


DisplayValueOnly

Consider you are creating a page where users can inspect data and change it. You might want to implement different states of the page, one where users can only see the data, and not change it (let's call it *view mode* `DISPLAY`), and another where users can also modify the data (let's call it *view mode* `EDIT`). In view mode `EDIT`, you obviously need HTML input tags, to have the browser show editable fields. For a clean user interface, you don't want editable fields in view mode `DISPLAY`.

One solution to make the difference, apart from creating 2 separate `.jsp`  pages, is to disable the fields in view mode `DISPLAY`:

```
<h:inputText ... disabled="#{myHandler.viewMode == 'DISPLAY'}" ...>
```

With the extended standard components of Tomahawk, it is also possible to render the value of the fields just as text, without the field. This is controlled by the boolean `displayValueOnly` attribute. If the value of this attribute evaluates to `true`, no HTML input tags are rendered, but the component behaves like the standard `<h:outputText>` component. If the value of this attribute evaluates to `false`, HTML input tags are rendered, and the component behaves like the equivalent standard input component.

When the component is rendered as value only, you still want to control it's representation. For that, the attribute `displayValueOnlyStyle` and `displayValueOnlyStyleClass` are at your disposal. The `css style` or `resp. style class` mentioned as values for these attributes are applied to a HTML span tag around the displayed text when the value of the `displayValueOnly` attribute evaluates to `true`