18 下拉刷新组件开发

更新时间：2019-08-28 15:37:02

> 我们活着不能与草木同腐，不能醉生梦死，枉度人生，要有所作为。
>
> ——方志敏

朋友圈首页 UI - 下拉刷新组件

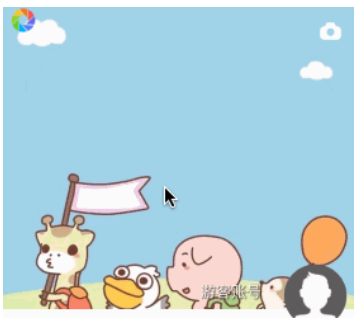
下拉刷新效果是移动 web 端独有的一种用户交互方式，大多出现在滚动列表的页面上，目的更新页面数据。相信大家都体验过微信朋友圈的下拉刷新效果，当页面滚动到顶部时，手指向下拖动，页面左上角会出现一个圆形的 icon 并有向下的位移，本章节将会模拟微信的这个下拉刷新效果，大家可以先预习一下 CSS3 的 transaction 动画效果，下面就进入开发吧。

本章节完整源代码地址，大家可以事先浏览一下：

Github

下拉刷新效果：

我们先来看一下实现的效果，如下图：



下拉刷新逻辑梳理：

1. 当页面滚动到最顶部时，手指在屏幕中向下拖动，触发下拉刷新。
2. 隐藏在页面左上角的圆形 icon 随着手指向下拖动而向下发生位移，并顺时针旋转。
3. 当向下位移到一定临界值后，手指离开屏幕，触发正在刷新操作，圆形 icon 在固定位置自转一定角度。
4. 之后圆形 icon 在向上发生位移，并逆时针旋转直到隐藏起来。

下拉刷新实现：

在前端项目的 components 文件夹下新建 pullRefreshView 文件夹，同时新建组件的 `index.vue`：

首先需要编写下拉刷新组件的 template，这里用到 `<slot>`，大家如果忘记了，可以去前面章节回顾一下，代码如下：

```
<template>

  <div class="pullRefreshView" @touchmove="touchmove" @touchstart="touchstart" @touchend="touchend">
    <div ref="circleIcon" class="circle-icon">
      <div ref="circleIconInner" class="circle-icon-inner"></div>
    </div>
    <slot></slot>

  </div>
</template>
```

上面代码中，最外层使用了一个 div 用来包裹，作为事件绑定的容器，同时新建一个圆形 icon 的 div `.circleIcon`，我们将此 icon 样式设置在*屏幕外*，达到隐藏的效果，代码如下：

```
.circle-icon {

  position: absolute;
  left: 10px;
  top: -30px;
}
```

下拉刷新组件的 UI 基本编写完毕，接下来就要绑定事件了，通过上述分析，加上我们之前章节开发图片查看器的原理，我们需要用到移动端 `touchstart`，`touchmove`，`touchend` 事件，可以实现下拉刷新效果。

首先，监听 `touchstart` 事件：

```
  touchstart (evt) {
    //手指接触屏幕起始时，记录一下位置
    this.pullRefresh.dragStart = evt.targetTouches[0].clientY
    //将圆形icon的动画效果先隐藏
    this.$refs.circleIcon.style.webkitTransition = 'none'
  },
```

在 `touchstart` 事件中，我们主要做的是记录一些初始值，包括手指第一次接触屏幕时的位置，然后将圆形 icon 的动画效果先隐藏。

然后，监听 `touchmove` 事件：

```
touchmove (evt) {
//如果没有touchstart设置的值，说明没进入下拉状态，不影响正常的滚动
if (this.pullRefresh.dragStart === null) {
  return
}

//获取手指的一个target
let target = evt.targetTouches[0]

//根据起始位置和屏幕高度计算一个相对位移量，正值为向上拖动，负值为向下拖动
this.pullRefresh.percentage = (this.pullRefresh.dragStart - target.clientY) / window.screen.height

//获取scrollTop，为了判断是否在页面顶部
let scrollTop = document.documentElement.scrollTop || document.body.scrollTop
//当页面处于顶部时，才能进入下拉刷新的逻辑
if (scrollTop === 0) {
  //当向下拖动时，才能进入下拉刷新的逻辑
  if (this.pullRefresh.percentage < 0 && evt.cancelable) {
    evt.preventDefault()

    //将进入下拉刷新刷新标志位true
    this.pullRefresh.joinRefreshFlag = true

    //根据moveCount速率系数计算位移的量
    let translateY = -this.pullRefresh.percentage * this.pullRefresh.moveCount

    //当位移到一定程度，还没达到临界值时，不断去位移和旋转圆形icon
    if (Math.abs(this.pullRefresh.percentage) <= this.pullRefresh.dragThreshold) {

      //计算圆形icon旋转的角度
      let rotate = translateY / 30 * 360

      //位移和旋转圆形icon，利用translate3d和rotate属性
      this.$refs.circleIcon.style.webkitTransform = 'translate3d(0,' + translateY + 'px,0) rotate(' + rotate + 'deg)'
    }
  } else {
    //向上拖动就没有进入下拉，要清除下拉刷新刷新标志位true
    if (this.pullRefresh.joinRefreshFlag == null) {
      this.pullRefresh.joinRefreshFlag = false
    }
  }
} else {
  //清除下拉刷新刷新标志位true
  if (this.pullRefresh.joinRefreshFlag == null) {
    this.pullRefresh.joinRefreshFlag = false
  }
}
},
```

在 `touchmove` 事件里，我们主要做的是根据手指移动的量来实时将圆形 icon 移动并旋转，大家可以看看代码中的注释，在这里说明一下：

1. 我们的下拉刷新触发的时机是在页面处于屏幕顶部并且手指向下拖动，这两个条件，缺一不可，在代码中，我们利用 `scrollTop == 0` 和 `this.pullRefresh.percentage < 0` 来判断。
2. 在进入下拉刷新状态时，此时手指不断向下拖动，首先圆形 icon `.circleIcon` 会向下滚动并旋转，当滚动到临界值时就只原地旋转。
3. 如果手指在向上拖动，圆形 icon `.circleIcon` 就会向上滚动并旋转。
4. 直到手指离开屏幕前，都不会触发下拉刷新，只是圆形 icon `.circleIcon` 在不停的上下移动。

监听 `touchend` 事件：

```
touchend (evt) {
//如果没有touchstart设置的值，说明没进入下拉状态，不影响正常的滚动
if (this.pullRefresh.percentage === 0) {
  return
}
//在手指离开时，位移量达到临界值时，并且也有进入下拉刷新的标志位，就表明要触发正在刷新
if (Math.abs(this.pullRefresh.percentage) > this.pullRefresh.dragThreshold && this.pullRefresh.joinRefreshFlag) {

  //通知父元素触发正在刷新
  this.$emit('onRefresh')

  //给circleIconInner一个正在旋转的动画，利用css的animation实现
  this.$refs.circleIconInner.classList.add('circle-rotate')

  //700ms之后，动画结束，立刻收起
  setTimeout(() => {

    this.$refs.circleIconInner.classList.remove('circle-rotate')
    this.$refs.circleIcon.style.webkitTransition = '330ms'
    this.$refs.circleIcon.style.webkitTransform = 'translate3d(0,0,0) rotate(0deg)'
  }, 700)
} else {
  //在手指离开时，位移量没有达到临界值，就自动收回，通过transition，设定一个终止值即可。
  if (this.pullRefresh.joinRefreshFlag) {

    this.$refs.circleIcon.style.webkitTransition = '330ms'
    this.$refs.circleIcon.style.webkitTransform = 'translate3d(0,0,0) rotate(0deg)'

  }
}


// 重置joinRefreshFlag
this.pullRefresh.joinRefreshFlag = null

// 重置dragStart
this.pullRefresh.dragStart = null

// 重置percentage
this.pullRefresh.percentage = 0
}
```

在 `touchend` 事件中，我们主要是做一些动画执行的操作，大家可以看看代码中的注释，这里说明一下：

1. 此时手指离开屏幕，位移量达到临界值时，并且也有进入下拉刷新的标志位，就表明要触发正在刷新。此时圆形 icon 原地旋转，并触发下拉刷新回调方法，延迟 700ms 后向上收起。
2. 我们在实现圆形 icon 时的旋转和位移动画时，用了两个 div，在 `touchmove` 时，我们主要对外层的 div 也就是 `ref=circleIcon`，来实现位移和旋转。
3. 在 `touchend` 时，我们主要给内层的 div 也就是 `ref=circleIconInner` 来加 `animation` 动画，因为无法给一个 div 同时使用位移旋转和 `animation` 动画，所以这里一个技巧就是给父元素设置位移和旋转，它的子元素在不设置任何 CSS 动画样式时，是会随着父元素而生效的。

小结

本章节主要模拟朋友圈实现了一个下拉刷新的动画效果。
相关技术点：

1. 下拉刷新操作的主要用户交互流程，包括开始下拉，执行刷新，下拉结束。
2. 利用 CSS3 的 `transform` 的 `rotate` 可以实现一个 div 元素的旋转，结合 `transition` 可以实现旋转动画。
3. 在下拉刷新结束时，我们主要给内层的 div 也就是 `ref=circleIconInner` 来加 `animation` 动画，因为无法给一个 div

同时使用位移旋转和 `animation` 动画，所以这里一个技巧就是给父元素设置位移和旋转，它的子元素在不设置任何 CSS 动画样式时，是会随着父元素而生效。

本章节完整源代码地址：

Github

}

17 图片查看器组件

19 评论输入框组件开发