

Задание

1. Установить python и flask
2. С их помощью создать сервис с REST API для своего варианта БД
3. Установить postman или написать простое тестовое приложение на python, выполняющее операции с данными с помощью разработанного сервиса и API
4. Продемонстрировать выполнение тестовых запросов CRUD (добавление/ выборка/ модификация/ удаление)

```

from flask import Flask, request, jsonify
import mysql.connector

app = Flask(__name__)

# Подключаемся к базе данных
mydb = mysql.connector.connect(
    host='localhost',
    user='root',
    password='zrc8=:&Eq,Hd',
    database='mydb1',
    port='3306'
)

# Создаем курсор для работы с базой данных
cursor = mydb.cursor()

# Маршрут для главной страницы
@app.route('/')
def index():
    return """
    <html>
    <head>
    </head>
    <body>
    <div style="text-align: center;">
        <h1>Добро пожаловать!</h1>
        <p>Лабораторная работа №12.</p>
    </div>
    </body>
    </html>
    """

@app.route('/players', methods=['GET'])
def get_all_players():
    cursor.execute("SELECT * FROM players")
    players = []
    for player in cursor.fetchall():
        players.append({
            'id': player[0],
            'nickname': player[1]
        })
    return jsonify({'players': players}), 200

@app.route('/tanks', methods=['GET'])
def get_all_tanks():
    cursor.execute("SELECT * FROM tanks")
    players = []
    for player in cursor.fetchall():

```

```

        players.append({
            'id': player[0],
            'name': player[1],
            'type': player[2]
        })
    return jsonify({'tanks': players}), 200

```

```

@app.route('/players_add', methods=['POST'])
def add_player():
    data = request.json
    nickname = data.get('nickname')
    # Выполняем SQL-запрос для добавления студента в базу данных
    cursor.execute("INSERT INTO players (nickname) VALUES (%s)", (nickname))
    mydb.commit()
    return jsonify({'message': 'Игрок успешно добавлен'}), 201

```

```

@app.route('/players_change', methods=['PUT'])
def change_nick():
    data = request.json
    nick_pred = data.get('nick_pred')
    nick_new = data.get('nick_new')
    # Выполняем SQL-запрос для добавления студента в базу данных
    cursor.execute(f"UPDATE players SET nickname = '{nick_new}' where nickname = '{nick_pred}' ")
    mydb.commit()
    return jsonify({'message': 'Игрок успешно обновлён'}), 201

```

```

@app.route('/players_delete', methods=['DELETE'])
def delete_student():
    data = request.json
    nick = data.get('nick')
    # Выполняем SQL-запрос для добавления студента в базу данных
    cursor.execute(f"DELETE FROM players WHERE nickname = '{nick}' ")
    mydb.commit()
    return jsonify({'message': 'Игрок успешно удалён'}), 201

```

```

if __name__ == '__main__':
    app.run(host='0.0.0.0')

```

После этого произошло подключение и создан сервис.

Добро пожаловать!

Лабораторная работа №12.

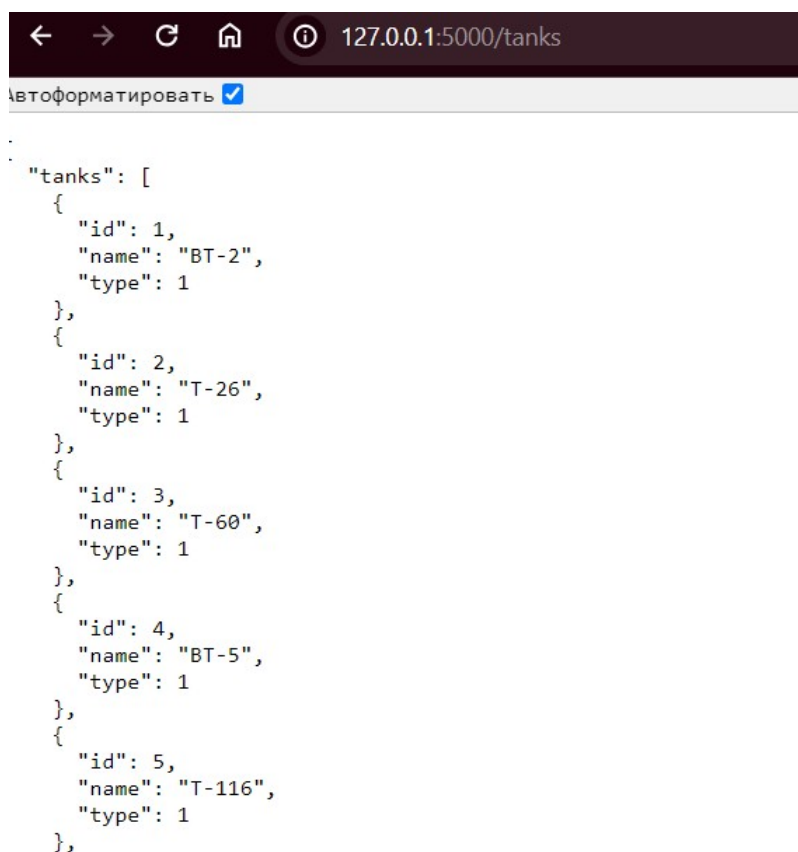
1 Запросы

1.1 Выборка всех игроков

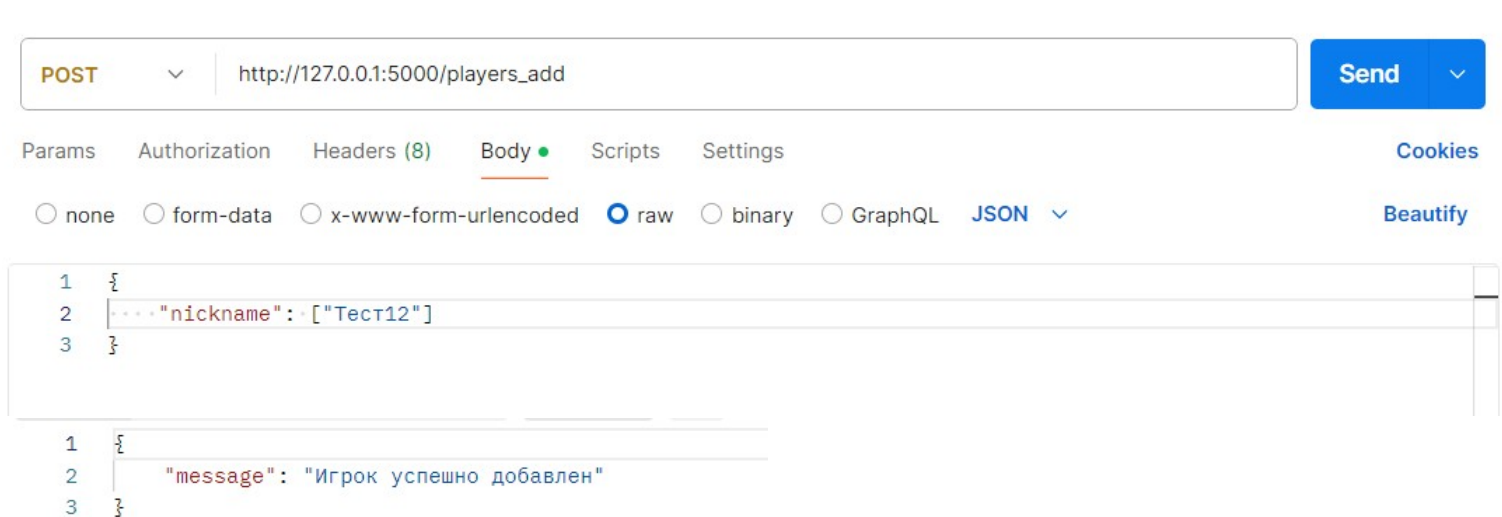


```
{
  "players": [
    {
      "id": 1,
      "nickname": "Yellow Flash"
    },
    {
      "id": 2,
      "nickname": "ahsaS"
    },
    {
      "id": 3,
      "nickname": "hunterKiller228"
    },
    {
      "id": 4,
      "nickname": "SaIyAjIn"
    },
    {
      "id": 5,
      "nickname": "dmitriyBrekotkin"
    },
    {
      "id": 6,
      "nickname": "your_master"
    },
    {
      "id": 29,
      "nickname": "dark_master"
    },
    {
      "id": 30,
      "nickname": "new_one"
    },
    {
      "id": 31,
      "nickname": "C"
    },
    {
      "id": 38,
      "nickname": "lab7"
    }
  ]
}
```

1.2 Выборка всех танков



1.3 Добавление нового игрока



```

    },
    "nickname": "new_one"
  },
  {
    "id": 31,
    "nickname": "C"
  },
  {
    "id": 38,
    "nickname": "lab7"
  },
  {
    "id": 44,
    "nickname": "Тест12"
  }
]
}

```

Игрок успешно добавлен

1.4 Изменение ника игрока

PUT

▼

http://127.0.0.1:5000/players_change

Send ▼

Params

Authorization

Headers (8)

Body ●

Scripts

Settings

Cookies

☐ none
 ☐ form-data
 ☐ x-www-form-urlencoded
 ☒ raw
 ☐ binary
 ☐ GraphQL

JSON ▼

Beautify

1 {

2 ... "nick_pred": "Тест12",

3 ... "nick_new": "new_name"

4 }

Pretty

Raw

Preview

Visualize

JSON ▼

≡

1 {

2 "message": "Игрок успешно обновлён"

3 }

```

{
  "id": 31,
  "nickname": "C"
},
{
  "id": 38,
  "nickname": "lab7"
},
{
  "id": 44,
  "nickname": "new_name"
}
]
}

```

Смена ника прошла успешно

1.5 Удаление игрока

DELETE

http://127.0.0.1:5000/players_delete

Send

Params

Authorization

Headers (8)

Body

Scripts

Settings

Cookies

☐ none

☐ form-data

☐ x-www-form-urlencoded

☒ raw

☐ binary

☐ GraphQL

JSON

Beautify

```
1 {
2   ... "nick": "new_name"
3 }
```

Body

Cookies

Headers (5)

Test Results

Pretty

Raw

Preview

Visualize

JSON

```
1 {
2   "message": "Игрок успешно удалён"
3 }
```

```
},
{
  "id": 30,
  "nickname": "new_one"
},
{
  "id": 31,
  "nickname": "C"
},
{
  "id": 38,
  "nickname": "lab7"
}
]
```

Игрок был удалён