TS_OpenWeatherADL

Execution Environment

Host name	sahat - SAHATHTRK
Local OS	Windows 11 64bit
Katalon version	9.3.2.212
Browser	null

Summary

ID	Test Suites/TS_OpenWeatherADL		
Description			
Total	2		
Passed	2	Failed	0
Error	0	Incomplete	0
Skipped	0		
Start	26-04-2025 19:53:13	End	26-04-2025 19:53:24
Elapsed	11.323s		

#	ID	Description	Status
1	Test Cases/TC_Get5DaysForecast		PASSED
2	Test Cases/TC_GetCurrentAirPollution		PASSED

TC_Get5DaysForecast

Information

ID	Test Cases/TC_Get5DaysForecast		
Description			
Tag			
Start	26-04-2025 19:53:13	End	26-04-2025 19:53:19
Elapsed	6.524s		
Status	PASSED		

Details

#	Description	Elapsed	Status
1	requestObjectGeocoding = findTestObject("Object Repository/Get_Geocoding")	1.166s	PASSED
2	responseGeocoding = sendRequest(requestObjectGeocoding) Send request successfully	3.353s	PASSED
3	verifyResponseStatusCode(responseGeocoding, 200) Verify response status code successfully	0.035s	PASSED
4	verifyResponseStatusCodeInRange(responseGeocoding, 200, 299) Verify response status code successfully	0.030s	PASSED
5	jsonResponseGeocoding = JsonSlurper().parseText(responseGeocoding.getResponseText())	0.064s	PASSED
6	lat = lat.toString()	0.045s	PASSED
7	lon = lon.toString()	0.005s	PASSED

TC_Get5DaysForecast

#	Description	Elapsed	Status
8	latitude = lat	0.003s	PASSED
9	longitude = lon	0.004s	PASSED
10	requestObject = findTestObject("Object Repository/Get_5DaysForecast")	0.353s	PASSED
11	response = sendRequest(requestObject) Send request successfully	0.613s	PASSED
12	verifyResponseStatusCode(response, 200) Verify response status code successfully	0.006s	PASSED
13	verifyResponseStatusCodeInRange(response, 200, 299) Verify response status code successfully	0.006s	PASSED
14	jsonResponse = JsonSlurper().parseText(response.getResponseText())	0.008s	PASSED
15	cityName = name.toString()	0.003s	PASSED
16	println("nama kota: " + cityName)	0.012s	PASSED
17	geoLat = Double.parseDouble(latitude.trim())	0.020s	PASSED

TC_Get5DaysForecast

#	Description	Elapsed	Status
18	geoLon = Double.parseDouble(longitude.trim())	0.004s	PASSED
19	airpolutionLat = lat	0.003s	PASSED
20	airpolutionLon = lon	0.002s	PASSED
21	tolerance = 0.001	0.003s	PASSED
22	assert ((java.lang.Math.abs((geoLat - airpolutionLat)) <= tolerance)) ? Latitude match : Latitude doesn't match	0.040s	PASSED
23	assert ((java.lang.Math.abs((geoLon - airpolutionLon)) <= tolerance)) ? Longitude match : Longitude doesn't match	0.002s	PASSED

TC_GetCurrentAirPollution

Information

ID	Test Cases/TC_GetCurrentAirPollution		
Description			
Tag			
Start	26-04-2025 19:53:20	End	26-04-2025 19:53:21
Elapsed	1.633s		
Status	PASSED		

Details

#	Description	Elapsed	Status
1	requestObjectGeocoding = findTestObject("Object Repository/Get_Geocoding")	0.221s	PASSED
2	responseGeocoding = sendRequest(requestObjectGeocoding) Send request successfully	0.421s	PASSED
3	verifyResponseStatusCode(responseGeocoding, 200) Verify response status code successfully	0.007s	PASSED
4	verifyResponseStatusCodeInRange(responseGeocoding, 200, 299) Verify response status code successfully	0.007s	PASSED
5	jsonResponseGeocoding = JsonSlurper().parseText(responseGeocoding.getResponseText())	0.003s	PASSED
6	lat = lat.toString()	0.002s	PASSED
7	lon = lon.toString()	0.018s	PASSED

TC_GetCurrentAirPollution

#	Description	Elapsed	Status
8	latitude = lat	0.003s	PASSED
9	longitude = lon	0.002s	PASSED
10	requestObject = findTestObject("Object Repository/Get_CurrentAirPollution")	0.299s	PASSED
11	response = sendRequest(requestObject) Send request successfully	0.397s	PASSED
12	verifyResponseStatusCode(response, 200) Verify response status code successfully	0.005s	PASSED
13	verifyResponseStatusCodeInRange(response, 200, 299) Verify response status code successfully	0.005s	PASSED
14	jsonResponse = JsonSlurper().parseText(response.getResponseText())	0.002s	PASSED
15	geoLat = Double.parseDouble(latitude.trim())	0.004s	PASSED
16	geoLon = Double.parseDouble(longitude.trim())	0.005s	PASSED
17	airpolutionLat = lat	0.004s	PASSED

TC_GetCurrentAirPollution

#	Description	Elapsed	Status
18	airpolutionLon = lon	0.002s	PASSED
19	tolerance = 0.001	0.004s	PASSED
20	assert Math.abs(geoLat - airpolutionLat) <= tolerance : "Latitude doesn't match"	0.003s	PASSED
21	assert Math.abs(geoLon - airpolutionLon) <= tolerance : "Longitude doesn't match"	0.004s	PASSED