

 Cập nhật tháng 8 năm 2024

[Bài Đọc] Session storage

Session Storage là một trong các API lưu trữ của trình duyệt, cho phép lưu trữ dữ liệu tạm thời trong một phiên làm việc. Dữ liệu này tồn tại cho đến khi tab hoặc cửa sổ trình duyệt bị đóng

1. Đặc điểm của Session Storage

- Thời gian sống:
 - Dữ liệu chỉ tồn tại trong một phiên làm việc (session)
 - Khi tab hoặc cửa sổ được đóng, dữ liệu sẽ bị xóa
- Dung lượng:
 - Tương tự Local Storage, Session Storage hỗ trợ lưu trữ khoảng 5 - 10 MB dữ liệu
- Truy cập:
 - Dữ liệu chỉ có thể truy cập từ tab nơi dữ liệu được lưu trữ. Không thể chia sẻ giữa các tab hoặc cửa sổ khác nhau, ngay cả khi chúng cùng domain
- Định dạng lưu trữ:
 - Chỉ lưu trữ chuỗi (string). Các kiểu dữ liệu khác (như đối tượng hoặc mảng) cần được chuyển đổi thành JSON trước khi lưu
- Bảo mật:
 - Không an toàn trước các cuộc tấn công XSS (Cross-Site Scripting). Không nên lưu trữ dữ liệu nhạy cảm

2. API của Session Storage

- Lưu dữ liệu
 - Sử dụng phương thức `setItem` để lưu dữ liệu vào Session Storage

```
sessionStorage.setItem(key: 'key', value: 'value');
```

- Ví dụ

```
sessionStorage.setItem(key: 'username', value: 'John Doe');  
sessionStorage.setItem(key: 'age', value: '25');
```

- Lấy dữ liệu

- Sử dụng phương thức **getItem** để lấy dữ liệu từ Session Storage

```
sessionStorage.getItem(key: 'key');
```

- Ví dụ

```
const username = sessionStorage.getItem(key: 'username');  
console.log(...data: username); // Output: John Doe
```

- Xóa dữ liệu

- Xóa một mục cụ thể bằng phương thức **removeItem**

```
sessionStorage.removeItem(key: 'key');
```

- Ví dụ

```
sessionStorage.removeItem(key: 'username');
```

3. Lưu trữ dữ liệu phức tạp

- Lưu đối tượng

- Chuyển đổi đối tượng thành chuỗi JSON trước khi lưu

```
const user = { name: 'John', age: 25 };  
sessionStorage.setItem(key: 'user', value: JSON.stringify(value: user));
```

- Lấy lại đối tượng bằng cách phân tích chuỗi JSON

```
const user = JSON.parse(text: sessionStorage.getItem(key: 'user'));  
console.log(...data: user.name); // Output: John
```

- Lưu mảng

- Chuyển đổi mảng thành chuỗi JSON trước khi lưu

```
const todos = ['Learn JavaScript', 'Build a project'];  
sessionStorage.setItem(key: 'todos', value: JSON.stringify(value: todos));
```

- Lấy lại mảng

```
const todos = JSON.parse(text: sessionStorage.getItem(key: 'todos'));  
console.log(...data: todos); // Output: ['Learn JavaScript', 'Build a project']
```

4. Trường hợp sử dụng Session Storage

- Lưu trạng thái trong phiên làm việc:

- Ví dụ: Lưu thông tin giỏ hàng, danh sách sản phẩm đã xem trong phiên làm việc hiện tại
- Lưu dữ liệu biểu mẫu tạm thời:
 - Lưu dữ liệu người dùng nhập vào biểu mẫu để phục hồi nếu tab bị làm mới
- Lưu trạng thái ứng dụng:
 - Trạng thái menu, giao diện hoặc bộ lọc trong phiên làm việc
- Lưu dữ liệu giữa các trang trong một phiên:
 - Khi người dùng chuyển giữa các trang, Session Storage có thể giúp lưu trữ dữ liệu cần thiết

5. So sánh Session Storage và Local Storage

Đặc điểm	Session storage	Local storage
Thời gian sống	Trong phiên làm việc hiện tại	Lâu dài (cho đến khi bị xóa)
Dung lượng	~5-10 MB	~5-10 MB
Truy cập	Chỉ trong tab hiện tại	Truy cập từ mọi tab cùng domain
Ứng dụng	Lưu trữ dữ liệu tạm thời	Lưu trữ dữ liệu lâu dài

6. Tài liệu tham khảo

<https://www.telerik.com/blogs/three-browser-storage-mechanismsses>

Danh sách các bài học

