**Mule 4 – Dataweave Expressions**

|  |  |
| --- | --- |
| **Description** | **Expression** |
| Flow name in an API | #[[flow.name](http://flow.name)] |
| Client/Caller to an API | #[authentication.properties.clientName] |
| Correlation ID | #[correlationId] |
| Loading a file | ${file::properties-file.txt} |
| Extracting the underlying java exception class | #[if (error.exception.cause.^class == null) error.cause.^class else error.exception.cause.^class] |
| Extracting the underlying exception message | #[if (error.exception.cause.message == null) error.cause.message else error.exception.cause.message] |
| Extracting error type and description | #[error.errorType.namespace], #[error.errorType.identifier], #[error.description] |
| Extracting the error payload | #[error.errorMessage.payload] |
| Accessing Global Functions maintained in a Dataweave Script | #[global::functions::urlEncode("sample value")] |
| Accessing fields in Web Service Response | payload.body |
| Status Code pertaining to a HTTP Response received | #[message.attributes.statusCode] |
| Soap Fault Code | #[error.exception.faultCode.localPart] |
| Soap Fault String | #[error.exception.message] |
| Coercing raw data to string | #[payload.^raw as String] |
| Removing multiple elements from an object | #[payload - <key1> - <key2>]  **In Transformer:**  %dw 2.0 output application/json --- {aa: "a", bb: "b", cc: "c"} - "bb" - "cc" |
| Removing a record from a list | #[(message.payload - message.payload[index]) as Array] |
| Forming dynamic keys | %dw 2.0  output application/json  ---  {       (vars.input1 ++ vars.input2): {                                 ........       }  }  Use the expression within () to be evaluated to a dynamic key |
| Dynamic SQL Queries | truncate table ${[table.name](http://table.name)} drop storage |
| Dynamically loading DWL files | %dw 2.0 output application/java var load = readUrl("classpath://somepath/" ++ vars.fileName ++ ".dwl","application/dw") --- load(payload)  **DWL File:**  %dw 2.0 output <output-type> --- (payload) ->  payload map {} |
| Loading SQL script from SQL Files in Database Connector | Place below line in SQL Query Text in the database connector  #[output text/plain --- readUrl("classpath://somepath/ ++ vars.filename ++ ".sql" ,"text/plain")] |
| Get/Remove Extension from filename | %dw 2.0 import java!org::apache::commons::io::FilenameUtils  // Custom functions to handle file extensions fun removeExtension(filename: String) = (FilenameUtils::removeExtension(filename)) fun getExtension(filename:String) = (FilenameUtils::getExtension(filename))  Note: Apache Commons Dependency needs to be added. |
| Munit assertion for json,csv,xml | Ex: #[MunitTools::equalTo(readUrl('classpath://somepath/abc.json', 'application/json'))] |