

Stwórz REST API w ASP.NET CORE 6.0 służące do wyszukiwania słów w dokumencie TXT znajdujących się w pobliżu zdefiniowanych słów kluczowych.

1. Powinna istnieć możliwość zdefiniowania przez jedną z metod Rest API profilu wyszukiwania, który powinien zawierać możliwe słowa kluczowe w pobliżu, których będzie odczytywana szukana wartość. Możesz przyjąć, że w body zapytania dla tej metody otrzymujesz dane w formacie JSON np.  
["Autor", "Autor książki", "pisarz", "autorem książki jest"]

Każdy profil szukania powinien posiadać unikalny identyfikator. Po którym można się do niego odwołać. Profile wyszukiwania możesz trzymać w pamięci - **nie ma potrzeby** zapisywania ich np. do bazy danych.

2. Druga metoda RestApi powinna przyjmować **plik** txt (patrz przykład poniżej), w którym będą wyszukiwane wartości. Dodatkowo powinna otrzymać jeszcze ID profilu wyszukiwania, żeby wiedzieć z jakich słów kluczowych skorzystać oraz np. w formie query string parametryzacje:
  - **direction** - która przyjmie wartości right, left, top lub bottom i od niej będzie zależało jakie słowa na lewo, prawo, nad czy pod spodem względem słów kluczowych będą zwrócone. np. left oznacza, że aplikacja ma zwrócić słowa na lewo od znalezionych słów kluczowych.
  - **maxWordsCount** - maksymalna liczba słów jaką metoda zwróci
  - **ignoreCase** - która przyjmie wartości true, false i będzie sprawdzała czy aplikacja ma zwracać uwagę na wielkość liter przy porównywaniu słów.

Metoda powinna zwrócić odpowiedź na żądanie w formie JSON zawierającą tablicę stringów z wszystkimi znalezionymi wartościami w dokumencie:

np. ["Adam Mickiewicz", "Bolesław Prus", "Eliza Orzeszkowa"]

#### Przykład:

Dla profilu wyszukiwania: ["Autorem jest", "napisane przez"]

I otrzymanego tekstu:

"Oda do młodości", "Pan Tadeusz", "Dziady" to tylko niektóre dzieła napisane przez Adam Mickiewicz polskiego poetę, działacza politycznego, publicystę i tłumacza. Inne powieści jakie powstały w okresie romantyzmu to "Kordian" i "Balladyna".  
Ich autorem jest Juliusz Słowacki.

- a. Dla direction=right, maxWordsCount=2 i ignoreCase=true metoda powinna zwrócić: ["Adam Mickiewicz", "Juliusz Słowacki."]
- b. Dla direction=right, maxWordsCount=2 i ignoreCase=false metoda powinna zwrócić: ["Adam Mickiewicz"]
- c. Dla direction=right, maxWordsCount=1 i ignoreCase=true metoda powinna zwrócić: ["Adam", "Juliusz"]
- d. Dla direction=left, maxWordsCount=1 i ignoreCase=true metoda powinna zwrócić: ["dzieła", "Ich"]
- e. Dla direction=top, maxWordsCount=2 i ignoreCase=true metoda powinna zwrócić: ["jakie powstały"]
- f. Dla direction=bottom, maxWordsCount=2 i ignoreCase=true metoda powinna zwrócić: ["tłumacza. Innym"]

**Uwaga:**

Dla kierunków left i right obsłuż zawijanie tekstu.

Dla wyszukiwania top i bottom załóż, że szerokość wszystkich znaków w tekście jest sobie równa (tz. monospace). I jeżeli choć jedna litera słowa znajduje się pod słowem poniżej drugiego to te słowa sąsiadują ze sobą:

np. Te słowa ze sobą sąsiadują w pionie:

		A	L	A
K	O	T		

	A	L	A	
	K	O	T	

A	L	A		
		K	O	T

A te nie:

			A	L	A
K	O	T			

Rozwiązanie wrzuć na publiczne repozytorium git i prześlij do niego link  
[rekrutacja@e-msi.pl](mailto:rekrutacja@e-msi.pl)

Postaraj się prowadzić ten projekt zgodnie z najlepszymi praktykami.

Powodzenia!