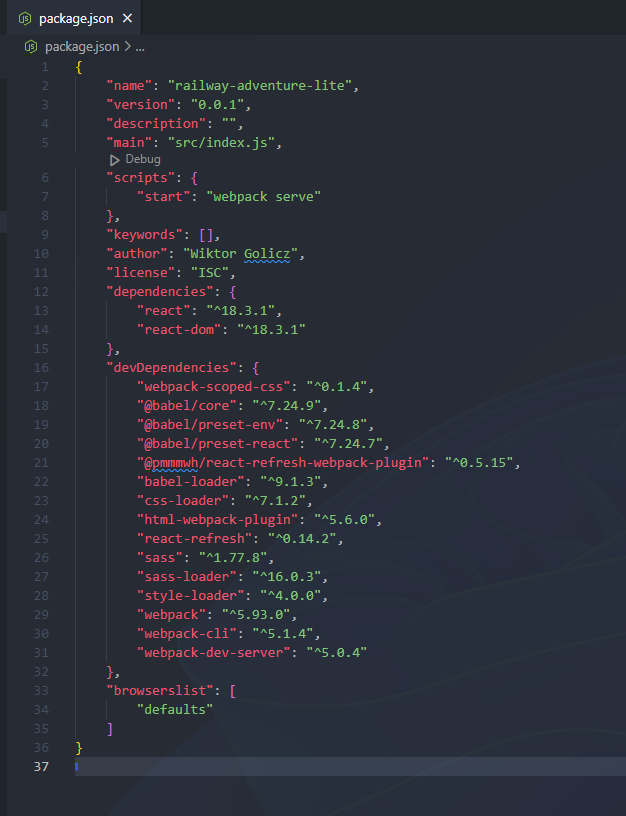
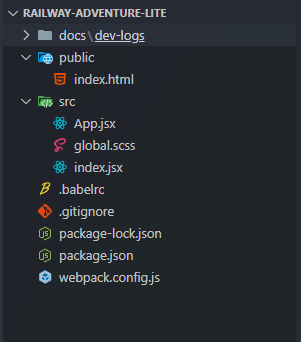
PROJEKT: Railway Adventure Lite (dev-log 2)

Wiktor Golicz | 02.12.2024

# Tworzenie projektu

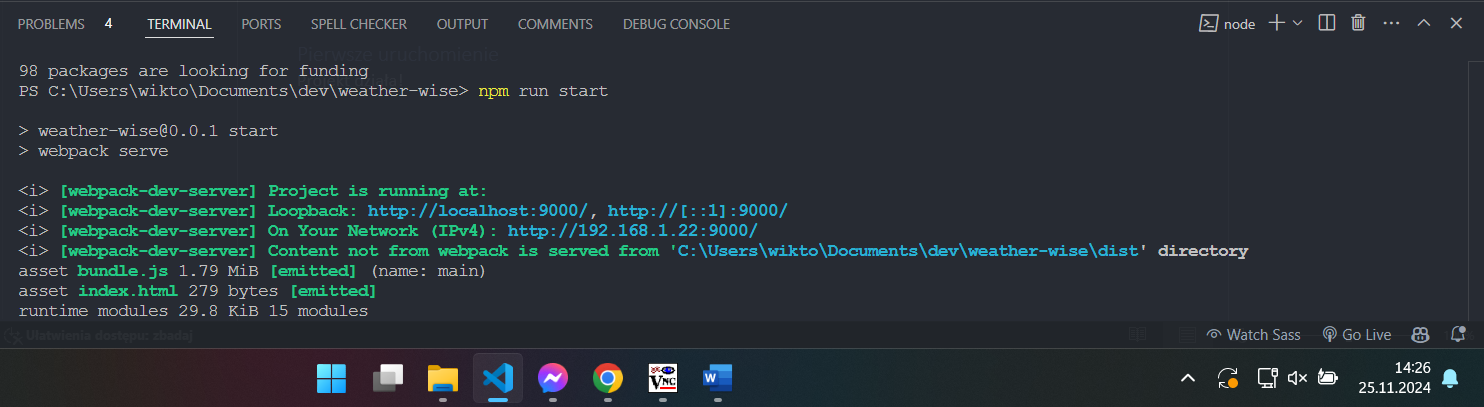
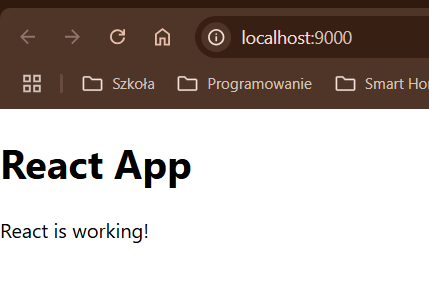
Mimo możliwości stworzenia automatycznego projektu (czy to z wykorzystaniem *create-react-app* czy *vite*), zdecydowałem się zainstalować wszystkie zależności samodzielnie i samodzielnie skonfigurować bundler – webpack’a. Tym sposobem będę miał pełną kontrolę nad wtyczkami. Przyda się to, ponieważ zamierzam użyć własnej wtyczki do lepszego zarządzania stylami tak by były odizolowane style konkretnych komponentów - <https://github.com/Wiktor102/railway-adventure-lite>.

## Pliki projektu

 Definiuję wszystkie potrzebne mi na ten moment zależności w pliku *package.json*.  
Tworzę także plik *webpack.config.js* i dodaję do niego konfigurację *webpacka*. Potrzebny jest też plik konfiguracyjny *.babelrc*. Oczywiście potrzebny jest też plik *index.html*, który umieszczam w katalogu *public*. Projekt będzie repozytorium, więc tworzę plik *.gitignore*. Dokumentację będę przechowywał w folderze *docs*. Kod aplikacji będzie się znajdował w folderze *src*. Cała struktura projektu prezentuje się na ten moment następująco:

## Pierwsze uruchomienie

Projekt działa!



# Routing

Na ten moment przewiduje jedynie potrzebę 2 ścieżek: dla strony głównej oraz dla gry. W przyszłości jednak może się pojawić konieczność dodatkowych ścieżek dla różnych zakładek / menu. Dlatego zdecydowałem się użyć biblioteki *react-router*.

Tworzę komponenty *Home* oraz *Game* i w pliku *App.js* umieszczam następujący kod:

## Strona główna

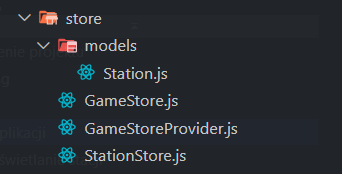
# Mapa

Gra będzie się odbywała na terenie województwa śląskiego, dlatego głównym jej elementem jest mapa. Do jej stworzenia użyję 2 bibliotek: *leaflet* oraz *react-leaflet*. Do wyświetlenia mapy mógłbym użyć kafelków  
np. z open-street-map, lecz tak dokładne szczegóły są zbędne w tym projekcie. Dlatego użyję formatu *geojson* by narysować jedynie kontury (przy okazji mam większą kontrolę nad stylami). Dane konturów pobrałem z geoportalu i przechowuję w pliku *slaskie.json*.

# Stan aplikacji

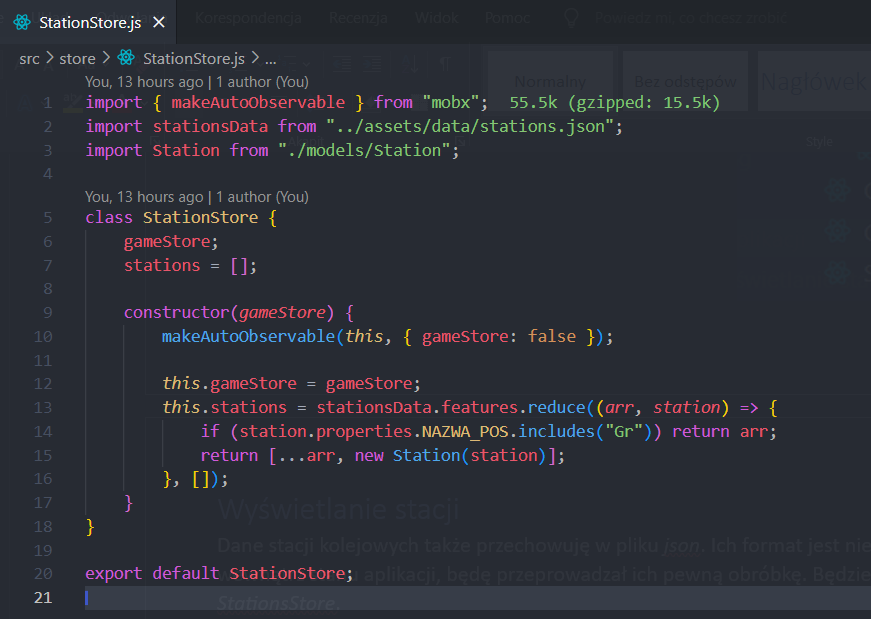
Ten projekt będzie dość złożony i będzie wymagał łączenia ze sobą danych na temat różnych obiektów w grze. Najlepszym sposobem na uproszczenie wszystkich zależności jest abstrakca, której najłatwiej dokonać w programowaniu obiektowym. React jest raczej biblioteką, która obrała inne podejście – niezmienność (immutability) danych. Oznacza to, że podejście OOP jest utrudnione, ale nie niemożliwe. Dlatego, do zarządzania stanem, wykorzystam bibliotekę *MobX*. Pozwala ona na przechowywanie danych w klasach a jej integracja z React’em sprawia, że wszystko działa jak należy.

W tym projekcie architektura będzie następująca: główny sore *GameStore* oraz mieszkające w nim *StationStore*, *TrainStore* itd. Dostęp do *GameStore* komponenty react’a uzyskają poprzez *context api*.

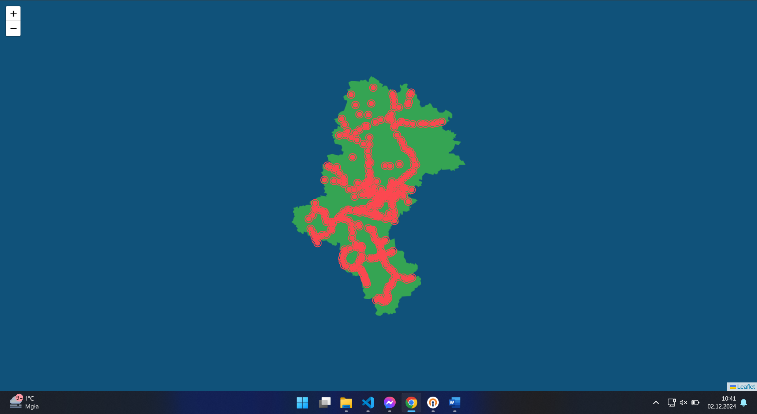


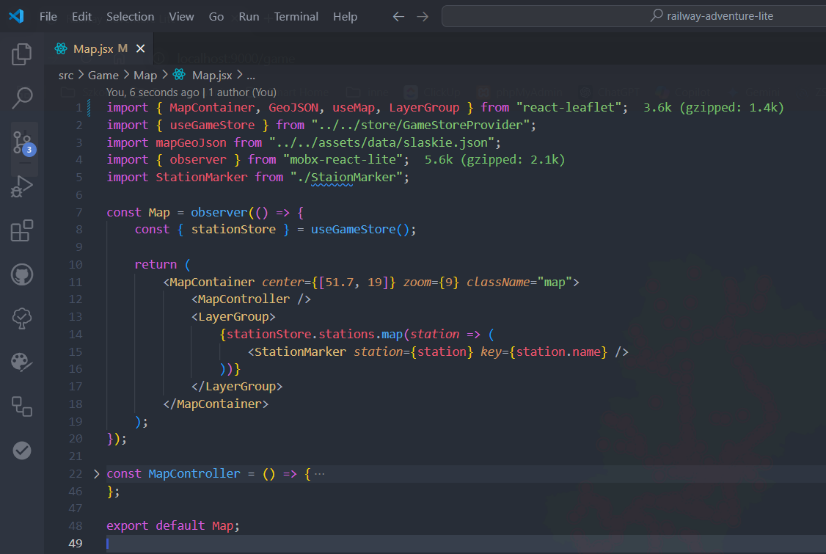
# Dane stacji oraz ich wyświetlanie na mapie

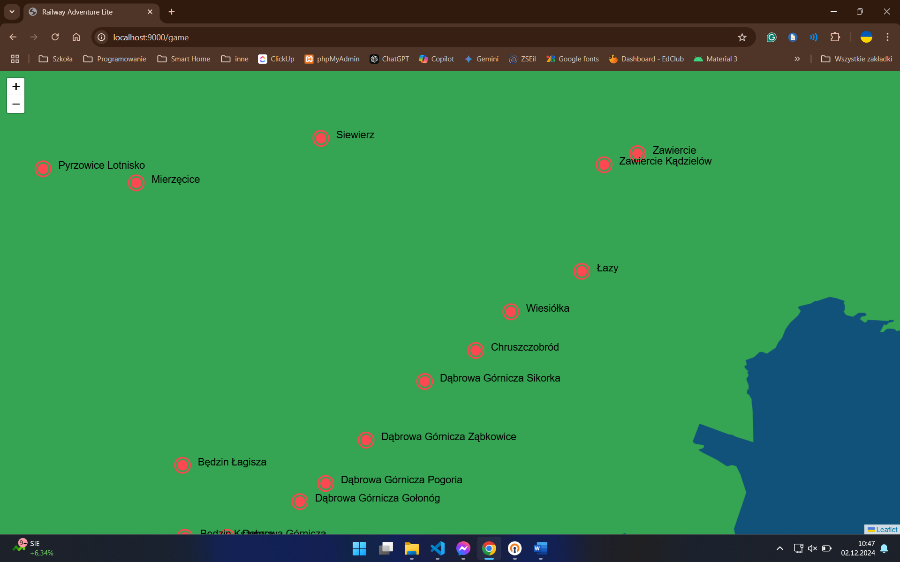
Dane stacji kolejowych także przechowuję w pliku *json*. Ich format jest nieco dziwny, więc przed ich zapisaniem wewnątrz stanu aplikacji, będę przeprowadzał ich pewną obróbkę. Będzie ona następowała w konstruktorze klasy *StationsStore*.

Na tym etapie, plik zawiera również stacje, które nie powinny się znaleźć w ostatecznej wersji gry (np. przystanki towarowe, posterunki graniczne). Brakuje również danych pokroju wielkości danej stacji (popytu / ilości pasażerów). Naprawą tych niedociągnięć zajmę się w późniejszym etapie gdy podstawowe mechaniki będą już działać.

## Wyświetlanie stacji

Samo wyświetlanie stacji, następuje w komponencie *Map* poprzez uzyskanie dostępu do *stationsStore* oraz wyrenderowaniu komponentu *StationMarker* dla każdej stacji.



Komponent *StationMarker* zarządza wyświetlaniem markera – zmienia jego rozmiar w zależności od przybliżenia mapy. Przy większym przybliżeniu pojawiają się dodatkowo podpisy poszczególnych stacji. W przyszłości niektóre mniejsze stacje będą całkowicie znikać przy małym przybliżeniu (a ich znaczniki będą mniejsze od pozostałych).