Aplikacja dla programistów

JSON TOOLS

Opis:

Aplikacja dla programistów, którzy potrzebują przeformatować lub filtrować struktury danych zapisane w formacie JSON a także porównać ze soba struktury. JSON tools pozwala zarówno na zminifikowanie niezminifikowanej reprezentacji JSON, a także na operację odwrotną.

Struktura:

pl.put.poznan.transformer.rest:
zawiera kontrolery REST odpowiedzialne za obsługę przychodzących żądań HTTP.

- TextTransformerController.java

Jest to kontroler do transformacji tekstu.

- TransformerController.java

Jest to kontroler do operacji na JSON.

pl.put.poznan.transformer.logic:
Ten pakiet zawiera klasy odpowiedzialne za transformację, dekorację i porównywanie JSON oraz tekstu.

pl.put.poznan.transformer.app:

- TextTransformerApplication.java

Jest to główna klasa aplikacji odpowiedzialna za uruchamianie aplikacji Spring Boot.
Przeszukuje ona pakiet pl.put.poznan.transformer.rest w poszukiwaniu komponentów Spring.
W jej skład wchodzą:

- Filter.java

Filter to klasa dekoratora implementująca interfejs JSONTransformer. Dodaje funkcję filtrowania wybranych pól w strukturze JSON.
Konstruktor:
Filter(JSONTransformer json, String[] parameters): Przyjmuje obiekt do dekoracji (implementujący JSONTransformer) oraz tablicę parametrów do filtrowania.
Metoda:
decorate(String text): Metoda dekorująca, która filtruje wybrane pola w strukturze JSON.
Metody prywatne:
filterString(String text, String[] filterParameter): Faktyczne filtrowanie na obiekcie JSON.
filterNodeRecursive(JsonNode jsonNode, String[] filter, List<JsonNode> forRemoval, ArrayList<String> forRemovalName): Metoda rekurencyjna używana do filtrowania węzłów w strukturze JSON.

- FilterOnly.java

FilterOnly to klasa dekoratora implementująca interfejs JSONTransformer. Oferuje funkcję filtrowania, ale zachowuje jedynie węzły, które zawierają przekazane filtry.
Konstruktor:
FilterOnly(JSONTransformer json, String[] filterParameter): Przyjmuje obiekt do dekoracji (implementujący JSONTransformer) oraz tablicę parametrów do filtrowania.
Metoda:
decorate(String text): Metoda dekorująca, która filtruje i zwraca jedynie węzły zawierające przekazane filtry.
Metody prywatne:
filterOutString(String text, String[] filterParameter): Faktyczne filtrowanie na obiekcie JSON.
filterNodeRecursive(JsonNode jsonNode, String[] filter, List<JsonNode> forRemoval, ArrayList<String> forRemovalName): Metoda rekurencyjna używana do

filtrowania węzłów w strukturze JSON.

- JSON.java

JSON reprezentuje strukturę JSON i umożliwia operacje na niej.
Konstruktor:
JSON(String jsonString): Przyjmuje ciąg znaków JSON i tworzy odpowiadający mu obiekt.
Metody:
get(): Zwraca wewnętrzną reprezentację JSON jako JsonNode.
getString() throws JsonProcessingException: Zwraca ciąg znaków JSON w formie przetworzonej.

- JSONTransformer.java

JSONTransformer to interfejs, który jest implementowany przez klasy realizujące różne transformacje.

- JsonComparator.java

JsonComparator oferuje funkcję porównywania dwóch plików JSON i wyświetlania różnic między nimi.

- JsonDecorator.java

JsonDecorator to klasa abstrakcyjna dekoratora implementująca interfejs JSONTransformer. Służy do tworzenia łańcucha dekoratorów.

- JsonDecoratorBuilder.java

JsonDecoratorBuilder jest klasą odpowiedzialną za tworzenie różnych dekoratorów w zależności od podanych parametrów.
Metoda:
getDecorator(String format, String[] filterParameter, String[] filterOnlyParameter): Metoda przyjmuje parametry takie jak format, parametry do filtrowania, oraz parametry do filtrowania jedynie określonych pól. W zależności od tych parametrów, tworzony jest odpowiedni łańcuch dekoratorów i zwracany jako obiekt implementujący interfejs JSONTransformer.

- MinifyDecorator.java oraz PrettifyDecorator.java

MinifyDecorator i PrettifyDecorator są klasami dekoratorów, które odpowiednio minimalizują i "upiększają" przekazany tekst JSON.
Konstruktor:
MinifyDecorator(JSONTransformer json), PrettifyDecorator(JSONTransformer json): Konstruktory przyjmują obiekt implementujący JSONTransformer jako argument.
Metoda:
decorate(String jsonString): Metoda dekorująca, która przekształca przekazany tekst JSON w zależności od rodzaju dekoratora.
Metody prywatne:
minify(String jsonString), prettify(String jsonString): Metody prywatne, które faktycznie wykonują minimalizację i "upiększanie" tekstu JSON.

- TextTransformer.java

TextTransformer jest przykładową klasą logiczną, która ilustruje, że logika przekształceń powinna być umieszczona poza warstwą serwisu REST.
Konstruktor:
TextTransformer(String[] transforms): Konstruktor przyjmuje tablicę transformacji.
Metoda:
transform(String text): Metoda, która implementuje przykładową transformację tekstu. W tym przypadku, zamienia tekst na jego dużą literę. Oczywiście w praktyce ta metoda powinna implementować rzeczywistą logikę na podstawie dostarczonych transformacji.

Instalacja:

Aby zbudować i uruchomić projekt, wykonaj metodę main w klasie TextTransformerApplication.java.

Funkcje:

Program umożliwia:
-zdalne korzystanie z dostępnych funkcji poprzez REST
-minifikację struktury w formacie JSON na podstawie pełnego zapisu w tym formacie w celu zmniejszenia rozmiaru danych
-uzyskanie pełnej struktury w formacie JSON ze zminifikowanego zapisu w celu polepszenia czytelności
-uzyskanie struktury zawierającej tylko określone własności w celu uproszczenia struktury
-porównywanie dwóch tekstów

Autorzy:

Józef Godlewski
Alicja Lis
Krzysztof Matyla
Szymon Haj
Michał Kwaśniowski