

Projekt z baz danych

Sebastian Misztal, Maciej Nessel

Temat: Sklep internetowy sprzedający ubrania

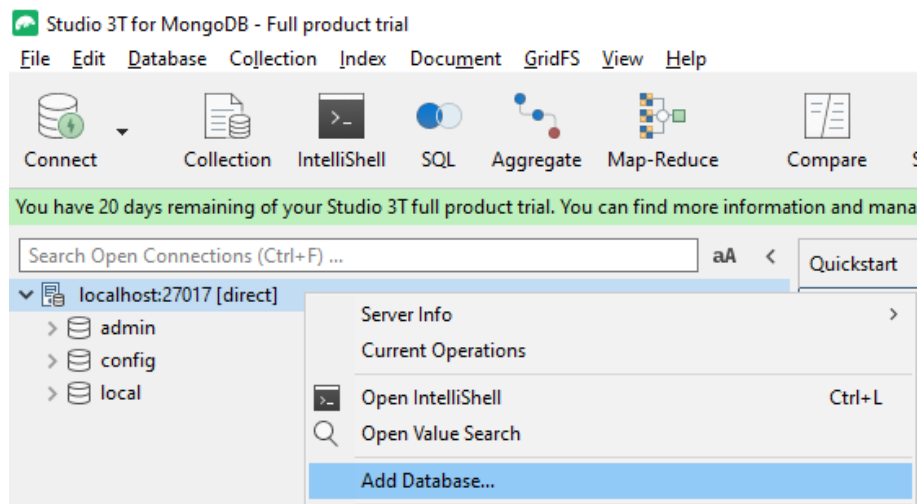
Technologie

- MongoDB
- Frontend: Angular, HTML, CSS, TypeScript
- Backend: Node.js, Express.js, Mongoose, JWT

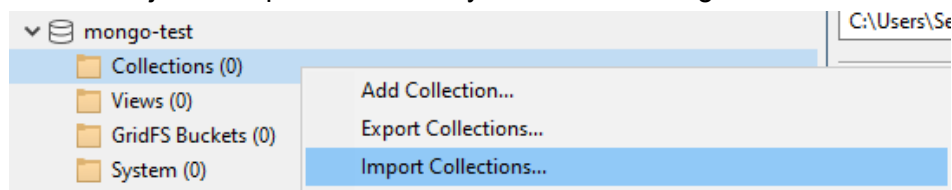
Uruchomienie

Baza Danych

0. Przewodnik został stworzony z myślą o używaniu Studio 3T.
1. Pierwszym krokiem jest pobranie, a następnie zainstalowanie mongoDB lokalnie na swoim urządzeniu.
2. Następnie, po połączeniu należy utworzyć nową bazę danych o nazwie mongo-test.



3. Kolejnym krokiem jest zaimportowanie danych z folderu mongo.



Import

Import format

To start the import process, please select an input format

☐ JSON - Mongo shell, Studio 3T, mongoexport

Import collections from JSON formatted files created with Studio 3T or 'mongoexport' compatible tools.

☐ CSV

Import a collection from a CSV formatted file created with Studio 3T or other applications like MS Excel. You will be able to preview the imported data as well as configure the import process.

☐ SQL database

Import data from a live SQL database.

☒ BSON - mongodump folder

Import databases and collections from a mongodump folder created with Studio 3T or MongoDB's 'mongodump' command-line tool using the 'mongorestore' executable. It expects a dump root folder with a subfolder for each database containing two files per collection: 'collection.bson' and 'collection.metadata.json', or 'collection.bson.gz' and 'collection.metadata.json.gz' if compression was used. Additionally, an 'oplog.bson' file generated with 'oplog.bson' can exist in the root folder.

☐ BSON - mongodump archive

Import databases and collections from a mongodump archive file created with Studio 3T or MongoDB's 'mongodump --archive' command line tool using the 'mongorestore' executable. Please note that the content of a mongodump archive cannot be browsed. It is still possible to select any single database or collection (if available).

☐ Another collection

Import data from another collection, allowing you to specify how duplicates are handled.

Configure

Cancel

Source folder

Select folder

Select mongodump folder

Ten komputer > Dokumenty >

Przeszukaj: Dokumenty

Organizuj > Nowy folder

| | Nazwa | Data modyfikacji | Typ | Rozmiar |
|--|-------|------------------|---------------|---------|
| | | 06.03.2022 07:04 | Folder plików | |
| | | 26.04.2022 18:45 | Folder plików | |
| | | 05.03.2022 19:27 | Folder plików | |
| | | 23.02.2022 00:16 | Folder plików | |
| | mongo | 25.05.2022 02:24 | Folder plików | |
| | | 17.05.2022 13:24 | Folder plików | |
| | | 01.04.2022 23:38 | Folder plików | |
| | | 22.02.2022 22:51 | Folder plików | |

Folder: mongo

Wybierz folder

Anuluj

Save task | Save task as... | Schedule | Run

Source folder

Run

\\Documents\mongo

Import sources

Available databases & collections

☒ mongo-test

☒ items

☒ orders

☒ users

Backend

1. W folderze backend edytować plik **.env** wpisując odpowiednie dane:

DB_URL=*{url bazy danych }*

JWT_SECRET=*{prywatny klucz JWT Tokenu, wygenerowany dla serwera, nie udostępniany publicznie}*

2. W folderze **backend** wpisać następujące komendy :

npm i

npm start

Frontend

W folderze **frontend** wpisać następujące komendy:

npm i

npm start

Serwis działa lokalnie na porcie 4200: **http://localhost:4200/**

Struktura bazy danych

Kolekcja przechowująca dane Przedmiotów

```
{
  "_id": {
    "$oid": "628d72cafc13ae06b6000066"
  },
  "name": "Gas Relief",
  "type": "Spodnie",
  "quantity_in_stock": 500,
  "gender": [
    "kid",
    null,
    null
  ],
  "size": "XL",
  "price": 2639.39,
  "imageUrl":
  "https://artsmidnorthcoast.com/wp-content/uploads/2014/05/no-image-available-icon-6.png"
}
```

Kolekcja przechowująca dane Użytkowników

```
{
  "_id": {
    "$oid": "628d7553773a291d01b84414"
  },
  "userName": "Uzytkownik3",
  "email": "takt@tak.tak",
  "password": "$2a$10$ZLkMruRvyYh/r42bkj7IP.OZ/rCgQ1hin3DWFakNkNYIG7XDkwqbq",
  "firstName": "Baron",
  "lastName": "Lios",
  "createdAt": {
    "$date": {
      "$numberLong": "1653437779709"
    }
  },
  "updatedAt": {
    "$date": {
      "$numberLong": "1653437779709"
    }
  },
  "___v": 0
}
```

Kolekcja przechowująca dane Zamówień

```
{
  "_id": {
    "$oid": "628d75cc773a291d01b84452"
  },
  "user_id": "628d756f773a291d01b84416",
  "date": {
    "$date": {
      "$numberLong": "1653437900752"
    }
  },
  "status": "unpaid",
  "products": [
    {
      "item_id": "628d72cefc13ae06b6000430",
      "quantity": 7,
      "price": 1195.72,
      "size": "S",
      "_id": {
        "$oid": "628d75cc773a291d01b84453"
      }
    },
    {
      "item_id": "628d72cefc13ae06b6000414",
      "quantity": 5,
      "price": 1586.98,
      "size": "L",
      "_id": {
        "$oid": "628d75cc773a291d01b84454"
      }
    },
    {
      "item_id": "628d72ccfc13ae06b600023c",
      "quantity": 31,
      "price": 1525.43,
      "size": "L",
      "_id": {
        "$oid": "628d75cc773a291d01b84455"
      }
    }
  ],
  "__v": 0
}
```

API

** Do części endpointów wymagany jest token **{{auth-token}}** wysyłany przez użytkownika w headers. Po poprawnym wprowadzeniu danych logowania jest on generowany przez backend i wysyłany do klienta.*

/item/

POST /item/ zwraca listę produktów używając paginację oraz filtrowanie danych

Body:

```
{
  "page": "1",
  "limit": "1",
  "name": "A",
  "type": "Koszule",
  "size": "S",
  "gender": ["man"],
  "price_min": "0",
  "price_max": "10000"
}
```

Result:

```
{
  "response": [
    {
      "_id": "628d72cbfc13ae06b6000109",
      "name": "Anti Aging",
      "type": "Koszule",
      "quantity_in_stock": 23,
      "gender": [
        null,
        "man",
        "woman"
      ],
      "size": "S",
      "price": 327.22,
      "imageUrl":
        "https://artsmidnorthcoast.com/wp-content/uploads/2014/05/no-image-available-icon-6.png"
    }
  ],
  "status": true
}
```

POST /item/read/ zwraca informacje o produkcie o wysłanym w body id

Body:

```
{
  "itemID": "628d72cbfc13ae06b6000109"
}
```

Result:

```
{
  "response": {
    "_id": "628d72cbfc13ae06b6000109",
    "name": "Anti Aging",
    "type": "Koszule",
    "quantity_in_stock": 23,
    "gender": [
      null,
      "man",
      "woman"
    ],
    "size": "S",
    "price": 327.22,
    "imageURL":
      "https://artsmidnorthcoast.com/wp-content/uploads/2014/05/no-image-available-icon-6.png"
  },
  "status": true
}
```

POST /item/create/ dodaje nowy produkt do bazy danych

Authentication: {{ auth-token }}

Body:

```
{
  "name": "Koszula w krate",
  "description": "Idealna na lato",
  "type": "Koszula",
  "quantity_in_stock": "20",
  "gender": ["man"],
  "size": "XL",
  "price": "69.99",
  "imageURL": "exampleurl.pl/image.img"
}
```

POST /item/update/ aktualizuje dane produktu (wysyłamy w body te dane które uległy zmianie)

Authentication: {{ auth-token }}

Body:

```
{
  "itemID": "628d72cafc13ae06b6000064",
  "name": "Koszula w krate",
  "description": "Idealna na lato.",
  "type": "Koszula",
  "quantity_in_stock": "20",
  "gender": ["man"],
  "size": "XL",
  "price": "69.99",
  "imageURL": "exampleurl.pl/image.img"
}
```

POST /item/delete/ usuwa produkt z bazy

Authentication: {{ auth-token }}

Body:

```
{
  "itemID": "628d72cafc13ae06b6000064"
}
```

/order/

POST /order/ zwraca listę zamówień, zrealizowanych przez danego użytkownika używając paginację

Authentication: {{ auth-token }}

Body:

```
{
  "page": 1,
  "limit": 5
}
```

Result:

```
{
  "history": [
    {
      "order_id": "628d82c574b1ea3691f11913",
      "date": "2022-05-25T01:13:41.926Z",
      "status": "unpaid",
      "products": [
        {
          "name": "Koszula w kratę",
          "price": 69.99,
          "quantity": 1
        },
        {
          "name": "Satohap",
          "price": 924.73,
          "quantity": 2
        }
      ],
      "resultPrice": 1919.45
    }
  ],
  "pageAmount": 1,
  "status": true
}
```


POST /order/create/ dodaje nowe zamówienie

Authentication: {{ auth-token }}

Body:

```
{
  "products": [
    {
      "item_id": "628d802074b1ea3691f11903",
      "quantity": "1",
      "size": "XL"
    },
    {
      "item_id": "628d72cafc13ae06b6000065",
      "quantity": "2",
      "size": "S"
    }
  ]
}
```

/user/

POST /user/register/ rejestracja nowego użytkownika

Body:

```
{
  "userName": "wLodek22",
  "password": "1234567890",
  "email": "wLodek22@wp.pl",
  "firstName": "WLodek",
  "lastName": "Nowak"
}
```

Result:

```
{
  "message": "User registered successfully",
  "status": true
}
```

POST /user/login/ logowanie

Body:

```
{
  "userName": "wlodek22",
  "password": "1234567890"
}
```

Result:

```
{
  "message": "Logged.",
  "status": true
}
```

Po poprawnym wprowadzeniu danych logowania klient otrzyma token autoryzacyjny:

headers: {{auth - token}}

GET /user/auth/ zwraca dane aktualnie zalogowanego użytkownika

Authentication: {{ auth-token }}

Result:

```
{
  "firstName": "Wlodek",
  "lastName": "Nowak",
  "status": true,
  "message": "Ok"
}
```