

Integracja Aplikacji i Systemów

LABORATORIUM

HUB INTEGRUJĄCY DANE O MODELACH SAMOCHODÓW

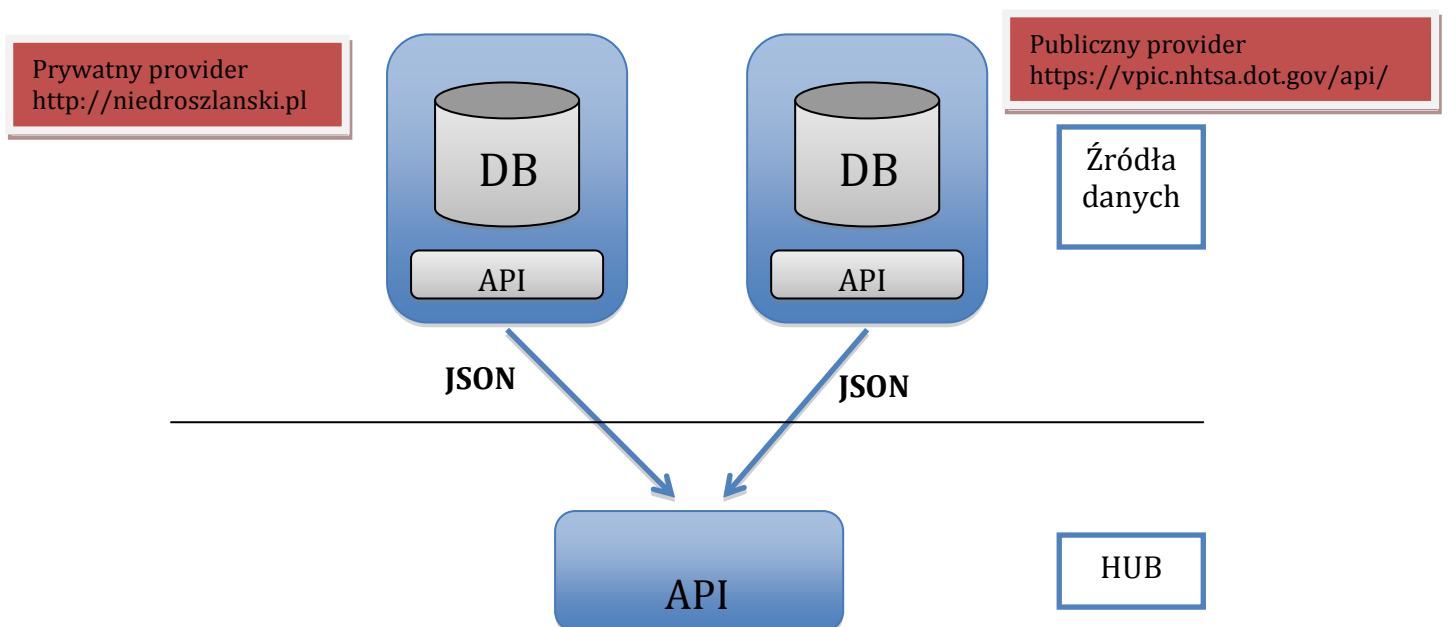
1. Zespół projektowy

Andrzej Furgał - 163303
Kamil Krzykowski - 163438
Przemysław Niedroszlański - 163307

2. Użyte technologie i języki programowania

- JSON format wymiany danych, pomiędzy providerami a brokerem. JSON jest formatem tekstowym, bazującym na podzbiorze języka JavaScript
- Java 8 – język w którym stworzyliśmy HUBa. Dzięki temu, że jest to wersja 8 można było wykorzystać m.in. funkcje lambda.
- IDE – korzystamy z IntelliJ w wersji 13;
- Glassfish - otwarty serwer aplikacji dla platformy Java EE, wykorzystany do uruchomienia HUBa
- Apache Maven - narzędzie automatyzujące budowę oprogramowania na platformę Java.

3. Schemat oraz opis architektury systemu



Nasz HUB łączy się z dwoma źródłami danych (providerami), dostępnymi w sieci i przy użyciu własnego API udostępnia zintegrowane dane.

4. Opis dostawców, struktura encji

- a) **<https://vpic.nhtsa.dot.gov/api/>** - urzędowa baza danych marek samochodowych dostępnych w USA.

Wykorzystujemy następujące webserwisy:

1) Typy samochodów według marek

URL	<code>/vehicles/GetVehicleTypesForMake/make?format=json</code>
Struktura encji:	<ul style="list-style-type: none">- Identyfikator Marki- Nazwa Marki- Id typu pojazdu- Typ pojazdu
Przykład pojedynczej encji	<pre>{ "MakeId":4463, "MakeName":"Trail Maker Trailers", "VehicleTypeId":6, "VehicleTypeName":"Trailer" }</pre>

2) Modele samochodów według Marek

URL	<code>/vehicles/GetModelsForMake/honda?format=json</code>
Struktura encji:	<ul style="list-style-type: none">- Identyfikator Marki- Nazwa Marki- Id modelu- Nazwa modelu
Przykład pojedynczej encji	<pre>{ "Make_ID":474, "Make_Name":"Honda", "Model_ID":1864, "Model_Name":"Pilot" }</pre>

3) Marki samochodów dla wybranego typu samochodu

URL	/vehicles/GetMakesForVehicleType/car?format=json
Struktura encji:	<ul style="list-style-type: none">- Identyfikator Marki- Nazwa Marki- Id typu pojazdu- Typ pojazdu
Przykład pojedynczej encji	<pre>{ "MakeId":452, "MakeName":"BMW", "VehicleTypeId":2, "VehicleTypeName":"Passenger Car" }</pre>

b) <http://niedroslanski.pl/car-api/> - baza danych samochodów polskich producentów.

Wykorzystujemy następujące Rest-pointy:

1) Typy samochodów

URL	http://niedroslanski.pl/car-api/typy-samochodow
Struktura encji:	<ul style="list-style-type: none">- Typ samochodu
Przykład pojedynczej encji	<pre>{ "typ":"car" }</pre>

2) Marki samochodów

URL	http://niedroslanski.pl/car-api/marki
Struktura encji:	<ul style="list-style-type: none">- Typ samochodu- Nazwa Marki
Przykład pojedynczej encji	<pre>{ "typ":"car", "NazwaMarki":"FSO" }</pre>

3) Modele samochodów według marek

URL	http://niedroslanski.pl/car-api/modele-wg-marek?marka
Struktura encji:	<ul style="list-style-type: none">- Nazwa modelu
Przykład pojedynczej encji	<pre>{ "NazwaModelu":"Polonez Caro" }</pre>

5. Opis huba, przebieg integracji encji, struktura wynikowa encji

Nasza implementacja HUBa jest umieszczona w repozytorium <https://github.com/nieprzemek/IAS-HUB>. Wykorzystując biblioteki `java.servlet`, `javax.org.json` odpytuje webservisy providerów, które są podane w punkcie 4.

Zwrot danych po integracji możliwy jest po odpytaniu webserwisu HUBa[hub]/.

Nasza implementacja HUBa przetwarza dane w locie – nie posiada własnej bazy danych. Jeśli wystąpi sytuacja w której któryś z providerów będzie niedostępny, hub zwróci dane tylko z dostępnego providera. Dane, które HUB zwraca są unikalne i posortowane. W razie podania błędnego URL'a HUB zwraca komunikat „Błędny URL”. Jeśli serwer nie znajdzie żadnych wyników (np. będziemy szukać modeli nieistniejącej marki) otrzymamy odpowiedź : „Brak danych”

Opis encji po integracji:

URL	<code>http://localhost:9999/IAS_HUB/models?fso</code>
Działanie funkcji	Funkcja zwraca wszystkie modele podanej marki (w tym przypadku jest to „fso”) z obu providerów.
Przykład wyniku	1500 POLONEZ ATU POLONEZ CARO

URL	<code>http://localhost:9999/IAS_HUB/cars-types</code>
Działanie funkcji	Funkcja zwraca wszystkie typy samochodów z obu providerów.
Przykład wyniku	BUS CAR INCOMPLETE VEHICLE MULTIPURPOSE PASSENGER VEHICLE (MPV) ...

URL	<code>http://localhost:9999/IAS_HUB/makes?car</code>
Działanie funkcji	Funkcja zwraca wszystkie marki na podstawie podanego typu samochodu (w tym przypadku jest to „car”) z obu providerów.
Przykład wyniku	ACURA ALAN AUTOSPORTS ALFA ROMEO AM GENERAL ARMBRUSTER STAGEWAY ASTON MARTIN ASUNA AUDI AVANTI MOTORS AVERA MOTORS ...

6. Napotkane problemy

7. Adres do repozytorium

<https://github.com/nieprzemek/IAS-HUB>

8. Podział pracy w grupie

- Andrzej Furgal – przygotowanie środowiska, przygotowanie dokumentacji, oprogramowanie HUBa
- Kamil Krzykowski – oprogramowanie HUBa, przygotowanie dokumentacji, testy HUBa
- Przemysław Niedroszlański – przygotowanie dokumentacji, znalezienie/przygotowanie źródeł danych, oprogramowanie HUBa