Coco-LG 的学习笔记《一》事件

微信小程序的"事件"挺有意思。看了说明文档后发现它的功能很全，事件可以向父节点传递，而且打印这个事件的信息很透明，调试起来应该非常方便。

接下来把文档 copy 过来，原文地址：

https://mp.weixin.qq.com/debug/wxadoc/dev/framework/view/wxml/event.html

什么是事件

- 事件是视图层到逻辑层的通讯方式。
- 事件可以将用户的行为反馈到逻辑层进行处理。
- 事件可以绑定在组件上，当达到触发事件，就会执行逻辑层中对应的事件处理函数。
- 事件对象可以携带额外信息，如 id, dataset, touches。

事件的使用方式

- 在组件中绑定一个事件处理函数。

如 `bindtap`，当用户点击该组件的时候会在该页面对应的 Page 中找到相应的事件处理函数。

[html] view plain copy

```
1.  <view id="tapTest" data-hi="MINA" bindtap="tapName"> Click me! </view>
```

- 在相应的 Page 定义中写上相应的事件处理函数，参数是 event。

[javascript] view plain copy

```
Page({
  tapName: function(event) {
    console.log(event)
  }
})
```

可以看到 log 出来的信息大致如下：

[javascript] view plain copy

```
{
"type": "tap",
"timeStamp": 1252,
"target": {
  "id": "tapTest",
  "offsetLeft": 0,
  "offsetTop": 0,
  "dataset": {
   "hi": "MINA"
  }
},
"currentTarget": {
  "id": "tapTest",
  "offsetLeft": 0,
```

```
    "offsetTop": 0,
    "dataset": {
      "hi": "MINA"
    }
  },
  "touches": [{
    "pageX": 30,
    "pageY": 12,
    "clientX": 30,
    "clientY": 12,
    "screenX": 112,
    "screenY": 151
  }],
  "detail": {
    "x": 30,
    "y": 12
  }
}
```

事件详解

事件分类

事件分为冒泡事件和非冒泡事件：

1. 冒泡事件：当一个组件上的事件被触发后，该事件会向父节点传递。
2. 非冒泡事件：当一个组件上的事件被触发后，该事件不会向父节点传递。

WXML 的冒泡事件列表：

类型	触发条件
touchstart	手指触摸
touchmove	手指触摸后移动
touchcancel	手指触摸动作被打断，如来电提醒，弹窗
touchend	手指触摸动作结束
tap	手指触摸后离开
longtap	手指触摸后，超过 350ms 再离开

注：除上表之外的其他组件自定义事件都是非冒泡事件，如<form/>的 submit 事件，<input/>的 input 事件，<scroll-view/>的 scroll 事件，(详见各个组件)

事件绑定

事件绑定的写法同组件的属性，以 key、value 的形式。

- key 以 bind 或 catch 开头，然后跟上事件的类型，如 bindtap, catchtouchstart
- value 是一个字符串，需要在对应的 Page 中定义同名的函数。不然当触发事件的时候会报错。

bind 事件绑定不会阻止冒泡事件向上冒泡，catch 事件绑定可以阻止冒泡事件向上冒泡。

如在下边这个例子中，点击 inner view 会先后触发 handleTap3 和 handleTap2(因为 tap 事件会冒泡到

middle view，而 middle view 阻止了 tap 事件冒泡，不再向父节点传递)，点击 middle view 会触发 handleTap2，点击 outter view 会触发 handleTap1。

[html] view plain copy

```
<view id="outter" bindtap="handleTap1">
  outer view
  <view id="middle" catchtap="handleTap2">
    middle view
    <view id="inner" bindtap="handleTap3">
      inner view
    </view>
  </view>
</view>
```

事件对象

如无特殊说明，当组件触发事件时，逻辑层绑定该事件的处理函数会收到一个事件对象。

事件对象的属性列表：

属性	类型	说明
type	String	事件类型
timeStamp	Integer	事件生成时的时间戳
target	Object	触发事件的组件的一些属性值集合
currentTarget	Object	当前组件的一些属性值集合

touches Array 触摸事件，触摸点信息的数组

detail Object 额外的信息

type

通用事件类型

timeStamp

该页面打开到触发事件所经过的毫秒数。

target

触发事件的源组件。

属性	说明
id	事件源组件的 id
dataset	事件源组件上由 data-开头的自定义属性组成的集合

offsetLeft, offsetTop事件源组件的坐标系统中偏移量

currentTarget

事件绑定的当前组件。

属性	说明

id	当前组件的 id
dataset	当前组件上由 data-开头的自定义属性组成的集合
offsetLeft, offsetTop	当前组件的坐标系统中偏移量

说明: target 和 currentTarget 可以参考上例中，点击 inner view 时，handleTap3 收到的事件对象 target 和 currentTarget 都是 inner，而 handleTap2 收到的事件对象 target 就是 inner，currentTarget 就是 middle

dataset

在组件中可以定义数据，这些数据将会通过事件传递给 SERVICE。书写方式：以 data-开头，多个单词由连字符-链接，不能有大写(大写会自动转成小写)如 data-element-type，最终在 event.target.dataset 中会将连字符转成驼峰 elementType。

示例：

[html] view plain copy

```
<view data-alpha-beta="1" data-alphaBeta="2" bindtap="bindViewTap"> DataSet Test </view>
```

[javascript] view plain copy

```
Page({
  bindViewTap:function(event){
    event.target.dataset.alphaBeta == 1 // - 会转为驼峰写法
    event.target.dataset.alphabeta == 2 // 大写会转为小写
```

```
  }
})
```

touches

touches 是一个触摸点的数组，每个触摸点包括以下属性：

属性	说明
pageX,pageY	距离文档左上角的距离，文档的左上角为原点 ，横向为 X 轴，纵向为 Y 轴
clientX,clientY	距离页面可显示区域（屏幕除去导航条）左上角距离，横向为 X 轴，纵向为 Y 轴
screenX,screenY	距离屏幕左上角的距离，屏幕左上角为原点，横向为 X 轴，纵向为 Y 轴

detail

特殊事件所携带的数据，如表单组件的提交事件会携带用户的输入，媒体的错误事件会携带错误信息，详见组件定义中各个事件的定义。