



**AKADEMIA GÓRNICZO-HUTNICZA IM. STANISŁAWA STASZICA W KRAKOWIE**

**WYDZIAŁ INFORMATYKI, ELEKTRONIKI I TELEKOMUNIKACJI**

**KATEDRA INFORMATYKI**

## **BAZY DANYCH**

Technologia: **Python + Django**

Serwer BD: **MS Sql**

Repository: <https://github.com/robertcialowicz/bazydanych2.git>

Autorzy:

*Chwała Paweł <pchwala@student.agh.edu.pl>*

*Ciałowicz Robert <robcial@student.agh.edu.pl>*

*Kozaczekiewicz Łukasz <kozaczki@student.agh.edu.pl>*

*Szpila Magdalena <mszpila@student.agh.edu.pl>*

Kraków, 2020

## Spis treści

Opis .....	3
Uruchomienie .....	4
REST API .....	5
Categories.....	5
GET /api/Categories/ .....	5
POST /api/Categories/ .....	5
GET /api/Categories/{CategoryID} .....	6
PUT /api/Categories/{CategoryID} .....	6
DELETE /api/Categories/{CategoryID}.....	6
Suppliers.....	7
GET /api/Suppliers/ .....	7
POST /api/Suppliers/ .....	7
GET /api/Suppliers/{SupplierID} .....	8
PUT /api/Suppliers/{SupplierID} .....	8
DELETE /api/Suppliers/{SupplierID} .....	8
Products .....	9
GET /api/Products/ .....	9
POST /api/Products/ .....	9
GET /api/Products/{ProductID}.....	10
PUT /api/Products/{ProductID} .....	10
DELETE /api/Products/{ProductID} .....	10

## Opis

Aplikacja napisana w języku Python z wykorzystaniem Django Rest Framework.

W projekcie wykorzystano serwer bazy danych MSSql.

Przykładową bazą, na której wykonywane będą operacje jest baza Northwind.

Celem projektu jest zaimplementowanie operacji CRUD na dowolnej tabeli, operacji składania zamówienia oraz możliwość tworzenia raportów (do zdefiniowania).

## Uruchomienie

W lokalizacji `./src` znajduje się plik `docker-compose.yml`. Aby wystartować aplikację należy z konsoli wywołać `docker-compose up --build`.

Uruchomią się dwa obrazy dockerowe zdefiniowane w pliku `yml`: jeden z serwerem MSSql z bazą danych Northwind oraz drugi z aplikacją pythonową.

### UWAGA!

W obecnej wersji aplikacja nie działa na systemach platformy Windows.

## REST API

Ten rozdział opisuje dostępne publicznie endpointy tworzonej aplikacji.

API systemu jest dostępne pod adresem /api/

### Categories

Ten rozdział opisuje użycie endpointu /api/Categories/

GET /api/Categories/

Zwraca listę kategorii.

*Przykładowa odpowiedź:*

```
{
  "Categories": [
    {
      "categoryid": 1,
      "categoryname": "Beverages",
      "description": "Soft drinks, coffees, teas, beers, and ales",
      "picture": "FRwvAAIAAAANAA4AFAAhAP...."
    },
    {
      "categoryid": 2,
      "categoryname": "Condiments",
      "description": "Sweet and savory sauces, relishes, spreads, and seasonings",
      "picture": "FRwvAAIAAAANAA4AFAAhAP...."
    },
    {
      "categoryid": 3,
      "categoryname": "Confections",
      "description": "Desserts, candies, and sweet breads",
      "picture": "FRwvAAIAAAANAA4AFAAhAP...."
    }
  ]
}
```

POST /api/Categories/

Tworzy nową kategorię.

*Przykładowe zapytanie:*

```
{
  "category": {
    "categoryname": "Przykład",
    "description": "Przykład"
  }
}
```

*Przykładowa odpowiedź:*

```
{
  "success": "category 'Przykład' created successfully"
}
```

GET /api/Categories/{CategoryID}

Zwraca pojedynczą kategorię wskazaną przez CategoryID.

*Przykładowa odpowiedź:*

```
{
  "Categories": {
    "categoryid": 4,
    "categoryname": "Dairy Products",
    "description": "Cheeses",
    "picture": "FRwvAAIAAAANAA4AFAAhAP....."
  }
}
```

PUT /api/Categories/{CategoryID}

Aktualizuje pola kategorii wskazanego przez CategoryID. Wszystkie pola są opcjonalne.

*Przykładowe zapytanie:*

```
{
  "category": {
    "categoryname": "ChangedName"
  }
}
```

*Przykładowa odpowiedź:*

```
{
  "success": "Category 'ChangedName' updated successfully"
}
```

DELETE /api/Categories/{CategoryID}

Kasuje kategorię ze wskazanym CategoryID.

*Przykładowa odpowiedź:*

```
{
  "message": "category with id `9` has been deleted."
}
```

## Suppliers

Ten rozdział opisuje użycie endpointu `/api/Suppliers/`

GET `/api/Suppliers/`

Zwraca listę dostawców.

Przykładowa odpowiedź:

```
{
  "Suppliers": [
    {
      "supplierid": 1,
      "companyname": "Exotic Liquids",
      "contactname": "Charlotte Cooper",
      "contacttitle": "Purchasing Manager",
      "address": "49 Gilbert St.",
      "city": "London",
      "region": null,
      "postalcode": "EC1 4SD",
      "country": "UK",
      "phone": "(171) 555-2222",
      "fax": null,
      "homepage": null
    },
    {
      "supplierid": 2,
      "companyname": "New Orleans Cajun Delights",
      "contactname": "Shelley Burke",
      "contacttitle": "Order Administrator",
      "address": "P.O. Box 78934",
      "city": "New Orleans",
      "region": "LA",
      "postalcode": "70117",
      "country": "USA",
      "phone": "(100) 555-4822",
      "fax": null,
      "homepage": "#CAJUN.HTM#"
    },
    {
      "supplierid": 3,
      "companyname": "Grandma Kelly's Homestead",
      "contactname": "Regina Murphy",
      "contacttitle": "Sales Representative",
      "address": "707 Oxford Rd.",
      "city": "Ann Arbor",
      "region": "MI",
      "postalcode": "48104",
      "country": "USA",
      "phone": "(313) 555-5735",
      "fax": "(313) 555-3349",
      "homepage": null
    }
  ]
}
```

POST `/api/Suppliers/`

Tworzy nowego dostawcę.

Przykładowe zapytanie:

```
{
  "supplier": {
    "companyname": "Przykład",
    "contactname": "Przykład",
    "city": "Krakow"
  }
}
```

```
}
```

*Przykładowa odpowiedź:*

```
{
  "success": "supplier 'Przykład' created successfully"
}
```

GET /api/Suppliers/{SupplierID}

Zwraca pojedynczego dostawcę wskazanego przez SupplierID.

*Przykładowa odpowiedź:*

```
{
  "Supplier": {
    "supplierid": 30,
    "companyname": "Przykład",
    "contactname": "Przykład",
    "contacttitle": null,
    "address": null,
    "city": "Krakow",
    "region": null,
    "postalcode": null,
    "country": null,
    "phone": null,
    "fax": null,
    "homepage": null
  }
}
```

PUT /api/Suppliers/{SupplierID}

Aktualizuje pola dostawcy wskazanego przez SupplierID. Wszystkie pola są opcjonalne.

*Przykładowe zapytanie:*

```
{
  "supplier": {
    "address": "Mogilska 100",
    "contacttitle": "Przykład"
  }
}
```

*Przykładowa odpowiedź:*

```
{
  "success": "Supplier 'Przykład' updated successfully"
}
```

DELETE /api/Suppliers/{SupplierID}

Kasuje dostawcę ze wskazanym SupplierID.

*Przykładowa odpowiedź:*

```
{
  "message": "supplier with id `30` has been deleted."
}
```



## Products

Ten rozdział opisuje użycie endpointu `/api/Products/`

GET `/api/Products/`

Zwraca listę produktów.

*Przykładowa odpowiedź:*

```
{
  "Products": [
    {
      "productid": 1,
      "productname": "Chai",
      "supplierid": 1,
      "categoryid": 1,
      "quantityperunit": "10 boxes x 20 bags",
      "unitprice": "18.0000",
      "unitsinstock": 39,
      "unitsonorder": 0,
      "reorderlevel": 10,
      "discontinued": false
    },
    {
      "productid": 2,
      "productname": "Chang",
      "supplierid": 1,
      "categoryid": 1,
      "quantityperunit": "24 - 12 oz bottles",
      "unitprice": "19.0000",
      "unitsinstock": 17,
      "unitsonorder": 40,
      "reorderlevel": 25,
      "discontinued": false
    },
    {
      "productid": 3,
      "productname": "Aniseed Syrup",
      "supplierid": 1,
      "categoryid": 2,
      "quantityperunit": "12 - 550 ml bottles",
      "unitprice": "10.0000",
      "unitsinstock": 13,
      "unitsonorder": 70,
      "reorderlevel": 25,
      "discontinued": false
    }
  ]
}
```

POST `/api/Products/`

Tworzy nowy produkt.

*Przykładowe zapytanie:*

```
{
  "product": {
    "productname": "nozyczki",
    "supplierid": 2,
    "categoryid": 3,
    "discontinued": false
  }
}
```

*Przykładowa odpowiedź:*

```
{
  "success": "product 'nozyczki' created successfully"
}
```

GET /api/Products/{ProductID}

Zwraca pojedynczy produkt wskazany przez ProductID.

*Przykładowa odpowiedź:*

```
{
  "Product": {
    "productid": 1,
    "productname": "Chai",
    "supplierid": 1,
    "categoryid": 1,
    "quantityperunit": "10 boxes x 20 bags",
    "unitprice": "18.0000",
    "unitsinstock": 39,
    "unitsonorder": 0,
    "reorderlevel": 10,
    "discontinued": false
  }
}
```

PUT /api/Products/{ProductID}

Aktualizuje pola produktu wskazanego przez ProductID. Wszystkie pola są opcjonalne.

*Przykładowe zapytanie:*

```
{
  "product": {
    "unitsinstock": 38
  }
}
```

*Przykładowa odpowiedź:*

```
{
  "success": "Product 'Chai' updated successfully"
}
```

DELETE /api/Products/{ProductID}

Kasuje produkt ze wskazanym ProductID

*Przykładowa odpowiedź:*

XXX