

How to create a new UI component in the WRTKit

WRTKit Introduction

The WRTKit is a complete library of JavaScript code, CSS style rules and graphical elements that are required to implement the kind of sophisticated, customizable, application-like user interfaces. It gives widget developers an easy way to create high quality user interfaces without having to worry about the problems that would otherwise plague their widgets.

The WRTKit folder has the following structure:

- Resources (Images and CSS files)
- UI (The user interface components, a lot of .js files)
- Utils (utils .js files, When this post was write there is only a Logger.js file)
- WRTKit.js (Import all the .js files in the UI folder)

Let's see in more details the UI folder, This folder contains all the UI components of the WRTKit, we have to understand them before start the creation of new UI components.

Extending WRTKit

In the UI folder, there is a Control.js file where is defined the Control class, this is an abstract base class for all user interface controls. Controls like FormButton, Label, TextArea, TextField and many more Controls are classes that inherit from Control class, directly or indirectly. Any new Control must inherits from Control class too.

The below code shows how to extend the Control class:

```
// Constructor.
function Map(id, caption,latitude,longitude) {
    if (id != UI_NO_INIT_ID) {
        this.init(id,caption,latitude,longitude);
    }
}

// Map inherits from Control.
Map.prototype = new Control(UI_NO_INIT_ID);
```

Remember that you have to put both files (Your new class and WRTKit.js) in the html code as showed below:

```
<script src="Map.js"/> <!-- If you save your new class as Map.js -->
<script src="WRTKit/WRTKit.js"/>
```

The Control class has a **controlElement** attribute, this is a div element. Any HTML element that you want to put inside your new Control has to be included inside the attribute **controlElement**. For example, if you want a image in your Map control, you can do this using the follow code:

```
//document.createElement is a Javascript default method to create any HTML element at
runtime.
var mapImage = document.createElement("img");
//After create you can get and set attributes of the element created.
mapImage.src = "url to the image";
//Now you can add the new element into the control element of the new Map control.
this.controlElement.appendChild(mapImage);
```

Complete example

be an imported CSS file)

```
/* Caption for controls (captionElement) */  
.ControlCaptionMap {  
    font-weight: bold;  
    padding: 3px 0px 0px 3px;  
    color: #FF0000;  
    horizontal-align: center;  
    text-align: center;  
}  
/* Caption for controls (captionElement) */  
.ControlMap {  
    padding: 3px 0px 0px 3px;  
    horizontal-align: center;  
    vertical-align: middle;  
    text-align: center;  
}
```

After import this class into your HTML file, you can use it with the below code:

```
listView = new ListView(null, "List Selection");  
mapControl = new Map("MyMap", "", -7, -35);  
listView.addControl(mapControl);  
mapControl.reloadMap("You're Here!");
```