

Dokumentacja projektu Integracji Aplikacji i Systemów

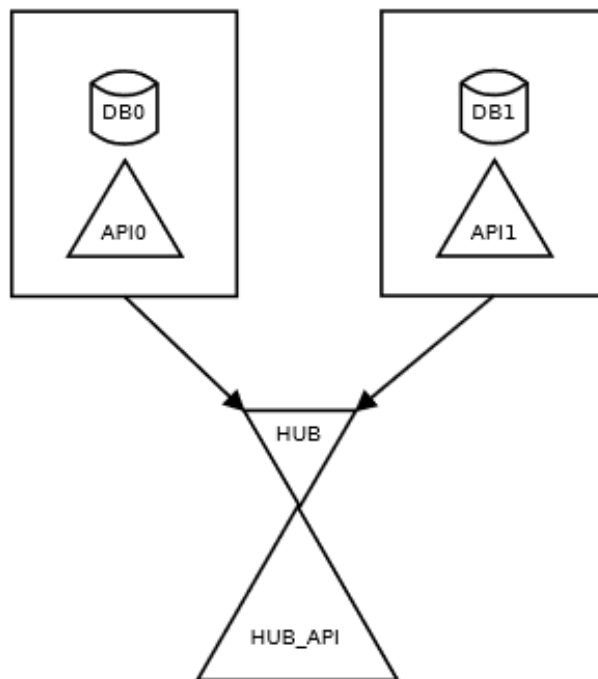
Aplikacja integrująca wyniki zapytań do sklepów odzieżowych.

1. Technologie

- Baza danych: sqllite3
- Provider, hub: Python 3, Flask
- Format danych: JSON i Protocol Buffers

2. Schemat i opis architektury systemu

Dwóch dostawców pobiera dane ze swoich baz danych. Hub nie korzysta z bazy danych, korzysta z API dostawców. Zwraca wyniki zapytań o dane z obu dostawców.



3. Opis dostawców

Dostawcy to sklepy odzieżowe korzystające z osobnych baz danych.

- Provider1 (JSON)
 - Informacje o produkcie:
 - id – nadawane jest automatycznie
 - name – nazwa produktu np. „Pants no 1234”
 - sex – id płci dla której jest przeznaczony produkt; może zawierać kilka (produkty typu unisex); osobne oznaczenie produktów dla dzieci
 - category – id kategorii produktu np. buty, spodnie, czapka
 - collection – kolekcja np. Jesienna zimowa itp.
 - price – cena produktu
 - previous_prices – historyczne ceny (data i cena)
 - description – dłuższy opis produktu
 - Struktura bazy danych:
 - items:

- (rowid) – automatyczne
- price (real) – cena produktu
- name (char(50)) – nazwa produktu
- description (char(250)) – opis produktu
- sex
 - (rowid) – automatyczne
 - id (int) – id produktu z items
 - sex (int) – płeć dla której jest przeznaczony produkt
- category
 - (rowid) – automatyczne
 - id (int) – id produktu z items
 - category (int) – id kategorii dla której jest przeznaczony produkt
- collection
 - (rowid) – automatyczne
 - id (int) – id produktu z items
 - collection (int) – kolekcja produktu
- prices
 - (rowid) – automatyczne
 - id (int) – id produktu z items
 - price (real) – cena produktu
 - timestamp (int) – timestamp ceny
- Obsługiwane metody:
 - items: zwraca id wszystkich produktów w bazie
 - item/<int:id>: informacja o produkcie z danym id
 - sex/<int:id>: zwraca id wszystkich produktów dla danej płci
 - price/more/<int:amount>: zwraca id wszystkich produktów których cena jest większa lub równa wskazanej
 - price/less/<int:amount>: zwraca id wszystkich produktów których cena jest mniejsza niż wskazana

Przykłady zapytań dla Providera1

- a) Zapytanie o produkt o numerze id=3. Otrzymujemy informację takie jak: name, description, previous_prices itd.

```
(ias) MacBook-Air-Joanna:ias joannakozak$ curl localhost:5000/api/v1.0/item/3
{
  "data": {
    "item": {
      "description": "Nice hat 3",
      "id": 3,
      "name": "Hat no 235",
      "previous_prices": [
        {
          "1480944884": 309.0
        },
        {
          "1480844584": 299.0
        }
      ],
      "price": 948.9,
      "sex": [
        {
          "category": [
            3
          ]
        },
        2
      ]
    }
  }
}
```

```

    ]
  }
}
}

```

b) Zapytanie o produkty z oznaczeniem płci=1. Otrzymujemy id tych produktów – 1, 2.

```

(ias) MacBook-Air-Joanna:ias joannakozak$ curl localhost:5000/api/v1.0/sex/1
{
  "data": {
    "items": [
      1,
      2
    ]
  }
}

```

- Provider2 (protobuf)
 - Informacje o produkcie:
 - id – UUID
 - name – nazwa produktu
 - price – cena produktu
 - sex – płeć dla której przeznaczony jest produkt
 - category – kolekcja
 - description – dłuższy opis produktu
 - photos – adresy do zdjęć
 - materials - lista materiałów z których wykonany jest produkt
 - Struktura bazy danych:
 - items:
 - (rowid) – automatyczne
 - uuid (char(50)) - UUID
 - price (real) – cena produktu
 - name (char(50)) – nazwa produktu
 - description (char(250)) – opis produktu
 - sex
 - (rowid) – automatyczne
 - id (int) – id produktu z items
 - sex (int) – płeć dla której jest przeznaczony produkt
 - category
 - (rowid) – automatyczne
 - id (int) – id produktu z items
 - category (int) – id kategorii dla której jest przeznaczony produkt
 - photos
 - (rowid) – automatyczne
 - id (int) – id produktu z items
 - url (char(250)) – URL do zdjęcia produktu
 - materials
 - (rowid) – automatyczne
 - id (int) – id produktu z items
 - materials (char(50)) – materiał
 - Obsługiwane metody:
 - get_items: zwraca id wszystkich produktów w bazie
 - get_item/<int:id>: zwraca informacje o produkcie z danym id
 - get_Sex/<int:id>: zwraca id wszystkich produktów dla danej płci
 - price/more/<int:amount>: zwraca id wszystkich produktów których cena jest większa lub równa wskazanej

- price/less/<int:amount>: zwraca id wszystkich produktów których cena jest mniejsza niż wskazana
- get_materials/<string:id>: zwraca id produktów wykonanych z danego materiału

Przykład zapytania dla Providera2

Poniższe zapytanie zwraca id wszystkich produktów z ceną mniejszą niż 500.

```
MacBook-Air-Joanna:~ joannakozak$ curl localhost:5001/api/v1.0/price/less/500
12720a2432636532643531302d626431662d313165362d393439362d3430386435633466323030650
a2432636532643531312d626431662d313165362d393439362d3430386435633466323030650a2432
636532643531322d626431662d313165362d393439362d343038643563346632303065
```

4. HUB

Korzysta z danych pobieranych od dostawców. Dane przez niego zwracane są w formacie JSON.

- Struktura wynikowa encji:
 - id: uuid lub id produktu
 - provider: nazwa providera produktu
 - name: nazwa produktu
 - price: cena produktu
 - sex: płeć dla której przeznaczony jest produkt
 - category: kolekcja z której pochodzi produkt
 - description: dłuższy opis produktu
 - provider_specific (jeżeli provider obsługuje te pola)
 - photos: adresy do zdjęć
 - materials: lista materiałów
 - previous_prices: ceny historyczne
- Obsługiwane metody:
 - get_items: zwraca id oraz nazwę providera wszystkich produktów w bazie
 - get_item/<string:provider>/<int:id>: informacja o produkcie z danym id od danego providera
 - get_Sex/<int:id>: zwraca id oraz nazwę povidera wszystkich produktów dla danej płci
 - price/more/<int:amount>: zwraca id oraz nazwę povidera wszystkich produktów których cena jest większa lub równa wskazanej
 - price/less/<int:amount>: zwraca id oraz nazwę povidera wszystkich produktów których cena jest mniejsza niż wskazana
 - get_materials/<string:id> zwraca id oraz nazwę povidera produktów wykonanych z danego materiału
- Jeżeli pola występowały w encjach obu dostawców to pole jest w wynikowej encji. Jeżeli nie - trafia do pola „provider specific”.

5. Link do repozytorium: <https://github.com/joannakozak/ias>

6. Podział pracy

143077 Joanna Kozak: Provider1; testowanie; przygotowanie dokumentacji
 140191 Łukasz Żerkowski: Provider2; testowanie
 163979 Paweł Radecki: Hub; testowanie