

Fullstack-разработка

Vue

POST-запросы

Цель урока:

1. Потренироваться делать разные запросы к серверу
2. Научиться следить за обновлениями на сервере

Разбор ДЗ

Что получилось?

Практика

Магазин растений

Повторение

POST-запрос

POST-запрос

- Безопасен для передачи данных, так как использует **тело запроса**
- Используется для добавления, удаления и редактирования данных в БД

Форма

Для Vue поля **формы** нужно привязать к **data-свойствам** через директиву **v-model**.

```
<form>
```

```
  <input type="text" v-model="username">
```

```
  <input type="password" v-model="password">
```

```
  <button type="submit">
```

```
    Отправить
```

```
  </button>
```

```
</form>
```

```
data() {  
  return {  
    username: '',  
    password: ''  
  }  
}
```

@submit

@submit – это событие, которое срабатывает на теге **form** в тот момент, когда пользователь нажимает на кнопку с **type="submit"**

```
<form @submit="send">  
  <input type="text" v-model="username">  
  <input type="password" v-model="password">  
  <button type="submit">  
    Отправить  
  </button>  
</form>
```

evt.preventDefault()

Браузер реагирует на событие submit
перезагрузкой страницы.

Чтобы он так не делал, нам нужно **остановить**
действие браузера **по умолчанию:**

```
methods: {  
  function send(evt) {  
    evt.preventDefault();  
    console.log('приём-приём');  
  }  
}
```

POST-запрос

При отправке запроса данные из **data-свойств** полей ввода передаются на сервер **в теле запроса**.

```
methods: {  
  async function send(evt) {  
    evt.preventDefault();  
    await axios.post('user/login', {  
      username: this.username,  
      password: this.password  
    })  
  }  
}
```

Практика

Сезон посадок

Теория

Long polling

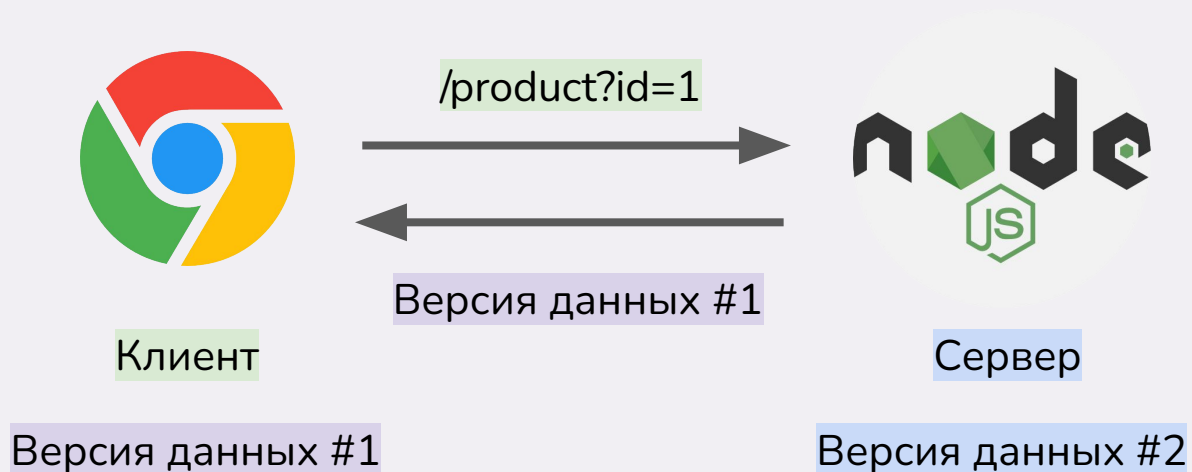
При загрузке страницы компонент
получает данные с сервера.

Но давайте задумаемся...

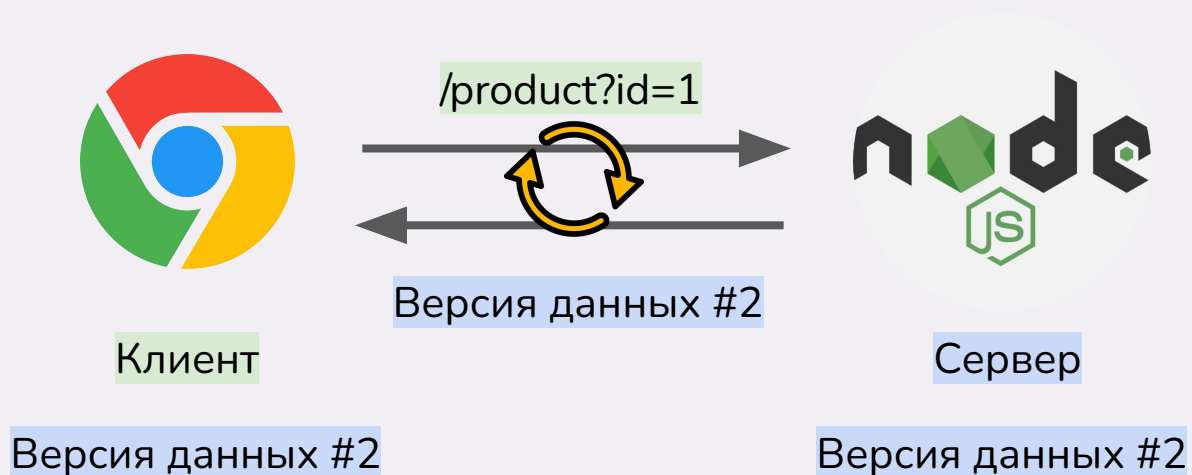
Могут ли данные на сервере измениться?

**Данные могут меняться
между перезагрузками
страницы**

1. Другая вкладка
2. Другие пользователи
3. Изменения в системе



Как клиенту узнать об изменениях?



Long polling – регулярный запрос за обновлениями

Например, 1 раз в минуту

setInterval

Чтобы запланировать повторение, в JS есть специальный метод, который работает как в браузере, так и в NodeJS:

```
let i = 0;

setInterval( function() {
    i++;
    console.log(i);
}, 1000);
```



setInterval

Метод `setInterval` вызывает переданную функцию 1 раз в указанное число миллисекунд.

```
let i = 0;

setInterval( function() {
    i++;
    console.log(i);
}, 1000); 1000мс = 1с
```



setInterval

При использовании **обычной функции** внутри таймера пропадает доступ к **this**.

Поэтому мы будем использовать **стрелочную функцию** – с ней такой проблемы не возникает.

```
mounted() {  
  setInterval( () => {  
    this.loadData();  
  }, 1000);  
}
```

Таймер бывает полезен...

1. Для показа рекламы
2. Для анимации прогресса
3. Для получения обновлений с сервера
4. Для игр!

Практика

Странные цены

Цели урока:

- ✓ Потренировались делать разные запросы к серверу
- ✓ Научились следить за обновлениями на сервере

Домашнее задание

Сбор урожая