

互动操作组件

本章主要介绍互动操作组件的属性和常见用法,包括底部菜单组件 `action-sheet`、弹出对话框组件 `modal`、消息提示框组件 `toast`、加载提示组件 `loading` 等组件的属性和用法。App 软件经常可以从底部弹出很多选项,这种效果可以在小程序中用 `action-sheet` 组件实现。弹出对话框组件 `modal` 常用来提示一些信息;例如,退出应用、修改资料等。消息提示框组件 `toast` 经常用来提示提交成功或者正在加载。加载提示组件 `loading` 通常使用在请求网络数据时的一种方式,通过 `hidden` 属性设置显示与否。

5.1 底部菜单组件 `action-sheet`

App 软件经常可以从底部弹出很多选项,这种效果可以在小程序中用 `action-sheet` 组件实现。`action-sheet` 组件是从底部弹出可选菜单项,`action-sheet` 有两个子组件:每个选项 `action-sheet-item` 和取消选项 `action-sheet-cancel`。`action-sheet-cancel` 和 `action-sheet-item` 的区别是:点击 `action-sheet-cancel` 会触发 `action-sheet` 的 `change` 事件,并且外观上会同 `action-sheet-cancel` 上面的内容间隔开来。`action-sheet` 的属性说明如表 5-1 所示。在 `action-sheet` 组件中可以放置任何组件。

表 5-1 `action-sheet` 属性相关信息

属 性	类 型	说 明
<code>hidden</code>	Boolean	是否隐藏,默认值为 <code>true</code>
<code>bindchange</code>	EventHandle	点击背景或 <code>action-sheet-cancel</code> 按钮时触发 <code>change</code> 事件,不携带数据

例 5-1 代码如下,其效果如图 5-1 所示。

例 5-1

```
<!-- index.wxml -->
<button type="default" bindtap="actionSheetTap">弹出 action sheet</button>
<action-sheet hidden="{{actionSheetHidden}}" bindchange="actionSheetChange">
  <block wx:for-items="{{actionSheetItems}}">
    <action-sheet-item class="item" bindtap="bind{{item}}">{{item}}
  </action-sheet-item>
```

```
</block>
<action-sheet-cancel class="cancel">取消</action-sheet-cancel>
</action-sheet>

//index.js
var items=['item1', 'item2', 'item3', 'item4']
var pageObject={
  data: {
    actionSheetHidden: true,
    actionSheetItems: items
  },
  actionSheetTap: function(e) {
    this.setData({
      actionSheetHidden: ! this.data.actionSheetHidden
    })
  },
  actionSheetChange: function(e) {
    this.setData({
      actionSheetHidden: ! this.data.actionSheetHidden
    })
  }
}
for (var i=0; i<items.length;++i) {
  (function(itemName) {
    pageObject['bind'+itemName]=function(e) {
      console.log('click'+itemName, e)
    }
  })(items[i])
}
Page(pageObject)
```



图 5-1 action-sheet 组件的应用

5.2 弹出对话框组件 modal

弹出对话框组件 modal 常用来提示一些信息;例如,退出应用、修改资料等。模态弹窗是对整个界面进行覆盖,防止用户对界面中的其他内容进行操作。对于需要用户明确知晓的操作结果状态可通过模态对话框来提示,并可附带下一步操作指引。modal 的属性说明如表 5-2 所示。

表 5-2 modal 属性相关信息

属 性	类 型	说 明
title	String	标题
hidden	Boolean	是否隐藏整个弹窗,默认值为 false
no-cancel	Boolean	是否隐藏 cancel 按钮,默认值为 false
confirm-text	String	confirm 按钮文字,默认值为确定
cancel-text	String	cancel 按钮文字,默认值为取消
bindconfirm	EventHandle	点击确认触发的回调
bindcancel	EventHandle	点击取消以及蒙层触发的回调

例 5-2 代码如下,其效果如图 5-2 所示。

例 5-2

```
<!-- index.wxml -->
<view>
  <modal title="两个按钮" confirm-text="确定"
    cancel-text="取消" hidden="{{modalHidden1}}"
    bindconfirm="modalChange1" bindcancel="modalChange1" >
    这是对话框的内容
  </modal>
  <modal title="一个按钮" no-cancel confirm-text="确定"
    cancel-text="取消" hidden="{{modalHidden2}}"
    bindconfirm="modalChange2" bindcancel="modalChange2" >
    <image src="http://geekori.cn/img/weixin_code.png" style="width:
      100px;height:100px"/>
  </modal>
  <view style="margin:20px">
    <button bindtap="modalTap1">点击弹出包含两个按钮的对话框</button>
    <button style="margin-top:20px" bindtap="modalTap2">点击弹出包含一个按钮的对
    话框</button>
  </view>
</view>

//index.js
```

```
var items=['第一项', '第二项', '第三项', '第四项'];
Page({
  data: {
    modalHidden1: true,
    modalHidden2: true
  },
  modalTap1: function(e) {
    this.setData({
      modalHidden1: false
    })
  },
  modalChange1: function(e) {
    this.setData({
      modalHidden1: true
    })
  },
  modalTap2: function(e) {
    this.setData({
      modalHidden2: false
    })
  },
  modalChange2: function(e) {
    this.setData({
      modalHidden2: true
    })
  }
})
```



图 5-2 modal 组件的应用

5.3 消息提示框组件 toast

消息提示框组件 toast 经常用来提示提交成功或者正在加载。toast 弹出式提示适用于轻量级的成功提示,1.5s 后自动消失,对用户影响较小,适用于不需要强调的操作提醒,例如成功提示。注意该形式不适用于错误提示,因为错误提示需要明确告知用户,不适合使用一闪而过的弹出式提示。toast 的属性说明如表 5-3 所示。

表 5-3 toast 属性相关信息

属性	类型	说明
duration	Float	hidden 设置 false 后,触发 bindchange 的延时,单位为毫秒(ms),默认值为 1500
hidden	Boolean	是否隐藏,默认值为 false
bindchange	EventHandle	duration 延时后触发

例 5-3 代码如下,其效果如图 5-3 所示。

例 5-3

```

<!-- index.wxml -->
<view class="body-view">
  <toast hidden="{{toast1Hidden}}" bindchange="toast1Change">
    默认
  </toast>
  <button type="default" bindtap="toast1Tap">点击弹出默认 toast</button>
</view>
<view class="body-view">
  <toast hidden="{{toast2Hidden}}" duration="3000" bindchange="
  toast2Change">
    设置 duration
  </toast>
  <button type="default" bindtap="toast2Tap">点击弹出设置 duration 的 toast
</button>
</view>

//index.js
var toastNum=2
var pageData={}
pageData.data={}
for (var i=0; i<=toastNum;++i) {
  pageData.data['toast'+i+'Hidden']=true;
  (function(index) {
    pageData['toast'+index+'Change']=function(e) {

```

```

    var obj={}
    obj['toast'+index+'Hidden']=true;
    this.setData(obj)
  }
  pageData['toast'+index+'Tap']=function(e) {
    var obj={}
    obj['toast'+index+'Hidden']=false
    this.setData(obj)
  }
  })(i)
}
Page(pageData)

```

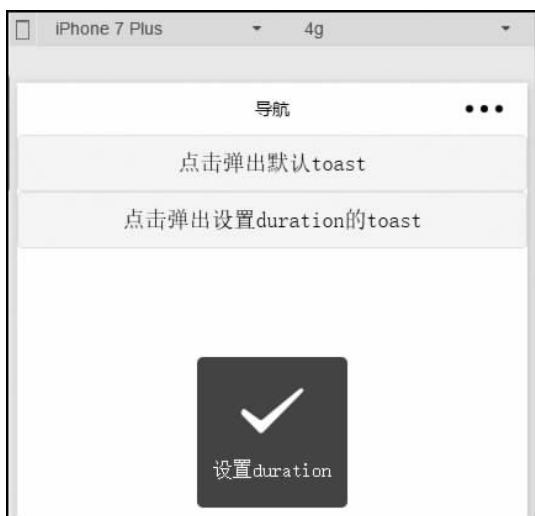


图 5-3 toast 组件的应用

5.4 加载提示组件 loading

加载提示组件 loading 通常使用在请求网络数据时的一种方式,通过 hidden 属性设置显示与否。如果加载样式覆盖整个页面,由于无法明确告知具体加载的位置或内容将可能引起用户的焦虑感,因此应谨慎使用。除了在某些全局性操作下不要使用模态的加载。局部加载反馈只在触发加载的页面局部进行反馈,这样的反馈机制更加有针对性,页面跳动小,是微信推荐的反馈方式。loading 属性为 hidden,默认为 false(不隐藏)。

例 5-4 代码如下,其效果如图 5-4 所示。

例 5-4

```

<!-- index.wxml -->
<view class="body-view">

```

```
<loading hidden="{{hidden}}">
  加载中...
</loading>
<button type="default" bindtap="loadingTap">点击弹出 loading</button>
</view>

//index.js
Page({
  data: {
    hidden: true
  },
  loadingTap: function() {
    this.setData({
      hidden: false
    });
    var that=this;
    setTimeout(function() {
      that.setData({
        hidden: true
      });
      that.update();
    }, 3000);
  }
})
```



图 5-4 loading 组件的应用



习 题 5

问答题

1. 请说明 action-sheet 组件的常见属性。
2. 请说明 modal 组件的常见属性。
3. 请说明 toast 组件的常见属性。
4. 请说明 loading 组件的常见属性。