

# Aplikacje Internetowe

Sprawozdanie – Lab 5 – REST API Client – OpenWeatherMap.org

Data laboratorium: 04.12.2021

Autor: Lewicki Maciej

Grupa: 42

Wersja: 2021/2022-zima

## Wprowadzenie

*Tutaj należy umieścić 2-3 zdania opisujące zawartość sprawozdania i streszczające przebieg laboratorium. Wypełnienie tej rubryki warunkuje dalsze sprawdzenie sprawozdania.*

Zadaniem było stworzyć stronę która korzystając z openweathermap.org API pobiera dane na temat pogody i wyświetla je użytkownikowi.

## Strona HTML ze wszystkimi elementami

*Należy przedstawić HTML ze wszystkimi wymaganymi elementami. 1 punkt.*

# Rest Weather

Szczecin, Poland

Check!

13.12.2021 18:42:05



**3.12 °C**

Odczuwalna: 0.51 °C  
zachmurzenie duże

13.12.2021 19:00:00



**3.12 °C**

Odczuwalna: 1.51 °C  
zachmurzenie duże

13.12.2021 22:00:00



**3.38 °C**

Odczuwalna: 1.29 °C  
zachmurzenie duże

14.12.2021 01:00:00



**3.93 °C**

Odczuwalna: 1.73 °C  
zachmurzenie duże

14.12.2021 04:00:00



**5.32 °C**

Odczuwalna: 2.8 °C  
zachmurzenie duże

14.12.2021 07:00:00



**5.17 °C**

Odczuwalna: 2.71 °C  
zachmurzenie duże

14.12.2021 10:00:00



**4.84 °C**

Odczuwalna: 2.13 °C  
zachmurzenie duże

14.12.2021 13:00:00



**6.18 °C**

Odczuwalna: 3.59 °C  
zachmurzenie duże

14.12.2021 16:00:00

14.12.2021 19:00:00

```

<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <title>Rest Weather</title>
  <link rel="stylesheet" href="rest-weather.css" type="text/css">
  <script defer src="rest-weather.js" type="text/javascript"></script>
</head>
<body>

  <div id="weather-search-params">
    <h1>Rest Weather</h1>
    <input type="text" value="Szczecin, Poland" id="locationInput">
    <button id="checkButton">Check!</button>
  </div>

  <div id="wrapper" class="gradient-border">
    <div id="weather-results-container"></div>
  </div>

</body>
</html>

```

## Żądanie XMLHttpRequest pod endpoint current

Należy udokumentować pobieranie danych z endpointu current z wykorzystaniem tradycyjnego XMLHttpRequest. Wyświetlać dane z wykorzystaniem console.log(). 1 punkt.

```

getCurrentWeather(query) {
  let url = this.currentWeatherLink.replace("{query}", query);
  let req = new XMLHttpRequest();
  req.open("GET", url, true);
  req.addEventListener("load", () => {
    this.currentWeather = JSON.parse(req.responseText);
    console.log(this.currentWeather);
    this.drawWeather();
  });
  req.send();
}

```

## Żądanie Fetch API pod endpoint forecast

Należy udokumentować pobieranie danych z endpointu forecast z wykorzystaniem nowoczesnego FETCH API. Wyświetlać dane z wykorzystaniem console.log(). 1 punkt.

```

getForecast(query) {
  let url = this.forecastLink.replace("{query}", query);
  fetch(url).then((response) => {
    return response.json();
  }).then((data) => {
    console.log(data);
    this.forecast = data.list;
    this.drawWeather();
  });
}

```

...

## Wizualizacja otrzymanych zwrotek

Należy udokumentować funkcjonalność wyświetlania uzyskanych zwrotek poprzez modyfikację drzewa DOM. 1 punkt.

```
drawWeather() {
  // clear previous blocks
  this.resultsBlock.innerHTML = '';

  // add current weather block
  if (this.currentWeather) {
    const date = new Date(this.currentWeather.dt * 1000);
    const dateTimeString = `${date.toLocaleDateString("pl-PL")} ${date.toLocaleTimeString("pl-PL")}`;

    const temperature = this.currentWeather.main.temp;
    const feelsLikeTemperature = this.currentWeather.main.feels_like;
    const iconName = this.currentWeather.weather[0].icon;
    const description = this.currentWeather.weather[0].description;

    const weatherBlock = this.createWeatherBlock(dateTimeString, temperature, feelsLikeTemperature, iconName, description);
    this.resultsBlock.appendChild(weatherBlock);
  }

  // add forecast weather blocks
  if (this.forecast && this.forecast.length > 0) {
    for (let i = 0; i < this.forecast.length; i++) {
      let weather = this.forecast[i];
      const date = new Date(weather.dt * 1000);
      const dateTimeString = `${date.toLocaleDateString("pl-PL")} ${date.toLocaleTimeString("pl-PL")}`;

      const temperature = weather.main.temp;
      const feelsLikeTemperature = weather.main.feels_like;
      const iconName = weather.weather[0].icon;
      const description = weather.weather[0].description;

      const weatherBlock = this.createWeatherBlock(dateTimeString, temperature, feelsLikeTemperature, iconName, description);
      this.resultsBlock.appendChild(weatherBlock);
    }
  }
}

createWeatherBlock(dateString, temperature, feelsLikeTemperature, iconName, description) {
  let weatherBlock = document.createElement("div");
  weatherBlock.className = "weather-block gradient-border";

  let dateBlock = document.createElement("div");
  dateBlock.className = "weather-date";
  dateBlock.innerText = dateString;
  weatherBlock.appendChild(dateBlock);

  let temperatureBlock = document.createElement("div");
  temperatureBlock.className = "weather-temperature";
  temperatureBlock.innerHTML = `${temperature} &deg;C`;
  weatherBlock.appendChild(temperatureBlock);

  let feelsLikeBlock = document.createElement("div");
  feelsLikeBlock.className = "weather-temperature-feels-like";
  feelsLikeBlock.innerHTML = `Odczuwalna: ${feelsLikeTemperature} &deg;C`;
  weatherBlock.appendChild(feelsLikeBlock);

  let weatherIcon = document.createElement("img");
  weatherIcon.className = "weather-icon";
  weatherIcon.src = this.iconLink.replace("{iconName}", iconName);
  weatherBlock.appendChild(weatherIcon);

  let weatherDescription = document.createElement("div");
  weatherDescription.className = "weather-description";
  weatherDescription.innerText = description;
  weatherBlock.appendChild(weatherDescription);

  return weatherBlock;
}
```

## Linki i uwagi

W tej sekcji należy umieścić link do działającej aplikacji listy zadań oraz dowolne uwagi.

Link do strony:

<http://www.ai-labo1.ml/rest-weather/rest-weather.html>