

Fullstack-разработка

Vue

Axios

Работа с датой

Цель урока:

1. Соединим клиент на Vue и сервер на Express
2. Узнаем основы работы с датой

Разбор ДЗ

Тестирование

Практика

Готовим клиент

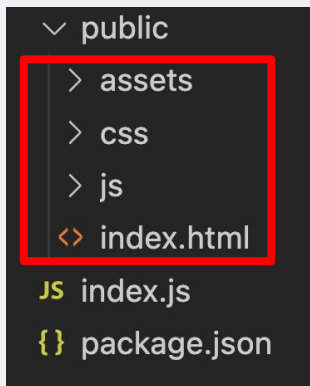
Теория

Клиент и сервер

Что мы уже знаем?

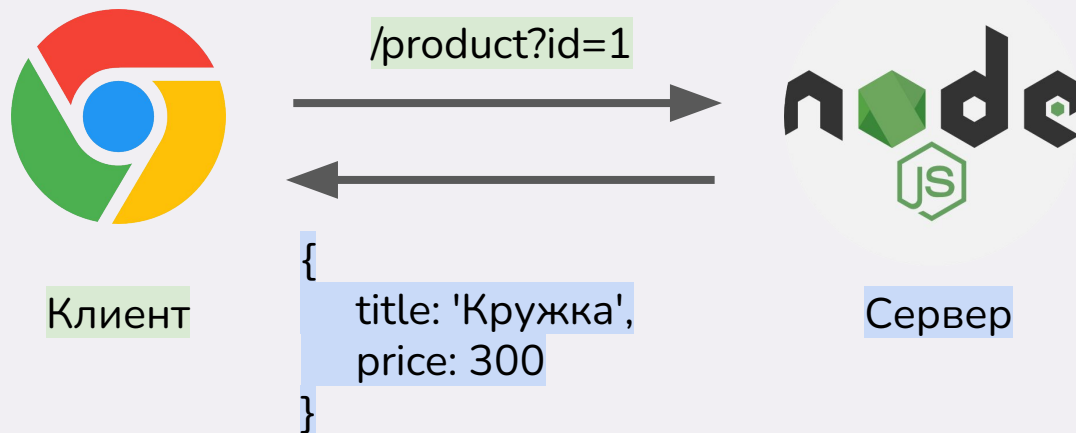
Статические файлы

```
app.use(express.static(`public`));
```



Весь клиентский код
живёт в папке **public**
внутри сервера

Single Page Application



Клиент запрашивает у сервера
данные и выводит их на экран

Axios

Клиент отправляет запросы на сервер
с помощью библиотеки **axios**

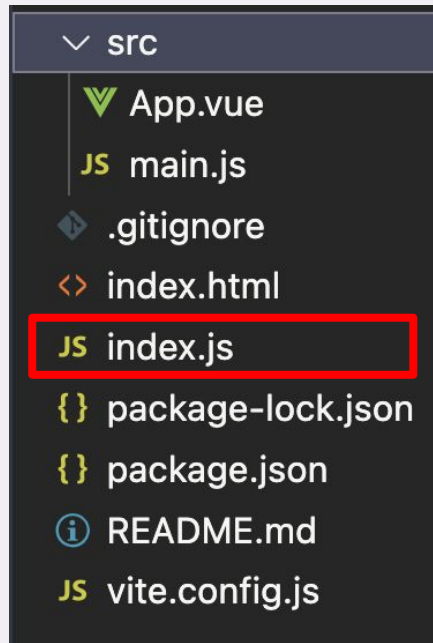
```
async function loadProducts() {  
  let response = await axios.get('/all');  
  let products = response.data;  
  
  console.log(products);  
}
```

С приходом Vue всё меняется!

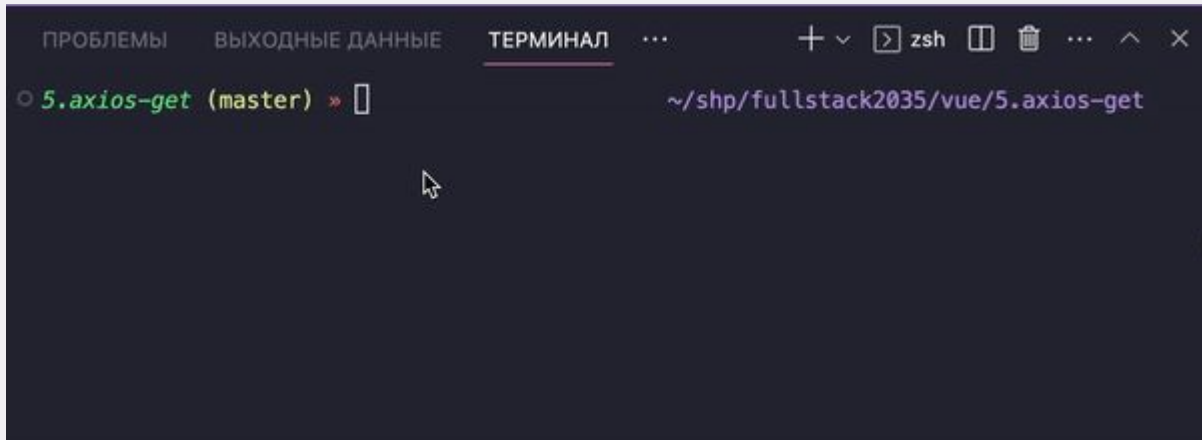
немножко :)

Структура проекта

Код сервера `index.js`
переезжает жить
в проект на **Vue**



Два терминала



<http://localhost:5173> – клиент

<http://localhost:3005> – сервер

CORS



CORS – это политика безопасности, которая защищает сервер от взаимодействия с **неправильными** клиентами

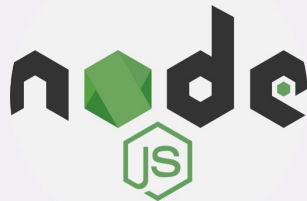
CORS

CORS срабатывает, если клиент и сервер живут на разных доменах.

<http://localhost:5173> – клиент

<http://localhost:3005> – сервер

Я хочу ответить,
но не могу.
Что делать?



1. Настроить CORS
2. Настроить AXIOS
3. Загрузить данные в компонент

Настройка CORS

Нужно явно разрешить серверу работать с нашим клиентом через пакет `cors`.

```
npm install cors
```

```
let cors = require('cors');  
  
app.use(cors({  
  origin: 'http://localhost:5173'  
}));
```

Настройка Axios

Для отправки запросов из компонента App.vue нам пригодится пакет axios:

```
npm install axios
```

```
import axios from "axios";
```

```
// Адрес сервера
```

```
axios.defaults.baseURL =
```

```
'http://localhost:3005';
```

Метод запроса

Вместо асинхронной функции теперь будем объявлять **асинхронный метод** компонента.

```
methods: {  
  async loadProducts() {  
    let response = await axios.get('/all');  
  }  
}
```

Метод запроса

Ответ сервера можно и нужно сохранять в **data-свойство**, чтобы произошла перерисовка компонента.

```
methods: {  
  async loadProducts() {  
    let response = await axios.get('/all');  
    this.products = response.data;  
  }  
}
```

mounted

Специальный метод `mounted` срабатывает при первой отрисовки приложения.

```
data() { ... },  
  
methods: { ... }  
  
mounted() {  
  this.loadProducts();  
}
```

Идеально для загрузки данных!



Практика

Подключаем сервер

Теория

Работа с датой

13 сентября – День программиста

Как можно представить
эту дату для человека?

**13 сентября –
День программиста**

13.09.23
Через 3 месяца
September 13

В каком виде MongoDB
хранит информацию
о дате и времени?

2023-09-13T15:00:00.000Z

Стандарт ISO 8601

Чтобы переводить между ISO
и человеческим, нам поможет
библиотека **datejs**

2023-09-13T15:00:00.000Z



13.09.23

Настройка datejs

Для работы с датой подключи **datejs** в компонент App.vue.

```
npm install datejs
```

```
import dayjs from 'dayjs';  
import 'dayjs/locale/ru';  
dayjs.locale('ru');
```

Создание даты

```
// сейчас
```

```
dayjs();
```

```
// без времени
```

```
dayjs('2023-09-13');
```

```
// со временем
```

```
dayjs('2023-09-13 15:20:00');
```

```
// из iso
```

```
dayjs('2023-09-13T15:20:00.000Z');
```

Формат даты

Метод `format` записывает дату в заданном по таблице формате.

```
let iso = '2023-09-13T15:00:00.000Z';  
let day = dayjs(iso);
```

```
day.format('YYYY'); // год  
day.format('HH:mm'); // время  
day.format('DD.MM'); // дата
```

Перевод в ISO

Метод `toISOString` приводит любую дату к формату ISO строки.

```
let day = dayjs('2020-09-15');  
  
let iso = day.toISOString();  
  
// 2020-09-14T21:00:00.000Z
```

Практика

Выбор даты

Поиск

Цели урока:

- ✓ Соединим клиент на Vue и сервер на Express
- ✓ Узнаем основы работы с датой

Домашнее задание