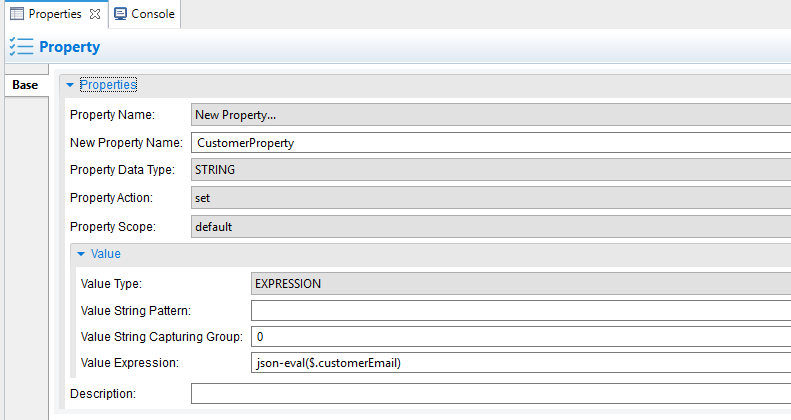
**WSO2 POC**

1. Installation component: -
   1. Install Oracle Java SE Development Kit (JDK) version 1.8. and set the JAVA\_HOME environment variable.
   2. Do not install higher versions of jdk as it does not support higher version.
   3. Go to the [**product page**](https://wso2.com/integration/) **(**<https://wso2.com/integration/>**)**  of WSO2 Enterprise Integrator, select Other Installation Options, and download the Binary distribution. Extract the ZIP file of the binary. This will be your <EI\_HOME> directory.
   4. Select and download the relevant WSO2 Integration Studio ZIP file based on your operating system from [here](https://wso2.com/integration/tooling/) and then extract the ZIP file.  
      The path to this folder is referred to as <EI\_TOOLING> throughout this tutorial.
   5. Getting an error message? See the troubleshooting tips given under [Installing WSO2 Integration Studio](https://docs.wso2.com/display/EI650/Installing+WSO2+Integration+Studio#InstallingWSO2IntegrationStudio-Troubleshooting). <https://docs.wso2.com/display/EI650/Installing+WSO2+Integration+Studio#InstallingWSO2IntegrationStudio-Troubleshooting>
   6. Download the postgresql jar and put it in lib folder of installed wso2.
   7. Install postgresql if you are connecting to it.
2. Tutorial:-
   1. Create an independent microservice and keep it up and running.
   2. Open the wso integration studio and create the project with all three service, application and registry checked.
   3. Drag the property mediator and configure the property for example

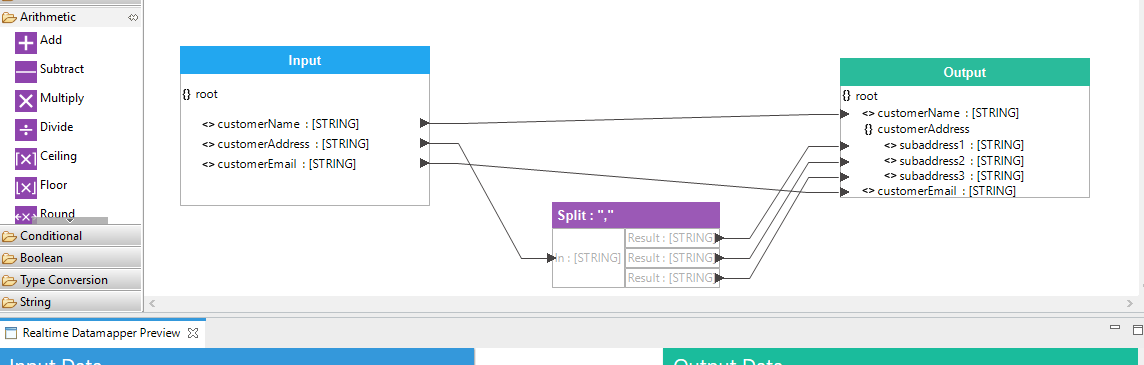


Where the value expression is taking the value from input payload with customerEmail tag.

* 1. We are going to use this property’s value in next flow
  2. Drag a data mapper next to the property mediator and configure it.
  3. To configure it follow the

<https://docs.wso2.com/display/EI650/Transforming+Message+Content>

Double click on data mapper configure accordingly for example : -



* 1. Add a switch mediator next to this data mapper.
  2. Configure the switch mediator using

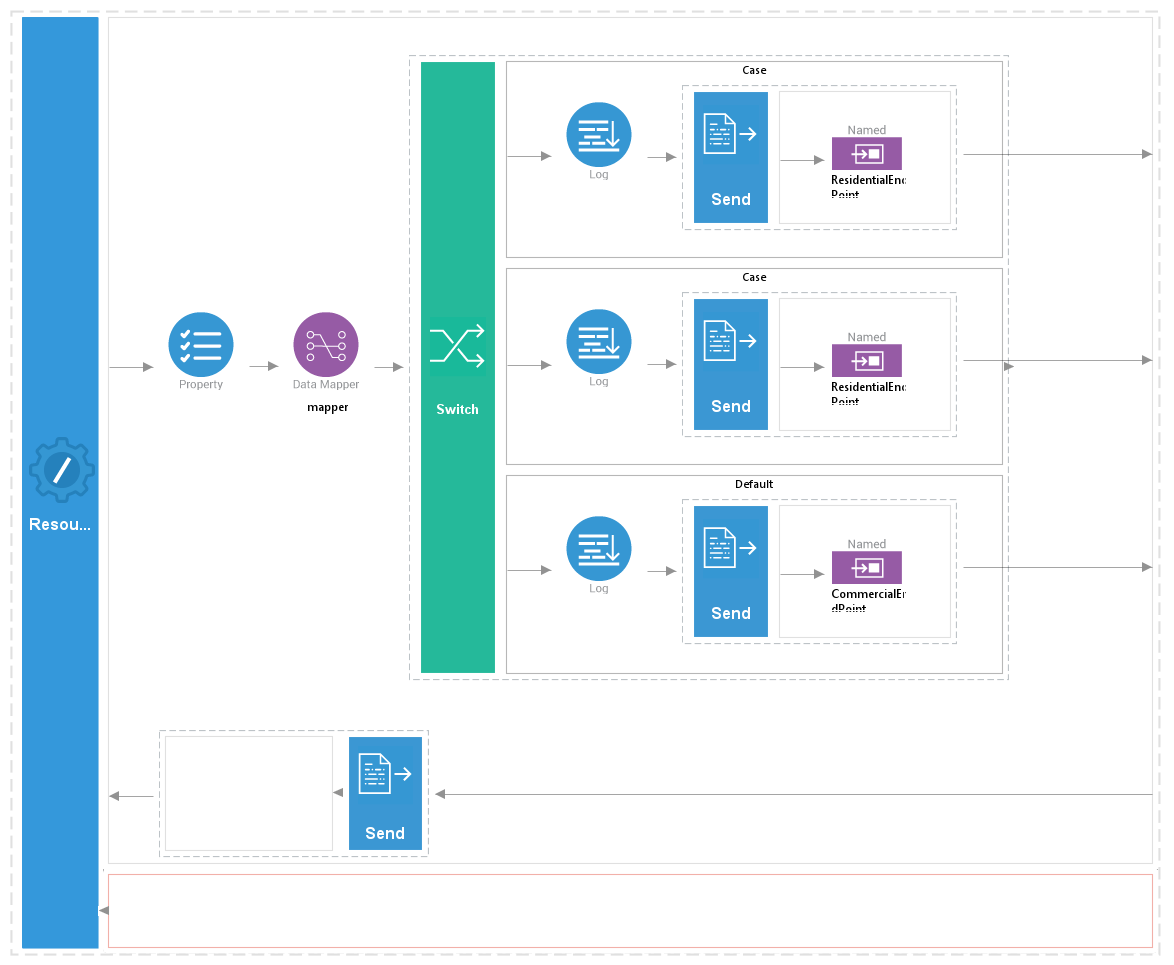
<https://docs.wso2.com/display/EI650/Routing+Requests+Based+on+Message+Content>

* 1. See below image for example :-



* 1. Add a log palette to each case and configure the property using [this](https://docs.wso2.com/display/EI650/Routing+Requests+Based+on+Message+Content).
  2. Add a send mediator palette to each log added above in h
  3. Now point the defined end point in this send so that the flow will go to these HTTP end point which will point to the independent services we wrote earlier.

See below image for reference-



* 1. To give the response to the client drag a send mediator palette in outsequnce.

Useful links :-

<https://docs.wso2.com/display/EI650/Sending+a+Simple+Message+to+a+Service>

<https://docs.wso2.com/display/EI650/Transforming+Message+Content>

<https://docs.wso2.com/display/EI650/Routing+Requests+Based+on+Message+Content>

Git URL : -

<https://github.com/MohitPatidar483/WSO2_CustomerRepository>

Points to remember :-

* + If any error comes like failed to save resource then restart the studio with run as admin.
  + While creating the http end point give the url of the independent service you created earlier, also while doing steps ignore [MSF4J service](https://docs.wso2.com/display/EI650/Key+Concepts#KeyConcepts-msf4j). As you have already created the independent micorservice.
  + Follow all the steps except MSF4J but do remember to start the carbon.sh, which will start the console page.
  + Now you have created an API with some end point you can see that in <https://lcoalhost:9443/carbon/> url with admin/admin user/pass in API section.
  + Hit the microservice with proper url and see the response.