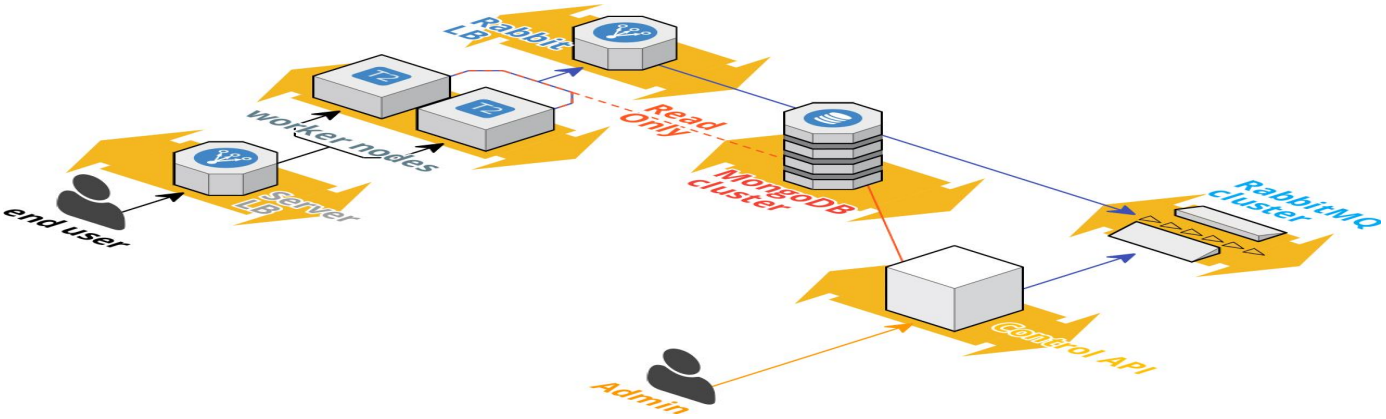
FLOW ORCHESTRATOR

User Guide - Power User



INTRODUCTION

Purpose

This document provides an overview and the required details to understand the semantics of constructing a new flow of execution based on the available/deployed functional components.

Intended Audience

The intended audience is “Power User” or any user who would construct / configure a “flow of execution”.

Definitions, Acronyms & Abbreviations

PU: Power User

FO: Flow Orchestrator

FOE: Flow of execution

Flow of execution (FOE) - Construction

Component Understanding

Please check the deployed components list and its metadata to understand the components that you could use in your flow construction and also to understand the input parameters expected by each component and the response/output fields the same component would produce.

For Sample, please refer: “deployed-components.csv”

Construction Format

You will need to use a csv template to construct/configure the flow of execution, this csv file would be provided as an input to the Flow Orchestrator. Ideally, a graph designer tool could be used to depict the flow diagram in a graphical manner, however for this sample application a simple csv template is used for the same.

Please use the provided csv template to construct the flow of execution as required using the deployed/available components.

CSV template: “flow-config-template.csv”

For Sample, please refer: “flow-config-create-employee.csv” OR “flow-config-retrieve-employee,csv”

Steps & Details

Following are the definition of the columns, know-how, supported config, rules/constraints of the csv template, please open the csv template / sample and refer the same while going through the below items:

1. “SNo” – This column denotes a serial number to denote the number of rows in the csv. Please provide a sequential number for this column.
2. “FlowName” – This column denotes a custom logical name for each flow/step in the flow of execution. This name can be used for tracing purposes. Please provide a logical name for each flow/step in your flow of execution.
3. "CallType" – This column denotes the mode/type of the call to the component, basically this indicated if this flow/step should be a synchronous call OR an Asynchronous call like “Fire-n-Forget” type. Only 2 values are supported which are “SYNC” or “Async”, the values can be either in upper case, lower case or even mixed case that does not matter.
4. "ComponentID" – This is a very critical column and denotes the ID of the component that you would like to invoke in a particular flow/step. This “ComponentID” should exactly match the “ComponentID” as per the deployed/available components. Please refer “deployed-components.csv” file to get the “ComponentID” of the deployed component that you intend to use.
5. Constructing a synchronous call on a component that is designed to be ASYNC by design will throw an error and stop the flow of execution. E.g. Message/PubSub component.
7. All the response fields (from Synchronous calls only) will be accumulated/aggregated and will be sent back to the caller as the overall response from this flow of execution.
8. Note: use backslash if you would like to use “,” comma character within any value in the transformation logic to ensure that the csv file is not corrupted because of this value.

Data Transformation

“DataTransformationRules”: This column is optional and provides the data transformation rules to perform any data processing before invoking the component/flow/step. The format in which the transformation rule has be provided is. Please refer to the sample provided in the “flow-config-create-employee”, “Flow Step 2” while going through the below.

a. “targetFieldName@valueType=dataType=value#operator#valueType=dataType=value|operator#valueType=datatype=value$.....n”.

b. sample: to denote “displayName=firstname + “, “ + lastname, PassportValidty = expiryDate “ the relevant transformation rule is: “displayName@field=string=firstname#concat#constant=string=, |concat#field=string=lastname$PassportValidity@field=string=expiryDate ”

c. “targetFieldName” – denotes the field name to which the transformed data needs to be assigned to and sent as an input to the component in this flow/step.

d. @ (at symbol) - targetFieldName level separator

e. "valueType" - denotes the type of value that is being provided. Supported values are “constant” or “field”

f. if "valueType" is "constant", then value can be a static constant value

g. if "valueType" is "field", then value can be a "field/parameter" from the responses of all previous components in the flow including the input from the original caller like “firstname”.

h. = (equal to symbol) - element level separator

i. "dataType" – denotes the data type of the value being provided. Supported values are "string" or "number" only

j. “value” – denotes the value.

k. # (hash symbol) - field/value level separator

l. "operator" – denotes the operation to be performed in the transformation. Supported values are "concat", "plus", "minus", "multiply", "divide" only

m. same as e, f, g points for valueType, datatype, value

n. | (pipe symbol) - expression level separator

o. repeat steps k (operator) and i (value) for how many ever value that needs to be concatenated with the value to arrive at the final value for targetFieldName.

p. $ (dollar symbol) - transformation rule separator to aid many rules/logics to be added for many target fields for each flow.