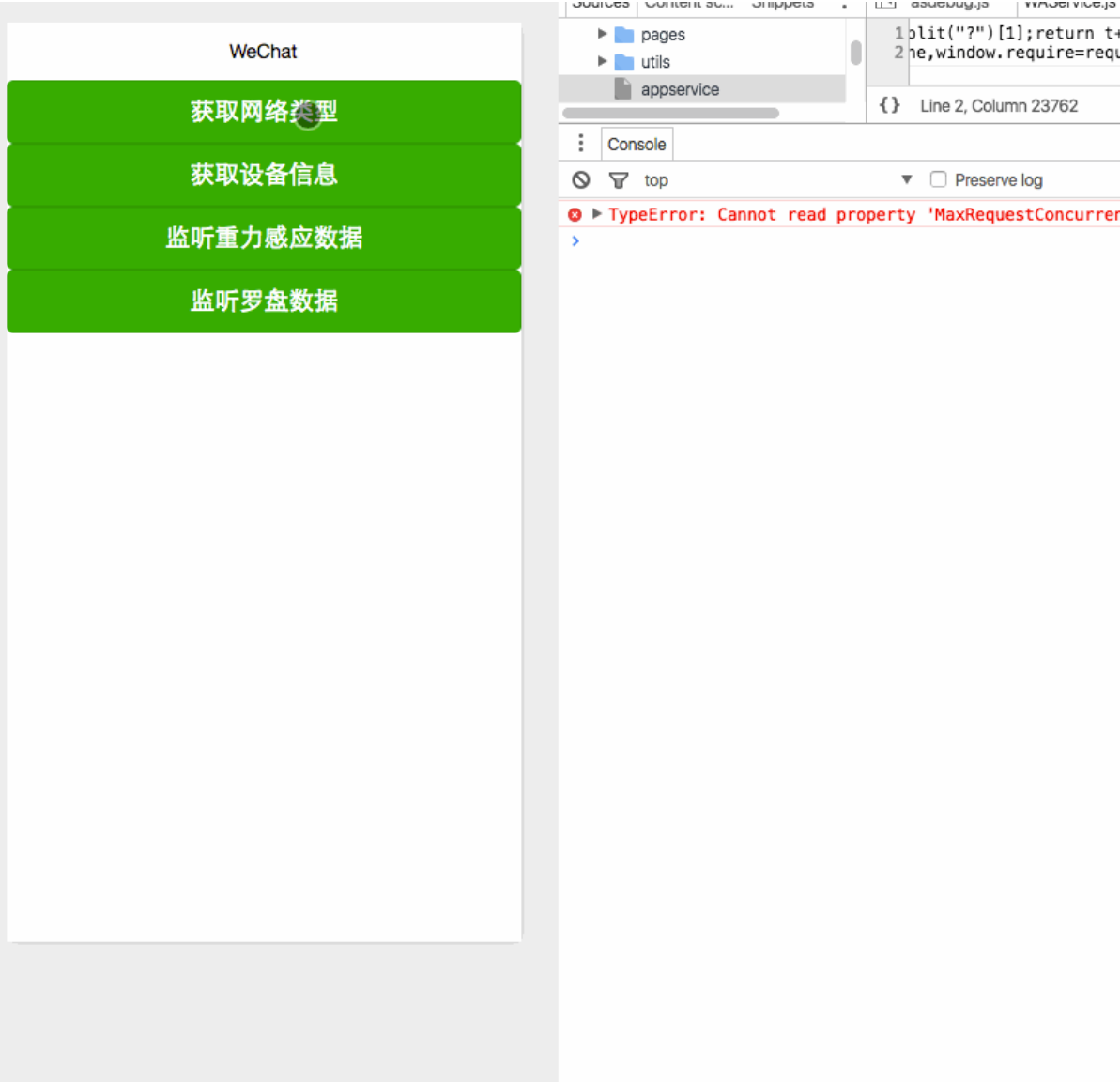


# 微信小程序把玩《十三》：获取设备信息 API，navigation API，animation API ...

## 一：获取设备信息 API



获取设备信息这里分为四种，

主要属性：

- 网络信息 wx.getNetWorkType,

参数	类型	必填	说明
success	Function	是	返回网络类型networkType
fail	Function	否	接口调用失败的回调函数
complete	Function	否	接口调用结束的回调函数（调用成功、失败都会执行）

- 系统信息 wx.getSystemInfo,

属性	说明
model	手机型号
pixelRatio	设备像素比
windowWidth	窗口宽度
windowHeight	窗口高度
language	微信设置的语言
version	微信版本号

- 重力感应数据 wx.onAccelerometerChange,

参数	类型	说明
x	Number	X 轴
y	Number	Y 轴
z	Number	Z 轴

- 罗盘数据 wx.onCompassChange

参数	类型	说明
direction	Number	面对的方向度数

wxml

```
<button type="primary" bindtap="getNetWorkType">获取网络类型</button>
```

```
<button type="primary" bindtap="getSystemInfo">获取设备信息</button>
```

```
<button type="primary" bindtap="onAccelerometerChange">监听重力感应数据</button>
```

```
<button type="primary" bindtap="onCompassChange">监听罗盘数据</button>
```

js

```
Page({
  data:{
    text:"Page system"
  },
  onLoad:function(options){
    // 页面初始化 options 为页面跳转所带来的参数
  },
  /**
```

```
* 获取当前网络状态

*/
getNetworkType: function() {
  wx.getNetworkType({
    success: function(res) {
      console.log(res)
    }
  })
},

/**

* 获取系统信息

*/
getSystemInfo: function() {
  wx.getSystemInfo({
    success: function(res) {
      console.log(res)
    }
  })
},

/**

* 监听重力感应数据

* - 带 on 开头的都是监听接收一个 callback

*/
onAccelerometerChange: function() {
  wx.onAccelerometerChange(function(res) {
    console.log(res)
  })
},
```

```
/**
 * 监听罗盘数据
 */
onCompassChange: function() {
  wx.onCompassChange(function(res) {
    console.log(res)
  })
},
onReady:function(){

  // 页面渲染完成

},
onShow:function(){

  // 页面显示

},
onHide:function(){

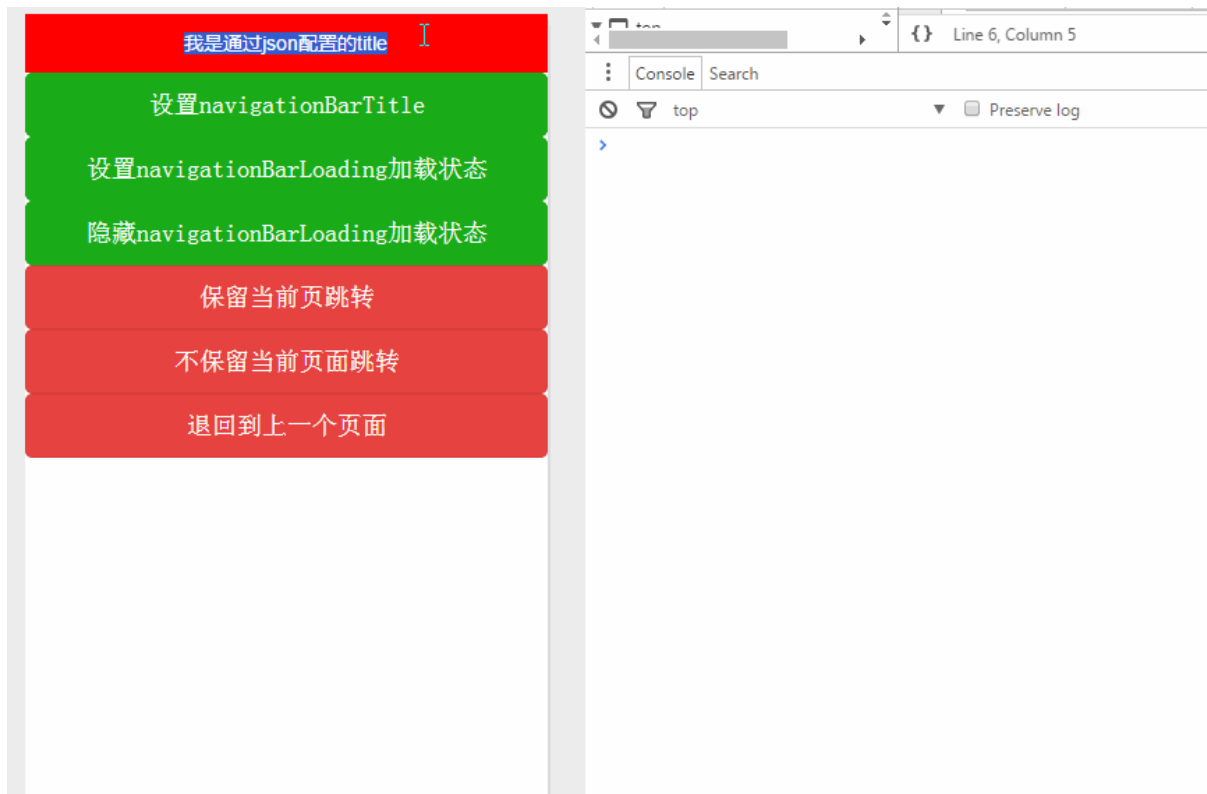
  // 页面隐藏

},
onUnload:function(){

  // 页面关闭

}
})
```

## 二: navigation API



演示效果也看到了小程序也就提供这几个处理导航控制。值得注意的是只能同时导航五个页面

**主要属性：**

导航条一些方法

`wx.setNavigationBarTitle(object)` 设置导航条的 Title

导航标题可以通过三种方式设置，第一种是通过全局配置名字统一，第二种就是在 page 中新建个 json 文件配置它会覆盖全局配置的 title，第三种就是通过 API 设置。

参数	类型	必填	说明
title	String	否	页面标题
success	Function	否	接口调用成功的回调函数
fail	Function	否	接口调用失败的回调函数
complete	Function	否	接口调用结束的回调函数（调用成功、失败都会执行）

•  
wx.showNavigationBarLoading()设置在导航条上显示 Loading 加载状态

wx.hideNavigationBarLoading() 隐藏 Loading 加载状态

导航视图的一些方法

wx.navigateTo() 保留当前页面进行跳转，传参时就像拼接 GET 参数一样即可，代码中会有体现

参数	类型	必填	说明
url	String	是	需要跳转的应用内页面的路径
success	Function	否	接口调用成功的回调函数
fail	Function	否	接口调用失败的回调函数
complete	Function	否	接口调用结束的回调函数（调用成功、失败都会执行）



wx.redirectTo() 销毁当前页面进行跳转，

参数	类型	必填	说明
url	String	是	需要跳转的应用内页面的路径
success	Function	否	接口调用成功的回调函数
fail	Function	否	接口调用失败的回调函数
complete	Function	否	接口调用结束的回调函数（调用成功、失败都会执行）

wx.navigateBack() 返回上一个页面

wxml

```
<button type="primary" bindtap="setNavigationBarTitle">设置 navigationBarTitle</button>

<button type="primary" bindtap="showNavigationBarLoading">设置 navigationBarLoading 加载状态</button>

<button type="primary" bindtap="hiddenNavigationBarLoading">隐藏 navigationBarLoading 加载状态</button>

<button type="warn" bindtap="navigateTo"> 保留当前页跳转</button>

<button type="warn" bindtap="redirectTo"> 不保留当前页面跳转</button>

<button type="warn" bindtap="navigateBack">退回到上一个页面</button>
```

json

```
{  
  "navigationBarTitleText": "我是通过 json 配置的 title"  
}
```

js

//导入另一个页面

```
var file = '../audio/audio'
```

```
Page({  
  data:{  
    text:"Page navigation"  
  },  
  onLoad:function(options){  
    // 页面初始化 options 为页面跳转所带来的参数  
  },  
  /**  
   * 设置 NavigationTitle  
   */  
  setNavigationBarTitle: function() {  
    wx.setNavigationBarTitle({  
      title: '我是通过 API 设置的 NavigationBarTitle'  
    })  
  },  
},
```

```
/**
 * 设置加载状态
 */
showNavigationBarLoading: function() {
  wx.showNavigationBarLoading()
},

/**
 * 隐藏加载状态
 */
hiddenNavigationBarLoading: function() {
  wx.hideNavigationBarLoading()
},

/**
 * 保留当前 Page 跳转
 */
navigateTo: function() {
  wx.navigateTo({

    //传递参数方式向 get 请求拼接参数一样

    url: file + '?phone=18939571&password=1992',
    success: function(res) {
      console.log(res)
    },
    fail: function(err) {
      console.log(err)
    }
  })
}
```

```
    })
  },
  /**
   * 关闭当前页面进行跳转当前页面会销毁
   */
  redirectTo: function() {
    wx.redirectTo({
      //传递参数方式向 get 请求拼接参数一样
      url: file + '?phone=189395719&password=1992'
    })
  },
  /**
   * 退回到上一个页面
   */
  navigateBack: function() {
    wx.navigateBack()
  },
  onReady: function() {
    // 页面渲染完成
  },
  onShow: function() {
    // 页面显示
  },
  onHide: function() {
    // 页面隐藏
  },
  onUnload: function() {
```

```
        // 页面关闭
    }
  })
```

### 三: animation API

动画水还是比较深的，这里只是简单介绍下小程序中动画的一些属性和注意事项，做动画前一定要整理好思路将动画一步步分解，再进行组合！这里只做引入。

#### **wx.createAnimation(object)**

- 看官方介绍
  - 1.创建一个动画实例 **animation**。调用实例的方法来描述动画。最后通过动画实例的 **export** 方法导出动画数据传递给组件的 **animation** 属性。
  - 2.调用动画操作方法后要调用 **step()** 来表示一组动画完成，可以在一组动画中调用任意多个动画方法，一组动画中的所有动画会同时开始，一组动画完成后才会进行下一组动画。**step** 可以传入一个跟 **wx.createAnimation()** 一样的配置参数用于指定当前组动画的属性

这还是比较好理解的比如第一条对应代码 **animation: this.animation.export()**

第二条比如缩放动画，也就说是一组 **scale, scaleX, scaleY...**为一缩放动画组的一个动画方法，缩放动画组和旋转动画组通过 **step()**链接，按顺序执行。代码中体验吧！看效果反过来看会更容易理解

主要属性:

参数	类型	必填	说明
duration	Integer	否	动画持续时间，单位ms，默认值 400
timingFunction	String	否	定义动画的效果，默认值"linear"，有效值："linear","ease","ease-in","ease-in-out","ease-out","step-start","step-end"
delay	Integer	否	动画持续时间，单位 ms，默认值 0 应该改为动画延迟多久开始
transformOrigin	String	否	设置transform-origin，默认为"50% 50% 0"

这里主要树下 timingFunction 和 transformOrigin

- timingFunction 设置动画效果
  - linear 默认为 linear 动画一直较为均匀
  - ease 开始时缓慢中间加速到快结束时减速
  - ease-in 开始的时候缓慢
  - ease-in-out 开始和结束时减速
  - ease-out 结束时减速
  - step-start 动画一开始就跳到 100% 直到动画持续时间结束 一闪而过
  - step-end 保持 0% 的样式直到动画持续时间结束 一闪而过
- transformOrigin 设置动画的基点 默认%50 %50 0
  - left,center right 是水平方向取值，对应的百分值为 left=0%;center=50%;right=100%
  - top center bottom 是垂直方向的取值，其中 top=0%;center=50%;bottom=100%

动画组及动画方法

样式：

样式：		
方法	参数	说明
opacity	value	透明度，参数范围 0~1
backgroundColor	color	颜色值
width	length	长度值，如果传入 Number 则默认使用 px，可传入其他自定义单位的长度值
height	length	长度值，如果传入 Number 则默认使用 px，可传入其他自定义单位的长度值
top	length	长度值，如果传入 Number 则默认使用 px，可传入其他自定义单位的长度值
left	length	长度值，如果传入 Number 则默认使用 px，可传入其他自定义单位的长度值
bottom	length	长度值，如果传入 Number 则默认使用 px，可传入其他自定义单位的长度值
right	length	长度值，如果传入 Number 则默认使用 px，可传入其他自定义单位的长度值

旋转：

方法	参数	说明
rotate	deg	deg的范围-180~180，从原点顺时针旋转一个deg角度
rotateX	deg	deg的范围-180~180，在X轴旋转一个deg角度
rotateY	deg	deg的范围-180~180，在Y轴旋转一个deg角度
rotateZ	deg	deg的范围-180~180，在Z轴旋转一个deg角度
rotate3d	(x,y,z,deg)	同 <a href="#">transform-function rotate3d</a>

缩放：



方法	参数	说明
scale	sx,[sy]	一个参数时，表示在X轴、Y轴同时缩放sx倍数；两个参数时表示在X轴缩放sx倍数，在Y轴缩放sy倍数
scaleX	sx	在X轴缩放sx倍数
scaleY	sy	在Y轴缩放sy倍数
scaleZ	sz	在Z轴缩放sy倍数
scale3d	(sx,sy,sz)	在X轴缩放sx倍数，在Y轴缩放sy倍数，在Z轴缩放sz倍数

偏移：

方法	参数	说明
translate	tx,[ty]	一个参数时，表示在X轴偏移tx，单位px；两个参数时，表示在X轴偏移tx，在Y轴偏移ty，单位px。
translateX	tx	在X轴偏移tx，单位px
translateY	ty	在Y轴偏移tx，单位px
translateZ	tz	在Z轴偏移tx，单位px
translate3d	(tx,ty,tz)	在X轴偏移tx，在Y轴偏移ty，在Z轴偏移tz，单位px

倾斜：

方法	参数	说明
skew	ax, [ay]	参数范围-180~180；一个参数时，Y轴坐标不变，X轴坐标延顺时针倾斜ax度；两个参数时，分别在X轴倾斜ax度，在Y轴倾斜ay度
skewX	ax	参数范围-180~180；Y轴坐标不变，X轴坐标延顺时针倾斜ax度
skewY	ay	参数范围-180~180；X轴坐标不变，Y轴坐标延顺时针倾斜ay度

矩阵变形：

方法	参数	说明
matrix	(a,b,c,d,tx,ty)	同transform-function matrix
matrix3d		同transform-function matrix3d

演示单个动画组效果



wxml

```
<view class="container">
  <view animation="{{animation}}" class="view">我在做动画</view>
</view>

<button type="primary" bindtap="rotate">旋转</button>
```

js

```
Page({
  data:{
    text:"Page animation",
    animation: ''
  },
  onLoad:function(options){

    // 页面初始化 options 为页面跳转所带来的参数

  },
  onReady:function(){

    // 页面渲染完成

    //实例化一个动画
    this.animation = wx.createAnimation({

      // 动画持续时间，单位 ms，默认值 400
      duration: 1000,
      /**
       * http://cubic-bezier.com/#0,0,.58,1
       * linear 动画一直较为均匀
       * ease 从匀速到加速再到匀速
       * ease-in 缓慢到匀速
       * ease-in-out 从缓慢到匀速再到缓慢
       *
       * http://www.tuicool.com/articles/neqMVr
       * step-start 动画一开始就跳到 100% 直到动画持续时间结束 一闪而过
```

```

    * step-end 保持 0% 的样式直到动画持续时间结束      一闪而过
    */
    timingFunction: 'linear',
    // 延迟多长时间开始
    delay: 100,
    /**
     * 以什么为基点做动画 效果自己演示
     * left,center right 是水平方向取值，对应的百分值为 left=0%;center=50%;right=100%
     * top center bottom 是垂直方向的取值，其中 top=0%;center=50%;bottom=100%
     */
    transformOrigin: 'left top 0',
    success: function(res) {
        console.log(res)
    }
    })
},
/**
 * 旋转
 */
rotate: function() {
    //顺时针旋转 10 度
    //
    this.animation.rotate(150).step()
    this.setData({

```

```
        //输出动画
        animation: this.animation.export()
    })
},

onShow:function(){

    // 页面显示

},
onHide:function(){

    // 页面隐藏

},
onUnload:function(){

    // 页面关闭

}
})
```

演示多个动画组效果



这里我们只需要更改以下代码即可

```
/**
 * 旋转
 */
rotate: function() {
    //两个动画组 一定要以 step() 结尾
}
```

\* 动画顺序 顺时针旋转 150 度>x,y 放大二倍>x, y 平移 10px>x,y 顺时针倾斜>改变样式和设置宽度宽度

```
*/
this.animation.rotate(150).step().scale(2).step().translate(10).step().skew(10).step().opacity(0.5).width(10).step({duration: 8000})
this.setData({

  //输出动画

  animation: this.animation.export()
})
}
```