



KISSY Component API Design

yiminghe@gmail.com

2012-06-05 draft

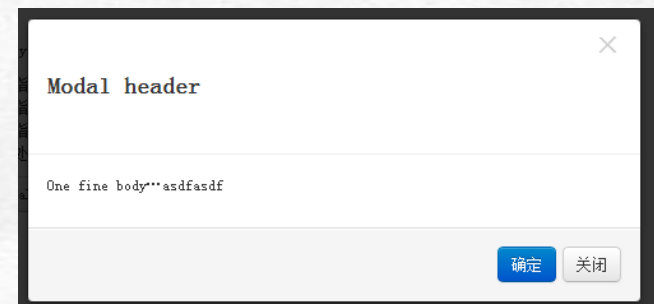
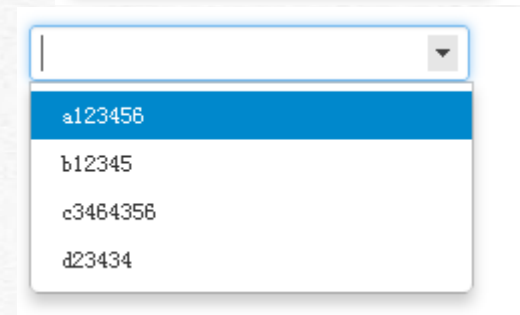
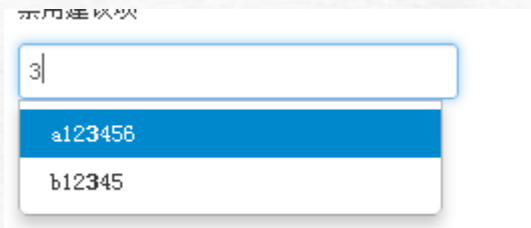
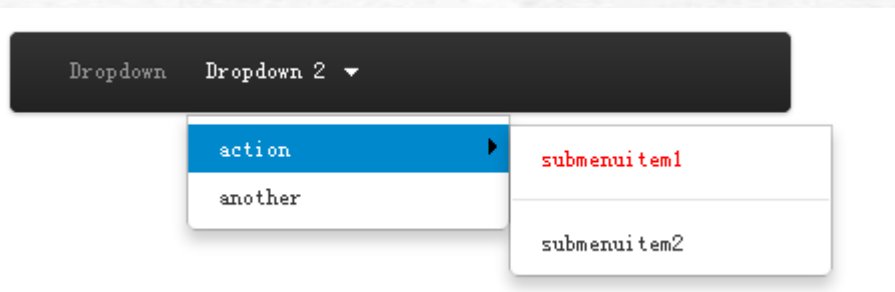
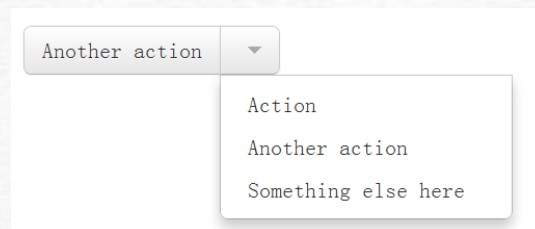


Outline

- Why create component
- Why API design
- KISSY Component API Design
- API Design principles

you probably haven't heard of me.
sustainable quinoa 8-bit and
will The last tip! four le
ustin [twitter handle](#) freega

Components



<http://docs.kissyui.com/kissy-bootstrap/docs/>

Components



昵图网 www.nipic.com 8y_0098hq

102-20001015202410000000

Why create components

- Ease of development

Why create components

- Ease of development
- Reusability

Why create components

- Ease of development
- Reusability
- Maintainability

Why API Design

- Contract between user and developer

Why API Design

- Contract between user and developer
- Stable platform to build on

Why API Design

- Contract between user and developer
- Stable platform to build on
- **Minimize waste**
 - Code-reuse instead of re-writing
 - Reduce code and complexity

Component API Design

- Structure
- Subcomponent
- Subclass
- Plugin
- Events
 - Lifecycle event
- Skin
- Creation

Structure

- Config
- Attribute
- Method



Config

- Config

```
new Overlay({  
    width:100,  
    height:100,  
    content:'i am overlay'  
})
```


Attribute

- Attribute

- Talk to your component

```
var o=new Overlay({});  
o.get("content");  
o.set("content","xx");
```



Method

- Method
 - Attribute first, method second

```
var o=new Tree({});  
o.collapseAll();  
o.set("collapsed",true);
```

Subcomponent

- Config/Attribute

- Children

- Method

- addChild/removeChild/removeChildren

Subcomponent

- Config/Attribute

- Children

```
var o=new Menu({  
  children:[{new Menu.Item(){}},{xclass:'menuitem', content:'menuitem-  
content2' }]  
});  
o.get("children");
```

Subcomponent

•Method

- addChild/removeChild/removeChildren

```
var o=new Menu ( { } );  
o.addChild ( new Menu . Item ( ) );
```


Subclass

- Easy to extend

```
var MyOverlay = Overlay.extend({
  initialize:function() {},
  createDom:function() {},
  renderUI:function() {},
  syncUI:function() {}
}, {
  ATTRS: {
    myAttr: {}
  }
});
```

Plugin

- enhance ability dynamically at runtime

```
var editor = new Editor({  
  plugins:[  
    FontSize,  
    new Image({  
      url:'upload.htm'  
    })  
  ]  
});
```

Events

- Event

- Native lifecycle event / attrChange event / Custom event
- Method: on / detach
- Config : listeners

```
{  
  type: ,  
  target: , // consider bubbling  
  yy: xx  
}
```



Events

- Custom Event

```
var o = new Overlay({  
    listeners : { hide : { fn : function() {} } }  
});  
o.on("show", function() {  
});
```

Events

- Lifecycle event

- beforeCreateDom/afterCreateDom/beforeRenderUI/afterRenderUI
/beforeBindUI/afterBindUI/..

```
var o=new Overlay();  
o.on("afterCreateDom",function(){  
    alert(o.get("el"));  
});
```


Events

- AttrChange event

- beforeAttrChange/afterAttrChangeDom

```
var o=new Button.Toggle();  
o.on("afterCheckedChange",function(){  
    alert(o.get("checked"));  
});
```

Skin

- Config

- prefixCls

```
new Button ({  
  prefixCls: "xx-"  
});
```



Creation

- new
- srcNode
- xclass



Creation

- new

```
var m = new Menu();  
m.addChild(new Menu.Item());  
m.on("click", function() {})  
m.render();
```

Creation

- srcNode

```
<div class='ks-menu'>  
<div class='ks-  
  menuitem'></div>  
</div>
```

```
new Menu ({  
  srcNode:div,  
  listeners:{  
    click:function() {  
    }  
  }  
}) ;
```


Creation

- xclass

```
new Menu({  
  children:[{  
    xclass:'menuitem',  
    content:'yy'  
  }],  
  render:container,  
  listeners:{  
    click:function(){}  
  })  
});
```

API Design Principle

- Hide implementation

```
new Overlay({  
  // view: new OverlayRender()  
});
```

API Design Principle

- Easy to learn
 - consistency
 - simple
- `elCls/el/srcNode/get()/set()`

API Design Principle

- Easy to read and write

```
new Overlay({  
  width:,  
  height:'',  
  children:[{  
    xclass:'menu'  
  }])});
```

API Design Principle

- Easy to extend

```
var MyOverlay = Overlay.extend(...);
```


API Design Principle

- make api complete
- separate concern
- **new/addChild/removeChild/destroy/on/detach**



KISSY



THANKS FOR
COMING

SEE YOU SOON!