
**NATIONAL TECHNICAL UNIVERSITY OF UKRAINE "IGOR SIKORSKY KYIV POLYTECHNIC INSTITUTE"**

**Faculty of Informatics and Computer Engineering Department of Computer Engineering**

# Laboratory work 5

# Creating APIs with Node.js and Express. Geolocation API. Using Leaflet library

**2nd year student**
**groups of IP-98**
**CEM KOYLUOGLU**

**Kyiv 2021**

## ===Task 5==

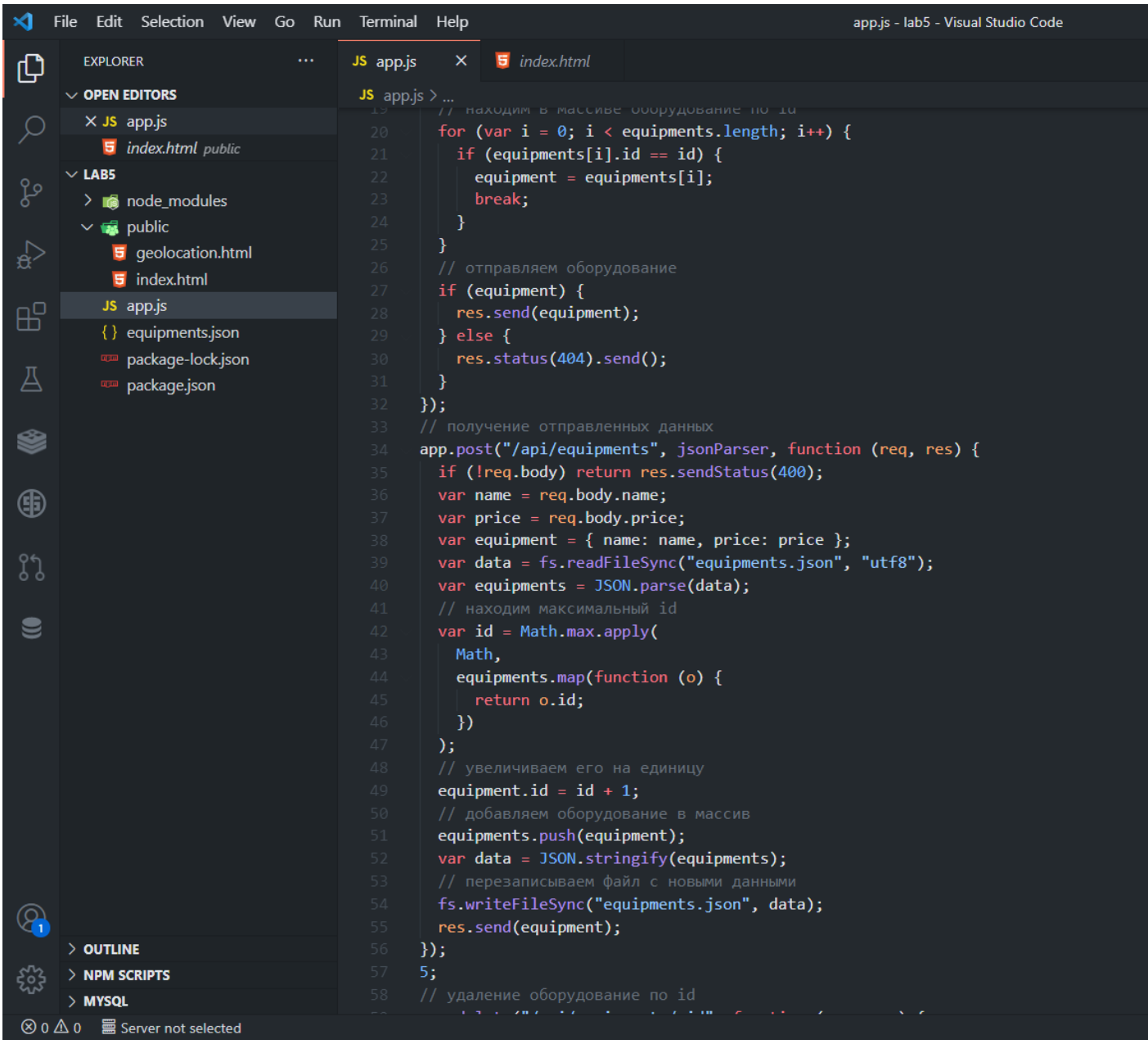
```
var express = require("express");
var bodyParser = require("body-parser");
var fs = require("fs");
var app = express();
var jsonParser = bodyParser.json();
app.use(express.static(__dirname + "/public"));
// получение списка данных
app.get("/api/equipments", function (req, res) {
var content = fs.readFileSync("equipments.json", "utf8");
var users = JSON.parse(content);
res.send(users);
});
// получение оборудования по id
app.get("/api/equipments/:id", function (req, res) {
var id = req.params.id; // получаем id
var content = fs.readFileSync("equipments.json", "utf8");
var equipments = JSON.parse(content);
var equipment = null;
// находим в массиве оборудование по id
for (var i = 0; i < equipments.length; i++) {
if (equipments[i].id == id) {
equipment = equipments[i];
break;
}
}
// отправляем оборудование
if (equipment) {
res.send(equipment);
} else {
```

```
res.status(404).send();
}
});
// получение отправленных данных
app.post("/api/equipments", jsonParser, function (req, res) {
if (!req.body) return res.sendStatus(400);
var name = req.body.name;
var price = req.body.price;
var equipment = { name: name, price: price };
var data = fs.readFileSync("equipments.json", "utf8");
var equipments = JSON.parse(data);
// находим максимальный id
var id = Math.max.apply(
Math,
equipments.map(function (o) {
return o.id;
})
);
// увеличиваем его на единицу
equipment.id = id + 1;
// добавляем оборудование в массив
equipments.push(equipment);
var data = JSON.stringify(equipments);
// перезаписываем файл с новыми данными
fs.writeFileSync("equipments.json", data);
res.send(equipment);
});
5;
// удаление оборудование по id
app.delete("/api/equipments/:id", function (req, res) {
```

```
var id = req.params.id;
var data = fs.readFileSync("equipments.json", "utf8");
var equipments = JSON.parse(data);
var index = -1;
// находим индекс оборудования в массиве
for (var i = 0; i < equipments.length; i++) {
if (equipments[i].id == id) {
index = i;
break;
}
}
if (index > -1) {
// удаляем оборудование из массива по индексу
var equipment = equipments.splice(index, 1)[0];
var data = JSON.stringify(equipments);
fs.writeFileSync("equipments.json", data);
// отправляем удаленного оборудование
res.send(equipment);
} else {
res.status(404).send();
}
});
// изменение оборудования
app.put("/api/equipments", jsonParser, function (req, res) {
if (!req.body) return res.sendStatus(400);
var id = req.body.id;
var name = req.body.name;
var price = req.body.price;
var data = fs.readFileSync("equipments.json", "utf8");
var equipments = JSON.parse(data);
```

```
var equipment = null;
for (var i = 0; i < equipments.length; i++) {
if (equipments[i].id == id) {
equipment = equipments[i];
break;
}
}
// изменяем данные у оборудования
if (equipment) {
equipment.name = name;
equipment.price = price;

var data = JSON.stringify(equipments);
fs.writeFileSync("equipments.json", data);
res.send(equipment);
} else {
res.status(404).send(equipment);
}
});
app.listen(8080, function () {
console.log("Сервер ожидает подключения...");
});
```


File Edit Selection View Go Run Terminal Help
app.js - lab5 - Visual Studio Code
EXPLORER
OPEN EDITORS
app.js
index.html public
LAB5
node_modules
public
geolocation.html
index.html
app.js
equipments.json
package-lock.json
package.json
OUTLINE
NPM SCRIPTS
MYSQL
app.js
index.html
app.js > ...
// находим в массиве оборудование по id
for (var i = 0; i < equipments.length; i++) {
if (equipments[i].id == id) {
equipment = equipments[i];
break;
}
}
// отправляем оборудование
if (equipment) {
res.send(equipment);
} else {
res.status(404).send();
}
});
// получение отправленных данных
app.post("/api/equipments", jsonParser, function (req, res) {
if (!req.body) return res.sendStatus(400);
var name = req.body.name;
var price = req.body.price;
var equipment = { name: name, price: price };
var data = fs.readFileSync("equipments.json", "utf8");
var equipments = JSON.parse(data);
// находим максимальный id
var id = Math.max.apply(
Math,
equipments.map(function (o) {
return o.id;
})
);
// увеличиваем его на единицу
equipment.id = id + 1;
// добавляем оборудование в массив
equipments.push(equipment);
var data = JSON.stringify(equipments);
// перезаписываем файл с новыми данными
fs.writeFileSync("equipments.json", data);
res.send(equipment);
});
5;
// удаление оборудование по id
0 0
Server not selected

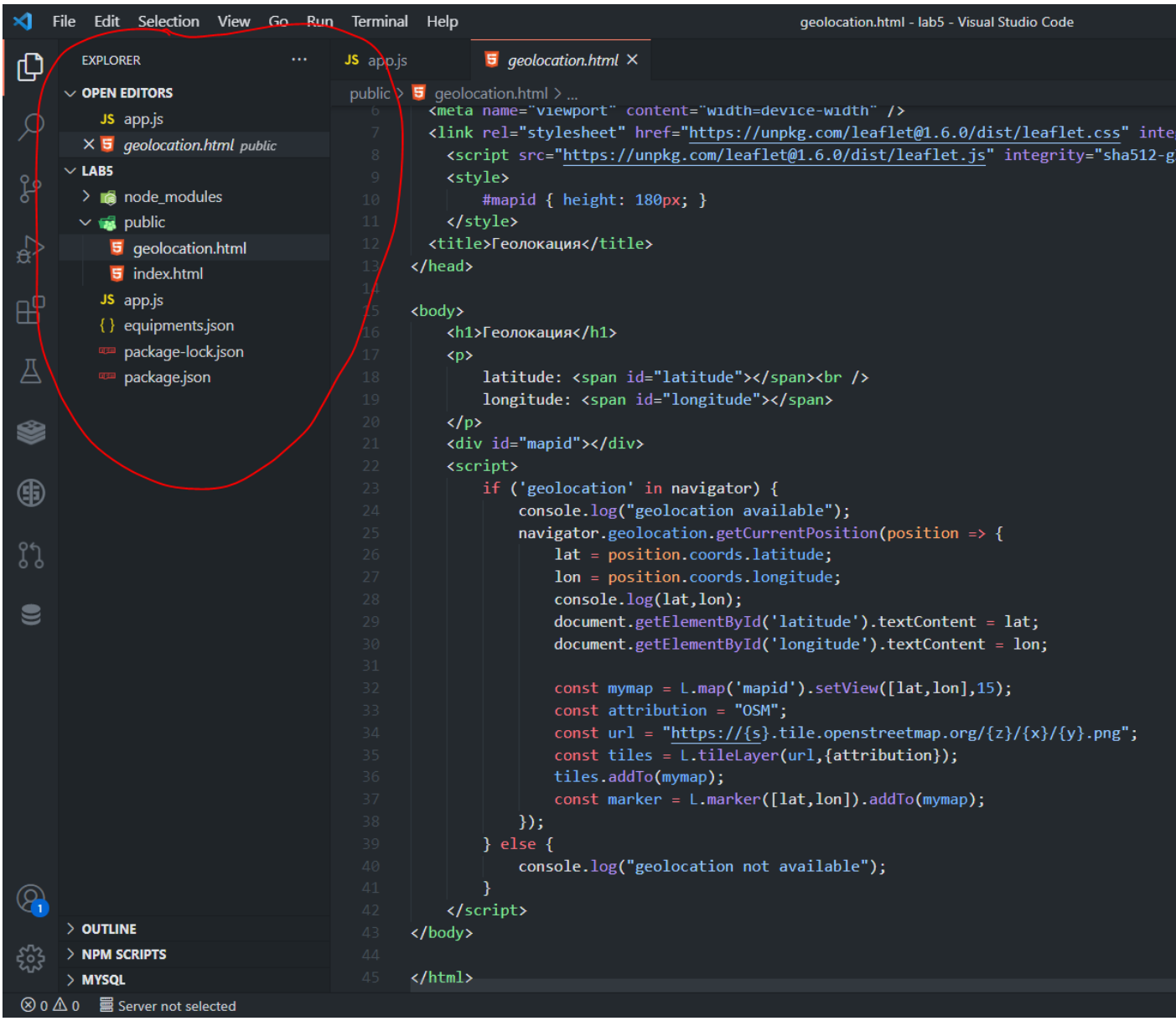

# Список оборудования

**Название:**

**Цена:**

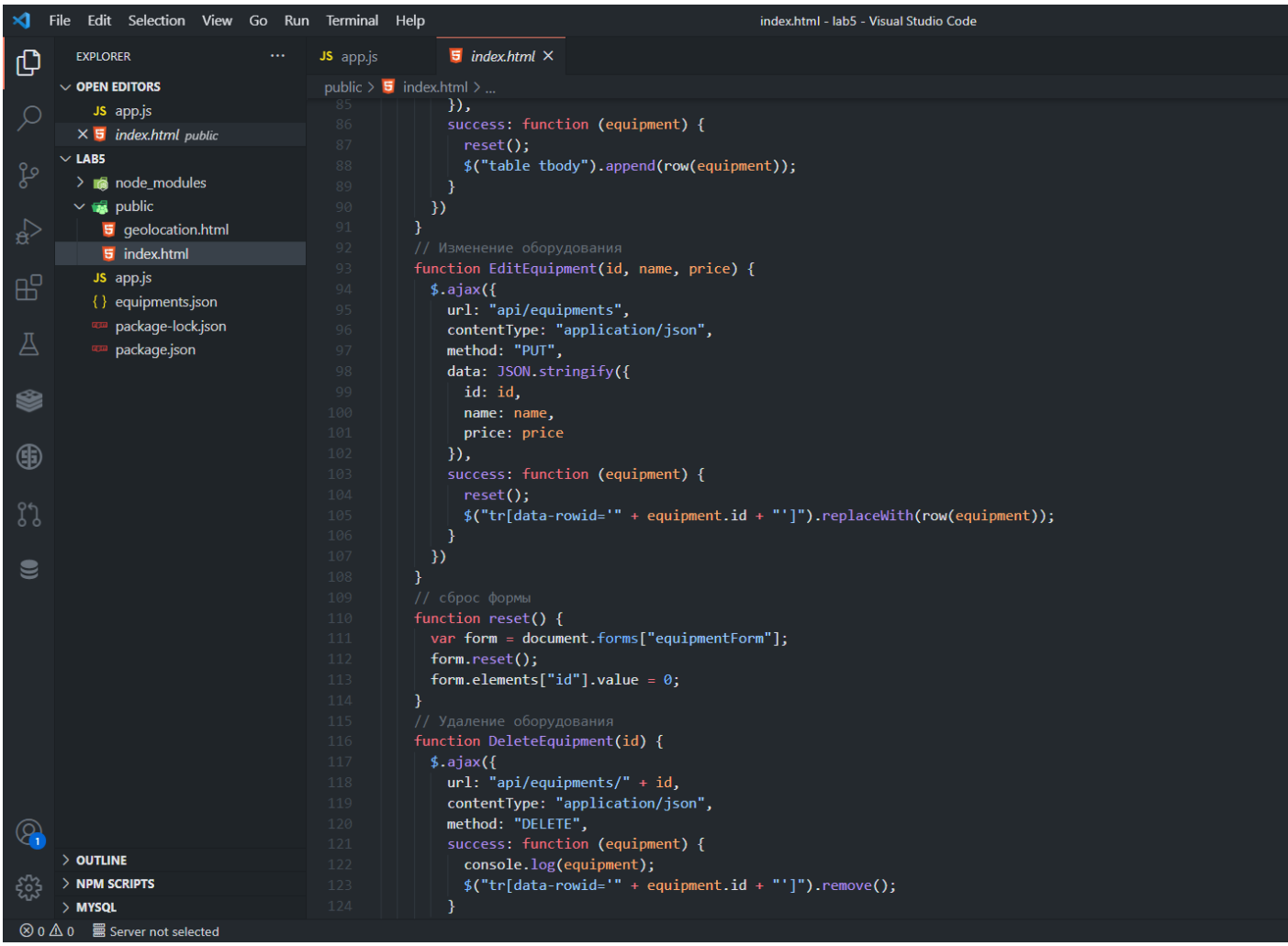
Сохранить Сбросить

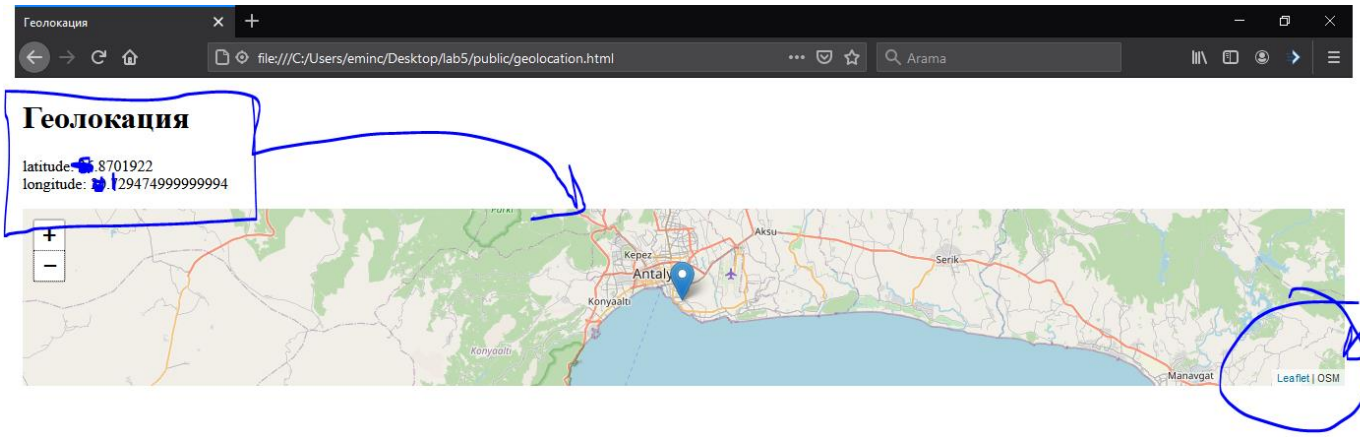
| Id | Название | Цена | |
|---|---|---|---|
| 1 | Монитор | 1000 | Изменить \| Удалить |
| 2 | Клавиатура | 2000 | Изменить \| Удалить |
| 3 | Мышь | 3000 | Изменить \| Удалить |
| 4 | Принтер | 4000 | Изменить \| Удалить |
| 5 | Наушники | 5000 | Изменить \| Удалить |

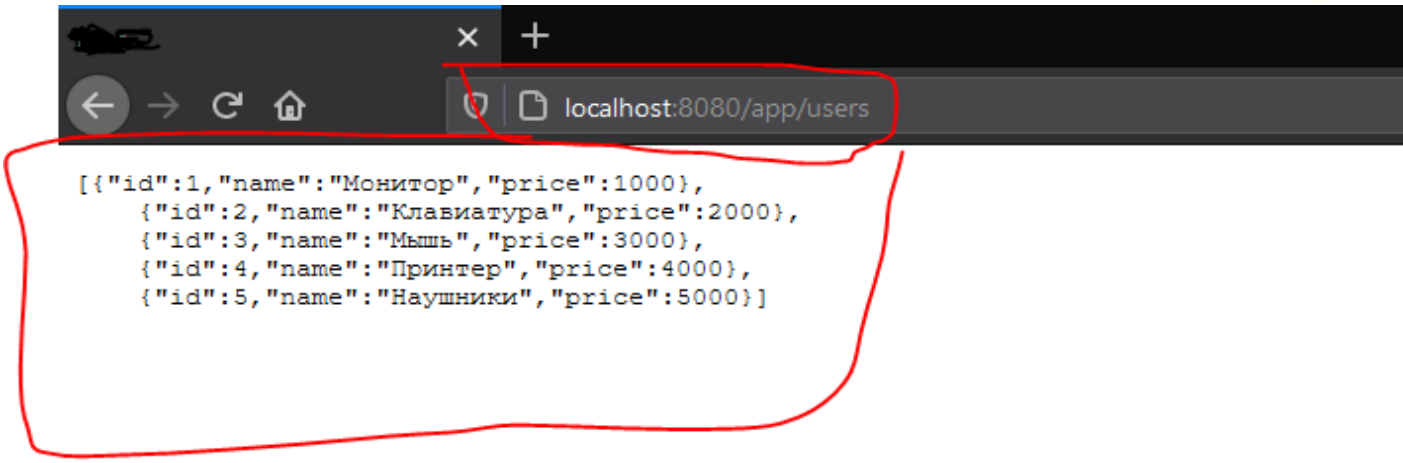

File Edit Selection View Go Run Terminal Help
geolocation.html - lab5 - Visual Studio Code
EXPLORER
OPEN EDITORS
app.js
geolocation.html public
LAB5
node_modules
public
geolocation.html
index.html
app.js
equipments.json
package-lock.json
package.json
OUTLINE
NPM SCRIPTS
MYSQL
app.js
geolocation.html
public > geolocation.html > ...
<meta name="viewport" content="width=device-width" />
<link rel="stylesheet" href="https://unpkg.com/leaflet@1.6.0/dist/leaflet.css" inte
<script src="https://unpkg.com/leaflet@1.6.0/dist/leaflet.js" integrity="sha512-g
<style>
#mapid { height: 180px; }
</style>
<title>Геолокация</title>
</head>
<body>
<h1>Геолокация</h1>

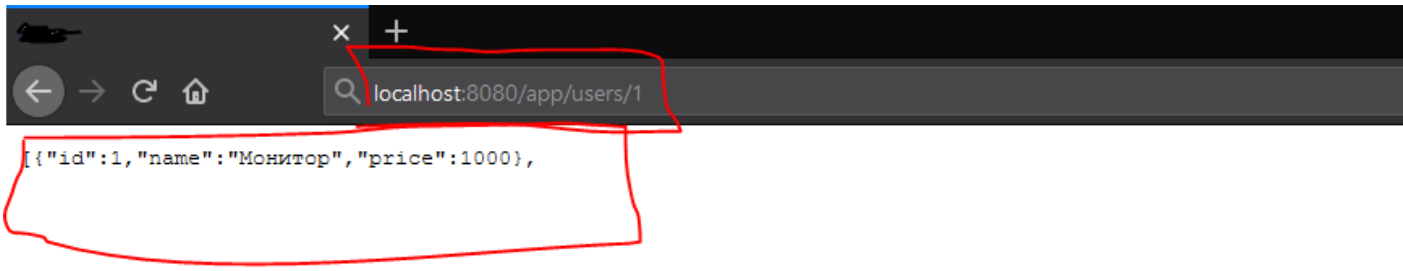
latitude: <br />
longitude: 

<script>
if ('geolocation' in navigator) {
console.log("geolocation available");
navigator.geolocation.getCurrentPosition(position => {
lat = position.coords.latitude;
lon = position.coords.longitude;
console.log(lat,lon);
document.getElementById('latitude').textContent = lat;
document.getElementById('longitude').textContent = lon;
const mymap = L.map('mapid').setView([lat,lon],15);
const attribution = "OSM";
const url = "https://{s}.tile.openstreetmap.org/{z}/{x}/{y}.png";
const tiles = L.tileLayer(url,{attribution});
tiles.addTo(mymap);
const marker = L.marker([lat,lon]).addTo(mymap);
});
} else {
console.log("geolocation not available");
}
</script>
</body>
</html>
0 0
Server not selected


File Edit Selection View Go Run Terminal Help
index.html - lab5 - Visual Studio Code
EXPLORER
OPEN EDITORS
app.js
index.html public
LAB5
node_modules
public
geolocation.html
index.html
app.js
equipments.json
package-lock.json
package.json
OUTLINE
NPM SCRIPTS
MYSQL
app.js
index.html
public > index.html > ...
}),
success: function (equipment) {
reset();
$("table tbody").append(row(equipment));
}
})
}
// Изменение оборудования
function EditEquipment(id, name, price) {
$.ajax({
url: "api/equipments",
contentType: "application/json",
method: "PUT",
data: JSON.stringify({
id: id,
name: name,
price: price
}),
success: function (equipment) {
reset();
$("tr[data-rowid='" + equipment.id + "']").replaceWith(row(equipment));
}
})
}
// сброс формы
function reset() {
var form = document.forms["equipmentForm"];
form.reset();
form.elements["id"].value = 0;
}
// Удаление оборудования
function DeleteEquipment(id) {
$.ajax({
url: "api/equipments/" + id,
contentType: "application/json",
method: "DELETE",
success: function (equipment) {
console.log(equipment);
$("tr[data-rowid='" + equipment.id + "']").remove();
}
0 0
Server not selected


Геолокация
file:///C:/Users/eminc/Desktop/lab5/public/geolocation.html
Arama
Геолокация
latitude: 36.8701922
longitude: 30.729474999999994
Kepez
Antalya
Aksu
Serik
Konyaalti
Manavgat
Leaflet | OSM


localhost:8080/app/users
[{"id":1,"name":"Монитор","price":1000},
{"id":2,"name":"Клавиатура","price":2000},
{"id":3,"name":"Мышь","price":3000},
{"id":4,"name":"Принтер","price":4000},
{"id":5,"name":"Наушники","price":5000}]


localhost:8080/app/users/1
[{"id":1,"name":"Монитор","price":1000},

==ANSWERS==

1. What is API?
   **API** is the acronym for Application Programming Interface, which is a software intermediary that allows two applications to talk to each other. Each time you use an app like Facebook, send an instant message, or check the weather on your phone, you're using an **API**

2. What is the REST architecture?
   Representational state transfer (**REST**) is a de-facto standard for a software **architecture** of interactive applications that use Web services. ... Such a Web service must provide its Web resources in a textual representation and allow them to be read and modified with a stateless protocol and a predefined set of operations.

3. Why are the methods app.get () / app.post () / app.delete () / app.put () used?

actually which function will be calling it depends on what request verb you are using to the path.

if use method='GET' in HTML form or type='GET' in ajax or $http.get() in service then call .get()

if method='POST' in HTML form or type='POST' in ajax or $http.post() in service then call .post()

if method='PUT' in HTML form or type='PUT' in ajax or $http.put() in service then call .put()

if method='DELETE' in HTML form or type='DELETE' in ajax or $http.delete() in service then call .delete()

4. How to get data from a file using the fs.readFileSync () method?
   The **fs**. **readFileSync**() method is an inbuilt application programming interface of **fs** module which is used to read the file and return its content. In **fs**. readFile() method, we can read a file in a non-blocking asynchronous way, but in **fs**. ... js to block other parallel process and do the current file reading process.

5. What is the JSON.parse () method used for?

The **JSON**. **parse() method** parses a **JSON** string, constructing the JavaScript value or object described by the string. An optional reviver function can be provided to perform a transformation on the resulting object before it is returned

6. What opportunities does Geolocation API have?

   The **Geolocation API** allows the user to provide their location to web applications if they so desire. For privacy reasons, the user is asked for permission to report location information. WebExtensions that wish to use the **Geolocation** object must add the "**geolocation**" permission to their manifest.

7. What is Leaflet for JavaScript?

**Leaflet** is the leading open-source **JavaScript** library for mobile-friendly interactive maps. Weighing just about 39 KB of JS , it has all the mapping features most developers ever need. **Leaflet** is designed with simplicity, performance and usability in mind.

8. How to use Leaflet to add the user's current location on the map?

   With a couple of modifications, you can use `setInterval()` as a timer, and remove the existing user position on each pass.