

Отчет

Настройка базы данных SQLite

```
JS server.js > ...
1  const express = require("express");
2  const bodyParser = require("body-parser");
3  const session = require("express-session");
4  const { Sequelize, DataTypes } = require("sequelize");
5  const bcrypt = require("bcrypt");
6  const path = require("path");
7
8  const app = express();
9  const port = 3000;
10
11
12  const sequelize = new Sequelize({
13    dialect: "sqlite",
14    storage: "db.sqlite",
15  });
16
```

Создаем таблицы роли и пользователи

```
18  const Role = sequelize.define(
19    "Role",
20    {
21      id: {
22        type: DataTypes.INTEGER,
23        autoIncrement: true,
24        primaryKey: true,
25      },
26      name: {
27        type: DataTypes.STRING,
28        allowNull: false,
29        unique: true,
30      },
31    },
32    { timestamps: false }
33  );
34
36  const User = sequelize.define(
37    "User",
38    {
39      id: {
40        type: DataTypes.INTEGER,
41        autoIncrement: true,
42        primaryKey: true,
43      },
44      login: {
45        type: DataTypes.STRING,
46        allowNull: false,
47        unique: true,
48      },
49      password: {
50        type: DataTypes.STRING,
51        allowNull: false,
52      },
53      roleId: {
54        type: DataTypes.INTEGER,
55        allowNull: false,
56        references: {
57          model: Role,
58          key: "id",
59        },
60      },
61    },
62    { timestamps: false }
63  );
```

Устанавливаем связь один ко многим и создаем администратора

```
66  Role.hasMany(User, { foreignKey: "roleId" });
67  User.belongsTo(Role, { foreignKey: "roleId" });
68
89  if (usersCount === 0) {
90    const hashedPassword = await bcrypt.hash("admin", 10); // Пароль admin
91    await User.create({
92      login: "admin",
93      password: hashedPassword,
94      roleId: adminRole[0].id,
95    });
96    console.log("Администратор создан");
97  }
98  })
99  .catch((err) => console.error(err));
```

Проверка авторизации и роли

```
116 function isAuthenticated(req, res, next) {
117   if (req.session.user) {
118     next();
119   } else {
120     res.redirect("/login");
121   }
122 }
123
124 function hasRole(roleName) {
125   return async (req, res, next) => {
126     if (req.session.user) {
127       const user = await User.findByPk(req.session.user.id, { include: Role });
128       if (user && user.Role.name === roleName) {
129         next();
130       } else {
131         res.status(403).send("Доступ запрещен");
132       }
133     } else {
134       res.redirect("/login");
135     }
136   };
137 }
```

Создаем страницы регистрации и авторизации и подключаем к серверу

```
1  <!doctype html>
2  <html>
3    <head>
4      <title>Авторизация</title>
5    </head>
6    <body>
7      <h1>Авторизация</h1>
8      <form method="post" action="/login">
9        <label for="login">Логин:</label>
10       <input type="text" id="login" name="login" required /><br /><br />
11       <label for="password">Пароль:</label>
12       <input
13         type="password"
14         id="password"
15         name="password"
16         required
17       /><br /><br />
18       <input type="submit" value="Войти" />
19     </form>
20     <p><a href="/register">Регистрация</a></p>
21   </body>
22 </html>
23
```

```

2  <html>
3  <head>
4  |   <title>Регистрация</title>
5  </head>
6  <body>
7  |   <h1>Регистрация</h1>
8  |   <form method="post" action="/register">
9  |       <label for="login">Логин:</label>
10 |       <input type="text" id="login" name="login" required /><br /><br />
11 |       <label for="password">Пароль:</label>
12 |       <input
13 |           type="password"
14 |           id="password"
15 |           name="password"
16 |           required
17 |       /><br /><br />
18 |       <input type="submit" value="Зарегистрироваться" />
19 |   </form>
20 |   <p><a href="/login">Авторизация</a></p>
21 </body>
22 </html>
23

```

Запускаем сервер на порту

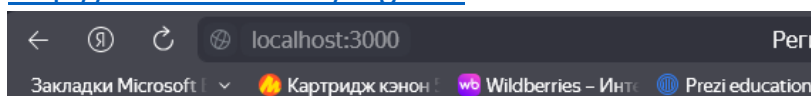
```

219  app.listen(port, () => {
220  |    console.log(`Сервер запущен на порту http://localhost:${port}`);
221  |  });
222

```

Проверка:

1. Переходим на страницу регистрации в браузере по адресу <http://localhost:3000/register>



Регистрация

Логин:

Пароль:

[Авторизация](#)

2. Создаем пользователя

Регистрация

Логин:

Пароль: 

[Авторизация](#)

3. Проверяем пользователя в бд

	id	login	password	roleId
	Фи...	Фильтр	Фильтр	Фильтр
1	1	admin	\$2b\$10\$xpFjtbzyCMJx8NW87dZH/...	2
2	2	dede	\$2b\$10\$pTr5JXlw74azh12XCU8/8O77kWzT...	1
3	3	disaeva	\$2b\$10\$5nDD5OohOXGdaHXkp1AUae1/...	1
4	4	isaeva	\$2b\$10\$JK3kjKhIOloejSFAEbHul.rcMeVQ...	1

4. Пробуем авторизоваться

Авторизация

Логин:

Пароль: 

[Регистрация](#)

Добро пожаловать, ddd! [Выйти](#)

5. Входим на страницу админа

Авторизация

Логин:

Пароль: 

[Регистрация](#)

Добро пожаловать, admin! [Выйти](#)