Dataweave 2.0 CheatSheet

DataWeave is the MuleSoft expression language for accessing and transforming data that travels through a Mule app

General

- Dataweave scripts can be externalized into DWL files.
- Dataweave is case sensitive.
- A script has 3 parts, the header that contains directives, variables and functions; the body that contains the actual transformation or expression; and the delimiter.
- Supported mime-types list here: https://docs.mulesoft.com/mule-runtime/4.3/dataweave-formats

Operators and Selectors

- @ is the operator for reading and creating attributes
- . is the single value selector
- .* is the multi-value selector, includes repeated or multiple elements.
- {( )}, the constructor-evaluation operator (composed of curly braces and parentheses) allows to extract values from an array and wrap them into an object, i.e. converts an array of objects into an object of objects. Useful when transforming arrays into XML.

Functions and Variables

- *map*, applies a transformation into each element of an array. **Only works with arrays.**
- $ is the operator to refer to the current element of the iteration.
- $$ refers to the index of the current element of the iteration.
- *var* is the directive to declare variables.
- *fun* is the directive to declare functions.
- *using* is the function to declare local variables.
- *contains* works for arrays and strings to check the presence of an element on them.
- *sizeOf* works for arrays, objects, strings.
- *orderBy, filter, distinctBy, groupBy* works for arrays only, the order on how these functions are applied also matters.
- *lookup* allows invoking flows (does not allow sublfows). It has two parameters, 1st the flow's name, 2nd the payload for the flow.

Formats

- Can format Numbers and Dates
- Use - *as* operator - for data coercion (e.g. payload.price as Number)
- Format a number or date using a metadata schema, e.g. *price as Number {format: "###.00"}, date as Date {format: "dd/mm/yyyy"}*

Types and Libraries

- Create new data types with *type* directive, e.g. *type User = Object {class: "my.company.User"}*
- Import additional dataweave modules and libraries with *import* directive, e.g. *import * from dw::core::Strings*

References

- https://docs.mulesoft.com/mule-runtime/4.3/dataweave-quickstart
- https://docs.mulesoft.com/mule-runtime/4.3/dataweave-cookbook
- https://developer.mulesoft.com/tutorials-and-howtos (search Dataweave section)
- https://docs.mulesoft.com/mule-runtime/4.3/dw-functions