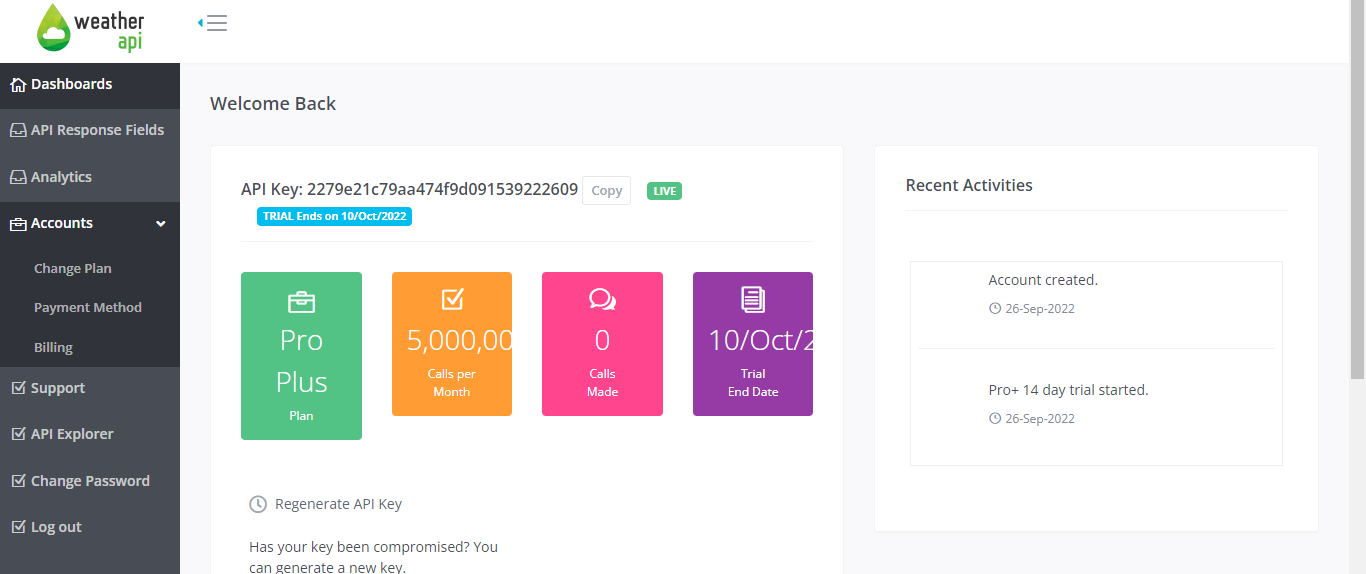
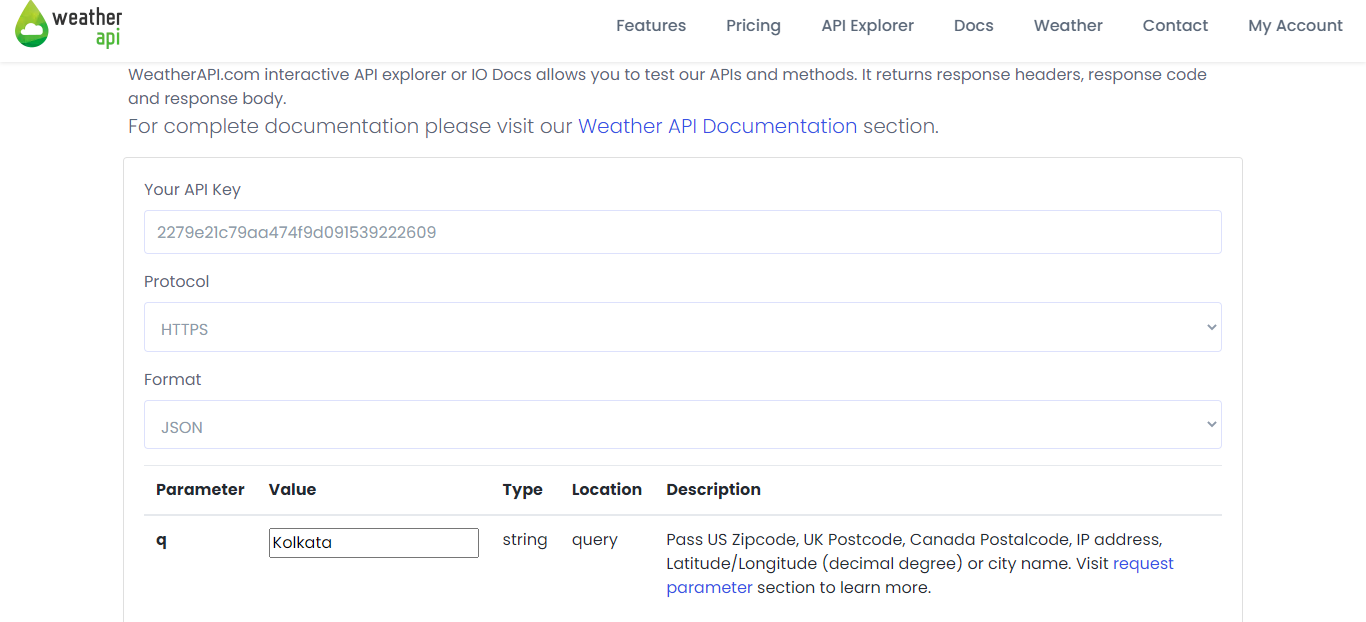
Main Agenda- To connect an external service through REST Connector

For this we would be needing an External API to do this. So we are going for the Weather API so we need the key to generate this

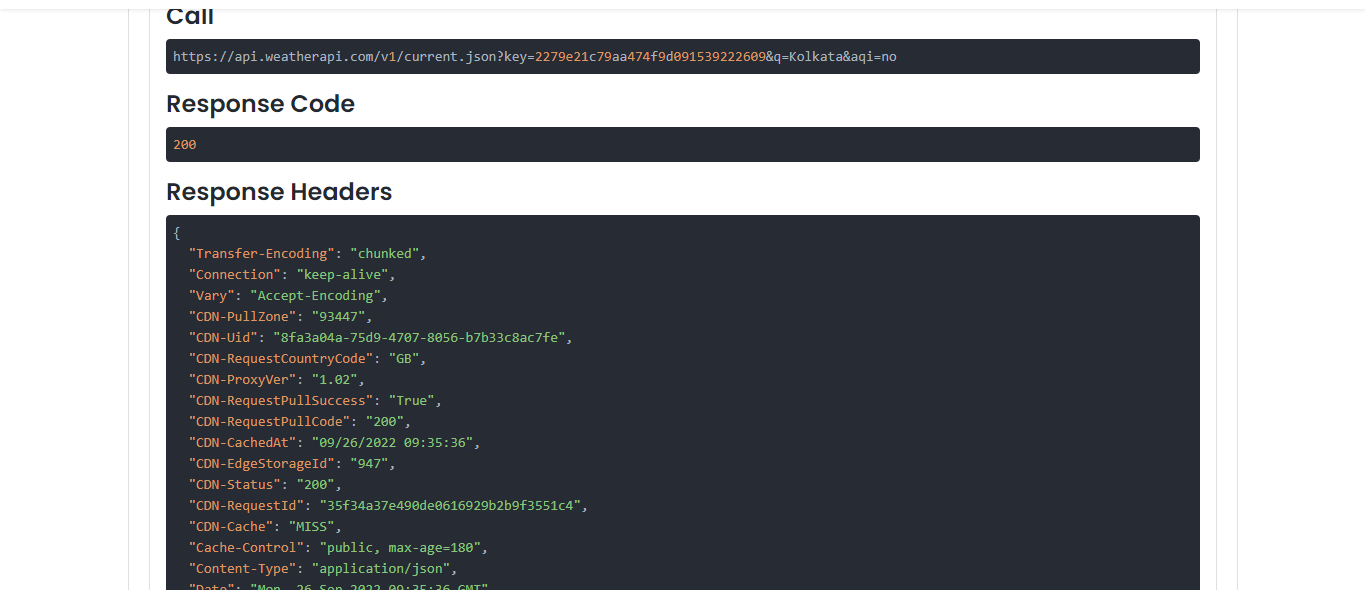
First we can create one account and Signup in the Weather API and then we would be getting our API Key.

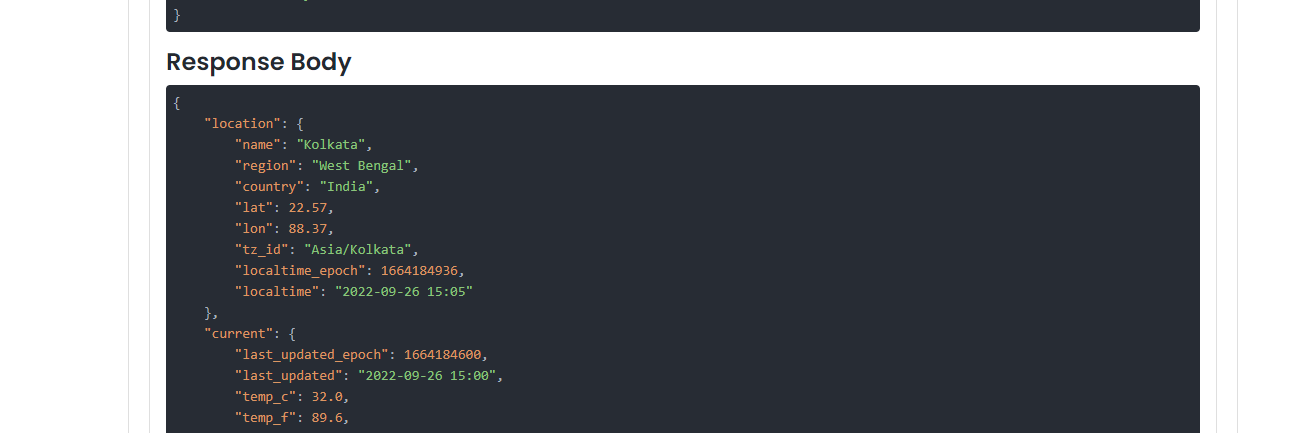


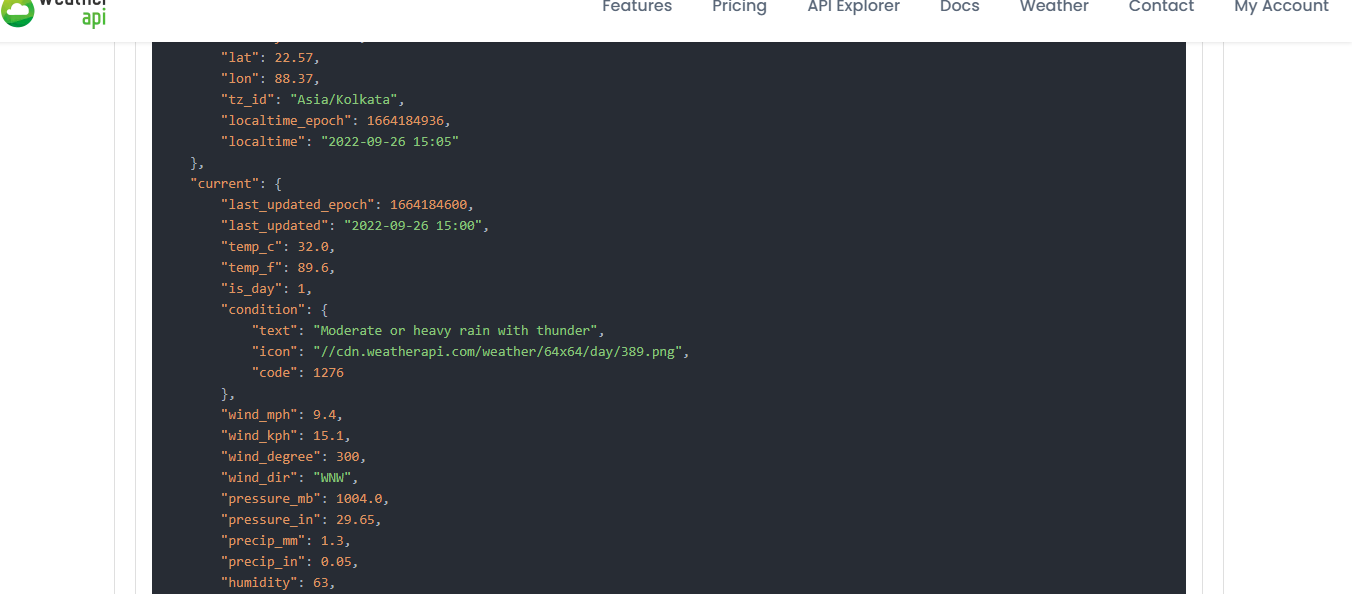
And after that we can get the interactive API Explorer where we will get our Key Credentials   


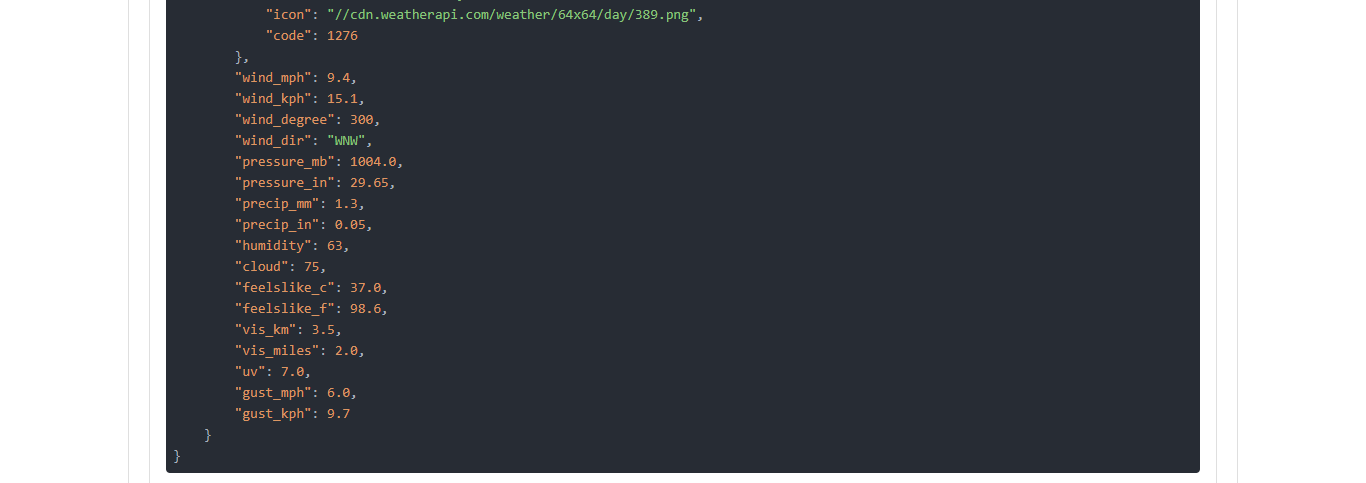
And In the query you can give on which basis we need the values so I entered one state Kolkata

And when we hit the show response then a call ID/Request ID is generated and response header sand body are generated.





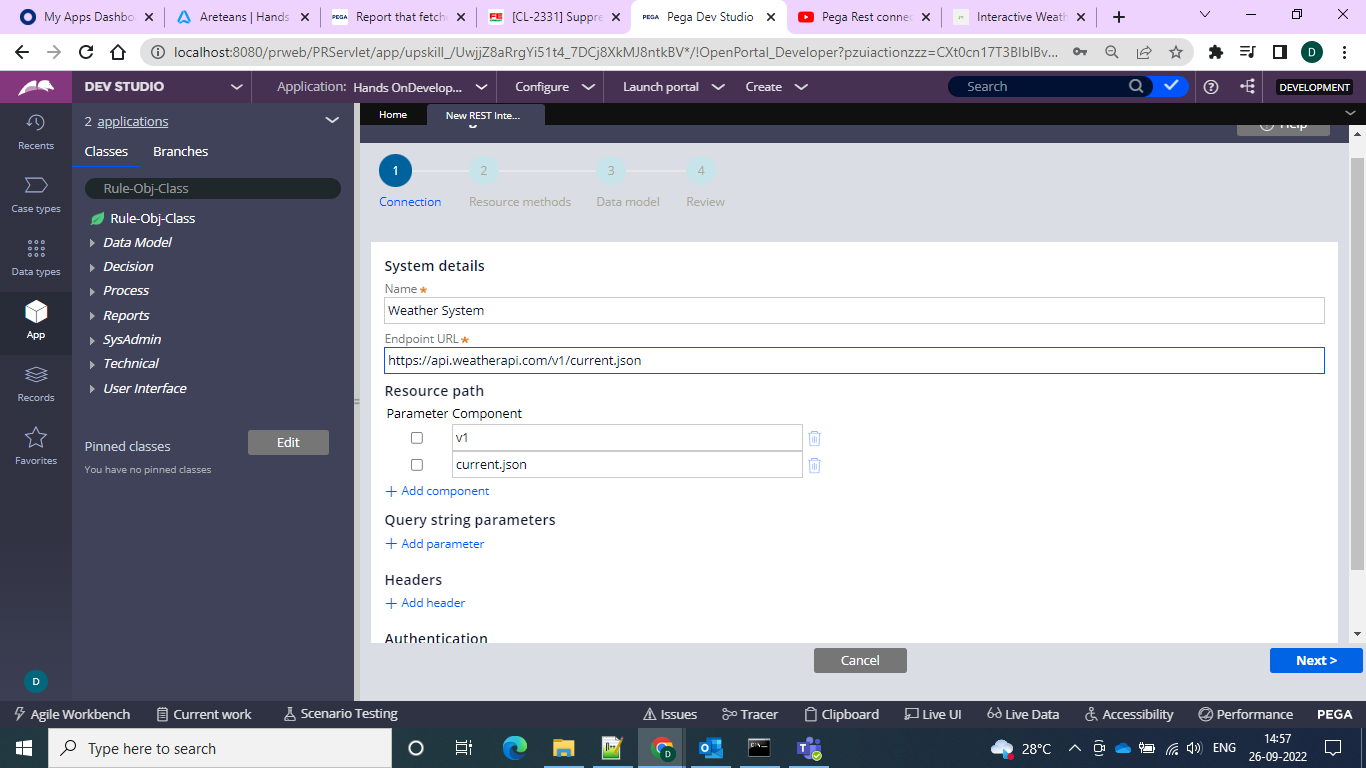




And after all this code generation from the API Service now we can configure our Rest API and can give the configurations.

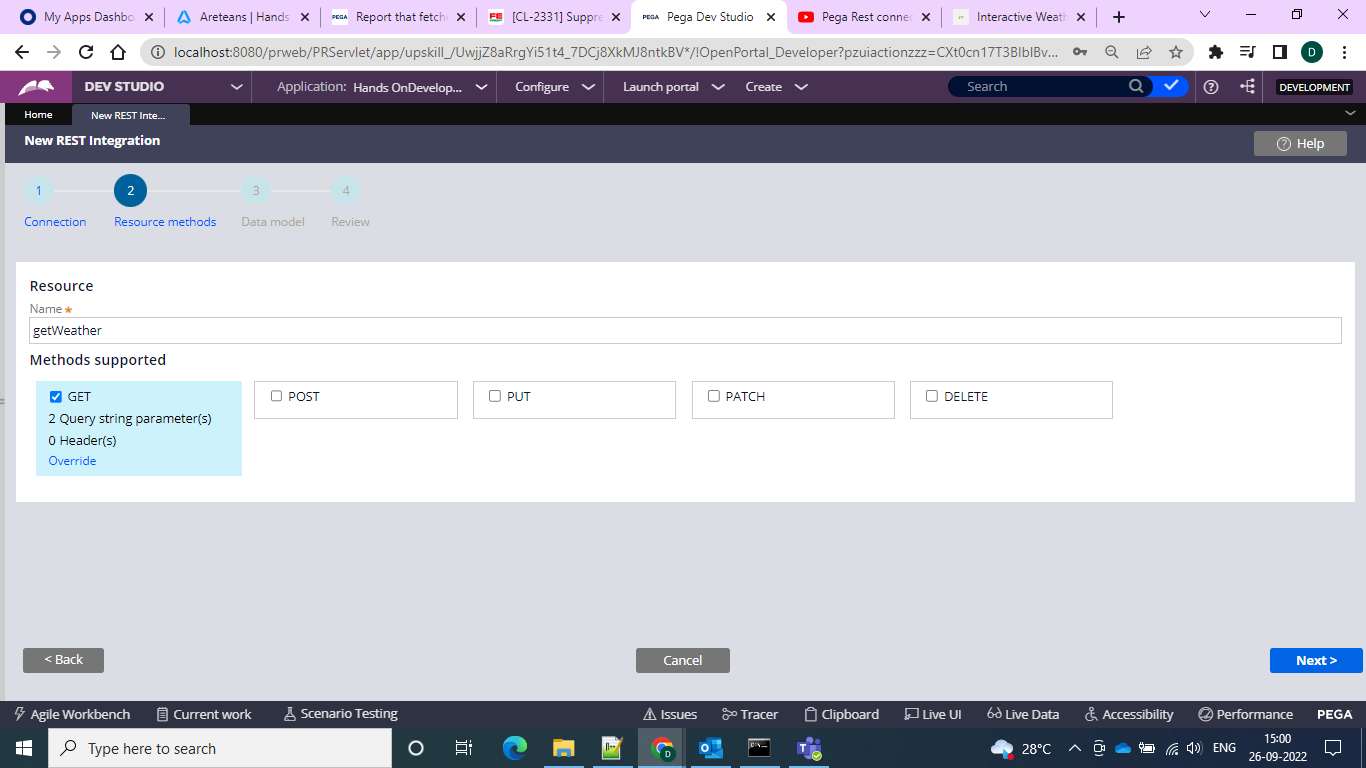
Enter into Pega

Go to integration Connectors And then go for Connect REST

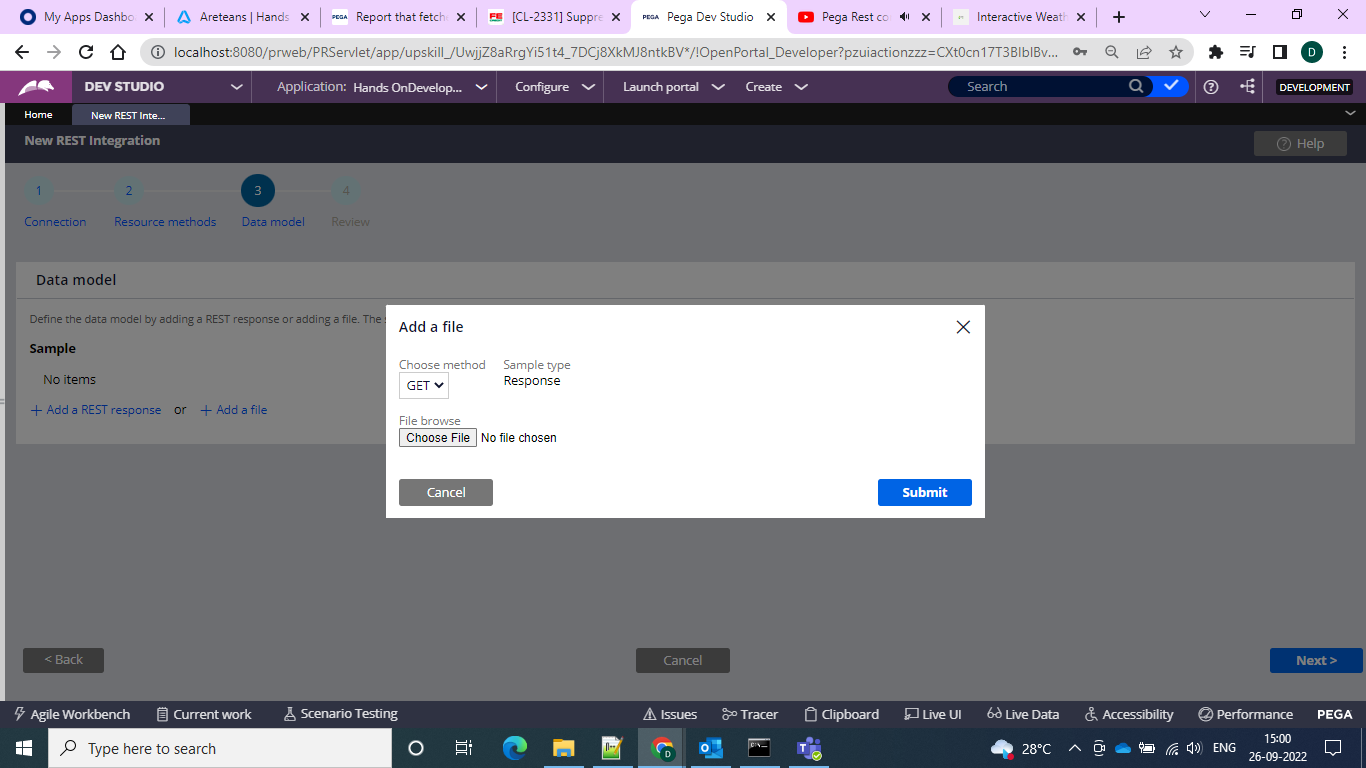


From the above screen give the Endpoint URL as the call path URL and then the resource path would already be generated.

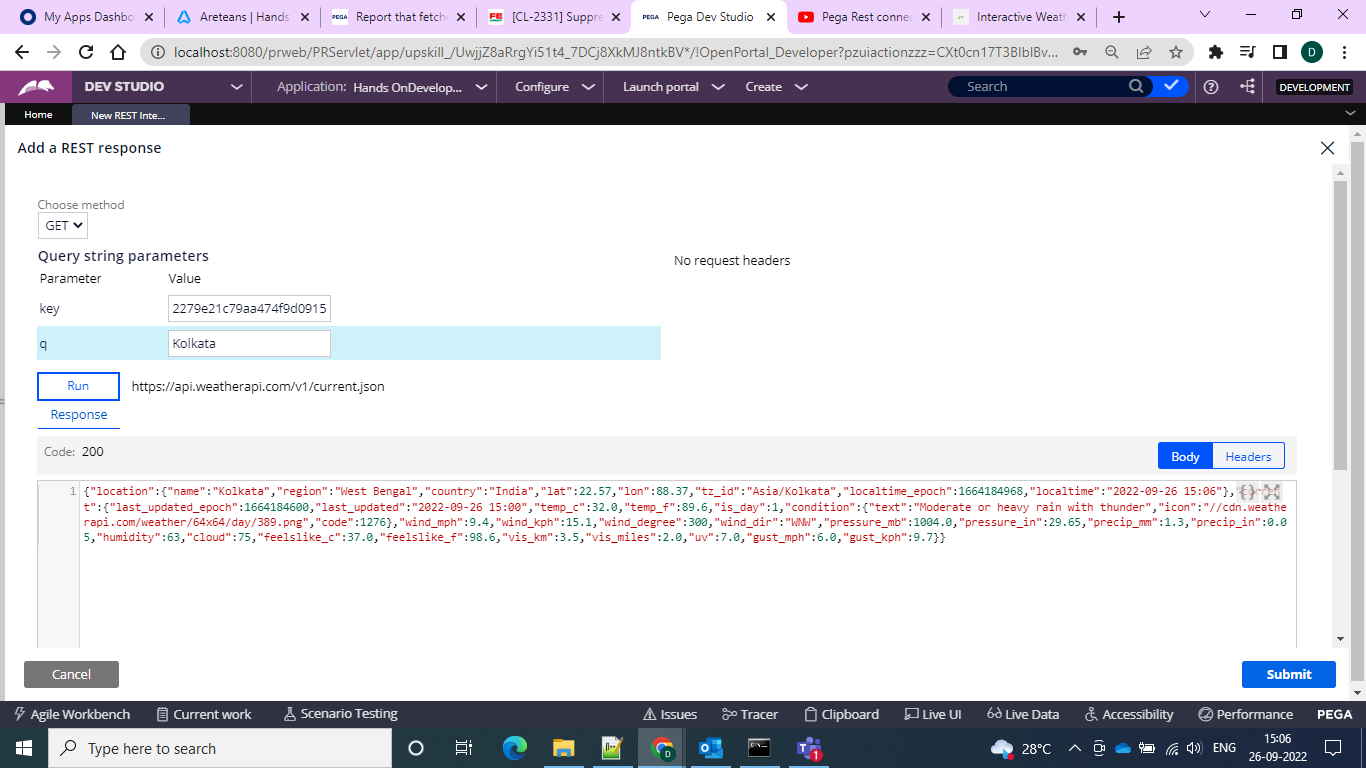
Then in the query string parameters give the “Key” and “q” parameters taken from the API so that it can filter in the following way and click next and then select a method I am selecting GET because we need to get the details from the API



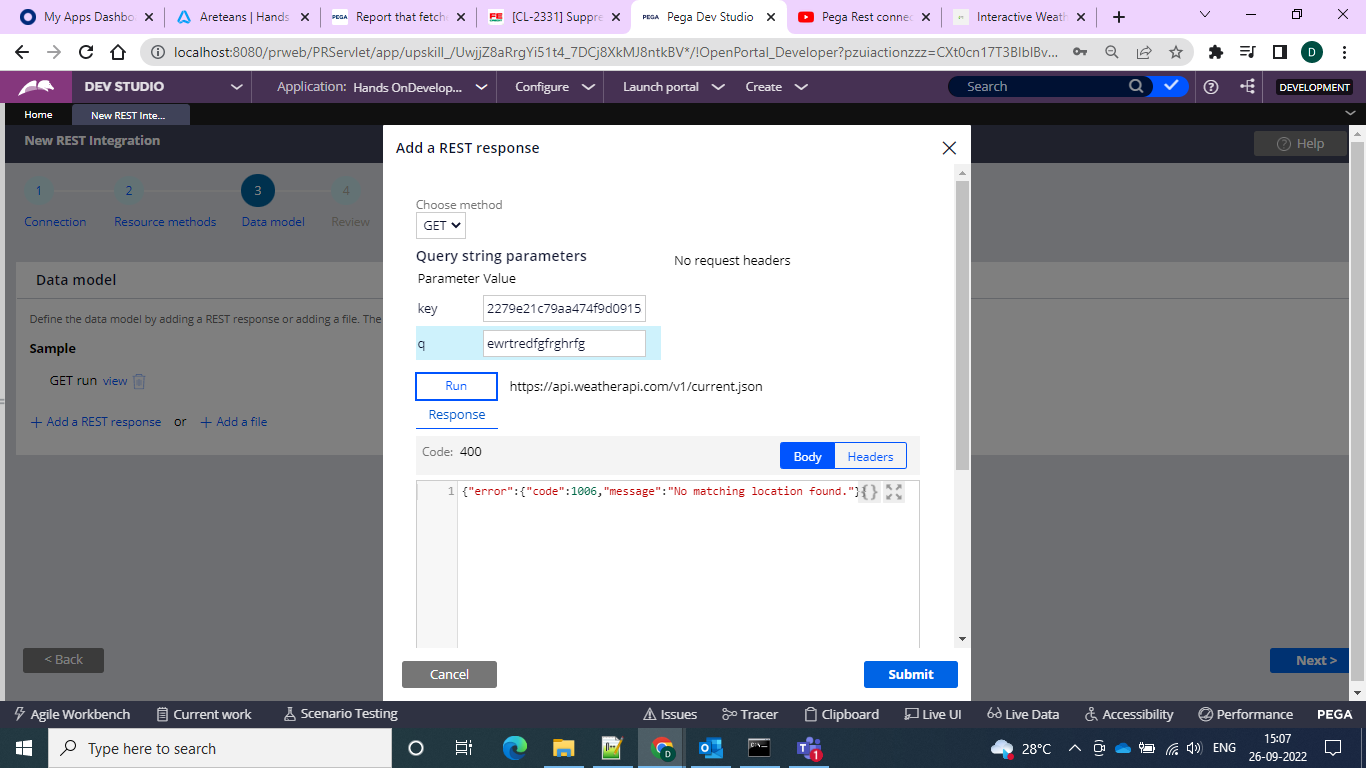
And then comes the data model where in since we selected GET Method so it would show as response and method as GET



And since we have the key and q so we can give the parameter values and get the result and test the code



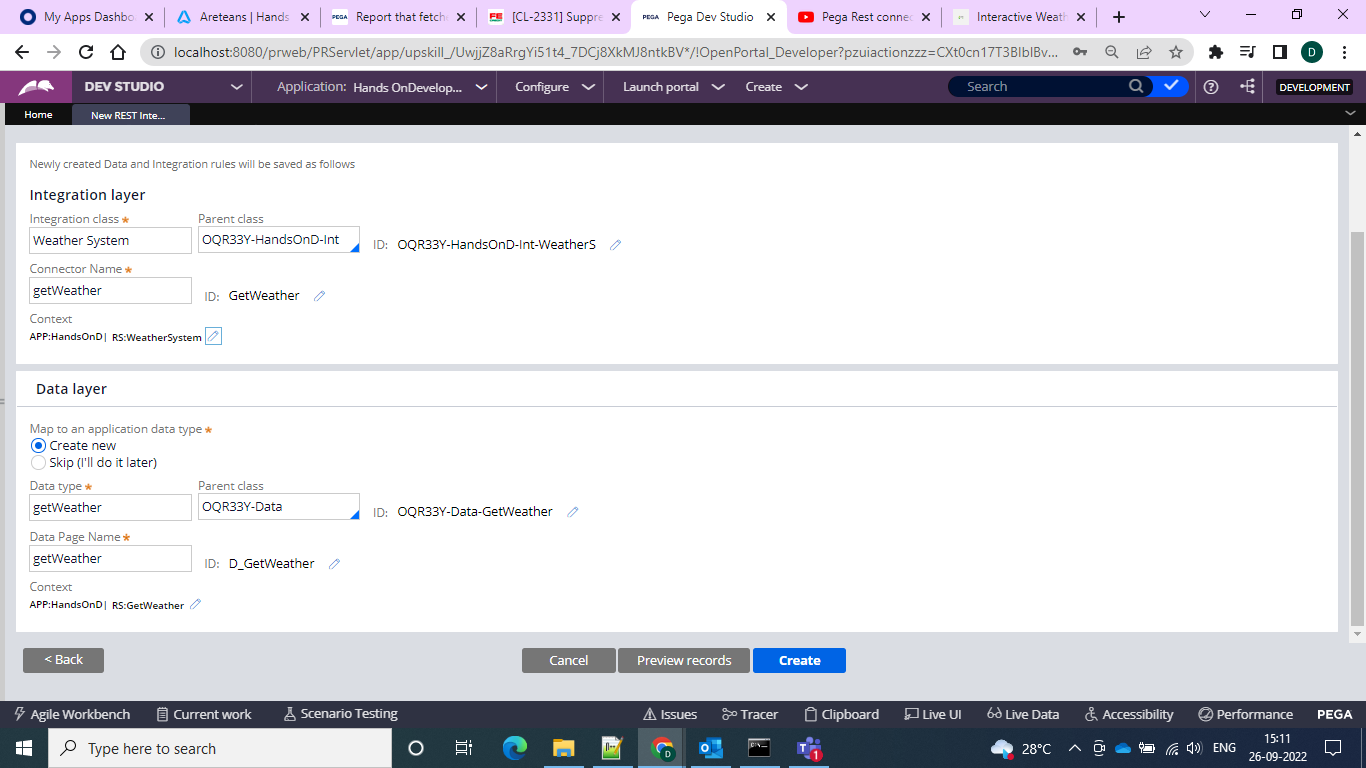
And if we give any random key the API would fail and then we get the error message as below



And then we can add both the sample model views and after that we can go for the review

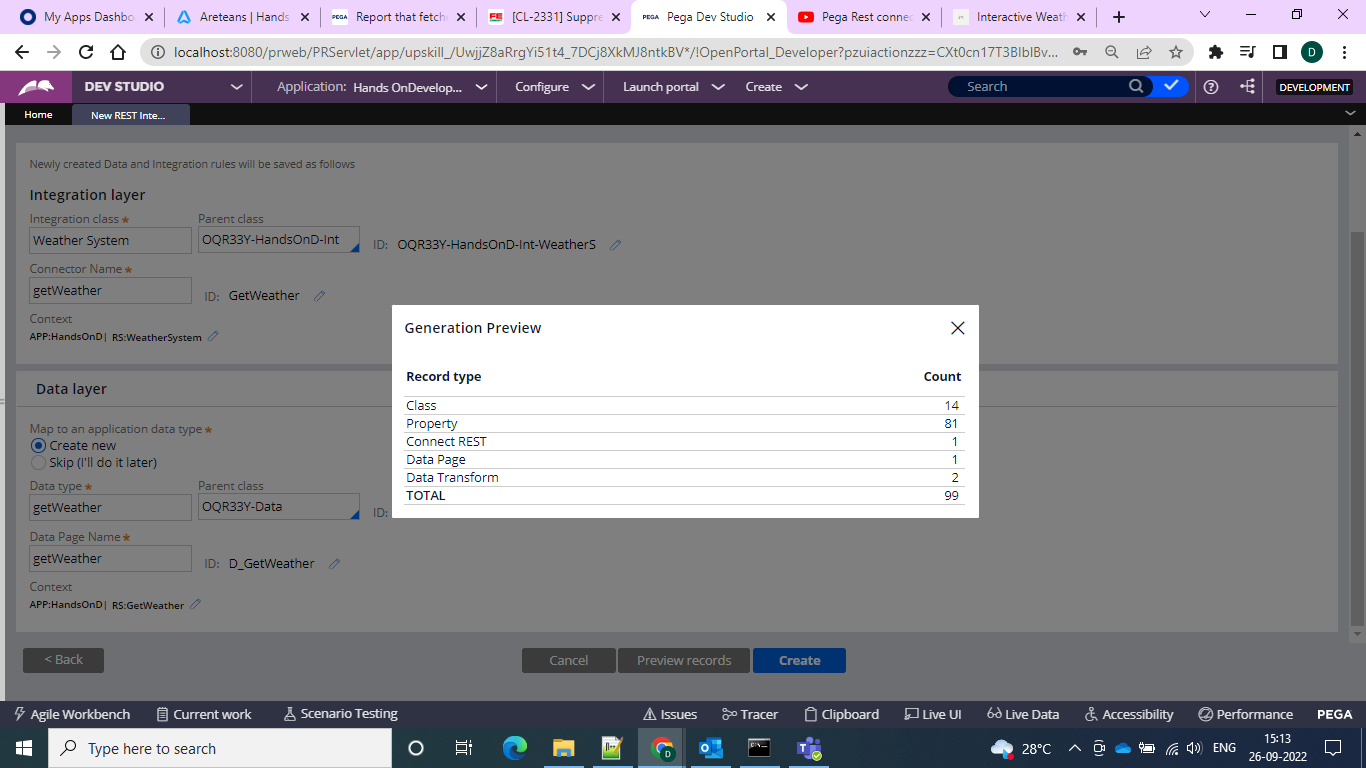
Where we can choose the layer and now we must use the Application layer and use it to the application we can check through tracer and get the info that it is mapped to integration layer only

And we can get the properties from layer

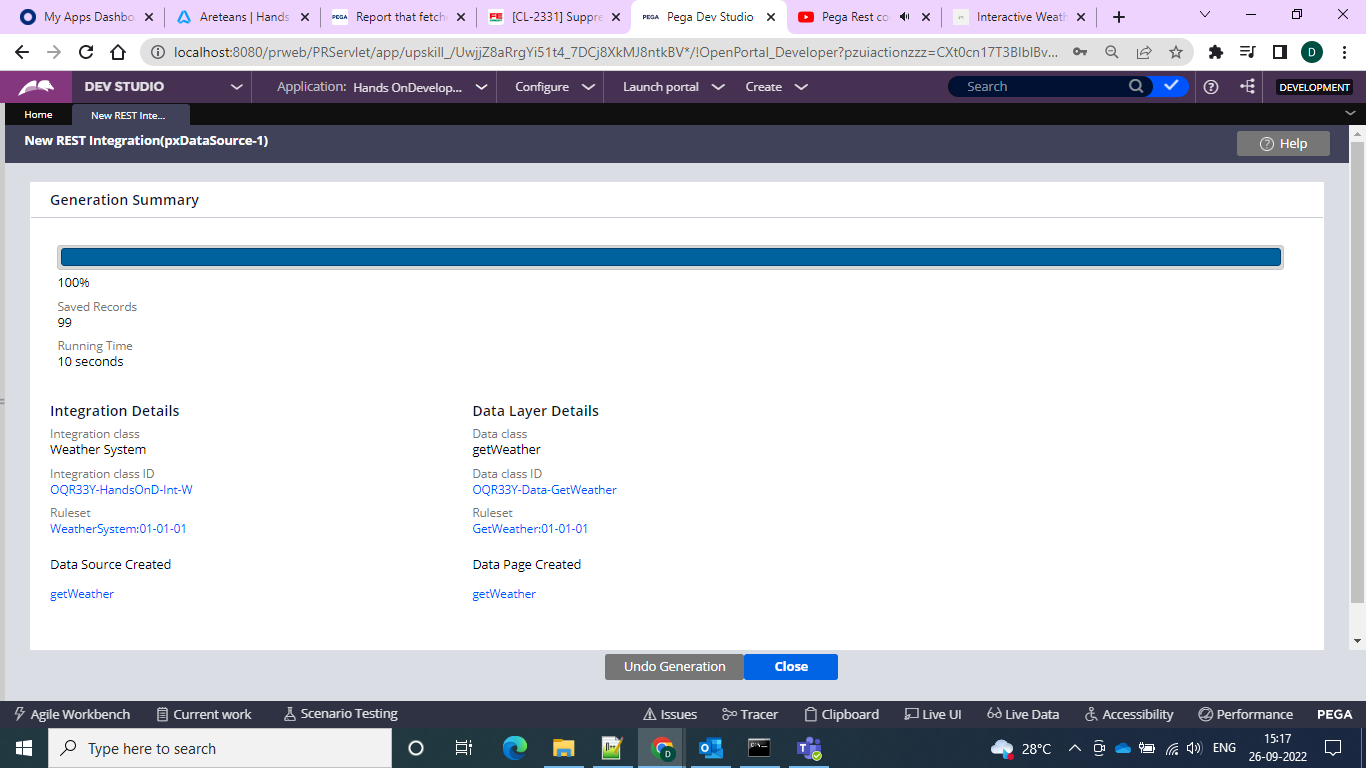


And note all these classes and details then we can go for the generation

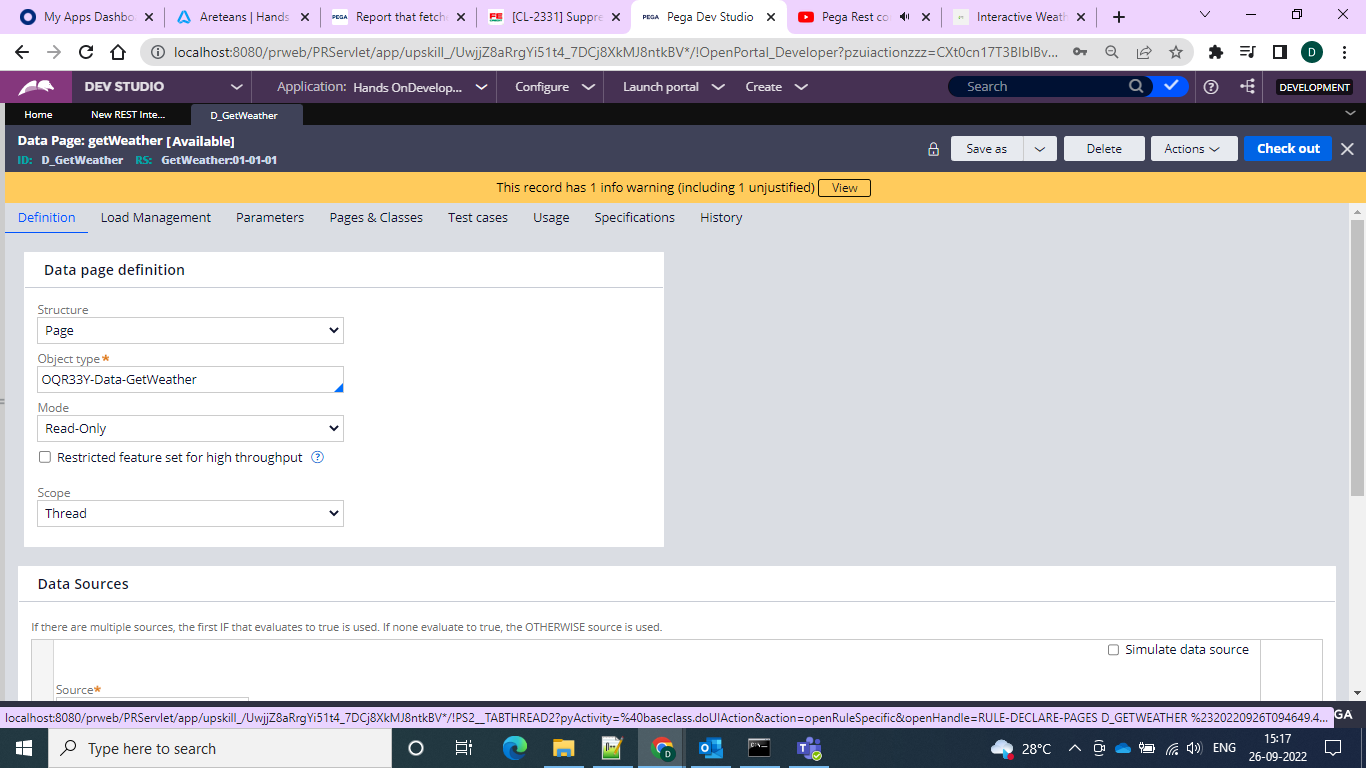
And we can do the preview what would be created for the ruleset

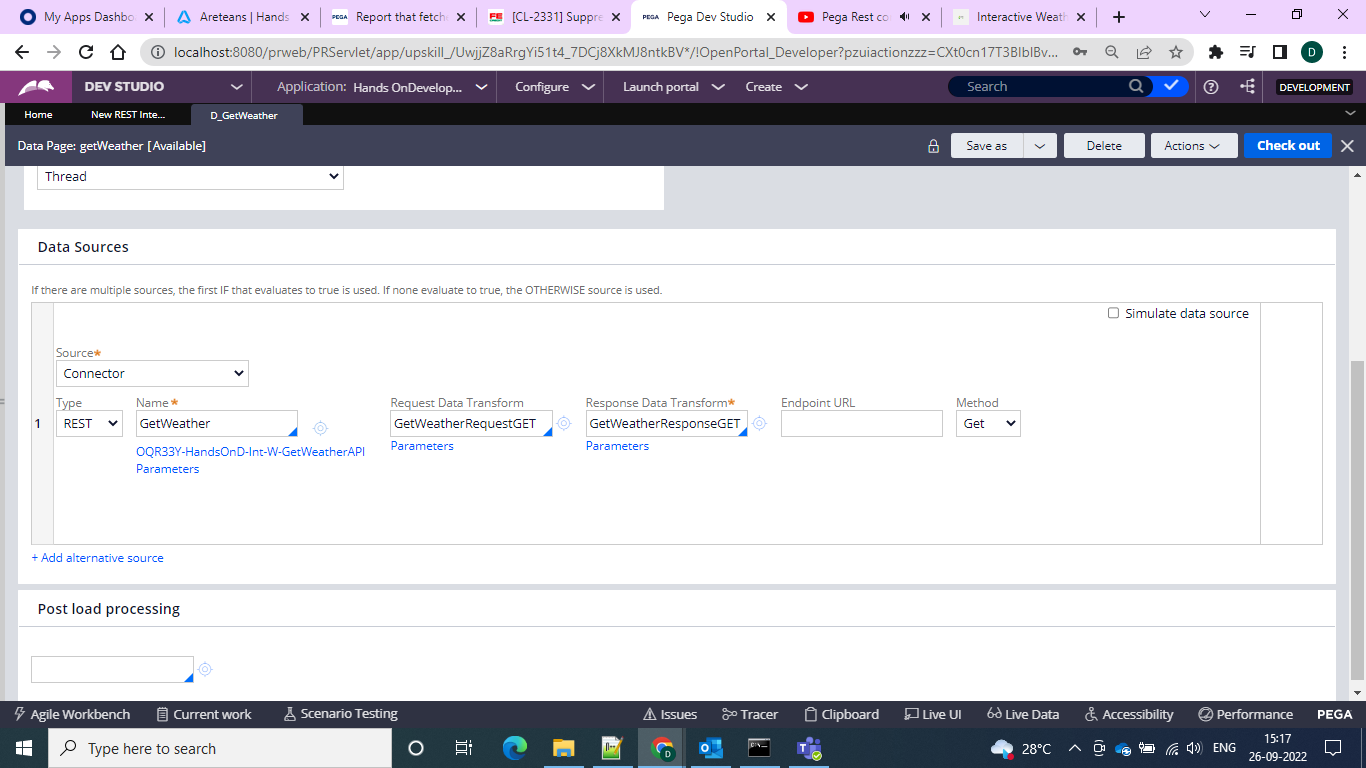


And the generation is successful

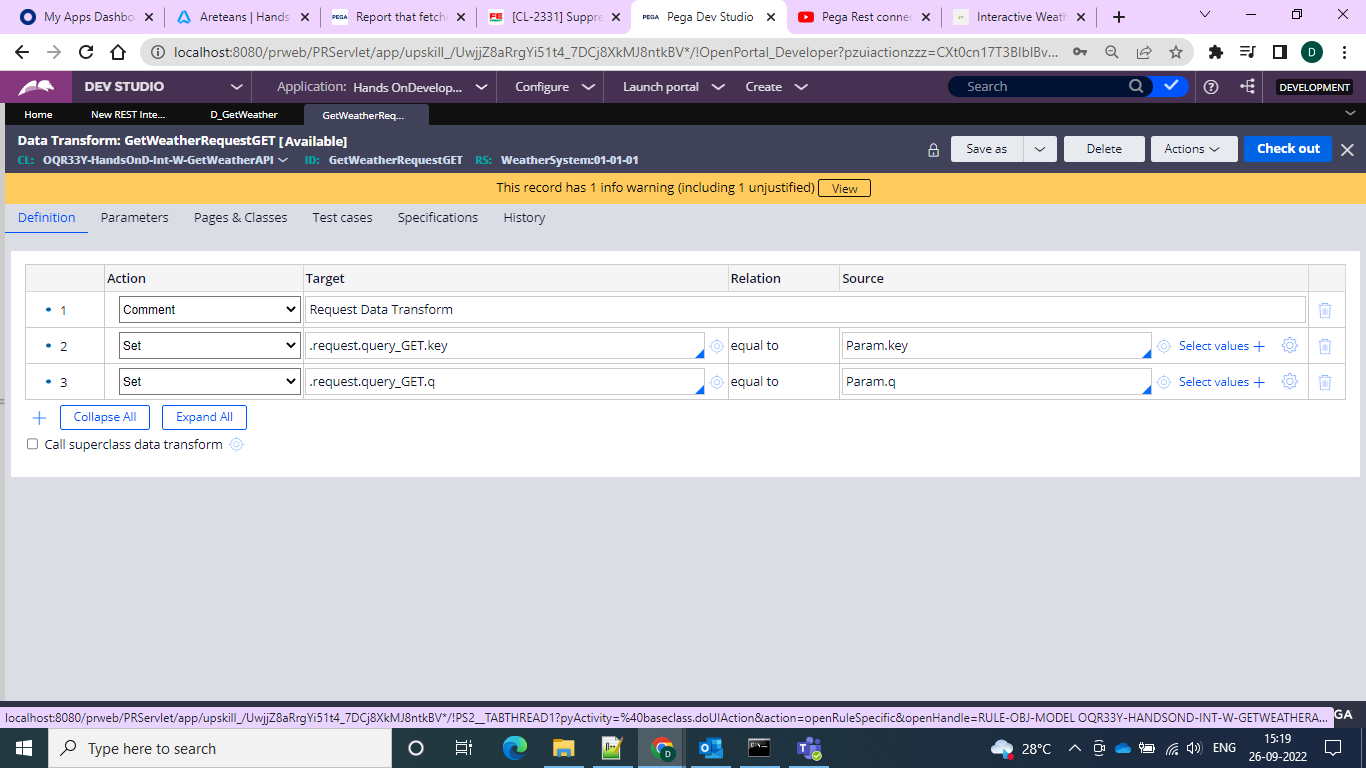


And after that we would be getting the Data Page with request and response data transforms

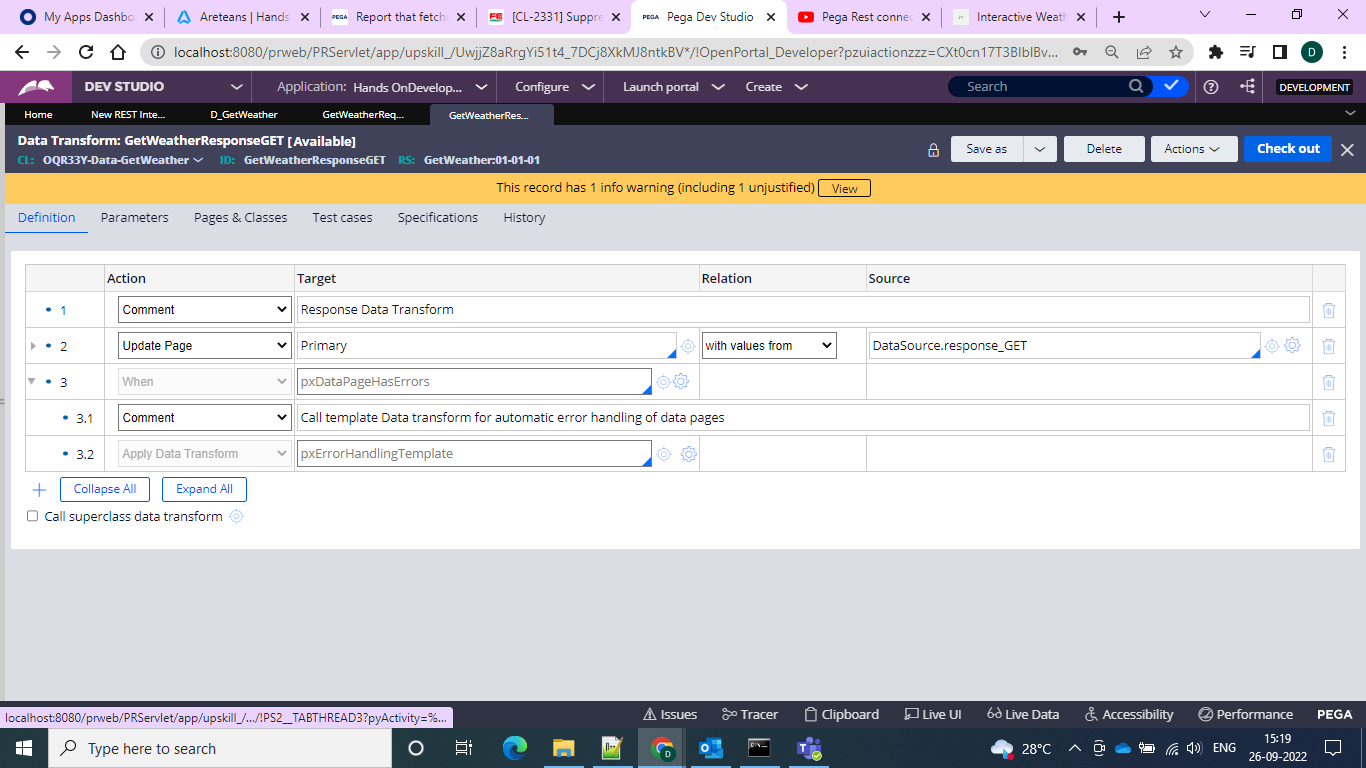




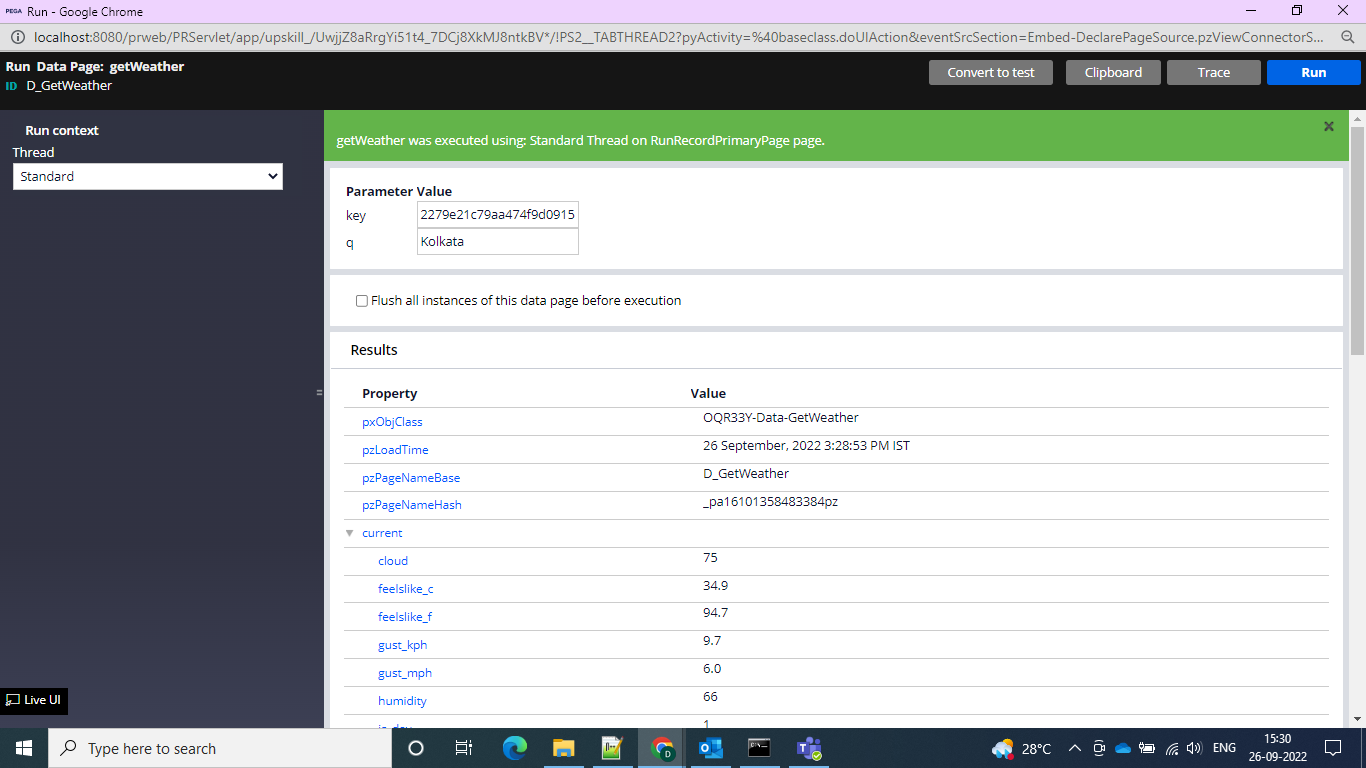
And in the request Data transform we get the following

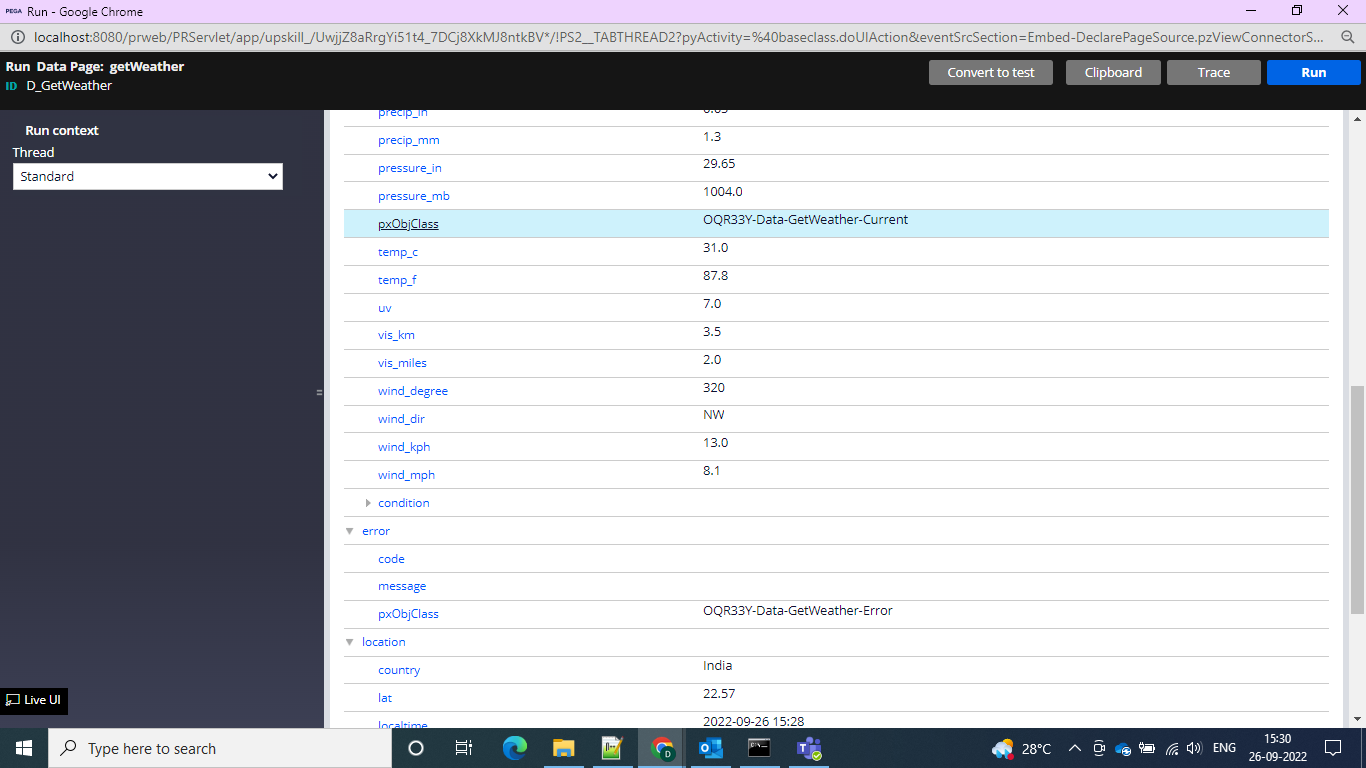


And the response would be like this

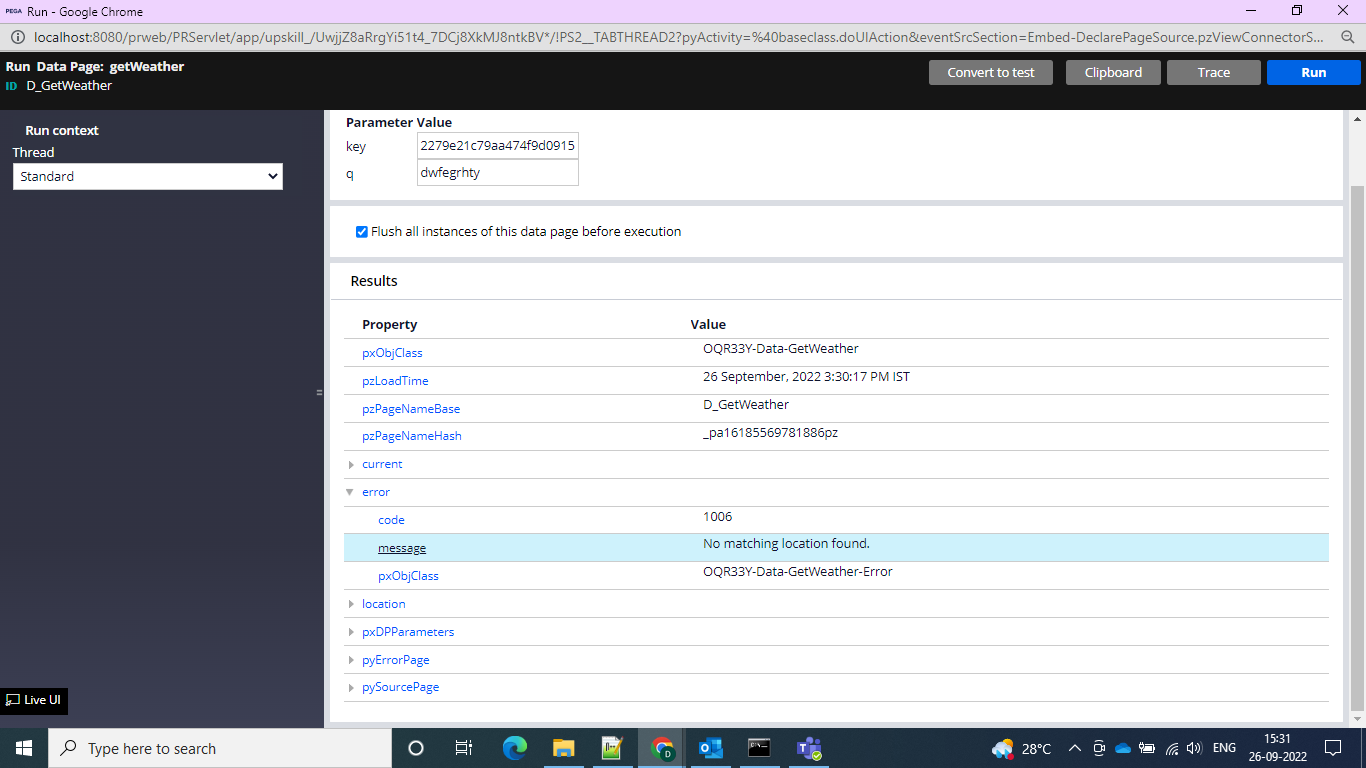


And when we run and test the D\_Page with the given key and param “q” on which it should get the values





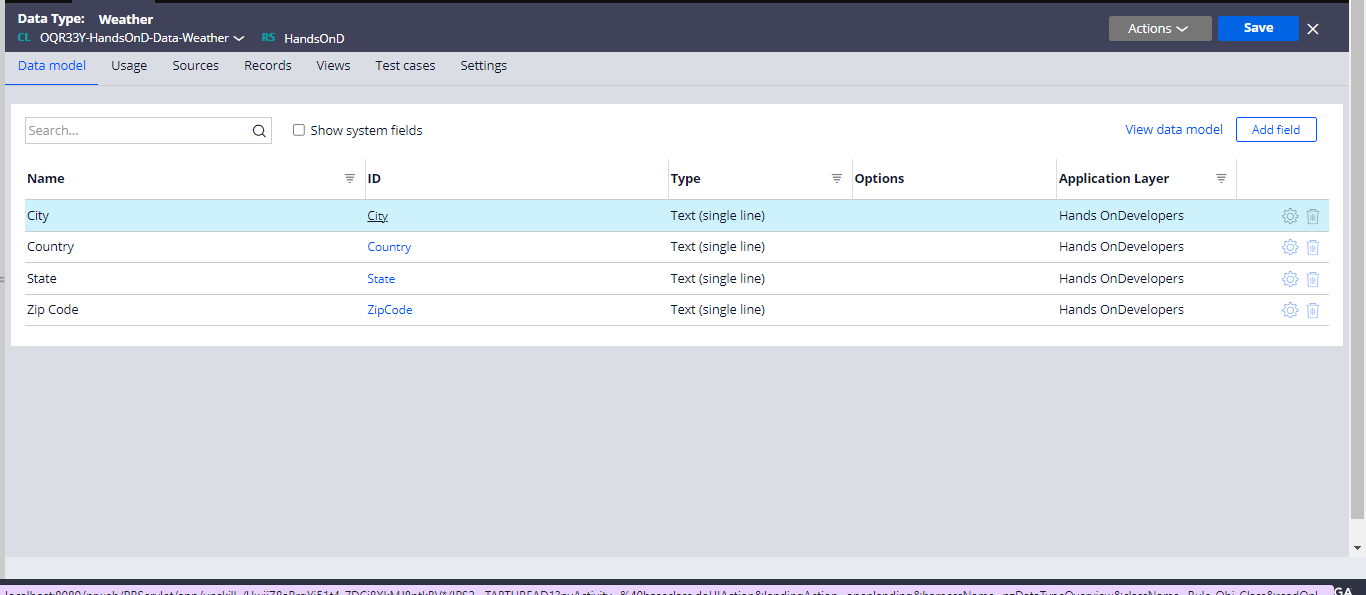
And if given wrong parameter instead of the correct city we would get the

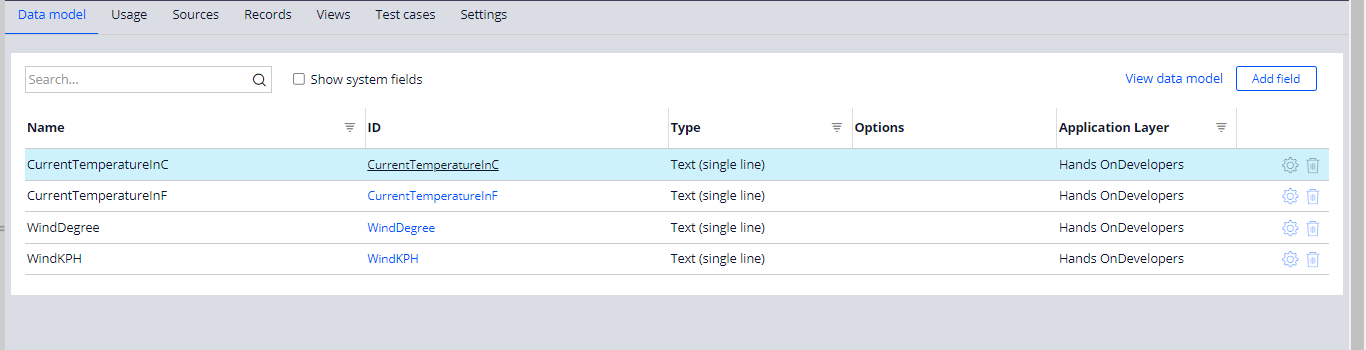


And error message would come

And then we can create 2 data types one for the mapping property and one for the param to which it should map and take the values

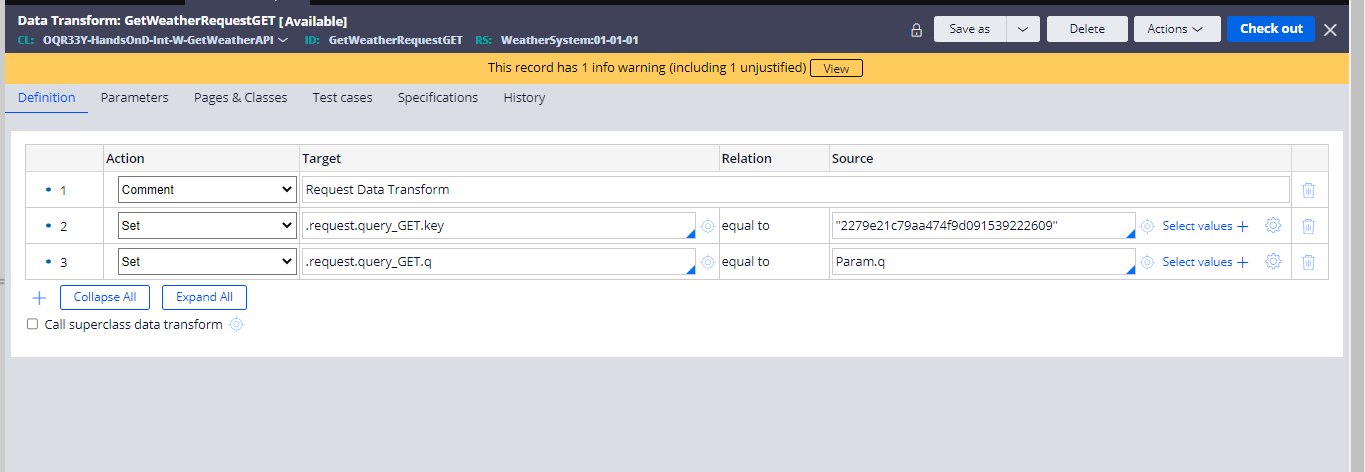
So let’s go for the data types





Amd then we can go for the request and response DT and our own one DT we need to create to update the Data page and map the properties to the data page

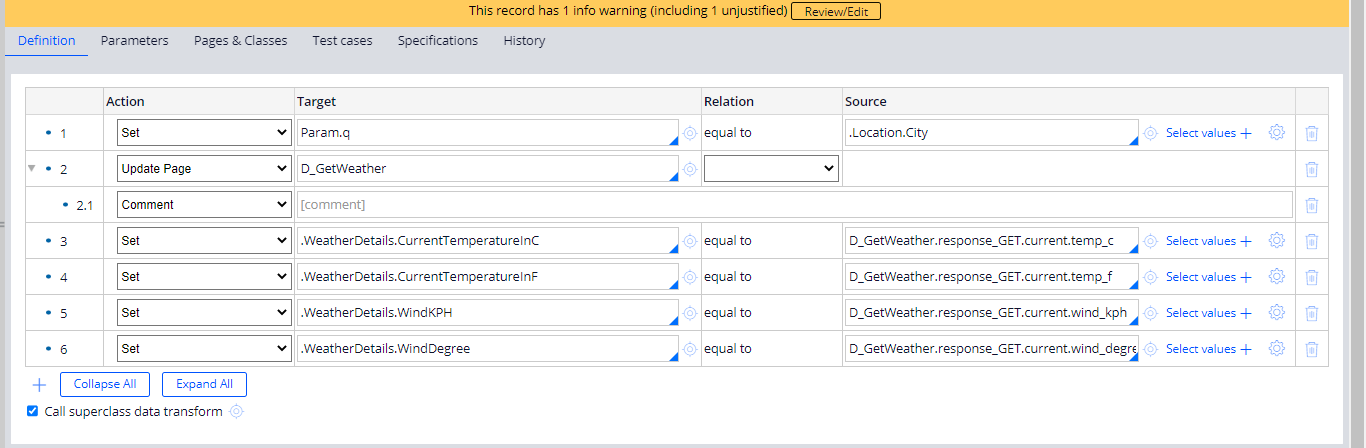
So in we can go for the request DT and can give the key and param



And then we can go for the response data transform and can give as to map and update the page by values from the response.GET and then can go for configuring of our own DT to call in the assignment shape to get the results

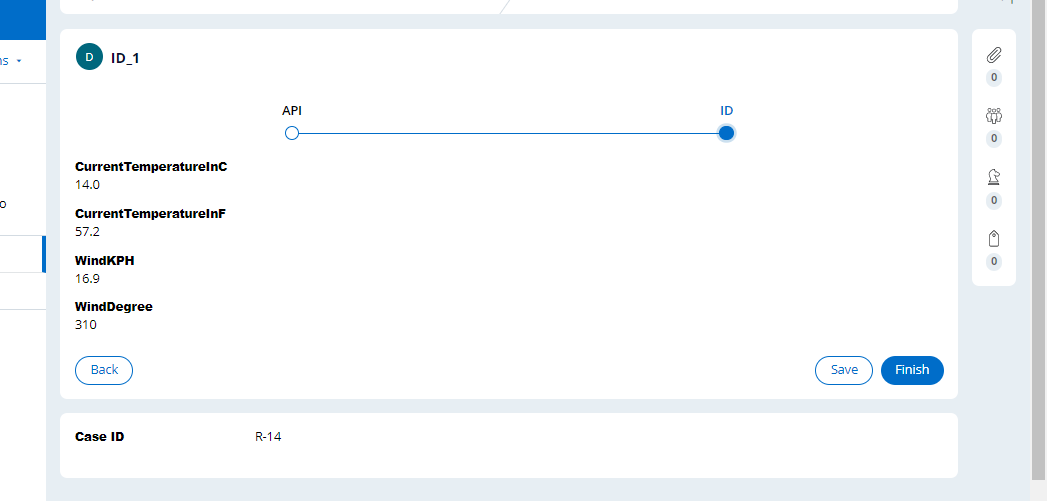
And in the DT in the first step we can give as City so that by taking city as parameter values can be displayed and in the second step can give as update page so that you can update the page with the values

And from the 3rd step we can map the properties on the source taken from the response and on the target can map the properties that which we are showing in our application.

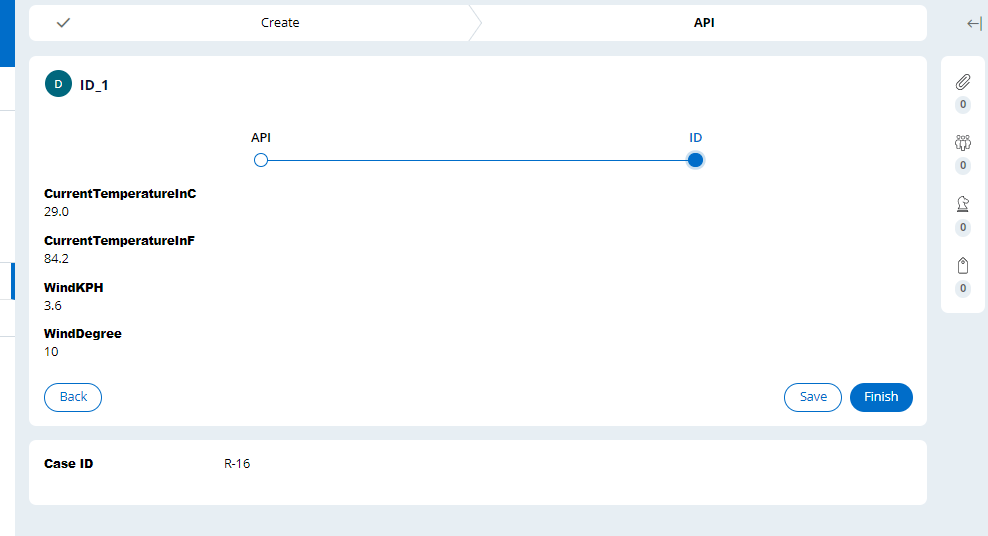


And the results are as below

London



Kolkata



Pune

