

Bazy danych 2



Temat: CRUD API sklepu internetowego

Technologie:

- **Node.js**
- **Express**
- **MongoDB**

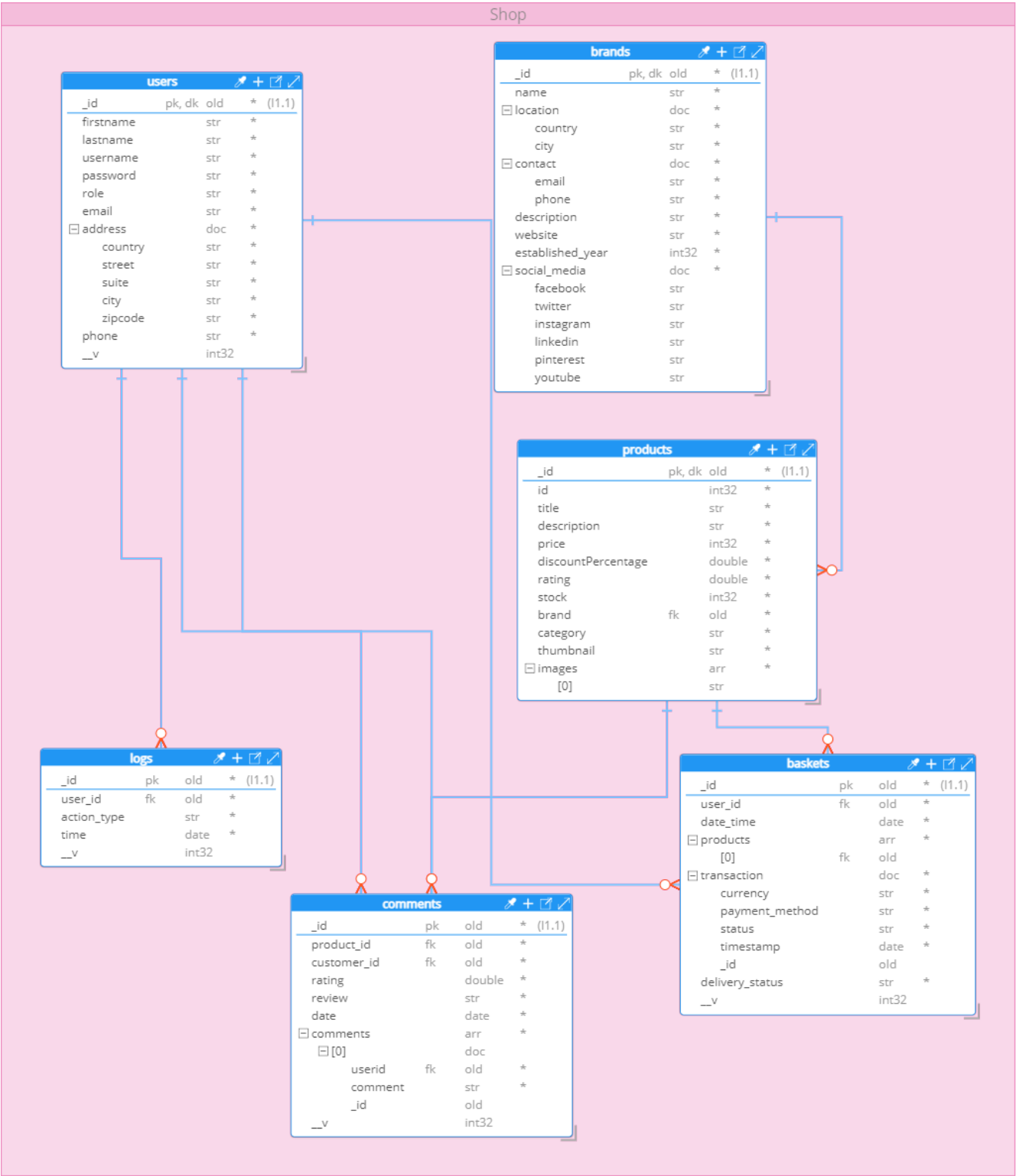
oraz biblioteki:

- **bcrypt** - Biblioteka pomagająca w haszowaniu haseł.
- **jsonwebtoken** - biblioteka służąca do tworzenia i weryfikacji tokenów sieciowych JSON (JWT).
- **mongoose** - to narzędzie do modelowania obiektów MongoDB zaprojektowane do pracy w środowisku asynchronicznym. Mongoose obsługuje Node.js i Deno (alfa).
- **mongodb-js/charts-embed-dom Wykonali:**

Piotr Śmiałek - piotrsmia@student.agh.edu.pl Michał Tomaszewski - mtomaszewski@student.agh.edu.pl

Diagram

diagram wygenerowany przy pomocy narzędzia Hackolade



Kolekcje

users

Zawiera informacje o użytkownikach sklepu internetowego

przykładowy dokument:

```
{
  "_id": {
    "$oid": "661ee379a881e0b8999aabee"
  },
  "firstname": "Ervin",
  "lastname": "Howell",
  "username": "Antonette",
  "password": "$2b$10$ARZw9AcYC0cat.csewnl/eNbRf/LKKkxTFRLi0dDSJ7PlIgwqP7oW",
  "role": "customer",
  "email": "Shanna@melissa.tv",
  "address": {
    "country": "France",
    "street": "Victor Plains",
    "suite": "Suite 879",
    "city": "Wisokyburgh",
    "zipcode": "90566-7771"
  },
  "phone": "010-692-6593 x09125"
}
```

- **_id (objectId)** - klucz główny
- **firstname (string)** - imie użytkownika
- **lastname (string)** - nazwisko użytkownika
- **username (string)** - nick użytkownika
- **password (string)** - zahasowane hasło użytkownika
- **role (string enum)** - jedna z dwóch ról obsługiwanych przez serwer (**customer** lub **admin**), ma pomóc przy chronieniu dostępu do niektórych endpointów
- **address (object)** - obiekt który trzyma pola typu string takie jak (**country**, **street**, **suite**, **city**, **zipcode**)
- **phone** - numer telefonu użytkownika

Validation rules - dla kolekcji users

```
{
  $jsonSchema: {
    additionalProperties: true,
    bsonType: 'object',
    required: [
      'firstname',
      'lastname',
      'username',
      'password',
      'email',
      'address',
      'phone'
    ],
    properties: {
      _id: {
        bsonType: 'objectId',
        description: 'must be an objectId and is required'
      },
      firstname: {
        bsonType: 'string',
        description: 'must be a string and is required'
      },
      lastname: {
        bsonType: 'string',
        description: 'must be a string and is required'
      },
      username: {
        bsonType: 'string',
        description: 'must be a string and is required'
      }
    }
  }
}
```

```
    },
    email: {
      bsonType: 'string',
      description: 'must be a string and is required',
      pattern: '.*@.*'
    },
    password: {
      bsonType: 'string',
      description: 'must be a string and is required'
    },
    role: {
      'enum': [
        'user',
        'admin',
        'customer'
      ],
      description: 'it must be user or admin'
    },
    address: {
      bsonType: 'object',
      required: [
        'country',
        'street',
        'city',
        'zipcode'
      ],
      properties: {
        country: {
          'enum': [
            'France',
            'Germany',
            'Austria',
            'Sweden',
            'South Korea',
            'China',
            'Saudi Arabia',
            'United States',
            'Pakistan',
            'Spain',
            'Vietnam',
            'Poland'
          ],
          description: 'must be a string and is required'
        },
        street: {
          bsonType: 'string',
          description: 'must be a string and is required'
        },
        suite: {
          bsonType: [
            'string',
            'null'
          ],
          description: 'must be a string or null'
        },
        city: {
          bsonType: 'string',
          description: 'must be a string and is required'
        },
        zipcode: {
          bsonType: 'string',
          description: 'must be a string and is required'
        }
      },
      description: 'must be an object with country, street, city, and zipcode fields'
    },
    phone: {
```

```
      bsonType: 'string',
      description: 'must be a string and is required'
    }
  }
}
```

products

Kolekcja zawiera wszystkie produkty dostępne w sklepie

przykładowy dokument:

```
{
  "_id": {
    "$oid": "661edcb6a881e0b8999aabcc"
  },
  "id": 6,
  "title": "MacBook Pro",
  "description": "MacBook Pro 2021 with mini-LED display may launch between September, November",
  "price": 1749,
  "discountPercentage": 11.02,
  "rating": 4.57,
  "stock": 83,
  "brand": {
    "$oid": "664e5eec1322d1e99418db6d"
  },
  "category": "laptops",
  "thumbnail": "https://cdn.dummyjson.com/product-images/6/thumbnail.png",
  "images": [
    "https://cdn.dummyjson.com/product-images/6/1.png",
    "https://cdn.dummyjson.com/product-images/6/2.jpg",
    "https://cdn.dummyjson.com/product-images/6/3.png",
    "https://cdn.dummyjson.com/product-images/6/4.jpg"
  ]
}
```

- **_id (objectId)** - klucz główny
- **title (string)** - nazwa produktu
- **description (string)** - krótki opis produktu
- **price ([int, double])** - cena produktu
- **discountPercentage ([int, double])** - % o jaki produkt jest przeceniony
- **rating ([int, double])** - rating produktu
- **stock (int)** - ilość produktów na stanie magazynu
- **brand (objectId)** - klucz obcy marki (kolekcja brands)
- **category (enum string)** - jedna ze zdefiniowanych kategorii

polia nieobowiązkowe

- **thumbnail** oraz **images** - linki do zdjęć produktów

Validation rules - dla kolekcji products

```
{
  $jsonSchema: {
    additionalProperties: true,
    bsonType: 'object',
    required: [
      'title',
      'description',
    ]
  }
}
```

```
    'price',
    'discountPercentage',
    'rating',
    'stock',
    'brand',
    'category'
  ],
  properties: {
    _id: {
      bsonType: 'objectId',
      description: 'must be an objectId and is required'
    },
    id: {
      bsonType: 'int',
      description: 'must be an integer and is optional'
    },
    title: {
      bsonType: 'string',
      description: 'must be a string and is required'
    },
    description: {
      bsonType: 'string',
      description: 'must be a string and is required'
    },
    price: {
      bsonType: [
        'int',
        'double'
      ],
      description: 'must be an integer or a double and is required'
    },
    discountPercentage: {
      bsonType: [
        'int',
        'double'
      ],
      description: 'must be an integer or a double and is required'
    },
    rating: {
      bsonType: [
        'int',
        'double'
      ],
      description: 'must be an integer or a double and is required'
    },
    stock: {
      bsonType: 'int',
      description: 'must be an integer and is required'
    },
    brand: {
      bsonType: 'objectId',
      description: 'must be a string and is required'
    },
    category: {
      'enum': [
        'smartphones',
        'food',
        'laptops',
        'fragrances',
        'skincare',
        'groceries',
        'home-decoration',
        'Example Category'
      ],
      description: 'must be one of the enum values and is required'
    },
    thumbnail: {
```

```
    bsonType: 'string',
    description: 'must be a string and is optional'
  },
  images: {
    bsonType: 'array',
    items: {
      bsonType: 'string',
      description: 'each image must be a string'
    },
    description: 'must be an array of strings and is optional'
  }
}
}
```

brands

Kolekcja zawiera wszystkie marki, których produkty są dostępne na stanie

przykładowy dokument:

```
{
  "_id": {
    "$oid": "664e5eec1322d1e99418db6d"
  },
  "name": "Apple",
  "location": {
    "country": "United States",
    "city": "Cupertino"
  },
  "contact": {
    "email": "contact@apple.com",
    "phone": "+1-800-275-2273"
  },
  "description": "Apple Inc. designs, manufactures, and markets mobile communication and media devices, personal computers, and portable digital music players.",
  "website": "https://www.apple.com",
  "established_year": 1976,
  "social_media": {
    "facebook": "https://www.facebook.com/Apple",
    "twitter": "https://twitter.com/Apple"
  }
}
```

- **_id (objectId)** - klucz główny
- **name (string)** - nazwa marki
- **location (object)** - obiekt, zawierający dokładny adres firmy
 - **country (string)** - kraj z jakiego pochodzi firma
 - **city (string)** - miasto, z którego pochodzi firma
- **contact (object)** - obiekt, zawierający kontakt do firmy
 - **email (string)** - email
 - **phone (string)** - numer telefonu
- **description (string)** - krótki opis marki
- **website (string)** - strona internetowa firmy
- **established_year (int)** - data powstania marki

- **social_media (object)** - media społecznościowe marki, gdzie można ją znaleźć
 - **facebook (string)**
 - **twitter (string)**
 - **linkedin (string)**
 - **youtube (string)**

**

Validation rules - dla kolekcji products

```
{
  $jsonSchema: {
    additionalProperties: true,
    bsonType: 'object',
    required: [
      'name',
      'location',
      'contact',
      'description',
      'established_year'
    ],
    properties: {
      _id: {
        bsonType: 'objectId',
        description: 'must be an objectId and is required'
      },
      name: {
        bsonType: 'string',
        description: 'must be a string and is required'
      },
      location: {
        bsonType: 'object',
        required: [
          'country',
          'city'
        ],
        properties: {
          country: {
            'enum': [
              'France',
              'Germany',
              'Austria',
              'Sweden',
              'South Korea',
              'China',
              'Saudi Arabia',
              'United States',
              'Pakistan',
              'Spain',
              'Vietnam',
              'Poland'
            ],
            description: 'country must exist'
          },
          city: {
            bsonType: 'string',
            description: 'must be a string and is required'
          }
        },
        description: 'must be an object with country and city fields'
      },
      contact: {
        bsonType: 'object',
        required: [
          'email',
```



```

    'phone'
  ],
  properties: {
    email: {
      bsonType: 'string',
      description: 'must be a string and is required'
    },
    phone: {
      bsonType: 'string',
      description: 'must be a string and is required'
    }
  },
  description: 'must be an object with email and phone fields'
},
description: {
  bsonType: 'string',
  description: 'must be a string and is required'
},
website: {
  bsonType: 'string',
  description: 'must be a string'
},
established_year: {
  bsonType: 'int',
  description: 'must be an integer and is required'
},
social_media: {
  bsonType: 'object',
  properties: {
    facebook: {
      bsonType: 'string',
      description: 'must be a string'
    },
    twitter: {
      bsonType: 'string',
      description: 'must be a string'
    }
  },
  description: 'must be an object with optional facebook and twitter fields'
}
}
}
}

```

comments

Kolekcja zawiera komentarze i podkomentarze użytkowników do danych produktów

przykładowy dokument:

```

{
  "_id": {
    "$oid": "665cc9700263fbb3b2f79b38"
  },
  "product_id": {
    "$oid": "661edcb6a881e0b8999aabe2"
  },
  "customer_id": {
    "$oid": "661ee379a881e0b8999aabee"
  },
  "rating": 2.5,
  "review": "Kupiłem go ponownie i jakość znacząco spadła względem wcześniejszego zakupu",
  "date": {
    "$date": "2024-06-02T19:35:12.954Z"
  },
}

```

```

    "comments": [
      {
        "userid": {
          "$oid": "661ee379a881e0b8999aabcd"
        },
        "comment": "NIe zgadzam sie z ta opinia, u wszystko jest w porzadku!!!",
        "_id": {
          "$oid": "665ccd5c01b697f932e229e0"
        }
      },
      {
        "userid": {
          "$oid": "661ee379a881e0b8999aabcd"
        },
        "comment": "Zmieniam zdanie! Miał pan racje, produkt nie spelnia wymagan",
        "_id": {
          "$oid": "665ccdc9b44a40a6224935af"
        }
      }
    ],
    "__v": 2
  }

```

- **_id (objectId)** - klucz główny
- **product_id (objectId)** - id produktu do którego została wystawiona opinia
- **customer_id (objectId)** - id użytkownika który wystawił opinię główną
- **rating ([int, double])** - opinia w zakresie od 0 do 5 wystawiona produktowi
- **review (string)** - komentarz do opinii
- **date (date)** - data wystawienia opinii
- **comments (array)** - lista obiektów (komentarze do opinii) w ramach jednego obiektu listy:
 - **userid (objectId)** - id użytkownika, który skomentował OPINIĘ
 - **comment (string)** - komentarz tego użytkownika

Validation rules - dla kolekcji comments

```

{
  $jsonSchema: {
    additionalProperties: true,
    bsonType: 'object',
    required: [
      '_id',
      'product_id',
      'customer_id',
      'rating',
      'review',
      'date'
    ],
    properties: {
      _id: {
        bsonType: 'objectId',
        description: 'must be an objectId and is required'
      },
      product_id: {
        bsonType: 'objectId',
        description: 'must be an objectId and is required'
      },
      customer_id: {
        bsonType: 'objectId',
        description: 'must be an objectId and is required'
      },
      rating: {
        bsonType: [
          'int',

```

```

        'double'
      ],
      minimum: 1,
      maximum: 5,
      description: 'must be an integer between 1 and 5 and is required'
    },
    review: {
      bsonType: 'string',
      description: 'must be a string and is required'
    },
    date: {
      bsonType: 'date',
      description: 'must be a date and is required'
    },
    comments: {
      bsonType: 'array',
      items: {
        bsonType: 'object',
        required: [
          'userid',
          'comment'
        ],
        properties: {
          userid: {
            bsonType: 'objectId',
            description: 'must be an objectId and is required'
          },
          comment: {
            bsonType: 'string',
            description: 'must be a string and is required'
          }
        }
      },
      description: 'must be an array of objects containing userid and comment'
    }
  }
}
}
}

```

baskets

Koszyki z produktami użytkowników

przykładowy dokument:

```

{
  "_id": {
    "$oid": "663fcf269d51de47edb6f0b5"
  },
  "user_id": {
    "$oid": "661ee379a881e0b8999aabee"
  },
  "date_time": {
    "$date": "2020-03-01T15:30:00.000Z"
  },
  "products": [
    {
      "$oid": "661edcb6a881e0b8999aabe4"
    },
    {
      "$oid": "661edcb6a881e0b8999aabd9"
    },
    {
      "$oid": "661edcb6a881e0b8999aabd5"
    }
  ]
}

```

```
  ],
  "transaction": {
    "currency": "PLN",
    "payment_method": "PayPal",
    "status": "in_progress",
    "timestamp": {
      "$date": "2023-05-01T15:30:00.000Z"
    }
  },
  "delivery_status": "order_placed"
}
```

- **_id (objectId)** - klucz główny
- **user_id (objectId)** - id użytkownika, który zamówił koszyk
- **date_time (date)** - data złożenia zamówienia
- **products (array of objectId's)** - lista z id'kami produktów, jakie zamówił klient
 - **product_id (objectId)** - id produktu (klucz obcy z kolekcji products)
- **transactiob (object)** - szczegóły transakcji
 - **currency (enum string)** - waluta w jakiej dokonał transakcji
 - **payment_method (enum string)** - sposób zapłacenia za zakupy
 - **status (enum string)** - status transakcji
 - **timestamp (date)** - data transakcji
- **delivery_status (enum string)** - status przesyłki

Validation rules - dla kolekcji comments

```
{
  $jsonSchema: {
    additionalProperties: true,
    bsonType: 'object',
    required: [
      '_id',
      'user_id',
      'date_time',
      'products',
      'transaction',
      'delivery_status'
    ],
    properties: {
      _id: {
        bsonType: 'objectId',
        description: 'must be an objectId and is required'
      },
      user_id: {
        bsonType: 'objectId',
        description: 'must be an objectId and is required'
      },
      date_time: {
        bsonType: 'date',
        description: 'must be a date and is required'
      },
      products: {
        bsonType: 'array',
        minItems: 1,
        items: {
          bsonType: 'objectId',
          description: 'each product must be an objectId'
        }
      }
    }
  }
}
```

```

    },
    description: 'must be an array of objectIds and is required'
  },
  transaction: {
    bsonType: 'object',
    required: [
      'currency',
      'payment_method',
      'status',
      'timestamp'
    ],
    properties: {
      currency: {
        'enum': [
          'PLN',
          'EUR',
          'USD'
        ],
        description: 'can only be one of the enum values and is required'
      },
      payment_method: {
        bsonType: 'string',
        description: 'must be a string and is required'
      },
      status: {
        'enum': [
          'rejected',
          'completed',
          'canceled',
          'in_progress'
        ],
        description: 'must be a string and is required'
      },
      timestamp: {
        bsonType: 'date',
        description: 'must be a date and is required'
      }
    },
    description: 'must be an object and is required'
  },
  delivery_status: {
    'enum': [
      'order_placed',
      'delivered',
      'on_the_way'
    ],
    description: 'must be \'order_placed\' and is required'
  }
}
}
}

```

logs

kollekcja zawierająca proste informacje na temat logowania / wylogowywania się użytkowników ze strony sklepu

przykładowy dokument:

```

{
  "_id": {
    "$oid": "663fc5499d51de47edb6f022"
  },
  "user_id": {
    "$oid": "661ee379a881e0b8999aabcd"
  },
}

```

```
"action_type": "login",
"time": {
  "$date": "2021-12-08T15:00:18.000Z"
}
}
```

- **_id (objectId)** - klucz główny
- **user_id (objectId)** - id użytkownika, który zamówił koszyk
- **action_type (enum string)** - login / logout
- **time (date)** - czas i data akcji

Validation rules - dla kolekcji comments

```
{
  $jsonSchema: {
    additionalProperties: true,
    bsonType: 'object',
    required: [
      'user_id',
      'action_type',
      'time'
    ],
    properties: {
      user_id: {
        bsonType: 'objectId',
        description: 'must be an objectId and is required'
      },
      action_type: {
        bsonType: 'string',
        'enum': [
          'login',
          'logout'
        ],
        description: 'must be one of the enum values (login/logout) and is required'
      },
      time: {
        bsonType: 'date',
        description: 'must be a date and is required'
      }
    }
  }
}
```

Logowanie

Każdy użytkownik w tabeli users posiada email i zahaslowane hasło. Do każdego użytkownika jest także przypisana rola, jest ich dwie: admin i customer. Większość endpointów na serwerze jest chrononych (tylko niektóre są dostępne dla gości).

Logowanie odbywa się przez endpoint:

POST /login w body podajemy password

```
POST http://localhost:8000/login
Content-Type: application/json

{
  "email": "Julianne.OConner@kory.org",
```

```
"password": "haslo123"
}
```

W odpowiedzi dostajemy token, którego używamy przy chronionych endpointach.

Logowanie jest realizowane przy pomocy jwt token'a, przy pomocy middleware.

```
function authenticateToken(req, res, next){
  const authHeader = req.headers['authorization']
  // we split because its: BEARER <token>
  const token = authHeader && authHeader.split(' ')[1]

  if(token == null){
    res.status(400)
  }
  jwt.verify(token, process.env.ACCESS_TOKEN_SECRET, (err, user) =>{
    if(err) return res.sendStatus(403)

    req.user = user;
    next()
  })
}

function authorizeRoles(allowedRoles) {
  return (req, res, next) => {
    if (!allowedRoles.includes(req.user.user.role)) {
      return res.status(403).send('Access forbidden: insufficient rights');
    }
    next();
  }
}
```

Customer

Enpointy do jakich ma dostęp customer:

POST /create-basket POST /my-shopping GET /search-products GET /products/:productId/all-reviews GET /all-products POST /add-review GET /all-brands GET /search-brands GET /my-reviews