What are HTTP Requests

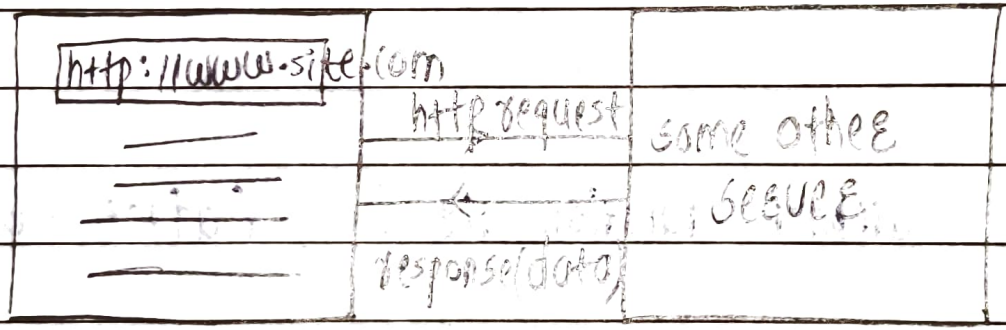
- Make HTTP requests to get data from another server.

- we make requests to API endpoints.

- API endpoints

example API endpoint

http://www.musicapi.com/artist/moby



- we are using this api

"https://jsonplaceholder.typicode.com"

Making HTTP Requests (XHR)

- XMLHttpRequest.readyState

Value	→ state	→ Description
0	→ UNSENT	→ client has been created. open() not called yet.
1	→ OPENED	→ open() has been called.
2	→ HEADERS_RECEIVED	→ send() has been called, and headers and status are available.
3	→ LOADING	→ Downloading; responseText holds partial data.
4	→ DONE	→ The operation is complete.

```
- const request = new XMLHttpRequest();
request.addEventListener('readystatechange', () => {
    console.log(request, request.readyState);
});
request.open('GET', 'https://jsonplaceholder.typicode.com/todos/');
request.send();
```

↓

XMLHttpRequest {onready —11— }
1
XMLHttpRequest {onready —11— }
2
XMLHttpRequest {onready —11— }
3
XMLHttpRequest {onready —11— }
4

```
const request = new XMLHttpRequest();
request.addEventListener('readystatechange', () => {
    if(request.readyState === 4){
        console.log(request.responseText);
    };
});
request.open('GET', 'https://jsonplaceholder.typicode.com/todos/');
request.send();
```

↓

```
[
  {
    "userId": 1,
    "id": 1,
    ---//---
    ---//---
    ---//---
  }
    ---//---
]
```

Page No.
Date

Response status

- Information responses
100 - continue
101 - switching protocol
102 - processing (webDAV)
103 - Eaely hints

- Successful Responses
200 - ok
201 - created
202 - accepted
203 - Non authaitative information
204 - No content
205 - reset content
206 - partial content

- Redirection messages
300 - multiple choice
301 - moved permanently
302 - found
303 - see other
304 - not modified
305 - use proxy
306 - unused
307 - temporary Redirect
308 - permanent redirect

- client error responses
400 - bad request
401 - unauthorized
402 - payment required
403 - forbidden
404 - not found
405 - method not allowed
406 - not acceptable
407 - proxy authentication required
408 - request timeout
409 - conflict

- server error responses
500 - internal server error
501 - not implemented
502 - bad gateway
503 - server unavailable
504 - gateway timeout

```
const request = new XMLHttpRequest();
request.addEventListener('readystatechange', () => {
    if (request.readyState === 4) {
        console.log(request, request.responseText);
    };
});
request.open('GET', 'https://jsonplaceholder.typicode.com/todoss/');
request.send();
```

Due to this

→

⊗ GET https://jsonplaceholder.typicode.com/todoss/ 404

▶ XMLHttpRequest {onreadystatechange —11— }

→ Why 404 error because that url does not exist or it do not found therefore we do not see any responseText.

```
const request = new XMLHttpRequest();
request.addEventListener('readystatechange', () => {
    if(request.readyState === 4 && request.status === 200){
        console.log(request,request.responseText);
    }else if(request.readyState === 4) {
        console.log('could not fetch data');
    }
});
request.open('GET','https://jsonplaceholder.typicode.com/todoss/');
request.send();
```

↓

due to this

⊗ GET https://jsonplaceholder.typicode.com/todoss/ 404
could not fetch data

↓

Error because our URL is wrong or URL not found.

Page No.
Date

callback functions

```
const getTodos = (callback) => {
    const request = new XMLHttpRequest();

    request.addEventListenee('readystatechange', () => {
        if(request.readystate === 4 && request.status === 200){
            callback(undefined, request.responseText);
        } else if(request.readystate === 4){
            callback('could not fetch data', undefined);
        }
    });
    request.open('GET', 'https://jsonplaceholder.typicode.com/todos/');
    request.send();
};
getTodos((eee, data) => {
    console.log('callback fieed');
    console.log(eee, data);
});
```

↓

callback fieed
undefined "[
{
"useeId": 1,
—11—
},
—11—
]"

(Note that oue uel is roeeect heee)

```
const getTodos = (callback) => {

    const request = new XMLHttpRequest();

    request.addEventListener('readystatechange', () => {
        if(request.readyState === 4 && request.status === 200){
            callback(undefined, request.responseText);
        }else if(request.readyState === 4){
            callback('could not fetch data', undefined);
        }
    });
    request.open('GET', 'https://jsonplaceholder.typicode.com/todoss/1');
    request.send();
};
getTodos((err, data) => {
    console.log('callback fired');
    console.log(err, data);
});
```

↓

GET https://jsonplaceholder.typicode.com/todoss/ 404
callback fired
could not fetch data undefined

(This is because our url is wrong)

```
const getTodos = (callback) => {
    const request = new XMLHttpRequest();

    request.addEventListener('readystatechange', () => {
        if(request.readyState === 4 && request.status === 200){
            callback(undefined, request.responseText);
        }else if(request.readyState === 4){
            callback('could not fetch data', undefined);
        }
    });
    request.open('GET', 'https://jsonplaceholder.typicode.com/todos/');
    request.send();
};
getTodos((err, data) => {
    console.log('callback fired');
    if(err){
        console.log(err);
    }else{
        console.log(data);
    }
});
```

↓

callback fired

```
[
    {
        "userId": 1,
        --11--
    },
    --11--
]
```

(Note that here url is correct)

And similarly when our url is wrong then we get output as

↓

GET https://jsonplaceholder.typicode.com/todoss 404
callback fired
could not fetch data

(Note that we get error because our url is wrong)

```
const getTodos = (callback) => {
    const request = new XMLHttpRequest();
    request.addEventListener('readystatechange', () => {
      if(request.readyState === 4 && request.status === 200){
        callback(undefined, request.responseText);
      }else if(request.readyState === 4){
        callback('could not fetch data', undefined);
      }
    });
    request.open('GET', 'https://jsonplaceholder.typicode.com/todos/');
    request.send();
};

console.log(1);
console.log(2);
```

```
getTodos((eee, data) => {
    console.log('callback fired');
    if (eee) {
        console.log(eee);
    } else {
        console.log(data);
    }
});

console.log(3);
console.log(4);
```

(This is non-blocking code)
(This is asynchronous code)

↓

```
1
2
3
4
callback fired
[
  {
    "userId": 1,
      ==//==
  },
    ==//==
]
```