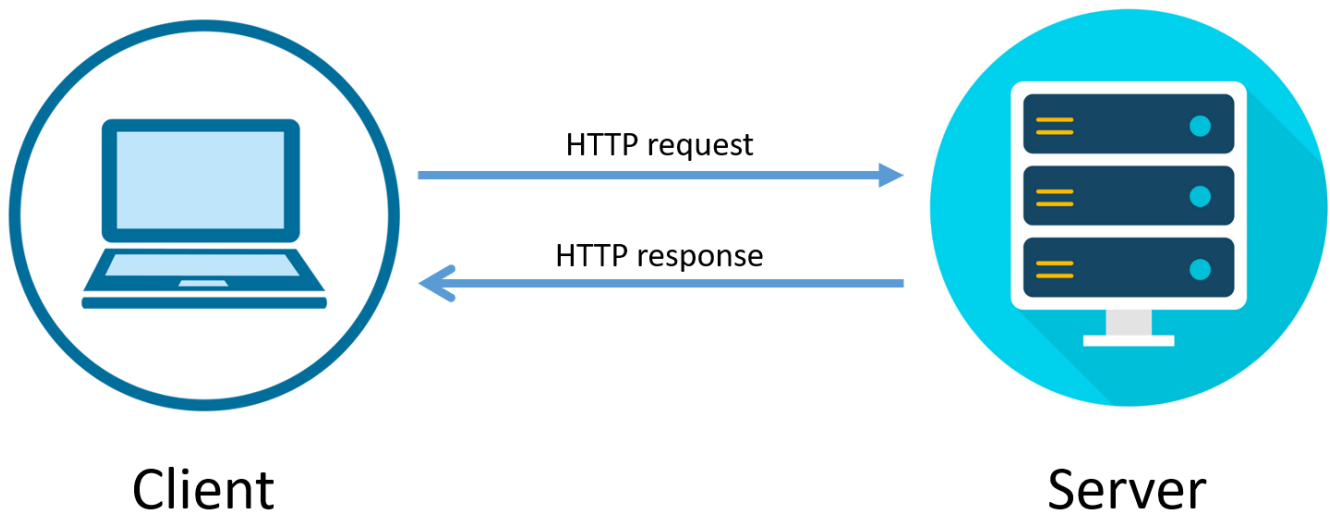# Fetch API

AGENDA

1. fetch overview
2. fetch with headers
3. Handle errors

1. fetch overview

- The Fetch API provides an interface for fetching resources source
- Web APIs -> global method **fetch()**.

fetch

- it starts the process of fetching a resource from the network
- returns a promise which fulfilled once the response is available
- it only rejects when a network error is encountered.
- it doesn't reject on HTTP errors (4xx, 5xx)
- you can check **Response.ok** to handle error properly.

```javascript
const url = 'https://js-post-api.herokuapp.com/api/students?_page=1';
const init = {
  method: 'POST', // GET, PUT, PATCH. DELETE
  headers: {
    'Content-Type': 'application/json',
    Authorization: 'Bearer YOUR_TOKEN_HERE'
  },
  body: JSON.stringify({ name: 'Easy Frontend' }),
}
```

```
const promise = fetch(url, init);
```

Init object

#	Name	Desc
1	method	GET, POST, PUT, PATCH, DELETE, ...
2	headers	'Content-Type', Authorization, x-*
3	body	URLSearchParams, FormData, Blob, ...
4	signal	an AbortSignal used to abort if desired read more

Common Content-Type

Content-Type	Desc
application/json	JSON data (most of the time)
multipart/form-data	FormData (upload file: images, pdf, ...)
application/x-www-form-urlencoded	key-value on URL

Source: https://developer.mozilla.org/en-US/docs/Web/API/fetch#parameters

```javascript
fetch('https://js-post-api.herokuapp.com/api/students?_page=1')
  .then(response => response.json())
  .then(data => console.log(data));
```

2. fetch with headers / body

- GET, DELETE: mình không dùng body.
- POST/PUT/PATCH: mình có gửi kèm body.

```javascript
// Add new student
fetch('https://js-post-api.herokuapp.com/api/students', {
  method: 'POST',
  headers: {
    'Content-Type': 'application/json',
  },
  body: JSON.stringify({
    name: 'Easy Frontend',
    age: 18,
    mark: 9,
    gender: 'male',
  })
}).then(...).catch(...)
```

```javascript
// Update partial info of student
fetch('https://js-post-api.herokuapp.com/api/students/e0df2354-1014-4f40-
97c4-76e1dac88ab5', {
  method: 'PATCH',
  headers: {
    'Content-Type': 'application/json',
  },
  body: JSON.stringify({
    name: 'Easy',
    age: 20,
  })
}).then(...).catch(...)
```

```javascript
// Delete a student
fetch('https://js-post-api.herokuapp.com/api/students/e0df2354-1014-4f40-
97c4-76e1dac88ab5', {
  method: 'DELETE',
  headers: {
    'Content-Type': 'application/json',
  },
}).then(...).catch(...)
```

Tham khảo: https://developer.mozilla.org/en-US/docs/Web/HTTP/Headers

3. Handle errors

```javascript
fetch('https://js-post-api.herokuapp.com/api/invalid-endpoint', {
  method: 'GET',
  headers: {
    'Content-Type': 'application/json',
  },
})
  .then(response => {
    if (response.ok) return response.json();

    // TODO: How you handle errors here? --> It depends on what your API
returns
    // Solution 1: return Promise.reject(new Error('Something wrong!'));
    // Solution 2: throw new Error(response.statusText);
    return response.json().then(data => {
      throw new Error(data?.message || 'Something went wrong!');
    })
  })
  .catch(error => {
    console.log(error);
    // Toast message
    // Send report to log server (Sentry)
  })
```

Response status codes

Status Code	Name
1xx	Informational responses
2xx	Successful responses
3xx	Redirects
4xx	Client errors
5xx	Server errors

Common status code

Status code	Name
200	OK
201	Created
301	Moved Permanently
400	Bad Request
401	Unauthorized
403	Forbidden
404	Not Found
429	Too Many Requests
500	Internal Server Error
502	Bad Gateway

Source: https://developer.mozilla.org/en-US/docs/Web/HTTP/Status

What to do when you get an error from API

- Check URL
- Check payload
- Check headers
- Then come to ask BE with the above infomation.

Tham khảo

- https://developer.mozilla.org/en-US/docs/Web/API/Fetch_API/Using_Fetch
- https://developer.mozilla.org/en-US/docs/Web/API/fetch#parameters
- https://developer.mozilla.org/en-US/docs/Web/API/Fetch_API
- https://developer.mozilla.org/en-US/docs/Web/HTTP/Basics_of_HTTP/MIME_types/Common_types

---

Khoá học Javascript cho người mới bắt đầu 2021🎉

- Tác giả: **Hậu Nguyễn** - Founder Easy Frontend
- Khoá học chỉ được published trên Udemy, không thông qua trung gian.
- Khoá học không bán dạng videos upload trên Google Drive hay bất cứ hình thức nào tương tự.
- Khoá học có nhóm discord để hỗ trợ trong quá trình học tập.

☎️ Liên hệ tác giả để được hỗ trợ:

- ✅ Facebook: https://www.facebook.com/nvhauesmn/
- ✅ Fanpage: https://www.facebook.com/learn.easyfrontend
- ✅ Youtube Channel: https://www.youtube.com/easyfrontend