EXPLORING JAVASCRIPT'S POWERHOUSE: AbortController

- Improved User Experience
- Resource Management
- Cleaner Code:

Decryptus_

Dominic Orefuwa

Dominic Orefuwa

Swipe →

What is AbortController?

AbortController is a vital feature in modern JavaScript, allowing you to manage and gracefully handle asynchronous operations such as **HTTP** requests. They give you the power to **cancel** these operations when they are no longer needed, improving resource management and user experience.

Swipe →

Key Benefits of AbortControllers in JavaScript

Improved User Experience:

AbortControllers allow you to enhance user experience by providing a mechanism to cancel asynchronous operations, such as **HTTP** requests, that may take longer to complete than expected. This ensures that your web applications remain responsive and user-friendly.

✨ Imagine a scenario where a user initiates a search on your website but decides to change their query or navigate away from the page before the search request completes. With **AbortControllers**, you can quickly **cancel** the initial request, preventing unnecessary data retrieval and presenting the user with more relevant and up-to-date information.

Swipe →

 Key Benefits of AbortControllers in JavaScript

 Resource Management:

Efficient resource management is critical for both server-side and client-side applications. **AbortControllers** empower you to make better use of network resources by allowing you to cancel requests that are no longer needed. This not only reduces the load on your server but also conserves precious bandwidth.

✦ Consider a situation where your web application sends multiple concurrent requests to a server. Some of these requests might become redundant due to user actions or changing requirements. **AbortControllers** enable you to swiftly terminate these requests, ensuring that your application operates efficiently and minimizes unnecessary data transfer.

Swipe →

Key Benefits of AbortControllers in JavaScript

Cleaner Code:

Clean and **maintainable** code is a goal for every developer. **AbortControllers** contribute to code cleanliness by providing a structured way to handle asynchronous operations. By incorporating them into your codebase, you can streamline the logic for managing and canceling requests, leading to more organized and readable code.

✦ In complex applications, managing asynchronous tasks without **AbortControllers** can lead to spaghetti code and error-prone handling of request cancellation. With **AbortControllers**, you can centralize the logic for managing requests and cancellations, making your codebase more maintainable and easier to debug.

Swipe →

Example Use Case: On a search Input

```
<script>
  const searchInput = document.getElementById('searchInput');
  const searchResults = document.getElementById('searchResults');
  let currentRequest = null;

  searchInput.addEventListener('input', () => {
    if (currentRequest) {
      currentRequest.abort();
    }

    const searchTerm = searchInput.value.trim();

    if (searchTerm === '') {
      return;
    }

    const controller = new AbortController();
    const signal = controller.signal;

    currentRequest = controller;

    fetch(`https://api.example.com/search?term=${searchTerm}`, { signal })
      .then(response => response.json())
      .then(data => {
        searchResults.innerHTML = data.map(item => `<li>${item.name}</li>`).join('');
      })
      .catch(error => {
        if (error.name === 'AbortError') {
          // Request was canceled, ignore this error
        } else {
          console.error('Error:', error);
        }
      })
      .finally(() => {
        currentRequest = null;
      });
  });
</script>
```

Swipe ⟶

</> Example Use Case: On a search Input

The image below shows the **canceled** request in the browser's network tab **in a case** when the user types consecutively and the previous request is still pending.

Name	Status	Type	Initiator	Size	T...	Waterfall
search?q=d	(cance...	fetch	index.js:24	0 B	2...	
search?q=df	200	fetch	index.js:24	862 B	2...	
search?q=dff	(cance...	fetch	index.js:24	0 B	1...	
search?q=dffk	200	fetch	index.js:24	126 B	2...	
search?q=dffkg	200	fetch	index.js:24	125 B	2...	

Did you Find it useful?

Follow me for more...

Decryptus_

Dominic Orefuwa