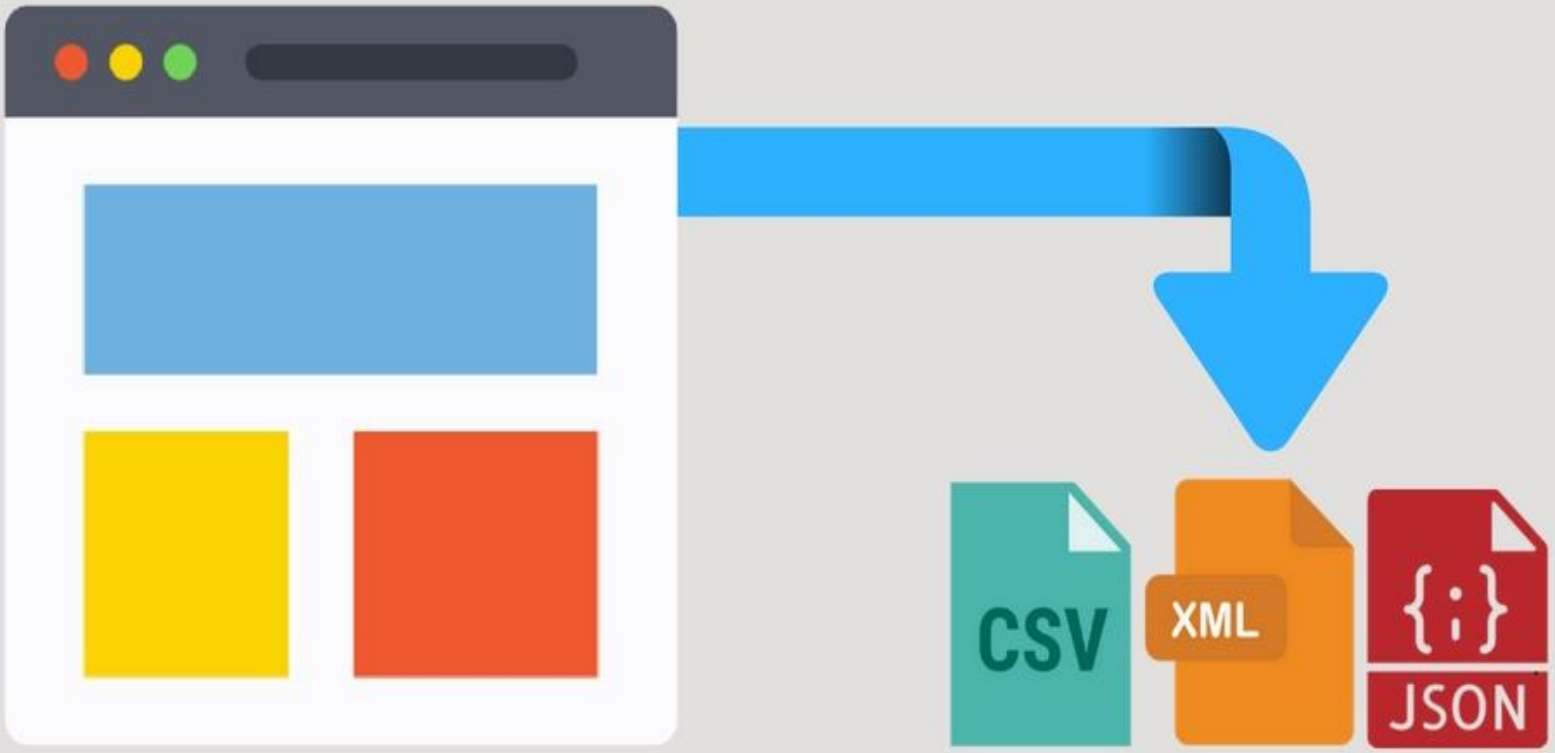
WEB SCRAPER

DESARROLLO DE APLICACIONES PARA CIENCIA DE DATOS

2º año | Grado en Ciencias e Ingeniería de Datos | Universidad de las Palmas de Gran Canaria



Índice

Resumen

En este trabajo además de lo pedido he incluido los conocimientos del proyecto anterior el del Spotify.

- **Clase "Main":** tiene cuatro métodos que actualizan la base de datos con la información correspondiente.
  Llama al método de la clase Scraper para obtener información, recorre la lista de elementos obtenidos y agrega cada uno de ellos a la base de datos mediante la llamada al método "add" en la clase "SqliteBookingDatabase".

Paquete webservice:

- **Clase "Scraper":** Es el scraper web para Booking.com es una clase que implementa una interfaz llamada "BookingSource". Utilizo la biblioteca JSoup para conectarme a la URL y parsear el HTML de la página extraer la información que se especifica en paquete model(Location, Rating, Comment, Service).
  El método "getLocations", "getRatings", "getComments" tienen una estructura similar: primero se obtiene el HTML de la página utilizando el método "getHtmlDocument", luego usa JSoup para seleccionar los elementos HTML relevantes y extraer la información necesaria. Finalmente se devuelven los objetos de "Location", "Rating" o "Comment" correspondientes a partir de información obtenida.
  El método "getServices" también tiene que extraer información diferente del HTML de la página, en este caso el nombre del grupo de servicios y la lista de servicios dentro de ese grupo. Por lo tanto, utiliza una lógica de selección y extracción de información ligeramente diferente a la utilizada por los otros métodos.

- **Clase "ScraperAPI":** es una clase que implementa una API REST con la biblioteca Spark que utiliza la clase "Scraper" para extraer información de una página web. La API expone cuatro rutas (/hotels/:name/locations, /hotels/:name/ratings, /hotels/:name/comments y /hotels/:name/services). Cada una de estas rutas permite obtener información de una página web dada su URL. Cuando se inicia la aplicación, se establece un servidor en el puerto 4567 y se especifica que la ubicación de los archivos estáticos es la carpeta "public". Luego se configuran las rutas de la API utilizando el método get y se proporciona una implementación de la clase abstracta "ScraperCommand" para manejar la solicitud de cada ruta.
  La clase abstracta ScraperCommand se encarga de verificar si se ha proporcionado un nombre de hotel válido y de generar la URL de la página web. Luego llama al método abstracto "handleScraper" para que sea implementado por las clases que heredan de ella.
  Las clases "GetLocationsCommand", "GetRatingsCommand", "GetCommentsCommand" y "GetServicesCommand" son clases que heredan de

"ScraperCommand" y proporcionan implementaciones específicas para cada una de las rutas.

- **Interfaz "BookingSource":** se utiliza en la clase "ScraperAPI" para permitir que cada una de las rutas de la API pueda utilizar cualquiera de los métodos de extracción de información disponibles en la clase "Scraper".

Paquete model: todas las clases de este paquete tienen un método toString que utiliza la clase Gson para convertir la información del objeto en una cadena en formato JSON.para devolver la información en una respuesta HTTP.

- **Clase "Location":** representa información sobre la ubicación, contiene dos atributos públicos (name y location). La clase proporciona un constructor que acepta dos parámetros y los utiliza para inicializar los atributos de la clase. También proporciona los métodos "getName" y "getLocation".

- **Clase "Rating":** representa información sobre la calificación, contiene dos atributos públicos (name y rating). La clase proporciona un constructor que acepta dos parámetros y los utiliza para inicializar los atributos de la clase. También proporciona los métodos "getName" y "getRating".

- **Clase "Service":** representa información sobre el servicio, contiene dos atributos públicos (name y service). La clase proporciona un constructor que acepta dos parámetros y los utiliza para inicializar los atributos de la clase. También proporciona los métodos "getName" y "getService".

- **Clase "Comment":** representa información sobre los comentarios, contiene siete atributos públicos (name, country, punctuation, review, positive, negative, days). La clase proporciona un constructor que acepta siete parámetros y los utiliza para inicializar los atributos de la clase. También proporciona los métodos "getName", "getCountry". "getPunctuation", "getReview", "getPositive", "getNegative", "getDays".

Paquete database:

- **Clase "DMLTranslater":** proporciona métodos estáticos para crear instrucciones de inserción de SQL para cada una de las clases del paquete model. Cada uno de estos métodos recibe un objeto del tipo correspondiente y devuelve una cadena que contiene una instrucción de inserción de SQL para insertar la información de ese objeto en una tabla de base de datos.
  El método "insertStatementOf()" recibe un objeto y devuelve una cadena que contiene una instrucción de inserción de SQL para insertar la información de ese objeto en la tabla correspondiente. La instrucción de inserción utiliza el método

String.format para reemplazar los placeholders '%s' en la plantilla de la instrucción de inserción con los valores de los atributos de la clase.

- ➔ **Clase "SqliteBookingDatabase":** es una implementación de la interfaz "BookingDatabase", se encarga de almacenar información del hotel en una base de datos SQLite. La base de datos se inicializa en el constructor de la clase con la creación de cuatro tablas. Cada una de estas tablas almacena información diferente sobre el hotel. Cada método add toma un objeto de un tipo específico y utiliza la clase "DMLTranslater" para crear una sentencia INSERT adecuada para insertar ese objeto en la base de datos.
- ➔ **Interfaz "BookingDataBase":** permite mediante al add añadir objetos a la base de datos.

Líneas futuras

Para comercializar mi producto podría ofrecer desde servicios de recopilación de datos a empresas e individuos interesados en obtener cierta información, hasta integrarlo en un en un servicio o producto más amplio.

Por ejemplo: integrar el scraper en un sistema de análisis de datos o en una plataforma de monitoreo de redes sociales, para proporcionar información valiosa a los usuarios de esos servicios.

Recursos utilizados

- IntelliJ
- Word

Conclusiones

Este proyecto ha sido una excelente oportunidad para poner en práctica los conocimientos adquiridos en esta materia de programación y base de datos. Ha sido un reto para mí el desarrollar una herramienta de web scraper y la integración con una base de datos, pero al final he podido superar las dificultades y completar el proyecto con éxito. Además, este trabajo me ha permitido profundizar en el uso de herramientas como SQLite y JSON, lo que me ha brindado una mayor confianza en mi capacidad de aplicar estos conocimientos en futuros proyectos. En resumen, este proyecto ha sido una experiencia muy valiosa y me ha permitido adquirir habilidades fundamentales para mi desarrollo como programadora.

Bibliografía

https://www.scrapingbee.com/blog/

https://mvnrepository.com/artifact/org.xerial/sqlite-jdbc/3.39.3.0

https://www.sqlitetutorial.net/sqlite-java/sqlite-jdbc-driver/

https://www.sqlitetutorial.net/sqlite-java/create-table/

https://www.sqlitetutorial.net/sqlite-java/insert/