

This plugin is used for getting information about users' locale language, date and time zone, currency, etc.

Step 1 - Installing Globalization Plugin

Open **command prompt** and install the plugin by typing the following code

```
C:\Users\username\Desktop\CordovaProject>cordova plugin add cordova-plugin-globalization
```

Step 2 - Add Buttons

We will add several buttons to **index.html** to be able to call different methods that we will create later.

```
<button id = "getLanguage">LANGUAGE</button>
<button id = "getLocaleName">LOCALE NAME</button>
<button id = "getDate">DATE</button>
<button id = "getCurrency">CURRENCY</button>
```

Step 3 - Add Event Listeners

Event listeners will be added inside **getDeviceReady** function in **index.js** file to ensure that our app and Cordova are loaded before we start using it.

```
document.getElementById("getLanguage").addEventListener("click",
getLanguage);
document.getElementById("getLocaleName").addEventListener("click",
getLocaleName);
document.getElementById("getDate").addEventListener("click",
getDate);
document.getElementById("getCurrency").addEventListener("click",
getCurrency);
```

Step 4A - Language Function

The first function that we are using returns BCP 47 language tag of the client's device. We will use **getPreferredLanguage** method. The function has two parameters **onSuccess** and **onError**. We are adding this function in **index.js**.

```
function getLanguage() {
    navigator.globalization.getPreferredLanguage(onSuccess,
onError);

    function onSuccess(language) {
        alert('language: ' + language.value + '\n');
    }
}
```

```
}  
  
function onError(){  
    alert('Error getting language');  
}  
}
```

Once we press the **LANGUAGE** button, the alert will be shown on screen.



Step 4B - Locale Function

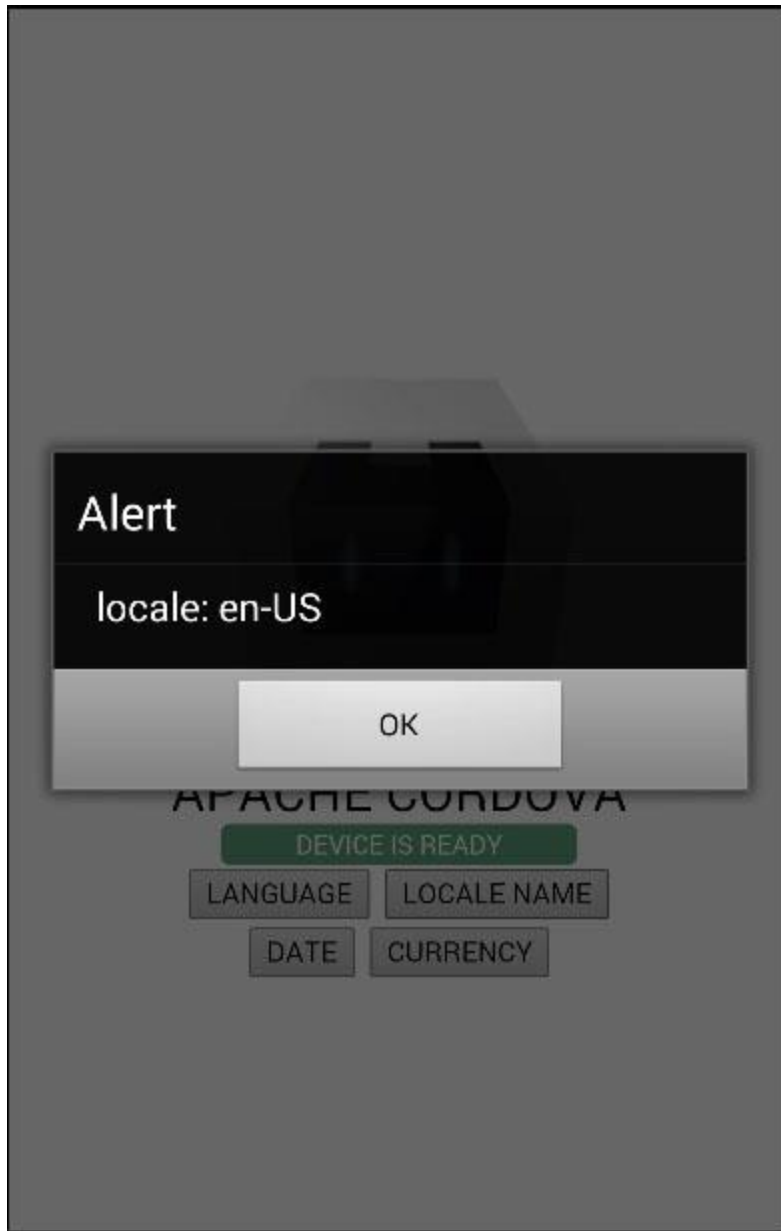
This function returns BCP 47 tag for the client's local settings. This function is similar as the one we created before. The only difference is that we are using **getLocaleName** method this time.

```
function getLocaleName() {
    navigator.globalization.getLocaleName(onSuccess, onError);

    function onSuccess(locale) {
        alert('locale: ' + locale.value);
    }

    function onError(){
        alert('Error getting locale');
    }
}
```

When we click the **LOCALE** button, the alert will show our locale tag.



Step 4C - Date Function

This function is used for returning date according to client's locale and timezone setting. **date** parameter is the current date and **options** parameter is optional.

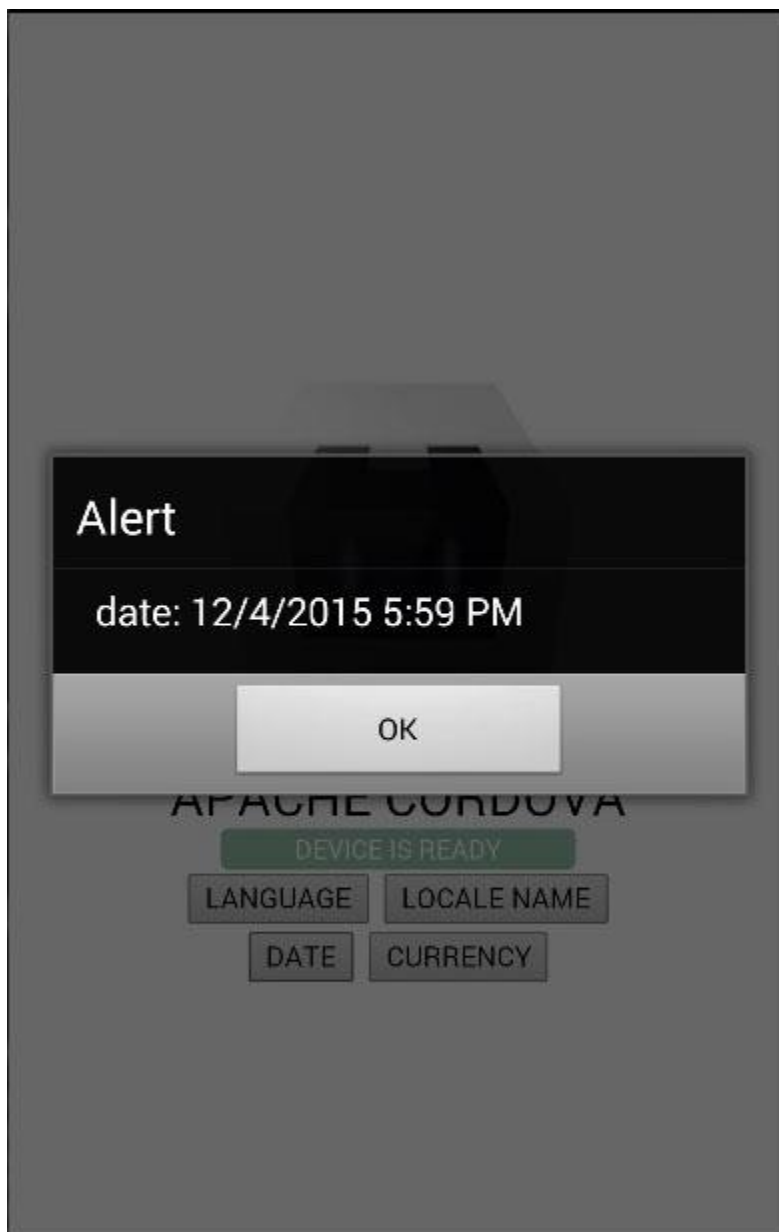
```
function getDate() {  
    var date = new Date();  
  
    var options = {  
        formatLength:'short',  
        selector:'date and time'  
    }  
}
```

```
    navigator.globalization.dateToString(date, onSuccess, onError,
options);

    function onSuccess(date) {
        alert('date: ' + date.value);
    }

    function onError(){
        alert('Error getting dateString');
    }
}
```

We can now run the app and press **DATE** button to see the current date.



The last function that we will show is returning currency values according to client's device settings and ISO 4217 currency code. You can see that the concept is the same.

```
function getCurrency() {
    var currencyCode = 'EUR';
    navigator.globalization.getCurrencyPattern(currencyCode,
    onSuccess, onError);

    function onSuccess(pattern) {
        alert('pattern: ' + pattern.pattern + '\n' +
            'code: ' + pattern.code + '\n' +
            'fraction: ' + pattern.fraction + '\n' +
            'rounding: ' + pattern.rounding + '\n' +
            'decimal: ' + pattern.decimal + '\n' +
            'grouping: ' + pattern.grouping);
    }

    function onError(){
        alert('Error getting pattern');
    }
}
```

The **CURRENCY** button will trigger alert that will show users currency pattern.

Alert

pattern: α#,##0.00;(α#,##0.00)
code: EUR
fraction: 2
rounding: 0
decimal: .
grouping: ,

OK

DATE

CURRENCY

Write Code for index.js file:-

```
function onLoad() {  
    document.addEventListener("deviceready", onDeviceReady, false);  
}
```

```
function onDeviceReady() {  
  
    document.getElementById("getLanguage").addEventListener("click",  
    getLanguage);  
    document.getElementById("getLocaleName").addEventListener("click",  
    getLocaleName);  
    document.getElementById("getDate").addEventListener("click",  
    getDate);  
    document.getElementById("getCurrency").addEventListener("click",  
    getCurrency);  
}
```

```
function getLanguage() {  
    navigator.globalization.getPreferredLanguage(onSuccess,  
    onError);
```

```
    function onSuccess(language) {  
        alert('language: ' + language.value + '\n');  
    }  
}
```

```
function onError(){  
    alert('Error getting language');  
}
```

```
}
```

```
function getLocaleName() {  
    navigator.globalization.getLocaleName(onSuccess, onError);
```



```
function onSuccess(locale) {
    alert('locale: ' + locale.value);
}

function onError(){
    alert('Error getting locale');
}
}

function getDate() {
    var date = new Date();

    var options = {
        formatLength:'short',
        selector:'date and time'
    }
    navigator.globalization.dateToString(date, onSuccess, onError,
options);

    function onSuccess(date) {
        alert('date: ' + date.value);
    }

    function onError(){
        alert('Error getting dateString');
    }
}

function getCurrency() {
    var currencyCode = 'EUR';
```

```
    navigator.globalization.getCurrencyPattern(currencyCode,  
onSuccess, onError);
```

```
function onSuccess(pattern) {  
    alert('pattern: ' + pattern.pattern + '\n' +  
        'code: ' + pattern.code + '\n' +  
        'fraction: ' + pattern.fraction + '\n' +  
        'rounding: ' + pattern.rounding + '\n' +  
        'decimal: ' + pattern.decimal + '\n' +  
        'grouping: ' + pattern.grouping);  
}
```

```
function onError(){  
    alert('Error getting pattern');  
}
```

```
}
```

Write code for index.html file:-

```
<!DOCTYPE html>
<html>
  <head>

    <title>Cordova Device Ready Example</title>

    <script type="text/javascript" charset="utf-8" src="cordova.js"></script>
      <script type="text/javascript" charset="utf-8" src="js/index.js"></script>
    </head>
  <body onload="onLoad()">
    <button id = "getLanguage">LANGUAGE</button>
    <button id = "getLocaleName">LOCALE NAME</button>
    <button id = "getDate">DATE</button>
    <button id = "getCurrency">CURRENCY</button>
  </body>
</html>
```