



链滴

React Native 系列【五】View 滑动时禁止 WebView 滚动

作者: [Vanessa](#)

原文链接: <https://ld246.com/article/1498446273352>

来源网站: [链滴](#)

许可协议: [署名-相同方式共享 4.0 国际 \(CC BY-SA 4.0\)](#)

<p>SymApp</p>

<h2 id="ToC">ToC</h2>

https://hacpai.com/article/1496906863683

https://hacpai.com/article/1497002998658

https://hacpai.com/article/1497406003289

https://hacpai.com/article/1497860440410

https://hacpai.com/article/1497235254333

<h2 id="知识点">知识点</h2>

<p>想快速解决的请直接看最后一节的解决方案</p>

StackNavigator Visual options

Gesture Responder System

WebView

<h2 id="问题描述">问题描述</h2>

<p>在使用 React Navigation 时,虽然各个 View 切换的性能效果都非常不错,也可以用 <code>transitionConfig</code> 很方便的进行自定义切换效果。但如果你从一个原生的 React Native 页面点击后跳转到一个使用 WebView 的页面时,问题就来了。</p>

<p>iOS 下,当我在 WebView 中用从最左滑到右的手势想回到前一个 View 时,我的手如果上下移动了。这个时候 WebView 也会跟着上下移动。</p>

<h2 id="解决过程">解决过程</h2>

查看如何让 WebView 不滚动 -> <code>scrollEnabled</code>

找 navigation 切换时的事件回掉,然而在返回上一页的切换开始时,<code>onTransitionStart</code> 并不会被触发。还好在整个过程中详细的了解了 navigation。


```
<pre><code class="highlight-chroma">onTransitionStart: () =&gt; {
  this.props.navigation.state.isScroll = 1;
  console.log(Article.props.isScroll = false);
},
onTransitionEnd: () =&gt; {
  console.log(arguments, 2);
},
```

```

transitionConfig: () => {
  screenInterpolator: (sceneProps) => {
    const { layout, position, scene } = sceneProps;
    const { index } = scene;

    const height = layout.initHeight;
    const translateY = position.interpolate({
      inputRange: [index - 1, index, index + 1],
      outputRange: [height, 0, 0]
    });

    const opacity = position.interpolate({
      inputRange: [index - 1, index - 0.99, index],
      outputRange: [0, 1, 1]
    });

    return { opacity, transform: [{ translateY }] };
  }
}
)
</code></pre>

```

<p>3 . 本想放弃的，要么不要切换效果，要么换个切换效果，要么就不用 WebView，要么就忽略在 Phone 上回退时手闲的去上下滑动的人（这样的人应该比较少吧）

4 . 最后在放弃的哪天，和同事吃晚饭，聊了下。我的方向错了，这个属于 gesture，并不是 navigation</p>

<h2 id="解决方案">解决方案</h2>

<p>其实蛮简单的，但是一开始的时候思路错了。一直往 navigator 组件上找，看各种 API，找各种件劫持回掉。。。其实只要用 gesture 就可以简单搞定了</p>

```
<pre><code class="highlight-chroma">class Article extends Component {
```

```

static propTypes = {
  navigation: PropTypes.object.isRequired
};

constructor(props) {
  super(props);
  this.state = {
    scrollEnabled: true
  };
}

componentWillMount() {
  this._gestureHandlers = {
    onStartShouldSetResponder: () => true,
    onResponderGrant: () => {
      this.setState({ scrollEnabled: true });
    }
  };
}

```

```
},  
onResponderTerminate: () => {  
  this.setState({ scrollEnabled: false });  
}  
};  
}  
  
render() {  
  const { params } = this.props.navigation.state;  
  return (  
    style={utils.statusBar} {...this._gestureHandlers}&gt;  
    scrollEnabled={this.state.scrollEnabled}  
    source={{ uri: https://hacpai.com/article/\${params.old} }}  
  /&gt;  
);  
}  
}  
  
</code></pre>
```