Database System (API)

 # Secondary school of electrotechnical engineering Jecna 30

 # 2022-2023

 # Matěj Šturma

 # C4c

Requirements

- **Node.js**
- **An API platform software, for example Postman.**
- **MySQL or MariaDB**

Starting the application

1. Create the database using the SQL script located at **/backend/sql/script.sql**
2. Run **npm install** or **npm i** inside the backend folder. This will install all the dependency packages necessary for the application.
3. Configure the **.env** file located inside the backend folder.
4. Run **npm run dev** inside the backend folder. You should see **Server started on port 4000** in the console.

Using the API

Each table has its own API path that can be used to **create**, **read**, **update**, **delete** and **import** data. There's 5 tables in total (see Database for more information).

The **create** function can be called with a **POST** request on http://localhost:<port>/<table_name> and takes JSON as a request body with parameters for each column (see Database for more information).

Example:

```
{
        "name":"Black T-shirt",
        "price":225,
        "size":"L"
}
```

Example JSON inserted into the **Produkt** table.

The **readAll** function can be called with a **GET** request on the same URI. It returns all items from the specified table.

The **readByID** function can be called with a **GET** request on http://localhost:<port>/<table_name>/<ID>. It takes the specified ID as a request parameter and returns an item with the ID.

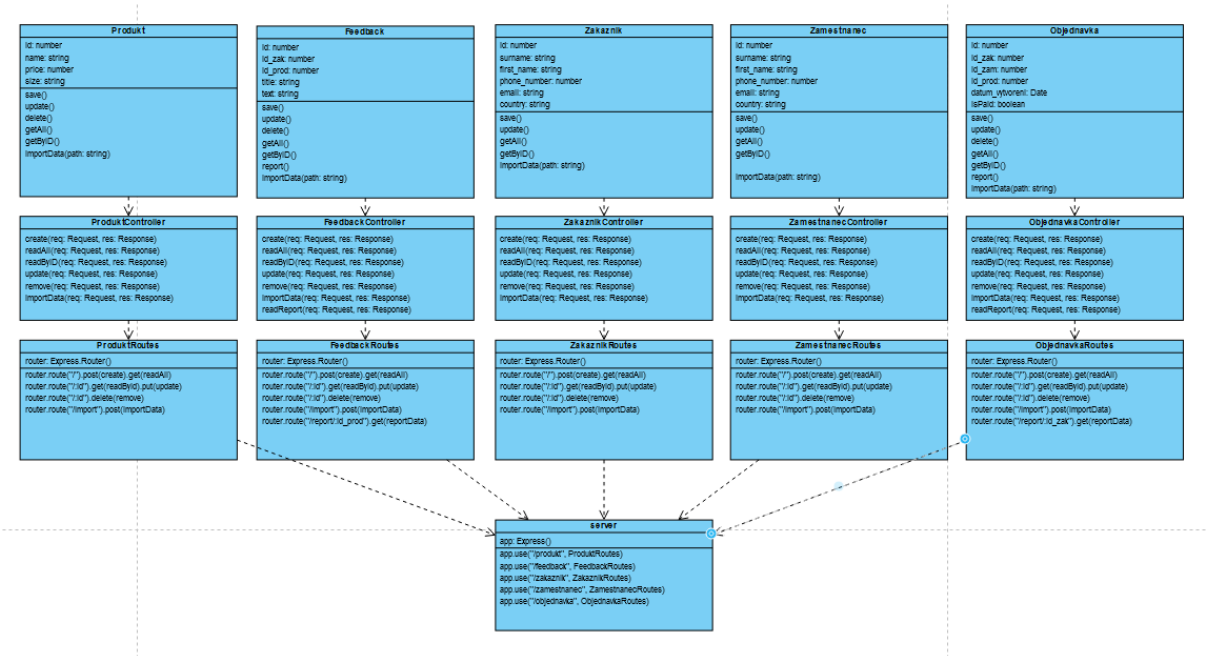
The **update** function can be called with a **PUT** request on the same URI and takes JSON as a request body with parameters for each column (see Database for more information). The ID you specify is a request parameter.

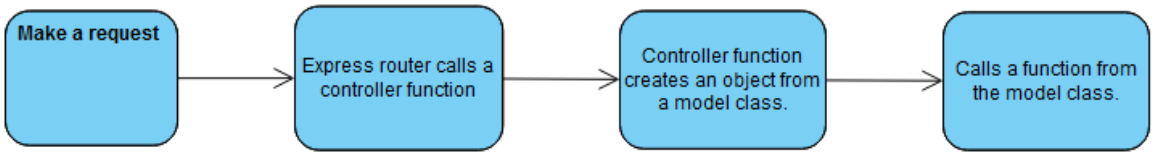
The **remove** function can be called with a **DELETE** request on the same URI. It takes the specified ID as a request parameter and deletes an item with the ID.

The **importData** function can be called with a **POST** request on http://localhost:<port>/<table_name>/import and takes JSON as a request body with **1 parameter "path"** (a local path to the CSV file). To use the importData function, you have to **make a CSV file** first. The file must consist of **at least one** line with values for each column in the table.

Architecture

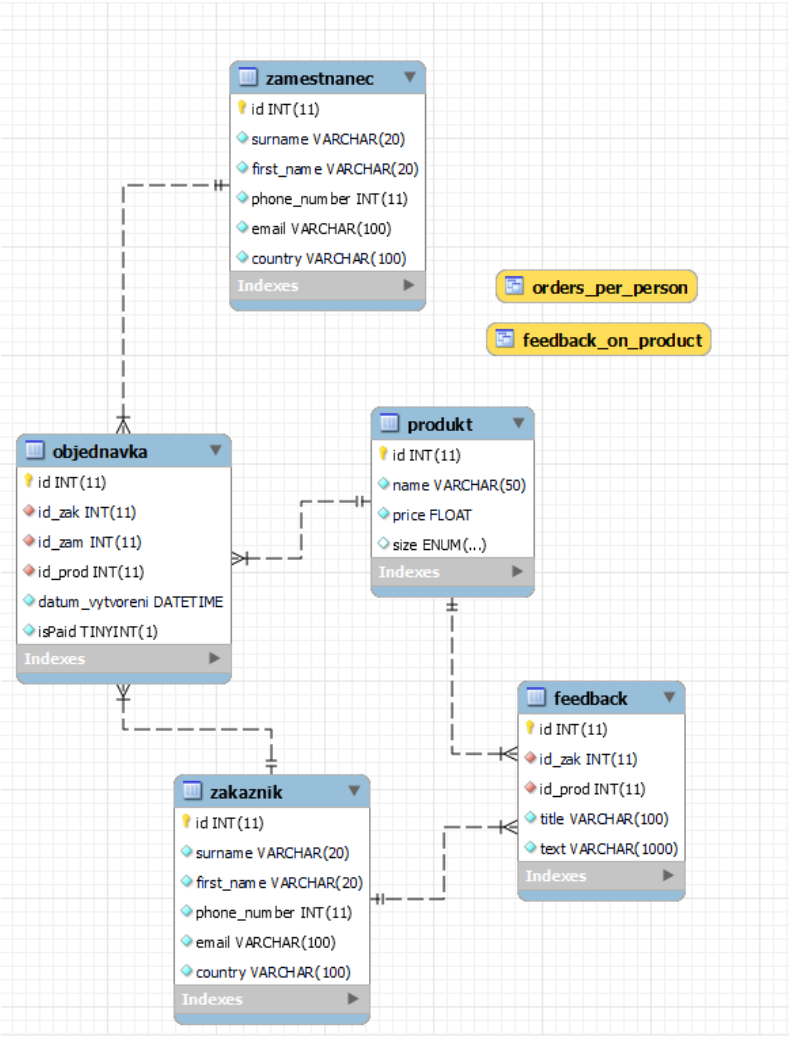
UML Class Diagram





Database

E-R Diagram



The **produkt.size** attribute is an ENUM consisting of:

'XXS', 'XS', 'S', 'M', 'L', 'XL', 'XXL', '3XL'

Configuration

The **.env** file consists of all environment variables and can be configured. What you're most likely looking to change is the **database connection info**.

```
DB_HOST=<your MySQL server>
DB_USER=<your MySQL username>
DB_PASSWORD=<your MySQL password>
DB_NAME=<name of the database>
```

If you cloned this software from GitHub, you need to **create a .env file** inside the backend folder and paste inside the following:

```
PORT=4000

DB_HOST=<your MySQL server>
DB_USER=<your MySQL username>
DB_PASSWORD=<your MySQL password>
DB_NAME=<name of the database>
```

Dependency packages

The application uses the Express.js framework and MySQL2 package for connecting and managing a MySQL database.