

# Prevent web app into sleep mode by Javascript

You need to take a look at navigator.wakeLock object if you do not want to prevent your web app into the sleep mode. Unfortunately it is not supported in Safari and Firefox, but Chrome browser supports very well.

You can check if the feature is available on the current browser by the following simple function:

```
const canWakeLock = () => 'wakeLock' in navigator;
```

To lock the wake state, you need to call request() on the wakeLock object in the navigator like below:

```
let wakelock;
async function lockWakeState() {
  if(!canWakeLock()) return;
  try {
    wakelock = await navigator.wakeLock.request();
    wakelock.addEventListener('release', () => {
      console.log('Screen Wake State Locked:', !wakelock.released);
    });
    console.log('Screen Wake State Locked:', !wakelock.released);
  } catch(e) {
    console.error('Failed to lock wake state with reason:', e.message);
  }
}
```

Followings are the function can release from wakeLock

```
function releaseWakeState() {
  if(wakelock) wakelock.release();
  wakelock = null;
}
```

Below is the working example based on above:

```
let wakelock;
const canWakeLock = () => 'wakeLock' in navigator;

async function lockWakeState() {
  if(!canWakeLock()) return;
  try {
    wakelock = await navigator.wakeLock.request();
    wakelock.addEventListener('release', () => {
      console.log('Screen Wake State Locked:', !wakelock.released);
    });
    console.log('Screen Wake State Locked:', !wakelock.released);
  } catch(e) {
    console.error('Failed to lock wake state with reason:', e.message);
  }
}

function releaseWakeState() {
  if(wakelock) wakelock.release();
  wakelock = null;
}

lockWakeState();
setTimeout(releaseWakeState, 5000); // release wake state after 5 seconds
```

