

## 23 聊天容器组件开发

更新时间：2019-09-09 10:47:27



“

没有引发任何行动的思想都不是思想，而是梦想。

—— 马丁

”

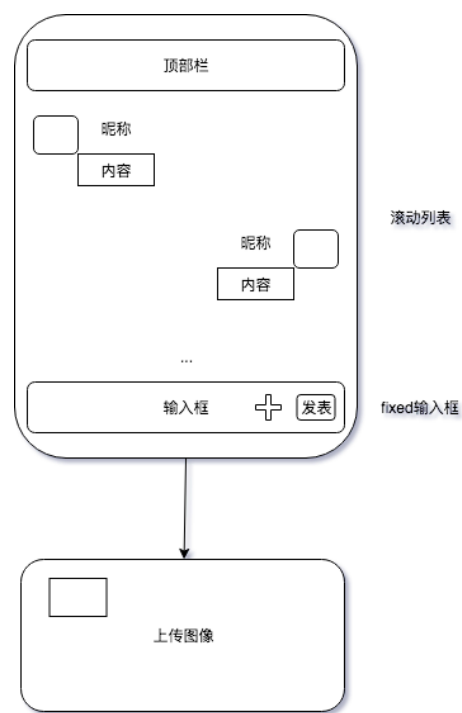
接下来就要进入私信聊天页面的开发工作了，这个页面我相信用过微信的同学都知道这个页面的交互，没错，就是模仿微信聊天来做的，只是减少了部分功能，接下来就进入页面开发吧。

本章节完整源代码地址，大家可以事先浏览一下：

[Github](#)

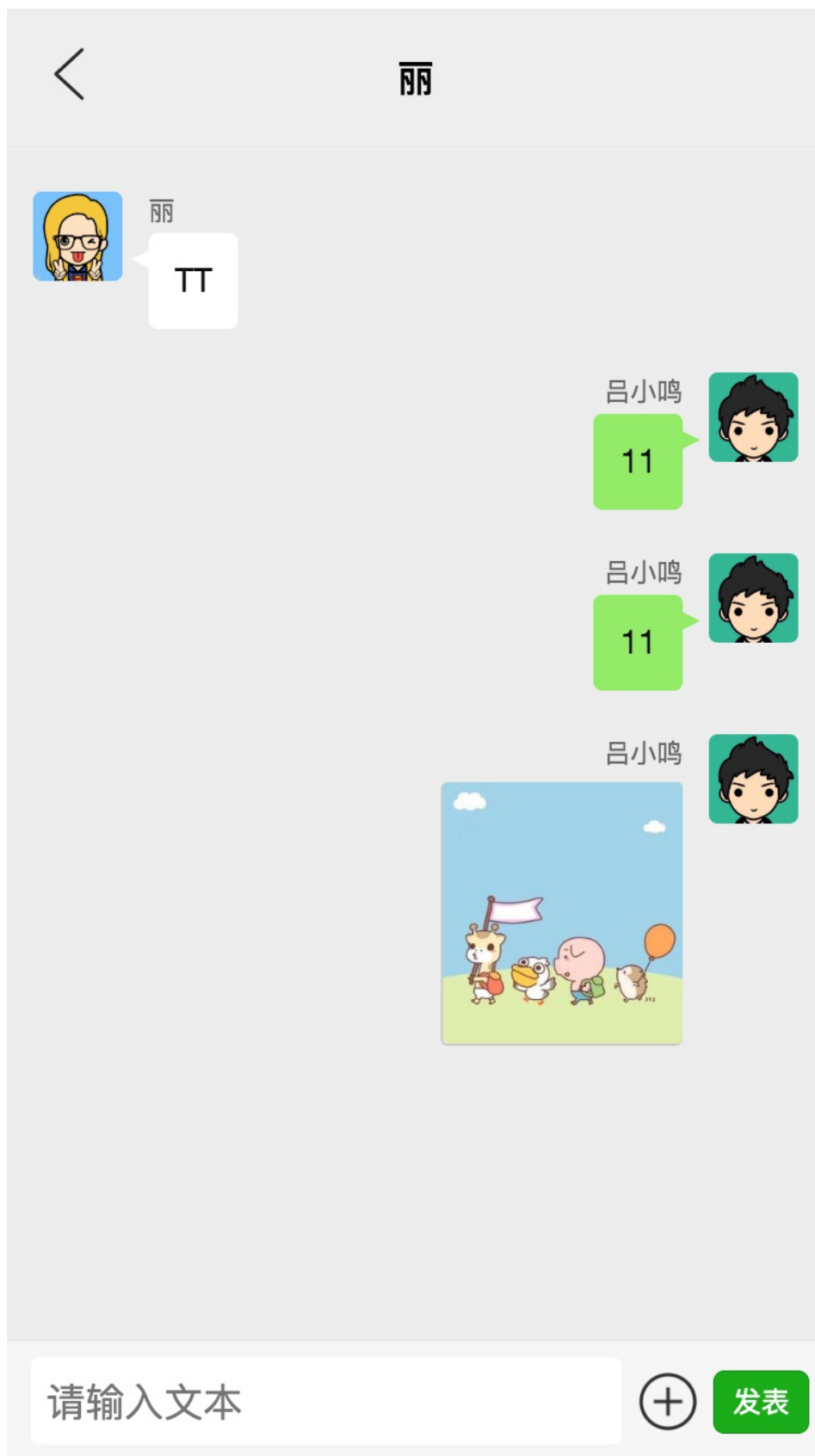
聊天页面交互图

我们先来看一下页面的逻辑交互图，基本上和微信聊天的交互一致，省去了发送语音和视频表情的功能，包括顶部的 `navHeader`，以及中间的聊天容器内容区域，以及下面的输入框，如下图：



聊天界面UI效果图

然后是页面的UI图，我们将会按照这个UI来实现页面，如下图：



## 聊天容器组件

首先，我们来开发聊天容器这个模块，同时包括了引入公用的 `navHeader` 组件和 公用的 `inputBar` 组件，其中 `navHeader` 在之前章节已经开发完成了，直接引入使用即可，但是 `inputBar` 组件，我们之前在朋友圈页面已经开发过，在这里我们还需要针对聊天页面添加一些功能，这里先引入，后面会讲解。

在前端项目的 **views** 文件夹下新建 **chat** 文件夹，同时新建 **index.vue**：

下面开始编写 **template** 内容，这段代码主要是整个页面的UI，包括了引入顶部使用公共的 **navHeader**，中间的聊天容器界面 **chatView**，以及下面的输入框 **inputBar** 组件。

```
<template>
  <div class="container">
    <navHeader :title="topName"/>
    <div ref="chatView" class="chat-view" @touchstart="touchstart">
      <chatItem v-for="(item) in dataList" :data="item" :key="item._id"/>
    </div>
    <div :class="bottomClass" :style="bottomStyle">
      <inputBar
        ref="inputBar"
        :option="{}"
        @publish="publish"
        @uploaded="uploaded"
        @showBottom="showBottom"
        @hideBottom="hideBottom"
        @hideBottomOnPanel="hideBottomOnPanel"
      />
    </div>
  </div>
</template>
```

其中，中间的 **chatView** 容器是一个可滚动的区域，也就是说当内容超过时，会出现滚动条，并且是一个局部滚动，即只在 **chatview** 这部分区域滚动，**chatview** 里面的内容是一条条聊天组件，来自 **v-for** 指令循环遍历渲染，我们采用局部滚动来实现这个容器，代码如下：

```
.container {
  display: flex;
  position: absolute;
  right: 0;
  left: 0;
  top: 0;
  bottom: 0;
  flex-direction: column;
}

.bottom-view {
  width: 100%;
  height: 56px;

  overflow: hidden;
  transition: height 200ms;
}

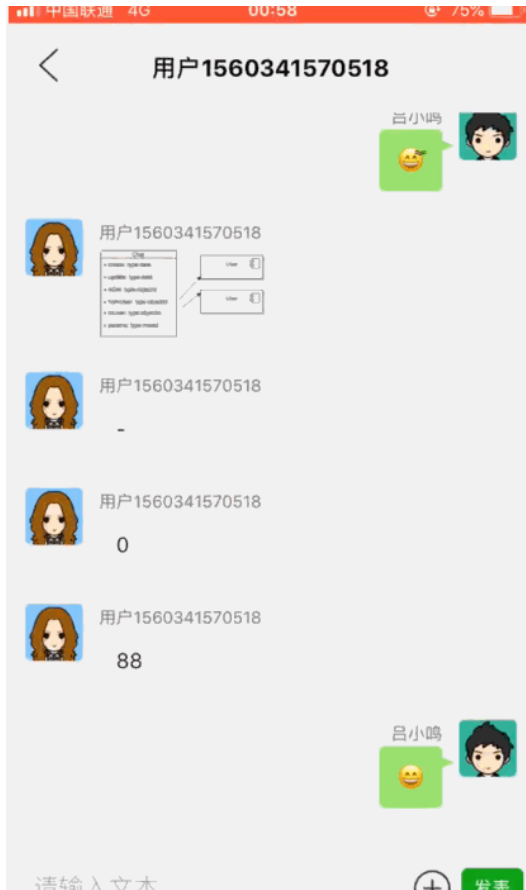
.chat-view {
  padding-top: 64px;
  flex: 1;
  background-color: rgb(237, 237, 237);

  overflow: auto;
  -webkit-overflow-scrolling: touch;
  padding-bottom: 30px;
}
```

上面这段代码包含以下逻辑和知识点：

1. 页面整体采用绝对定位，同时是 **display:flex** 布局，这样顶部 **navHeader** 和底部 **inputBar** 设置固定值，剩余的 **chatview** 设置 **flex:1**，高度可以达到适配效果。
2. **overflow: auto**；可以实现一个 **div** 的局部滚动。

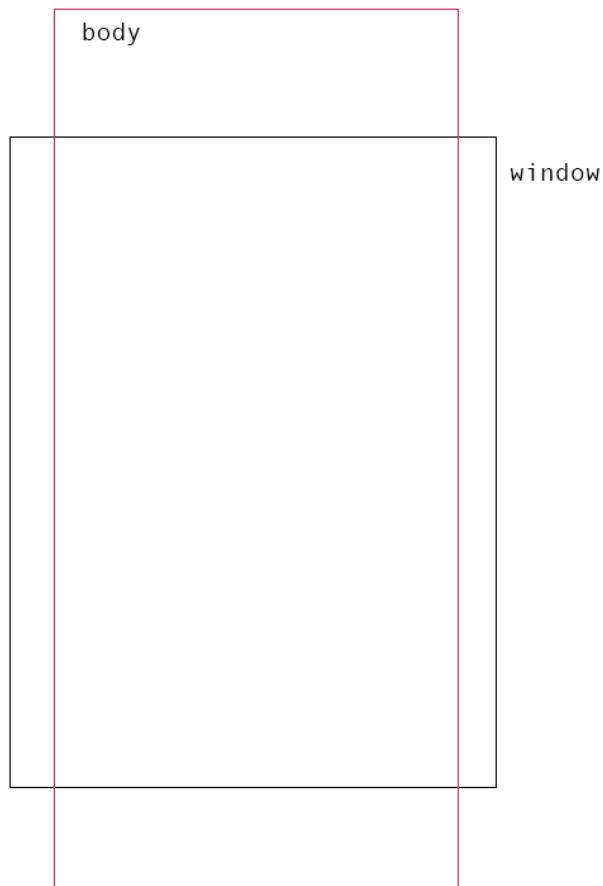
3. `-webkit-overflow-scrolling: touch;` 表示在iOS上，使用顺滑滚动，有弹性。效果如下图：



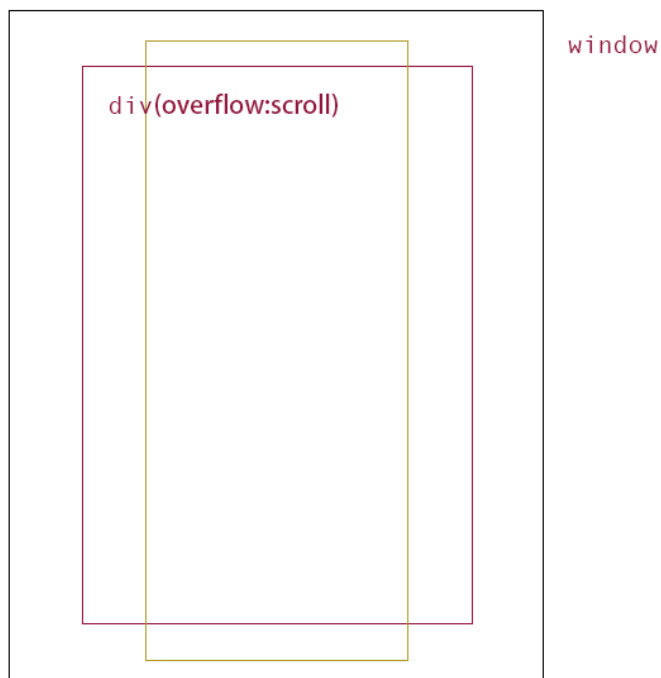
#### 移动web滚动讲解

针对移动端的滚动方式，以及滚动的原理，总结一下：

- **body滚动**：意思就是页面的高度由内容自动撑大，**body** 自然形成滚动条，一般如果不做处理，默认用的都是body 滚动。



- **局部滚动**：意思就是我们的滚动在一个固定宽高的 `div` 内触发，将该`div`设置成 `overflow:scroll/auto` 来形成`div`内部的滚动，滚动条出现在`div`内部。

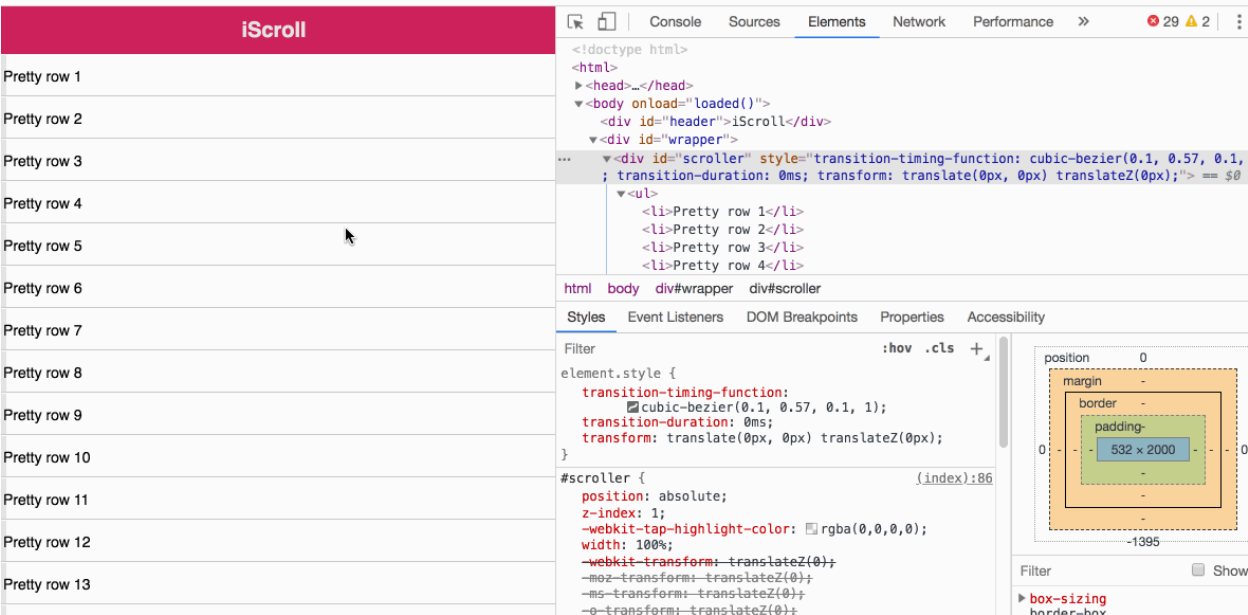


- **模拟滚动**：\*\*最典型的例子就是`iscroll`了，原理一般有两种：
- 1). 监听滚动元素的 `touchmove` 事件，当事件触发时修改元素的 `transform` 属性来实现元素的位移，让手指离开时触发 `touchend` 事件，然后采用 `requestAnimationFrame` 来在一个线型函数下不断的修改元素的 `transform` 来实现手指离开时的一段惯性滚动距离。
- 2). 监听滚动元素的 `touchmove` 事件，当事件触发时修改元素的 `transform` 属性来实现元素的位移，让手指离开时触发 `touchend` 事件，然后给元素一个 `css` 的 `animation`，并设置好 `duration` 和 `function` 来实现手指离开时的一段惯

性距离。

在使用模拟滚动时，浏览器在 javascript 层面会消耗更多的性能去改变 dom 元素的位置，在 dom 复杂层级深的页面更为高，所以在长列表滚动时还要使用正常滚动更好。

如果之前章节的《朋友圈首页UI-下拉刷新组件》和《朋友圈首页UI-图片查看器组件》你学会的话，关于模拟滚动应该很好理解。



在朋友圈首页，我们用了body滚动，在聊天页面，我们使用局部滚动来实现，需要注意的是，局部滚动不能乱用，每当你用一个局部滚动时，webview都会在终端生成一个原生的 `scrollView`，可能会带来更多的手机webview 内存消耗。

小结

本章节主要讲解了聊天容器组件开发以及采用的滚动方式，其中主要是容器组件的开发，后面会单独讲解单个聊天内容组件和底部输入框组件。

相关知识点：

- 1. 移动web中的body滚动，局部滚动和模拟滚动之间的区别和原理，需要注意的是，局部滚动不能乱用，可能会带来更多的手机webview 内存消耗。
- 2. `-webkit-overflow-scrolling: touch;` 表示在iOS上，使用顺滑滚动，有弹性。

本章节完整源代码地址：

[Github](#)

}