DOCUMENTO TEST:

GuardianContentApiTest contiene i test del client java reperito dal seguente link.

ToolsTest effettua il test dei metodi della classe Tools escludendo quelli di utility (ovvero i metodi di interazione con l'utente per richiedere che operazione svolgere). I metodi sono divisi in 4 sezioni principali:

- Serializzatore
  - testSerializer
- Deserializzatore
  - testGenericDeserializer
- Download
  - testDownload
- Words Counter
  - testWordsCounter
  - testExtraction

testSerializer:
Test per il metodo Serializer() il quale verifica la corretta creazione di un file .json. Il test effettua una richiesta all'API TheGuardian utilizzando una parola chiave di test ("bitcoin") e ottiene una risposta. Viene successivamente invocato il metodo Serializer() per salvare la risposta in un file .json. Il test verifica che il file sia stato creato correttamente nel percorso specificato ed infine elimina il file creato per pulire l'ambiente di test

testGenericDeserializer:

Test per il metodo GenericDeserializer() il quale verifica la corretta deserializzazione di file .json e .csv. Il test esegue la deserializzazione dei file presenti nelle directory specificate per i file .json e .csv. Per entrambe le directory dei file, il test verifica che il risultato della deserializzazione non sia nullo e che contenga almeno un elemento

testDownload:

Test per il metodo Download() che verifica il corretto download dei file .json e .csv. Il test simula l'input e verifica che i file vengano scaricati correttamente. Per il download dei file .json: Viene impostato l'input di test per la fonte "J" (.json). Prima del download, viene creato un array di file nella directory "src/main/json_source". Viene eseguito il metodo Download() per scaricare i file .json. Dopo il download, viene creato un nuovo array di file nella directory "src/main/json_source". Viene verificato che la lunghezza dell'array dopo il download sia diversa da quella precedente. Viene controllato che il primo file nell'array dopo il download sia diverso dal primo file nell'array precedente, confermando così che un file è stato sovrascritto o creato durante il download. Se un nuovo file è stato creato durante il download, viene eliminato per ripristinare lo stato precedente. Se nessun nuovo file è stato creato, viene verificato che il primo file nell'array

dopo il download sia uguale al primo file nell'array precedente, confermando che non ci sono stati cambiamenti. Per il download dei file .csv: Viene creato un array di file nella directory "src/main/csv_source" come input di partenza. Viene eseguito il metodo Download() per scaricare i file .csv. Viene creato un array di file nella directory "src/main/csv_to_json" dopo il download. Viene verificato che la lunghezza dell'array dei file di output sia uguale a quella dell'array di input, confermando che non ci sono stati cambiamenti nel numero di file. Se l'array di file di output iniziale è vuoto, vengono eliminati tutti i file nella directory "src/main/csv_to_json" per ripristinare lo stato precedente

testExtraction:

Metodo di test per la funzione Extraction(), che verifica l'estrazione di parole chiave dai file .json. Verifica se il metodo di Extraction funziona correttamente analizzando la directory di input e controllando se l'output contiene la stringa desiderata..
Il test esegue i seguenti passaggi:

- Verifica se il file "output.txt" esiste nella directory corrente.
- Se il file esiste, ne effettua una copia in "temp.txt".
- Prepara l'input specificando la directory di origine, ovvero "src/main/json_source".
- Reindirizza lo stream di output per catturare l'output del metodo Extraction.
- Esegue il metodo di Extraction.
- Aggiorna la lista dei file nella directory corrente.
- Ripristina lo stream di output originale.
- Ottiene l'output del metodo Extraction.
- Verifica se l'output contiene la stringa desiderata.
- Elimina il file "output.txt" se presente.
- Rinomina il file "temp.txt" in "output.txt" se il file originale esisteva in precedenza

testWordsCounter:

Metodo di test per la funzione WordsCounter() della classe Tools per verificare se conta correttamente il numero di occorrenze delle parole in un testo. Utilizza un testo di input predefinito e confronta il risultato atteso con il risultato ottenuto.
Se i due risultati corrispondono, il test viene considerato superato, altrimenti fallito