

# BBS

姓名： 傅琦 学号： 2017210218007

## 1、项目需求

设计一个“BBS”软件，要求实现以下功能：

- 登录识别身份 (Log in)
- 浏览所有根问题(browseroot)
- 浏览对应问题的回复(browereplyerbyid)
- 回复(reply)
- 删除自己的留言(deletproblembyid)
- 新建问题(createproblem)

## 2、功能分析

- (1) 登录功能是实现用户验证账户秘密的操作，系统会对用户的账户信息进行验证，通过验证后才可以进行下面的操作。
- (2) 浏览所有根问题的功能是通过 API 请求得到包含所有根问题的数据集，在 WXML 上进行列表渲染，设计成人性化。
- (3) 浏览回复功能是当用户点击一个问题时候，能够获取当前问题的 userid，并使用当前 userid 为 parentid 进行发起 API 请求，从而获得该问题的所有回复。
- (4) 删除留言功能需要用户知道哪天留言为自己创建的前提下才能进行删除，因此使用条件渲染功能，把 userid 与当前用户一致的留言添加一个“删除”的按钮以及监听器，使用户能够完成删除操作。

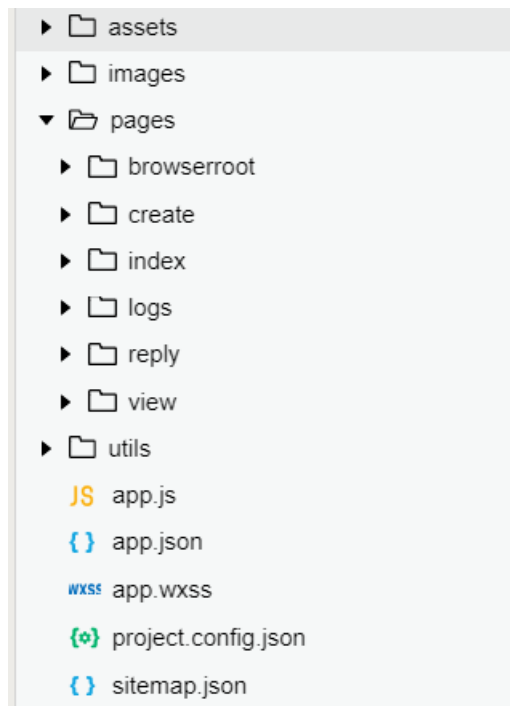
## 3、总体设计

将程序分成三个模块，将对应的操作和按钮接入到对应的页面上。达到用户可以在三个界面上无缝衔接，并且要求程序的逻辑关系符合人性化设计，与平时使用的话题贴的使用习惯保持一致，增强其实用性。

## 4、实现原理

该程序的核心操作使用全局变量的赋值和使用便利性，将要传递的参数赋给全局变量，在后续操作时候能方便快捷地取用。根问题渲染和回复渲染的视图层是程序最重要的设计，使用 `wx:for` 进行循环渲染，以达到根据实际情况渲染所需要的列表数。

## 5、模块划分

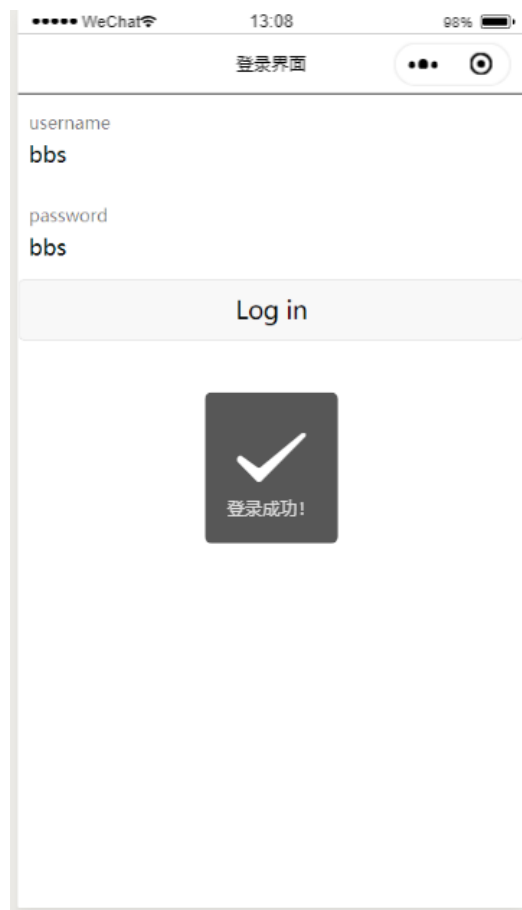
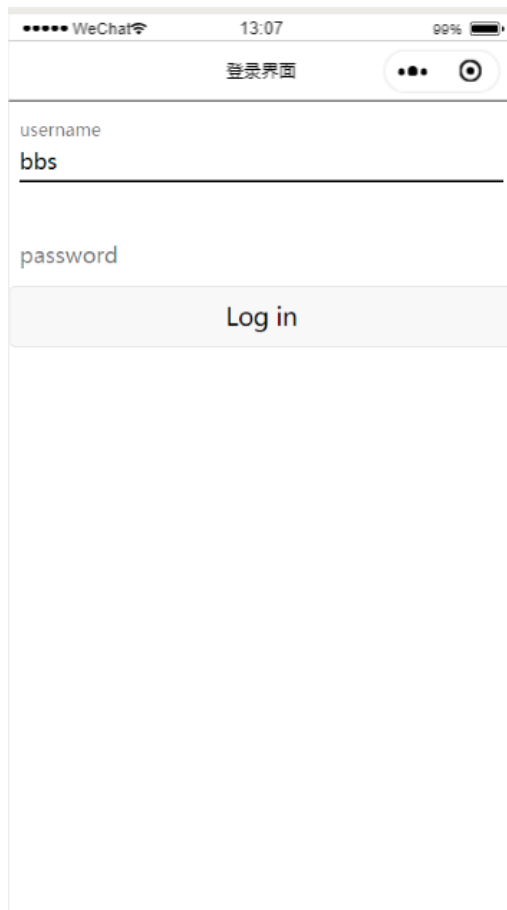


大致分为三个模块：即登录模块、根贴操作模块、回复操作模块。登录模块只负责验证用户账户密码；根贴操作模块负责浏览所有根问题、创建跟问题以及跳转根问题的回复；回复模块负责打印所有回复、删除自己的回复和添加回复的操作。

## 6、各模块实现过程

### a) 模块一：登录操作的实现

#### I. 设计 UI



## II. 各类算法实现

(1) 在 `wxml` 文件上使用 `input` 组件，在屏幕上打印出登录账号和密码的输入框，用户输入的账号信息会赋值到对应的 `username` 和 `userpassword` 上。

(2) 当用户点击“log in”按钮，`js` 文件会发起请求，使用 `API` 连接服务器，验证用户的信息，成功登录后，会返回用户的基础信息。

(3) 在 `app.json` 文件中声明全局变量 `app.myData.userInfo`，返回的信息会被赋值到全局变量中，以便于后续操作使用作者的 `id` 信息。

## III. 各类实现代码

Logs.js:

```
var app = getApp()
Page({
  data: {
    username: '',
    password: ''
  },
```

```

bindusername: function (e) {
  console.log(e.detail.value)
  this.setData({
    username: e.detail.value
  })
},
bindpassword: function (e) {
  console.log(e.detail.value)
  this.setData({
    password: e.detail.value
  })
},

Log: function () {
  wx.request({
    url: 'http://172.22.130.33/index.php/Ajaxapi/Ajaxapi/login' + '/username/' + this.data.username
    + '/password/' + this.data.password,
    success: function (res) {
      console.log(res.data.data)
      if (res.data.info == 1){
        wx.showToast({
          title: '登录成功!',
        })
        app.myData.userInfo = { myid: res.data.data.id }
        wx.navigateTo({
          url: '/pages/browserroot/browserroot',
        })
      }
      else{
        wx.showToast({
          title: '密码或账号错误',
        })
      }
    }

  })

}

},
onLoad: function () {
}
})
Log.json:
{
  "navigationBarTitleText": "登录界面",

```

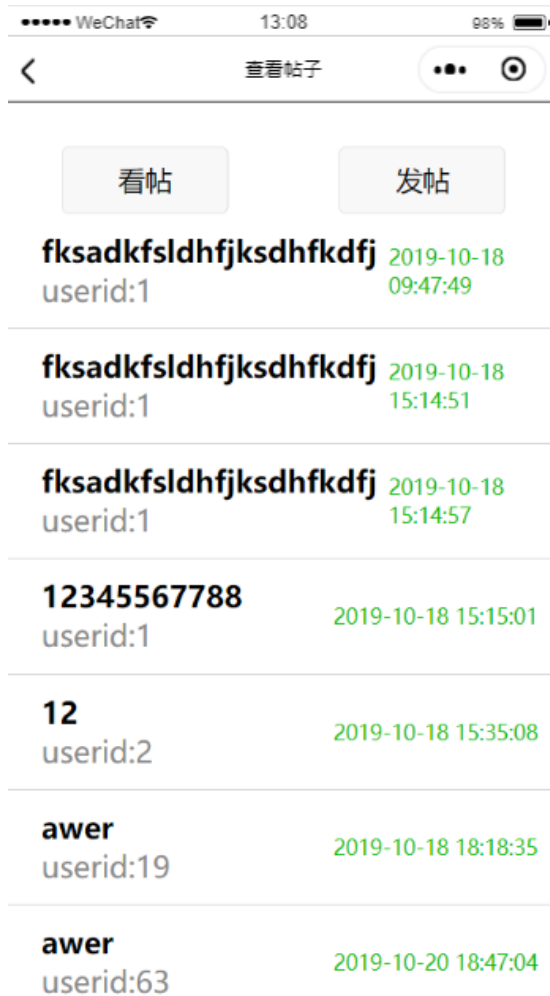
```
"navigationBarBackgroundColor": "#FFFFFF",
"navigationBarTextStyle": "black",
"usingComponents": {
  "sc-input": "/images/scInput/sc-input",
  "sc-textarea": "/images/scTextarea/sc-textarea"
}
}
```

Logs.wxml:

```
<!--logs.wxml-->
<view class="ui-test-container">
  <view class="ui-test-item">
    <view class="ui-test-item-subheader">
    </view>
    <view class="ui-test-item-content">
      <sc-input sc-class="sc-input" label="username" bindinput="bindusername" ></sc-input>
    </view>
    <view class="ui-test-item-content">
      <sc-input sc-class="sc-input" label="password" bindinput="bindpassword"></sc-input>
    </view>
  </view>
</view>
<button bindtap="Log" style="default">Log in</button>
```

b) 模块二：根问题的加载

I. 设计 UI



## II. 各类算法实现

(1) 在 wxml 文件中设计“看贴”和“发帖”两个按钮，跳转到该页面后，需要用户手动选择看贴还是发帖；

(2) 当用户点击“看贴”按钮时，js 文件会发起请求跟问题的操作，并将 pid=0 的根问题的数据返回，并以渲染列表的方式呈现给用户。同时系统会给每行列表添加一个点击监听器，当用户想要查看某个话题的评论时，只需要点击该行，系统会通过 `data-userid = "{{item.userid}}"` 记录下创建当前话题的用户 id，并创建全局变量，传递到下一个页面。

(3) 当用户点击“发帖”按钮时，页面会跳出一个输入框，以便用户发表根贴。创建完成后将自动返回到看贴界面。

## III. 各类代码实现

js:  
`var app = getApp()`

```

Page({

  data: {
    browerRoot:'',
    pid:''

  },
  BrowserRoot:function() {
    var that = this
    wx.request({
      url: 'http://172.22.130.33/index.php/Ajaxapi/Ajaxapi/browserroot',
      success: function(res){
        console.log(res.data.data)
        if(res.data.info == 1){
          that.setData({
            browerRoot:res.data.data
          })
        }
      }
    })
  },
  toview: function (e) {
    var that = this
    that.setData({
      pid: e.currentTarget.dataset.userid
    })
    console.log(this.data.pid)
    wx.request({
      url: 'http://172.22.130.33/index.php/Ajaxapi/Ajaxapi/browsereplybyid',
      data: {
        pid: this.data.pid
      },
      success: function (res) {
        console.log(res.data.data)
        if(res.data.info == 1){
          app.brower.userInfo = { myData: res.data.data, mypid: e.currentTarget.dataset.userid}
          wx.navigateTo({
            url: '/pages/view/view',
          })
        }
      }
    })
  },
  createproblem:function() {

```

```

wx.navigateTo({
  url: '/pages/create/create',
})
}
})
json:
{
  "navigationBarTitleText": "查看帖子",
  "navigationBarBackgroundColor": "#FFFFFF",
  "navigationBarTextStyle": "black",
  "usingComponents": {
    "sc-cell": "/assets/lib/scui/dist/components/scCell/sc-cell",
    "sc-cell-group": "/assets/lib/scui/dist/components/scCellGroup/sc-cell-group",
    "sc-checkbox": "/assets/lib/scui/dist/components/scCheckbox/sc-checkbox",
    "sc-checkbox-group": "/assets/lib/scui/dist/components/scCheckboxGroup/sc-checkbox-group",
    "sc-switch": "/assets/lib/scui/dist/components/scSwitch/sc-switch",
    "sc-radio": "/assets/lib/scui/dist/components/scRadio/sc-radio",
    "sc-radio-group": "/assets/lib/scui/dist/components/scRadioGroup/sc-radio-group"
  }
}
wxml:
<form bindsubmit="formBindsubmit" bindreset="formReset">
<view style="display:flex;margin-top:30px;">
<button style="width:30%;" formType="submit" bindtap="BrowserRoot">看帖</button>
<button style="width:30%" formType="reset" bindtap="createproblem">发帖</button>
</view>
</form>

<view class="ui-test-container">
<view class="ui-test-item">
<block wx:for="{{browerRoot}}" wx:key="key" wx:for-item = "item">
<view class="ui-test-item-content" bindtap="toview" data-userid =
"{{item.userid}}"><!--data-userid 记录 userid-->
<sc-cell-group>
<sc-cell right-icon="iconfont icon-right icon-right-color" left-icon="iconfont icon-cloudqueue
icon-cloudqueue-color">
<text style="color: color: RGB(53,53,53);font-weight:bold;font-size:20px;
">{{item.problem}}</text>
<text slot="subheader" style="color:
RGB(136,136,136);font-size:20px;">userid:{{item.userid}}</text>
<text slot="righttext" style="color: RGB(9,187,7);">{{item.createtime}}</text>
</sc-cell>
</sc-cell-group>
</view>

```



```

</block>
</view>
</view>
WXSS:
.icon-right-color{
color: #66c6f2;
}

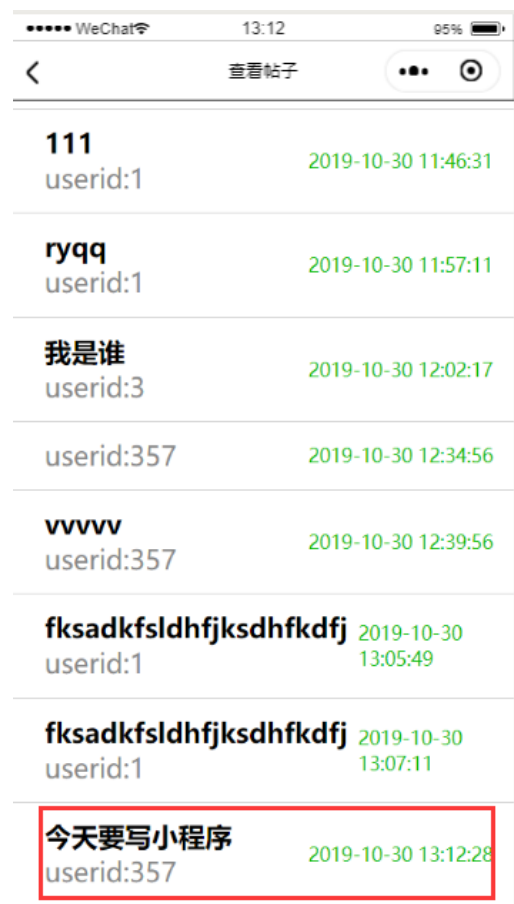
.icon-cloudqueue-color{
color: #47a86c;
}

.redText{
color:white;
background-color: #0b93d5;
}

```

### C) 模块三：创建问题

#### I、设计 UI



#### II、各类算法实现

(1) 用户在输入框输入帖子内容后，点击“创建”按钮，可以自动跳转回查看帖子内容页面。

### III、实现代码

js 文件：

```
var app = getApp()
Page({

  data: {
    problem: ''
  },
  bindFormSubmit: function (e) {
    console.log(e.detail.value)
    this.setData({
      problem: e.detail.value
    })
  },
  bindcreate:function(){
    wx.request({
      url: 'http://172.22.130.33/index.php/Ajaxapi/Ajaxapi/createproblem/',
      data:{
        userid: app.globalData.userInfo.myid,
        problem:this.data.problem
      },
      success:function(res){
        console.log(res.data.data)
        wx.showToast({
          title: '创建成功!',
        })
        wx.navigateBack({
          url:'/pages/browserroot/browserroot'
        })

      }
    })
  }
})
```

json 文件：

```
{
  "navigationBarTitleText": "创建帖子",
  "navigationBarBackgroundColor": "#FFFFFF",
```

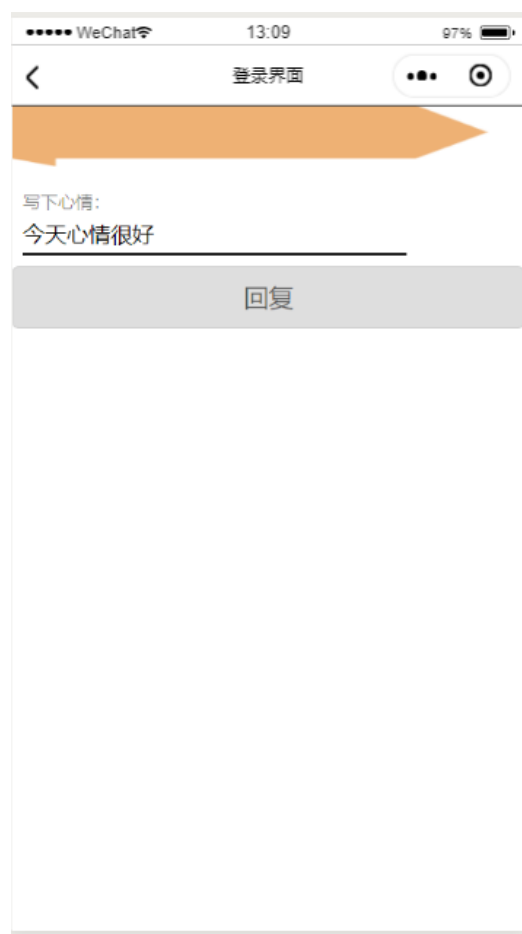
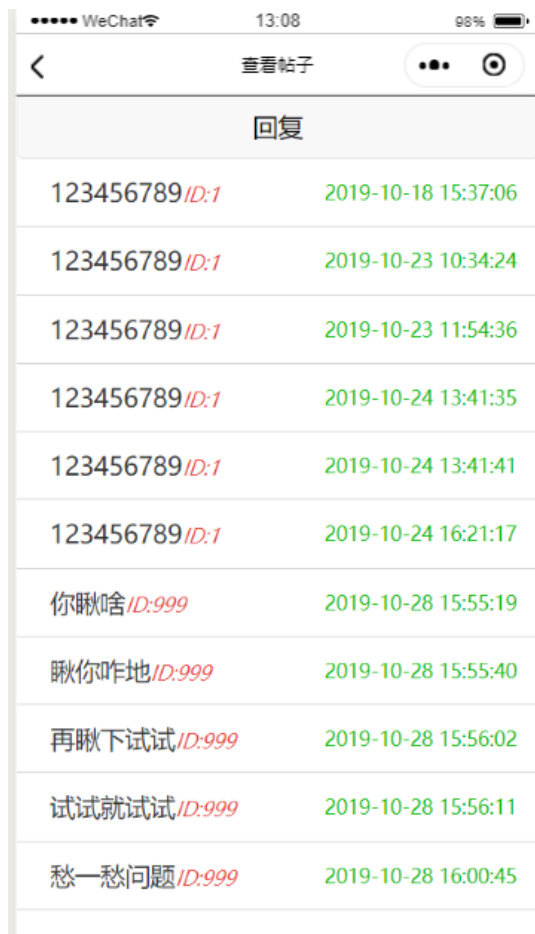
```
"navigationBarTextStyle": "black",
"usingComponents": {
  "sc-input": "/images/scInput/sc-input",
  "sc-textarea": "/images/scTextarea/sc-textarea"
}
}
```

wxml 文件

```
<view>
  <image src="/images/biaoti.png" class="tp1"></image>
</view>
<view class="ui-test-container">
  <view class="ui-test-item">
    <view class="ui-test-item-subheader">
    </view>
    <view class="ui-test-item-content">
      <sc-input sc-class="sc-input" label="写下心情: " bindinput="bindFormSubmit"></sc-input>
    </view>
  </view>
</view>
<button form-type="submit" bindtap="bindcreate"> 提交 </button>
```

D)模块四：回复(查看回复、添加回复、删除回复)

I:设计 UI





## II: 算法实现

- (1) 查看回复的界面有三个基础功能：查看所有回复、添加回复、删除自己的回复。
- (2) 查看所有回复是使用的的列表渲染功能；broweroot 文件中再跳转至当前页面时，以及发起了 API 请求，并将该问题的回复数据传递到当前页面。
- (3) 用户点击“回复”时，可以为该问题添加自己的回复，并返回当前页面。
- (4) 而在渲染时候，添加一个条件渲染，当该回复是当前用户添加的时候，会在该条回复下生成一个“删除”按钮；用户点击该按钮可以出发删除操作的 API 请求, 并根据用户 id 删除该问题的下的当前条回复。

## III: 实现代码。

- (1) 查看及删除回复：

js 文件

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

```

Page({

  data: {
    thisUserId: '',//{{app.myData.userInfo.myid}},
    data: '',//{{app.brower.userInfo.thisData}},
    pid: '',//{{app.brower.userInfo.mypid}}'
  },
  onShow: function (options) {
    this.setData({
      thisUserId: app.myData.userInfo.myid
    })
    this.setData({
      data: app.brower.userInfo.myData
    })
    this.setData({
      pid: app.brower.userInfo.mypid
    })
  },

  myDelet: function() {
    console.log(this.data.thisUserId)
    console.log(this.data.pid)
    wx.request({
      url: 'http://172.22.130.33/index.php/Ajaxapi/Ajaxapi/deleteproblembyid',
      data:{
        userid: this.data.thisUserId,
        pid: this.data.pid
      },
      success:function(res){
        console.log(res.data)
        wx.showToast({
          title: '删除成功!'
        })
      }
    })
  },
  reply:function(){
    wx.navigateTo({
      url: '/pages/reply/reply',
    })
    wx.request({
      url:
        'http://172.22.130.33/index.php/Ajaxapi/Ajaxapi/reply/userid/1/problem/123456789/parentid/1',
      data:{

```

```

userid: this.data.thisUserId,
problem: this.data.problem,
parentid: this.data.pid
},
success: function(res){

}
})
}

```

```

})

```

wxml 文件:

```

<view class="ui-test-container">
<view class="ui-test-item">
<button bindtap="reply" style="default">回复</button>
<view wx:for="{{data}}" wx:key="key" wx:for-item="item">
<view class="ui-test-item-content" ><!--data-userid 记录 userid-->
<block wx:if="{{item.userid == thisUserId}}">
<sc-cell-group>
<sc-cell right-icon="iconfont icon-right" left-icon="iconfont icon-cloudqueue"
bindtap="myDelet">
<text style="color: RGB(53,53,53);font-size:18px;">{{item.problem}}</text>
<text style="font-style:italic;color:RGB(230,67,64)">ID:{{item.userid}}</text>
<text slot="righttext" style="color: RGB(9,187,7);">删除{{item.createtime}}</text>
</sc-cell>
</sc-cell-group>
</block>
<block wx:else="{{item.userid != thisUserId}}">
<sc-cell-group>
<sc-cell right-icon="iconfont icon-right" left-icon="iconfont icon-cloudqueue">
<text style="color: RGB(53,53,53);font-size:18px;">{{item.problem}}</text>
<text style="font-style:italic;color:RGB(230,67,64)">ID:{{item.userid}}</text>
<text slot="righttext" style="color: RGB(9,187,7);">{{item.createtime}}</text>
</sc-cell>
</sc-cell-group>
</block>
</view>
</view>
</view>
</view>

```

(2) 回复

js 文件:

```

var app = getApp()
Page({

```

```

data: {
  problem: ''
},

bindFormSubmit: function (e) {
  console.log(e.detail.value)
  this.setData({
    problem: e.detail.value
  })
},
bindreply:function(){
  wx.request({
    url: 'http://172.22.130.33/index.php/Ajaxapi/Ajaxapi/reply/',
    data:{
      userid: app.myData.userInfo.myid,
      problem: this.data.problem,
      parentid: app.brower.userInfo.mypid
    },
    success:function(res){
      wx.showToast({
        title: '回复成功! ',
      })
      wx.navigateBack({
        url:'/pages/view/view'
      })
    }
  })
}
}

```

json 文件:

```

{
  "navigationBarTitleText": "添加回复",
  "navigationBarBackgroundColor": "#FFFFFF",
  "navigationBarTextStyle": "black",
  "usingComponents": {
    "sc-input": "/images/scInput/sc-input",
    "sc-textarea": "/images/scTextarea/sc-textarea"
  }
}

```

wxml 文件:

```

<view>
<image src="/images/biaoti.png"class="tp1"></image>

```



```
</view>
<view class="ui-test-container">
  <view class="ui-test-item">
    <view class="ui-test-item-subheader">
      </view>
      <view class="ui-test-item-content">
        <sc-input sc-class="sc-input" label="写下心情: " bindinput="bindFormSubmit"></sc-input>
      </view>
    </view>
  </view>
  <button form-type="submit" bindtap="bindreply"> 回复 </button>
```