Storage: LocalStorage & Cookie

> Type-safe wrappers for browser storage mechanisms: `localStorage`, `cookie`, and more.

This module provides **reusable, consistent, and safe functions** to work with client-side storage.
Currently, it supports `LocalStorage` and `Cookie`, but it can be extended to other storage mechanisms (e.g., sessionStorage, IndexedDB) in the future.

Contents:

---

1. LocalStorage

File Structure

There are two main files for configuration:

- `functions.ts` — contains functions to work with LocalStorage.
- `paths.ts` — defines keys (paths) used for storage.

Usage

Paths Example

```
const LOCAL_STORAGE_PATHS = {
  user: 'user',
  is_logged_in: 'is_logged_in',
  score: 'score',
} as const;

export default LOCAL_STORAGE_PATHS;
```

GET Method

```
import { LOCAL_STORAGE_PATHS, customLocalStorage } from '@services/storage';

const user = customLocalStorage.get(LOCAL_STORAGE_PATHS.user, { id: 0, name: 'Guest' });
console.log(user); // { id: 1, name: 'Alice' }

const isLoggedIn = customLocalStorage.get(LOCAL_STORAGE_PATHS.is_logged_in, false);
console.log(isLoggedIn); // true

const score = customLocalStorage.get(LOCAL_STORAGE_PATHS.score, 0n);
console.log(score); // 123n
```

SET Method

```
import { LOCAL_STORAGE_PATHS, customLocalStorage } from '@services/storage';

customLocalStorage.set(LOCAL_STORAGE_PATHS.user, { id: 1, name: 'Alice' });

customLocalStorage.set(LOCAL_STORAGE_PATHS.is_logged_in, true);

customLocalStorage.set(LOCAL_STORAGE_PATHS.score, 123n);
```

REMOVE Method

```
import { LOCAL_STORAGE_PATHS, customLocalStorage } from '@services/storage';

customLocalStorage.remove(LOCAL_STORAGE_PATHS.user);

const removedUser = customLocalStorage.get(LOCAL_STORAGE_PATHS.user, null);
console.log(removedUser); // null
```

---

2. Cookie

File Structure

There are two main files for configuration:

- `functions.ts` — contains functions to work with Cookie.
- `paths.ts` — defines keys (paths) used for storage.

Usage

Paths Example

```
const COOKIE_PATHS = {
  user: 'user',
  is_logged_in: 'is_logged_in',
  session_token: 'session_token',
} as const;

export default COOKIE_PATHS;
```

GET Method

```
import { COOKIE_PATHS, customCookieStorage } from '@services/storage';

const user = customCookieStorage.get(COOKIE_PATHS.user, { id: 0, name: 'Guest' });
console.log(user); // { id: 1, name: 'Alice' }

const isLoggedIn = customCookieStorage.get(COOKIE_PATHS.is_logged_in, false);
console.log(isLoggedIn); // true
```

SET Method

```
import { COOKIE_PATHS, customCookieStorage } from '@services/storage';

customCookieStorage.set(COOKIE_PATHS.user, { id: 1, name: 'Alice' });
customCookieStorage.set(COOKIE_PATHS.is_logged_in, true);
customCookieStorage.set(COOKIE_PATHS.session_token, 'abc123', { path: '/', maxAge: 3600 });
```

REMOVE Method

```
import { COOKIE_PATHS, customCookieStorage } from '@services/storage';

customCookieStorage.remove(COOKIE_PATHS.user);

const removedUser = customCookieStorage.get(COOKIE_PATHS.user, null);
console.log(removedUser); // null
```
