

1. Register a new user at <https://openweathermap.org/api> to obtain an API key.
2. Open the Katalon Studio project named 'Testing API'.
3. Create a Web Service Request object in the Object Repository.
  - 3.1. Input the name as get\_details.
  - 3.2. Select the Request Type as RESTful and enter the URL:  
<http://api.openweathermap.org/geo/1.0/direct>.
  - 3.3. Select the request method as GET.
  - 3.4. Add Query Parameters:
    - 3.4.1. Name = appid, Value = the API key obtained from the website.
    - 3.4.2. Name = q, Value = Jakarta Selatan.
  - 3.5. Execute the request to view the results.
  - 3.6. Copy the latitude value from the response.
  - 3.7. Copy the longitude value from the response.
4. Create another Web Service Request object in the Object Repository named weather\_api.
  - 4.1. Select the Request Type as RESTful and enter the URL:  
<http://api.openweathermap.org/data/2.5/forecast>.
  - 4.2. Select the request method as GET.
  - 4.3. Add Query Parameters:
    - 4.3.1. Name = appid, Value = the API key obtained from the website.
    - 4.3.2. Name = lat, Value = latitude copied from the previous response.
    - 4.3.3. Name = lon, Value = longitude copied from the previous response.
    - 4.3.4. Name = cnt, Value = 5, because only a 5-day forecast is needed.
5. Create another Web Service Request object in the Object Repository named pollution\_api.
  - 5.1. Select the Request Type as RESTful and enter the URL:  
[http://api.openweathermap.org/data/2.5/air\\_pollution](http://api.openweathermap.org/data/2.5/air_pollution).
  - 5.2. Select the request method as GET.
  - 5.3. Add Query Parameters:
    - 5.3.1. Name = appid, Value = the API key obtained from the website.
    - 5.3.2. Name = lat, Value = latitude copied from the previous response.
    - 5.3.3. Name = lon, Value = longitude copied from the previous response.
6. Create a test case named Forecast\_Weather.
  - 6.1. Call the object weather\_api.
  - 6.2. Send the request.
  - 6.3. Verify that the response status code is 200.
  - 6.4. Retrieve the response body and store it as a plain text string.
  - 6.5. Convert the string variable into JSON format.
  - 6.6. Check that the total number of forecast days is 5, based on the dt\_txt field containing the date.
  - 6.7. Validate that there are 5 forecast data entries; if not, raise an error.

- 6.8. Validate the JSON schema to ensure that the structure and data types of the API response are correct.
  - 6.8.1. Copy the API response content from weather\_api.
  - 6.8.2. Open a site like [quicktype.io](https://quicktype.io).
  - 6.8.3. Paste the JSON response into the input field and let the system generate a JSON Schema.
  - 6.8.4. Copy the generated JSON Schema.
  - 6.8.5. Save the schema into the Katalon project, usually under Include/scripts/groovy/schemas/.
  - 6.8.6. Save the file in .json format (e.g., weather.schema.json).
  - 6.8.7. The file will be read locally from your computer.
  - 6.8.8. Convert the read file into a JSON object for processing in Groovy/Katalon code.
  - 6.8.9. Convert the JSON Schema into a JSONObject.
  - 6.8.10. Convert the API response body into a JSONObject.
  - 6.8.11. Validate whether the API response matches the schema; if successful, print "SUCCESS", otherwise print all validation errors.
7. Create a test case named Air\_Pollution.
  - 7.1. Call the object pollution\_api.
  - 7.2. Send the request.
  - 7.3. Verify that the response status code is 200.
  - 7.4. Retrieve the response body and store it as a plain text string.
  - 7.5. Convert the string variable into JSON format.
  - 7.6. Convert the dt value (Unix timestamp) into a UTC date format and validate it against the current time, then store dt as a long type variable.
  - 7.7. Calculate the corresponding milliseconds.
  - 7.8. Format the calculated date into UTC timezone using the pattern yyyy-MM-dd HH:mm:ss with SimpleDateFormat.
  - 7.9. Convert the Date object into a neatly formatted readable string, then print it.
  - 7.10. Retrieve the current system time into a variable of type long.
  - 7.11. Compare the current time (variable 1) and the API response time (dt \* 1000) to check if they match; if they match, it means the API provides real-time data correctly.
  - 7.12. Validate the JSON schema to ensure the API response structure and data types are correct.
    - 7.12.1. Copy the API response content from pollution\_api.
    - 7.12.2. Open a site like [quicktype.io](https://quicktype.io).
    - 7.12.3. Paste the JSON response into the input field and let the system generate a JSON Schema.
    - 7.12.4. Copy the generated JSON Schema.
    - 7.12.5. Save the schema into the Katalon project, usually under Include/scripts/groovy/schemas/.

- 7.12.6. Save the file in .json format (e.g., pollution.schema.json).
- 7.12.7. The file will be read locally from your computer.
- 7.12.8. Convert the read file into a JSON object for processing in Groovy/Katalon code.
- 7.12.9. Convert the JSON Schema into a JSONObject.
- 7.12.10. Convert the API response body into a JSONObject.
- 7.12.11. Validate whether the API response matches the schema; if successful, print "SUCCESS", otherwise print all validation errors.