

1. Instalacja Ubuntu wersja 24.04

- Pobierz obraz instalacyjny Ubuntu 24.04 z oficjalnej strony: <https://ubuntu.com/download>.
- Skonfiguruj maszynę wirtualną w VirtualBox lub Vmware, aby uruchomić obraz ISO.
- Przeprowadź instalację systemu, konfigurując użytkownika, hasło oraz pozostałe ustawienia.

2. Instalacja potrzebnych zależności(OpenJDK 17, Maven, Docker, IDE np. IntelliJ)

- Instalacja OpenJDK 17:

W terminalu wykonaj polecenie :

```
sudo apt install openjdk-17-jdk -y
```

Następnie zweryfikuj czy instalacja się powiodła wpisując:

```
java -version
```

- Instalacja Maven:

W terminalu wykonaj polecenie :

```
sudo apt install maven -y
```

Następnie zweryfikuj czy instalacja się powiodła wpisując:

```
mvn -version
```

- Instalacja Dockera:

W terminalu wykonaj polecenie :

```
sudo apt install docker.io -y  
sudo systemctl start docker  
sudo systemctl enable docker
```

Następnie zweryfikuj czy instalacja się powiodła wpisując:

```
docker --version
```

- Instalacja IDE

Możesz pobrać i zainstalować IntelliJ IDEA Community Edition, które jest bezpłatne. Pobierz ze strony <https://www.jetbrains.com/idea/download>. Następnie zainstaluj i skonfiguruj środowisko zgodnie z wymaganiami.

3. Utworzenie projektu Spring Boot za pomocą Spring Initializr

- Ustaw parametry projektu:
 - 1) **Project:** *Maven Project*
 - 2) **Language:** *Java*
 - 3) **Spring Boot Version:** *3.3.5*
 - 4) **Group:** *com.pbs.edu*
 - 5) **Artifact:** *WeatherDataService*
 - 6) **Packaging:** *Jar*
 - 7) **Java:** *17*
- Dodanie zależności:
 - 1) Spring Web
 - 2) Spring Cloud Gateway
 - 3) Spring Boot DevTools
 - 4) Spring Data JPA
 - 5) H2 Database
 - 6) Lombok
- Pobierz projekt i rozpakuj plik ZIP, a następnie otwórz go w wybranym IDE (np. IntelliJ IDEA).

4. Konfiguracja mikroservisu do pobierania danych np. z OpenWeatherMap API.

- Przejdź do folderu *src/main/resources* i otwórz plik *application.properties*.
- Dodaj klucz API:

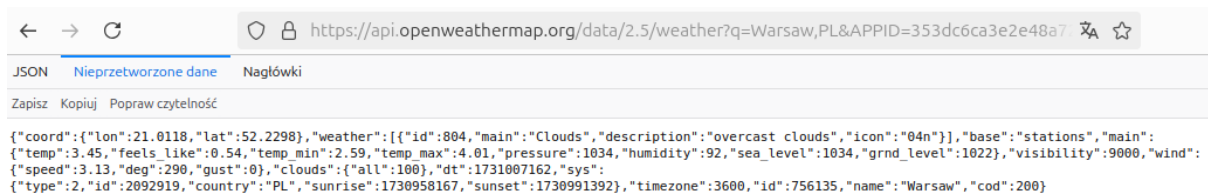
```
weather.api.key=353dc6ca3e2e48a7296714f30d5f88c4
```

- Dodaj url API:

```
weather.api.url=https://api.openweathermap.org/data/2.5/weather
```

5. Utwórz klasę WeatherData i zdefiniuj zmienne

- W katalogu *src/main/java/com/pbs/edu/WeatherDataService/* utwórz nową klasę o nazwie *WeatherData*
- Dodaj odpowiednie zmienne, wiedząc, że struktura odpowiedzi z API wygląda następująco:



```
{
  "coord": {
    "lon": 21.0118,
    "lat": 52.2298
  },
  "weather": [
    {
      "id": 804,
      "main": "Clouds",
      "description": "overcast clouds",
      "icon": "04n"
    }
  ],
  "base": "stations",
  "main": {
    "temp": 3.45,
    "feels_like": 0.54,
    "temp_min": 2.59,
    "temp_max": 4.01,
    "pressure": 1034,
    "humidity": 92,
    "sea_level": 1034,
    "grnd_level": 1022,
    "visibility": 9000,
    "wind": {
      "speed": 3.13,
      "deg": 290,
      "gust": 0
    },
    "clouds": {
      "all": 100
    },
    "dt": 1731007162,
    "sys": {
      "type": 2,
      "id": 2092919,
      "country": "PL",
      "sunrise": 1730958167,
      "sunset": 1730991392,
      "timezone": 3600,
      "id": 756135,
      "name": "Warsaw",
      "cod": 200
    }
  }
}
```

6. Utwórz serwis WeatherService do pobierania danych z wybranego API.

- W folderze *src/main/java/com/pbs/edu/WeatherDataService/* utwórz klasę *WeatherService*.

- Wewnątrz klasy stwórz metodę *getWeatherData* :

```
public WeatherData getWeatherData(String city, String countryCode) {
    // Logika do pobierania danych z API
    // Przypisanie wyników do obiektu WeatherData
}
```

- Użyj biblioteki *RestTemplate* z Springa do komunikacji z API.

7. Utwórz kontroler *WeatherController* w którym pod adresem */weather* z parametrem *city* wyświetli pogodę w wybranym mieście

- W katalogu *src/main/java/com/pbs/edu/WeatherDataService/* utwórz klasę *WeatherController*.
- Dodaj metodę obsługującą zapytanie HTTP GET:

```
@RestController
public class WeatherController {
    @Autowired
    private final WeatherService weatherService;

    @GetMapping(value = "/weather", produces = "application/json")
    public WeatherData getWeather(@RequestParam String city, @RequestParam String
        countryCode) {
        return weatherService.getWeatherData(city, countryCode);
    }
}
```

8. Uruchom oraz przetestuj lokalnie aplikację.

- Uruchom aplikację Spring Boot np. używając komendy:

```
./mvn spring-boot:run
```

- W przeglądarce wpisz adres <http://localhost:8080/weather?city=Warsaw&countryCode=PL>

9. Utwórz plik *docker file*, a następnie zbuduj obraz Dockera.

- W folderze głównym projektu utwórz plik *Dockerfile* o następującej zawartości:

```
FROM openjdk:17-jdk-slim
WORKDIR /app
COPY target/WeatherDataService-0.0.1-SNAPSHOT.jar app.jar
```

```
ENTRYPOINT ["java", "-jar", "app.jar"]
```

- Zbuduj obraz dockera:

```
sudo docker build -t weather-data-service .
```

10. Uruchom mikroservis w kontenerze.

```
sudo docker run -p 8080:8080 weather-data-service
```

11. Przetestuj działanie mikroservisu w kontenerze uruchamiając przeglądarkę i wpisując adres <http://localhost:8080/weather?city=Warsaw&countryCode=PL>