



20.35
УНИВЕРСИТЕТ



Минцифры
России



Fullstack-разработка

JSON

POST запросы

Цель урока:

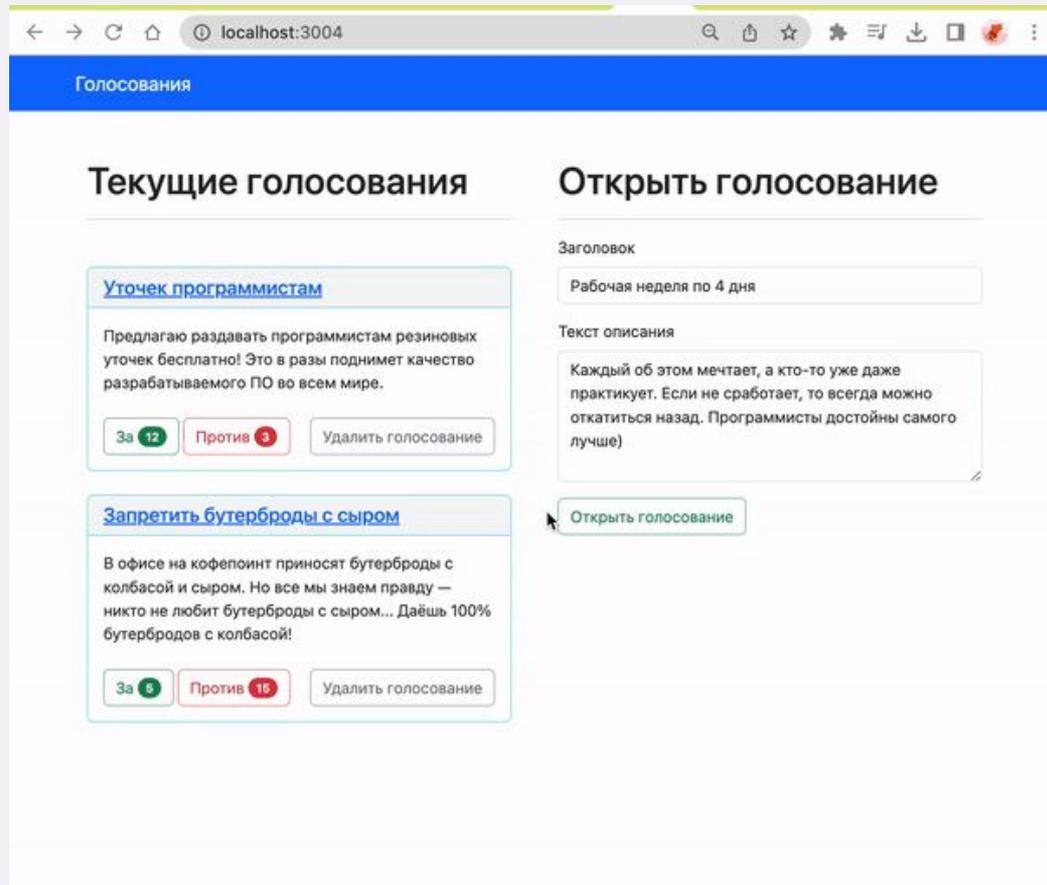
Научимся отправлять
POST запросы через axios

Узнаем, как отправить форму
незаметно для пользователя

Разбор ДЗ

Что получилось?

Что было трудно?



The screenshot shows a web browser window with the address bar displaying "localhost:3004". The page title is "Голосования". The main content is divided into two columns. The left column, titled "Текущие голосования", contains two voting items. The first item is "Уточек программистам" with 12 "За" votes and 3 "Против" votes. The second item is "Запретить бутерброды с сыром" with 5 "За" votes and 15 "Против" votes. The right column, titled "Открыть голосование", contains a form for a new poll with the title "Рабочая неделя по 4 дня" and a description: "Каждый об этом мечтает, а кто-то уже даже практикует. Если не сработает, то всегда можно откатиться назад. Программисты достойны самого лучше)". A button labeled "Открыть голосование" is visible at the bottom of the right column.

Голосования

Текущие голосования

[Уточек программистам](#)

Предлагаю раздавать программистам резиновых уточек бесплатно! Это в разы поднимет качество разрабатываемого ПО во всем мире.

За 12 Против 3 Удалить голосование

[Запретить бутерброды с сыром](#)

В офисе на кофепоинт приносят бутерброды с колбасой и сыром. Но все мы знаем правду — никто не любит бутерброды с сыром... Дайшь 100% бутербродов с колбасой!

За 5 Против 15 Удалить голосование

Открыть голосование

Заголовок

Рабочая неделя по 4 дня

Текст описания

Каждый об этом мечтает, а кто-то уже даже практикует. Если не сработает, то всегда можно откатиться назад. Программисты достойны самого лучше)

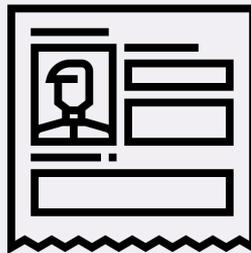
Открыть голосование

Практика

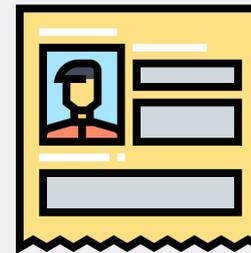
Настрой сервер

Теория

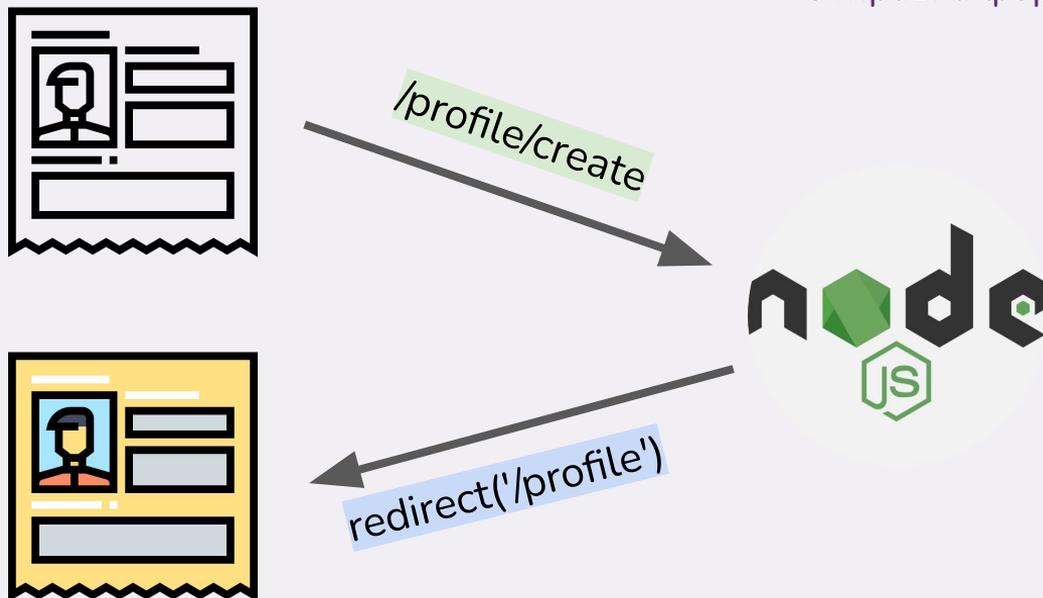
Отправка формы



Форма создания

Страница с
результатом

Представим, что у нас сайт с серверной шаблонизацией и на нём **есть 2 элемента**



При отправке формы **браузер** делает запрос на сервер и перезагружает страницу

С помощью **axios** мы сможем отправлять данные формы без перезагрузки страницы.

Для этого клиенту и серверу нужно начать общаться в формате **JSON**.

Настройка сервера

Было:

```
app.use(express.urlencoded({ extended: true }));
```

Стало:

```
app.use(express.json());
```



Обработка запроса

Начинай решать любую задачу с создания роута, который будет обрабатывать запрос.

```
app.post(`/login`, function (req, res) {  
  let username = req.body.username;  
  ...  
})
```

Допустим, сервер готов.

Дальше нужно настроить клиент...

Вот где начинается настоящая работа!

Настройка axios

```
axios.defaults.headers.post['Content-Type'] =  
'application/json';
```

Content-Type – это заголовок, через который клиент и сервер сообщают друг другу тип пересылаемого сообщения.

Так axios скажет серверу, что отправляет **JSON**.

Тег form

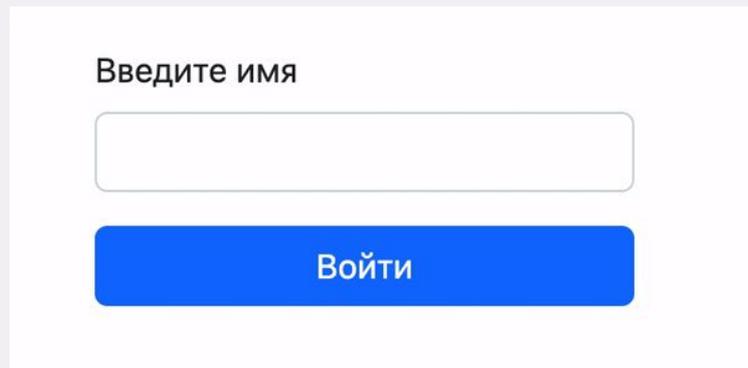
Для тега **form** больше не нужно указывать **method** или **action**.

Зато **id** или **class** пригодятся: так ты сможешь обратиться к ним через JS.

```
<form id="register">  
  <input type="text" name="username">  
  <input type="password" name="password">  
  <button type="submit">  
    Отправить  
  </button>  
</form>
```

Событие submit

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



Введите имя



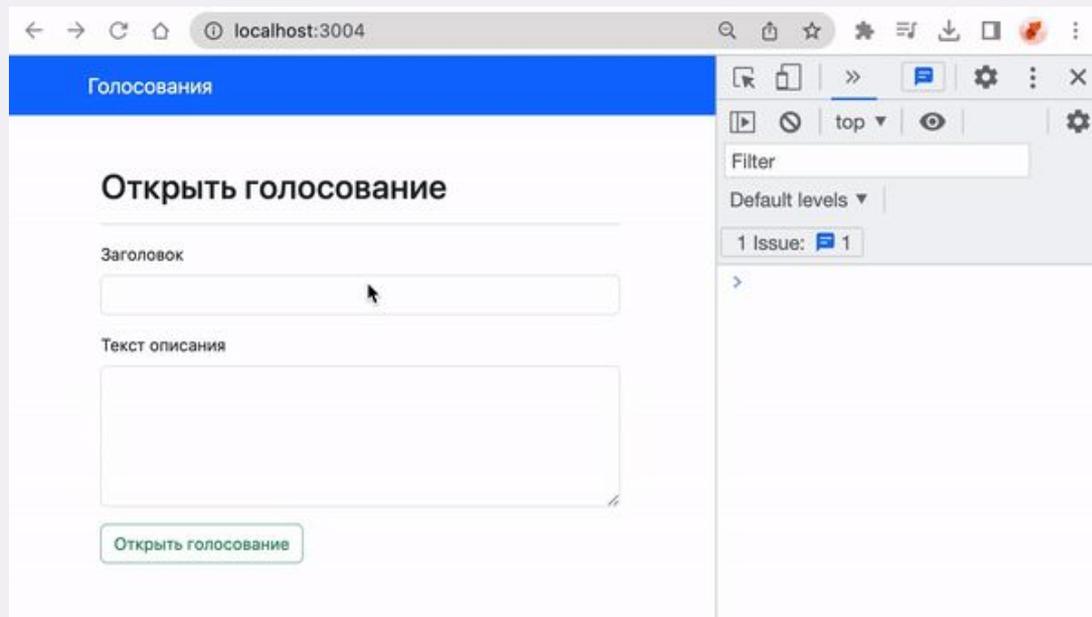
Событие submit

Раз это событие – давайте подпишемся!

```
let form = document.querySelector('#register');  
form.addEventListener('submit', function () {  
    console.log('приём-приём');  
});
```

Событие submit

Сообщение появляется на долю секунды, но сразу исчезает. Почему???



evt.preventDefault()

Браузер реагирует на событие submit **перезагрузкой** страницы.

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

```
let form = document.querySelector('#register');
form.addEventListener(`submit`, function (evt) {
  evt.preventDefault();
  console.log('приём-приём');
});
```

Отправка формы

Метод `post` у `axios` отправляет POST-запрос на роут `/register`, передавая данные полей ввода из формы `form` в виде JSON

```
axios.post('/register', form);
```

Отправка формы

Полный код отправки формы на сервер:

```
let form = document.querySelector('#register');
form.addEventListener('submit', async function (evt) {
  evt.preventDefault();
  let response = await axios.post('/register', form);
  console.log(response.data);
});
```

Если мы отправляем
запрос на создание, что
должен вернуть сервер?

Объект, который создал

Итоги

1. Создай **роут** для POST-запроса на сервере
2. Создай **форму** в HTML
3. Подпишись на **submit** формы
4. Останови **действие по умолчанию**
5. Отправь POST-запрос через **axios**

Практика

Создаём новое голосование

Теория

Отправка POST-запроса

При серверном
рендеринге все действия
на сайте совершаются
через **формы** или **ссылки**.

Это очень ограничивает!



Пылесос

Техника: 10000 руб

Пылесос в рабочем состоянии, продаю, так как покупаем новый.

Удалить

Ссылки обновляют страницу
(собственно, в этом их идея...)



Пылесос

[Удалить](#)

Техника: 10000 руб

Пылесос в рабочем состоянии, продаю, так как покупаем новый.

Ссылки могут отправлять
только GET-запросы

А это значит, что пора вернуть в нашу жизнь старые добрые кнопки!



POST-запрос

```
button.addEventListener(`click`,  
function () {  
    Так, а evt.preventDefault нужен?  
})
```

Нет, вызываем этот метод только для формы.

Кнопки и так прекрасно работают)

POST-запрос

```
button.addEventListener(`click`,  
  async function () {  
  
  })
```

Если в обработчике делаешь запросы,
добавляй **async**.

POST-запрос

```
button.addEventListener(`click`,
  async function () {
    await axios.post(`/edit`, {
      id: 123,
      title: 'Покормить кота'
    })
  })
})
```

*Тело запроса, попадёт
в req.body на сервере*

Если мы отправляем
запрос на удаление, что
должен вернуть сервер?

Сообщение об успехе!

```
res.send('OK');
```

Практика

Удаляем голосование

Результат урока:

Научились отправлять
POST запросы через axios

Узнали, как отправить форму
незаметно для пользователя

Домашнее задание