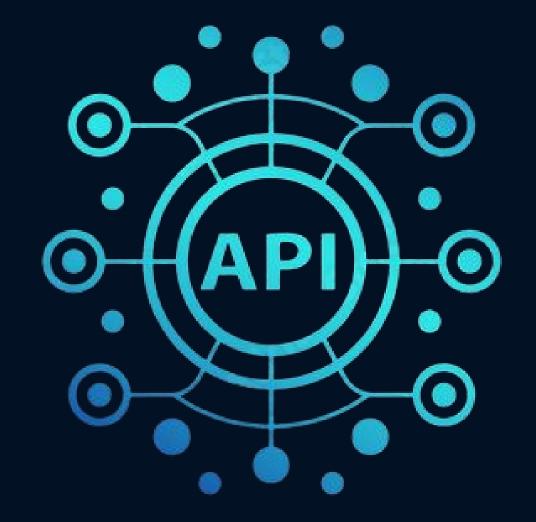


# 5 USE FULL WAR STATES AND STATES



**Presented by Sewak Dhakal** 

# 1. LOCALHOST API (DEV SERVER)

This isn't technically a Web API, but let's clear it up.

Localhost is your local machine acting as a server. Used for testing APIs or websites before deploying.

#### 1. SAVING DATA

```
// Save a string
localStorage.setItem("username", "Madara");

// Save a number (it gets saved as string)
localStorage.setItem("score", 100);

// Save an object (must stringify!)
const user = {
   name: "Madara",
   level: "Legendary",
   role: "Web Dev"
};
localStorage.setItem("userInfo", JSON.stringify(user));
```

### **2. RETRIEVING DATA**

```
// Get string
const username = localStorage.getItem("username");
console.log("Username:", username); // Madara
// Get number (but it's string, so convert if needed)
const score = parseInt(localStorage.getItem("score"));
console.log("Score:", score); // 100
// Get object (parse JSON string)
const userInfo = JSON.parse(localStorage.getItem("userInfo"));
console.log("User Info:", userInfo);
// { name: "Madara", level: "Legendary", role: "Web Dev" }
```

## **⋖ 3. REMOVING DATA**

```
// Remove a specific item
localStorage.removeItem("score");
// Remove the whole storage
localStorage.clear(); // A All keys gone
```

#### **BONUS: TEST IT IN BROWSER** YOU CAN OPEN DEVTOOLS > CONSOLE AND PASTE THIS:

```
localStorage.setItem("anime", "One Piece");
console.log(localStorage.getItem("anime")); // One Piece
localStorage.removeItem("anime");
console.log(localStorage.getItem("anime")); // null
```



#### It lets you:

- Add new entries to browser history
- Update the URL without reloading the page
- Detect when the user navigates back or forward.



# **CORE METHODS**

Method	Description
history.pushState()	Push a new state to browser history
history.replaceState()	Replace current state
window.onpopstate	Handle back/forward button
history.back()	Go one step back
history.forward()	Go one step forward

# **EXAMPLE**

```
index.js

//go back in history
window.history.back();

//go forward in history
window.history.forward();
```



Used to make HTTP requests (GET, POST, etc.) in modern JavaScript.

# Example: GET Data

```
async function fetchData() {
  try {
    const response = await fetch('add Url');
    const data = await response.json();
    console.log(data);
  } catch (error) {
    console.error('Error:', error);
fetchData();
```

# Used to copy or read text from the clipboard.

```
// Copy text
async function copyText() {
  await navigator.clipboard.writeText("Text to copy");
// Read text
async function readText() {
  const text = await navigator.clipboard.readText();
 console.log(text);
```