

Отчет по разделу #7

Цель работы

Ознакомиться с взаимодействием серверов в Node.js.

Задание 1

Условие

Создать сервер А. На стороне сервера хранится файл с содержимым в формате JSON. При получении запроса на `/insert/record` идёт добавление записи в файл. При получении запроса на `/select/record` идёт получение записи из файла. Каждая запись хранит информацию о машине (название и стоимость).

Создать сервер Б. На стороне сервера хранится файл с содержимым в формате JSON. Каждая запись в файле хранит информацию о складе и массиве машин, находящихся на данном складе. То есть каждая запись хранит в себе название склада (строку) и массив названий машин (массив строк). При получении запроса на `/insert/record` идёт добавление записи в файл. При получении запроса на `/select/record` идёт получение записи из файла.

Создать сервер С. Сервер выдаёт пользователю страницы с формами для ввода информации. При этом сервер взаимодействует с серверами А и Б. Реализовать для пользователя функции:

- создание нового типа машины
- получение информации о стоимости машины по её типу
- создание нового склада с находящимися в нём машинами
- получение информации о машинах на складе по названию склада

Реализовать удобный для пользователя интерфейс взаимодействия с системой (использовать поля ввода и кнопки).

Код программы

Язык: **Javascript**

Сервер А

src/app.js

```
const express = require("express");
const body_parser = require("body-parser");

const router = require("./router");
const logger = require("./controllers").logger;

const port = process.env.PORT | 4001;

const app = express();

app.use(body_parser.json());
app.use(logger);

app.use(router);

app.listen(port);
console.log(`Running on port ${port}`);
```

src/controllers.js

```

"use strict";

const utils = require("../utils");

module.exports.insert_record = (req, res) => {
  try {
    const { name, price } = req.body;
    if (!name || !price)
      throw Error("");

    utils.put({ name, price });
    res.status(200).send(JSON.stringify({ result: "Ok" }));
  } catch(_) {
    res.status(400).send(JSON.stringify({ result: "No object in body" }));
  }
}

module.exports.select_record = (req, res) => {
  const car = utils.get(req.query.name);
  if (car)
    res.status(200).send(JSON.stringify({ result: "Ok", car }));
  else
    res.status(400).send(JSON.stringify({ result: `No car with name ${req.query.name}` }));
}

module.exports.bad_request_controller = (req, res) => {
  res.status(400).send(JSON.stringify({ result: `Unknown request: ${req.method} ${req.url}` }));
}

module.exports.logger = (req, _, next) => {
  console.log(`${req.url} ${req.method}`);

  if (req.method === "GET")
    console.log(req.query);
  else if (req.method === "POST")
    console.log(req.body);

  next();
}

```

src/router.js

```

const Router = require("express").Router;

const controllers = require("../controllers");

const router = Router();

router.post("/insert/record", controllers.insert_record);
router.get("/select/record", controllers.select_record);
router.use(controllers.bad_request_controller);

module.exports = router;

```

src/utils.js

```
"use strict";

const fs = require("fs");
const path = require("path");

const DB_FILE_FOLDER = path.join(__dirname, "..", "meta");
const DB_FILE = path.join(DB_FILE_FOLDER, "db.txt");

if (!fs.existsSync(DB_FILE_FOLDER))
  fs.mkdirSync(DB_FILE_FOLDER);

if (!fs.existsSync(DB_FILE))
  fs.writeFileSync(DB_FILE, JSON.stringify([]));

const db = JSON.parse(fs.readFileSync(DB_FILE));

console.log(db);

module.exports.get = (name) => {
  return db.find(car => car.name === name);
}

module.exports.put = (car) => {
  db.push(car);
  fs.writeFileSync(DB_FILE, JSON.stringify(db));
}
```

Сервер Б

src/app.js

```
const express = require("express");
const body_parser = require("body-parser");

const router = require("./router");
const logger = require("./controllers").logger;

const port = process.env.PORT | 4002;

const app = express();

app.use(body_parser.json());
app.use(logger);

app.use(router);

app.listen(port);
console.log(`Running on port ${port}`);
```

src/controllers.js

```

"use strict";

const utils = require("./utils");

module.exports.insert_record = (req, res) => {
  try {
    const { name, cars } = req.body;
    if (!name || !cars)
      throw Error("");
    utils.put({ name, cars });

    res.status(200).send(JSON.stringify({ result: "Ok" }));
  } catch(_) {
    res.status(400).send(JSON.stringify({ result: "No object in body" }));
  }
}

module.exports.select_record = (req, res) => {
  const stock = utils.get(req.query.name);
  if (stock)
    res.status(200).send(JSON.stringify({ result: "Ok", stock }));
  else
    res.status(400).send(JSON.stringify({ result: `No stock with name ${req.query.name}` }));
}

module.exports.bad_request_controller = (req, res) => {
  res.status(400).send(JSON.stringify({ result: `Unknown request: ${req.method} ${req.url}` }));
}

module.exports.logger = (req, _, next) => {
  console.log(`${req.url} ${req.method}`);

  if (req.method === "GET")
    console.log(req.query);
  else if (req.method === "POST")
    console.log(req.body);

  next();
}

```

src/router.js

```

const Router = require("express").Router;

const controllers = require("./controllers");

const router = Router();

router.post("/insert/record", controllers.insert_record);
router.get("/select/record", controllers.select_record);
router.use(controllers.bad_request_controller);

module.exports = router;

```

src/utils.js

```

"use strict";

const fs = require("fs");
const path = require("path");

const DB_FILE_FOLDER = path.join(__dirname, "..", "meta");
const DB_FILE = path.join(DB_FILE_FOLDER, "db.txt");

if (!fs.existsSync(DB_FILE_FOLDER))
    fs.mkdirSync(DB_FILE_FOLDER);

if (!fs.existsSync(DB_FILE))
    fs.writeFileSync(DB_FILE, JSON.stringify([]));

const db = JSON.parse(fs.readFileSync(DB_FILE));

module.exports.get = (name) => {
    return db.find(stock => stock.name === name);
}

module.exports.put = (stock) => {
    db.push(stock);
    fs.writeFileSync(DB_FILE, JSON.stringify(db));
}

```

Сервер C

src/app.js

```

const express = require("express");
const body_parser = require("body-parser");
const path = require("path");

const router = require("./router");
const logger = require("./controllers").logger;

const port = process.env.PORT | 3000;
const VIEWS_FOLDER = path.join(__dirname, "..", "views");
const PUBLIC_FOLDER = path.join(__dirname, "..", "public");

const app = express();

app.set("view engine", "ejs");
app.set("views", VIEWS_FOLDER)

app.use(body_parser.urlencoded({ extended: false }));
app.use(express.static(PUBLIC_FOLDER))
app.use(logger);

app.use(router);

app.listen(port);
console.log(`Running on port ${port}`);

```

src/controllers.js

```

"use strict";

const axios = require("axios");

const CARS_SERVER = "http://localhost:4001/";
const INSERT_QUERY = "insert/record";
const SELECT_QUERY = "select/record";
const STOCKS_SERVER = "http://localhost:4002/";

```

```

const HOME_PAGE = "index.ejs";
const CREATE_CAR_PAGE = "create_car.ejs";
const CREATE_STOCK_PAGE = "create_stock.ejs";
const SEARCH_CAR_PAGE = "search_car.ejs";
const SEARCH_STOCK_PAGE = "search_stock.ejs";
const CAR_PAGE = "car.ejs";
const STOCK_PAGE = "stock.ejs";

module.exports.render_create_car = (_, res) => {
  res.render(CREATE_CAR_PAGE, { meta: {} });
}

module.exports.create_car = (req, res) => {
  axios.post(CARS_SERVER + INSERT_QUERY, {
    name: req.body.name,
    price: req.body.price
  }).then(result => {
    if (result.status !== 200)
      throw Error(result.data.result);

    res.status(200).render(CREATE_CAR_PAGE, { meta: { log: "Car has been created!" } });
  }).catch(err => {
    res.status(400).render(CREATE_CAR_PAGE, { meta: { error: `Can't create a car: ${err}` } });
  });
}

module.exports.search_car = (_, res) => {
  res.render(SEARCH_CAR_PAGE);
}

module.exports.show_car = (req, res) => {
  axios.get(CARS_SERVER + SELECT_QUERY, {
    params: {
      name: req.query.name
    }
  }).then(result => {
    res.status(200).render(CAR_PAGE, { car: result.data.car, error: null });
  }).catch(() => {
    res.status(400).render(CAR_PAGE, { car: {}, error: "Car wasn't found!" });
  })
}

module.exports.render_create_stock = (_, res) => {
  res.render(CREATE_STOCK_PAGE, { meta: {} });
}

module.exports.create_stock = (req, res) => {
  axios.post(STOCKS_SERVER + INSERT_QUERY, {
    name: req.body.name,
    cars: req.body.cars.split(' ').filter(str => str !== '')
  }).then(result => {
    console.log(`RESULT: ${result}`)
    res.status(200).render(CREATE_STOCK_PAGE, { meta: { log: "Stock has been created!" } });
  }).catch(err => {
    res.status(400).render(CREATE_STOCK_PAGE, { meta: { error: `Can't create a stock: ${err}` } });
  });
}

module.exports.search_stock = (_, res) => {
  res.render(SEARCH_STOCK_PAGE);
}

module.exports.show_stock = (req, res) => {
  axios.get(STOCKS_SERVER + SELECT_QUERY, {
    params: {
      name: req.query.name
    }
  })
}

```

```

        }
    }).then(result => {
        console.log(result.data.stock);
        res.status(200).render(STOCK_PAGE, { stock: result.data.stock, error: null });
    }).catch(() => {
        res.status(400).render(STOCK_PAGE, { stock: {}, error: "Stock wasn't found" });
    })
}

module.exports.default_controller = (_, res) => {
    res.render(HOME_PAGE);
}

module.exports.logger = (req, _, next) => {
    console.log(`${req.url} ${req.method}`);

    if (req.method === "GET")
        console.log(req.query);
    else if (req.method === "POST")
        console.log(req.body);

    next();
}

```

src/router.js

```

const Router = require("express").Router;

const controllers = require("../controllers");

const router = Router();

router.get("/create-car", controllers.render_create_car);
router.post("/create-car", controllers.create_car);

router.get("/search-car", controllers.search_car);
router.get("/car", controllers.show_car);

router.get("/create-stock", controllers.render_create_stock);
router.post("/create-stock", controllers.create_stock);

router.get("/search-stock", controllers.search_stock);
router.get("/stock", controllers.show_stock);

router.post("/create-car", controllers.create_car);
router.use(controllers.default_controller);

module.exports = router;

```

views/car.ejs

```

<%- include("includes/header.ejs") %>
  <link rel="stylesheet" href="/style.css">
  <title>Car <%= car.name %></title>
</head>
<body>
  <%- include("includes/navbar.ejs", { page: "car" }) %>
  <% if (error) { %>
    <div class="err-card">
      <h1><%=error%></h1>
    </div>
  <% } else { %>
    <div class="simple-card">
      <h1>Found car!</h1>
      <p class="bp">Name: <%=car.name%></p>
      <p class="bp">Price: <%=car.price%></p>
    </div>
  <% } %>

  <%- include("includes/end.ejs") %>

```

views/create_car.ejs

```

<%- include("includes/header.ejs") %>
  <link rel="stylesheet" href="/style.css">
  <title>Create car</title>
</head>
<body>
  <%- include("includes/navbar.ejs", { page: "ccar" }) %>

  <% if (meta.log) { %>
    <div class="log-card">
      <h1><%=meta.log%></h1>
    </div>
  <% } %>
  <% if (meta.error) { %>
    <div class="err-card">
      <h1><%=meta.error%></h1>
    </div>
  <% } %>

  <div class="simple-card">
    <h1>Create your car today!</h1>
    <form method="POST" action="/create-car">
      <input type="text" name="name" placeholder="Enter name...">
      <br>
      <input type="number" name="price" placeholder="Enter price...">
      <br>
      <button type="submit">Create</button>
    </form>
  </div>

  <%- include("includes/end.ejs") %>

```

views/create_stock.ejs


```

<%- include("includes/header.ejs") %>
  <link rel="stylesheet" href="/style.css">
  <title>Create stock</title>
</head>
<body>
  <%- include("includes/navbar.ejs", { page: "cstock" }) %>

  <% if (meta.log) { %>
    <div class="log-card">
      <h1><%=meta.log%></h1>
    </div>
  <% } %>
  <% if (meta.error) { %>
    <div class="err-card">
      <h1><%=meta.error%></h1>
    </div>
  <% } %>

  <div class="simple-card">
    <h1>Create your stock today!</h1>
    <form method="POST" action="/create-stock">
      <input type="text" name="name" placeholder="Enter name...">
      <br>
      <input type="text" name="cars" placeholder="Enter cars splitted by space...">
      <br>
      <button type="submit">Create</button>
    </form>
  </div>

<%- include("includes/end.ejs") %>

```

views/index.ejs

```

<%- include("includes/header.ejs") %>
  <link rel="stylesheet" href="/style.css">
  <title>Lab 7</title>
</head>
<body>
  <%- include("includes/navbar.ejs", { page: "Home" }) %>

  <div style="flex-direction: space-between; display: flex;" class="simple-card">

    <a class="menu" href="/search-car">Search a car</a>
    <a class="menu" href="/search-stock">Search a stock</a>
    <a class="menu" href="/create-car">Create a car</a>
    <a class="menu" href="/create-stock">Create a stock</a>

  </div>

<%- include("includes/end.ejs") %>

```

views/search_car.ejs

```
<%- include("includes/header.ejs") %>
  <link rel="stylesheet" href="/style.css">
  <title>Search car</title>
</head>
<body>
  <%- include("includes/navbar.ejs", { page: "cinfo" }) %>
  <div class="simple-card">
    <h1>Here you can find any car I have!</h1>
    <form method="GET" action="/car">
      <input type="text" name="name" placeholder="Enter name...">
      <br>
      <button type="submit">Find</button>
    </form>
  </div>

<%- include("includes/end.ejs") %>
```

views/search_stock.ejs

```
<%- include("includes/header.ejs") %>
  <link rel="stylesheet" href="/style.css">
  <title>Search stock</title>
</head>
<body>
  <%- include("includes/navbar.ejs", { page: "sinfo" }) %>
  <div class="simple-card">
    <h1>Here you can find any stock I have!</h1>
    <form method="GET" action="/stock">
      <input type="text" name="name" placeholder="Enter name...">
      <br>
      <button type="submit">Find</button>
    </form>
  </div>

<%- include("includes/end.ejs") %>
```

views/stock.ejs

```

<%- include("includes/header.ejs") %>
  <link rel="stylesheet" href="/style.css">
  <title>Stock <%= stock.name %></title>
</head>
<body>
  <%- include("includes/navbar.ejs", { page: "car" }) %>
  <% if (error) { %>
    <div class="err-card">
      <h1><%=error%></h1>
    </div>
  <% } else { %>
    <div class="simple-card">
      <h1>Found stock <span class="inv"><%=stock.name%></span>!</h1>

      <% if (stock.cars.length) { %>
        <h1>Cars:</h1>

        <% for (const car of stock.cars) { %>
          <div class="simple-card">
            <p class="bp">Name: <%=car%></p>
          </div>
        <% } %>

      <% } else { %>
        <div class="err-card">
          <h1>No cars in stock!</h1>
        </div>
      <% } %>
    </div>
  <% } %>

<%- include("includes/end.ejs") %>

```

views/include/navbar.ejs

```

<header>
  <nav>
    <a class=<%= page === "Home" ? "active" : "passive"%> href="/">Home</a>
    <a class=<%= page === "ccar" ? "active" : "passive"%> href="/create-car">Create car</a>
    <a class=<%= page === "cinfo" ? "active" : "passive"%> href="/search-car">Car info</a>
    <a class=<%= page === "cstock" ? "active" : "passive"%> href="/create-stock">Create stock</a>
    <a class=<%= page === "sinfo" ? "active" : "passive"%> href="/search-stock">Stock info</a>
  </nav>
</header>

```

views/include/header.ejs

```

<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">

```

views/include/end.ejs

```

</body>
</html>

```

Задание 2

Условие

Написать скрипт, который принимает на вход число и считает его факториал. Скрипт должен получать параметр через process.argv.

Написать скрипт, который принимает на вход массив чисел и выводит на экран факториал каждого числа из массива. Скрипт принимает параметры через process.argv.

При решении задачи вызывать скрипт вычисления факториала через execSync.

Код программы

Язык: **JavaScript**

main.js

```
"use strict";

const exec = require("child_process").execSync;

const fact = (num) => {
  const value = exec(`node fact.js ${num}`);
  return parseInt(value);
}

const array = process.argv.slice(2, process.argv).map(num => parseInt(num));
for (const number of array)
  console.log(`fact(${number}) = ${fact(number)}`);
```

fact.js

```
"use strict";

const fact = (num) => {
  let res = 1;

  for (let i = 2; i <= num; i++)
    res *= i;

  return res;
}

console.log(fact(process.argv[2]));
```

Результаты тестирования

Все тесты пройдены успешно.

Вывод

В результате работы было освоено взаимодействие между серверами и командами терминала.