

## вебка 4.2

```
C: > Users > Компик > Desktop > вебка 5 > <> index.html > ...
1  <!DOCTYPE html>
2  <html lang="ru">
3  <head>
4      <meta charset="UTF-8">
5      <meta name="viewport" content="width=device-width, initial-scale=1.0">
6      <title>Управление постами</title>
7      <link rel="stylesheet" href="style.css">
8  </head>
9  <body>
10     <h1>Управление постами</h1>
11
12     <div class="form-container">
13         <h2>Создать новый пост</h2>
14         <form id="createPostForm">
15             <input type="text" id="title" placeholder="Заголовок" required>
16             <input type="text" id="body" placeholder="Текст поста" required>
17             <button type="submit">Добавить пост</button>
18         </form>
19     </div>
20
21
22     <div id="postsContainer"></div>
23
24     <script src="script.js"></script>
25 </body>
26 </html>
27
```

```
body {
    font-family: Arial, sans-serif;
    margin: 20px;
}

.form-container {
    margin-bottom: 20px;
}

input, button {
    padding: 10px;
    margin: 5px;
}

.post {
    border: 1px solid #ccc;
    padding: 10px;
    margin-bottom: 10px;
}
```

```

document.addEventListener('DOMContentLoaded', () => {
  // Загружаем посты при загрузке страницы
  fetchPosts();
});

// Функция для получения всех постов
function fetchPosts() {
  fetch('https://jsonplaceholder.typicode.com/posts')
    .then(response => {
      if (!response.ok) {
        throw new Error('Сетевая ошибка при получении постов');
      }
      return response.json();
    })
    .then(data => displayPosts(data))
    .catch(error => console.error('Ошибка при получении постов:', error));
}

// Отображение постов на странице
function displayPosts(posts) {
  const postsContainer = document.getElementById('postsContainer');
  postsContainer.innerHTML = ''; // Очищаем контейнер перед повторной загрузкой

  posts.forEach(post => {
    const postElement = document.createElement('div');
    postElement.className = 'post';
    postElement.innerHTML = `
      <h3>${post.title}</h3>
      <p>${post.body}</p>
      <button onclick="updatePost(${post.id})">Обновить</button>
      <button onclick="deletePost(${post.id})">Удалить</button>
    `;
    postsContainer.appendChild(postElement);
  });
}

```

```
// Обработка отправки формы для создания нового поста
document.getElementById('createPostForm').addEventListener('submit', function(event) {
    event.preventDefault(); // Предотвращаем перезагрузку страницы

    const title = document.getElementById('title').value;
    const body = document.getElementById('body').value;

    // Логируем данные, которые будут отправлены
    console.log('Отправка поста:', { title, body });

    fetch('https://jsonplaceholder.typicode.com/posts', {
        method: 'POST',
        headers: {
            'Content-Type': 'application/json'
        },
        body: JSON.stringify({
            title: title,
            body: body,
            userId: 1 // Используем фиктивный userId
        })
    })
    .then(response => {
        if (!response.ok) {
            throw new Error('Ошибка при создании поста');
        }
        return response.json();
    })
    .then(data => {
        console.log('Пост успешно создан:', data);
        alert('Пост успешно создан!');
        fetchPosts(); // Обновляем список постов
    })
    .catch(error => console.error('Ошибка при создании поста:', error));
});
```

```
// Обновление поста
function updatePost(id) {
  const newTitle = prompt('Введите новый заголовок');
  const newBody = prompt('Введите новый текст поста');

  if (newTitle && newBody) {
    fetch(`https://jsonplaceholder.typicode.com/posts/${id}`, {
      method: 'PUT',
      headers: {
        'Content-Type': 'application/json'
      },
      body: JSON.stringify({
        title: newTitle,
        body: newBody,
        userId: 1
      })
    })
    .then(response => {
      if (!response.ok) {
        throw new Error('Ошибка при обновлении поста');
      }
      return response.json();
    })
    .then(data => {
      console.log('Пост обновлен:', data);
      alert('Пост обновлен!');
      fetchPosts(); // Обновляем список постов
    })
    .catch(error => console.error('Ошибка при обновлении поста:', error));
  } else {
    alert('Поле не может быть пустым!');
  }
}
}
```

```
// Удаление поста
function deletePost(id) {
  fetch(`https://jsonplaceholder.typicode.com/posts/${id}`, {
    method: 'DELETE'
  })
  .then(response => {
    if (!response.ok) {
      throw new Error('Ошибка при удалении поста');
    }
    alert('Пост удален!');
    fetchPosts(); // Обновляем список постов
  })
  .catch(error => console.error('Ошибка при удалении поста:', error));
}
}
```

## Управление постами

### Создать новый пост

Вауууу новый пост от кирю

поставьте 5 пж я старался

Добавить пост

**sunt aut facere repellat provident occaecati excepturi optio reprehenderit**  
quia et suscipit suscipit recusandae consequuntur expedita et cum reprehenderit molestiae ut ut quas totam nostrum rerum est autem sunt rem eveniet architecto

Обновить

Удалить

**qui est esse**  
est rerum tempore vitae sequi sint nihil reprehenderit dolor beatae ea dolores neque fugiat blanditiis voluptate porro vel nihil molestiae ut reiciendis qui aperiam non debitis possimus qui neque nisi nulla

Обновить

Удалить

**ea molestias quasi exercitationem repellat qui ipsa sit aut**  
et iusto sed quo iure voluptatem occaecati omnis eligendi aut ad voluptatem doloribus vel accusantium quis pariatur molestiae porro eius odio et labore et velit aut

Обновить

Удалить

**eum et est occaecati**  
ullam et saepe reiciendis voluptatem adipisci sit amet autem assumenda provident rerum culpa quis hic commodi nesciunt rem tenetur doloremque ipsam iure quis sunt voluptatem rerum illo velit

Обновить

Удалить

**nesciunt quas odio**

Уведомление

Пост успешно создан!

Заккрыть

## Управление постами

### Создать новый пост

Вауууу новый пост от кирю

поставьте автомат пж я ста

Добавить пост

**Уведомление**

Пост удален!

Заккрыть

## Управление постами

### Создать новый пост

ыиф

Текст поста

Добавить пост

**Ввод данных**

Введите новый заголовок

Продолжить

Отмена

**sunt aut facere repellat provident occaecati excepturi optio reprehenderit**  
quia et suscipit suscipit recusandae consequuntur expedita et cum reprehenderit molestiae ut ut quas totam nostrum rerum est autem sunt rem eveniet architecto

Обновить

Удалить

**qui est esse**  
est rerum tempore vitae sequi sint nihil reprehenderit dolor beatae ea dolores neque fugiat blanditiis voluptate porro vel nihil molestiae ut reiciendis qui aperiam non debitis possimus qui neque nisi nulla

Обновить

Удалить

**ea molestias quasi exercitationem repellat qui ipsa sit aut**  
et iusto sed quo iure voluptatem occaecati omnis eligendi aut ad voluptatem doloribus vel accusantium quis pariatur molestiae porro eius odio et labore et velit aut

Обновить

Удалить

**eum et est occaecati**  
ullam et saepe reiciendis voluptatem adipisci sit amet autem assumenda provident rerum culpa quis hic commodi nesciunt rem tenetur doloremque ipsam iure quis sunt voluptatem rerum illo velit

Обновить

Удалить

**nesciunt quas odio**

## Уведомление

Поле не может быть пустым!

☐ Больше не показывать сообщения от этого сайта

Закреть

## Ввод данных

Введите новый текст поста

выфвфыв|

☐ Больше не показывать сообщения от этого сайта

Продолжить

Отмена

## Уведомление

Пост обновлен!

Закреть

ehenderit