Ng-Zorro

注：

1.在行文中没有对属性和指令做明确的区分，从何种角度去理解需要思考。

2.通过我的写法可以判断这是一个指令还是一个组件。（就以下面的按钮为例，明显nz-button的用法是<button nz-button>,而按钮组就直接是一个组件了<nz-button-group>）。

1. 按钮（nz-button, <nz-button-group>）

按钮没有什么好记录的。直接上图就好了。



1. 图标（<i class=”aniicon anticon-${type}”></i>）

图标也没有什么好记录的，type是图标的名字。

1. 栅格（24栅格系统，nz-row/nz-col）

对于CSS3的Flex布局不太熟悉，这里的Flex布局比较好理解，关注一下。

[nz-row]

nzGutter比较强大，这个成员与24栅格系统不造成冲突，就是说它的大小不会算在[nz-col]的[nzSpan]指令里面，设置没有限制，它会去改变每一个[nz-col]的大小，但是对整体没有影响。



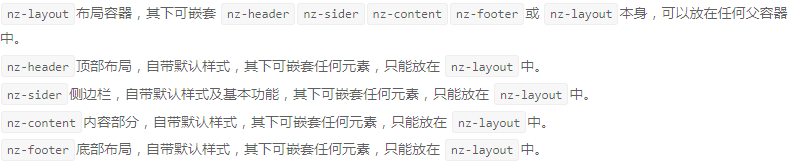
[nz-col]

span pull push offset order 属性可以通过内嵌到 nzXs nzSm nzMd nzLg nzXl 属性中使用,[nzXs]="{ span: 5, offset: 1 }"还是比较牛B的。



1. 布局（<nz-layout>）

在刚看到这里的时候其实没什么好记录的，注意一下官方文档开头的嵌套顺序



这下面都是nz-sider的参数，其实使用起来还好，官方文档最后一个例子有使用方法。



1. 固钉(<ng-affix>)

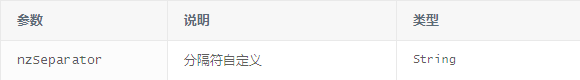
这个组件的属性虽然不多，但是不是很好理解。首先我们从[nzTarget]说起，不设置它的时候默认值是Window,也就是说这个固钉组件能力的发挥是参照Window的。当我滚动Window的时候，在Window里马上会看不见它时，[nzoffsetTop]指令勇敢地站了出来，如果没有设置的话，固钉此时的位置不考虑其他因素，就一个：距离Window顶部的距离为0. (nzChange)作为一个输出指令（触发事件:$event），在固钉的能力开始（结束）发挥作用的时候被触发。

PS:[nzTarget]的用法在官方文档第二个例子里可以看到。



1. 面包屑(<nz-breadcrumb>)

<nz-breadcrumb-item></nz-breadcrumb-item>是面包屑里面的单个项目，让面包屑带上图标的话，可以将图标嵌套在<nz-breadcrumb-item>里面。在<nz-breadcrumb>设置[nzSeparator]指令自定义分隔符。注意：它的类型是string.



1. 下拉菜单(<nz-dropdown>)

下拉菜单组件比较复杂，待我慢慢道来。

注意：需要在触发下拉菜单的元素上加入[nz-dropdown]的directive标记用于定位元素位置。 这是官方文档上面的一句话，说的意思是nz-dropdown不仅是一个组件，而且还是一个指令。要触发下拉菜单当然需要用组件<nz-dropdown>完全包裹起来，而如果你想要将一个<a>变成下拉的，那么加上nz-dropdown指令。

到这儿，也有没讲清楚的属性，因为后面还有很多组件没有学习。nzVisible指令在官网和别的指令一起展示，不太清晰，它的作用其实是当你第一次进入页面的时候，下拉菜单就已经铺开了，它支持双向数据绑定。

在nz-dropdown组件层面没有官网没有涉及到的就是ng-dropdown-custom属性了，但它是却是ng-content类型，不容小觑。(直到第三十一个组件表格，我才见它第一次被使用)

nz-dropdown



nz-dropdown-button

这也是一个牛B的组件，它和nz-dropdown不一样的地方在于它并没有nz-dropdown-button指令，成为一个普通的即拿即用的组件。它的样子是本来就定好了的，左边按钮，右边下拉菜单，它的指令和下拉菜单差不多，在官网有很清晰的用法。

菜单可由 nz-menu取得，菜单还包括菜单项 nz-menu-item分割线 nz-menu-divider和 子菜单nz-submenu。在官网倒数第二个例子。另外官网提到的一个指令nzDisable可以让按钮式菜单GG。



1. 菜单(nz-menu)

这一节的指令信息量有点大，让我慢慢道来。先上图，后说话。

nz-menu

一般用在<ul>上的指令，nzInlineCollapsed比较有趣，但用法也不是很难，参看官方文档的第三个例子就好。nzClickActive官网上面没有使用的例子，我试了一下，它的效果是你按了一下子菜单，那个子菜单会一直亮着，表示你曾经选择了它，而false的话，点击一下亮一下。



nz-menu-submenu

它的作用简单来说就是让你点击它了以后还能继续拓展，前两个指令控制开合与回调在官网倒数第四个例子里面用到。title和下面nz-menu-group的title一样，用在<span title>上作为标题使用，去掉后格局会不妥，也是很重要的指令。



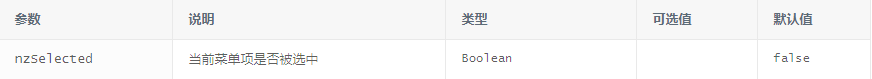
nz-menu-group

将两个相关的项目写在一起。



nz-menu-item

估计nzSelected又是一个双向绑定。



nz-menu-item-divider

菜单子项目分隔线。

1. 分页(<nz-pagination>)

直接上图不废话,其实感觉虽然指令很多，但是并不太复杂。实际显示总共有多少页是根据nzTotal(输入属性)/nzPageSize自动算出来的。 加上nzShowSizeChanger属性的话，分页组件的右边出现一个下拉菜单，用于显示nzPageSize的可选项(貌似以10为倍数)，很明显选择以后实际的总页数就会改变。 如果不加nzShowQuickJumper，你是无法跨越跳的，即使你在43页点击41页，你仍然是跳到42页。该属性还会在分页组件右边加一个小部件。



1. 步骤条(<nz-steps>)

nz-steps

nzCurrent指定了当前的步骤，注意是从0开始计数的，也就是说nzStatus上的状态是上给了第number+1个步骤身上。nzProgressDot属性在官网最后一个例子使用。



nz-step

#nzIcon在官网第三个例子使用到。



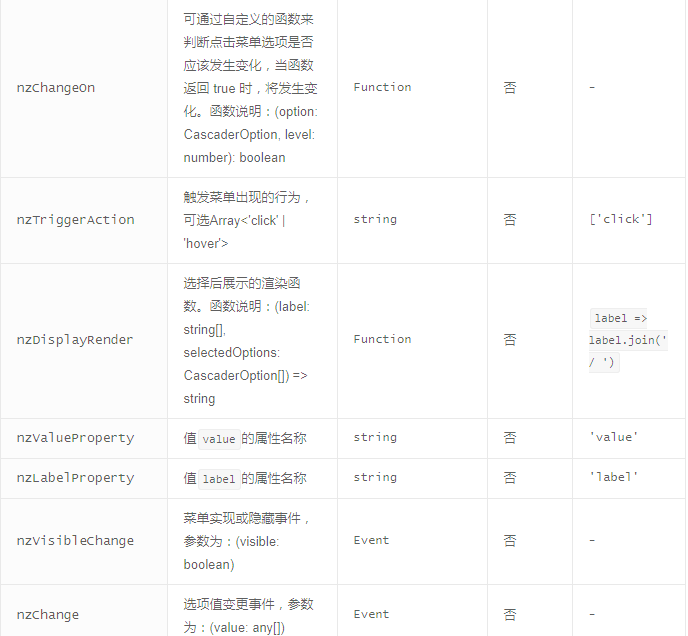
1. 级联选择(<nz-cascader>)

nz-cascader

这个组件的内容有点夸张了，待我慢慢道来。ngModel是默认值，和input元素的placehoder差不多，就是一开始就显示在那儿了，不过它不仅仅是提示。刚说到这就发现它有nzPlaceHolder指令，代表默认文字。nzShowInput在官网第三个例子使用。这个指令之后有很多官网为拿出来介绍的指令，可以自己尝试一下。nzOptions是真正的数据。nzChangOnSelect使你不必等待把每一级的选项都选完后才显示到级联选框中，而是你选一级，就显示到级联框这一部分，nzChangeOn是它的升级版，自定义怎么显示，官网没有说明。nzExpandTrigger和nzTriggerAction的区别在于触发次级菜单和整个菜单的行为。nzDisplayRender在官网倒数的例子里使用。nzChange是几乎在每一个例子里都可以见到的身影，它输出为一个数组，nzSelectionChange和前面说到的nzShowInput对应上了，把输入框隐藏后必须用这个定义显示，同样是第三个例子。nzSelect在官网那样使用,nzLoad在倒数第二个例子使用。









1. 多选框(nz-checkbox，<nz-checkbox-group>)

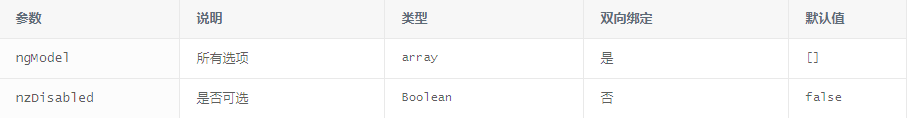
[nz-checkbox]

nzIndeterminate如下面所说，只控制样式，在官网最后一个实现全选的例子里有使用，当它为true时，checkbox里面是一道横杠。



nz-checkbox-group

使用不是很复杂，参见官网倒数第二个例子。



1. 日期选择框(<nz-datepicker>)

nz-datepicker

nzModel就是一开始就出现在日期选择框的日期，并不是placeholder占位那种。nzDisableDate在官网右边最后一个和左边最后一个都有展示。nzShowTime在倒数第二个例子里展示，将它设置成了true,然后就可以选择时间了。



1. 表单

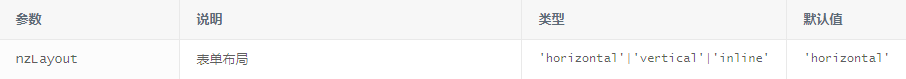
表单这一块指令不多，但怎么用需要搞清楚。整个架构还是<form>加上<nz-input>,注意传统的<input>已经被替换了。继续往下看，当以一个<form nz-form>开头了以后接下来会是什么呢？没错，必然是form item了,这是一个指令包裹在<div>之中。回忆Angular表单项的层级，formGroup接下来就是formControl了，所以接下来嵌套的该是<div nz-form-control>,终于到了<nz-input>了，和输入同级的便是nz-form-explain所在的<div>了。 这只是一个大概的分析，官网第一个例子就可以很好的看见。

在nz-form-control那一层，nzvalidateStatus的用法其实已经很简单了，将它的值设置成当前formControl就行啦，至于这个怎么获取，还是第一个例子。nzHasFeedback在第三个例子使用，它是个属性，直接添上去就好了。 nz-form-label的使用在第四个例子。

后面有几个动态加载的例子，用到了\*ngFor,注意这个结构指令是添加在那一层级的,如今可以把nz-form-item看成最小单位了。

[nz-form-item-required]在倒数第二个例子使用，就是给标签前面加上一个\*，表示必须添加。nz-form-label和nz-form-control同级在ng-form-item下。

[nzform]



[nz-form-control]



[nz-form-itme]

用于标定一个form item区域

[nz-form-label]

用于标定label区域

[nz-form-item-required]

给label添加require样式

[nz-form-explain]

用于显示提示信息，会根据当前nzValidateStatus显示不同的颜色

[nz-form-extra]

用于显示表单额外提示信息

[nz-form-split]

用于显示 - 分隔符

1. 数字输入框(<nz-input-number>)

非常简单的一个组件，它的指令全部可以在一瞬间看懂。



1. 输入框(<nz-input>)

nz-input



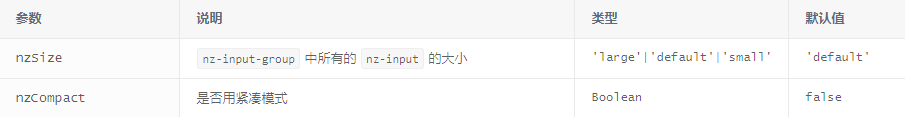
nz-input[type=textarea]

nzAutosize的意义在于只要你不手动拉它，改变它的大小，当输入到六行以后就会出现滑块。而没有此指令的话，它会继续增加输入框的大小。



<nz-input-group>

在nz-input-group中需要使用[nz-input]的directive，而非component进行组合，具体见demo,官网给出的注意事项，此时nz-input作为指令加到<input>组件上面。



1. 评分(<nz-rate>)

非常容易，不再赘述。



1. 单选框（nz-radio）

nz-radio-group



[nz-radio]/[nz-radio-button]

同样非常好用，一般用在<label>当中。



1. 选择器(<nz-select>)

nz-select

选择器也比较繁琐，待我一一道来。nzSearchChange的使用，必须把nzShowSearch启用搜索框的指令带上，它们在官网左边倒数第二个例子用到。nzMode多选的使用在官网右边例子倒数第二个，tags模式和multiple模式在官网右边第一个例子上演，在我看来没有任何差别，反倒是tags模式可以自己新增tag，而multiple模式不能。nzOpenChange看意思应该能明白怎么用。nzFilter我亲自试了一下都没大明白，但是把他设置成false的话还是会有一点差别，在官网的淘宝搜索例子中，输入：草泥马，明显不可能有完全匹配的东西，此时正常情况nzFilter为true的情况会说查找不到此物，但是为false的话会显示马草泥系列产品。nzKeepUnListOptions相当于一个记录功能吧。nzAllowClear把你的输入以下全部清掉，第一个例子就使用了。nzNotFoundContent在你没有找到的时候出现的话语，在前面淘宝搜索的例子里面有用到，而且我发现nzFilter为false的话，不可能触发nzNotFoundContent，因为此时根本就不会过滤。



nz-option

注意nzValue的使用，第一个例子就可以拿来研究，与nz-select选择option后的ngModel属性对应。



二十.滑动输入条(<nz-slider>)

这个组件感觉还不错，待我慢慢道来。nzStep提到要被max-min整除，nzMarks应该是最重的重头戏了，在官网的最后一个例子使用，它的值该怎么设置在官网的例子上书写的非常清楚，它的外表可以理解为在输入条上打上断点标记，而且这个标记的样式自定义，很牛B的一个指令。然后再回到nzStep上，nzMarks设置以后，它可以为设置为null（注意它默认是有设置的，为一），Slider的可选值就只有那些断点了，这个现在再来理解变得非常清晰。到了nzIncluded，对于这个指令我不太清楚，除了看到官网例子输入条颜色的变化，不清楚有什么作用。nzOnAfterChange这个指令非常厉害，官网例子当onmouseup触发，我试了一下，就是你的鼠标按键抬起来的时候触发，当前值会作为value返回，很不错的一个指令。nzTipFormatter在官网右边倒数第二个例子使用，它的外在表现是你滑动滑块时此刻的位置提示，此时的值他会反给你，输出怎样一个字符串来表示随你自己，比较不错的一个指令。





二十一.开关(<nz-switch>)

这个组件非常简单，最后两个指令的用法在官网左边最后一个例子里使用。



二十二.时间选择框(nz-timepicker)

需得提一提的便是最后四个属性了，官网最后两个例子分别演示了nzDisabled(Hours/Minutes/Seconds)和nzHideDisabledOptions的组合，nzHideDisabledOptions属性相对来说简单一些，只是将前面设置的不能选择的数字隐藏起来。nzDisabled(...)系列全部都是函数，根据例子来看，返回值全部是数组，在这段取件内的数字不能选择，有趣的是分钟的控制能根据小时来掐，秒钟的控制能根据小时和分钟同时来掐，例子很清晰。



二十三.头像(<nz-avatar>)

没什么好说的，一目了然，官网最后一个例子，带徽标的头像，和下一节徽标一起学习。  


二十四.徽标数(<nz-badge>)

这个组件直接看官网比较好一点。



二十五.日历(<nz-calendar>)

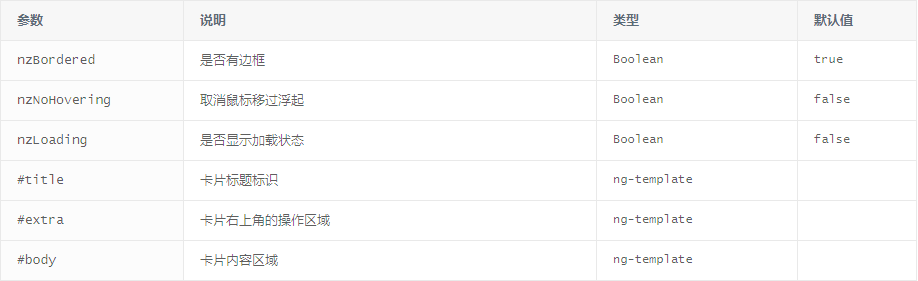
官网最后一个例子展示了天月模板的使用。





二十六.卡片(<nz-card>)

这个组件同样不用怎么解释，看官网的例子应该能够看懂。对于nzNoHovering的效果和它的字面意思一样，鼠标扫过卡片的时候，整个卡片不再浮起。



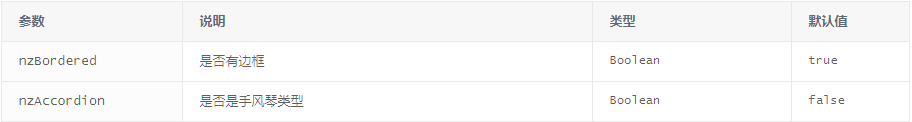
二十七.走马灯(<nz-carousel>)



二十八.折叠面板(<nz-collapseset>/<nz-collapse>)

nz-collapseset

手风琴类型即一下只能打开一个面板(nz-collapse)。



nz-collapse

在官网倒数第三个例子嵌套面板让我们知道了在<nz-collapse>里面是可以放很多内容的，包括collapseset以形成嵌套的面板，在官网最后一个例子给它加上[ngStyle]可以自定义面板样式。



二十九.气泡卡片(<nz-popover>)

首先这不仅是一个组件，在官网的第一个例子button便要搭载nz-popover Directive，其次最重要的莫过于nzTemplate定义Content的内容了。至于最后的两个指令官网只字未提。

另外[nzTitle]对气泡卡片标题的设置下面的参数没有记录下来。



三十.文字提示(<nz-tooltip>)

如果我说文字提示组件和上面的气泡卡片组件的用法一模一样，我想应该没有人会拒绝。同样以<nz-tooltip>为首，再去包含含有nz-tooltip Directive的元素，就连官网的最后两个指令说明都还是写着卡片...，明摆着粘贴复制的好吗。但是讲道理，官网开头给出的何时使用提到气泡浮层不承载复杂文本和操作，这就使得它和气泡卡片有着些许不同，下面的参数里也有说明，就是nzTemplate内容会覆盖掉nzTitle。而气泡卡片是都显示。

请确保 Tooltip的子元素能接受 onMouse EnteronMouseLeave onFocus onClick事件



三十一.表格(<nz-table>)

nz-table

这个组件比较复杂，以第一个例子从结构方面分析一下。首先它以<nz-table>来嵌套一切，接下来便是表格头<thead>和表格身体<tbody>，但是它们分别带有nz-thead和nz-tbody指令。在表格头里面和常规的写表格差不多，<tr>里面嵌套<th>（表格头列），不同的是在下面有一个指令nz-th可以看到用来标定th，因此将nz-th作为<th>的指令使用。然后再来看看表格身体，同样的以<tr>和<td>（表格身体列）作为主要结构，同样的有nz-td指令来标定<td>。 官网第五个例子是对筛选和排序的控制，用到了<nz-table-sort>和nz-table-filter指令，<nz-table-sort>在官网展示的样子是上下箭头堆叠在一起，点击重新排序。 令人感到惊喜的是，在官网第六个例子中给我们看到了前面讲下拉框时没有提到的nz-dropdown-custom的用法。官网第七个例子谈起了分页，感觉上<nz-table>是内置pagination的，因为它本身就有nzPageSize指令，而且没有使用pagination组件。

倒数第三个例子实现带编辑功能的表格，没看过不知道怎么实现的，看了才知道原来是一个幌子，非常巧妙。你看倒数第二个例子设置nzIspagination为false就不帮你分页了。

nzAjaxData只在官网远程数据例子看到过，没有什么特别。#模板引用变量在第一个例子就有使用。官网固定表头例子使用了nzScroll和#nzFixedHeader的组合，非常清晰。在这后面的几个指令依我看完全暴露了表格内部实现了Pagination。随后nzCustomNoResult好像没有看见例子，但是看到最后一个例子nz-table-footer Directive和nzShowFooter，nz-table-title Directive和nzShowTitle的表现，似乎不是很难理解。最后几个boolean指令无伤大雅，告辞！





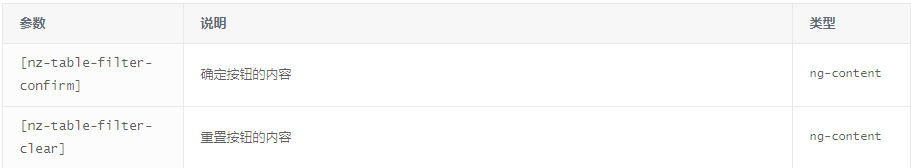


nz-table-sort

嵌入th中显示排序状态

nz-table-filter

与nz-dropdown结合使用，用于过滤表中的数据



[nz-thead]

标定thead

[nz-th]

标定th



[nz-table-body]

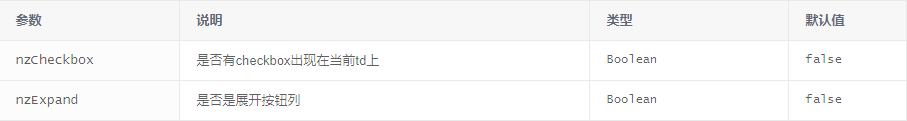
标定tbody

[nz-tbody-tr]

标定tbody中tr

[nz-td]

标定td



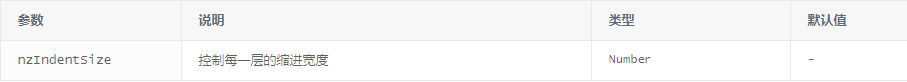
[nz-table-divider]

用于td中分隔数据

nz-row-indent

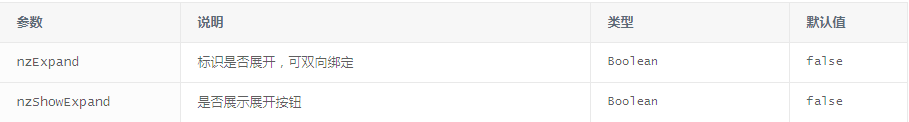
在官网树形结构例子里看得分明。

与树形数据展示配合使用



nz-row-expand-icon

与可展开和树形数据展示配合使用



三十二.标签页(<nz-tabset>)

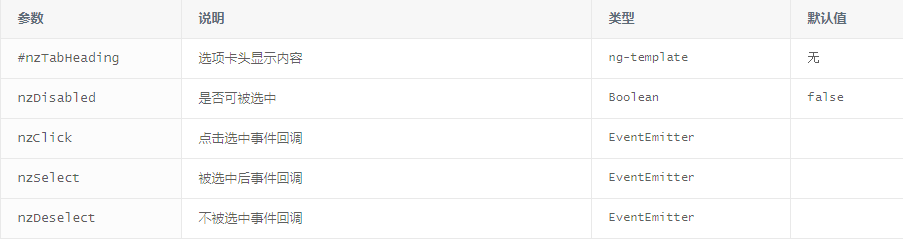
nz-tabset

这个组件比起上面的表格良心了不少。nzSelectedIndex,nzSelectedIndexChange,nzSelectChange在第四个例子开始它们的表演，#nzTabBarExtraContent又是个十分重要的模板参数，在官网第五个例子展现了它的威力，作为标签页题目上额外的元素。



nz-tab

#nzTabHeading又来啦，它提供每个标签的题目。在官网第四个例子可以看到所有的回掉，点击确认的那个小标签会触发nzClick和nzSelect,其它所有的标签触发nzDeselect,你会发现它们的回掉顺序是这样的:nzClick>nzDeselect>nzSelect。



三十三.标签（<nz-tag>）

nz-tag

这四个指令在官网第一个例子就已经全部使用到了，描写的十分的好。



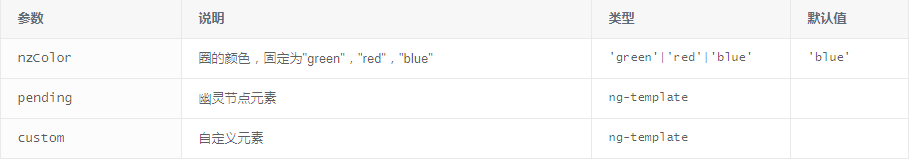
nz-checkable-tag

这个组件的名字多了一个checkable,可以想象它带有了多选框的色彩。在官网最后一个例子有描述。nzChange点击标签时触发的事件，最后一个例子里面把它的参数直接赋值给nzChecked,可见在未选中时点击一下它参数为true，选中时点击一下它参数为false。



三十四.时间轴（<nz-timeline>）

这个组件比较简单，官网的例子应该一看就懂，#pengding和#custom模板参数还是很厉害的。



三十五.警告提示(<nz-alert>)

指令看上去很多的样子，其实十分容易。首先正如官网最后一个例子所说的，nzCloseText自定义关闭按钮会覆盖nzCloseable默认的关闭Icon。nzMessage比nzDescription大一号，

nzShowIcon在官网左边倒数第一个例子用到，它会根据你这个提示类型nzType的不同在提示框开头显示不同的图标。最后三个指令没见使用。



三十六.全局提示(NzMessageService)

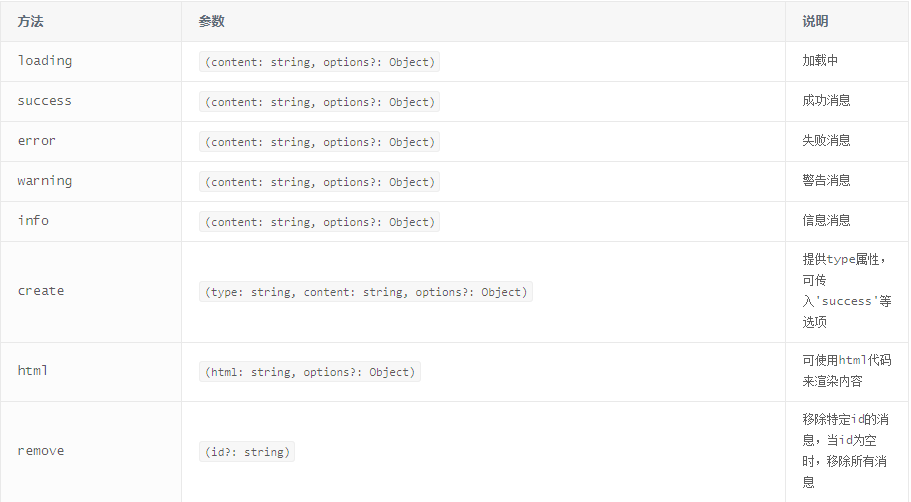
这个厉害了，这是一个服务很牛B。但是官网给出的例子已经很好了，清晰易懂。

全局配置(NZ\_MESSAGE\_CONFIG)



NzMessageService服务

提示：以下options参数支持全局配置中的 nzDuration/nzAnimate/nzPauseOnHover



三十七.对话框(<nz-modal>)

来了，又一个重量级组件，让我们慢慢道来。

nzConfirmLoading的表现在官网右边第一个例子有展示，效果很清晰，只要一按确定按钮（默认的按钮）确定按钮就进入Loading状态。nzTitle,nzBody,nzFooter是一样的用法，它们都支持传入TemplateRef,nzFooter不支持string,所以官网的例子这三个设置都是用Template写的，具体怎么操作看官网哈。nzOnOk,nzOnCancel的回掉描述的非常清晰，没有什么特别描述的。一直到nzStyle都写的比较明白，nzStyle在官网左边倒数第一个例子使用。







NzModalService.xxx()

包括：

NzModalService.open

NzModalService.info

NzModalService.success

NzModalService.error

NzModalService.warning

NzModalService.confirm

以上均为一个函数，参数为 object，具体属性如下：

这里面有几个很厉害的参数，可配置性极强，分别是title,content,footer以及componentParams,通过直接在<nz-modal>添加指令的方式，我们看到了nzTitle,nzBody,nzFooter传入Template的神级操作，然而官网左边第二个例子，以服务来构造对话框的形式，content参数更加变态，看下面的说明写到，它居然可以传Component，其实真正用法没什么区别，但是这个内容的可配置性就高了一个档次了，另外componentParams传参没啥子说的了。





NzModalSubject对象

这个的使用也牛B，还是官网左边第二个例子。

在弹出component中实例化NzModalSubject对象后可以通过next方法向外传输数据

所有的NzModalService.xxx()方法都会返回一个NzModalSubject对象。

这个对象是对rxjs的Subject对象的封装。除了支持Subject对象原生的方法外，还支持：

NzModalSubject.destroy ( type ) - 用于销毁当前的modal对象

NzModalSubject.on ( eventType, cb ) - 用于监听modal对象各个阶段的事件

destroy方法中的type支持传入 onOk 和 onCancel ，表示销毁modal的时候会执行用户传入的options中的onCancel还是onOk方法

NzModalSubject支持的所有eventType包括：

onShow

onShown

onHide

onHidden

onOk

onCancel

onDestroy

其他（ 用户可以自定义任何事件，都可以通过该对象监听 ）

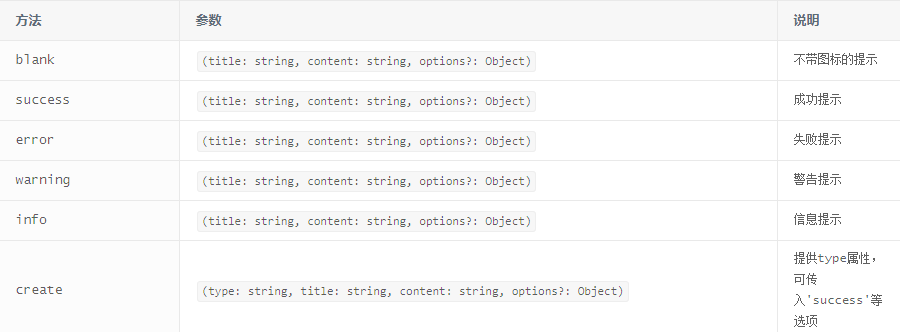
三十八.通知提醒框(没有组件)

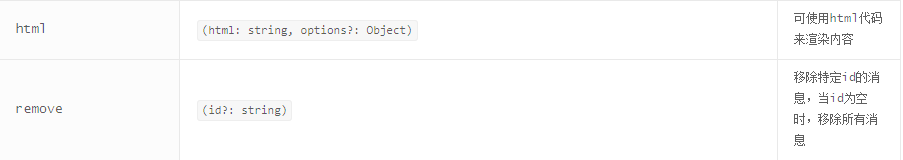
全局配置(NZ\_NOTIFICATION\_CONFIG)



NzNotificationService服务

提示：以下options参数支持全局配置中的 nzDuration/nzAnimate/nzPauseOnHover





三十九.进度条(<nz-progress>)

这不是一个很难的组件。只有官网最后一个例子用nzFormat更改内容模板还算比较新颖。



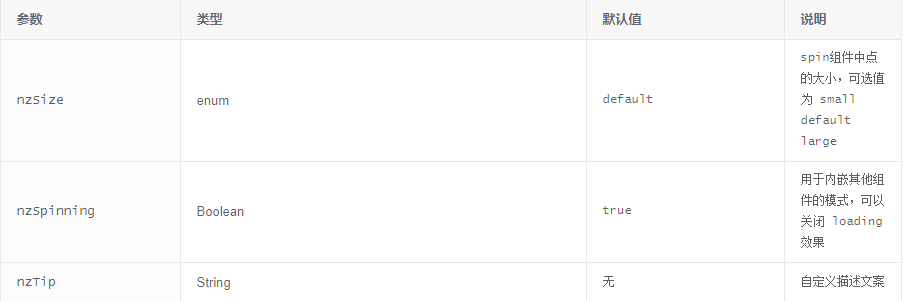
四十.气泡确认框(<nz-popconfirm>)

同样不复杂的一个组件，nzCondition为true的话，直接进行nzOnConfirm的回掉，不会弹出气泡确认框再让你选择。



四十一.加载中(<nz-spin>)

可以直接把内容内嵌到 nz-spin 中，将现有容器变为加载状态。整个官网例子唯独这句话最为有用。



四十二.锚点(<nz-anchor>)

nz-anchor



nz-link



四十三.回到顶部(<nz-back-top>)

官网的例子完全覆盖。

