JMP web service design notes (Ver 2.2.0)

Date: 10.01.2021

|  |  |  |
| --- | --- | --- |
| Ver. | changes | date |
| 2.0.0 | Changes in all aspects – process flow and functionality | 10.01.2021 |
| 2.2.0 | Changes marked in green  REQ\_ID – cancel validation | 28.02.2021 |

# Objectives

* Enable using pre-defined JMP modules (in Python code), for calculating estimated cost or partial cost , while running SAP quotation (VA22) or other
* Process will support changes in the modules calculation , and using different modules - based on pre-determined attributes

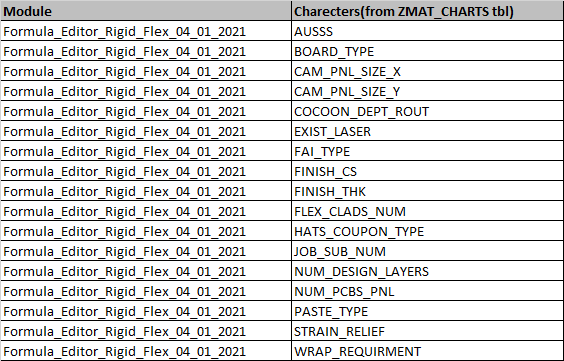
# Process steps

**Step 1** – temporary quick solution \*

Create dedicated Tcode with the following functionality

* User will select a Module name from dedicated Z\_table
* Relevant material characteristics will be available according to the Z\_table mapping

Example: (file attached)





* The program will copy all relevant material characteristics
* User will set values from each character, from a close list of values (inc. in ZMAT\_CHARTS), strings only, integers/floats can be manually override.
* User will post a request to the WS with the selected Parameters + req\_id + model name (see example in paragraph 3)
* Note – it is mandatory to fill all the associated characters with values , in order to post the request

**Step 2** – the event will call a web service in the follwing path :

* + <http://websrvdev-01.pcb-il.co.il:5001/jmp/> (DEV & TESTING)
  + <http://websrvprd-01.pcb-il.co.il:5001/jmp/> (PRD\*)

sending relevant parameters, in a JSON format.

**Step 4** – A local REST web service, will handle the JMP’s formula files,

* + receiving the SAP “POST” with JSON data
  + validating data
  + Assigning it to the relevant module & function
  + “POST” back the returned result in a JSON format

**Step 5** – temporary quick solution

* The program will collect all the parameters and values that were send + the WS answer values, and introduce it to the user screen

# 3. The WS functionality & rules:

1. The WS expect t get a request (POST)in JSON Array format, in the message body :

{

 "REQ\_ID":"I3333333",

 "SAP\_ITEM\_CHARECTERISTIC\_1" : Value(str,Float),

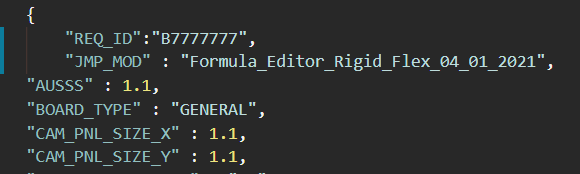
 "SAP\_ITEM\_CHARECTERISTIC\_2" : Value(str,Float),

 "SAP\_ITEM\_CHARECTERISTIC\_3" : Value(str,Float)

  …more ,

}

The messege will include all the valuess relevant for the chosen module (paragraph 2), attached example for the JSON structore





1. ~~REQ\_ID – messege will include is identifier for the request and it is unique fo~~*~~r qoute\_no &qoute\_line\_no~~*

~~REQ\_ID format include digits, the last digit is a key validation- includes the average of all the digits - rounded down. no length limit.~~

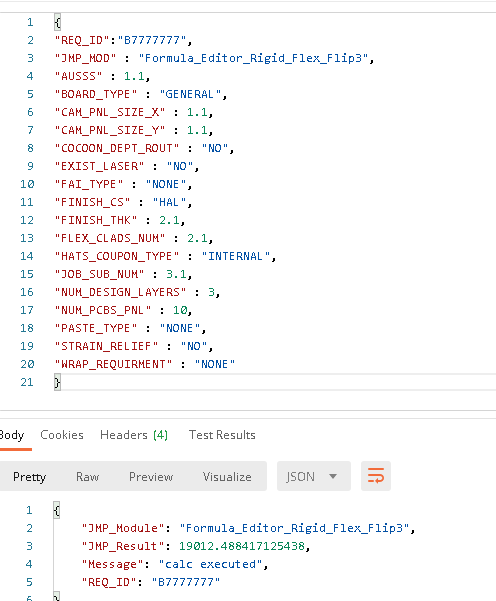
~~In case the REQ\_ID, from some reason, does include non-digit characters, the base format still applies (the program will consider only the digits in the string for the calculation).~~

Not Valid in this ver. ,there is no need anymore , as now the program validate the requested module name

1. The WS will validate that the ‘module name’ exist in its internal modules pull (python modules files that maintained in a dedicated folder).

If exits the WS will valdate all characteristic exist in the post request

1. WS will run the relevant function in the selected module and return results in a JSON format , see example (param are as the one in the JSON file example, for quick simulation) :



Loading new JMP modules:

1. The new JMP files (Python format) need to be named with a “JMP” prefix and a “.py” extension (example: JMP\_RIGID\_FLEX.py)
2. The files need to be set at the “pkg” directory in the app root (example: “\\JMPapp\pkg\” )