联系人导航栏插件

本文旨在说明联系人导航栏插件的使用和设计思路，已经存在的问题。发现新问题，欢迎提出。

-----能效通研发部【刘新琼】

# 使用说明

具体使用主要是三个步骤，引入插件，控制器中使用，前台标签的使用。

Ps:引入插件放入工程的lib文件夹下，然后在页面引入相关元素。相关元素内容如下：

引入完成之后，在模块中关联该插件，名称为：**hyd.directives**

<script src="lib/hydContactsNav/directive.js"></script>

<link rel="stylesheet" href="lib/hydContactsNav/contactNavStyle.css"/>

## 1.1控制器中使用

在控制器中使用步骤如下：

1.注入插件提供的工具service，名称为：【contactsNavUtil】

2.在控制器中得到联系人数据之后，调用工具类方法，会对数据按照首字母分类整理并排序，具体使用如下：

|  |
| --- |
| contactsNavUtil.sortByFirstCode(data,attrName)；  参数解释：   1. 联系人数据数组。 2. 根据联系人对象的哪个字段按照首字母排序，需要传入字段名。   返回数据格式大概如下：  [  {  "id": "hydContactsNav2",  "firstCode": "C",  "data": [  {  "id": 2,  "name": "刘新琼"  },  {  "id": 3,  "name": "陈龙"  }  ]  },  {  "id": "hydContactsNav13",  "firstCode": "$",  "data": [  {  "id": 19,  "name": "$%^"  }  ]  }  ] |

得到整理后的数据之后，和控制器的$scope对象进行绑定。接下来就是前台标签的使用。

## 1.2前台标签的使用

为方便解释，贴出例子，得到整理后的数据之后，要怎么显示还是有开发人员进行控制，这里只是一个简单使用demo如下：

|  |
| --- |
| <ion-content delegate-handle="contactsScroll">  <ion-list>  <div ng-repeat="friends in friendsArray">  <a class="contacts\_a" href="javascript:void(0)" id="{{friends.id}}">{{friends.firstCode}}</a>  <ion-item ng-repeat="friend in friends.data">  <h3>{{friend.name}}</h3>  </ion-item>  </div>  </ion-list> </ion-content>  <hyd-contacts-nav hyd-delegate-handle="contactsScroll" hyd-contacts-data="friendsArray"></hyd-contacts-nav> |

1.自定义标签的使用

标签名hyd-contacts-nav，有两个属性，解释如下：

|  |
| --- |
| hyd-delegate-handle：用来关联当前滚动对象的手柄，在这里和<ion-content>的delegate-handle属性对应。  hyd-contacts-data：关联整理分组排序后的对象。 |

2.迭代联系人数据的时候要处理的地方

迭代的时候，一般通过a标签进行分组划分，此时a标签必须具有的id属性，用来关联分组后数据对象的id值，因为在指令中逻辑是通过找寻这个id对象，读取scrollTop值进行滚动的。

# 设计思路

## 2.1 事件选择

|  |  |
| --- | --- |
| on-drag | 本插件开发暂时采用on-drag事件进行拖动逻辑处理，在拖动的过程中具体选择哪个元素，是通过计算的得到的。 |
| on-touch | 配合on-touch事件，实现即使不拖动，只是单纯点击也可以进行相应。 |
| on-release | 配置on-release事件，实现松开时提示框消失。 |

## 2.2 对象分类并排序

我们从服务器得到的数据通常是杂乱的，我们首先需要对数据进行分类整理并排序。

在插件中，我们使用了pinyin.js用来得到字符的首字母。

在contactsNavUtil中，我们首先通过SortByFirstCode对数据进行分类，分类完成后，扩展了js提供的sort方法，使得sort可以针对对象的某一个属性来进行排序。

## 2.3 模板文件

我们需要模板文件来承载我们的导航栏，通过上面的数据处理之后，我们的导航栏存在的导航都是有对应数据的。具体模板文件如下：

|  |
| --- |
| <ul class="alpha\_sidebar" on-drag="goList($event)" on-release="hidePromptBox()" on-touch="goListByTouch($event)">  <li ng-repeat="contacts in contactsData" data-id="{{contacts.id}}">{{contacts.firstCode}}</li>  <div class="promptBox-hydNav"><h3></h3></div> </ul> |

我们使用fixed定位将ul列表定位在可视窗口的右边，构造了一些简单的css进行了修饰，文件名为【contactNavStyle.css】，如果有需要可以自己进行升华。

这里有三点需要说明

1. fixed布局失效问题

在ionic指令下，有时候为为你附加一个会自动附加一个.scroll的父容器的父容器，如果在这个容器下面使用fixed布局，因为transfrom属性会对界面进行单独渲染，会导致fixed布局失效。

具体导致失效源代码如下：

|  |
| --- |
| .overflow-scroll .scroll {  /\* position: static; \*/  height: 100%;  -webkit-transform: translate3d(0, 0, 0);  } |

1. 提示框设计

设计了一个简单的div来对当前的选择的内容进行提示。

1. 数据来源

Li里面的contactsData数据是在自定义指令时，通过scope属性，创建一个隔离作用域来实现的，这样就可以对模板的数据实现灵活可控。

在li上传递一个data-id属性，主要是用来传递数据给指令link方法。

## 2.4 自定义指令

在自定义指令中主要逻辑在link中，在link中实现获取定位导航元素并使视图滚动到对应位置。主要是对三个事件进行方法定义和逻辑处理。

1. goListByTouch :这里逻辑比较简单，我们可以根据事件对象，直接得到事件源，根据事件源的data-id属性得到id值，通过id值找到对应a标签元素，接着得到a标签的scrollTop值，调用ionic提供的内置服务$ionicScrollDelegate实现滚动。

2. hidePromptBox：调用util提供的方法，使提示框消失即可。

3. goList：用来相应用户的拖动，具体逻辑如下：

3.1 根据事件源得到具体对象

3.2 得到对象的高度和父容器的总高度

3.3 得到当前相对于父容器的top偏移值

3.4 计算多少个元素

3.5 根据event.gesture.deltaY+top值计算当前元素

3.6 实现滚动

# 存在问题

## 3.1滚动时元素确定有误差

因为暂时没有找到合适的事件，拖动的实现是基于on-drag事件，具体元素的确定是通过计算得到的，会有大约小于一个单元高度的误差。具体原因如下：

ionic.DomUtil.getPositionInParent(element);得到的值总是为当前对象前面所有兄弟节点的高度值，但是由于元素本身有大小，误差如下：

如图所示我们点击圆圈处，上面箭头的高度无法得到，在原生js中，可以使用事件对象的.offsetY得到，但是在这里不知为何没有这个属性。