Proyecto01: Web Service

Integrantes:
Arizmendi López Alexis de Jesús
Cortés Jiménez Carlos Daniel
Cruz Hérnandez Víctor Emiliano

16/03/2022.

1. Definición del problema

En este proyecto nos centramos en la elaboracion de un web service para el aeropuerto de la cuidad de México, en el cual se realizará un programa para consultar características, e informacion relevante de un punto A a otro punto B(**en este caso ciudades**) para ello nos apoyaremos sobre otro web service para conocer la inforacion requerida para la solucion del problema planteado, como lo es OpenWeatherMap que nos proposrciona datos metereorologicos historicos y actuales.

2. Análisis del problema

Empecemos por el informe del clima de la ciudad, para lograr esto como anterirormente se menciono utilizremos OpenWeatherMap para mostrar informe de la ciudad en la que se encuentra el cline o usuario y dezplegarla por pantalla caracteristicas como temperatura de la ciudad, la humedad, la presion, la longitud, latitud y el tipo de clima en la ciudad que se ecnuentra asi como tambien mostrar todas estas caracteristicas a la ciudad a la que se dirige dicha persona, un dato importante a resaltar es que el programa no es interactivo, esto quiere decir que, el usuario no podra manipular dicha aplicacion, si no que la aplicacion misma desplegara toda la informacion sin ser manipulada.
Ademas este despligue de informacion debera realizarse para 3000 tickes una vez que corra el programa, tambien deberemos contar con un archivo .csv donde se tenga el codigo IATA de las ciudades a las que se puede realizar un viaje, para esto se hara dos classes una que contenga la estrcutura para obtner el clima y otra para los vuelos(que contiene el codigo IATA, longitud y latitud de la ciudad origen y destino).
Otra dato a resaltar es que se debera realizar un cache para guardra todos aqueloos datos que sean recurrentes para asi acceder de forma mas rapida a ellos. Una vez se tengan hecho todo lo anterior dicho se motrara en terminal por medio de una tabla toda la informacion que debera contener nuestro ticket

3. Selección de la mejor alternativa

Para a resolucion de este proyecto decidimos usar Apache Maven ya que es un software enfocado a la gestion y manipulacion para la creacion de proyecto en java, asi como para facilitarnos el uso de librerias de teceros, y ademas ayuda mucho a simpleficar la creacion de codigo, y ademas nos ayuda a validar, compilar, "testear"(Pruebas unitarias), verificar y emepaquetar nuestro proyecto, dados esto puntos es por lo que nos decidimos por usarlo para la resolucion rapida y eficaz del proyecto.

4. Pseudocódigo

1.- Inicio

2.- rutaCSV = "src/main/resources/archivoBoletos/dataset1.csv";
3.- maximoNumeroPeticiones = 55;
4.- maxTiempoCache = (1000 * 60 * 60 * 5);
5.- diccionarioRegiones = new HashMap<String, Lugar>();
6.- listaVuelos = new HashMap<String, Vuelo>();
7.- diccionarioClimas = new HashMap¡String, Clima¿();

8.- meterADiccionarios(String sOrigen, String sDestino, String so_lat, String so_lon, String sd_lat, String sd_lon);

Inicializamos Lugar origen = null;
Inicializamos Lugar destino = null;

8.1.- Si diccionarioRegiones.containsKey(sOrigen)

origen = diccionarioRegiones.get(sOrigen);

De lo contario:

origen = new Lugar(sOrigen, so_lat, so_lon);
diccionarioRegiones.put(sOrigen, origen);

Clima clima = new Clima(sOrigen, so_lat, so_lon);
diccionarioClimas.put(sOrigen, clima);

8.1.1.- Si diccionarioRegiones.containsKey(sDestino)

destino = diccionarioRegiones.get(sDestino);

De lo contarrio haz:

destino = new Lugar(sOrigen, sd_lat, sd_lon);
diccionarioRegiones.put(sDestino, destino);
Clima clima = new Clima(sDestino, sd_lat, sd_lon);
diccionarioClimas.put(sDestino, clima);

Además

int id = Vuelo.getNuevoIdDeVuelo();
8.1.1.1.- Mostramos "id :"+id
Vuelo vuelo = new Vuelo(origen, destino), id;
listaVuelos.add(vuelo);

9.- llenarValoresDiccionarioClima()

9.1.- Mostramos: diccionarioClimas.size();
9.2.- Mostramos: Çargando peticiones...";

inicializamos contador = 0;

9.3.- Mientras ap.Entry<String,Clima> v: diccionarioClimas.entrySet() realizamos

9.3.1.- Intenta

9.3.1.1.- Si contador>0 and contador %maximoNumeroPeticiones==0

9.3.1.1.1.- Mostramos: "Hay más de "+maximoNumeroPeticiones+" peticiones, hay que esperar 1 min entre ellas.";

9.3.1.1.2.- Mostramos: ."Esperando para un bloque";
Thread.sleep(1000*60);

Ademas

v.getValue().llenarAtributos();
contador++;

9.3.2.- Atrapa InterruptedException e

9.3.2.1.- Mostramos "ERROR en Esperar peticiones";

9.4.- Mostramos: "Terminó"

10.- imprimirMenu()

10.1.- Mostramos: "**********VUELOS**********";

10.2.- Mientras Vuelo v: listaVuelos realizamos

String origen = v.getOrigen().getId();
String destino = v.getDestino().getId();

String temp_origen = diccionarioClimas.get(origen).getTemperatura();
String temp_destino = diccionarioClimas.get(destino).getTemperatura();

String clima_origen = diccionarioClimas.get(origen).getTipoClima();
String clima_destino = diccionarioClimas.get(destino).getTipoClima();

String descClima_origen = diccionarioClimas.get(origen).getDescripcionDelClima();
String descClima_destino = diccionarioClimas.get(destino).getDescripcionDelClima();

String presion_origen = diccionarioClimas.get(origen).getPresionA();
String presion_destino = diccionarioClimas.get(destino).getPresionA();

String humedad_origen = diccionarioClimas.get(origen).getHumedad();
String humedad_destino = diccionarioClimas.get(destino).getHumedad();

dataO = String.format(
" %-10s %-10s %-15s %-10s %-10s %-10s",
origen, clima_origen, temp_origen, descClima_origen, presion_origen, humedad_origen
);

dataD = String.format(
" %-10s %-10s %-15s %-10s %-10s %-10s",
destino, clima_destino, temp_destino, descClima_destino, presion_destino,
humedad_destino
);

10.3.- Mostramos: Vuelo de "+origen+" a "+destino+". Ticket: "+v.getId() ;

10.4.- Mostramos: "->Origen: "+dataO+"/n->Destino: "+dataD+"/n";

11.- main(String[] args)

Date ahora = new Date();
Date primeraEjecucionDelDia = EscritorLectorCache.leerCacheHoraEjecucionDelDia();
long tiempo24Horas = (1000 * 60 * 60 * 24);
long diferencia = 0;

11.1.- Si primeraEjecucionDelDia != null entonces

diferencia = ahora.getTime() - primeraEjecucionDelDia.getTime();

De lo contrario

diferencia = tiempo24Horas *2;

11.1.1.- Si diferencia >= tiempo24Horas

11.1.1.1.- Mostramos "Crea desde csv nuevo";

CSVReader reader = null;

11.1.1.1.- Intenta

reader = new CSVReader(new FileReader(rutaCSV.replaceAll(/",//")));
String[] linea = reader.readNext();

Mientras (linea = reader.readNext()) != null

sOrigen = linea[0];
sDestino = linea[1];
so_lat = linea[2];
so_lon = linea[3];
sd_lat = linea[4];
sd_lon = linea[5];

101.1.1.1.1- Mostramos "Mete algo"

meterADiccionarios(sOrigen, sDestino, so_lat, so_lon, sd_lat, sd_lon);

11.1.1.2.- Atrapa FileNotFoundException e

11.1.1.2.1.- Mostrams: .ErrorÖal leer CSV"

11.1.1.3.- Atrapa IOException ee

11.1.1.3.1.- Mostrams: ÏOException"

11.1.1.4.- Atrapa CsvValidationException eee

11.1.1.4.1.- Mostrams: ÇsvValidationExveption"

Además

llenarValoresDiccionarioClima();

EscritorLectorCache.escribirUltimaFechaEjecucion(ahora);
EscritorLectorCache.escribirFechaEjecucionDelDia(ahora);

EscritorLectorCache.escribirClimas(
(HashMap<String,Clima>) diccionarioClimas
);
EscritorLectorCache.escribirLugares(
(HashMap<String,Lugar>) diccionarioRegiones
);
EscritorLectorCache.escribirVuelos(
listaVuelos

);

11.1.1.4.2.- Mostramos "número de tickets: "+listaVuelos.size();
11.1.1.4.3.- Mostramos "número de climas: "+diccionarioClimas.size();
11.1.1.4.4.- Mostramos "número de regiones: "+diccionarioRegiones.size();
imprimirMenu();

De lo contario haz

Date horaActual = new Date();
Date horaAnterior = EscritorLectorCache.leerCacheUltimaHoraEjecucion();

long diferencia2 = horaActual.getTime() - horaAnterior.getTime();

listaVuelos = EscritorLectorCache.leerCacheVuelos();
diccionarioRegiones = EscritorLectorCache.leerCacheLugares();
diccionarioClimas = EscritorLectorCache.leerCacheClima();

11.2.- Si diferencia2 >= maxTiempoCache entonces

11.2.1.- Mostramos "más de "+maxTiempoCache+" nueva búsqueda"

llenarValoresDiccionarioClima();

EscritorLectorCache.escribirUltimaFechaEjecucion(horaActual);
EscritorLectorCache.escribirClimas(
(HashMap<String,Clima>) diccionarioClimas
);

imprimirMenu();

11.- Fin

5. Finalmente piensa a futuro

Se podrian mejorar varias muchos aspectos uno de ellos seria aumentar la capacidad de tickets que se tiene en primera instancia del problema, ya que esto podria provocar un saturamiento de la pagina y poder hacerla colapsar, otro punto seria poder hacerla interactiva algo con lo que el usuario o persona pueda sentirse mas comodo al usar nuestros servicios, implemnetar una interfaz grafica mas amigable para el usuario, tambien requerira a futuro mantenimiento para una version actualizada a dicho web service que se realizara, actualizacion de redes, implementar una pagina web.