

Министерство образования Республики Беларусь

Учреждение образования
Белорусский государственный университет информатики и
радиоэлектроники

Факультет компьютерного проектирования
Кафедра инженерной психологии и эргономики
Дисциплина: Базы данных

Лабораторная работа № 5
«Реализация библиотеки функций для взаимодействия с базой данных на
языке высокого уровня»

Выполнил:
ст.гр. 113802
Разумов Д.А.

Проверила:
Василькова А.Н.

Минск 2022

Цель работы - изучение реализации библиотеки функций для взаимодействия с БД с использованием языка программирования JavaScript (Node.js), средствами разработки Visual Studio, Google Chrome и с использованием сервера баз данных MySQL Server.

Для начала, создадим новый main.js проект в среде разработки Visual Studio. Далее установим два пакета «mysql2» - для работы с базой данных и «express» - для работы с локальным сервером. И автоматически получили папку «node_modules» и два файла «package-log.json», «package.json» Пример подключения показан на рисунке 1.

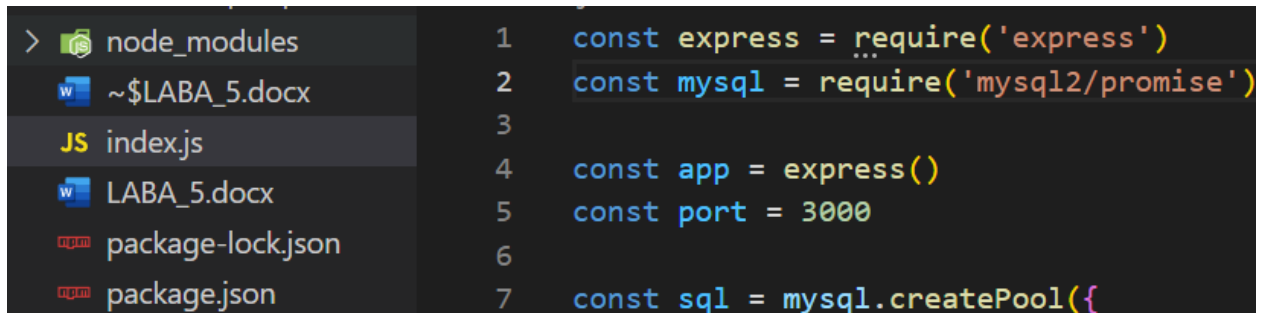


Рисунок 1 – Установка пакетов

Далее подключаемся к базе данных MySQL, для этого создадим класс sql с методом createPool, где подключимся к базе указав соответствующие данные.

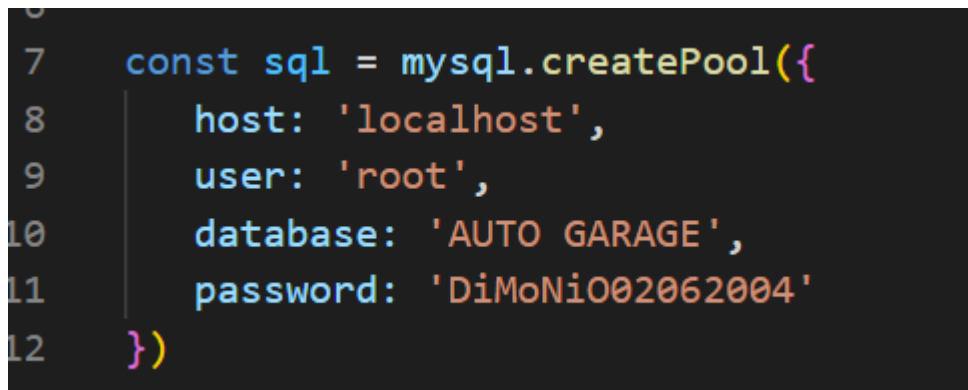


Рисунок 2 – Подключение к базе MySQL

После этого проверим успешность соединения, с помощью метода `listen`. При успешном/неуспешном подключении, получим соответствующее сообщение в терминале. Перед этим запустив локальный сервер на порту 3000 командой «`node main.js`».

```
189 app.listen(port, (err) => {
190   if (err) {
191     return console.log('something bad happened', err)
192   }
193   console.log(`server is listening on ${port}`)
194 })
195
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL

```
PS E:\Study BSUIR\STUDY_3_SEMESTER\Базы данных\LABA_5> node index.js
server is listening on 3000
```

Рисунок 3 – Проверка соединения с базой MySQL

Создадим стартовую страницу с выбором таблиц при помощи метода `get` и асинхронной функции. Пример кода и результирующая таблица представлена на рисунках 4-5.

```
app.get('/', async function(req, res) {
  res.send('<!DOCTYPE html>
  <html>
  <body>
    <div style="max-width: 1600px; margin: 100px auto 0 auto; border: 1.5px solid black; background-color: #ede4e4; padding: 100px 0;">
      <h1 style="font-size: 32px; text-align:center; margin-bottom: 100px;">Авторапах</h1>
      <div style="display: flex; flex-wrap: wrap; justify-content: space-around;">
        <a href="/cars">
          <button style="width: 200px; height: 100px; border-radius: 10px; cursor: pointer; font-size: 16px; font-weight: bold; text-decoration: none; color: black;">Cars</button>
        </a>
        <a href="/drivers">
          <button style="width: 200px; height: 100px; border-radius: 10px; cursor: pointer; font-size: 16px; font-weight: bold; text-decoration: none; color: black;">Drivers</button>
        </a>
        <a href="/traffics">
          <button style="width: 200px; height: 100px; border-radius: 10px; cursor: pointer; font-size: 16px; font-weight: bold; text-decoration: none; color: black;">Traffics</button>
        </a>
      </div>
    </div>
  </body>
</html>`);
});
```

Рисунок 4 – Пример кода создания начальной страницы

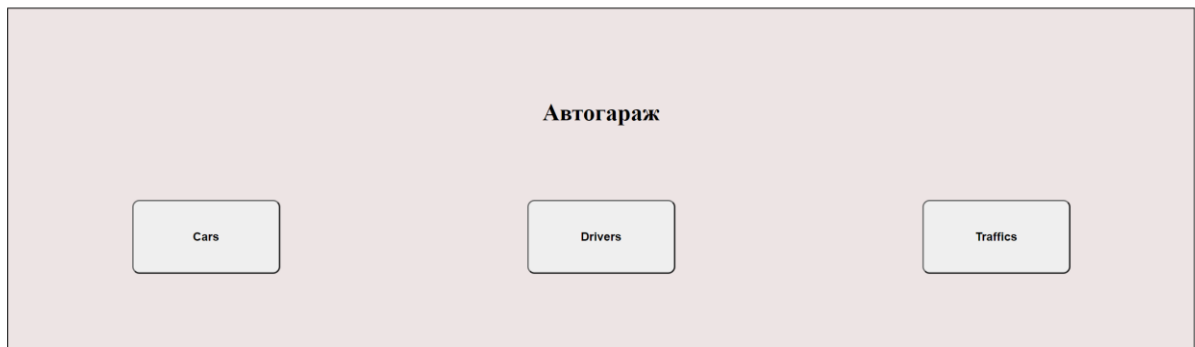


Рисунок 5 – Вывод начальной таблицы на сервер

Кнопкам соответствует определенный адрес сервера, который указан для каждой таблицы уникальным. Затем для каждой таблицы при помощи асинхронной функции, методов `get`, `send`, `query` создадим вывод в браузер.

Код для таблицы «Cars» и конечный результат представлены на рисунках 6-7.

```
app.get('/cars', async function(req, res) {
  const data = await sql.query('SELECT * FROM cars')
  const cars = data[0];

  res.send('<!DOCTYPE html>
  <html>
  <body>
    <div style="max-width: 1500px; margin: 100px auto 0 auto; border: 1.5px solid black; background-color: #ede4e4; padding: 50px;">
      <a href="/">
        <button style="width: 200px; height: 50px; border-radius: 10px; cursor: pointer; font-size: 12px; text-decoration: none; color: black;">Перейти на главную страницу</button>
      </a>
      <h1 style="font-size: 32px; text-align: center; margin-bottom: 30px;">Таблица 'Cars'</h1>

      <table style="width: 100%">
        <tbody>
          <th style="border: 1px solid black; background-color: grey">ID</th>
          <th style="border: 1px solid black; background-color: grey">Марка</th>
          <th style="border: 1px solid black; background-color: grey">Модель</th>
          <th style="border: 1px solid black; background-color: grey">Год выпуска</th>
          <th style="border: 1px solid black; background-color: grey">Пробег</th>
          <th style="border: 1px solid black; background-color: grey">Гос.номер</th>
          ${cars.map(car => `
            <tr>
              <td style="border: 1px solid black; background-color: grey">${car.idCars}</td>
              <td style="border: 1px solid black">${car.brand}</td>
              <td style="border: 1px solid black">${car.model}</td>
              <td style="border: 1px solid black">${car.yearOfIssue}</td>
              <td style="border: 1px solid black">${car.mileage}</td>
              <td style="border: 1px solid black">${car.stateNumber}</td>
            </tr>
          `).join('')}
        </tbody>
      </table>
    </div>
  </body>
  </html>`);
});
```

Рисунок 6 – Пример кода для заполнения таблицы «Cars»

Перейти на главную страницу

Таблица 'Cars'

ID	Марка	Модель	Год выпуска	Пробег	Гос.номер
1	HYUNDAI	XCIENT	2014	322067.00	2105-AM 5
2	KAMAZ	54901	2017	132503.00	2203-ЛС 7
3	MAZ	5440M9	2015	188945.00	1205-ЛЧ 7
4	VOLVO	FE 2	2019	95671.00	2218-KX 4
5	SCANIA	S730 V8	2020	66743.00	6262-AC 7

Рисунок 7 – Результат вывода таблицы «Cars»

Код для таблицы «Drivers» и конечный результат представлены на рисунках 8-9.

```

app.get('/drivers', async function(req, res) {
  const data = await sql.query(`SELECT * FROM drivers`)
  const drivers = data[0];

  res.send(`<!DOCTYPE html>
  <html>
  <body>
    <div style="max-width: 1500px; margin: 100px auto 0 auto; border: 1.5px solid black; background-color: #ede4e4; padding: 50px;">
      <a href="/">
        <button style="width: 200px; height: 50px; border-radius: 10px; cursor: pointer; font-size: 12px; text-decoration: none; color: black;">Перейти на главную страницу</button>
      </a>
      <h1 style="font-size: 32px; text-align:center; margin-bottom: 30px;">Таблица 'Drivers'</h1>

      <table style="width: 100%">
        <tbody>
          <th style="border: 1px solid black; background-color: grey">ID</th>
          <th style="border: 1px solid black; background-color: grey">ФИО</th>
          <th style="border: 1px solid black; background-color: grey">Категория ргуаа</th>
          <th style="border: 1px solid black; background-color: grey">Дом. адрес</th>
          <th style="border: 1px solid black; background-color: grey">Рейтинг</th>
          <th style="border: 1px solid black; background-color: grey">Семеное положение</th>
          <th style="border: 1px solid black; background-color: grey">Телефон</th>
          <th style="border: 1px solid black; background-color: grey">Код номера</th>
          <th style="border: 1px solid black; background-color: grey">Дата рождения</th>
          <th style="border: 1px solid black; background-color: grey">Стаж работы</th>
          <th style="border: 1px solid black; background-color: grey">Гражданство</th>
          ${drivers.map(driver => `
            <tr>
              <td style="border: 1px solid black; background-color: grey">${driver.idDrivers}</td>
              <td style="border: 1px solid black">${driver.fullName}</td>
              <td style="border: 1px solid black">${driver.category}</td>
              <td style="border: 1px solid black">${driver.address}</td>
              <td style="border: 1px solid black">${driver.reputation}</td>
              <td style="border: 1px solid black">${driver.familyStatus}</td>
              <td style="border: 1px solid black">${driver.phone}</td>
              <td style="border: 1px solid black">${driver.callCode}</td>
              <td style="border: 1px solid black">${driver.dateOfBirth}</td>
              <td style="border: 1px solid black">${driver.experience}</td>
              <td style="border: 1px solid black">${driver.citizenship}</td>
            </tr>
          `).join('')}}
        </tbody>
      </table>
    </div>
  </body>
</html>`);

```

Рисунок 8 – Пример кода для заполнения таблицы «Drivers»

Перейти на главную страницу

Таблица 'Drivers'

:

ID	ФИО	Категория груза	Дом. адрес	Рейтинг	Семеновое положение	Телефон	Код номера	Дата рождения	Стаж работы	Гражданство
1	Роголевич Виктор Семенович	Крупногабаритный груз	г.Минск ул.Волоха 3-17	null	null	447104585	+375	Mon Dec 01 1975 00:00:00 GMT+0300 (Москва, стандартное время)	24	null
2	Ефремов Геннадий Викторович	Крупногабаритный груз	г.Минск ул.Маяковского 23-102	null	null	445629451	+375	Wed May 22 1991 00:00:00 GMT+0300 (Москва, летнее время)	8	null
3	Чернов Юрий Павлович	Древесина	г.Минск ул.Дзержинского 88-137	Профи	null	331144185	+375	Fri Mar 12 1971 00:00:00 GMT+0300 (Москва, стандартное время)	27	null
4	Керножицкий Степан Дмитриевич	Техника	г.Минск ул.Слуцкое шоссе 61-67	Профи	null	294104582	+375	Mon Jan 22 1968 00:00:00 GMT+0300 (Москва, стандартное время)	26	null
5	Соколовский Валерий Александрович	Стекло	г.Минск ул.Карпова 39-14	null	null	295012127	+375	Thu Oct 13 1988 00:00:00 GMT+0300 (Москва, стандартное время)	9	null

Рисунок 9 – Результат вывода таблицы «Drivers»

Код для таблицы «Traffics» и конечный результат представлены на рисунках 10-11.

```

app.get('/traffics', async function(req, res) {
  const data = await sql.query(`SELECT * FROM traffics`)
  const traffics = data[0];

  res.send(`<!DOCTYPE html>
<html>
  <body>
    <div style="max-width: 1500px; margin: 100px auto 0 auto; border: 1.5px solid black; background-color: #ede4e4; padding: 50px;">
      <a href="/">
        <button style="width: 200px; height: 50px; border-radius: 10px; cursor: pointer; font-size: 12px; text-decoration: none; color: black;">Перейти на главную страницу</button>
      </a>
      <h1 style="font-size: 32px; text-align:center; margin-bottom: 30px;">Таблица 'Traffics'</h1>
      <table style="width: 100%">
        <tbody>
          <tr>
            <th style="border: 1px solid black; background-color: grey">ID</th>
            <th style="border: 1px solid black; background-color: grey">Пункт отправления</th>
            <th style="border: 1px solid black; background-color: grey">Конечный пункт</th>
            <th style="border: 1px solid black; background-color: grey">Длина пути</th>
            <th style="border: 1px solid black; background-color: grey">Количество остановок</th>
            <th style="border: 1px solid black; background-color: grey">ID Машин</th>
            <th style="border: 1px solid black; background-color: grey">ID Водителя</th>
          <tr>
            <td colspan="7">
              ${traffics.map(traffic => `
                <tr>
                  <td style="border: 1px solid black; background-color: grey">${traffic.idTraffics}</td>
                  <td style="border: 1px solid black">${traffic.startPoint}</td>
                  <td style="border: 1px solid black">${traffic.endPoint}</td>
                  <td style="border: 1px solid black">${traffic.trafficlelength}</td>
                  <td style="border: 1px solid black">${traffic.countStops}</td>
                  <td style="border: 1px solid black">${traffic.idCars}</td>
                  <td style="border: 1px solid black">${traffic.idDrivers}</td>
                </tr>
              `).join('')};
            </td>
          </tr>
        </tbody>
      </table>
    </div>
  </body>
</html>`);
});

```

Рисунок 10 - Пример кода для заполнения таблицы «Traffics»

Перейти на главную страницу

Таблица 'Traffics'

ID	Пункт отправления	Конечный пункт	Длина пути	Количество остановок	ID Машины	ID Водителя
2	Брест	Москва	1062	5	2	3
3	Могилев	Берлин	1313	7	3	4
4	Минск	Прага	1297	7	4	5

Рисунок 11 - Результат вывода таблицы «Traffics»

Для каждой страницы имеется кнопка выхода в главное меню. А также для каждой таблицы формируется свой уникальный адрес. На рисунке 12 показан один из примеров адреса.

localhost:3000/traffics

Рисунок 12 – Адрес одной из страниц