,
  "message": "Unsupported media type",
  "description": "The server is refusing to service the request
because the entity of the request is in a format not supported by
the requested resource for the requested method",
  "dateTime": "2020-05-29T05:58:02",
  "transactionId": "44b32520-61ee-47b4-907d-fa15869f3c4d"
}
```

HTTP: BAD_REQUEST: This block will execute when server will not process the request due to either malformed requests or invalid request messaging formats.

```json
{
  "code": 400,
  "message": " Bad Request",
  "description": "The 400 (Bad Request) status code indicates that
the server cannot or will not process the request due to something
that is perceived to be a client error (e.g., malformed request
syntax, invalid request message framing, or deceptive request
routing)",
  "dateTime": "2020-05-29T05:58:02",
  "transactionId": "44b32520-61ee-47b4-907d-fa15869f3c4d"
}
```

HTTP: FORBIDDEN: This block will execute when server refuses the request because of authorization issues. There might be multiple reasons why server is refuses the request will be shown as part of response payload.

```json
{
  "code": 403,
  "message": " Access to the upstream service is forbidden.",
  "description": "The 403 (Forbidden) status code indicates that the
server understood the request but refuses to authorize it.  A server
that wishes to make public why the request has been forbidden can
describe that reason in the response payload (if any)",
  "dateTime": "2020-05-29T05:58:02",
  "transactionId": "44b32520-61ee-47b4-907d-fa15869f3c4d"
}
```

HTTP: CLIENT_SECURITY: This block indicates an external entity (e.g., calling an external endpoint) produced a security error.

```json
{
  "code": 401,
  "message": "Security exception occurred",
  "description": "This error indicates an external entity
/application produced a security error.",
  "dateTime": "2020-05-29T05:58:02",
  "transactionId": "44b32520-61de-47b4-907d-fa15869f3c4d"
}
```

HTTP: CONNETIVITY: This block executes when upstream is not available to handle the requests.

```
{
  "code": 503,
  "message": " Service unavailable",
  "description": "The (upstream) service is temporarily not
available",
  "dateTime": "2020-05-29T05:58:02",
  "transactionId": "44b32520-61ee-47b4-907d-fa15869f3c4d"
}
```

HTTP: INTERNAL_SERVER_ERROR: This block will execute when any un expected condition that occurred during the flow to full the request.

```
{
  "code": 500,
  "message": "Upstream service unable to fulfil request.",
  "description": "The 500 (Internal Server Error) status code
indicates that the server encountered an unexpected condition that
prevented it from fulfilling the request",
  "dateTime": "2020-05-29T05:58:02",
  "transactionId": "44b65520-61ee-47b4-907d-fa06469f3c4d"
}
```

HTTP: METHOD_NOT_ALLOWED: This block will execute when method specified in the given request is not allowed.

```
{
  "code": 405,
  "message": "The method specified in the request is not allowed for
this resource",
  "description": "The method identified in the request is not
allowed for the resource given in the request.
  "dateTime": "2020-05-29T05:58:02",
  "transactionId": "22ab0987-61ee-47b4-907d-bn15869f3c4d"
}
```

HTTP: NOT_ACCEPTABLE: This block handles when client error response code indicates that the server cannot produce a response matching the list of acceptable values defined in the request's proactive content negotiation headers, and that the server is unwilling to supply a default representation.

```json
{
  "code": 406,
  "message": " Not Acceptable",
  "description": "The resource identified by the request is not capable of generating response entities according to the request accept headers",
  "dateTime": "2020-05-29T05:58:02",
  "transactionId": "62p09876-61ee-47b4-907d-fa15869f3c4d"
}
```

HTTP: NOT_FOUND: This block will execute when sever unable to find the matching request URI.

```json
{
  "code": 404,
  "message": " The server has not found anything matching the Request-URI ",
  "description": "The resource identified by the request is not find the matching request.",
  "dateTime": "2020-05-29T05:58:02",
  "transactionId": "98b32520-61ee-34n5-907d-fa15869f3c4d"
}
```

HTTP: PARSING: This block will execute when an HTTP request parser. It takes chunks of text as received and returns a 'hint' as to what is required, or returns the http: Request when a complete request has been read.

```json
{
  "code": 400,
  "message": "To read chunks of text message",
  "description": "The http request parser will read request in chunks to read the complete request.",
  "dateTime": "2020-05-29T05:58:02",
  "transactionId": "78f47431-61ee-47b4-907d-fa15869f3c4d"
}
```

HTTP: RETRY_EXHAUSTED: This block handles when The request is unable to interact with external application and exhausted the given retries to connect to external application.

```
{
  "code": 503,
  "message": "Retry exhausted for the given request",
  "description": "The request for the external application unable to
connect and exhausted the given retry count.",
  "dateTime": "2020-05-29T05:58:02",
  "transactionId": "74n32920-98sd-47b4-907d-fa15869f3c4d"
}
```

HTTP: SECURITY: This block indicates a security error occurred, like invalid credentials being received or an expired token being used.

```
{
  "code": 401,
  "message": "Method Not Allowed",
  "description": "The resource identified by the request is not
capable of generating response entities according to the request
accept headers",
  "dateTime": "2020-05-31T05:45:02",
  "transactionId": "44b32520-61ee-47b4-907d-fa15869f3c4d"
}
```

HTTP: TIMEOUT: This block will execute when the given request for external application is not responded in given interval of time timeout exceptions will occurred.

```
{
  "code": 504,
  "message": "Timeout exceeded",
  "description": "The request for the external application is
returned the response in given span of time.",
  "dateTime": "2020-05-31T05:48:02",
  "transactionId": "44b32520-61ee-47b4-907d-fa15869f3c4d"
}
```

HTTP: TOO_MANY_REQUESTS: This block will execute when request throws 429 response status code indicates the user has sent too many requests in a given amount of time.

```
{
  "code": 429,
  "message": "Too Many Request for given amount of time",
  "description": "The 429 response code indicates that too many requests are trying to process in given span of time",
  "dateTime": "2020-05-31T05:58:02",
  "transactionId": "42a23520-98sd-47b4-907d-fa15869f3c4d"
}
```

HTTP: UNAUTHORIZED: This block will execute that the request has not been applied because it lacks valid authentication credentials for the target resource.

```
{
  "code": 401,
  "message": "Unauthorized request",
  "description": "The given request has lack of valid authentication details",
  "dateTime": "2020-05-31T06:00:02",
  "transactionId": "54b56520-61ee-47b4-907d-fa15869f3c4d"
}
```

HTTP: UNSUPPORTED_MEDIA_TYPE: This block will execute when the server refuses to accept the request because the payload format is in an unsupported format.

```
{
  "code": 415,
  "message": "Unsupported Media Type",
  "description": "The server refuses to execute accept the given request due to the payload format is unsupported.",
  "dateTime": "2020-05-31T06:05:02",
  "transactionId": "98ba7864-61ee-47b4-907d-fa15869f3c4d"
}
```

STREAM_MAXIMUM_SIZE_EXCEEDED: This block will execute when the buffer uses the more size than configured.

```
{
  "code": 500,
  "message": "Buffer Size exceeded the maximum",
  "description": "This will return when the buffer size is
exceeded the maximum size that given",
  "dateTime": "2020-05-31T06:08:02",
  "transactionId": "42d37820-61ee-47b4-907d-fa15869f3c4d"
}
```

RETRY_EXHAUSTED: This block will indicate that retries of a certain execution block have been exhausted. For example, for a given operation, or using Until Successful Scope.

```
{
  "code": 503,
  "message": "Retry Exhausted for given request",
  "description": "The resource identified by the request is not
capable of generating response entities according to the request
accept headers",
  "dateTime": "2020-05-31T06:12:02",
  "transactionId": "21h70954-61ee-47b4-907d-fa15869f3c4d"
}
```

REDELIVERY_EXHAUSTED: This block indicates that max attempts to reprocess a message from a source have been exhausted.

```
{
  "code": 503,
  "message": "Redelivery Exhausted for the given request",
  "description": "The resource identified by the request is not
capable of generating response entities according to the request
accept headers",
  "dateTime": "2020-05-31T06:15:02",
  "transactionId": "49b39520-61ee-47b4-907d-fa15869f3c4d"
}
```

CONNECTIVITY: This block indicates that there was a problem establishing a connection. This could occur while using a connector, for example, an HTTP requester

```json
{
  "code": 503,
  "message": " Service unavailable ",
  "description": "The (upstream) service is temporarily not available",
  "dateTime": "2020-05-31T06:18:02",
  "transactionId": "48n32920-69ne-47b4-907d-fa15869f3c4d"
}
```

TIMEOUT: This block indicates timeout occurred while processing a message.

```json
{
  "code": 504,
  "message": "Timeout occurred to process the request",
  "description": "This indicates that the timeout occurred while processing the request",
  "dateTime": "2020-05-31T06:45:02",
  "transactionId": "49d32520-61ee-47b4-907d-fa15869f3c4d"
}
```

TRANSFORMATION: This block indicates an error occurred while transforming a value. This involves Mule Runtime internal transformations and not Data Weave transformations

```json
{
  "code": 400,
  "message": "Transformation exceptions occurred",
  "description": "The exception occurred while transforming a value",
  "dateTime": "2020-05-31T06:55:02",
  "transactionId": "87g32520-61ee-47b4-907d-fa15869f3c4d"
}
```

EXPRESSION: This block indicates an error occurred while evaluating a Data Weave expression.

```
{
  "code": 400,
  "message": "Error while parsing expression",
  "description": "This error will have occurred while evaluating
the expression in Data weave,
  "dateTime": "2020-05-31T06:58:02",
  "transactionId": "42b32520-61ps-47b4-907d-fa15869f3c4d"
}
```

ROUTING: This block indicates an error occurred while routing a message.

```
{
  "code": 500,
  "message": "Routing exception occurred",
  "description": "This indicates an error occurred while routing a
message. For example, using the Round Robin router",
  "dateTime": "2020-05-29T07:02:02",
  "transactionId": "68f32520-61ee-49b4-907d-fa15869f3c4d"
}
```

SECURITY: This block indicates a security error occurred, like invalid credentials being received or an expired token being used.

```
{
  "code": 401,
  "message": "Security exception occurred",
  "description": "This indicates a security error occurred, wheen
invalid credentials being received or an expired token being used.,
  "dateTime": "2020-05-29T07:15:02",
  "transactionId": "49q32520-98fg-47b4-907d-fa15869f3c4d"
}
```

ANY: This block executes if any other error occurs out of handled above.

```
{
  "code": 500,
  "message": " Internal server error ",
  "description": "The server encountered an unexpected condition
which prevented it from fulfilling the request ",
  "dateTime": "2020-05-29T05:58:02",
  "transactionId": "44b32520-61ee-47b4-907d-fa15869f3c4d"
}
```