YES, YOUR BROWSER CAN DO THAT (probably)

A Look At Modern Web APIs You Might Not Know

JULIAN BURR @ ../NEW 2024

PROGRESSIVE ENHANCEMENT

Progressive enhancement is a design philosophy that provides a baseline of essential content and functionality to as many users as possible while delivering the best possible experience only to users of the most modern browsers

https://developer.mozilla.org/docs/Glossary/Progressive_Enhancement

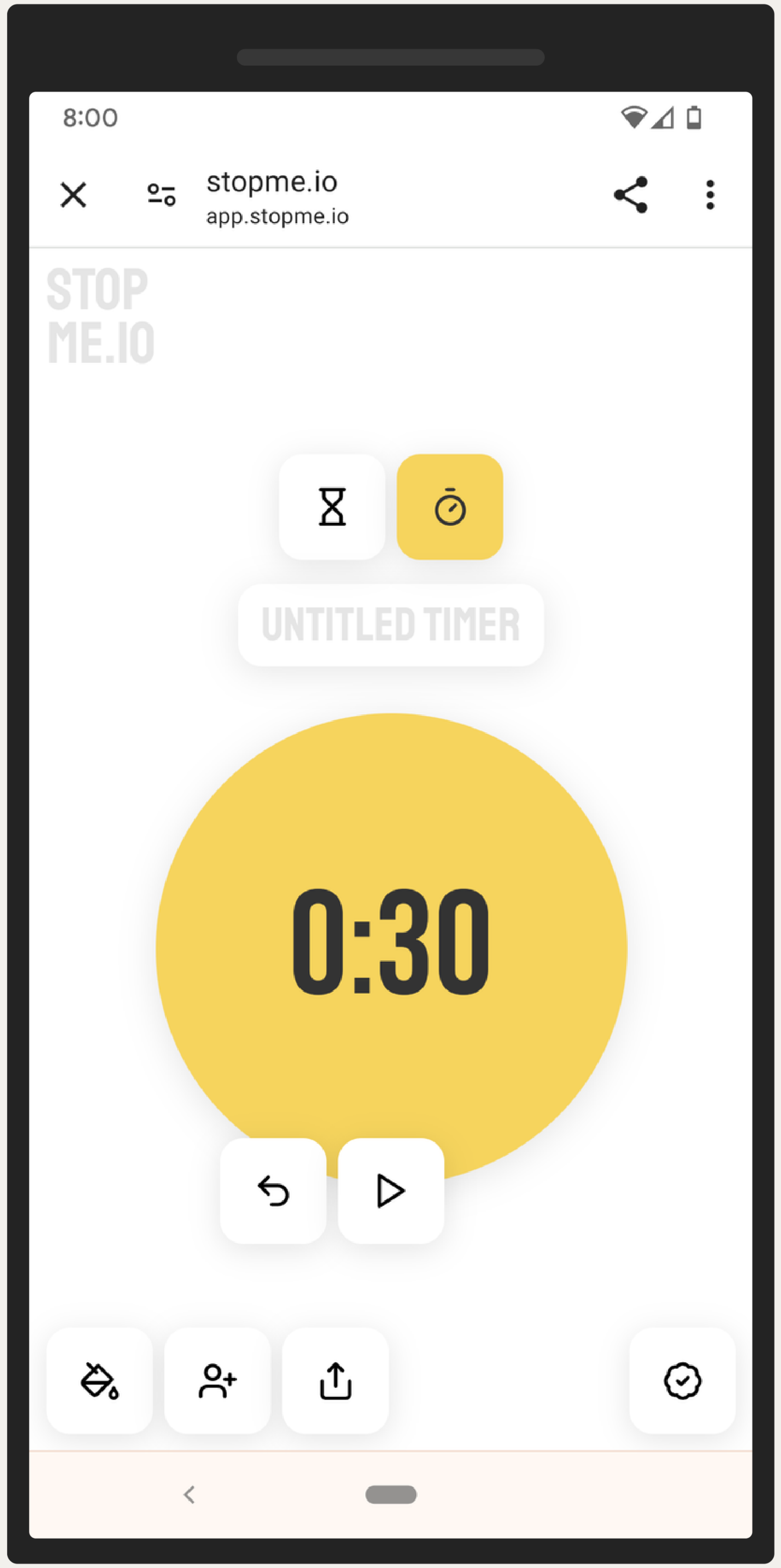
MOVING THE NEEDLE & PHONE GAP

aka Cordova

LET'S LOOK AT SOME BROWSER APIS

STOPME.IO

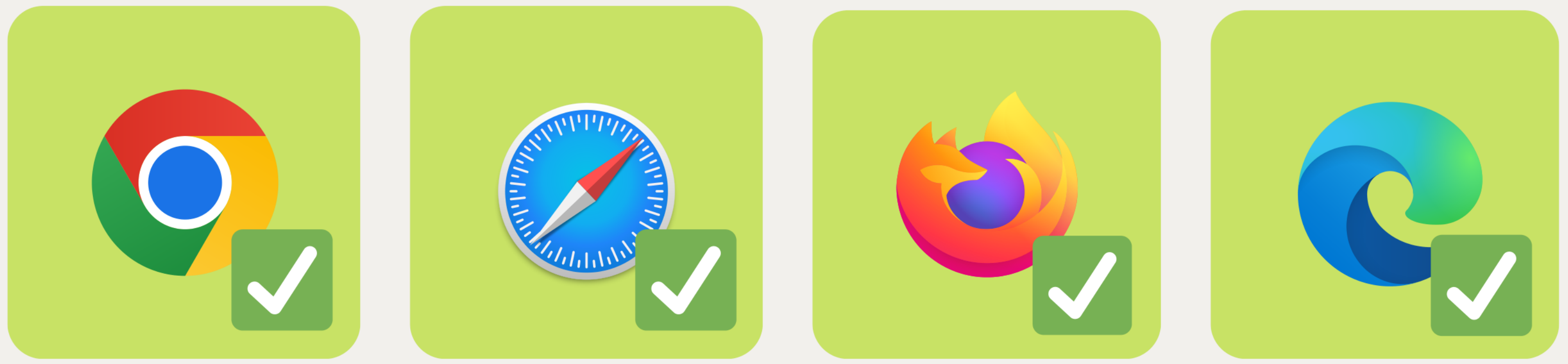
The ultimate SaaS (Stopwatch as a Service) product for everyone



OBSERVE YOUR APP …

01

RESIZE OBSERVER



01

RESIZE OBSERVER

```
const callback = (entries) => {
  entries.forEach((entry) => {
    // *.contentBoxSize
  });
}

const observer =
  new ResizeObserver(callback, options);

const el = document.querySelector("element");
observer.observe(el);
```

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

02

INTERSECTION OBSERVER

✓ ✓ ✓ ✓

02

INTERSECTION OBSERVER

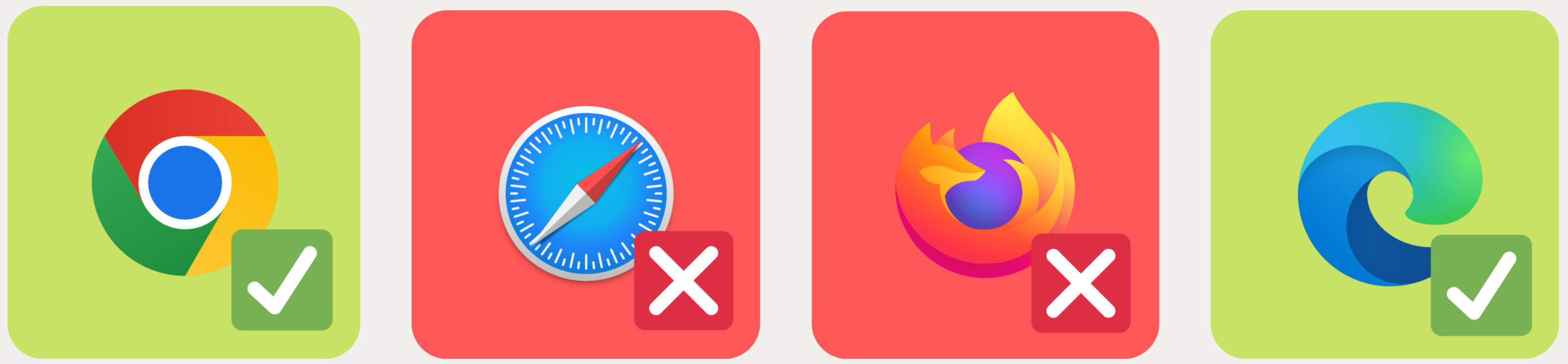
```javascript
const callback = (entries) => {
  entries.forEach((entry) => {
    // *.isIntersecting
    // *.intersectionRect
    // *.intersectionRatio
    // ...
  });
}

const observer =
  new IntersectionObserver(callback, options);

const el = document.querySelector("element");
observer.observe(el);
```

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

… AND EVEN THE USER'S DEVICE

03

NETWORK INFORMATION API



03

NETWORK INFORMATION API

```
// Progressive enhancement
const isSupported = “connection” in navigator;

let connection = navigator.connection;
// *.type (e.g. wifi, ...)
// *.effectiveType (e.g. 2g, 3g, ...)
// *.downlink
// *.downlinkMax
// *.rtt
// *.saveData

const callback = () => {
  connection = navigator.connection;
}

navigator.connection.addEventListener(
  "change",
  callback
);
```

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

NETWORK INFORMATION API

✓ ✗ ✗ ✓

```
// Progressive enhancement
const isSupported = “connection” in navigator;

let connection = navigator.connection;
// *.type (e.g. wifi, ...)
// *.effectiveType (e.g. 2g, 3g, ...)
// *.downlink
// *.downlinkMax
// *.rtt
// *.saveData

const callback = () => {
  connection = navigator.connection;
}

navigator.connection.addEventListener(
  "change",
  callback
);
```

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

03

NETWORK INFORMATION API

✓ ✗ ✗ ✓

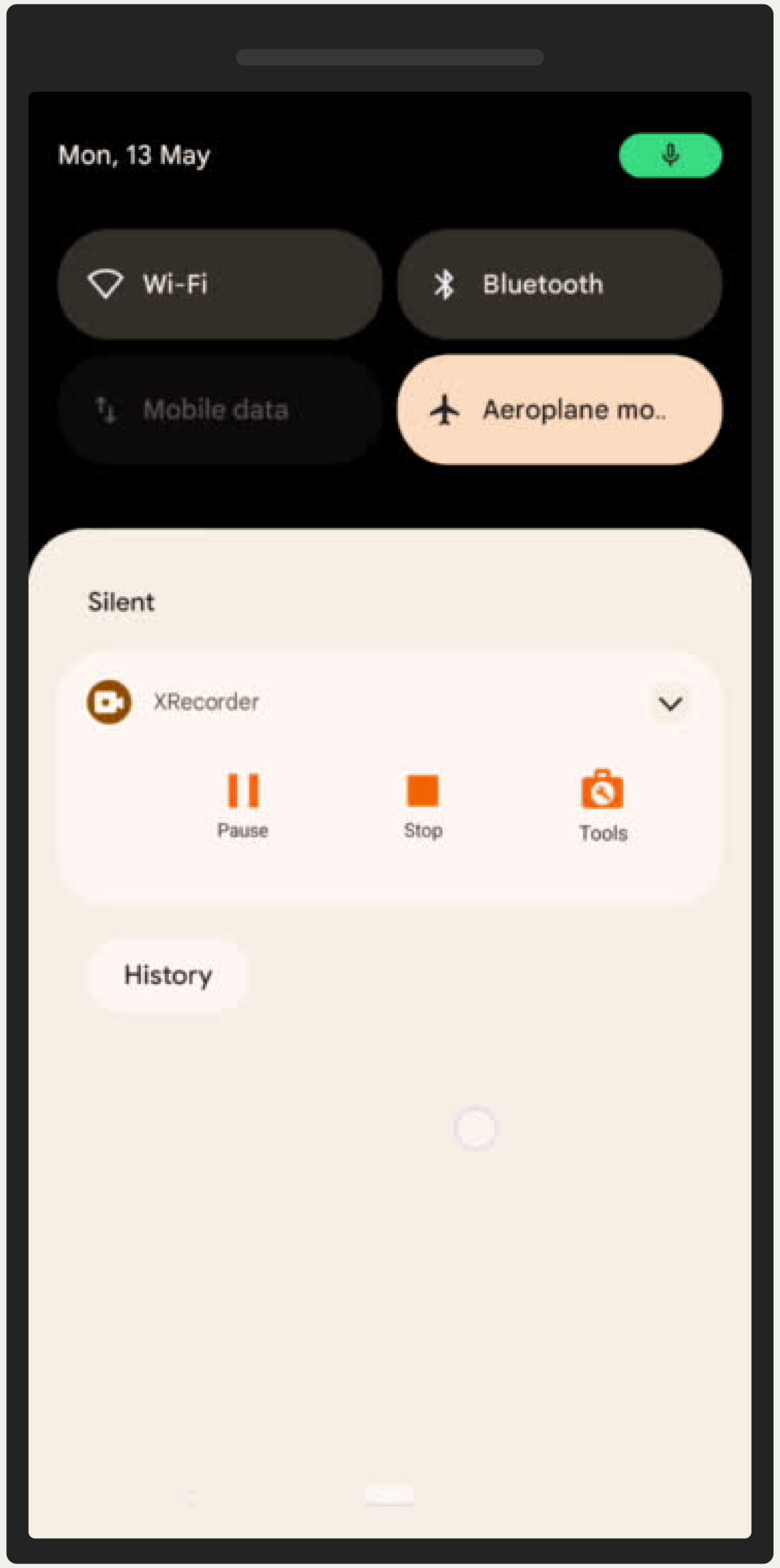
```
// Progressive enhancement
const isSupported = “connection” in navigator;

let connection = navigator.connection;
// *.type (e.g. wifi, ...)
// *.effectiveType (e.g. 2g, 3g, ...)
// *.downlink
// *.downlinkMax
// *.rtt
// *.saveData

const callback = () => {
  connection = navigator.connection;
}

navigator.connection.addEventListener(
  "change",
  callback
);
```

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

03

NETWORK INFORMATION API

✓ ✗ ✗ ✓


Mon, 13 May

Wi-Fi
Bluetooth
Mobile data
Aeroplane mo..

Silent

XRecorder

Pause
Stop
Tools

History

04

PAGE VISIBILITY API

04

PAGE VISIBILITY API

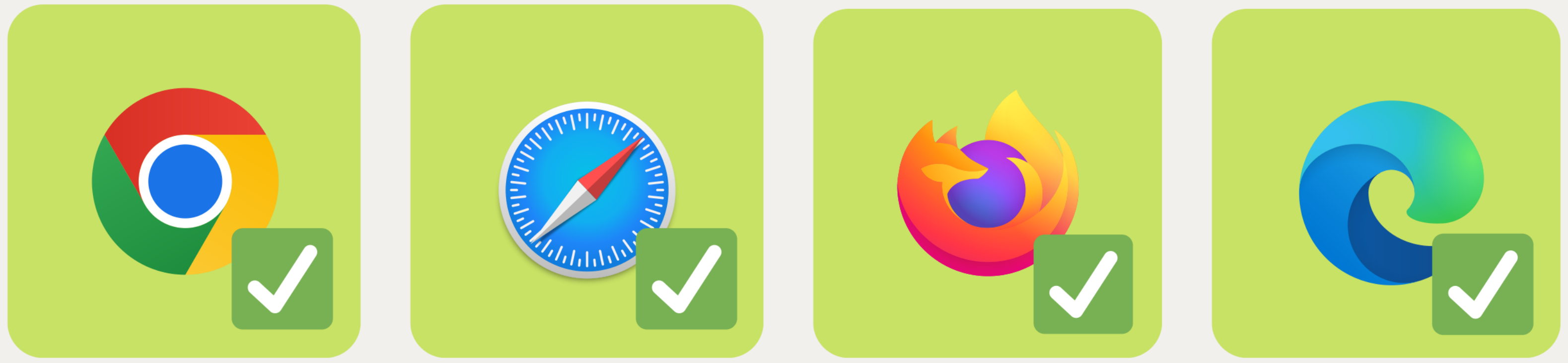
✓ ✓ ✓ ✓

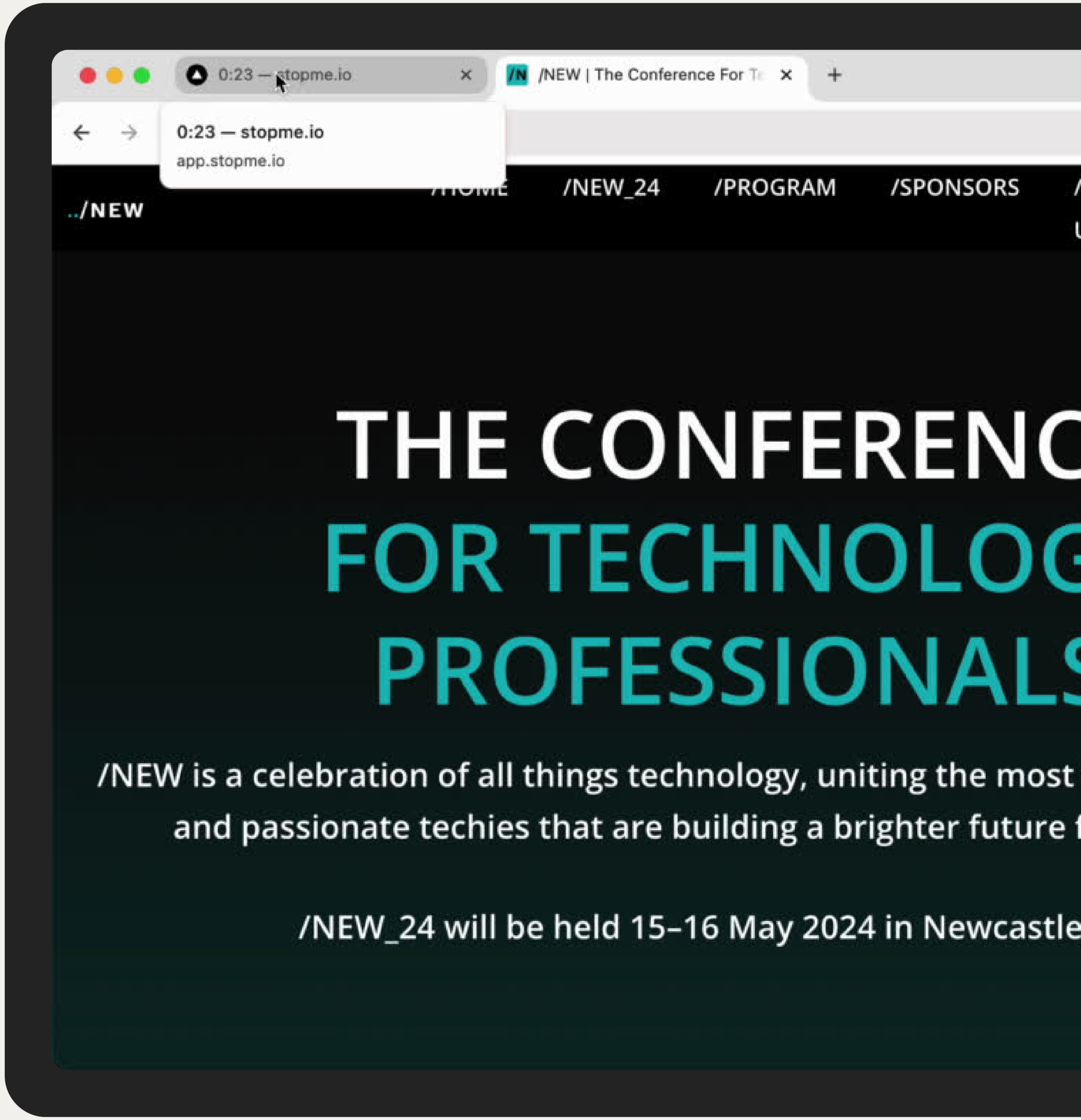
```
let isHidden = document.hidden;

const callback = () => {
  isHidden = document.hidden;
}

document.addEventListener(
  "visibilitychange",
  callback
);
```

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

04

PAGE VISIBILITY API





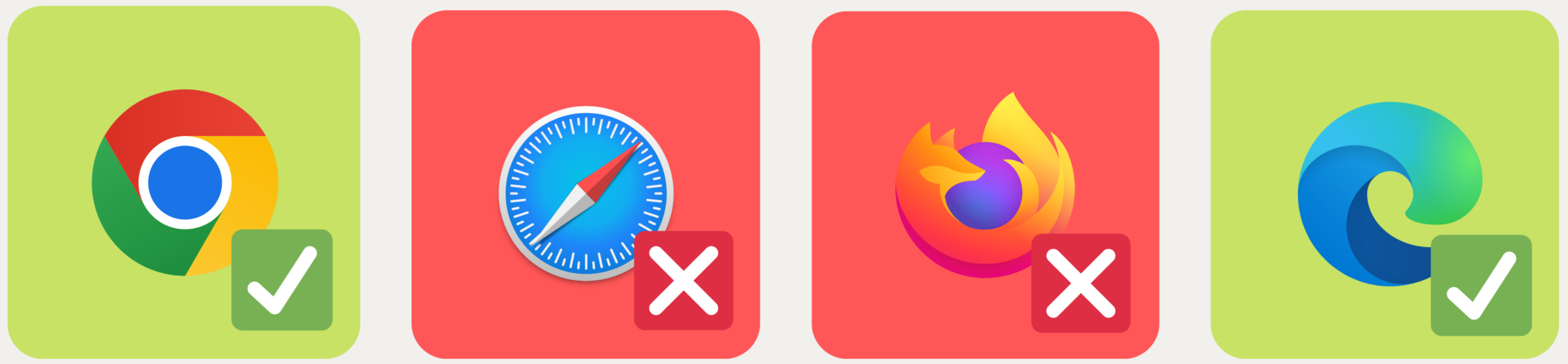
THE CONFERENC
FOR TECHNOLOG
PROFESSIONALS

/NEW is a celebration of all things technology, uniting the most
and passionate techies that are building a brighter future f

/NEW_24 will be held 15–16 May 2024 in Newcastle

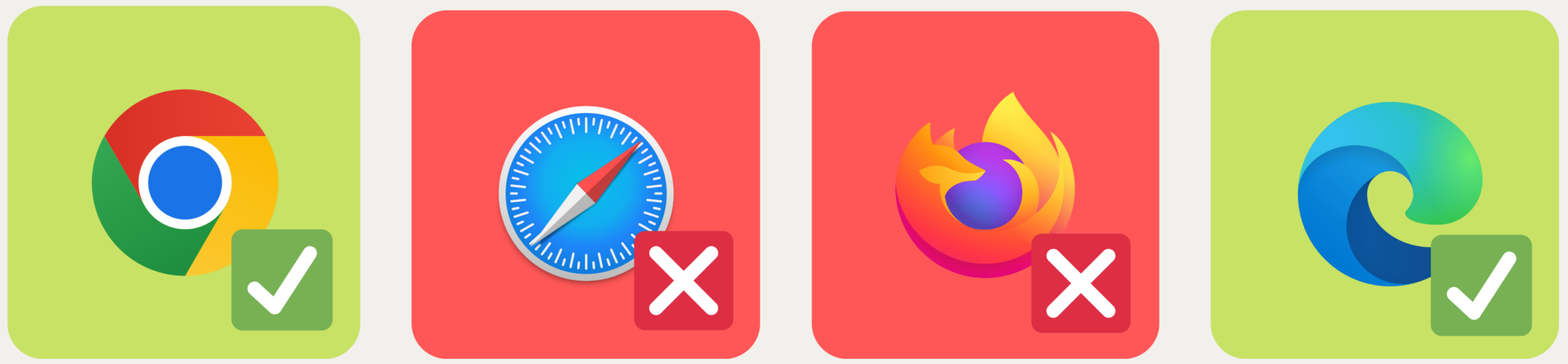
05

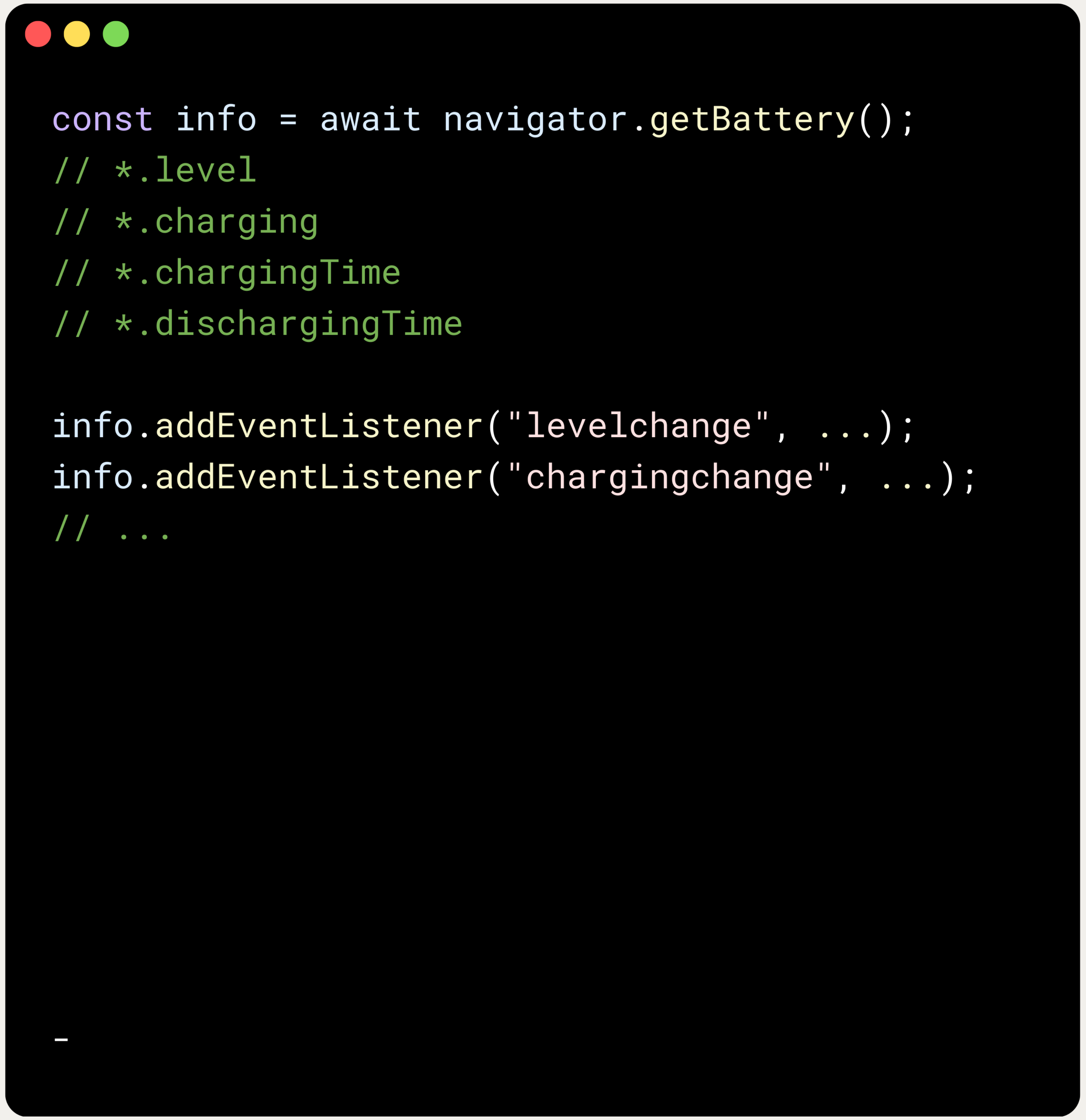
BATTERY STATUS API



05

BATTERY STATUS API





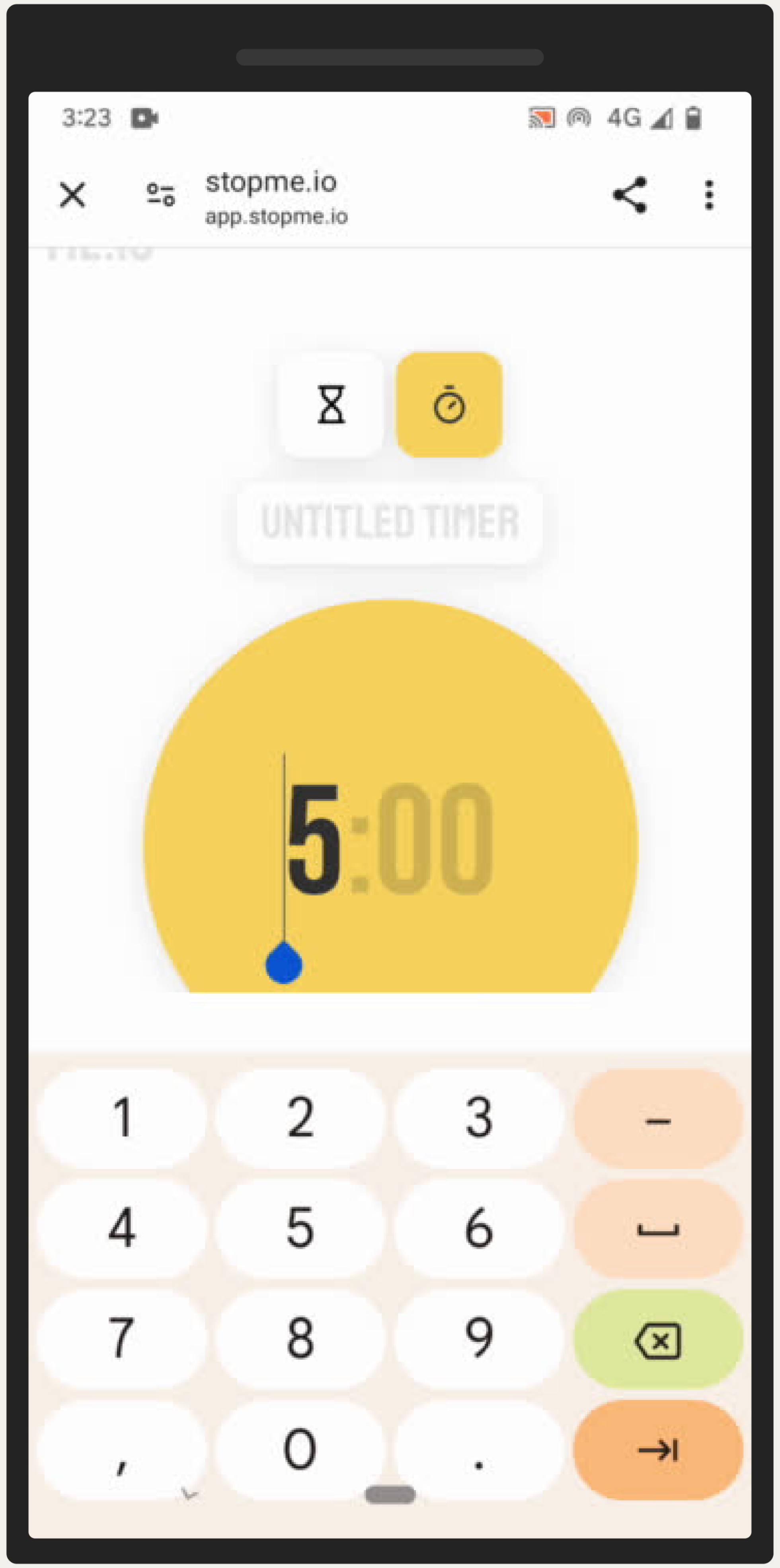
```
const info = await navigator.getBattery();
// *.level
// *.charging
// *.chargingTime
// *.dischargingTime

info.addEventListener("levelchange", ...);
info.addEventListener("chargingchange", ...);
// ...



_
```

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

05

BATTERY STATUS API

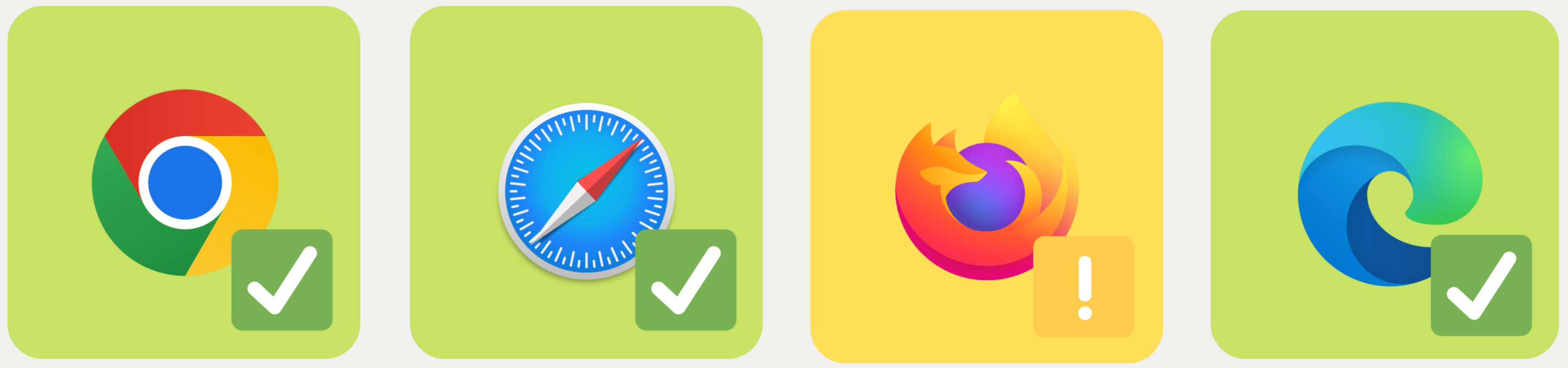
Chrome	Safari	Firefox	Edge
✓	✗	✗	✓



ENHANCE YOUR COMPONENTS

06

SCREEN WAKE LOCK API



06

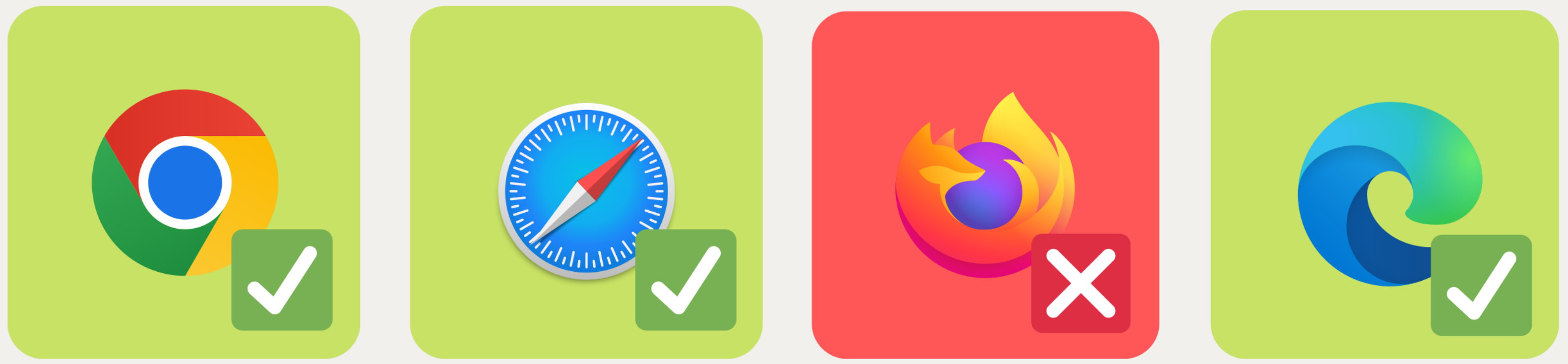
SCREEN WAKE LOCK API

```
const wakeLock = await navigator
  .wakeLock
  .request("screen")
  .catch((e) => {
    // Lock request failed, often because
    // of low battery etc
  });

await wakeLock.release();
```

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

07

VIBRATION API



07

VIBRATION API

✓ ✓ ✗ ✓

```
navigator.vibrate(duration);

// Patterns
// e.g. vibrate for 100ms with 50ms pauses
navigator.vibrate([100, 50, 100, 50, 100]);

// Stop long vibration or pattern
navigate.vibrate(0);
```

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

07

VIBRATION API

✓ ✓ ✗ ✓

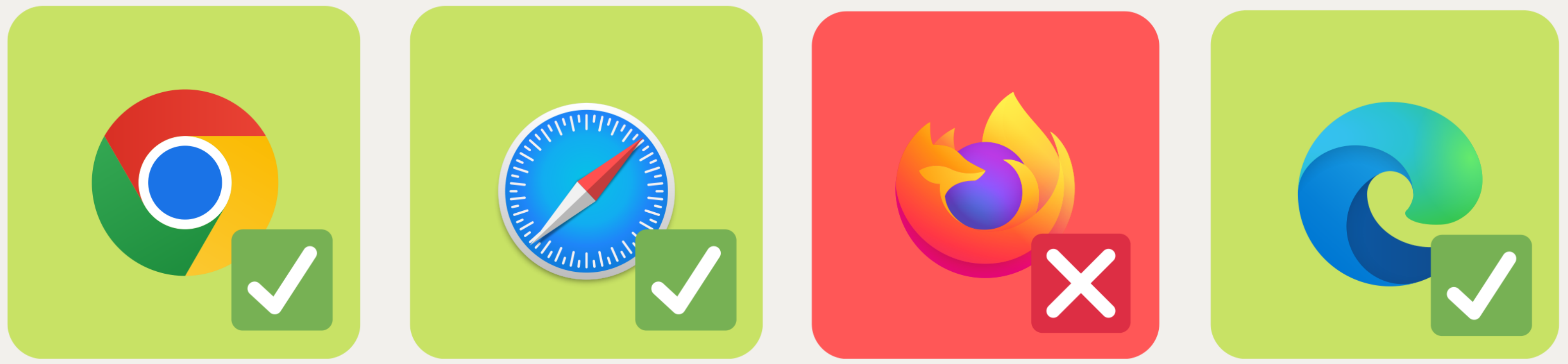
```
// SOS in Morse
navigator.vibrate([
  100, 30, 100, 30, 100, 30, 200, 30, 200,
  30, 200, 30, 100, 30, 100, 30, 100
]);

// Super Mario
navigator.vibrate([
  125, 75, 125, 275, 200, 275, 125, 75,
  125, 275, 200, 600, 200, 600
]);

// Star Wars
navigator.vibrate([
  500, 110, 500, 110, 450, 110, 200, 110,
  170, 40, 450, 110, 200, 110, 170, 40, 500
]);
```

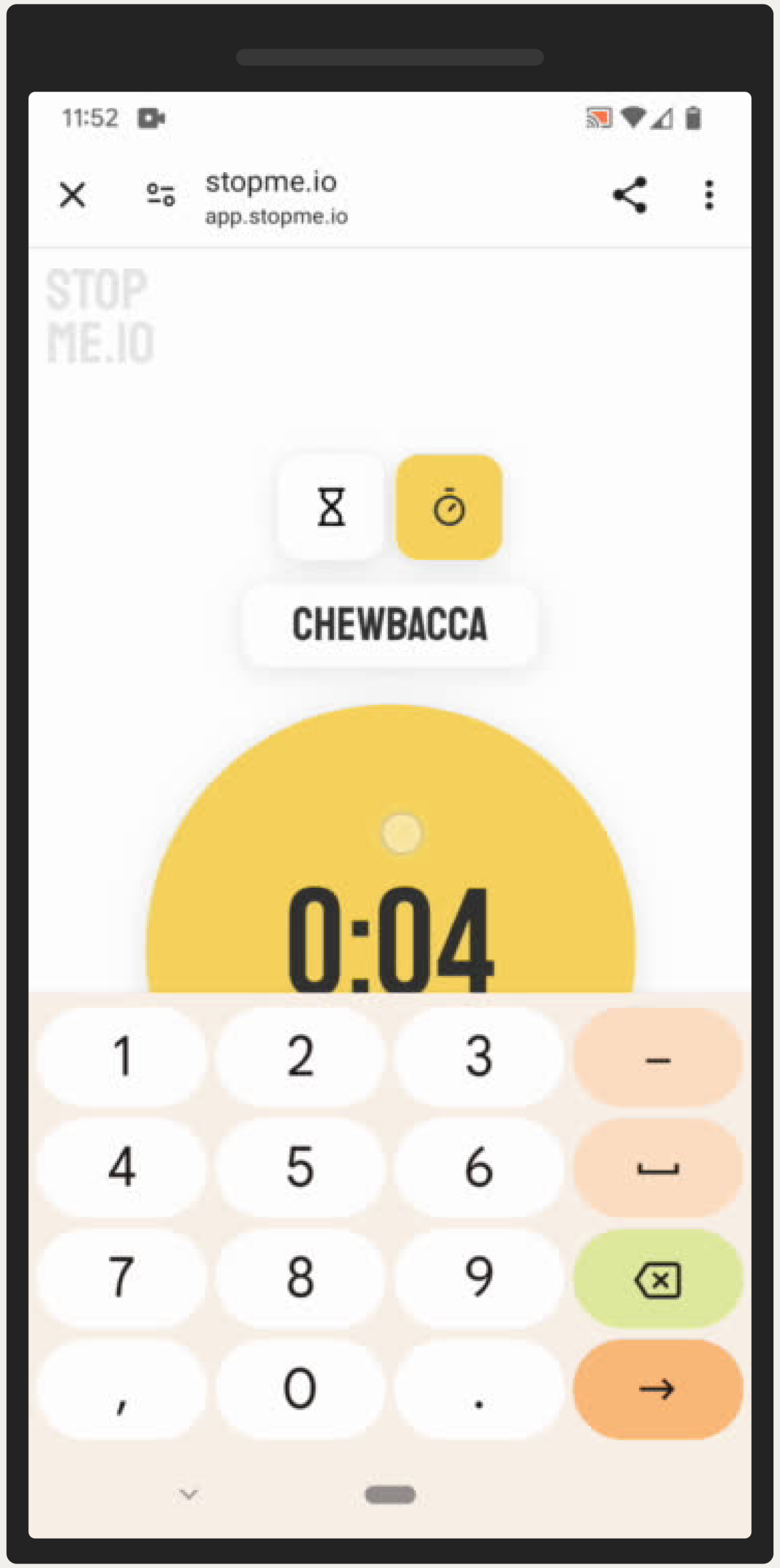
https://gearside.com/custom-vibration-patterns-mobile-devices/

07

It vibrates,
I swear!

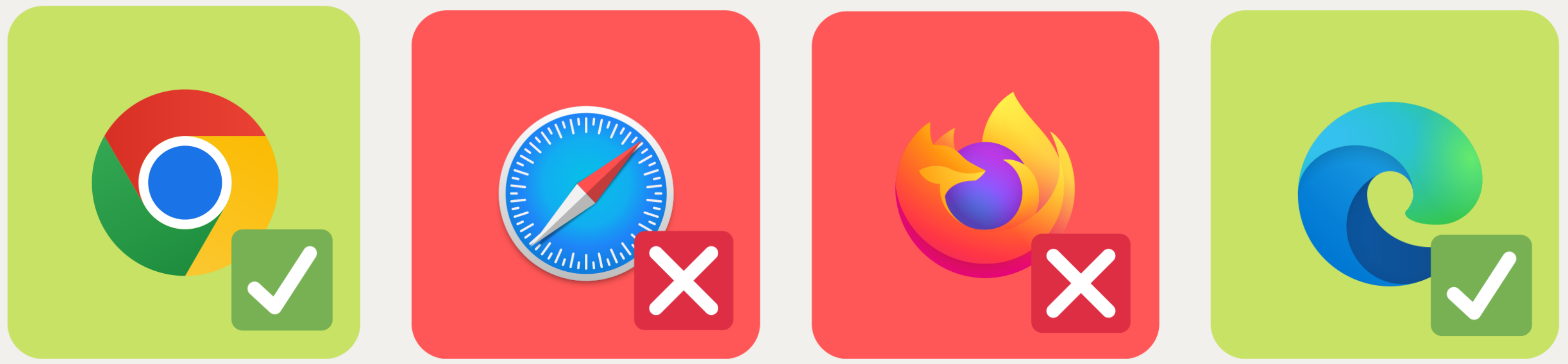
VIBRATION API





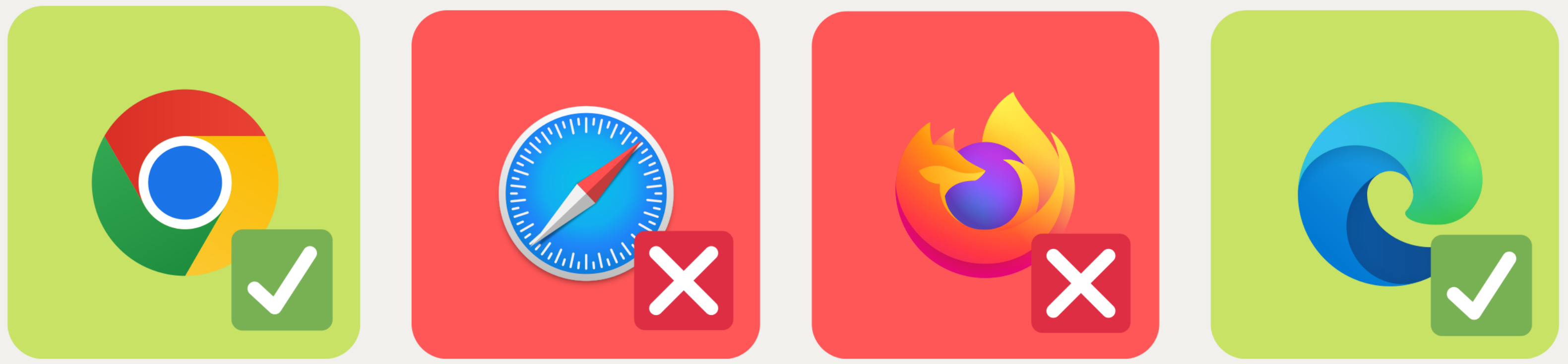
08

EYEDROPPER API



08

EYEDROPPER API

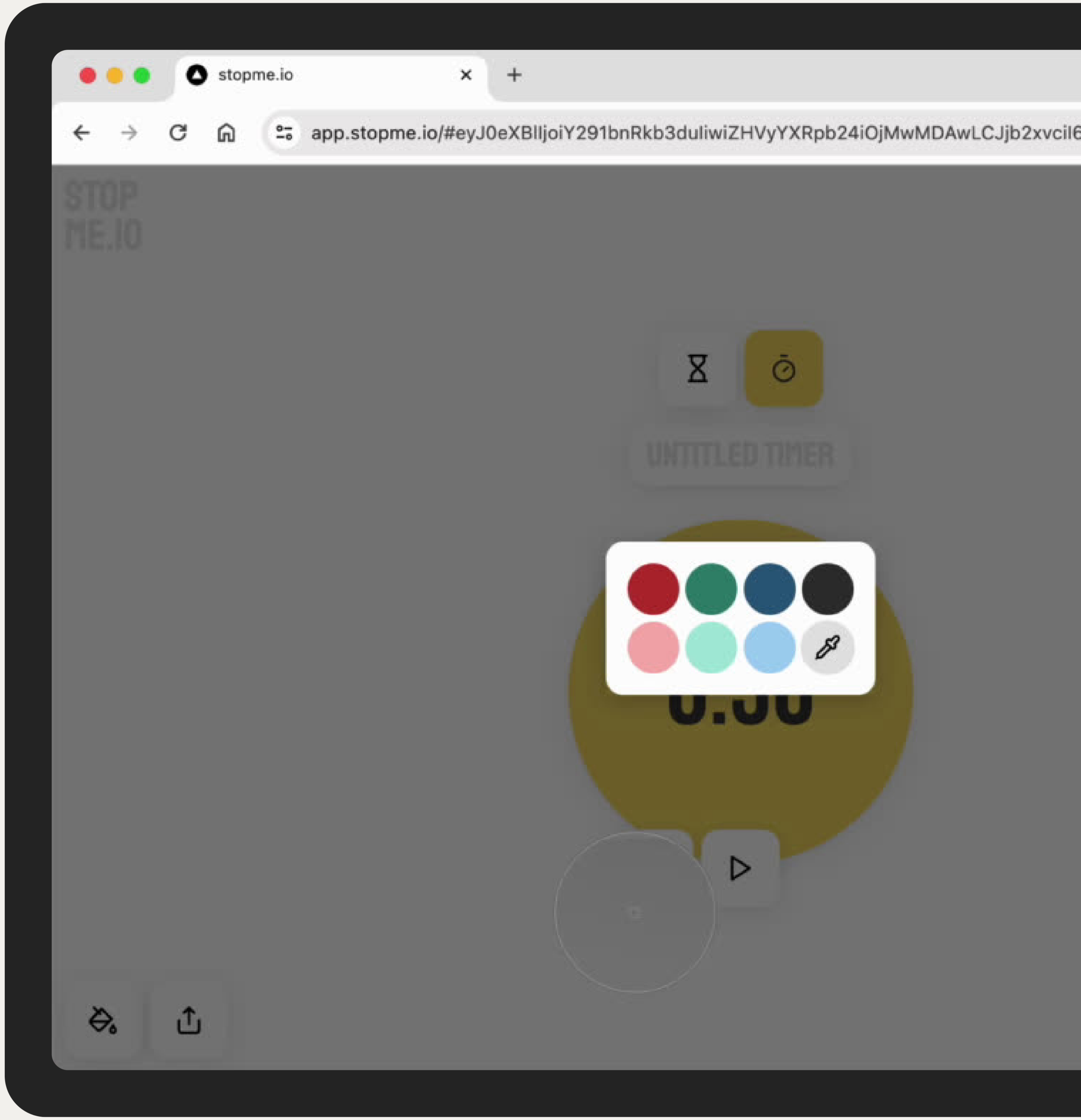


```
const eyeDropper = new EyeDropper();

document.getElementById("btn")
  .addEventListener("click", () => {
    // Needs to be triggered by user action
    eyeDropper.open()
      .then((result) => {
        // Returns the selected color
        // *.sRGBHex
      })
      .catch((e) => {
        // Catches any errors, including
        // when the user cancels selection
      });
  });
```

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

EYEDROPPER API

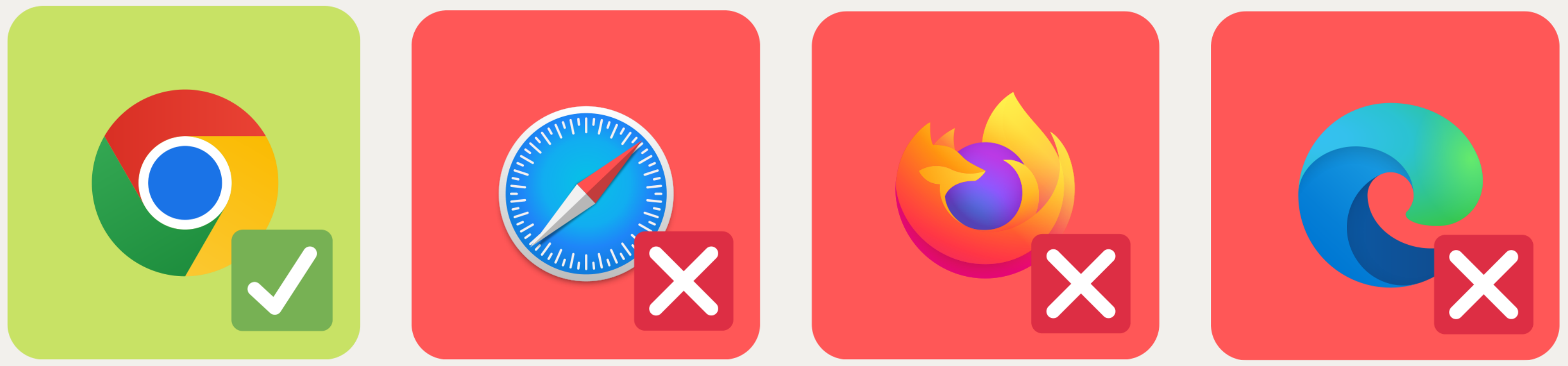
✓ ✗ ✗ ✓


stopme.io
app.stopme.io/#eyJ0eXBlIjoiY291bnRkb3duIiwiZHVyYXRpb24iOjMwMDAwLCJjb2xvcil
STOP ME.IO
UNTITLED TIMER
0:30

ALOMOST AS GOOD AS NATIVE

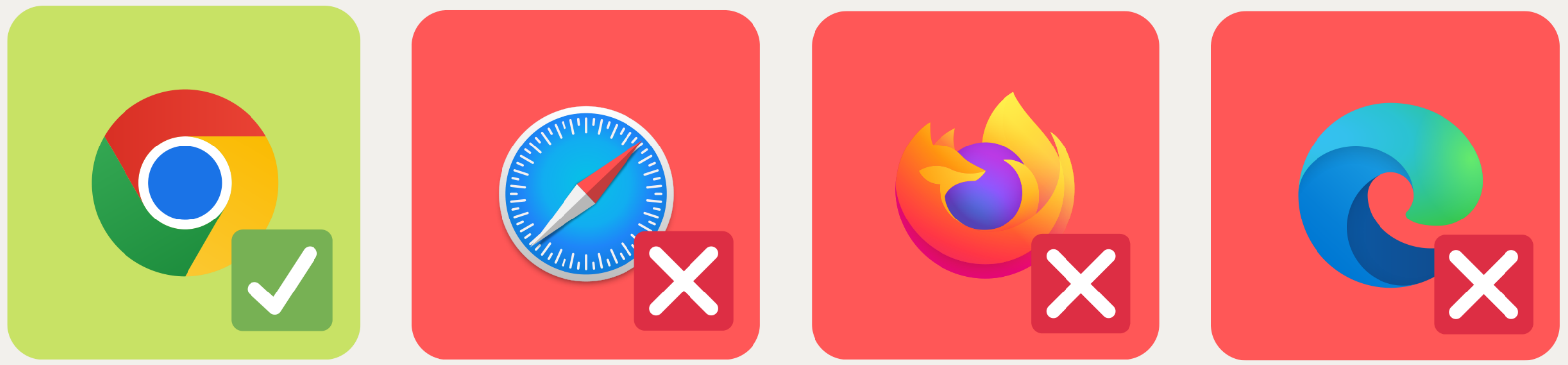
09

CONTACT PICKER API



09

CONTACT PICKER API

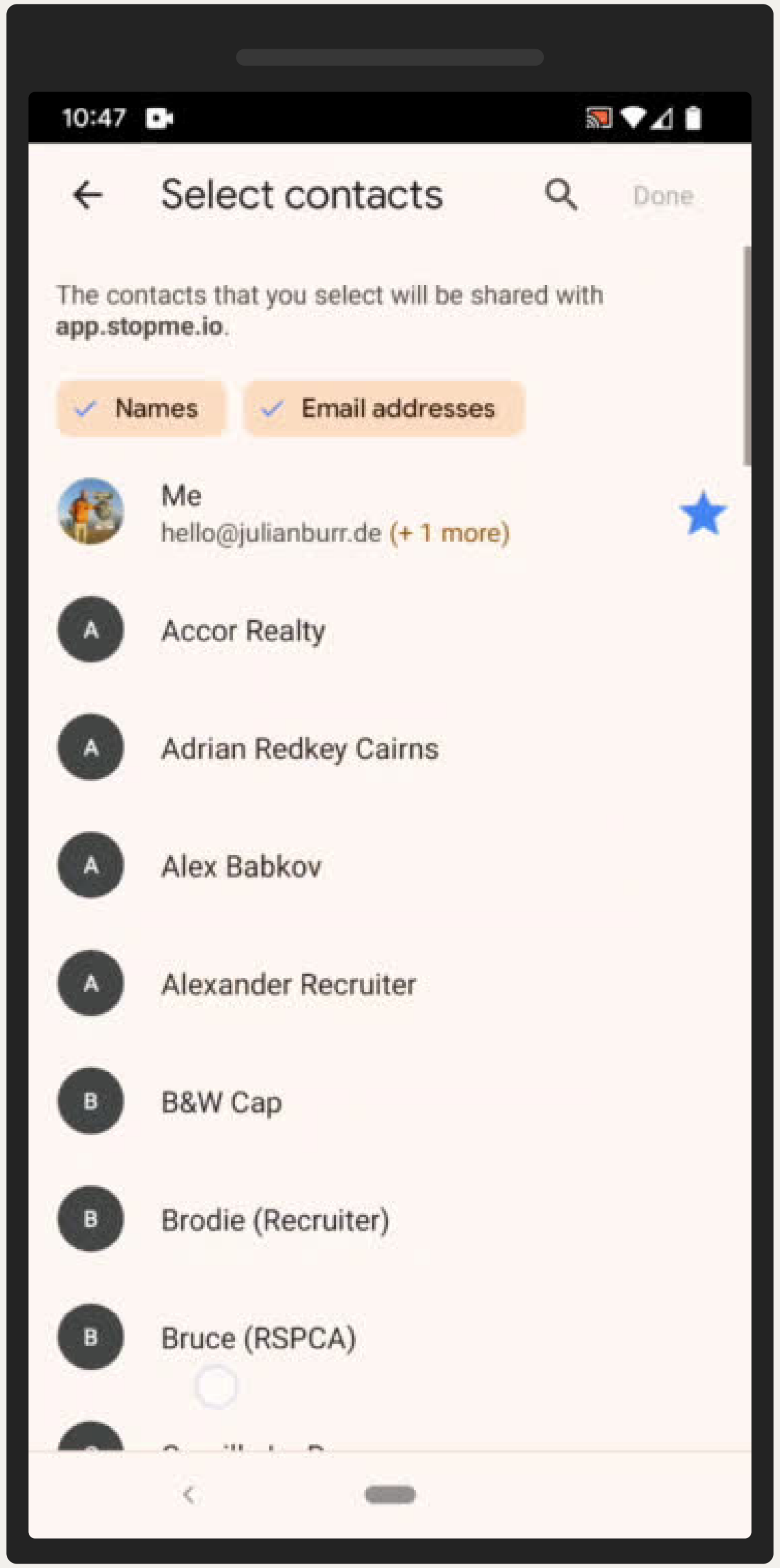


```
const props = ["name", "email", "tel", "icon"];
const opts = { multiple: true };

const contacts = await navigator.contacts
  .select(props, opts);
  .catch((e) => {
    // Handle any errors
  });
```

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

CONTACT PICKER API

✓ ✗ ✗ ✗


10:47

Select contacts

Done

The contacts that you select will be shared with **app.stopme.io**.

✓ Names ✓ Email addresses

Me
hello@julianburr.de (+ 1 more)

Accor Realty

Adrian Redkey Cairns

Alex Babkov

Alexander Recruiter

B&W Cap

Brodie (Recruiter)

Bruce (RSPCA)

10

WEB SHARE API

✓ ✓ ! ✓

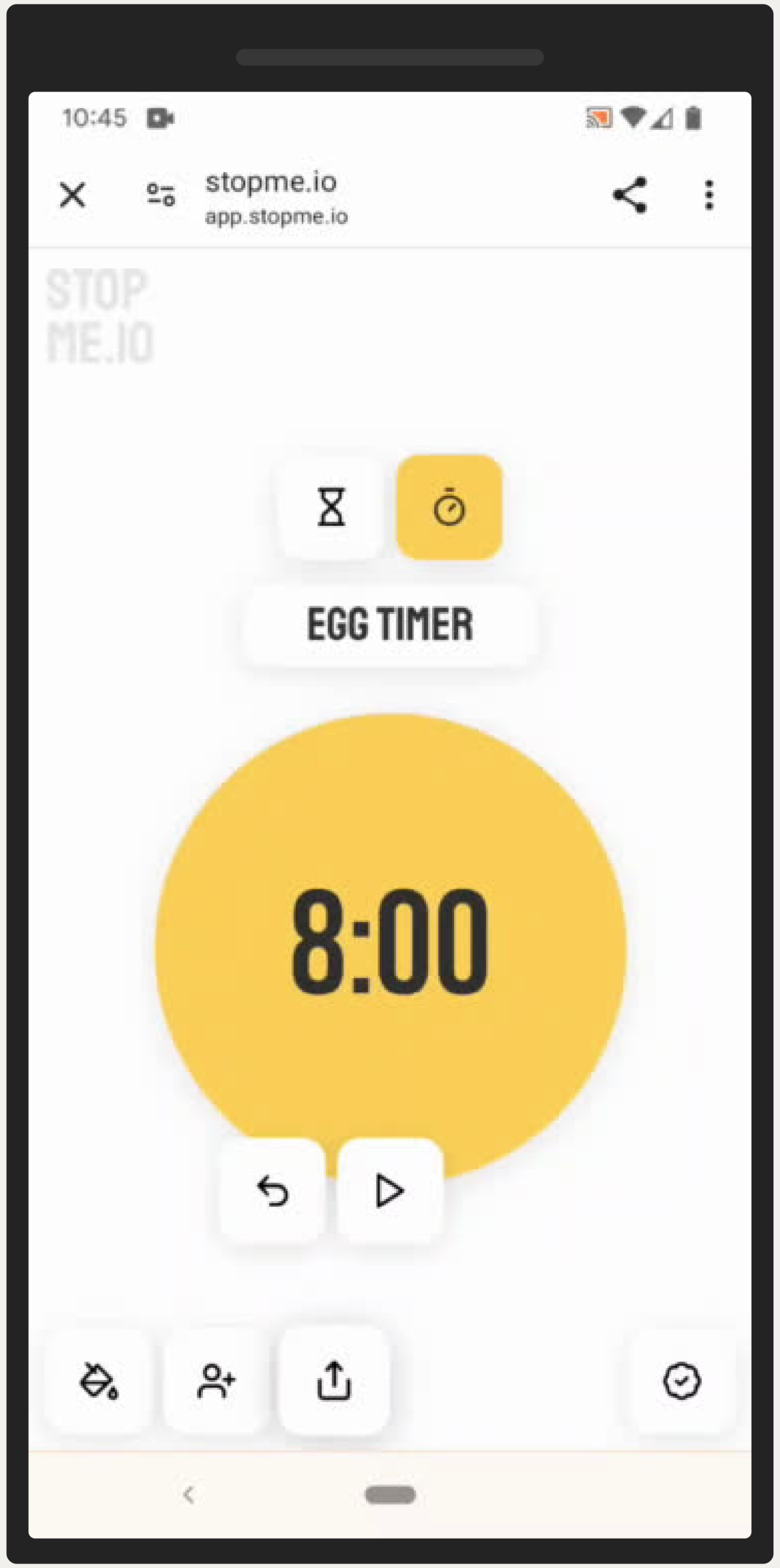
WEB SHARE API

```
const shareData = {
  title: "../NEW",
  text: "Celebrate all things technology",
  url: "https://slashnew.tech"
};

document.getElementById("btn")
  .addEventListener("click", () => {
    navigator.share(shareData)
      .catch((e) => {
        // Handle any errors
      });
  });
```

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

WEB SHARE API

✓ ✓ ! ✓



AUTHENTICATION DONE BETTER

passwordless

11

WEB AUTHN API

11

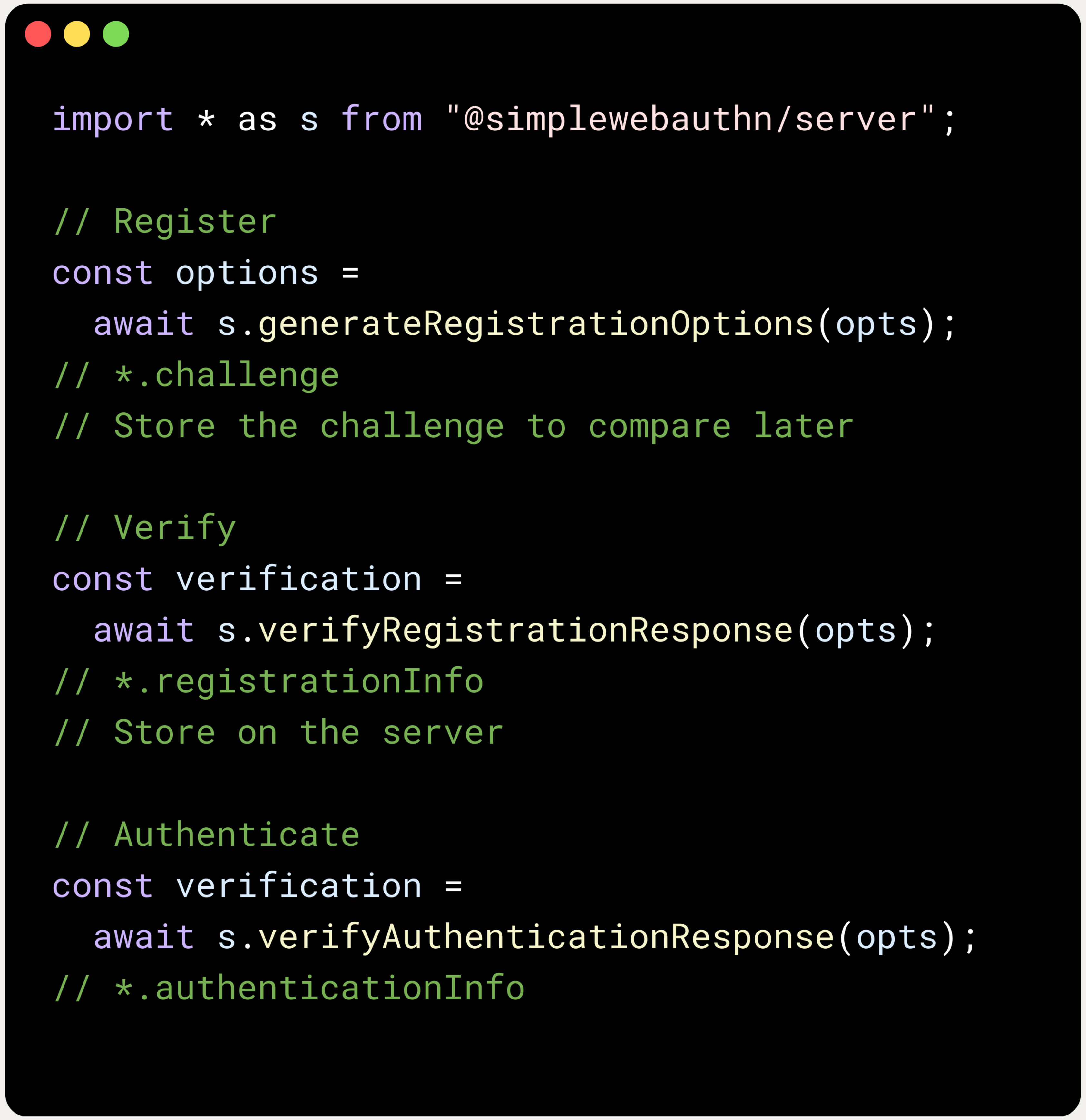
WEB AUTHN API

✓ ✓ ✓ ✓

```
// Create credentials object on the client
// using challenge generated on the server
const registerCredential =
  await navigator.credentials.create({
    publicKey
  });

// Credentials are stored with the user
// identity are sent back and stored on the
// server

// Authenticate again using server challenge
const authCredential =
  await navigator.credentials.get({
    publicKey
  });

// Use stored public key to verify validity
// of auth crendentials
```

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

WEB AUTHN API

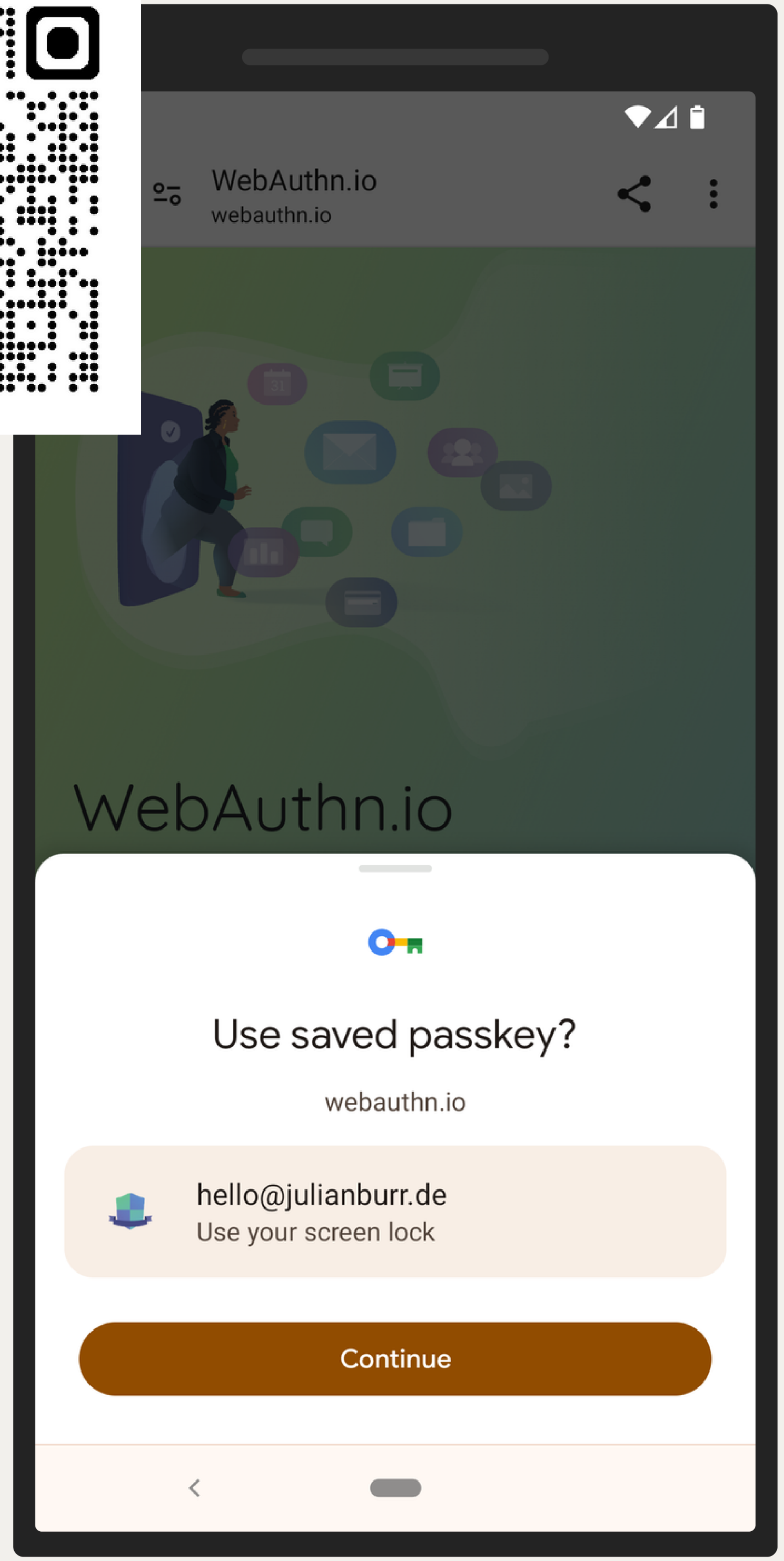


```
import * as s from "@simplewebauthn/server";

// Register
const options =
  await s.generateRegistrationOptions(opts);
// *.challenge
// Store the challenge to compare later

// Verify
const verification =
  await s.verifyRegistrationResponse(opts);
// *.registrationInfo
// Store on the server

// Authenticate
const verification =
  await s.verifyAuthenticationResponse(opts);
// *.authenticationInfo
```

https://github.com/MasterKale/SimpleWebAuthn

WEB AUTHN API

https://webauthn.io/



WebAuthn.io
webauthn.io

WebAuthn.io

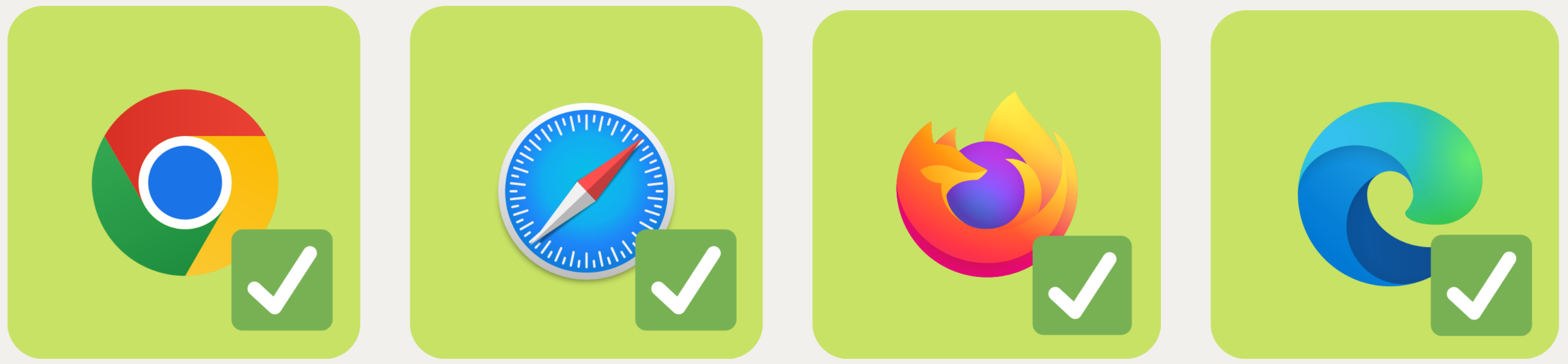
Use saved passkey?

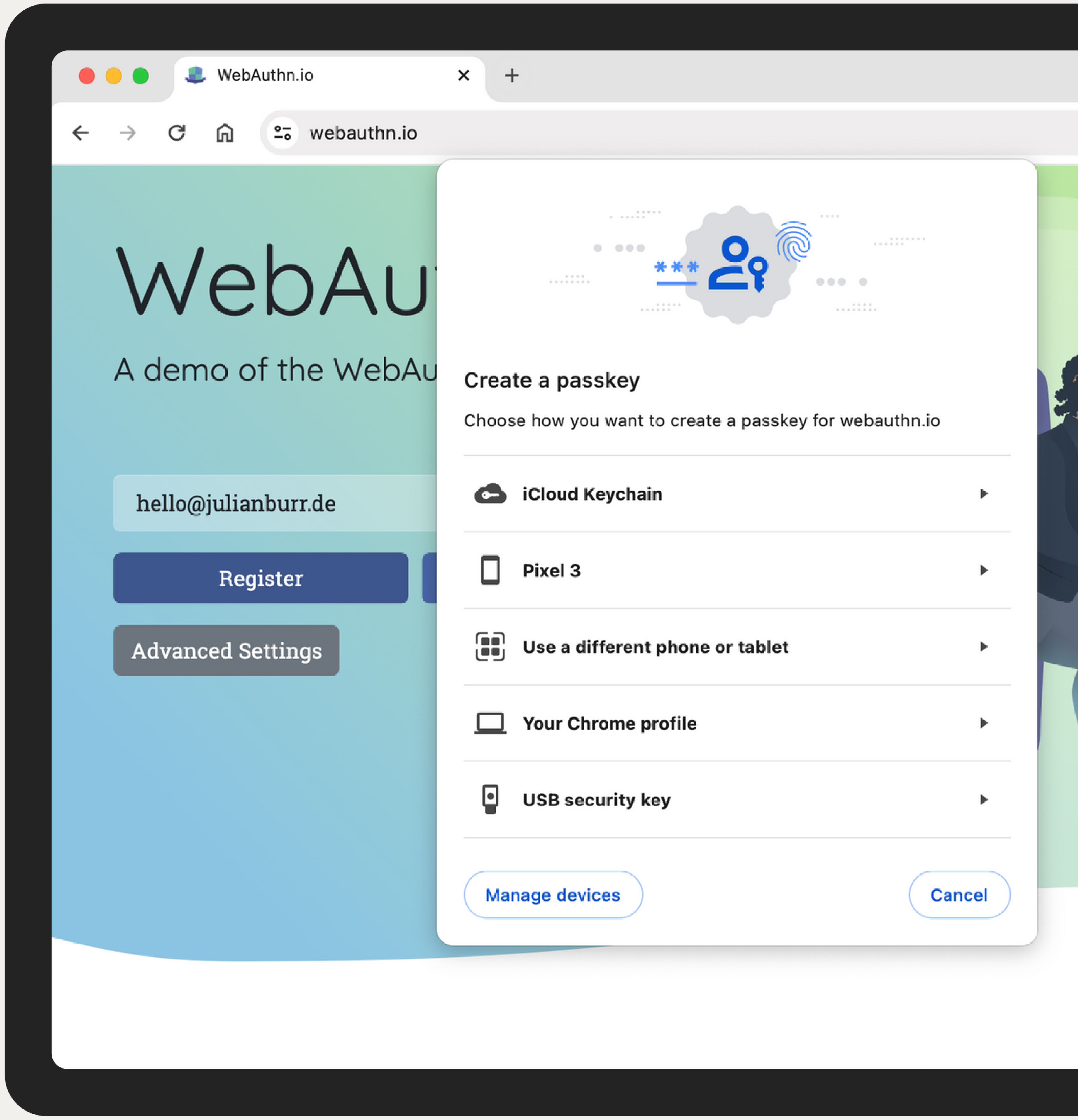
webauthn.io

hello@julianburr.de
Use your screen lock

https://webauthn.io/

WEB AUTHN API





WebAuthn.io

webauthn.io

WebAu

A demo of the WebAu

hello@julianburr.de

Register

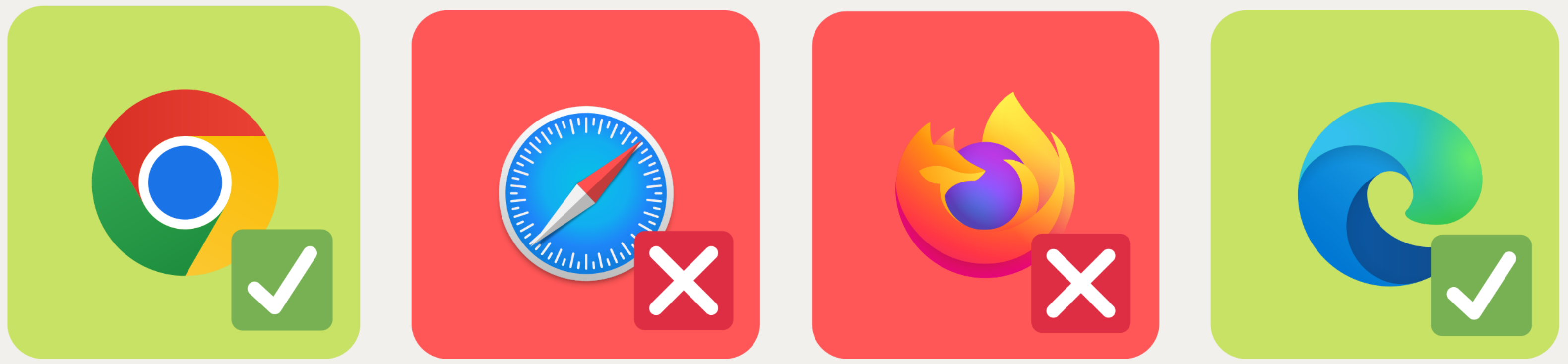
Advanced Settings

Create a passkey

Choose how you want to create a passkey for webauthn.io

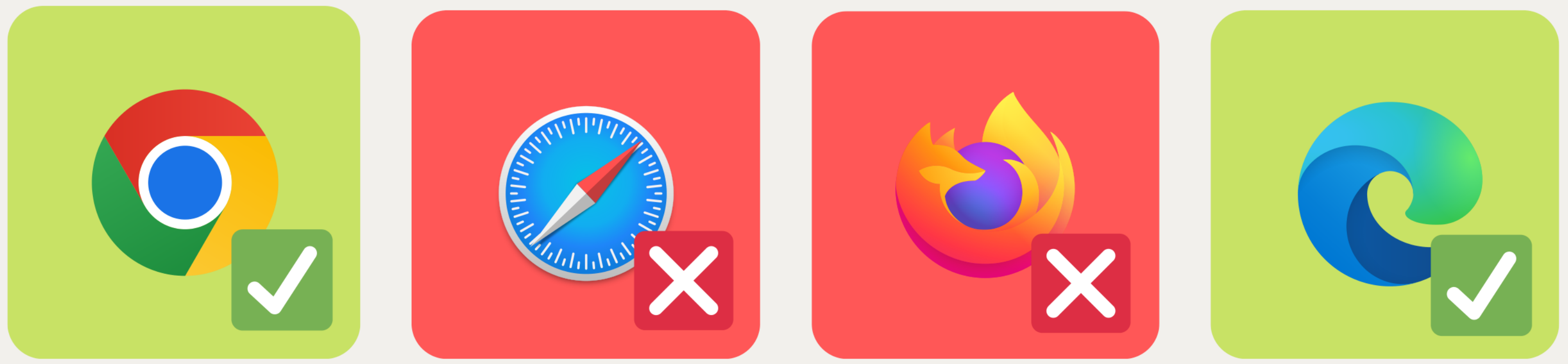
12

WEB OTP API



12

WEB OTP API



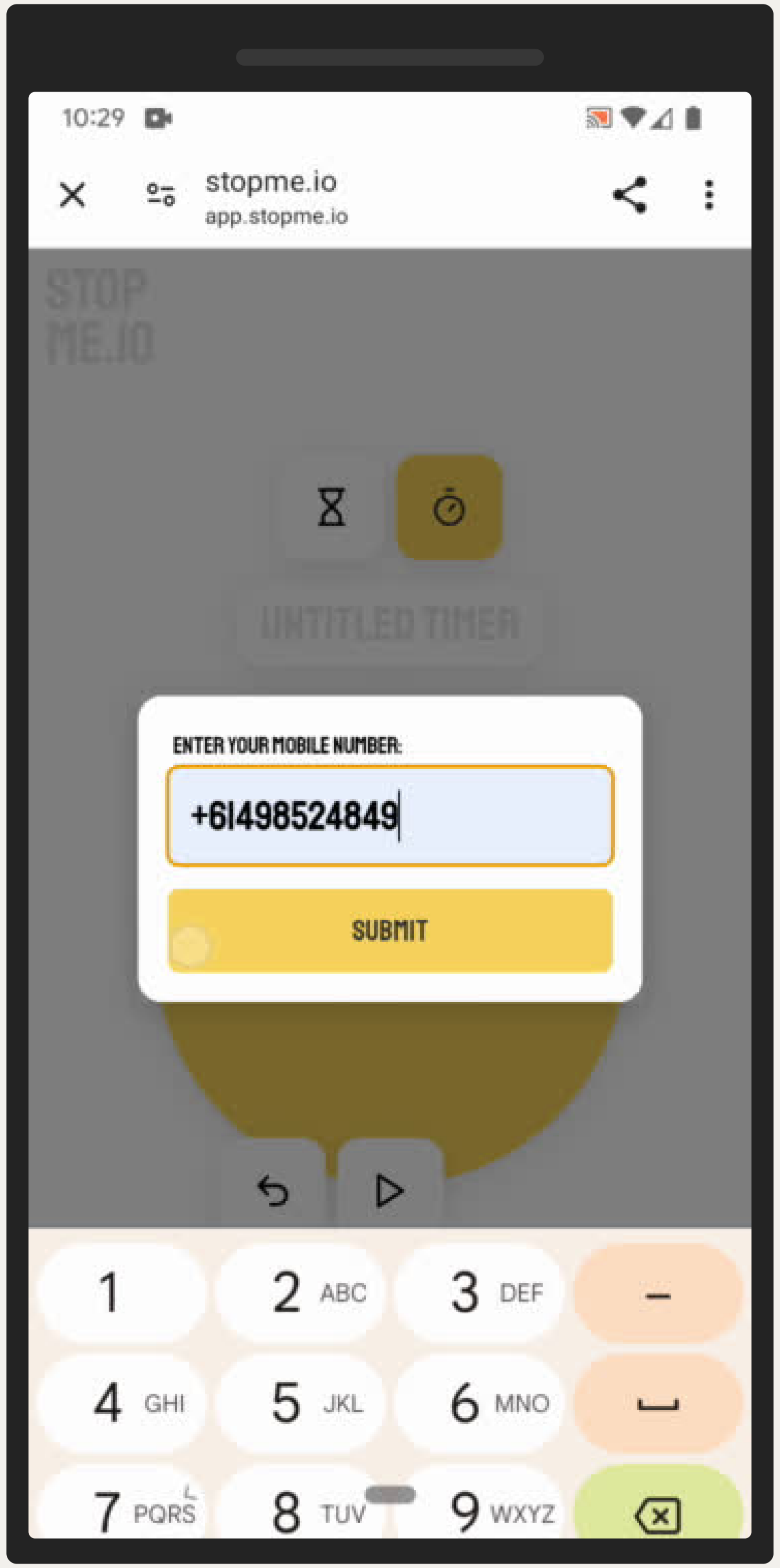
```
navigator.credentials
  .get({
    otp: { transport: ["sms"] },
    signal: ac.signal,
  })
  .then((otp) => {
    // *.code
  })
  .catch((e) => {
    // Handle any errors
  });

// Format of the SMS so it can be processed
//   Your verification code is 123456.\n\n
//   @app.stopme.io #123456
```

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

12

WEB OTP API

🟩 ✓ 🟥 ✗ 🟥 ✗ 🟩 ✓


10:29

stopme.io
app.stopme.io

STOP ME.IO

UNTITLED TIMER

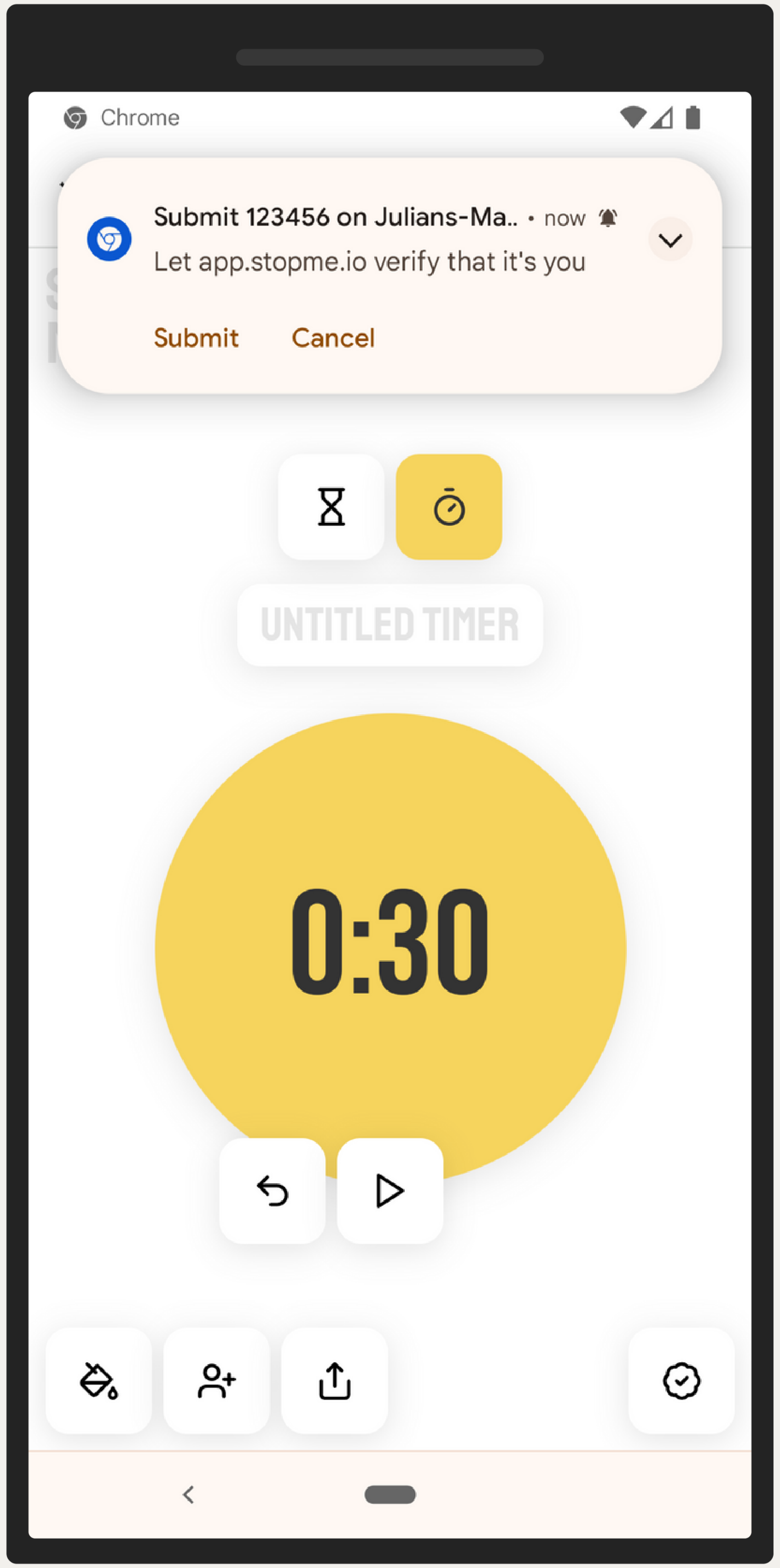
ENTER YOUR MOBILE NUMBER:

+61498524849

SUBMIT

WEB OTP API

✓	✗	✗	✓



Chrome

Submit 123456 on Julians-Ma.. • now

Let app.stopme.io verify that it's you

Submit Cancel

HONOURABLE MENTIONS

DEBUGGING

CONSOLE

PERFORMANCE

MEMORY

DEBUGGING

CONSOLE

PERFORMANCE

MEMORY

```
console.log(...);

console.assert(...);
console.count(...);
console.countReset(...);

console.dir(...);
console.table(...);
console.group(...);
console.groupCollapsed(...);
console.groupEnd(...);

console.time(...);
console.timeEnd(...);

console.trace(...);

console.clear();
```

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

DEBUGGING

CONSOLE

PERFORMANCE

MEMORY

```javascript
const time = performance.now();

performance.mark("start");
performance.mark("end", { detail: { ... } });

performance.measure("login", "start", "end");

const observer =
  new PerformanceObserver((list, obj) => {
    list.getEntries().forEach((entry) => {
      // *.name
      // *.startTime
      // *.duration
      // *.detail
    });
  });
observer.observe({ type: "resource" });
```

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

DEBUGGING

CONSOLE

PERFORMANCE

MEMORY

```
const memory = navigator.deviceMemory;
// Device has at least ${memory}GiB of RAM

const memorySample = await performance
  .measureUserAgentSpecificMemory();
// *.bytes
// *.breakdown[].bytes
// *.breakdown[].attribution
// *.breakdown[].types
```

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

GETTING CREATIVE

MIDI

MEDIA CAPTURE

WEB RTC

CODECS

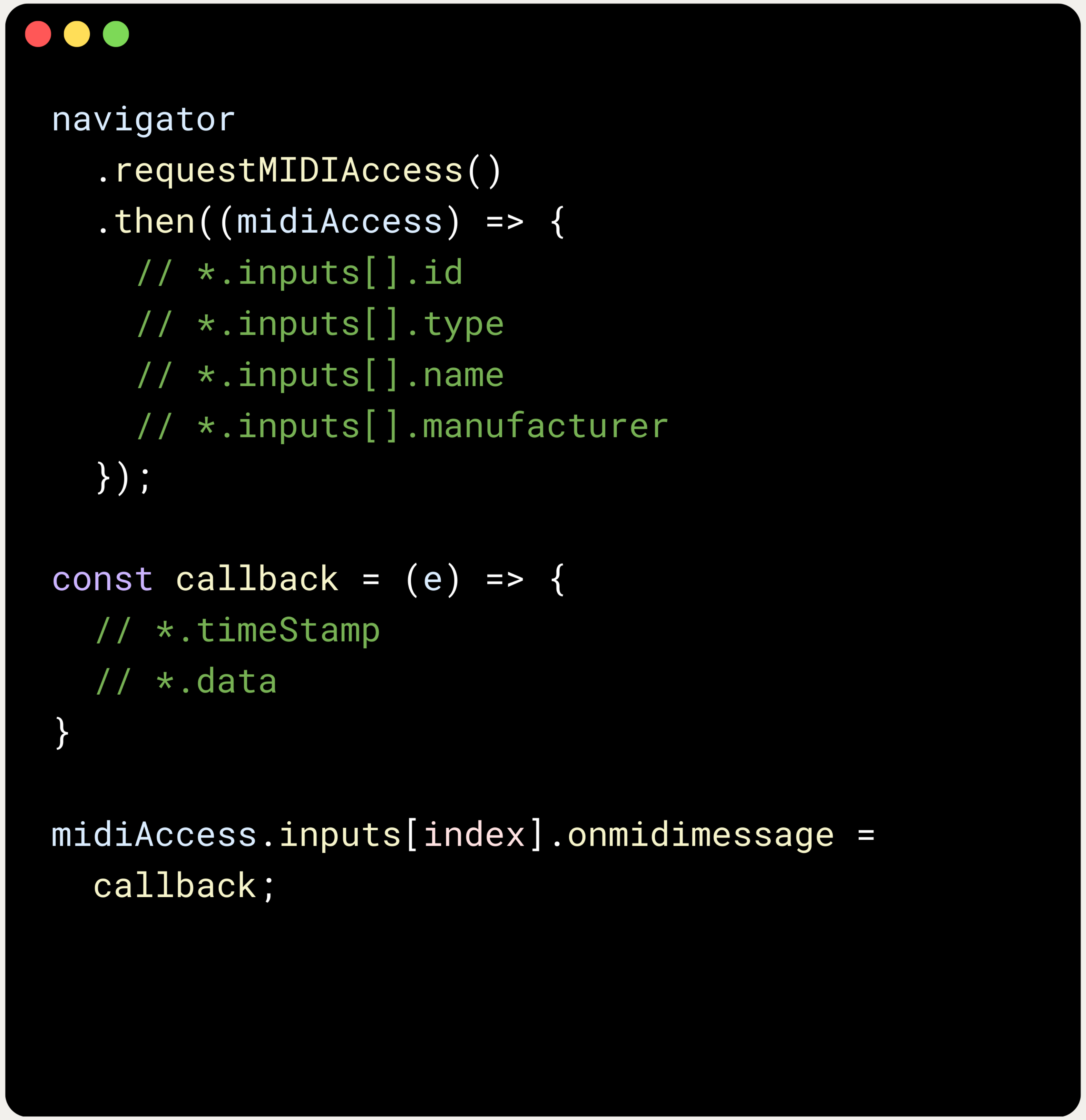
GETTING CREATIVE

MIDI

MEDIA CAPTURE

WEB RTC

CODECS



```
navigator
  .requestMIDIAccess()
  .then((midiAccess) => {
    // *.inputs[].id
    // *.inputs[].type
    // *.inputs[].name
    // *.inputs[].manufacturer
  });

const callback = (e) => {
  // *.timeStamp
  // *.data
}

midiAccess.inputs[index].onmidimessage =
  callback;
```

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

GETTING CREATIVE

MIDI

MEDIA CAPTURE

WEB RTC

CODECS

```
// Capture media from input devices
const stream = await navigator.mediaDevices
  .getUserMedia({ video: true, audio: false });

const videoEl =
  document.getElementById("video");
videoEl.srcObject = stream;
videoEl.play();

// Capture video from element or canvas
const el =
  document.getElementById("record-me");
const stream = await el.captureStream();
el.play();
```

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

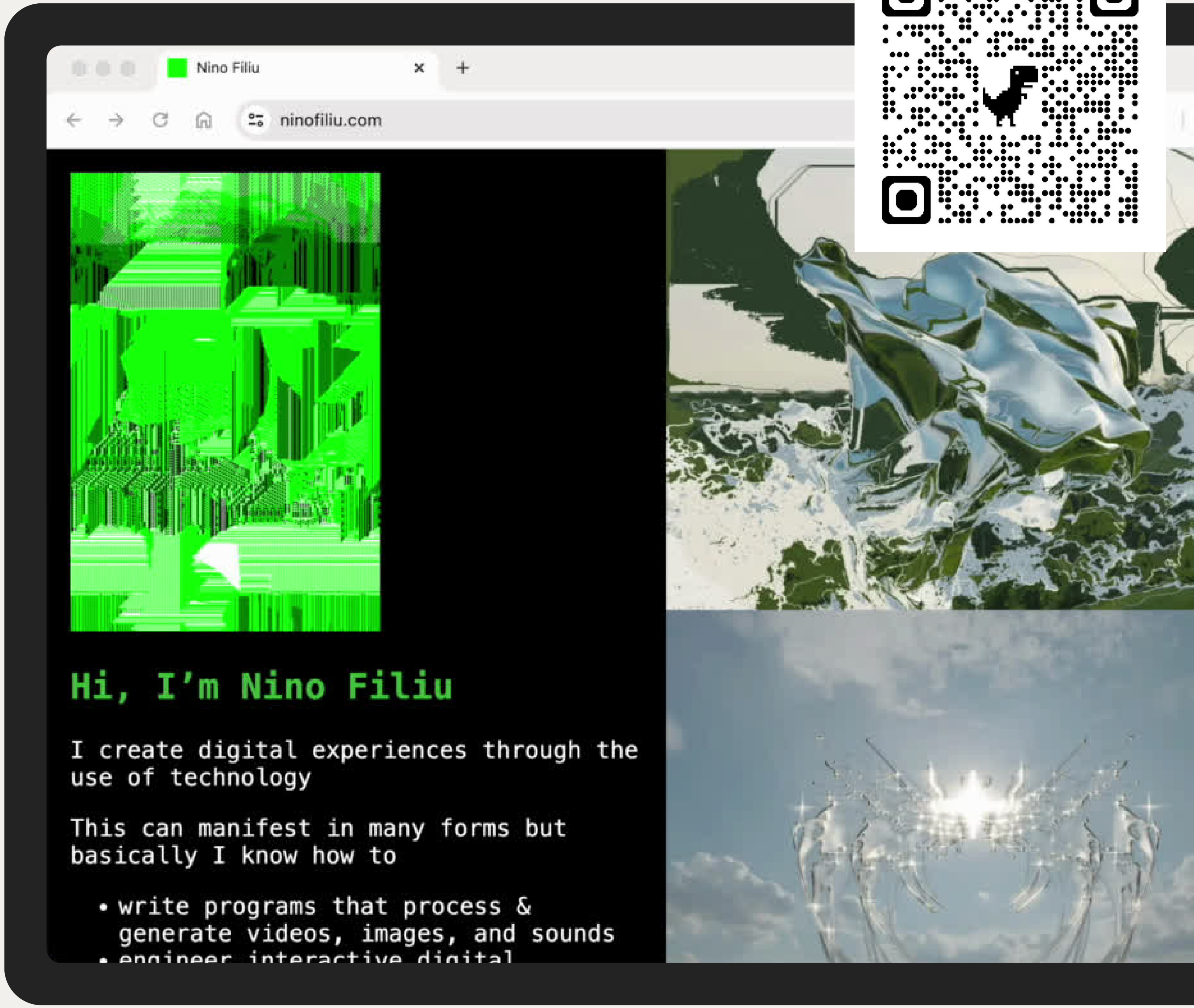
GETTING CREATIVE

MIDI

MEDIA CAPTURE

WEB RTC

CODECS



Hi, I'm Nino Filiu

I create digital experiences through the use of technology

This can manifest in many forms but basically I know how to

- write programs that process & generate videos, images, and sounds
- engineer interactive digital

SHARING CONTENT

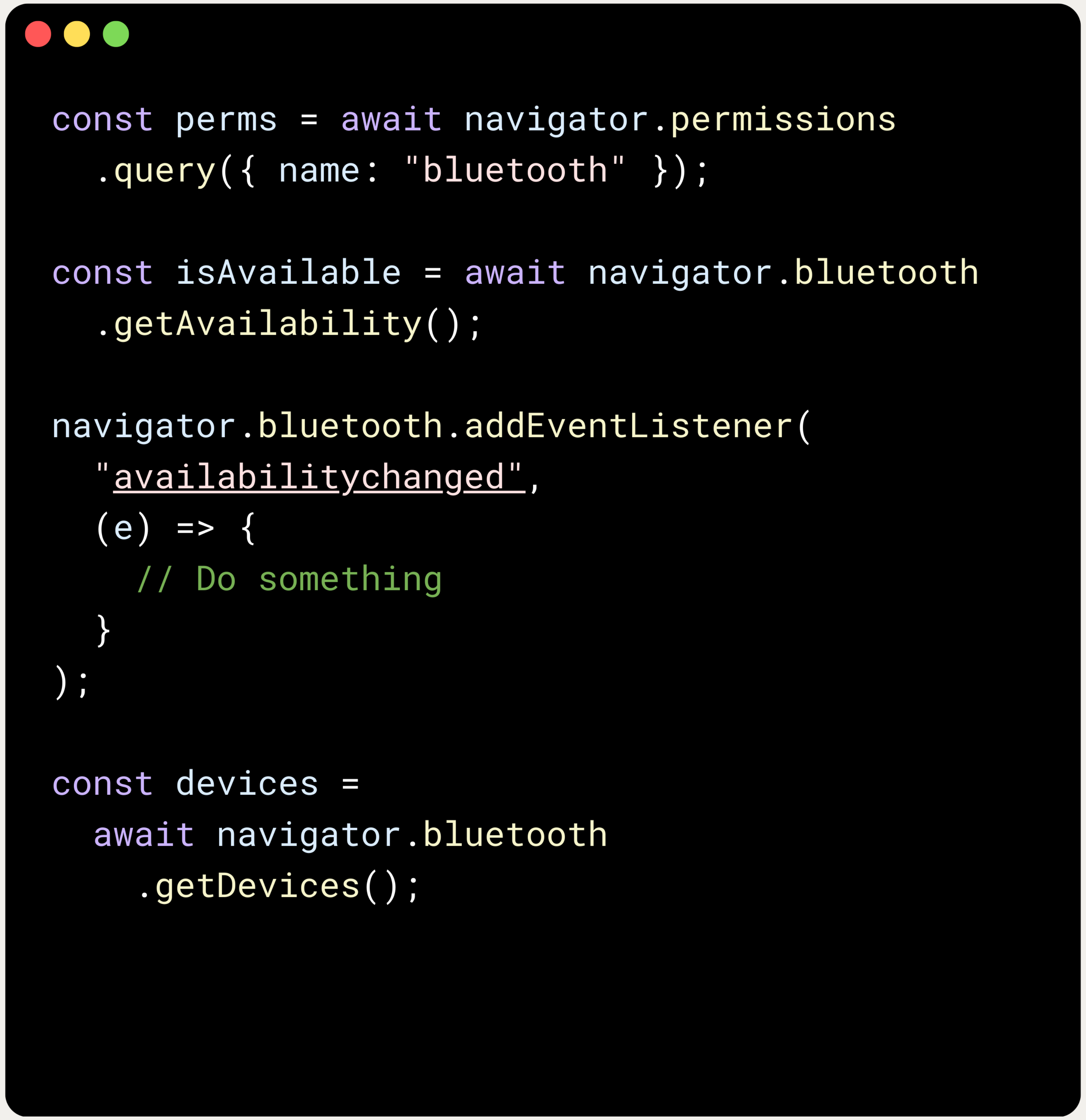
BLUETOOTH

USB API

SHARING CONTENT

BLUETOOTH

USB API



```
const perms = await navigator.permissions
  .query({ name: "bluetooth" });

const isAvailable = await navigator.bluetooth
  .getAvailability();

navigator.bluetooth.addEventListener(
  "availabilitychanged",
  (e) => {
    // Do something
  }
);

const devices =
  await navigator.bluetooth
    .getDevices();
```

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

SHARING CONTENT

BLUETOOTH

USB API

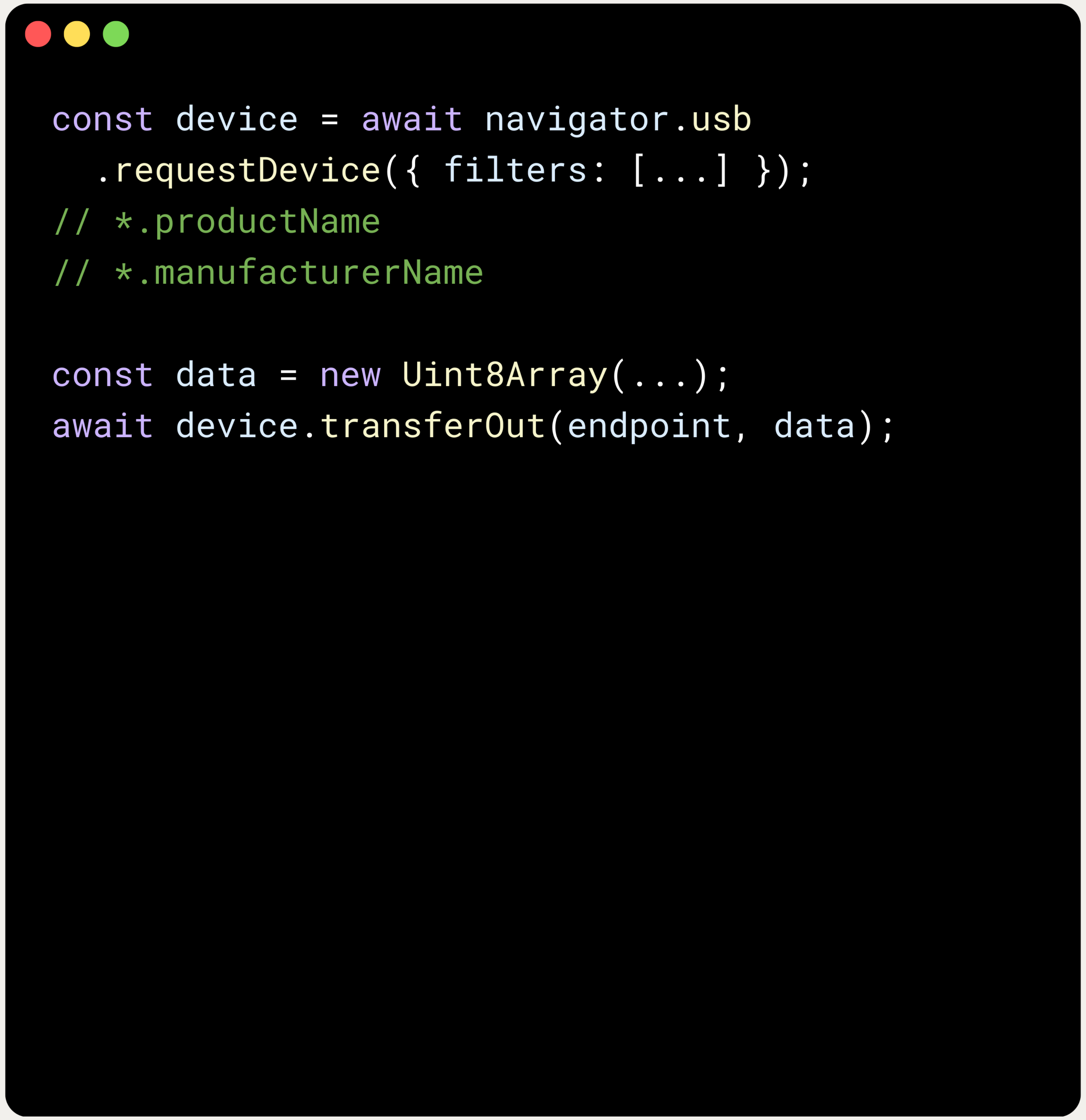
```
const device = await navigator.bluetooth
  .requestDevice({ filters: [...] });
// *.name

const server = await device.gatt.connect();
const service = await server
  .getPrimaryService("health_thermometer");

const char = await service
  .getCharacteristic("measurement_interval");
```

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

SHARING CONTENT

BLUETOOTH

USB API



```
const device = await navigator.usb
  .requestDevice({ filters: [...] });
// *.productName
// *.manufacturerName

const data = new Uint8Array(...);
await device.transferOut(endpoint, data);
```

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

THE NEXT GENERATION

INTL API

TEMPORAL*

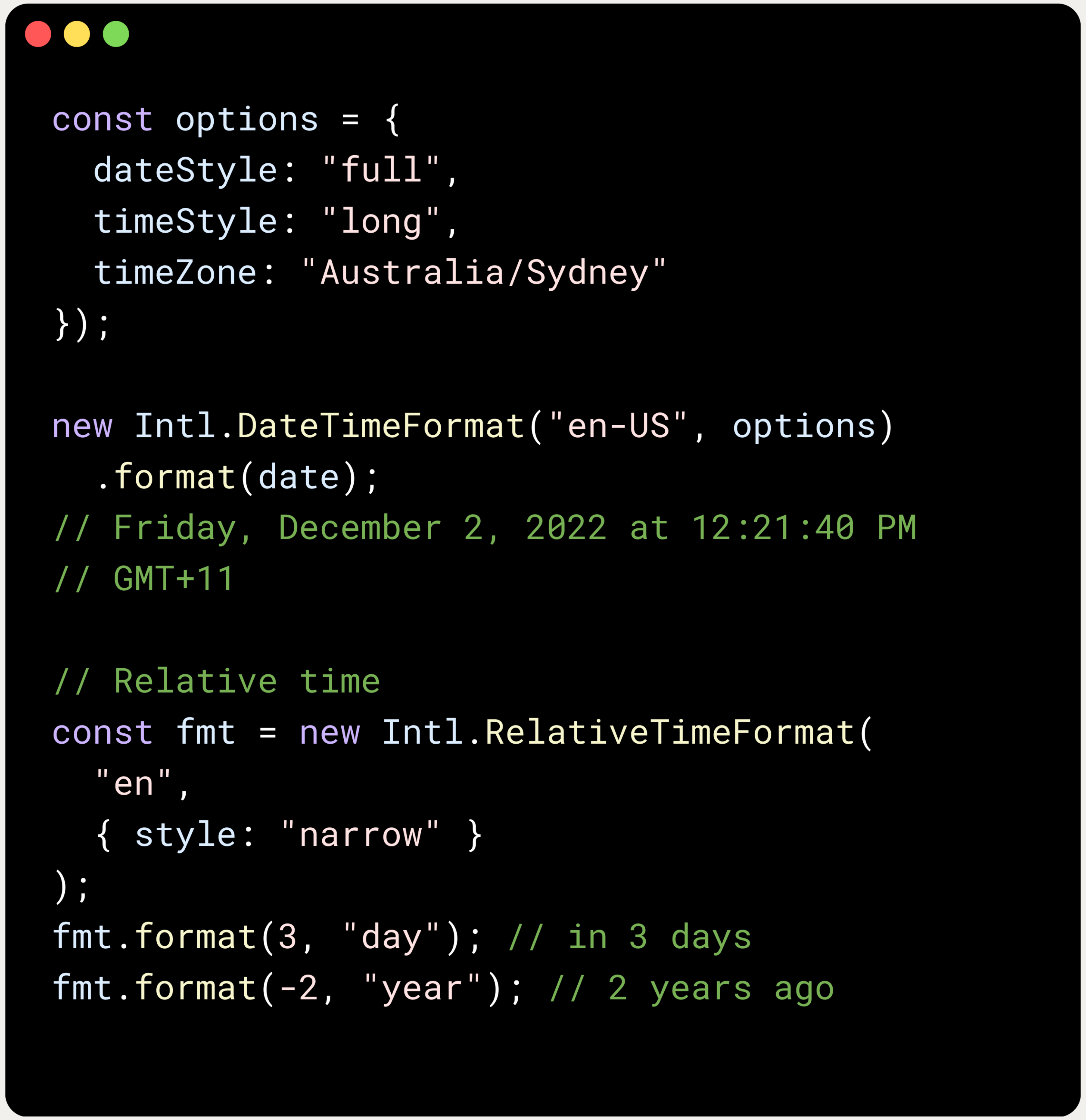
NAVIGATION API

THE NEXT GENERATION

INTL API

TEMPORAL*

NAVIGATION API



```
const options = {
  dateStyle: "full",
  timeStyle: "long",
  timeZone: "Australia/Sydney"
});

new Intl.DateTimeFormat("en-US", options)
  .format(date);
// Friday, December 2, 2022 at 12:21:40 PM
// GMT+11

// Relative time
const fmt = new Intl.RelativeTimeFormat(
  "en",
  { style: "narrow" }
);
fmt.format(3, "day"); // in 3 days
fmt.format(-2, "year"); // 2 years ago
```

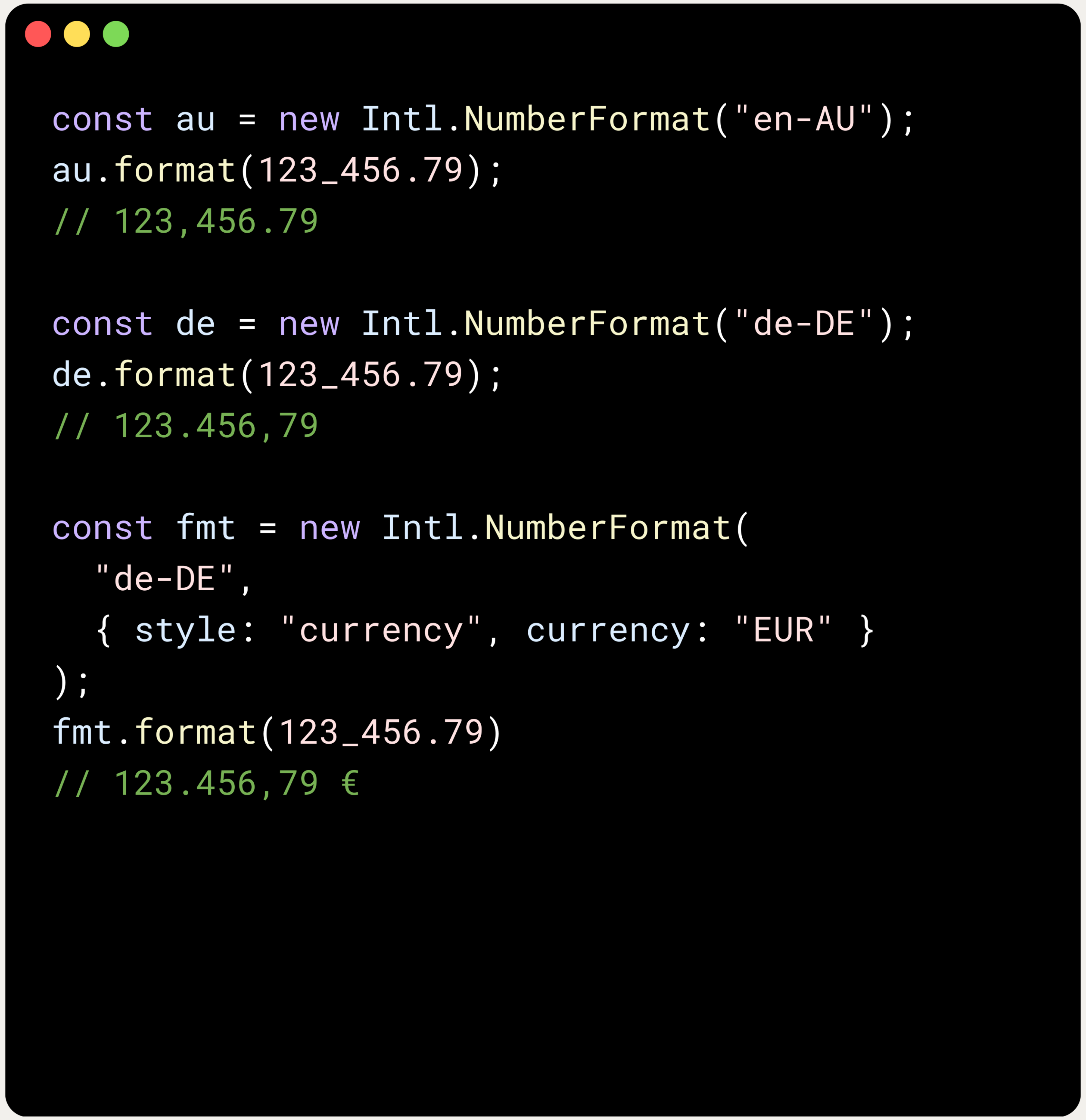
https://developer.mozilla.org/en-US/docs/Web/JavaScript/Reference/Global_Objects/Intl

THE NEXT GENERATION

INTL API

TEMPORAL*

NAVIGATION API



```
const au = new Intl.NumberFormat("en-AU");
au.format(123_456.79);
// 123,456.79

const de = new Intl.NumberFormat("de-DE");
de.format(123_456.79);
// 123.456,79

const fmt = new Intl.NumberFormat(
  "de-DE",
  { style: "currency", currency: "EUR" }
);
fmt.format(123_456.79)
// 123.456,79 €
```

https://developer.mozilla.org/en-US/docs/Web/JavaScript/Reference/Global_Objects/Intl

THE NEXT GENERATION

INTL API

TEMPORAL*

NAVIGATION API

```
// Get exact current system time
Temporal.Now.instant();

// Get time zone
Temporal.Now.timeZoneId();

// Useful utilities
const date = Temporal.PlainDate.from(dateStr);
// *.year
// *.inLeapYear
// *.toString()
// ...

// Manipulate time
date.add({ hours: 1 });
```

https://tc39.es/proposal-temporal/docs/

THE NEXT GENERATION

INTL API

TEMPORAL*

NAVIGATION API

```
// Promises for the win (finally!)
await navigation.reload({ info, state });

// Navigate around
await navigation.navigate(url, options);
await navigation.back(options);
await navigation.forward(options);
await navigation.traverseTo(key, options);

// Events
navigation
  .addEventListener("currententrychange");
  .addEventListener("navigate");
  .addEventListener("navigatesuccess");
  .addEventListener("navigateerror");
```

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

THE NEXT GENERATION

URL PATTERNS

POPOVER

VIEW TRANSITION

```
const p = new URLPattern({ pathname: "/foo" });
p.test("https://example.com/books")); // true

const p =
  new URLPattern({ pathname: "/books/:id" });
const match =
  p.exec("https://example.com/books/123");
// *.pathname.groups.id = 123

const p =
  new URLPattern(
    "/books/:id(\\d+)",
    "https://example.com"
  );
```

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

THE NEXT GENERATION

URL PATTERNS

POPOVER

VIEW TRANSITION

```
<button popovertarget="mypopover">
  Toggle the popover
</button>

<div id="mypopover" popover>
  Popover content
</div>

document.addEventListener("keydown", (e) => {
  if (e.key === "h") {
    mypopover.togglePopover();
  }
});
```

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

THE NEXT GENERATION

URL PATTERNS

POPOVER

VIEW TRANSITION

```javascript
function updateView(e) {
  e.preventDefault();
  const details =
    e.target.closest("details");

  if (!document.startViewTransition) {
    // Fallback for older browsers
    details.toggleAttribute("open")
  }

  document.startViewTransition(() =>
    details.toggleAttribute("open")
  );
}
```

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

THANKS!

@JBURR90 / JULIAN BURR

HTTPS://WWW.JULIANBURR.DE/SLASHNEW-2024-SLIDES.PDF