Pipalia®
Software House
SPECIALIST IN SAGE AND NOTES/DOMINO DEVELOPMENT

Using Validation and Vibration API in HTML5

August 10, 2014

HTML5 provides bevy of features which enable developers to build some really interesting applications. In this example I will build a small HTML5 based validation which will vibrate the mobile device in case of invalid input.

VALIDATION API

Validation in HTML5 is as simple as adding an attribute. Just add the *required* attribute and you are good to go.

```
<input type="text" required />
```

In the above code snippet the text field is required and browser supporting this attribute will not let you submit the form if the field value is empty.

Now we need to check if the user has entered valid value in the field or not. It he hasn't then we vibrate the mobile device. To check it in JavaScript use the statement *input.validity.valid*. This returns *true* in case valid value is entered in the field else returns *false*. In case its invalid you can then vibrate the mobile device using Vibration API.

VIBRATION API

This article describes Vibration API and how easy it is to implement.

First you need check if the device supports Vibration API:

```
if ("vibrate" in navigator) {...}
```

Then check the prefixed versions:

```
navigator.vibrate = navigator.vibrate || navigator.webkitVibrate || navigator.mozVibrate || navigator.msVibrate;
```

Then vibrate the device by calling:

```
navigator.vibrate(1000); // Vibrate for one second
```

PUTTING EVERYTHING TOGETHER

So you have an input field like this:

When user exits this field we call a function *check(this)* which checks if a valid value is entered in it or not. In this case the field cannot be empty. Here's how the function would be:

```
function check(input) {
        // Check if field value is valid or not
        if (!input.validity.valid) {
                // Add class invalidInput to input fields
                input.classList.add("invalidInput");

                // Check support for vibration API
                if ("vibrate" in navigator) {
                        // Check for the prefixed version
                        navigator.vibrate = navigator.vibrate || navigator.webkitVibrate || navigator.mozVibrate ||
                        if (navigator.vibrate) {
                                // Vibrate mobile device
                                navigator.vibrate([100, 200, 300]);
                        }
                }
        }
}
```

Here if *input.validity.valid* returns *false* then we set the class of input to *invalidInput*. Using CSS we style the input to show a red border around it and an exclamation mark. And then we vibrate the device three times.

This is how we style the *invalidInput*:

```
.invalidInput:required:invalid {
        background: url(http://dl.dropboxusercontent.com/u/44420778/exclamation16.png) no-repeat right -10px center;
        background-position: right 2px center;
        outline: none;
        box-shadow: 0px 0px 8px red;
    -moz-box-shadow: 0px 0px 8px red;
    -webkit-box-shadow: 0px 0px 8px red;
}
```

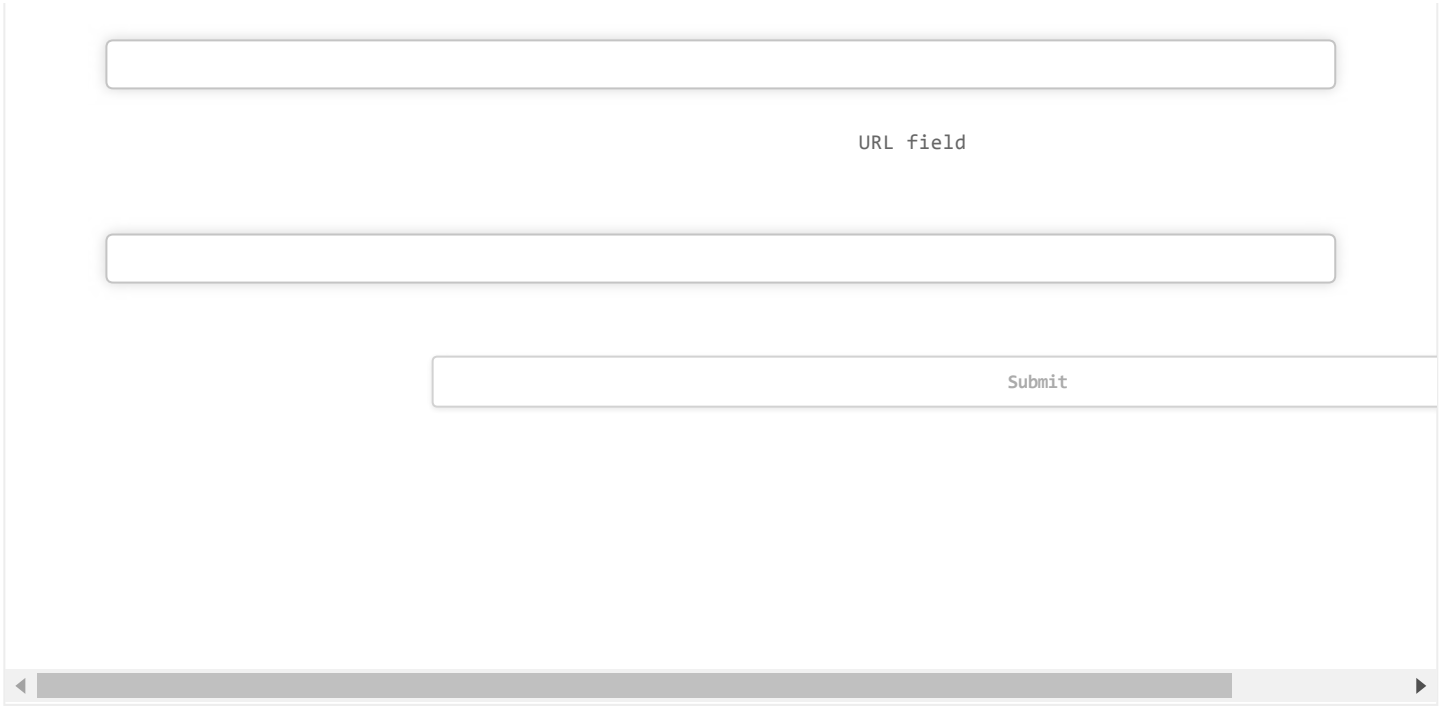
You can also validate for a particular pattern, email input, URL input and more. Just make sure you include the *required="required"* attribute. For e.g.:

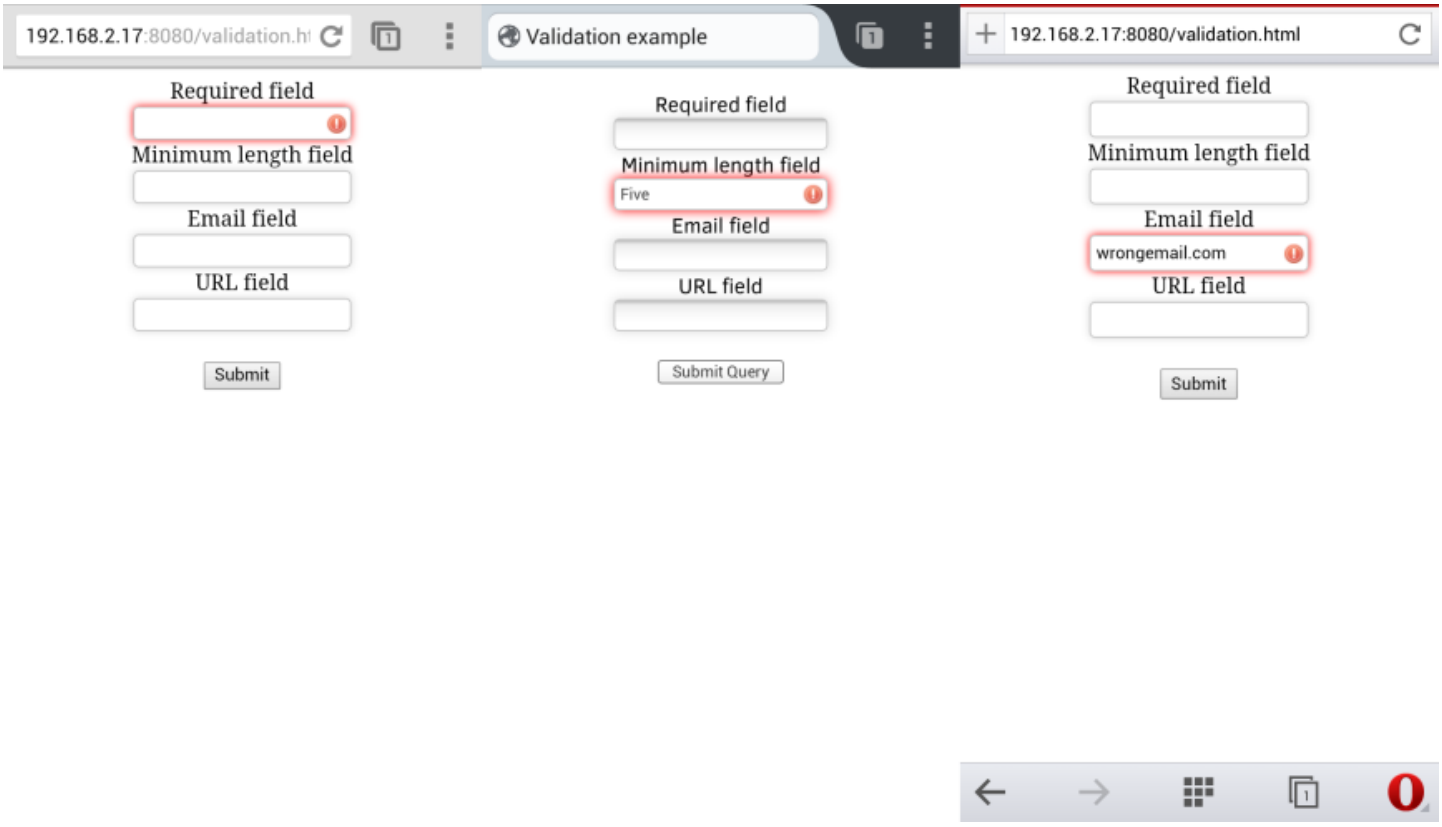
Below is the entire HTML code for my example. Go ahead and give it a try on your mobile.

Required field

Minimum length field

Email field





Share:

MORE
Tweet
Like 0
Share
Pin

August 10, 2014 [http://www.pipalia.co.uk/using-validation-vibration-api-html5/] | by Naveen Maurya | in HTML5 .

Comment