TouchBar

Class: TouchBar

> *Create TouchBar layouts for native macOS applications*

Process: Main

`new TouchBar(options)`

- `options` Object
  - `items` (TouchBarButton | TouchBarColorPicker | TouchBarGroup | TouchBarLabel | TouchBarPopover | TouchBarScrubber | TouchBarSegmentedControl | TouchBarSlider | TouchBarSpacer)[] (optional)
  - `escapeItem` (TouchBarButton | TouchBarColorPicker | TouchBarGroup | TouchBarLabel | TouchBarPopover | TouchBarScrubber | TouchBarSegmentedControl | TouchBarSlider | TouchBarSpacer | null) (optional)

Creates a new touch bar with the specified items. Use `BrowserWindow.setTouchBar` to add the `TouchBar` to a window.

Note: The TouchBar API is currently experimental and may change or be removed in future Electron releases.

Tip: If you don't have a MacBook with Touch Bar, you can use Touch Bar Simulator to test Touch Bar usage in your app.

Static Properties

`TouchBarButton`

A `typeof TouchBarButton` reference to the `TouchBarButton` class.

`TouchBarColorPicker`

A `typeof TouchBarColorPicker` reference to the `TouchBarColorPicker` class.

`TouchBarGroup`

A `typeof TouchBarGroup` reference to the `TouchBarGroup` class.

`TouchBarLabel`

A `typeof TouchBarLabel` reference to the `TouchBarLabel` class.

`TouchBarPopover`

A `typeof TouchBarPopover` reference to the `TouchBarPopover` class.

`TouchBarScrubber`

A `typeof TouchBarScrubber` reference to the `TouchBarScrubber` class.

`TouchBarSegmentedControl`

A `typeof TouchBarSegmentedControl` reference to the `TouchBarSegmentedControl` class.

`TouchBarSlider`

A `typeof TouchBarSlider` reference to the `TouchBarSlider` class.

`TouchBarSpacer`

A `typeof TouchBarSpacer` reference to the `TouchBarSpacer` class.

`TouchBarOtherItemsProxy`

A `typeof TouchBarOtherItemsProxy` reference to the `TouchBarOtherItemsProxy` class.

Instance Properties

The following properties are available on instances of `TouchBar`:

`touchBar.escapeItem`

A `TouchBarItem` that will replace the "esc" button on the touch bar when set. Setting to `null` restores the default "esc" button. Changing this value immediately updates the escape item in the touch bar.

Examples

Below is an example of a simple slot machine touch bar game with a button and some labels.

```
const { app, BrowserWindow, TouchBar } = require('electron')

const { TouchBarLabel, TouchBarButton, TouchBarSpacer } = TouchBar

let spinning = false

// Reel labels
const reel1 = new TouchBarLabel()
const reel2 = new TouchBarLabel()
const reel3 = new TouchBarLabel()

// Spin result label
const result = new TouchBarLabel()

// Spin button
const spin = new TouchBarButton({
  label: '🎰 Spin',
  backgroundColor: '#7851A9',
  click: () => {
    // Ignore clicks if already spinning
    if (spinning) {
      return
    }

    spinning = true
    result.label = ''

    let timeout = 10
```

```
    const spinLength = 4 * 1000 // 4 seconds
    const startTime = Date.now()

    const spinReels = () => {
      updateReels()

      if ((Date.now() - startTime) >= spinLength) {
        finishSpin()
      } else {
        // Slow down a bit on each spin
        timeout *= 1.1
        setTimeout(spinReels, timeout)
      }
    }

    spinReels()
  }
})

const getRandomValue = () => {
  const values = ['🍒', '💎', '7️⃣', '🍊', '🔔', '⭐', '🍇', '🍀']
  return values[Math.floor(Math.random() * values.length)]
}

const updateReels = () => {
  reel1.label = getRandomValue()
  reel2.label = getRandomValue()
  reel3.label = getRandomValue()
}

const finishSpin = () => {
  const uniqueValues = new Set([reel1.label, reel2.label, reel3.label]).size
  if (uniqueValues === 1) {
    // All 3 values are the same
    result.label = '💰 Jackpot!'
    result.textColor = '#FDFF00'
  } else if (uniqueValues === 2) {
    // 2 values are the same
    result.label = '😃 Winner!'
    result.textColor = '#FDFF00'
  } else {
    // No values are the same
    result.label = '☹️ Spin Again'
    result.textColor = null
  }
  spinning = false
}

const touchBar = new TouchBar({
  items: [
    spin,
    new TouchBarSpacer({ size: 'large' }),
```

```
    reel1,
    new TouchBarSpacer({ size: 'small' }),
    reel2,
    new TouchBarSpacer({ size: 'small' }),
    reel3,
    new TouchBarSpacer({ size: 'large' }),
    result
  ]
})

let window

app.whenReady().then(() => {
  window = new BrowserWindow({
    frame: false,
    titleBarStyle: 'hiddenInset',
    width: 200,
    height: 200,
    backgroundColor: '#000'
  })
  window.loadURL('about:blank')
  window.setTouchBar(touchBar)
})
```

Running the above example

To run the example above, you'll need to (assuming you've got a terminal open in the directory you want to run the example):

1. Save the above file to your computer as `touchbar.js`
2. Install Electron via `npm install electron`
3. Run the example inside Electron: `./node_modules/.bin/electron touchbar.js`

You should then see a new Electron window and the app running in your touch bar (or touch bar emulator).