

# Solar

A stylized sun logo consisting of a red circle with white rays emanating from the center, positioned behind the letter 'r' in the word 'Solar'.

## Mikroprojekt

# Co w module

- Budujemy API do przeszukiwania artykułów CNN
- Podsumowujemy całą wiedzę z kursu!



# Założenia

- Mamy zbudować w Solr kolekcję, która będzie przechowywać artykuły z CNN.
- Dodatkowo budujemy aplikację, która pozwoli na przeszukiwanie kolekcji poprzez API, po HTTP.

# Z czego korzystamy

- **Dane:** artykuły CNN z Kaggle (<https://www.kaggle.com/datasets/hadasu92/cnn-articles-after-basic-cleaning>)
- **Aplikacja:** Java, Springboot.
- **Czym będziemy wysyłać requesty:** Postman
- **To będzie POC.** Zachęcam do rozbudowy!

# Zapytania

- Przeszukiwanie po konkretnych słowach kluczowych lub frazach
- Możliwość wybrania dodatkowo: *autora*, *kategorii* - jako obligatoryjnych pól, spośród których chcemy wybierać
- Możliwość dodania statystyk do zapytania (po kategorii, autorze)
- Możliwość uwzględnienia literówek

# Plan tworzenia

- **Eksploracja czystych danych**
- **Budowanie schematu.** Pomożemy sobie schema designerem.
- **Tworzenie kolekcji** i indeksacja danych.
- **Tworzenie** i testowanie **zapytań** (query)
- **Budowa biblioteki** do przeszukiwania oraz **API**

# Aplikacje Webowe (Przeglądarkowe)

# Aplikacja webowa

- Aplikacje, które mogą działać w przeglądarkach
- Backend
- Frontend
- API





**BACK-END**

**FRONT-END**

**APIs**

[:funny.com](http://funny.com)



**Backend Programmer**



**Frontend Programmer**



**FRONTEND VS BACKEND**



**WHEN FRONTEND IS  
READY BEFORE BACKEND**

# Springboot

- Technologia, która pozwala (między innymi) budować aplikacje webowe
- W naszym przypadku - będziemy tworzyć bardzo proste API, które będzie obsługiwało requesty typu GET.
- Do wysyłania samych requestów użyjemy Postmana (choć można użyć także zwykłej przeglądarki)

**A teraz...**  
**Do roboty!;-)**