Progress Meter Component

The goal of this project is to produce a JavaScript component to that implements a Bootstrap (v3.x) Progress Bar, and a possible list of messages. Once instantiated, the component will poll a server at a configurable URL, at a configurable time interval, for a JSON response and using that response to update the progress meter.

The component will be used to provide a live progress display of a background process that is running on the server. The libraries available for use are jQuery, Bootstrap v3, and Backbone.

The implementation should be implemented as a Backbone construct, but also include a single wrapping invocation function, this function could be implemented as a jQuery plugin, for example, but that is optional. The component should be easily declarable as a JavaScript function call, which takes a configuration object as it's argument in the hosting HTML page.

For example:

```
<script type="text/javascript">
  // As a simple JavaScript component
   namespace.LiveProgressMeter({
       'interval' : 10 // in seconds,
     'url' : '../runsProcess,
     'el' : '#cssSelector'
   });

   // Or, as a jQuery plugin
   $('#cssSelector').LiveProgressMeter({
       'interval' : 10 // in seconds,
     'url' : '../runsProcess
   });

  // The name 'LiveProgressMeter' is for example only, and need not be the function name.
</script>
```

The (ajax) JSON response from the server will be of the form:

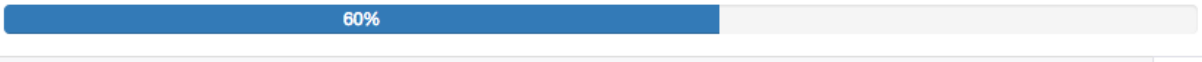
```
{
    percentComplete : .60,
    messages : [{'cssClass' : 'alert alert-info', 'text' : 'Working...'}]
}
```

The component should specify that it expects a JSON response from the server, as a default HTML response from that same URL may provide different, and not compatible output.

The percentComplete value should update the Progress bar display. The array of messages may or may not be present, if present, it will be a set of objects each of which will include a CSS class, and text. They can be rendered as an unordered list with the CSS classes set on each list item. It is the components responsibility to create and maintain this list.

The percent value should be displayed as shown:

EXAMPLE



Copy

```
<div class="progress">
  <div class="progress-bar" role="progressbar" aria-valuenow="60" aria-valuemin="0" aria-
valuemax="100" style="width: 60%;">
    60%
  </div>
</div>
```

With message list displayed below the progress meter.

When the percentComplete value reaches 1.0 the component should stop polling the server. In addition if there is a network or communication error, the component should place an appropriate message in the message list. Use appropriate Bootstrap styles to indicate error conditions.

The component, at a minimum, should fire the following events:

1. An update event.
2. A completion event.

It must provide a “remove” method that will stop the component, remove any event bindings, and remove the entire construct from the host HTML DOM.