



Localization Module

☰ PAGE CONTENTS

[Methods](#)

[Usage](#)

The Localization module provides information about the locale based on device settings.

PRO TIP

If you want your app to display strings in the user's current language, we recommend the [localized-strings library](#) . Use `LocalizationModule.getLanguageTag()` to get the current language, and then use `localized-strings` to display strings in that language.

Methods

`getCalendar()`

Description: Returns the user-preferred calendar format.

Return Type: String

Method Usage Example: `LocalizationModule.getCalendar()`

Platforms Supported: All except PS4 and webOS. For these platforms, the return value is `Gregorian` .

`getCountryCode()`

Description: Returns the current country code (based on the device's locale setting, not on its geographical location); for example, `CA` or `US` .

Return Type: String

Method Usage Example: `LocalizationModule.getCountryCode()`

Platforms Supported: All

`getCurrencies()`

Description: Returns the user-preferred currency codes, in order of preference; for example: `["CAD", "USD"]` .

Return Type: String

Method Usage Example: `LocalizationModule.getCurrencies()`

Platforms Supported: All except Tizen, PS4, and webOS. On these platforms, which don't provide a currency setting, this method returns an empty array.

NOTE

On Apple platforms, certain languages, such as Hebrew, have no associated currency. To determine the preferred currency in this case, the region setting is also used. If the user's primary preferred language is Hebrew and the region is Russia, but the list of preferred languages also includes English (Canada), `getCurrencies()` returns the following array: `["CAD", "RUB"]`

getDateFormat()

Description: Returns the date format based on device settings. If the device's locale setting is US, for example, the return value is `mm-dd-yyyy`; for CA, the return value is `yyyy-mm-dd`.

Return Type: String

Method Usage Example: `LocalizationModule.getDateFormat()`

Platforms Supported: Android. All other platforms return the default date format, `yyyy-mm-dd`.

getIsRTL()

Description: Returns true if the user-preferred language setting is RTL and false otherwise.

Return Type: Boolean

Method Usage Example: `LocalizationModule.getIsRTL()`

Platforms Supported: All

getLanguageTag()

Description: Returns the language tag (for example, `en-CA` or `fr-CA`), based on the locale.

Return Type: String

Method Usage Example: `LocalizationModule.getLanguageTag()`

Platforms Supported: All

NOTE

Language tags have the following format: `[language designator]-[region designator]`. On Apple platforms, some languages have a region association—for example, English (Canada) and English (US). For these languages, the region-designator part of the language tag is not controlled by the region setting. As an example, if the user-preferred language is English (Canada) and the region is Russia, `getLanguageTag()` returns `en-CA`.

Other languages have no region association—for example, Hebrew and Korean. In these cases, the language tag is comprised of the preferred language plus the user's region setting. For example, if the user-preferred language is Hebrew and the region is Canada, `getLanguageTag()` returns `he-CA`.

getLocales()

Description: Returns the user-preferred locales, in order of preference. Note that not all devices allow multiple locales.

Return Type: String

Method Usage Example: `LocalizationModule.getLocales()`

Platforms Supported: All

getNumberFormatSettings()

Description: Returns number formatting settings for decimal separator and grouping separator, based on the locale. For example, if the locale is `en-CA`, this method returns `","` and if the locale is `en-FR`, the method returns `" "` (space). This method ignores the setting for grouping separator that's available on certain platforms, such as macOS.

Return Type: Object with `decimalSeparator` and `groupingSeparator` attributes

Method Usage Example: `LocalizationModule.getNumberFormatSettings()`

Platforms Supported: Decimal separator: All; grouping separator: All except PS4

`getTemperatureUnits()`

Description: Returns the user-preferred temperature unit: `celsius` or `fahrenheit`.

Return Type: Boolean

Method Usage Example: `LocalizationModule.getTemperatureUnits()`

Platforms Supported: iOS and tvOS. On other platforms, returns a value based on the current country code (which is based on the locale); for the country codes US, BS, BZ, KY, PR, and PW, this method returns Fahrenheit on unsupported platforms.

`getTimeZone()`

Description: Returns the user-preferred time zone (based on the device's settings, not on its geographical location).

Return Type: String

Method Usage Example: `LocalizationModule.getTimeZone()`

Platforms Supported: All

`getUses24HourClock()`

Description: Returns true if the user prefers the 24-hour clock format and false if the 12-hour clock format is preferred.

Return Type: Boolean

Method Usage Example: `LocalizationModule.getUses24HourClock()`

Platforms Supported: All

`getUsesMetricSystem()`

Description: Returns true if the officially supported measurement system is metric, based on the current country code. Returns true for all countries except US, MM (Myanmar), and LR (Liberia).

Return Type: Boolean

Method Usage Example: `LocalizationModule.getUsesMetricSystem()`

Platforms Supported: All except Android, PS4, Tizen, and webOS.

Usage

```

import React, { Component } from "react";
import { AppRegistry, StyleSheet, Text, View } from "react-native";
import { LocalizationModule } from "@youi/react-native-youi";

export default class LocalizationModuleExample extends Component {
  constructor(props) {
    super(props);
    this.state = {};
  }

  componentDidMount() {
    LocalizationModule.getLanguageTag()
      .then(languageTag => {
        this.setState({ languageTag: languageTag });
      })
      .catch(() => {
        console.log("Could not retrieve lanugage tag");
      });
    LocalizationModule.getUses24HourClock()
      .then(uses24HourClock => {
        this.setState({ uses24HourClock: uses24HourClock });
      })
      .catch(() => {
        console.log("Could not retrieve 24 hour clock");
      });
    LocalizationModule.getCurrencies()
      .then(currencies => {
        this.setState({ currencies: JSON.stringify(currencies) });
      })
      .catch(() => {
        console.log("Could not retrieve currencies");
      });
    LocalizationModule.getDateFormat()
      .then(dateFormat => {
        this.setState({ dateFormat: dateFormat });
      })
      .catch(() => {
        console.log("Could not retrieve date format");
      });
    LocalizationModule.getNumberFormatSettings()
      .then(res => {
        this.setState({
          decimalSeparator: res.decimalSeparator,
          groupingSeparator: res.groupingSeparator
        });
      })
      .catch(() => {
        console.log("Could not retrieve number format settings");
      });
  }

  render() {
    return (
      <View style={styles.container}>
        <Text style={styles.heading}>Localization module exposes information about the device the
app is running on.</Text>
        <Text style={styles.text}>Language tag: {this.state.languageTag}</Text>
        <Text style={styles.text}>uses24HourClock: {this.state.uses24HourClock}</Text>
        <Text style={styles.text}>Currencies: {this.state.currencies}</Text>
        <Text style={styles.text}>Date Format: {this.state.dateFormat}</Text>
      </View>
    );
  }
}

```

```
    <Text style={styles.text}>Decimal Separator: {this.state.decimalSeparator}</Text>
    <Text style={styles.text}>Grouping Separator: {this.state.groupingSeparator}</Text>
  </View>
);
}
}

const styles = StyleSheet.create({
  container: {
    flex: 1,
    alignItems: 'center',
    backgroundColor: '#d0d0d0'
  },
  heading: {
    fontSize: 16
  },
  text: {
    fontSize: 12
  }
});
```



[LEGAL](#)

[PRIVACY POLICY](#)

[TERMS AND CONDITIONS](#)

©2021 You.i TV. All rights reserved.