

**UNIWERSYTET EKONOMICZNY WE WROCŁAWIU**

**WYDZIAŁ ZARZĄDZANIA, INFORMATYKI I FINANSÓW**

**Kierunek Informatyka w Biznesie**

# **PROJEKT ZALICZENIOWY**

**INTEGRACJA APLIKACJI I SYSTEMÓW**

---

**Magdalena Warzyńska (145799)**

**Mateusz Zakrzewski (163312)**

Prowadzący:

inż. Paweł Szpak

## **1. Technologie użyte do zaprojektowania HUB'a.**

Aplikacja HUB'a została napisana w języku skryptowym Python w wersji 3.4.

Warstwa odpowiedzialna za obsługę żądań HTTP została napisana przy użyciu framework'a webowego Flask wraz z rozszerzeniem Flask-REST.

Do komunikacji z providerami danych użyto biblioteki HTTP – Requests.

## **2. Architektura systemu (część po stronie serwera).**

Jako providerów danych wykorzystano dwa serwisy sieciowe, dostarczające dane o aktualnej pogodzie dla wybranych współrzędnych geograficznych:

- I. Darksy - <https://darksy.net/dev/>
- II. OpenWeatherMap - <http://openweathermap.org/API>

Powyższe usługi sieciowe zwracają dane w lekkim formacie wymiany danych - JSON.

Konsumentem danych zwracanych przez powyższe usługi są dwie klasy parsujące odpowiedzi za pomocą biblioteki Requests. Powyższe klasy znajdują się w następujących modułach:

- I. `weather_api.providers.dark_sky.DarkSkyProvider`
- II. `weather_api.providers.openweathermap.OpenWeatherMapProvider`

Obydwe klasy w ramach interfejsu dziedziczą z abstrakcyjnej klasy bazowej `'weather_api.providers.base.BaseProvider'` i implementują dwie metody wymagane przez klasę agregującą HUB'a. Wyżej wymienione metody to `'current_weather'` oraz `'normalize_current_weather'`.

Klasa HUB'a jest zaimplementowana w module `weather_api.hub`. Agreguje ona dwie wyżej zadeklarowane klasy, tworząc jednolity interfejs dostępu do danych wszystkich providerów.

Następnie klasa HUB'a jest wykorzystywana przez framework Flask, który obsługuje żądania klientów używając interfejsu klasy HUB'a.

## **3. Architektura systemu (część po stronie klienta).**

Jako aplikację klienta utworzono prostą stronę internetową, pozwalającą na pobranie danych na temat aktualnej pogody dla wskazanej przez użytkownika lokalizacji. Użytkownik może wybrać za pomocą interfejsu GoogleMaps lokalizację, dla której chce sprawdzić, jakie warunki pogodowe panują w danym miejscu.

Aplikację klienta zaimplementowano przy pomocy języka JavaScript, biblioteki GoogleMaps oraz biblioteki jQuery.

#### **4. Napotkane problemy.**

Największym problemem było zapewnienie zgodności jednostek zwracanych przez wyżej wymienionych providerów. Wymagało to dokładnego przeszukiwania dokumentacji providerów.

Drugim z problemów było wyznaczenie części wspólnej dostarczanych danych tak, aby stworzyć jednolity interfejs API.

#### **5. Co byśmy zmienili, gdybyśmy robili projekt jeszcze raz.**

Gdybyśmy mogli zrobić projekt jeszcze raz, zaimplementowalibyśmy przechowywanie informacji o użytkownikach tak, aby użytkownicy mogli przechowywać ulubione lokalizacje.

Następną rzeczą, jaką chcielibyśmy dodać byłoby, pobieranie informacji na temat prognozy pogody, jako że obydwa providery dostarczają takich informacji.

#### **6. Podział pracy w grupie.**

Część projektowa oraz dokumentacja powstała wspólnymi siłami członków grupy. Każdy miał swój pomysł na projekt, ale dzięki analizie i dyskusji, udało nam się stworzyć jedną całość.

#### **7. Adres do repozytorium oraz instancji aplikacji.**

I. [https://github.com/magdawrz/ias\\_project](https://github.com/magdawrz/ias_project)

II. <http://magdawrz.pythonanywhere.com/>