



خفن ترین API های Javascript

JS

خلاصه ای از مفهوم API

API مخفف عبارت "Application Programming Interface" است و به عبارت دیگر یک مجموعه از دستورات و توابع مورد استفاده برنامه نویسان برای ارتباط با یک سیستم یا خدمت خاص اشاره دارد. در زبان برنامه نویسی JavaScript نیز مفهوم API بسیار مهمی است. API ها در JavaScript به شما اجازه می‌دهند تا از ویژگی‌ها و توابع مختلفی که توسط مرورگر یا سیستم عامل ارائه می‌شود، استفاده کنید.



Gamepad API

Gamepad API اجازه می‌دهد تا در برنامه‌های وب با دسته‌ها و کنترلرهای بازی تعامل کرد. این API به شما اطلاعاتی مانند وضعیت دکمه‌ها، محورها (مثلاً جوی‌استیک)، و دسته کاربر را ارائه می‌دهد. این اطلاعات به شما امکان می‌دهند تا بازی‌ها و تجربیات تعاملی دیگر را از طریق دسته‌ها و کنترلرها ایجاد کنید. در ادامه یک قطعه کد نمونه برای مشاهده وضعیت دسته بازی در JavaScript آورده شده است:



Gamepad API

```
1 // تابعی برای بررسی وضعیت کنترلرهای بازی
2 function checkGamepads() {
3     const gamepads = navigator.getGamepads(); // لیست تمام کنترلرهای بازی متصل
4     for (const gamepad of gamepads) {
5         if (gamepad) {
6             console.log("Gamepad ID: " + gamepad.id);
7             console.log("Buttons Pressed: " + gamepad.buttons.map(button => button.pressed));
8             console.log("Axes Values: " + gamepad.axes);
9         }
10    }
11 }
12
13 // یک تابع متداول برای به روزرسانی وضعیت کنترلرها
14 function updateGamepadStatus() {
15     requestAnimationFrame(updateGamepadStatus);
16     checkGamepads();
17 }
18
19 // شروع به روزرسانی وضعیت کنترلرها
20 updateGamepadStatus();
```

Web MIDI API

Web MIDI API به شما امکان می‌دهد با دستگاه‌های موسیقی MIDI از طریق مرورگر تعامل کنید. با استفاده از این API می‌توانید دستگاه‌های MIDI را شناسایی کنید و داده‌ها را برای آنها ارسال کنید. در زیر یک قطعه کد نمونه برای استفاده از Web MIDI API در JavaScript آورده شده است:

در این قطعه کد، ما ابتدا چک می‌کنیم که مرورگر از Web MIDI API پشتیبانی می‌کند یا خیر. سپس با استفاده از `navigator.requestMIDIAccess` درخواست دسترسی به دستگاه‌های MIDI را ارسال می‌کنیم. بعد از دریافت دسترسی، لیست دستگاه‌های MIDI متصل را دریافت کرده و می‌توانیم از آنها برای ارسال و دریافت پیام‌های MIDI استفاده کنیم.

Web MIDI API

```
1 // در مرورگر Web MIDI API چک کردن پشتیبانی از
2 if (navigator.requestMIDIAccess) {
3   // MIDI درخواست دسترسی به دستگاه‌های
4   navigator.requestMIDIAccess()
5     .then(function(midiAccess) {
6       // متصل MIDI دریافت لیست دستگاه‌های
7       const inputs = midiAccess.inputs.values();
8       const outputs = midiAccess.outputs.values();
9
10      for (const input of inputs) {
11        input.onmidimessage = function(event) {
12          // دریافتی MIDI پردازش پیام‌های
13          console.log("Received MIDI Message:", event.data);
14        };
15      }
16
17      // MIDI به یک دستگاه MIDI ارسال پیام
18      for (const output of outputs) {
19        output.send([0x90, 60, 100]); // MIDI نمونه ارسال یک نوت
20      }
21    })
22    .catch(function(error) {
23      console.error("MIDI Access Error:", error);
24    });
25 } else {
26   console.error("Web MIDI API is not supported in this browser.");
27 }
28
```


MediaStream Image Capture API

MediaStream Image Capture API به شما اجازه می‌دهد تا از دوربین دستگاه کاربر تصاویر بگیرید و با آنها کار کنید. در ادامه یک قطعه کد نمونه برای استفاده از این API در JavaScript آورده شده است:

در این قطعه کد، ما ابتدا از

`navigator.mediaDevices.enumerateDevices` برای

شناسایی دستگاه‌های موجود در دستگاه کاربر استفاده می‌کنیم.

سپس از این دستگاه‌ها، دستگاه‌های ورودی ویدئو را انتخاب می‌کنیم

و اولین دوربین موجود را انتخاب می‌کنیم. سپس با استفاده از

`navigator.mediaDevices.getUserMedia` دسترسی به

دوربین را درخواست می‌دهیم و تصویر از دوربین گرفته می‌شود.

سپس با استفاده از کلاس `ImageCapture`، می‌توانیم یک عکس از

دوربین گرفته و آن را در صفحه نمایش دهیم.

```
1 // انتخاب دوربین با استفاده از MediaDevices API
2 navigator.mediaDevices.enumerateDevices()
3 .then(function(devices) {
4     const videoDevices = devices.filter(device => device.kind === 'videoinput');
5
6     if (videoDevices.length > 0) {
7         // انتخاب اولین دوربین موجود
8         const selectedDevice = videoDevices[0];
9
10        // دسترسی به دوربین و دریافت تصویر
11        navigator.mediaDevices.getUserMedia({ video: { deviceId: selectedDevice.deviceId } })
12        .then(function(stream) {
13            // برای تصویرگیری ImageCapture ایجاد یک شیء
14            const imageCapture = new ImageCapture(stream.getVideoTracks()[0]);
15
16            // گرفتن یک عکس
17            imageCapture.takePhoto()
18            .then(function(blob) {
19                // نمایش تصویر در یک تگ تصویر در صفحه
20                const img = document.createElement('img');
21                img.src = URL.createObjectURL(blob);
22                document.body.appendChild(img);
23            })
24            .catch(function(error) {
25                console.error("Error capturing image:", error);
26            });
27        })
28        .catch(function(error) {
29            console.error("Error accessing camera:", error);
30        });
31    } else {
32        console.error("No video input devices found.");
33    }
34 })
35 .catch(function(error) {
36     console.error("Error enumerating devices:", error);
37 });
38
```


Ambient Light Sensor API

Ambient Light Sensor API به شما امکان می‌دهد تا نور محیطی در محیط دستگاه کاربر را اندازه‌گیری کرده و بر اساس تغییرات نور محیطی واکنش نشان دهید. این API می‌تواند در برنامه‌ها و تجربیات وب مرتبط با تغییرات نور محیطی، مفید باشد. در ادامه یک قطعه کد نمونه برای استفاده از Ambient Light Sensor API در JavaScript آورده شده است:

در این قطعه کد، ما ابتدا چک می‌کنیم که مرورگر از Ambient Light Sensor API پشتیبانی می‌کند یا خیر. سپس یک شیء `AmbientLightSensor` ایجاد می‌کنیم و تابعی برای بررسی تغییرات نور محیطی با استفاده از `event onreading` تعریف می‌کنیم. در هر بار اندازه‌گیری، مقدار نور محیطی به وسیله `sensor.illuminance` دریافت می‌شود و واکنش‌های مختلف بر اساس این مقدار اعمال می‌شوند.

Ambient Light Sensor API

```
1 // در مرورگر Ambient Light Sensor API چک کردن پشتیبانی از
2 if ('AmbientLightSensor' in window) {
3   // AmbientLightSensor ایجاد یک شیء
4   const sensor = new AmbientLightSensor();
5
6   // تعریف تابعی برای بررسی تغییرات نور محیطی
7   sensor.onreading = function(event) {
8     console.log("Ambient Light Level: " + sensor.illuminance);
9
10    // اعمال واکنش بر اساس تغییرات نور محیطی
11    if (sensor.illuminance < 100) {
12      // باشد lux اگر نور کمتر از 100
13      document.body.style.backgroundColor = 'darkgray';
14    } else {
15      // در غیر این صورت
16      document.body.style.backgroundColor = 'white';
17    }
18  };
19
20  // شروع خواندن داده‌های نور محیطی
21  sensor.start();
22 } else {
23   console.error("Ambient Light Sensor API is not supported in this browser.");
24 }
25
```

Ambient Light Sensor API

Wake Lock API به توسعه دهندگان وب اجازه می‌دهد تا درخواست‌های نگه داشتن دستگاه کاربر را کنترل کنند. این API به کاربران اجازه می‌دهد تا تنظیمات مربوط به مدت زمانی که صفحه وب فعال باشد و مدت زمانی که صفحه خاموش است را تنظیم کنند. این کار برای تجربه کاربری بهتر در برنامه‌های وب مهم است. در ادامه، نمونه‌ای از کار با Wake Lock API در JavaScript آورده شده است:

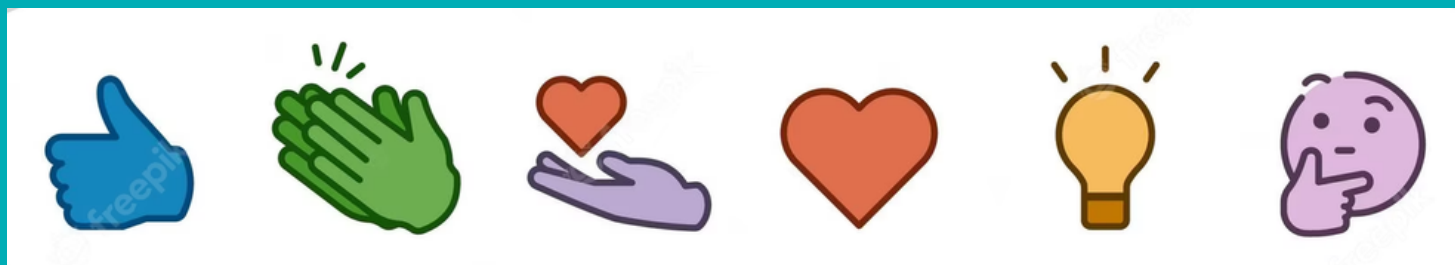
در این قطعه کد، ما ابتدا چک می‌کنیم که مرورگر از Wake Lock API پشتیبانی می‌کند یا خیر. سپس توابعی برای درخواست نگه داشتن صفحه (`requestWakeLock`) و رها کردن نگه داشتن صفحه (`releaseWakeLock`) تعریف می‌کنیم.

در تابع `requestWakeLock`، ما از `navigator.wakeLock.request` استفاده می‌کنیم تا نگه داشتن صفحه را درخواست دهیم و در تابع `releaseWakeLock` نگه داشتن صفحه را رها می‌کنیم. سپس با استفاده از دکمه‌های مربوطه در صفحه وب، کاربران می‌توانند نگه داشتن صفحه را درخواست یا رها کنند.

Wake Lock API

```
1 // در مرورگر Wake Lock API چک کردن پشتیبانی از
2 if ('wakeLock' in navigator) {
3   let wakeLock = null;
4
5   // تابعی برای درخواست نگه داشتن صفحه
6   async function requestWakeLock() {
7     try {
8       wakeLock = await navigator.wakeLock.request('screen');
9       console.log('Screen Wake Lock is active!');
10    } catch (error) {
11      console.error('Error requesting Screen Wake Lock:', error);
12    }
13  }
14
15   // تابعی برای رها کردن نگه داشتن صفحه
16   function releaseWakeLock() {
17     if (wakeLock) {
18       wakeLock.release().then(() => {
19         console.log('Screen Wake Lock is released.');
```


ممنون که تا اینجا همراهی کردی رفیق
خوشحال میشم نظرت رو درباره پست
گامنت کنی 😊



از اینا یادت نره :