

微信小程序简易全攻略《三》创建轮播图与数据请求，表单的创建、提交、与接收 ...

一：创建轮播图

应用中最常见的就是轮播图了，今儿个就讲讲**微信**小程序中轮播图的使用

轮播图，不外乎两个要素，跳转链接 和 图片地址

1. 设置数据

我在 pages/test/test.js 中添加如下数据

```
//test.js
```

```
//获取应用实例
```

```
var app = getApp()
```

```
Page({
```

```
  data: {
```

```
    imgUrls: [
```

```
      {
```

```
        link:'/pages/index/index',
```

```
        url:'http://img02.tooopen.com/images/20150928/tooopen_sy_143912755726.jpg'
```

```
      },{
```

```
    link:'/pages/logs/logs',
    url:'http://img06.tooopen.com/images/20160818/tooopen_sy_175866434296.jpg'
  },{
    link:'/pages/test/test',
    url:'http://img06.tooopen.com/images/20160818/tooopen_sy_175833047715.jpg'
  }
],
indicatorDots: true,
autoplay: true,
interval: 5000,
duration: 1000,
userInfo: {}
},
onLoad: function () {
  console.log('onLoad test');
}
```

```
}}
```

其中 `imgUrls` 是我们轮播图中将要用到的 图片地址和 跳转链接

`indicatorRots` 是否出现焦点

`autoplay` 是否自动播放

`interval` 自动播放间隔时间

`duration` 滑动动画时间

更多样式编辑请参看文

档 <https://mp.weixin.qq.com/debug/wxadoc/dev/component/swiper.html?t=1475052054228>

2. 部署到页面

找到 文件 `pages/test/test.wxml`

```
<swiper indicator-dots="{{indicatorDots}}"
```

```
    autoplay="{{autoplay}}" interval="{{interval}}" duration="{{duration}}">
```

```
<block wx:for="{{imgUrls}}">
```

```
    <swiper-item>
```

```
<navigator url="{{item.link}}" hover-class="navigator-hover">
```

```
<image src="{{item.url}}" class="slide-image" width="355" height="150"/>
```

```
</navigator>
```

```
</swiper-item>
```

```
</block>
```

```
</swiper>
```

注意：**swiper** 千万不要在外面 加上任何标签 例如 **<view>** 之类的，如果加了可能会导致轮播图出不来

3. 添加页面样式

创建文件 pages/test/test.wxss

```
.slide-image{
```

```
width: 100%;
```

```
}
```

加上上面的样式可以使 轮播图的宽度达到 100% 然后更漂亮点，当然可以根据自己的喜好罗！

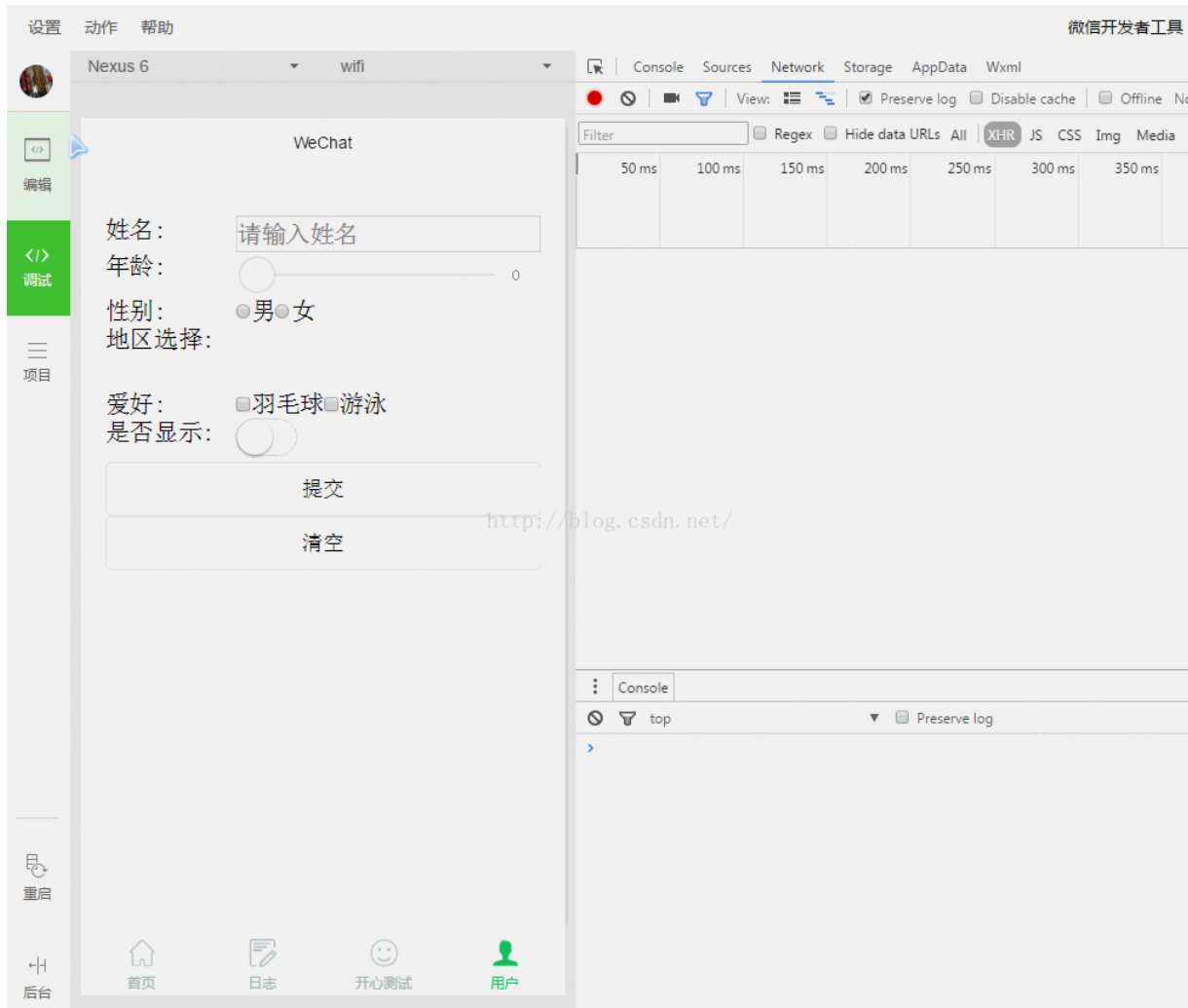
看效果



二：数据请求 表单的创建 提交 与接收

开始正题了，本节小小研究了下 **微信** 小程序的表单创建与提交

先看看效果



1. 表单页面

<view id="adduser">

<form bindsubmit="formSubmit" bindreset="formReset">

<view class="section">

```
<view class="section__title">姓名:</view>
```

```
<view class='form-group'>
```

```
  <input type="text" class="input-text" name="username" placeholder="请输入姓名" />
```

```
</view>
```

```
</view>
```

```
<view class="section section_gap">
```

```
  <view class="section__title">年龄:</view>
```

```
  <view class='form-group'>
```

```
    <slider name="age" show-value > </slider>
```

```
  </view>
```

```
</view>
```

```
<view class="section section_gap">
```

```
  <view class="section__title">性别:</view>
```

```
  <view class='form-group'>
```

```
    <radio-group name="gender">
```

```
      <label> <radio value="1"/>男</label>
```

```
<label> <radio value="0"/>女</label>
```

```
</radio-group>
```

```
</view>
```

```
</view>
```

```
<view class="section">
```

```
<view class="section__title">地区选择:</view>
```

```
<view class='form-group'>
```

```
<picker bindchange="bindPickerChange" value="{{index}}" range="{{array}}">
```

```
<view class="picker">
```

```
<input type="hidden" disabled="true" name="addr" value="{{array[index]]}}"/>
```

```
</view>
```

```
</picker>
```

```
</view>
```

```
</view>
```

```
<view class="section section_gap">
```

```
<view class="section__title">爱好:</view>
```



```
<view class='form-group'>
  <checkbox-group name="hobby">
    <label> <checkbox value="羽毛球"/>羽毛球</label>
    <label> <checkbox value="游泳"/>游泳</label>
  </checkbox-group>
</view>
```

```
</view>
```

```
<view class="section section_gap">
  <view class="section__title">是否显示:</view>
  <view class='form-group'>
    <switch name="isshow"/>
  </view>
</view>
```

```
</view>
```

```
<view class="section btn-area">
  <button formType="submit">提交</button>
```

```
<button formType="reset">清空</button>
```

```
</view>
```

```
</form>
```

```
<!-- 黑框提示并消失 -->
```

```
<toast hidden="{{toast1Hidden}}" bindchange="toast1Change">
```

```
  {{notice_str}}
```

```
</toast>
```

```
<!-- 确认框 及 提示框 -->
```

```
<view class="page_bd">
```

```
  <modal title="确认" confirm-text="确定" cancel-text="取消" hidden="{{modalHidden}}" mask
```

```
  bindconfirm="confirm_one" bindcancel="cancel_one">
```

```
    确认提交么？
```

```
  </modal>
```

```
  <modal class="modal" hidden="{{modalHidden2}}" no-cancel bindconfirm="modalChange2"
```

```
  bindcancel="modalChange2">
```

```
    <view> 提示 </view>
```

```
<view> 清空成功 </view>
```

```
</modal>
```

```
</view>
```

```
</view>
```

2.js 代码

```
var app = getApp();
```

```
Page({
```

```
  data:{
```

```
    // text:"这是一个页面"
```

```
    array:["中国","美国","巴西","日本"],
```

```
    toast1Hidden:true,
```

```
    modalHidden: true,
```

```
    modalHidden2: true,
```

```
    notice_str: "",
```

```
    index:0
```

```
  },
```

```
toast1Change:function(e){  
    this.setData({toast1Hidden:true});  
  
},  
  
//弹出确认框  
modalTap: function(e) {  
    this.setData({  
        modalHidden: false  
    })  
},  
  
confirm_one: function(e) {  
    console.log(e);  
    this.setData({  
        modalHidden: true,  
        toast1Hidden:false,  
        notice_str: '提交成功'  
    });  
};
```

```
},  
  
cancel_one: function(e) {  
    console.log(e);  
    this.setData({  
        modalHidden: true,  
        toast1Hidden:false,  
        notice_str: '取消成功'  
    });  
},  
  
//弹出提示框  
  
modalTap2: function(e) {  
    this.setData({  
        modalHidden2: false  
    })  
},  
  
modalChange2: function(e) {
```

```
this.setData({  
  modalHidden2: true  
})  
  
,  
  
bindPickerChange: function(e) {  
  console.log('picker 发送选择改变，携带值为', e.detail.value)  
  
  this.setData({  
    index: e.detail.value  
  })  
},  
  
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  
},
```

```
onShow:function(){  
    // 页面显示  
},  
onHide:function(){  
    // 页面隐藏  
},  
onUnload:function(){  
    // 页面关闭  
},  
formSubmit: function(e) {  
    var that = this;  
    var formData = e.detail.value;  
    wx.request({  
        url: 'http://test.com:8080/test/socket.php?msg=2',  
        data: formData,  
        header: {
```

```
        'Content-Type': 'application/json'

    },

    success: function(res) {

        console.log(res.data)

        that.modalTap();

    }

})

},

formReset: function() {

    console.log('form 发生了 reset 事件');

    this.modalTap2();

}

})
```

3. 部分样式

```
#adduser{

    background: #f5f5f5;
```



```
}
```

```
.section__title{
```

```
    float: left;
```

```
    width:30%;
```

```
}
```

```
.form-group{
```

```
    float: right;
```

```
    width: 70%;
```

```
}
```

```
.section{
```

```
    clear: both;
```

```
    width:90%;
```

```
    margin: 2rem auto;
```

```
}
```

```
.input-text{
```

```
    border: 1px solid #ddd;
```

```
}  
  
.button{  
    border: 1px solid #1aad19;  
    border-radius: 2px;  
}  
  
.picker{  
    padding: 13px;  
    background-color: #FFFFFF;  
}
```

4. 服务器端

```
<?php  
  
var_dump($_REQUEST);
```

本节 集合了

表

单 <https://mp.weixin.qq.com/debug/wxadoc/dev/component/form.html?t=1476197486816>

数据请求 <https://mp.weixin.qq.com/debug/wxadoc/dev/api/network-request.html?t=1476197484427>

提示

框 <https://mp.weixin.qq.com/debug/wxadoc/dev/component/toast.html?t=1476197486420>

确认和非确认框

<https://mp.weixin.qq.com/debug/wxadoc/dev/component/modal.html?t=1476197489278>

以及相关表单 的信息 ， 之后将分解讲解 对应的小功能。