



# JavaScript hashchange Event

**Summary:** in this tutorial, you will learn about the JavaScript `hashchange` event and how to handle it effectively.

## Introduction to the JavaScript hashchange event

The URL hash is everything that follows the pound sign ( `#` ) in the URL. For example, suppose that you have the following URL:

```
https://www.javascripttutorial.net/javascript-dom/javascript-hashchange/#header
```

In this URL, the hash is `header` .

When the hash changes to `footer` , the `hashchange` event will fire:

```
https://www.javascripttutorial.net/javascript-dom/javascript-hashchange/#footer
```

In this example, the hash changes from `#header` to `#footer` .

To attach an event listener to the `hashchange` event, you call the `addEventListener()` method on the `window` object:

```
window.addEventListener('hashchange', () => {  
    console.log('The URL has has changed');  
});
```

To get the current URL hash, you access the `hash` property of the `location` object:

```
window.addEventListener('hashchange', () => {  
    console.log(`The current URL hash is ${location.hash}`);  
});
```

```
});
```

Additionally, you can handle the `hashchange` event by assigning an event listener to the `onhashchange` property of the `window` object:

```
window.onhashchange = () => {  
    // handle hashchange event here  
};
```

## JavaScript hashchange event example

The hashchange event can be useful for creating a single-page application (SPA) or updating content dynamically without reloading the whole page.

The following example shows how to use the `hashchange` event to update the content of a div based on the URL hash.

First, create a simple HTML file (index.html) with some navigation links and a main section:

```
<!DOCTYPE html>  
<html lang="en">  
  <head>  
    <meta charset="UTF-8">  
    <meta name="viewport" content="width=device-width, initial-scale=1.0">  
    <title>JavaScript HashChange Example</title>  
    <link rel="stylesheet" href="css/style.css">  
    <script src="js/app.js" defer></script>  
  </head>  
  <body>  
    <header>  
      <nav>  
        <a href="#home">Home</a>  
        <a href="#about">About</a>  
        <a href="#contact">Contact</a>  
      </nav>  
    </header>  
    <main id="content">
```

```
    <h1>HashChange Event Demo!</h1>
    <p>Click the links above to change the content.</p>
  </main>
</body>
</html>
```

Second, create a JavaScript file `app.js` that handles the `hashchange` event and updates the content respectively:

```
const updateContent = () => {
  const content = document.getElementById('content');
  const hash = window.location.hash.substring(1); // Remove the '#' from the hash

  switch (hash) {
    case 'home':
      content.innerHTML = `
        <h1>Home</h1>
        <p>Welcome to the home page.</p>
      `;
      break;
    case 'about':
      content.innerHTML = `
        <h1>About</h1>
        <p>Learn more about us on this page.</p>
      `;
      break;
    case 'contact':
      content.innerHTML = `
        <h1>Contact</h1>
        <p>Get in touch with us.</p>
      `;
      break;
  }
};

// Event listener for hashchange
window.addEventListener('hashchange', updateContent);
```

Output:

Output

[Home](#)   [About](#)   [Contact](#)

# Welcome!

Click the links above to change the content.

## Summary

- The `hashchange` event fires when the URL hash changes.
- To register an event handler, you call the `addEventListener()` method or assign an event handler to the `onhashchange` property of the `window` object.