







Frontender[1.0] JavaScript - fetch, async, await, try, catch. Запросы на сервер. Асинхронная функция

 YouTube	https://youtu.be/5SzUJT0xGXk
 Telegram	https://t.me/Dmitry_Kolotilshikov
 Github	https://github.com/DmitryKolotilshikov/
 Boosty	https://boosty.to/dmitry_ko
# Номер урока	49



Полезные ссылки:

- <https://learn.javascript.ru/fetch>
- <https://jsonplaceholder.typicode.com>
- <https://jsonplaceholder.typicode.com/guide>

```
<!DOCTYPE html>
<html lang="en">

<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <script src="script.js" defer></script>
  <title>fetch, async, await</title>
</head>

<body>
</body>

</html>
```

```
// fetch, async, await, try, catch, finally

fetch('https://jsonplaceholder.typicode.com/todos?_limit=10')
  .then(response => response.json())
```

```

        .then(json => console.log(json))

fetch('https://jsonplaceholder.typicode.com/todos?_limit=5')
    .then(response => response.json())
    .then(json => console.log(json))

// ----- Renders posts -----

fetch('https://jsonplaceholder.typicode.com/posts?_limit=10', {
    method: "GET",
    headers: {
        'Content-type': 'application/json; charset=UTF-8',
    }
})
    .then(response => response.json())
    .then(posts => { console.log(posts); renderPosts(posts) })

const renderPosts = (posts) => {
    posts.forEach(post => {
        const postContainer = document.createElement("div");
        const postTitle = document.createElement("h3");
        const postBody = document.createElement("p");

        postTitle.textContent = post.title;
        postBody.textContent = post.body;
        postContainer.append(postTitle, postBody);

        document.body.append(postContainer);
    })
}

// ----- Renders posts (async & await) -----

// const renderPostsAsync = async () => {
//     const response = await fetch('https://jsonplaceholder.typicode.com/posts?_limit=10');
//     const todos = await response.json();

//     todos.forEach(todo => {
//         const todoContainer = document.createElement("div");
//         const todoTitle = document.createElement("h3");
//         const todoBody = document.createElement("p");

//         todoTitle.textContent = todo.title;
//         todoBody.textContent = todo.body;
//         todoContainer.append(todoTitle, todoBody);

//         document.body.append(todoContainer);
//     })
// }

// renderPostsAsync();

// ----- Try, catch, finally -----

const renderPostsAsync = async () => {
    try {
        console.log("СТАРТ ПРОЦЕССА")
    }
}

```

```
const response = await fetch('https://jsonplaceholder.typicode.com/posts?_limit=10');
const todos = await response.json();

todos.forEach(todo => {
  const todoContainer = document.createElement("div");
  const todoTitle = document.createElement("h3");
  const todoBody = document.createElement("p");

  todoTitle.textContent = todo.title;
  todoBody.textContent = todo.body;
  todoContainer.append(todoTitle, todoBody);

  document.body.append(todoContainer);
})
} catch (error) {
  console.error("ПОЙМАННАЯ ОШИБКА: ", error.message)
} finally {
  console.log("ФИНИШ ПРОЦЕССА")
}
}

renderPostsAsync();
```