

**САНКТ-ПЕТЕРБУРГСКИЙ НАЦИОНАЛЬНЫЙ
ИССЛЕДОВАТЕЛЬСКИЙ УНИВЕРСИТЕТ ИТМО**

Дисциплина: Фронтэнд разработка

Отчёт

Лабораторная работа №2

Выполнила:

Шурубова Прасковья

Группа: К3343

Проверил:

Добряков Д. И.

Санкт-Петербург

2024 г.

Задача

Варианты остаются прежними. Теперь Вам нужно привязать то, что Вы делали в ЛР1 к внешнему API средствами fetch/axios/xhr.

Ход работы

С помощью команды `npm install json-server` скачала json-server, затем создала `db.json` с таким наполнением.

```
{
  "restaurants": [
    {
      "id": "1",
      "name": "Mircuccio Osteria",
      "cuisine": "Итальянская",
      "details": "📍 ул.Моисеенко, д.27",
      "station": "Площадь Александра Невского 2",
      "price": "2000",
      "restaurant-phone": "+7 (812) 407-1908",
      "workingHours": "Пн - Вс: 12:00 - 23:00",
      "description": "Mircuccio — остерия с итальянской атмосферой, тёплым интерьером и авторскими блюдами.",
      "menu": [
        "Блюдо 1",
        "Блюдо 2",
        "Блюдо 3"
      ],
      "restaurant-photo": "photo/mercuccio1.jpg",
      "reviews": []
    },
    {
      "id": "2",
      "name": "My Asiatique",
      "cuisine": "Китайская, Японская, Индийская",
      "details": "📍 ул.Разъезжая, д.3",
      "station": "Владимирская, Достоевская, Звенигородская",
      "price": "1000",
      "restaurant-phone": "+7 (812) 407-1908",
      "workingHours": "Пн - Чт, Вс: 12:00 - 23:00  Пт - Сб: 12:00 - 1:00",
      "description": "Ресторан создан от команды старейшего ресторана национальной индийской кухни Tandoor.",
      "menu": [
        "Блюдо 1",
        "Блюдо 2",
        "Блюдо 3"
      ],
      "restaurant-photo": "photo/my_asiatique1.jpg",
      "reviews": []
    },
    {
      "id": "3",
      "name": "Roze & Dance",
      "cuisine": "Французская",
      "details": "📍 ул.Инженерная, д.13",
      "station": "Гостиный двор, Невский проспект",
      "price": "2500",
      "restaurant-phone": "+7 (812) 407-1908",
      "workingHours": "Пн - Чт, Вс: 12:00 - 00:00  Пт - Сб: 12:00 - 3:00",
      "description": "ROZÉ & Dance: fine dining-БОЖ от шеф-повара Ульяны Суздалкиной.",
      "menu": [
```

```

        "Блюдо 1",
        "Блюдо 2",
        "Блюдо 3"
    ],
    "restaurant-photo": "photo/rozel.jpg",
    "reviews": []
},
{
    "id": "4",
    "name": "Meal",
    "cuisine": "Европейская, Скандинавская",
    "details": "★ Литейный пр-т, 17-19",
    "station": "Чернышевская",
    "price": "1200",
    "restaurant-phone": "+7 (812) 407-1908",
    "workingHours": "Пн - Вс: 13:00 - 23:00",
    "description": "Meal на Литейном - ресторан-гастробистро в центре
города.",
    "menu": [
        "Блюдо 1",
        "Блюдо 2",
        "Блюдо 3"
    ],
    "restaurant-photo": "photo/meal1.jpg",
    "reviews": []
},
{
    "id": "5",
    "name": "Тбилисо",
    "cuisine": "Грузинская, Кавказская",
    "details": "★ ул.Сытнинская, д.10",
    "station": "Горьковская",
    "price": "800",
    "restaurant-phone": "+7 (812) 407-1908",
    "workingHours": "Пн - Вс: 12:00 - 23:00",
    "description": "«Тбилисо» - это залитый мягким светом уголок
грузинского двора.",
    "menu": [
        "Блюдо 1",
        "Блюдо 2",
        "Блюдо 3"
    ],
    "restaurant-photo": "photo/тбилисо1.jpg",
    "reviews": []
}
],
"articles": [
    {
        "id": 1,
        "title": "Halloween в ресторанах Петербурга",
        "image": "photo/halloween.jpg",
        "description": ""
    },
    {
        "id": 2,
        "title": "Где устроить свадьбу в Петербурге",
        "image": "photo/свадьба.jpg",
        "description": ""
    },
    {
        "id": 3,
        "title": "Интересные мероприятия в ресторанах Петербурга",
        "image": "photo/class.jpg",

```

```

    "description": ""
  },
],
"users": []
}

```

Далее передала мои данные в json server:

`npx json-server --watch db.json`

```

PS C:\Users\User\Desktop\ИТМО\3 курс\Фронтенд\2_lab> npx json-server --watch db.json
(node:4000) ExperimentalWarning: CommonJS module C:\Users\User\AppData\Nvm\v23.2.0\node_modules\npm\node_modules\deb
ers\User\AppData\Nvm\v23.2.0\node_modules\npm\node_modules\supports-color\index.js using require().
Support for loading ES Module in require() is an experimental feature and might change at any time
(Use `node --trace-warnings ...` to show where the warning was created)

Static files:
Serving ./public directory if it exists

Endpoints:
http://localhost:3000/restaurants
http://localhost:3000/articles
http://localhost:3000/users

```

Изначально были только рестораны, список статей, которые на главной странице, также создала массив users для дальнейшего автоматического заполнения. Элементы, которые вносятся в массив к users.

```

const user = {
  username: document.getElementById('register-username').value,
  surname: document.getElementById('register-surname').value,
  email: document.getElementById('register-email').value,
  password: document.getElementById('register-password').value,
  favorites: [],
  bookings: []
};

```

Всё записывается с помощью метода POST на <http://localhost:3000/users>.

При авторизации мы запоминаем currentUser, чтобы в дальнейшем была возможность обновлять его данные в профиле пользователя, сохранять избранные рестораны, делать бронирования и оставлять комментарии под именем пользователя, то есть персонализировать его действия.

Я использовала и библиотеку axios, и библиотеку fetch, но чаще обрабатывала запросы через axios, так как синтаксис показался более простым и понятным.

В профиле пользователь может обновить следующие данные:

```

const updatedUserData = {
  username: document.getElementById('register-username').value,
  surname: document.getElementById('register-surname').value,
  email: document.getElementById('register-email').value,
  password: document.getElementById('register-password').value,
};

```

История бронирования, избранные рестораны и id пользователя остаются в первоначальном виде.

```
const updatedUserDataFromServer = await updateResponse.json();
const updatedCurrentUser = {
  ...currentUser,
  ...updatedUserDataFromServer,
};
```

Все дальнейшие действия будут доступны только авторизовавшемуся пользователю, и это будут проверяться:

```
const currentUser = JSON.parse(localStorage.getItem('currentUser'));
if (!currentUser) {
  alert('Для выполнения этой операции требуется авторизация');
  return;
}
```

Данные, которые я записываю при бронировании.

```
const bookingData = {
  date: document.getElementById('booking-date').value,
  restaurantName: document.getElementById('restaurant').innerText,
  restaurantLocation: document.getElementById('details').innerText,
  people: document.getElementById('number-of-people').value,
  name: document.getElementById('name_book').value,
  phone: document.getElementById('phone').value,
  email: document.getElementById('email').value,
  comments: document.getElementById('comments').value
};
```

Пример отображения:

```
"bookings": [
  {
    "date": "2024-11-23",
    "restaurantName": "Roze & Dance",
    "restaurantLocation": "📍 ул.Инженерная, д.13",
    "people": "4",
    "name": "Михаил Сидоров",
    "phone": "+7 (812) 1234567",
    "email": "jackson@gmail.com",
    "comments": "караоке"
  }
]
```



Столик, пожалуйста!

Выйти Личный кабинет ▾

История заказов

Бронирование на 2024-11-23 в Roze & Dance
Количество человек: 4
Имя: Михаил Сидоров
Телефон: +7 (812) 1234567
Email: jackson@gmail.com
Пожелания: караоке

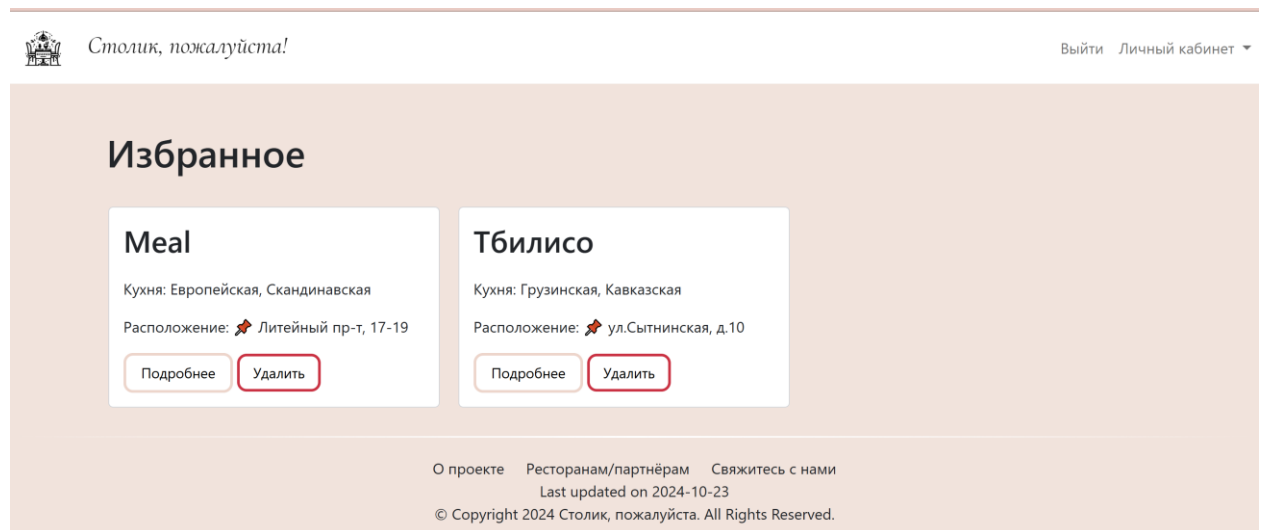
О проекте Ресторанам/партнёрам Свяжитесь с нами
Last updated on 2024-10-23
© Copyright 2024 Столик, пожалуйста. All Rights Reserved.

Данные, которые фиксируется при добавлении:

```
const restaurant = {
  id: restaurantId,
  name: document.getElementById('restaurant').innerText,
  cuisine: document.getElementById('cuisine').innerText,
  location: document.getElementById('details').innerText,
};
```

Пример отображения:

```
"favorites": [
  {
    "id": "4",
    "name": "Meal",
    "cuisine": "Европейская, Скандинавская",
    "location": "📍 Литейный пр-т, 17-19"
  },
  {
    "id": "5",
    "name": "Тбилисо",
    "cuisine": "Грузинская, Кавказская",
    "location": "📍 ул.Сытнинская, д.10"
  }
]
```



При добавлении отзывов данные заносятся в массив ресторана.

```
const newReview = {
  user: user.username,
  text: reviewText,
  rating: reviewRating,
  date: new Date().toISOString()
};
```

```
"reviews": [
  {
    "user": "Christian",
    "text": "Вкусные блюда по доступным ценам!",
    "rating": 5,
    "date": "2024-11-13T14:46:43.742Z"
  },
  {
```

```
"user": "Bobby",  
"text": "Всё супер, но большое время ожидания",  
"rating": 4,  
"date": "2024-11-13T14:47:36.848Z"  
}
```

Для отображения отзыва определённого ресторана, извлекаем id из url.

```
document.addEventListener('DOMContentLoaded', () => {  
  const restaurantId = getRestaurantIdFromUrl();  
  
  if (restaurantId) {  
    loadReviews(restaurantId);  
  } else {  
    console.error("Не удалось получить ID ресторана.");  
  }  
});
```

Пример отображения отзывов:

Отзывы

Christian (13.11.2024):

Вкусные блюда по доступным ценам!

Рейтинг: 5

Bobby (13.11.2024):

Всё супер, но большое время ожидания

Рейтинг: 4

Michael (13.11.2024):

Атмосфера не понравилась, очень громко играла музыка

Рейтинг: 3

Вывод: в ходе работы я научилась работать с имитационным API, json-server, обрабатывать запросы с помощью fetch и axios, а также структурировала свой проект и научилась обрабатывать POST, GET, PUT, DELETE запросы.