

# Դաս 11: Fetch հարցումներ, get, post, put, delete, patch մեթոդներ

HTTP հարցումները հնարավորություն են տալիս կապ հաստատել սերվերի հետ՝ տվյալներ ստանալու, փոխելու, կամ հեռացնելու համար:

HTTP մեթոդները հետևյալն են՝

- **GET** — տվյալներ ստանալու համար:
- **POST** — նոր տվյալներ ավելացնելու համար:
- **PUT** — տվյալները ամբողջությամբ փոխելու համար:
- **DELETE** — տվյալները հեռացնելու համար:
- **PATCH** — տվյալների մասամբ թարմացնելու համար:

**fetch** ֆունկցիան օգտագործվում է **JavaScript**-ում՝ **HTTP** հարցումներ կատարելու համար: Այն վերադարձնում է **Promise**, որը կարելի է մշակել **then** կամ **async/await**-ի միջոցով:

```
fetch('https://jsonplaceholder.typicode.com/posts')
  .then(response => response.json())
// Տվյալների վերաճում JSON-ի
  .then(data => console.log(data))
  .catch(error => console.error('Error:', error));
```

**JSON (JavaScript Object Notation)** տեքստային ձևաչափ է տվյալների փոխանակման համար: Այն հեշտ է հասկանալ թե՛ մարդու, թե՛ համակարգչին: **JSON**-ը հաճախ օգտագործվում է հաճախորդի (օրինակ՝ բրաուզերի) և սերվերի միջև տվյալների փոխանցման համար:

```
{  
  "name": "John",  
  "age": 30,  
  "isAdmin": false  
}
```

Երբ սերվերին հարցում եք ուղարկում **fetch**-ի միջոցով, սերվերն հաճախ պատասխանում է **JSON** ձևաչափով տվյալներով: Այդ տվյալները հաճախ տեքստային տեսքով են (**string**), ուստի հարկավոր է դրանք վերածել **JavaScript**-ի օբյեկտի, որպեսզի դրանց հետ աշխատեք:

```
fetch('https://jsonplaceholder.typicode.com/posts')  
  .then(response => response.text())  
// Պատասխանի տեսքումը որպես string  
  .then(text => {  
    console.log('Տեքստային տվյալներ (մինչև JSON  
վերածելը):', text); // Տեքստի ցուցադրում  
    const jsonData = JSON.parse(text);  
// Տվյալների վերածում JSON-ի  
    console.log('JSON տվյալներ (վերածված):', jsonData);  
// JSON-ի ցուցադրում  
  })  
  .catch(error => console.error('Միայն:', error));
```

**response.text()** - Վերցնում է պատասխանն իր հում տեքստային ձևաչափով: Այսինքն՝ այն հենց այն տեքստն է, որը սերվերն ուղարկել է:

**JSON.parse(text)** - Տեքստը (**string**) վերցնում է և վերածում **JavaScript**-ի օբյեկտի:

**GET** հարցումներով մենք սերվերից ստանում ենք տվյալներ: Այս մեթոդը չի փոփոխում սերվերի տվյալները:

```

async function getPosts() {
  try {
    const response = await
fetch("https://jsonplaceholder.typicode.com/posts");
    const posts = await response.json();
    console.log(posts);
  } catch (error) {
    console.error("Error fetching posts:", error);
  }
}

getPosts();

```

**POST** մեթոդը օգտագործվում է նոր տվյալներ ավելացնելու համար:

## Օրինակ՝ ստեղծել նոր գրառում.

```

async function createPost() {
  try {
    const response = await
fetch('https://jsonplaceholder.typicode.com/posts', {
      method: 'POST',
      headers: {
        'Content-Type': 'application/json',
      },
      body: JSON.stringify({
        title: 'Նոր վերնագիր',
        body: 'Գրառման պարունակություն',
        userId: 1,
      }),
    });
    const newPost = await response.json();

```

```

    console.log('Created post:', newPost);
  } catch (error) {
    console.error('Error creating post:', error);
  }
}

createPost();

```

**method: 'POST'** - Նշում է, որ հարցումը պետք է օգտագործի **HTTP POST** մեթոդը, որն օգտագործվում է սերվերին նոր տվյալներ փոխանցելու համար:

**Headers** - Ասում է սերվերին, որ փոխանցվող տվյալների **Ֆորմատը JSON** է: **Content-Type: application/json** գլխամասը պարտադիր է, եթե մենք տվյալներ եք ուղարկում **JSON** **Ֆորմատով**:

**Body** - ուղարկում է **JSON** տվյալներ որպես հարցման բովանդակություն (**body**):

**JSON.stringify** մեթոդը **JavaScript**-ի օբյեկտը վերածում է **JSON** տեքստի, որը սերվերը կարող է հասկանալ:

**PUT** մեթոդը սերվերի տվյալները ամբողջությամբ փոխարինելու համար է:

```

async function updatePost(postId) {
  try {
    const response = await fetch(
      `https://jsonplaceholder.typicode.com/posts/${postId}`,
      {
        method: "PUT",
        headers: {
          "Content-Type": "application/json",
        },
        body: JSON.stringify({

```

```

        id: postId,
        title: "Փոփոխված վերնագիր",
        body: "Փոփոխված պարունակություն",
        userId: 1,
    )),
    }
);
const updatedPost = await response.json();
console.log("Updated post:", updatedPost);
} catch (error) {
    console.error("Error updating post:", error);
}
}

updatePost(1);

```

**DELETE** մեթոդը օգտագործվում է տվյալներ հեռացնելու համար:

```

async function deletePost(postId) {
    try {
        const response = await fetch(
`https://jsonplaceholder.typicode.com/posts/${postId}`,
        {
            method: "DELETE",
        }
        );
        if (response.ok) {
            console.log(`Post ${postId} was deleted.`);
        }
    } catch (error) {
        console.error("Error deleting post:", error);
    }
}

```

```
    }  
  }  
  
deletePost(1);
```

**PATCH** մեթոդը տվյալները մասամբ թարմացնելու համար է:

```
async function patchPost(postId) {  
  try {  
    const response = await fetch(  
      `https://jsonplaceholder.typicode.com/posts/${postId}`,  
      {  
        method: "PATCH",  
        headers: {  
          "Content-Type": "application/json",  
        },  
        body: JSON.stringify({  
          title: "Թարմացված վերնագիր",  
        }),  
      }  
    );  
    const patchedPost = await response.json();  
    console.log("Patched post:", patchedPost);  
  } catch (error) {  
    console.error("Error patching post:", error);  
  }  
}  
  
patchPost(1);
```