

20215118

phuc.dth215118@sis.hust.edu.vn

1	2	3	4	5
6	7	8	9	10
11	12	13	14	15
16	17	18	19	20
21	22	23	24	25
26	27	28	29	30
31				

Fetch API


Câu hỏi #2ec174

1 điểm (không tích lũy, không hiển thị kết quả)

Trong Fetch API, phương thức `.json()` trả về đối tượng kiểu gì?

- ☐ JSON
- ☐ Text
- ☐ String
- ☒ Promise

Gửi

 Câu trả lời đã được gửi.


Câu hỏi #5caed9

1 điểm (không tích lũy, không hiển thị kết quả)

Trong Fetch API, khi gửi yêu cầu HTTP đến server, những công việc nào cần thực hiện để gửi dữ liệu dưới dạng JSON?

- ☒ Đặt method: 'POST' và chuyển đối tượng JSON thành `string` trong body.
- ☐ Thêm header `Content-Type: xml/json`.
- ☒ Thêm header `Content-Type: application/json`.
- ☐ Thực hiện phương thức `fetchPost()`.

Gửi

 Câu trả lời đã được gửi.


Câu hỏi #693fcd

1 điểm (không tích lũy, không hiển thị kết quả)

Để chuyển đổi dữ liệu phản hồi từ Fetch API thành đối tượng JSON, cần gọi phương thức nào?

- ☐ `response.toJson()`
- ☐ `response.parseJSON()`
- ☐ `response.parseTextToJson()`
- ☒ `response.json()`

Gửi

 Câu trả lời đã được gửi.


Câu hỏi #5a9d65

1 điểm (không tích lũy, không hiển thị kết quả)

Khi sử dụng Fetch API, nếu muốn đảm bảo đoạn mã được thực thi khi yêu cầu thành công hay thất bại, bạn cần sử dụng phương thức nào?

- ☒ `.then()`
- ☒ `.catch()`
- ☒ `.finally()`
- ☐ `.response()`

Gửi

 Câu trả lời đã được gửi.


Câu hỏi #887df7

1 điểm (không tích lũy, không hiển thị kết quả)

Trong Fetch API, để gửi yêu cầu và nhận phản hồi dưới dạng văn bản (text), có thể sử dụng phương thức nào?

- ☒ `response.text()`
- ☐ `response.xml()`
- ☐ `response.json()`
- ☐ `response.body()`

Gửi

 Câu trả lời đã được gửi.

Câu hỏi #bac78d

1 điểm (không tích lũy, không hiển thị kết quả)

Fetch API có sự khác biệt gì so với `XMLHttpRequest` trong việc sử dụng AJAX?

- ☐ Fetch API yêu cầu sử dụng callback để xử lý dữ liệu, trong khi `XMLHttpRequest` sử dụng `Promise`.
- ☒ Fetch API hỗ trợ `Promise`, giúp mã dễ đọc và xử lý bất đồng bộ một cách trực quan hơn, trong khi `XMLHttpRequest` sử dụng callback.

- ☐ Fetch API không hỗ trợ hết các phương thức HTTP, trong khi `XMLHttpRequest` hỗ trợ hết.
- ☐ `XMLHttpRequest` không thể xử lý đồng bộ, còn Fetch API có thể.

Gửi

i Câu trả lời đã được gửi.

Câu hỏi #182350

1 điểm (không tích lũy, không hiển thị kết quả)

Fetch API trả về một đối tượng gì sau khi gọi thành công?

- ☐ Response
- ☐ JSON
- ☒ Promise
- ☐ XMLHttpRequest

Gửi

i Câu trả lời đã được gửi.

Câu hỏi #1b224a

1 điểm (không tích lũy, không hiển thị kết quả)

Trong việc thực hiện các yêu cầu AJAX, `XMLHttpRequest` có nhược điểm gì so với Fetch API?

- ☐ `XMLHttpRequest` không hỗ trợ hết các phương thức HTTP, trong khi Fetch API hỗ trợ hết.
- ☐ `XMLHttpRequest` chỉ hỗ trợ các yêu cầu không đồng bộ, trong khi Fetch API chỉ hỗ trợ các yêu cầu đồng bộ.
- ☐ `XMLHttpRequest` không hỗ trợ xử lý lỗi, trong khi Fetch API tự động xử lý lỗi.
- ☒ `XMLHttpRequest` có cấu trúc mã phức tạp hơn và phải sử dụng nhiều callback, trong khi Fetch API sử dụng `Promise` giúp dễ dàng quản lý các yêu cầu bất đồng bộ.

Gửi

i Câu trả lời đã được gửi.

Câu hỏi #12869e

1 điểm (không tích lũy, không hiển thị kết quả)

Nhược điểm lớn nhất của Fetch API khi so sánh với `XMLHttpRequest` trong việc xử lý lỗi là gì?

- ☐ Fetch API không hỗ trợ phát hiện lỗi JSON, trong khi `XMLHttpRequest` làm được điều này.
- ☐ Fetch API không hỗ trợ xử lý lỗi đồng bộ, trong khi `XMLHttpRequest` hỗ trợ.
- ☐ Fetch API không thể gửi lỗi về server, trong khi `XMLHttpRequest` có thể.
- ☒ Fetch API không tự động `throw` lỗi khi nhận mã trạng thái HTTP không thành công (ví dụ: 404 hoặc 500), trong khi `XMLHttpRequest` làm vậy.