

How to use Intl API in JavaScript

Reach a wider audience by directing content to any language using the Intl API. Here's how to use the Intl API in JavaScript.

The Intl API simplifies formatting and manipulating text, numbers, dates, and internationalized currencies. It allows you to handle different data formats by language.

This API solves the difficulty of formatting data for different cultures and languages. It makes it easier to format numbers with currency or date symbols for specific regions.

Using the Intl API, you can create web applications that are accessible and easy to use for people from different regions and cultures.

Get user language

The JavaScript constructors provided by the International API define the language they will use to format dates, text, numbers, etc., according to a familiar pattern. Each constructor takes a language and an options object as arguments. Using these arguments, the constructor matches the requested language(s) with the currently supported languages.

To get the user's language, you can use the `navigator.language` property . This property returns a string, representing the browser's language version.

The value of the `navigator.language` attribute is the BCP 47 language tag. It contains a language and area code (optional) and other sub-tags. For example, 'en-US' represents English spoken in the US.

You can also use the `navigator.languages` ??property to get an array of user-favorite languages, and then sort them in order of preference. The first item in the array represents the main language configuration.

You can create a simple JavaScript function that helps you get the user's language. Here is the sample code:

```
const getUserLocale = () => { ??  
if (navigator.languages && navigator.languages.length) { ????  
return navigator.languages[0]; ??} ??  
return navigator.language; }; console.log(getUserLocale());
```

The `getUserLocale` function returns the first element of the `navigator.languages` ??property if it is available. Otherwise, it will fall back to the `navigator.language` property, which represents the user's preferred language in older browsers.

Date format for different languages

Format dates using the Intl API, you can use the Intl.DateTimeFormat() constructor. It takes two arguments: a language string and an options object.

Options objects can contain properties that control date formatting.

Some popular choices include:

1. Weekday : Format the day of the week.
2. Year : Year format.
3. Month : Month format.
4. Day : Date format.

Example how to format date using Intl.DateTimeFormat() constructor:

```
const date = Date.now() const locale = getUserLocale(); const options = {
  ??weekday: "long", ??year: "numeric", ??month: "long", ??
  day: "numeric", }; const formatter = new Intl.DateTimeFormat(locale, options); /
```

This code sets up a formatter object by passing the user's locale to Intl.DateTimeFormat() , along with a group of choices. It then uses the formatter to turn the current date into a string. The options object contains properties that control the format of the date.

Above is how to use Intl API in JavaScript . Hope the article is useful to you.

You finished reading the article "**How to use Intl API in JavaScript**" edited by the [TipsMake](#) team. We hope this article has provided you with many useful tech tips and tricks. You can search for similar articles on tips and guides. Thank you for reading and for following us regularly.