

Buy it

Spis treści

1. Architektura projektu	1
1.1. Komponenty	2
1.1.1. Frontend	2
1.1.1.1. Buy It Web	2
1.1.1.2. Buy It Web Admin	3
1.1.1.2.1. Administrator	3
1.1.1.2.2. Operator magazynu	3
1.1.1.2.3. Operator zamówień	3
1.1.2. Backend	3
1.1.3. Buy It BE	4
2. Aktualny stan projektu	4
2.1. Buy It BE	4
2.1.1. Uruchomienie lokalnie	5
2.1.2. Testy API	6
2.1.2.1. Utworzenie marki	6
2.1.2.2. Modyfikacja kodu marki	8
2.1.2.3. Utworzenie produktu	9
2.1.2.4. Modyfikacja produktu	11
2.1.2.5. Wyszukanie produktu po ID	13
2.1.2.6. Wyszukanie produktu po kodzie	15
2.1.2.7. Stan aplikacji po testach	17
2.1.2.7.1. PostgreSQL	17
2.1.2.7.2. Redpanda	18

Projekt realizowany w ramach przedmiotu Projekt Zespołowy prowadzony przez Ośrodek Kształcenia Na Odległość Politechniki Warszawskiej. W ramach projektu zdecydowano się zrealizować uproszczoną implementację sklepu internetowego, sprzedającego obuwie.

1. Architektura projektu

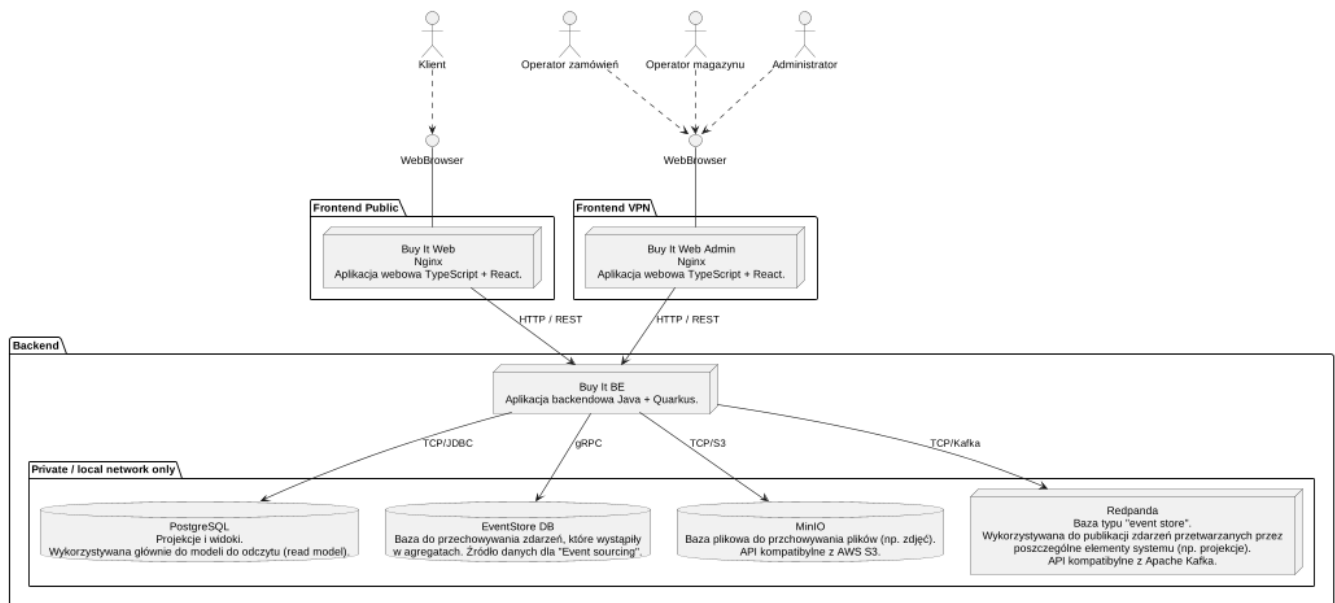
Projekt został podzielony na 2. części - 2. aplikacje backendowe oraz 1. frontendowa.

Aplikacja "Buy It Web" to ogólnodostępna, publiczna aplikacja sklepu internetowego. Klienci mogą za jej pomocą przeglądać i wyszukiwać produkty czy składać zamówienia.

Aplikacja "Buy It Web Admin" dostępna jest tylko dla zweryfikowanych pracowników sklepu, a

połączenie sieciowe do aplikacji możliwe powinno być jedynie w ramach wydzielonej sieci [VPN](#).

Na rysunku poniżej przedstawiono diagram komponentów wykorzystywanych przez aplikację.



Rysunek 1. Komponenty aplikacji Buy It - Level 2 modelu C4

Powyższy diagram przygotowany został z wykorzystaniem [C4 model - level 2](#).

1.1. Komponenty

1.1.1. Frontend

Obie aplikacje frontendowe to aplikacje Webowe serwowana jako pliki statyczne za pośrednictwem serwera [Nginx](#).

Aplikacje Webowe zostały napisane z wykorzystaniem:

1. Języka [TypeScript](#) (kompilowanego do [JavaScript](#)).
2. Języka [CSS](#) / [Saas SCSS](#).
3. Języka [HTML](#).
4. Frameworka [React](#).

Komunikacja między aplikacjami frontendowymi a aplikacją backendową została zrealizowana z wykorzystaniem HTTP z zastosowaniem REST ([Richardson Maturity Level 2](#)).

1.1.1.1. Buy It Web

Publicznie dostępna aplikacja internetowego sklepu z obuwem, której odbiorcami są klienci sklepu.

Aplikacja udostępnia takie funkcje jak:

1. Utworzenie i logowanie do konta klienta.

2. Wyszukiwanie produktów wraz z informacją o ich dostępności i wariantów (np. różnych rozmiarów czy kolorów).
3. Zarządzanie koszykiem zakupowym - dodawanie / usuwanie produktów.
4. Złożenie o opłacenia zamówienia na podstawie koszyka zakupowego.
5. Przegląd historii zamówień.

1.1.1.2. Buy It Web Admin

Aplikacja typu "backoffice" dla pracowników sklepu. Z założenia, niedostępna publicznie w sieci Internet.

Z aplikacji korzystać mogą pracownicy, posiadający różne role w aplikacji:

1. Administrator
2. Operator magazynu
3. Operator zamówień

1.1.1.2.1. Administrator

1. Wyszukiwanie i modyfikacja kont pracowniczych. [Administrator]
2. Modyfikacja ról użytkowników.

1.1.1.2.2. Operator magazynu

1. Przeglądanie / wyszukiwanie produktów. [Operator magazynu]
2. Modyfikację produktów (dodawanie / usuwanie, edycja atrybutów - np. rozmiaru czy koloru). [Operator magazynu]

1.1.1.2.3. Operator zamówień

1. Przeglądanie / wyszukiwanie zamówień. [Operator zamówień]
2. Zmiana statusów zamówień (Złożone, Opłacone, W trakcie realizacji, Wysłane, Dostarczone). [Operator zamówień]

1.1.2. Backend

Aplikacja backendowa to aplikacja serwerowa, która zarządza stanem magazynów, ofertą, zamówieniami itp. Ta aplikacja jest głównym źródłem danych dla całego systemu i dba o zachowanie ich spójności.

W ramach aplikacji backendowej możemy wyróżnić takie komponenty jak:

1. [Buy It BE](#) - aplikacja serwerowa napisana w języku Java, która implementuje logikę biznesową systemu po stronie backendu. Udostępnia 2. rodzaje API HTTP dla klientów aplikacji webowych - osobne dla aplikacji dla klientów sklepu [Buy It Web](#) oraz osobne dla pracowników [Buy It Web Admin](#).
2. [EventStore DB](#) - baza danych noSQL, przechowująca strumień zdarzeń domenowych,

wykonanych / zaaplikowanych przez agregaty domenowe w architekturze [Event Sourcing](#).

3. [PostgreSQL](#) - relacyjna baza danych, przechowująca dane do podglądu. Wygenerowane z użyciem projekcji na podstawie zdarzeń domenowych.
4. [MinIO](#) - baza obiektowa, przechowująca zdjęcia produktów czy logo marek. Posiada API kompatybilne z [AWS S3](#).
5. [Redpanda](#) - broker zdarzeń, baza strumieniowa, służąca jako medium wymiany zdarzeń pomiędzy komponentami logiki biznesowej (np. agregatami domenowymi i projekcjami). Posiada API kompatybilne z [Apache Kafka](#).

1.1.3. Buy It BE

Jest to główny komponent backendowy systemu. W tym komponencie zaimplementowana została logika biznesowa części backendowej sklepu.

Aplikacja została napisana z wykorzystaniem architektury [Event Sourcing](#) oraz elementami architektur [CQRS](#) i [Heksagonalnej](#).

Wykorzystane języki i frameworki do utworzenia aplikacji to:

1. [Java 17](#).
2. [SQL](#).
3. [Quarkus](#).
4. [SmallRye Mutiny](#).

2. Aktualny stan projektu

2.1. Buy It BE

W ramach części backendowej projektu, aktualnie zrealizowano:

1. Architekturę aplikacji backendowej.
 - Przygotowanie abstrakcji [Event Sourcing](#) oraz [CQRS](#) - dla agregatów i zdarzeń domenowych, publikacji zdarzeń, projekcji, obsługi komend i zapytań.
 - Przygotowanie abstrakcji dla zapytań z wykorzystaniem [Keyset Pagination](#)
 - Podział aplikacji [Quarkus](#) na moduły [Maven](#), starając się odwzorować architekturę [Heksagonalną](#).
 - Konfiguracja testów integracyjnych [Quarkus](#) z wykorzystaniem [Testcontainers](#).
2. Integrację z bazą [EventStore DB](#).
3. Integrację z bazą [PostgreSQL](#) wraz z wykorzystaniem narzędzia migracji bazy danych [Liquibase].
4. Integrację z brokerem komunikatów / zdarzeń [Redpanda](#).
5. Integrację z bazą [MinIO](#).

6. Konfigurację środowiska lokalnego z wykorzystaniem [Docker](#) i [docker compose](#).
7. Implementacja pierwszych usług do zarządzania produktem.
 - Proste wyszukiwanie produktu po podstawowych atrybutach (id produktu, kod produktu, id marki)
 - Tworzenie nowego produktu wraz z jego wariacjami / zestawami atrybutów
 - Aktualizacja produktu wraz z jego wariacjami / zestawami atrybutów
 - Aktualizacja liczby wariacji produktów w magazynie
 - Tworzenie i modyfikacja marek.
8. Przygotowanie dokumentacji [OpenAPI](#) dla wystawionych usług HTTP wraz z wykorzystaniem [Quarkus Swagger UI](#).

2.1.1. Uruchomienie lokalnie

Część backendową można uruchomić na lokalnym środowisku. Wymagane jest posiadanie na maszynie [Docker](#) wraz z [docker compose](#) oraz [Java 17](#).

W celu ułatwienia uruchomienia, przygotowano plik [Makefile](#) z przygotowanymi "celami":

1. [local-infra-up](#) - uruchamia infrastrukturę potrzebną do działania aplikacji ([EventStore DB](#), [PostgreSQL](#), [MinIO](#), [Redpanda](#)) z wykorzystaniem [docker compose](#).
2. [run-local](#) - uruchamia aplikację Quarkus na lokalnej maszynie.

```
$: make local-infra-up
docker compose -f local/docker-compose.yml up -d
[+] Building 0.0s (0/0)
[+] Running 9/9
✓ Container buy-it-eventstoredb-certs Started 0.4s
✓ Container buy-it-postgres-adminer Running 0.0s
✓ Container buy-it-postgres Running 0.0s
✓ Container buy-it-minio Running 0.0s
✓ Container buy-it-redpanda Running 0.0s
✓ Container buy-it.eventstore.node3 Running 0.0s
✓ Container buy-it.eventstore.node2 Running 0.0s
✓ Container buy-it-redpanda-console Running 0.0s
✓ Container buy-it.eventstore.node1 Running 0.0s

$: make run-local
./mvnw compile quarkus:dev -pl application -Dquarkus.profile=local --also-make
# ...
# logi z uruchomienia aplikacji Quarkus
# ...
2023-07-01 15:08:21,097 INFO [io.quarkus] (Quarkus Main Thread) application 0.0.0-227e5a29 on JVM (powered by Quarkus 3.1.0.Final) started in 7.773s. Listening on: http://localhost:4000
2023-07-01 15:08:21,098 INFO [io.quarkus] (Quarkus Main Thread) Profile local activated. Live Coding activated.
```

```
2023-07-01 15:08:21,099 INFO [io.quarkus] (Quarkus Main Thread) Installed features:
[agroal, cdi, config-yaml, hibernate-validator, jdbc-postgresql, kafka-client,
liquibase, narayana-jta, reactive-pg-client, resteasy-reactive, resteasy-reactive-
jackson, smallrye-context-propagation, smallrye-graphql, smallrye-openapi, smallrye-
reactive-messaging, smallrye-reactive-messaging-kafka, swagger-ui, vertx]
```

```
--
```

```
Tests paused
```

```
Press [r] to resume testing, [o] Toggle test output, [:] for the terminal, [h] for
more options>
```

Po uruchomieniu aplikacji w ten sposób, dostępny jest panel developerski Quarkus pod adresem <http://localhost:4000/q/dev-ui/extensions>.

2.1.2. Testy API

W ramach aktualnego stanu projektu, wykonano proste manualne testy API z wykorzystaniem Swagger UI.

2.1.2.1. Utworzenie marki

POST /brands

Parameters

No parameters

Request body

multipart/form-data

logoImage object

code string

ADIDAS

Execute Clear

Responses

Curl

```
curl -X 'POST' \
  'http://localhost:4000/brands' \
  -H 'accept: application/json' \
  -H 'Content-Type: multipart/form-data' \
  -F 'logoImage={}' \
  -F 'code=ADIDAS'
```

Request URL

http://localhost:4000/brands

Server response

Code	Details
201 <i>Undocumented</i>	<p>Response body</p> <pre>{ "id": "04abd190-656e-45e5-827a-0c8a426af8f7" }</pre> <p>Response headers</p> <pre>content-length: 45 content-type: application/json;charset=UTF-8</pre>

Rysunek 2. Utworzenie marki - Swagger UI

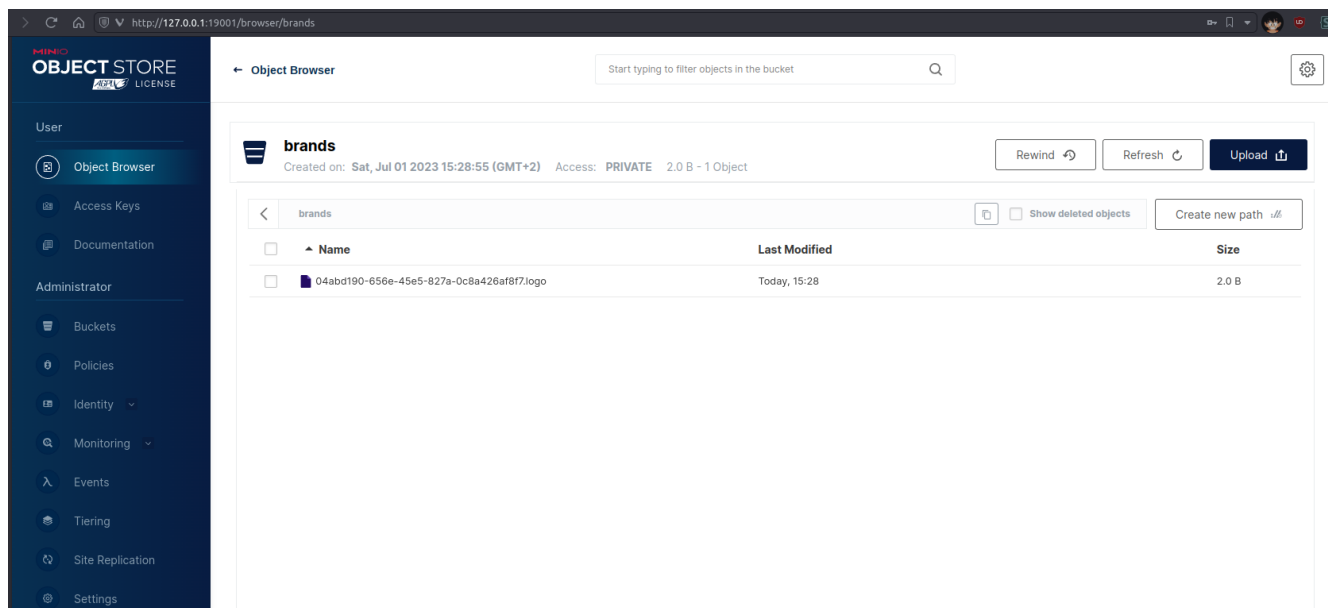
Wykonane żądanie odpowiada wywołaniu cURL:

```
curl -X 'POST' \
  'http://localhost:4000/brands' \
  -H 'accept: application/json' \
  -H 'Content-Type: multipart/form-data' \
  -F 'logoImage={}' \
  -F 'code=ADIDAS'
```

Otrzymano odpowiedź o statusie 201 CREATED i ID utworzonej marki w treści:

```
{
  "id": "04abd190-656e-45e5-827a-0c8a426af8f7"
}
```

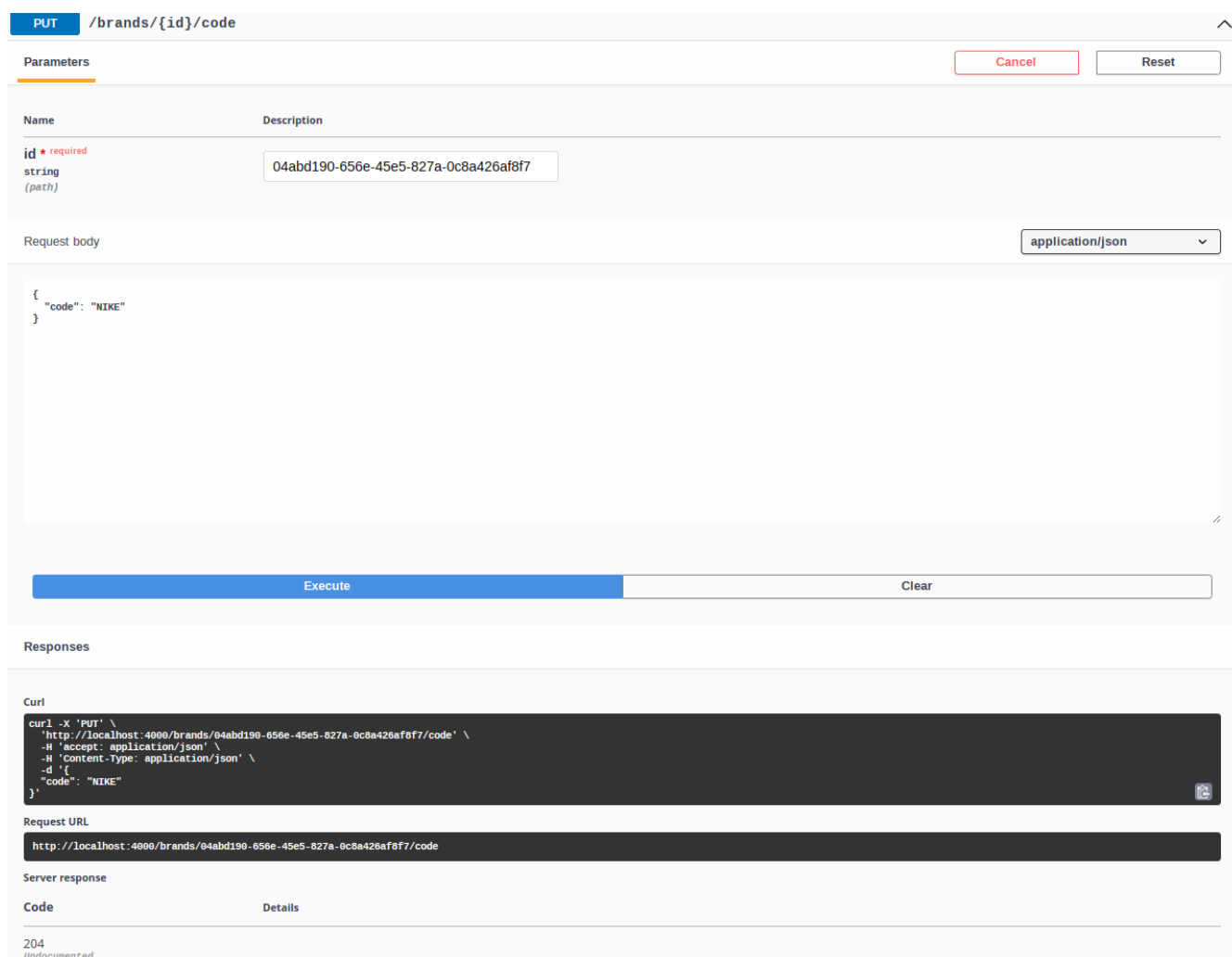
W konsoli MinIO widzimy zapisane logo dla marki o tym samym ID:



Rysunek 3. Logo marki - MinIO

Logo ma 2 bajty, ponieważ w testach jako logo przestany został "string" {}.

2.1.2.2. Modyfikacja kodu marki



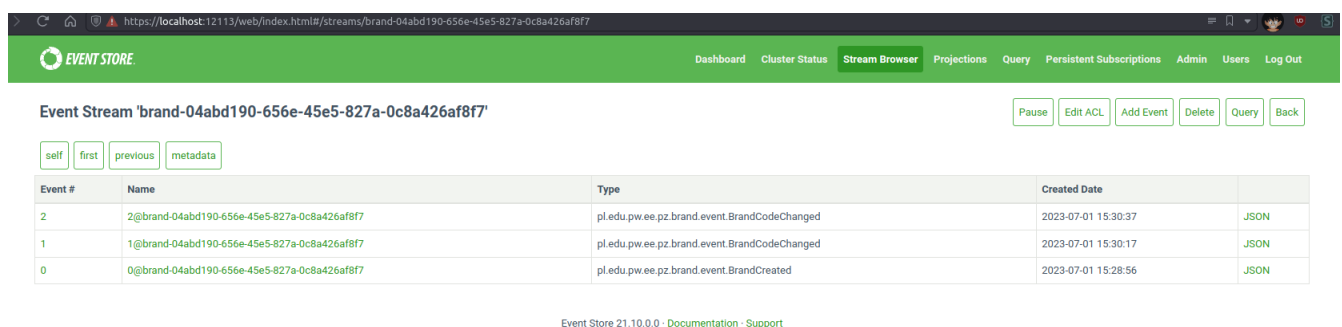
Rysunek 4. Modyfikacja kodu marki - Swagger UI

Wykonane żądanie odpowiada wywołaniu cURL:

```
curl -X 'PUT' \
  'http://localhost:4000/brands/04abd190-656e-45e5-827a-0c8a426af8f7/code' \
  -H 'accept: application/json' \
  -H 'Content-Type: application/json' \
  -d '{
    "code": "NIKE"
  }'
```

Otrzymano odpowiedź o statusie 204 NO_CONTENT.

W konsoli EventStoreDB (<https://localhost:12113/>) możemy zaobserwować zdarzenia domenowe zarejestrowane dla marki:



Event Stream 'brand-04abd190-656e-45e5-827a-0c8a426af8f7'

Event #	Name	Type	Created Date
2	2@brand-04abd190-656e-45e5-827a-0c8a426af8f7	pl.edu.pw.ee.pz.brand.event.BrandCodeChanged	2023-07-01 15:30:37
1	1@brand-04abd190-656e-45e5-827a-0c8a426af8f7	pl.edu.pw.ee.pz.brand.event.BrandCodeChanged	2023-07-01 15:30:17
0	0@brand-04abd190-656e-45e5-827a-0c8a426af8f7	pl.edu.pw.ee.pz.brand.event.BrandCreated	2023-07-01 15:28:56

Event Store 21.10.0.0 - Documentation - Support

Rysunek 5. Zdarzenia dla marki - EventStoreDB

2.1.2.3. Utworzenie produktu

POST

/products

Parameters

No parameters

Request body

application/json

```
{
  "code": "AIR_MAX_PULSE",
  "brandId": "04abd190-656e-45e5-827a-0c8a426af8f7",
  "variations": [
    {
      "id": "42009870-2248-4f82-84cb-7965c4a2b2ff",
      "attributes": [
        {
          "type": "SIZE_EU",
          "value": {
            "type": "Integer",
            "value": 36
          }
        },
        {
          "type": "COLOR_1",
          "value": {
            "type": "String",
            "value": "WHITE"
          }
        },
        {
          "type": "COLOR_2",
          "value": {
            "type": "String",
            "value": "GREY"
          }
        }
      ]
    }
  ]
}
```

Execute

Clear

Responses

Curl

```
curl -X 'POST' \
'http://localhost:4000/products' \
-H 'accept: application/json' \
-H 'Content-Type: application/json' \
-d '{
  "code": "AIR_MAX_PULSE",
  "brandId": "04abd190-656e-45e5-827a-0c8a426af8f7",
  "variations": [
    {
      "id": "42009870-2248-4f82-84cb-7965c4a2b2ff",
      "attributes": [
        {
          "type": "SIZE_EU",
          "value": {
            "type": "Integer",
            "value": 36
          }
        },
        {
          "type": "COLOR_1",
          "value": {
            "type": "String",
            "value": "WHITE"
          }
        },
        {
          "type": "COLOR_2",
          "value": {
            "type": "String",
            "value": "GREY"
          }
        }
      ]
    }
  ]
}'
```

Request URL

http://localhost:4000/products

Server response

Code	Details
201	<div>Response body</div> <pre>{ "id": "7a5cab46-2763-4841-a74d-45fe1b7ef313" }</pre> <div>Response headers</div> <pre>content-length: 45 content-type: application/json;charset=UTF-8</pre>

Rysunek 6. Utworzenie produktu - Swagger UI

Wykonane żądanie odpowiada wywołaniu cURL:

```
curl -X 'POST' \
'http://localhost:4000/products' \
-H 'accept: application/json' \
-H 'Content-Type: application/json' \
-d '{
  "code": "AIR_MAX_PULSE",
  "brandId": "04abd190-656e-45e5-827a-0c8a426af8f7",
  "variations": [
    {
      "id": "42009870-2248-4f82-84cb-7965c4a2b2ff",
      "attributes": [
        {
          "type": "SIZE_EU",
          "value": {
            "type": "Integer",
            "value": 36
          }
        },
        {
          "type": "COLOR_1",
          "value": {
            "type": "String",
            "value": "WHITE"
          }
        },
        {
          "type": "COLOR_2",
          "value": {
            "type": "String",
            "value": "GREY"
          }
        }
      ]
    }
  ]
}'
```

```

    "id": "42009870-2248-4f82-84cb-7965c4a2b2ff",
    "attributes": [
      {
        "type": "SIZE_EU",
        "value": {
          "type": "Integer",
          "value": 36
        }
      },
      {
        "type": "COLOR_1",
        "value": {
          "type": "String",
          "value": "WHITE"
        }
      },
      {
        "type": "COLOR_2",
        "value": {
          "type": "String",
          "value": "GREY"
        }
      }
    ]
  }
]
}'

```

Otrzymano odpowiedź o statusie 201 CREATED wraz z ID produktu w treści:

```

{
  "id": "7a5cab46-2763-4841-a74d-45fe1b7ef313"
}

```

2.1.2.4. Modyfikacja produktu

POST

/products

Parameters

No parameters

Request body

application/json

```
{
  "code": "AIR_MAX_PULSE",
  "brandId": "04abd190-656e-45e5-827a-0c8a426af8f7",
  "variations": [
    {
      "id": "42009870-2248-4f82-84cb-7965c4a2b2ff",
      "attributes": [
        {
          "type": "SIZE_EU",
          "value": {
            "type": "Integer",
            "value": 36
          }
        },
        {
          "type": "COLOR_1",
          "value": {
            "type": "String",
            "value": "WHITE"
          }
        },
        {
          "type": "COLOR_2",
          "value": {
            "type": "String",
            "value": "GREY"
          }
        }
      ]
    }
  ]
}
```

Execute

Clear

Responses

Curl

```
curl -X 'POST' \
'http://localhost:4000/products' \
-H 'accept: application/json' \
-H 'Content-Type: application/json' \
-d '{
  "code": "AIR_MAX_PULSE",
  "brandId": "04abd190-656e-45e5-827a-0c8a426af8f7",
  "variations": [
    {
      "id": "42009870-2248-4f82-84cb-7965c4a2b2ff",
      "attributes": [
        {
          "type": "SIZE_EU",
          "value": {
            "type": "Integer",
            "value": 36
          }
        },
        {
          "type": "COLOR_1",
          "value": {
            "type": "String",
            "value": "WHITE"
          }
        },
        {
          "type": "COLOR_2",
          "value": {
            "type": "String",
            "value": "GREY"
          }
        }
      ]
    }
  ]
}'
```

Request URL

http://localhost:4000/products

Server response

Code	Details
201	<div>Response body</div> <pre>{ "id": "7a5cab46-2763-4841-a74d-45fe1b7ef313" }</pre> <div>Response headers</div> <pre>content-length: 45 content-type: application/json; charset=UTF-8</pre>

Rysunek 7. Utworzenie produktu - Swagger UI

Wykonane żądanie odpowiada wywołaniu cURL (zmiana "COLOR_2" na "STEEL_GRAY"):

```
curl -X 'PUT' \
'http://localhost:4000/products/7a5cab46-2763-4841-a74d-45fe1b7ef313' \
-H 'accept: */*' \
-H 'Content-Type: application/json' \
-d '{
  "code": "AIR_MAX_PULSE",
  "brandId": "04abd190-656e-45e5-827a-0c8a426af8f7",
  "variations": [
    {
      "id": "42009870-2248-4f82-84cb-7965c4a2b2ff",
      "attributes": [
        {
          "type": "SIZE_EU",
          "value": {
            "type": "Integer",
            "value": 36
          }
        },
        {
          "type": "COLOR_1",
          "value": {
            "type": "String",
            "value": "WHITE"
          }
        },
        {
          "type": "COLOR_2",
          "value": {
            "type": "String",
            "value": "STEEL_GRAY"
          }
        }
      ]
    }
  ]
}'
```

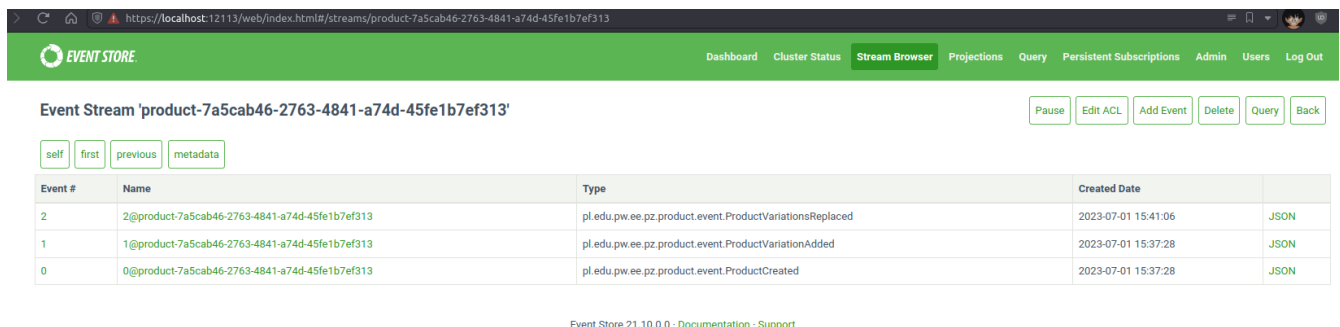
```

    "id": "d9f82ef7-95c9-4b42-b724-10589aceb2d4",
    "attributes": [
      {
        "type": "COLOR_1",
        "value": {
          "type": "String",
          "value": "WHITE"
        }
      },
      {
        "type": "COLOR_2",
        "value": {
          "type": "String",
          "value": "STEEL_GREY"
        }
      },
      {
        "type": "SIZE_EU",
        "value": {
          "type": "Integer",
          "value": 36
        }
      }
    ]
  }
}

```

Otrzymano odpowiedź o statusie 204 NO_CONTENT.

W konsoli EventStoreDB (<https://localhost:12113/>) możemy zaobserwować zdarzenia domenowe zarejestrowane dla produktu:



Event Stream 'product-7a5cab46-2763-4841-a74d-45fe1b7ef313'

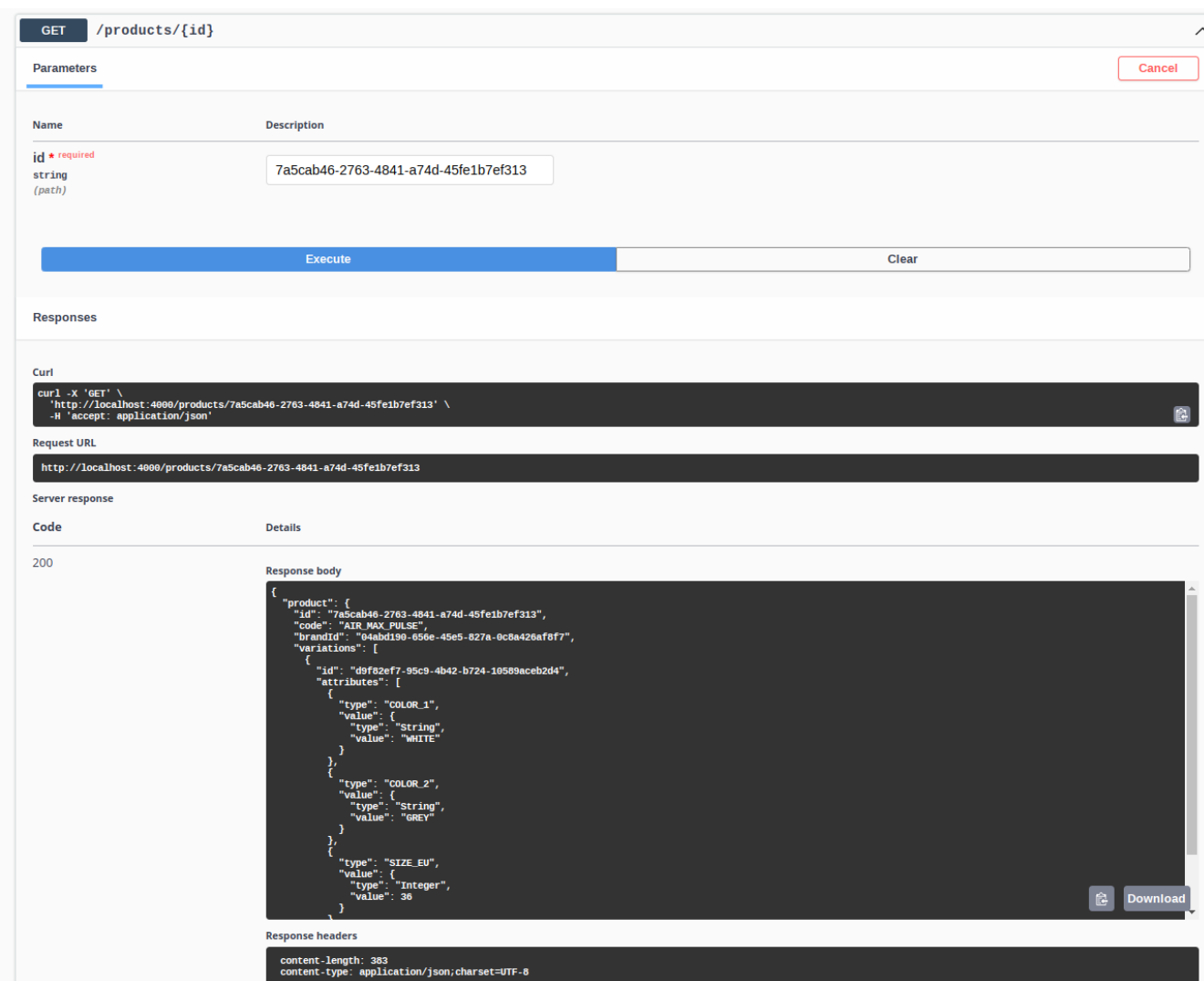
self first previous metadata

Event #	Name	Type	Created Date	
2	2@product-7a5cab46-2763-4841-a74d-45fe1b7ef313	pl.edu.pw.ee.pz.product.event.ProductVariationsReplaced	2023-07-01 15:41:06	JSON
1	1@product-7a5cab46-2763-4841-a74d-45fe1b7ef313	pl.edu.pw.ee.pz.product.event.ProductVariationAdded	2023-07-01 15:37:28	JSON
0	0@product-7a5cab46-2763-4841-a74d-45fe1b7ef313	pl.edu.pw.ee.pz.product.event.ProductCreated	2023-07-01 15:37:28	JSON

Event Store 21.10.0.0 - Documentation - Support

Rysunek 8. Zdarzenia dla produktu - EventStoreDB

2.1.2.5. Wyszukanie produktu po ID



Rysunek 9. Wyszukanie produktu po ID - Swagger UI

Wykonane żądanie odpowiada wywołaniu cURL:

```
curl -X 'GET' \
  'http://localhost:4000/products/7a5cab46-2763-4841-a74d-45fe1b7ef313' \
  -H 'accept: application/json'
```

Otrzymano odpowiedź o statusie 200 OK wraz z danymi znalezionej produktu:

```
{  "product": {    "id": "7a5cab46-2763-4841-a74d-45fe1b7ef313",    "code": "AIR_MAX_PULSE",    "brandId": "04abd190-656e-45e5-827a-0c8a426af8f7",    "variations": [      {        "id": "d9f82ef7-95c9-4b42-b724-10589aceb2d4",        "attributes": [          {            "type": "COLOR_1",            "value": {              "type": "String",              "value": "WHITE"            }          },          {            "type": "COLOR_2",            "value": {              "type": "String",              "value": "GREY"            }          }        ],        "type": "SIZE_EU",        "value": {          "type": "Integer",          "value": 36        }      }    ]  }  }
```

```
        "type": "String",
        "value": "WHITE"
    },
    {
        "type": "COLOR_2",
        "value": {
            "type": "String",
            "value": "GREY"
        }
    },
    {
        "type": "SIZE_EU",
        "value": {
            "type": "Integer",
            "value": 36
        }
    }
]
}
```

2.1.2.6. Wyszukanie produktu po kodzie

GET/products

Parameters
Cancel

Name	Description
brandId string (query)	<input type="text" value="brandId"/>
code string (query)	<input type="text" value="AIR_MAX"/>
keySetItemId integer(\$int64) (query)	<input type="text" value="keySetItemId"/>
pageSize integer(\$int64) (query)	<input type="text" value="pageSize"/>

Execute
Clear

Responses

Curl

```
curl -X 'GET' \
'http://localhost:4000/products?code=AIR_MAX' \
-H 'accept: application/json'
```

Request URL

```
http://localhost:4000/products?code=AIR_MAX
```

Server response

Code	Details
200	<div> Response body <pre>{ "products": { "type": "Single", "value": { "id": "7a5cab46-2763-4841-a74d-45fe1b7ef313", "code": "AIR_MAX_PULSE", "brandId": "04abd190-050e-45e5-827a-0c8a426af8f7", "variations": [{ "id": "fbba4116-c416-4d04-bc1c-414e44ef705a", "attributes": [{ "type": "COLOR_1", "value": { "type": "String", "value": "WHITE" } }, { "type": "COLOR_2", "value": { "type": "String", "value": "STEEL_GREY" } }] }] } }, "type": "SIZE_EU", "value": { "type": "String", "value": "42" } }</pre> Download </div> <div> Response headers <pre>content-length: 497 content-type: application/json;charset=UTF-8</pre> </div>

Rysunek 10. Wyszukanie produktu po kodzie - Swagger UI

Wykonane żądanie odpowiada wywołaniu cURL:

```
curl -X 'GET' \ 'http://localhost:4000/products?code=AIR_MAX' \ -H 'accept: application/json'
```

Otrzymano odpowiedź o statusie 200 OK wraz z danymi znalezionej produktu i informacją dot. paginacji:

```
{
  "products": {
    "type": "Single",
    "value": {
      "id": "7a5cab46-2763-4841-a74d-45fe1b7ef313",
      "code": "AIR_MAX_PULSE",
```



```

    "brandId": "04abd190-656e-45e5-827a-0c8a426af8f7",
    "variations": [
      {
        "id": "fbba4116-c416-4d04-bc1c-414e44ef785a",
        "attributes": [
          {
            "type": "COLOR_1",
            "value": {
              "type": "String",
              "value": "WHITE"
            }
          },
          {
            "type": "COLOR_2",
            "value": {
              "type": "String",
              "value": "STEEL_GREY"
            }
          },
          {
            "type": "SIZE_EU",
            "value": {
              "type": "Integer",
              "value": 36
            }
          }
        ]
      }
    ],
    "pageCount": 1,
    "page": {
      "size": 1,
      "prevKeySetId": 0,
      "nextKeySetId": 0
    },
    "itemsCount": 1
  }
}

```

2.1.2.7. Stan aplikacji po testach

Po wykonanych w/w testach, możemy zweryfikować stan danych w bazie danych PostgreSQL oraz przesłane zdarzenia w Redpanda.

2.1.2.7.1. PostgreSQL

buy_it.public.products

keyset_id	id	code	brand_id
1	7a5cab46-2763-4841-a74d-45fe1b7ef313	AIR_MAX_PULSE	04abd190-656e-45e5-827a-0c8a426af8f7

buy_it.public.brands

keyset_id	id	code
1	04abd190-656e-45e5-827a-0c8a426af8f7	NIKE

buy_it.public.products_variation_attributes

variation_id	product_id	type	value	value_type
fbba4116-c416-4d04-bc1c-414e44ef785a	7a5cab46-2763-4841-a74d-45fe1b7ef313	COLOR_1	WHITE	STRING
fbba4116-c416-4d04-bc1c-414e44ef785a	7a5cab46-2763-4841-a74d-45fe1b7ef313	COLOR_2	STEEL_GREY	STRING
fbba4116-c416-4d04-bc1c-414e44ef785a	7a5cab46-2763-4841-a74d-45fe1b7ef313	SIZE_EU	36	INTEGER

Rysunek 11. Projekcje po testach - PostgreSQL

Na powyższym rysunku możemy zaobserwować, utworzony produkt, markę oraz aktualne atrybuty produktu.

2.1.2.7.2. Redpanda

Korzystając z konsoli Webowej Redpanda (<http://localhost:18080/topics/domain-events>), możemy podejrzeć przesłane zdarzenia w ramach naszych testów:

Cluster > Topics > domain-events

3.73 kiB Size | 6 Messages | delete cleanup.policy | 7 days retention.ms | Infinite retention.bytes

Messages | Consumers | Partitions | Configuration | ACL | Documentation

PARTITION: All | START OFFSET: Newest-50 | MAX RESULTS: 50 | FILTER: ON

Offset	Partition	Timestamp	Key	Value
5	0	7/1/2023, 3:41:06 PM	7a5cab46-2763-4841-a74d-45fe1b7ef313	{"event": "eyJ0ZWZkZXI0bnN1dmFsdWU10jN9LCJhZ2dyZWdhdGVJZCI6eyJ...
4	0	7/1/2023, 3:37:28 PM	7a5cab46-2763-4841-a74d-45fe1b7ef313	{"event": "eyJ0ZWZkZXI0bnN1dmFsdWU10jN9LCJhZ2dyZWdhdGVJZCI6eyJ...
3	0	7/1/2023, 3:37:28 PM	7a5cab46-2763-4841-a74d-45fe1b7ef313	{"event": "eyJ0ZWZkZXI0bnN1dmFsdWU10jN9LCJhZ2dyZWdhdGVJZCI6eyJ...
2	0	7/1/2023, 3:30:37 PM	04abd190-656e-45e5-827a-0c8a426af8f7	{"event": "eyJ0ZWZkZXI0bnN1dmFsdWU10jN9LCJhZ2dyZWdhdGVJZCI6eyJ...
1	0	7/1/2023, 3:30:17 PM	04abd190-656e-45e5-827a-0c8a426af8f7	{"event": "eyJ0ZWZkZXI0bnN1dmFsdWU10jN9LCJhZ2dyZWdhdGVJZCI6eyJ...
0	0	7/1/2023, 3:28:56 PM	04abd190-656e-45e5-827a-0c8a426af8f7	{"event": "eyJ0ZWZkZXI0bnN1dmFsdWU10jN9LCJhZ2dyZWdhdGVJZCI6eyJ...

Total 6 items < 1 > 20 / page

Save Messages

Redpanda Console (Platform Version v23.1) (built March 22, 2023) B887CD4

Rysunek 12. Zdarzenia po testach - Redpanda