

Работа с запросами в JavaScript и теория HTTP

Кормышев Егор ИСиП-301

21 марта 2024 г.

Определение HTTP запроса

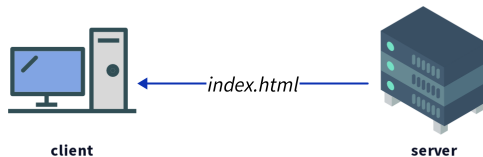
HTTP-запрос - Обмен данными между клиентом и сервером

Виды HTTP-сообщений

Запрос (Request)



Ответ (Response)



HTTP-запрос состоит из нескольких частей:

- Строка запроса - `https://example.com`
- Заголовки (headers) - {
 Accept: application/json
 User-Agent: Mozilla/5.0
 Host: example.com
 ...
}
- Тело запроса (body) - данные, передаваемые серверу при запросе в формате json/xml/text

HTTP-ответ состоит из нескольких частей:

- Строка ответа - HTTP/1.1 200 OK
- Заголовки (headers) - {
Accept: application/json
User-Agent: Mozilla/5.0
Host: example.com
...
}
- Тело запроса (body) - данные, передаваемые от сервера в формате json/html/text

Все коды делятся на 5 групп:

- 1.x.x - Информация
- 2.x.x - Успех
- 3.x.x - Перенаправление
- 4.x.x - Клиентская ошибка
- 5.x.x - Серверная ошибка

Подробнее в [таблице](#)

Способы выполнения HTTP запроса в javascript:

- Стандартная функция `fetch()`
- Библиотека `axios`

Использование функции fetch()

Пример запроса в js

```
fetch(url)
  .then(promise => {
    return promise.json()
  })
  .then(data => {
    console.log(data)
  })
  .catch(err => {
    console.log(err)
  })
  .catch(err => {
    console.log(err)
  })
```

Структура функции fetch:

- ❶ `fetch(url)` - запрос
- ❷ `then(res)` - результат запроса
- ❸ `catch(err)` - объект ошибки

Типы в fetch

```
export interface Root {  
  id: number  
  name: string  
  username: string  
  email: string  
  address: Address  
  phone: string  
  website: string  
  company: Company  
}  
{...}
```

Типы в fetch II

```
let url = "https://jsonplaceholder.typicode.com/users";

fetch(url)
  .then((promise: Response) => {
    return promise.json() as Promise<User[]>;
  })
  .then((data: User[]) => {
    console.log(data);
  })
  .catch((err: Error) => {
    console.log(err);
  });
```