PRÁCTICA 2 : Acceso a Datos XML

ACCESO A DATOS

DYLAN HURTADO LÓPEZ

Índice:

- ## Paquete model (pojos)

model.pojos_CSV_To_XML: Hay estan las clases pojos para leer los csv y meterlos a ArrayList

UtilMeasures: Clase que tiene un map con los tipos de mediciones y medidas

Station: Clase pojo con los atributos del csv de estaciones

Meteo: Clase pojo con los atributos del csv de meteoMes

model.pojosToRead_XML : Pojos para crear los objetos a partir de temperatura.xml

Temperatura , **Municipio** y **Medida**: Clases pojo para leer temperatura con JAXB.

model.pojosToWriteXML: estos no son utilizados pero queria usarlos para crear el esquema con JAXB

Resultado,**Resultados**: No usados

- Paquete io
    - io.csv

Clase : MeteoReader :lee el csv calidadAireMeteoMes y lo map a objetos y lo guarda en arraylist

ReadDataOfPath(): Lee su csv y lo mapea y asigna a clases pojo

Clase : StationReader: lee el csv estaciones y lo mapea a objetos también.

ReadDataOfPath(): Lee su csv y lo mapea y asigna a clases pojo

- io.xml

Clase : JaxbController : Controlador de JAXB usado para leer temperatura.xml

Usa un singleton para instanciarse

loadData: Se encarga de leer temperatura.xml con la ayuda de las clase pojos.

Clase: JdomController: Controlador de JDOM usado para crear los informes markdown y los XML.

Usa un singleton para instanciarse

generateXML: genera el xml en la ruta DATA_URI

xmlFormat: da formato y ordena los xml cuando se crean

initData: crea el Document e instancia el nodo root para crear el xml

getId: Obtiene el id a traves del nombre de la estacion

initDocumentCities: crea los Elementos municipio y sus id

initChildrens: inicializa los hijos de municipios

setCityData:inicializa dentro de cada medicion los maximos,minimos,medias,etc

loadDocument: carga el documento mediante el nombre

generarResultado: Usa una consulta de Xpath para obtener el elemento municipio gracias a su id

xmlSafer: salva el archivo en la ruta BD_URI

printFileInConsole: Imprime por pantalla los xml y el informe markdown

initMediciones: Inicializa los campos de mediciones

agregaMediciones: Metodo que permite agrandar el archivo xml con cada ejecucion

mostrarPantallaUltimoElemento:Muestra por pantalla el ultimo elemento guarda en mediciones.xml

loadMediciones: Carga mediciones.ml si existe lo crea y agrega y genera las mediciones

generaMediciones:metodo llamado en loadMediciones, generaMediciones

loadMD:Crea el informe MarkDown y lo guarda.

Clase : XpathController: Usado para ejecutar expresiones o consultas de Xpath
Usa un singleton para instanciarse

- **setExpresion**:Metodo para almacenar empresion de Xpath

  -**setDocument**: Inicializa el Document para guardarlo como variable local

  -**consultsXpath**: Obtiene consultas realizadas con Xpath.

- Paquete service

Facade : actua como fachada para organizar todo y no ensuciar el main
- Se usa un singleton para instanciarla

  -**mainRun**(): Metodo el cual valida los argumentos dados en el main y es el metodo desencadenante en la ejecucion desde la clase Main

  -**mainData**(): metodo que adjunta pruebas y hechos de lo que realizado en la práctica

- **ComprobarNombre**(): Comprueba si el primer argumento (nombre) es valido. Se hace referencia en mainRun().

MeteoUtil y StationUtil: Son las clases que tienen los filtros hechos con API Stream.

- **GetListDataMeasure**: Filtra para recibir una lista con el tipo de medida(Radicion solar, velocidad viento,etc).
- **GetDataAverageMinMax**():Devuelve una lista con los valores máximos, mínimos y la media.
- **getDataFechaFin** y **getDataFechaIncio**: Filtra las fechas.
- **getPuntoMuestreo** y **getPuntosMuestreo**: devuelven listas filtradas por puntos de muestreo.

- Paquete Main

Main: clase principal donde se ejecuta el programa solo contiene una instancia de la fachada y un método de esta misma.

Enlace al video Explicativo de la práctica:

https://drive.google.com/file/d/1vOg9Z43s3TEDpuf35nKasxoczGbcg8yZ/view?usp=sharing

Repositorio de Github:

https://github.com/DyLaNHurtado/ADPractica2XML

Certificados de los cursos de OpenWebinars:


OpenWebinars
CERTIFICA QUE
Dylan Hurtado López
HA SUPERADO CON ÉXITO
CURSO DE GIT
Manuel Agudo
CEO de OpenWebinars
openwebinars.net/cert/KnPI3
Duración del curso
5 horas
Fecha de expedición
7 de Noviembre de 2021