package com.dont\_forget\_your\_umbrella.v2.data.api
import okhttp3.OkHttpClient
import okhttp3.Request
import org.json.JSONArray
import org.json.JSONObject
suspend fun main() {
val apiImpl = WeatherAPIImpl(OkHttpClient(),"IuUWPuoe40Ot6QYCYOG1Ve6PrleeZseD")
val key = apiImpl.getLocationKey("53.189999", "-2.890000")
val weatherResponse = apiImpl.getWeather(key)
}
class WeatherAPIImpl (
private val client: OkHttpClient,
private val apiKey: String) {
private fun assembleLatLongQuery(lat: String, long: String): String =
"?apikey=$apiKey&q=$lat%2C$long"
suspend fun getLocationKey(lat: String, long: String): String {
val locationUrl = locationKeyUrl + assembleLatLongQuery(lat, long)
val request = Request.Builder().url(locationUrl).build()
val response = client.newCall(request).execute().body()?.string()
val obj = JSONObject(response)
return obj.get("Key") as String
}
fun getWeather(locationKey: String): WeatherInstance {
val weatherUrl = weatherFromLocationUrl + locationKey + "?apikey=$apiKey"
val request = Request.Builder().url(weatherUrl).build()
val response = client.newCall(request).execute().body()?.string()
val obj = JSONObject(response)
val headline = obj.getJSONObject("Headline")
val category = headline.get("Category") as String
val effectiveDate = headline.get("EffectiveDate") as String
val text = headline.get("Text") as String
return WeatherInstance(category, effectiveDate, text)
}
companion object {
private const val baseUrl = "http://dataservice.accuweather.com/"
const val locationKeyUrl = "${baseUrl}locations/v1/cities/geoposition/search"
const val weatherFromLocationUrl = "${baseUrl}forecasts/v1/daily/1day/"
}
}package com.dont\_forget\_your\_umbrella.v2.data.api
/\*\*
\* A specific weather instance in time, such as rain today at 10am.
\*/
data class WeatherInstance(
val category : String,
val effectiveDate : String,
val text : String
)