

Brower API

Page Visibility API

With tabbed browsing, there is a reasonable chance that any given webpage is in the background and thus not visible to the user. The Page Visibility API provides events you can watch for to know when a document becomes visible or hidden, as well as features to look at the current visibility state of the page.

Notes: The Page Visibility API is especially useful for saving resources and improving performance by letting a page avoid performing unnecessary tasks when the document isn't visible

When the user minimizes the window or switches to another tab, the API sends a visibilitychange event to let listeners know the state of the page has changed. You can detect the event and perform some actions or behave differently. For example, if your web app is playing a video, it can pause the video when the user puts the tab into the background, and resume playback when the user returns to the tab. The user doesn't lose their place in the video, the video's soundtrack doesn't interfere with audio in the new foreground tab, and the user doesn't miss any of the video in the meantime.

Use cases:

Let's consider a few use cases for the Page Visibility API.

- A site has an image carousel that shouldn't advance to the next slide unless the user is viewing the page
- An application showing a dashboard of information doesn't want to poll the server for updates when the page isn't visible
- A page wants to detect when it is being prerendered so it can keep accurate count of page views
- A site wants to switch off sounds when a device is in standby mode (user pushes power button to turn screen off)
-

Developers have historically used imperfect proxies to detect this. For example, watching for blur and focus events on the window helps you know when your page is not the active page, but it does not tell you that your page is actually hidden to the user. The Page Visibility API addresses this.

Ex:

The example, which pauses the video when you switch to another tab and plays again when you return to its tab, was created with the following code:

```
var hidden, visibilityChange;
if (typeof document.hidden !== "undefined") { // Opera 12.10 and Firefox 18 and later
support
    hidden = "hidden";
    visibilityChange = "visibilitychange";
} else if (typeof document.msHidden !== "undefined") {
    hidden = "msHidden";
    visibilityChange = "msvisibilitychange";
} else if (typeof document.webkitHidden !== "undefined") {
    hidden = "webkitHidden";
    visibilityChange = "webkitvisibilitychange";
}

var videoElement = document.getElementById("videoElement");

// If the page is hidden, pause the video;
// if the page is shown, play the video
function handleVisibilityChange() {
    if (document[hidden]) {
        videoElement.pause();
    } else {
        videoElement.play();
    }
}
```

```

// Warn if the browser doesn't support addEventListener or the Page Visibility API
if (typeof document.addEventListener === "undefined" || hidden === undefined) {
  console.log("This demo requires a browser, such as Google Chrome or Firefox, that
  supports the Page Visibility API.");
} else {
  // Handle page visibility change
  document.addEventListener(visibilityChange, handleVisibilityChange, false);

  // When the video pauses, set the title.
  // This shows the paused
  videoElement.addEventListener("pause", function(){
    document.title = 'Paused';
  }, false);

  // When the video plays, set the title.
  videoElement.addEventListener("play", function(){
    document.title = 'Playing';
  }, false);
}

```

Ex 2:

Automatic signOut of a tab when signout action done in another tab.

```

export function handleVisibility () {
  if (document[hidden]) {
    return true
  } else {
    stillSignedIn()
    return false
  }
}

```

```
}  
}  
export function stillSignedIn () {  
  if (!(localStorage.getItem('emailId') && localStorage.getItem('password'))) {  
    store.dispatch(actions.signOutOnClick())  
  }  
}
```

For more,

https://developer.mozilla.org/en-US/docs/Web/API/Page_Visibility_API