

Dojo to Dijit

Bill Keese

Philosophy

- Remove degrees of freedom
- Less magic
- Streamlined code
 - Remove options
 - Only include code you need
- Leverage browser (CSS, etc.)
- Optimize for HTML (not SVG/VML)
- Optimize templates for tundra theme

Goals

- Standard set of widgets just like in swing
- Standard API
- Standard look & feel
- I18N
- A11Y
- dojo.data integration

- Support simple web page or application similar to [yahoo mail](#)
- Customization via CSS

Directories

- dojo/
 - ...
- dijit/
 - base/
 - util/
 - form/
 - themes/
- dojox/
 - grid
 - binding code

Programatic Creation

```
dojo.require("dijit.form.Button");
```



Replaces `dojo.widget.createWidget()`
since multiple renderers not
supported

```
var node=dojo.byId('myButton');  
new dijit.form.Button({disabled: true}, node)
```

OR to create a new node:

```
new dijit.form.Button({disabled: true})
```

Parser, Markup creation

```
dojo.require("dijit.util.parser");
```

For
parsing

```
dojo.require("dijit.form.Button");
```

No auto-
require

```
<button dojoType=dijit.form.Button  
disabled=true>
```

Replaces namespaces.
Case sensitive

Styling

- No “default” CSS styling except essentials in `dijit.css`

Tundra theme

tundra.css

```
.tundra .dojoButton  
{ ... }
```

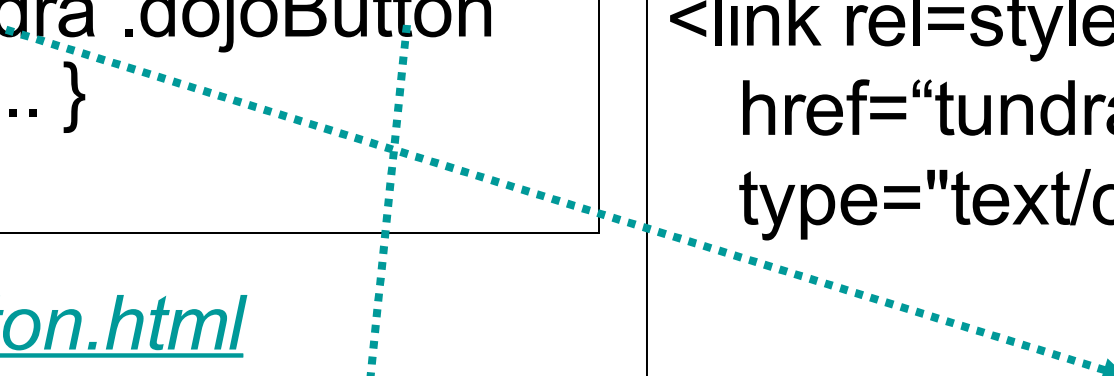
Button.html

```
<button  
  class="dojoButton"  
> ...
```

index.html

```
<link rel=stylesheet  
  href="tundra.css"  
  type="text/css">
```

```
<body class="tundra">  
  <div  
    dojoType=dijit.Button>  
</body>
```



User defined two-theme layout

Two column layout:

```
<div class="beach" style="float:left">  
  <button dojoType=dijit.form.Button>...
```



```
</div>
```

```
<div class="ocean">
```

```
  <button dojoType=dijit.form.Button> ...
```

```
</div>
```

Declare widget

```
dojo.declare( dojo.widget.defineWidget() removed  
  "dijit.ProgressBar",  No svg/vml/html distinction  
  [dijit.base.Widget,  
   dijit.base.TemplatedWidget],  
  null,  
  {  
    ...
```

Widget base class

- Widget.js, Domwidget.js, HtmlWidget.js combined
- TemplatedWidget mixin
- creation: postMixinProperties(), buildRendering() and postCreate().
- When postCreate() is called the widget children don't yet exist.
- onResized() removed
- templateCssPath ignored. User must manually include tundra.css file

Dojo.data Grid

Grid standard paging:

User code:

```
<table dojoType="dijit.Grid" data="store1">
```

```
function onItem(item){  
  for(var i=0; i < this.columns.length; i++){  
    var value = store.getValue(item,  
      this.columns[i].attribute); ...  
  }  
}
```

dojo.data Select

- Select uses query interface:
- `<select`
 `dojoType=dijit.Select`
 `store="stateStore"`
 `searchField="name"` *pattern match against this field*
 `keyField="abbreviation"` *the hidden value*
 `labelField="label"` *the displayed value*
 `>`
 `</select>`

dojo.data Select (cont)

```
store.fetch({  
  query: {name: "c*"},  
  onItem: myCallbackFunction  
});
```

Form widgets

- Standard attributes:
 - id, name, value, etc.
- Standard methods:
 - setValue(), getValue()
 - getTextValue()

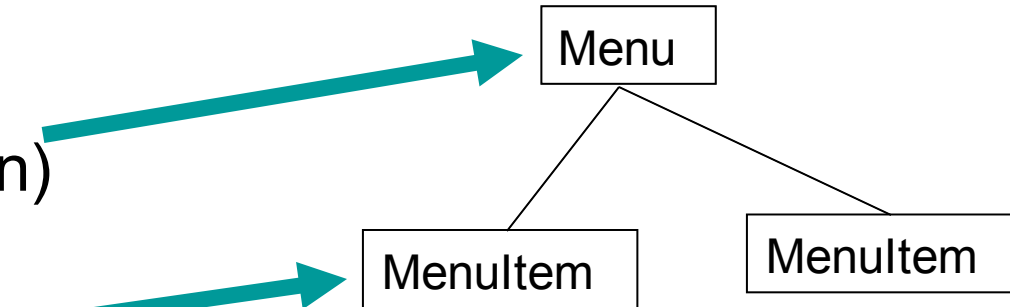
```
<input dojoType=DropDownCalendar  
value="2007-07-07"> ->
```

07/07/2007

Layout widgets

- Base classes

- Container (has children)
- Contained (has parent)



-
- Sizable (can set outer dom node size)

- layout() method recurses

