



ZSYCNav 3.0 界面导航模块

说明文档

V3.0



目 录

目 录	- 1 -
1 升级相关信息	- 3 -
2 概述	- 4 -
2.1 使用标准 HTML5 技术打造	- 4 -
2.2 定制登录页面	- 4 -
2.3 基本界面	- 5 -
2.4 界面自定义	- 6 -
2.5 警报	- 8 -
3 使用前准备工作	- 9 -
3.1 安装模块	- 9 -
3.2 安装授权文件	- 9 -
3.3 设置 NAV 文件	- 10 -
3.4 设置用户组	- 10 -
3.5 设置 Role	- 11 -
3.6 设置用户	- 11 -
4 框架模块的简单使用	- 13 -
4.1 打开 ZSYCNav-ux 模块的 palate	- 13 -
4.2 将 Page_Horizontal 拖入文件夹中	- 13 -
4.3 Page_Horizontal 模块的 Nav_Link_Ord 可设置网站登陆后的默认视图	- 14 -
5 导航目录结构	- 15 -

5.1	例子.....	- 16 -
5.2	上例效果.....	- 17 -
5.3	基本界面布局.....	- 19 -
6	参数详解.....	- 21 -
6.1	Page_Horizontal 模块下属性.....	- 21 -
6.2	按钮属性.....	- 25 -
7	附录 1：几种属性的涵义、格式.....	- 29 -
7.1	颜色：.....	- 29 -
7.2	外边距 margin_String.....	- 29 -
7.3	边框宽度 border_Width：.....	- 30 -
7.4	圆角 border_Radius_String.....	- 31 -
7.5	字体 Font_Family:.....	- 33 -
8	附录 2：特殊视图的链接.....	- 34 -
9	附录 3：总报警数的取得.....	- 36 -
10	附录 4：自定义图标.....	- 38 -
10.1	下载图标文件.....	- 38 -
10.2	引入图标.....	- 41 -
11	附录 5：自定义登陆页面.....	- 43 -
11.1	设置登录模板.....	- 43 -
11.2	设置登陆首页相关文件夹.....	- 43 -
11.3	关于网页定制的相关说明.....	- 48 -

1 升级相关信息

- 本模块适用于 Niagara N4.X 系列软件，在 Niagara AX 3.X 系列软件上**不能**使用。
- 由于大量增加了属性，ZSYCNav 3.0 版本与之前的 ZSYCNav 2 系列模块不能兼容，如有需要升级已含有 ZSYCNav2 系列模块的站点，请记得先删除站点内的 ZSYCNav2 对象，然后再更新 ZSYCNav.jar 模块文件，然后在站点中重新配置导航界面。
- 如有测试需求，请联系我获得模块及测试授权。
- 本人联系方式： Email : nightycd@163.com QQ:183548471

2 概述

2.1 使用标准 HTML5 技术打造

本系统使用标准 HTML5 技术打造，使用通用的浏览器即可浏览，无需专用客户端，电脑、手机、平板均可登录使用本系统。

本模块主要适用于有大可视面积的 PC 端或平板端的浏览器浏览，手机可浏览用本模块搭建的导航界面，但与电脑浏览无异，本模块未专门针对手机优化界面。

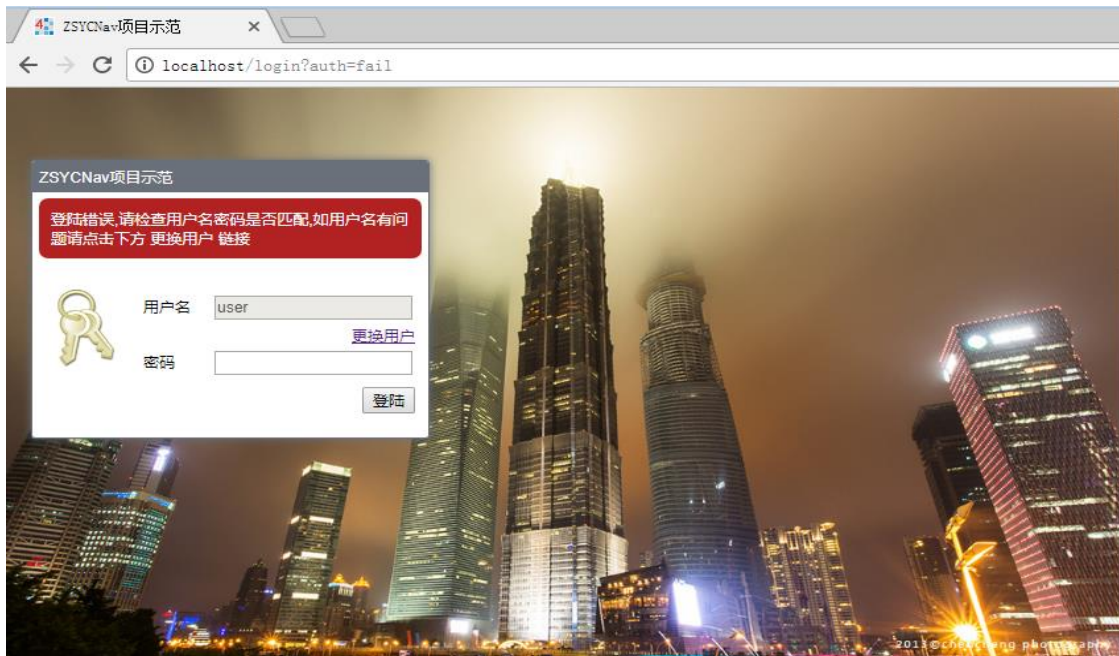
用户启动系统自带浏览器(如 IE11,Firefox 等,推荐 Firefox),并输入系统网址，即进入初次登录页面

2.2 定制登录页面

可定制登录页面。

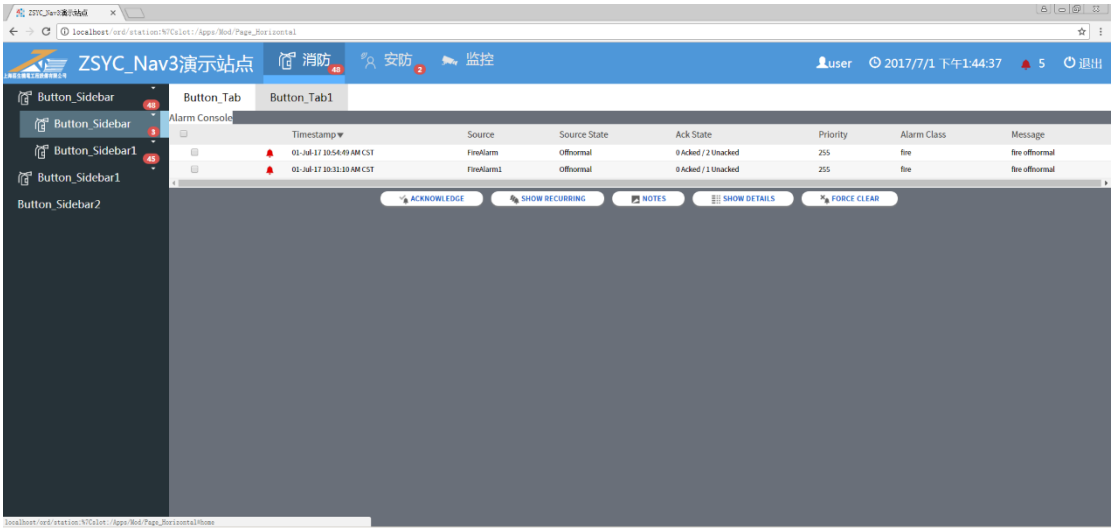
可将已定制好的网页放入 Station 内，然后将其指定为登陆页面。

具体如何设置请参见本文第 11 章节所述，下图为简单举例：

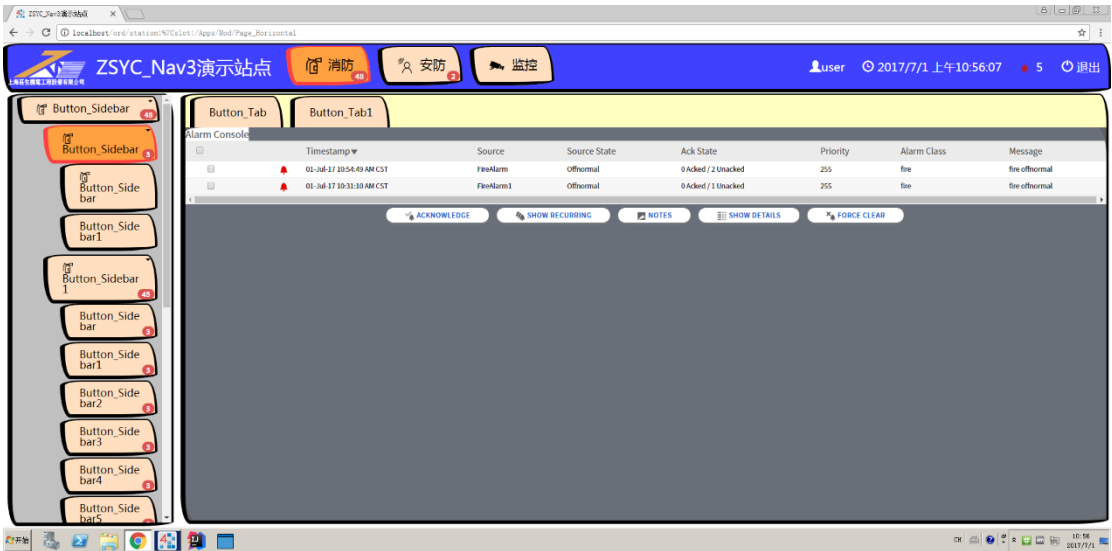


2.3基本界面

默认界面为 Metro 风格界面，如下图



本控件灵活多变，可具体设置各个区域的背景色、外边距、边框、圆角、字体、字号、字颜色、图标等等各种属性，所以可以方便快捷的更改界面风格，如下图即为我测试时所更改出的一个诡异风格。

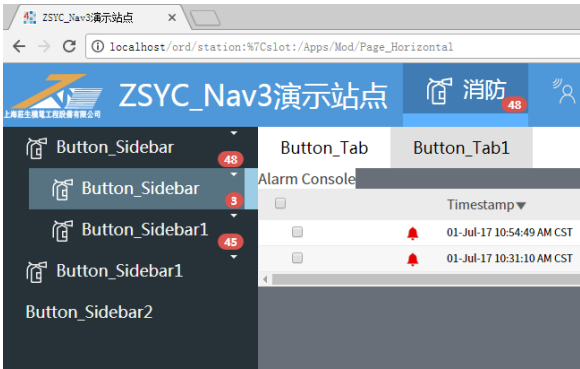


2.4界面自定义

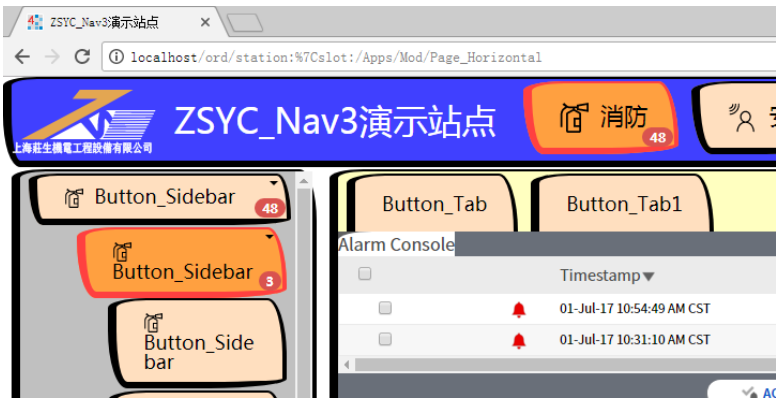
- 可自定义区域及元素间的间隔

例如：

默认的是无间隔，如下图



通过配置则可设置为如下效果

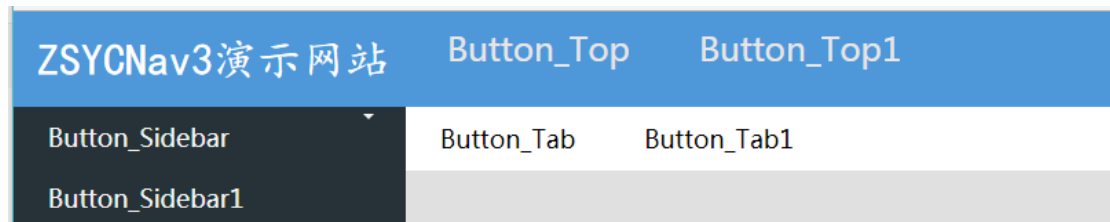


上图中的侧边栏按钮的设置如下所示

Top Button Property	Button Property
Margin _ String	2px 4px 6px 8px {ok}
Back Ground _ Color	#ffdfbf
Front _ Color	black
Border _ Color	black
Border _ Width	2px 4px 6px 8px {ok}
Border _ Radius _ String	2em 1em 4em / 0.5em 3em {ok}
Button _ Active _ Back Ground _ Color	#ffa040
Button _ Active _ Front _ Color	black
Button _ Active _ Border _ Color	#ff4040
Icon _ Height	28.00 {ok}
Button _ Font _ Size	22.00 {ok}
Button _ Font _ Family	Microsoft Yahei, Heiti SC, SourceSansPro, ...

● 自定义字体、字号、字体颜色

例如下图中网站标题试用了楷体字体



● 自定义按钮图标

可自引入外部字体按钮 JS 文件，自定义各个按钮的图标



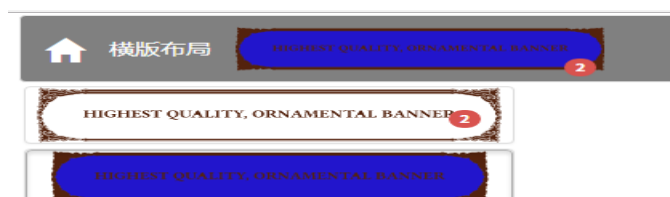
● 自定义界面元素颜色

可自定义各个部分的背景颜色以及字体颜色，可自定义按钮按下后的背景色及字体颜色



● 自定义界面元素背景图片

可自定义按钮的背景图片，可自定义按钮按下后的背景图片



2.5 警报

- 可直接在按钮中以小印章形式显示警报条数



- 可集中显示：

可在单一页面显示所有系统已发生警报的点的警报记录

- 可分系统显示警报记录

可在各个系统内部有各自分系统的警报记录。

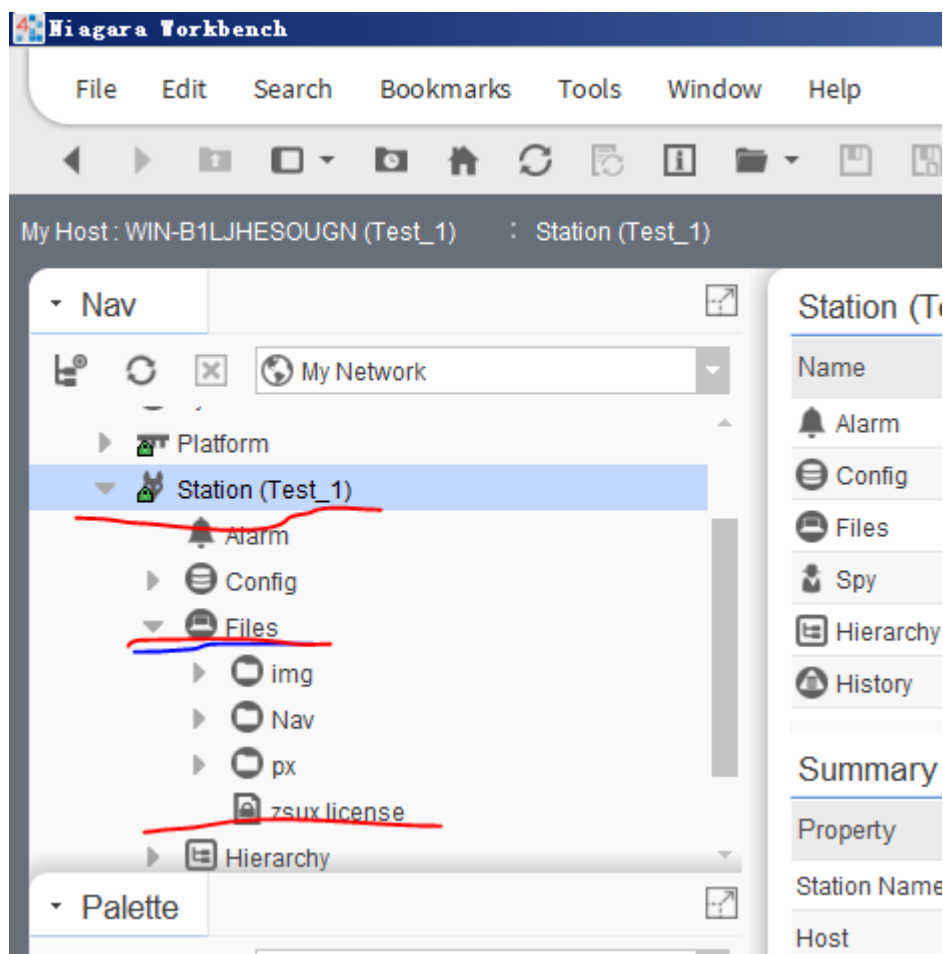
3 使用前准备工作

3.1 安装模块

将 ZSYCNV-ux.jar 和 ZSYCNV-rt.jar 拷贝至 niagara4 安装目录下的 modules 目录下

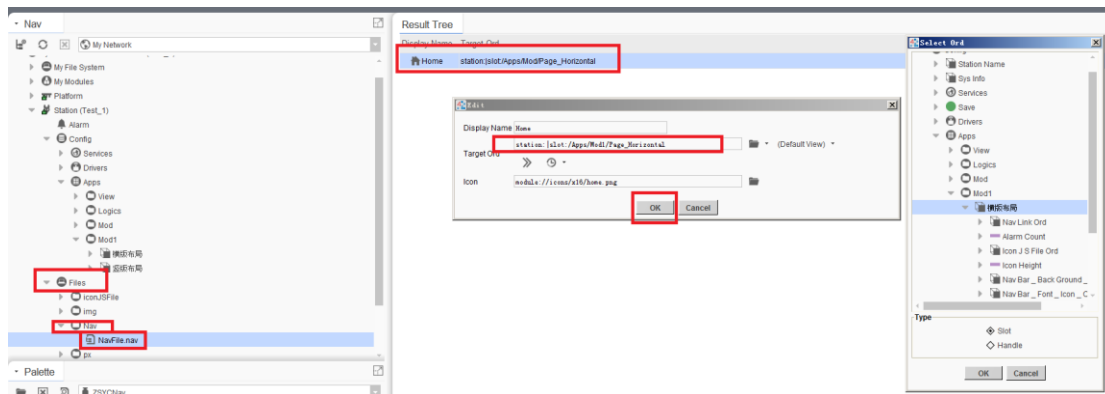
3.2 安装授权文件

在 Niagara4 的 WorkBench 界面中，将相应 Host ID 的 zsux.license 文件，拷贝至需要运行该模块的站点的 files 目录下。**注意必须在 Station 下的 files 的根目录下**

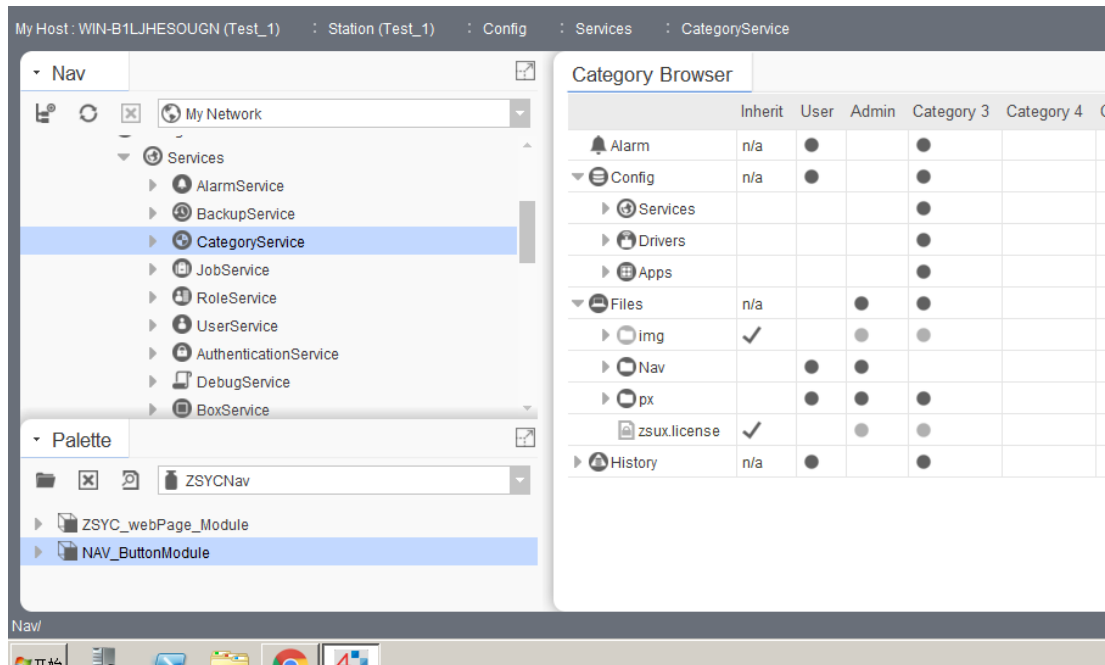


3.3 设置 NAV 文件

要想实现一登陆 Niagara4 就显示该网页框架，需设置用户首页指向，首先得设置 NAV 文件：在 Station 下新建 Nav 文件夹，在该文件夹下新建 NavFile.nav 文件，并修改该文件的 Home 属性

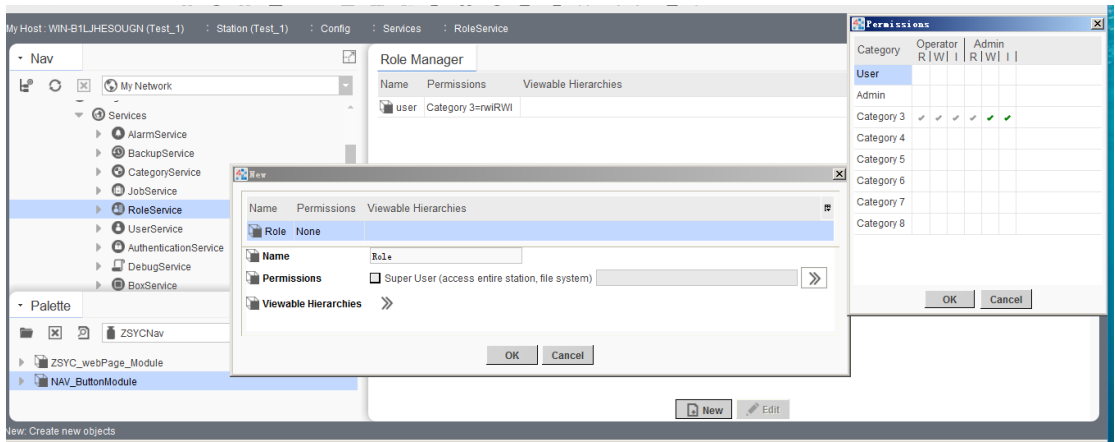


3.4 设置用户组



注意：此处请根据需要设置安全组，此图只是一个例子，无需跟本图设置的一模一样，另外在使用当中遇到链接出 404 错误页面，请先检查或修改此处安全组设置，看看是否是该用户所在安全组无访问链接目标点的权限。

3.5 设置 Role

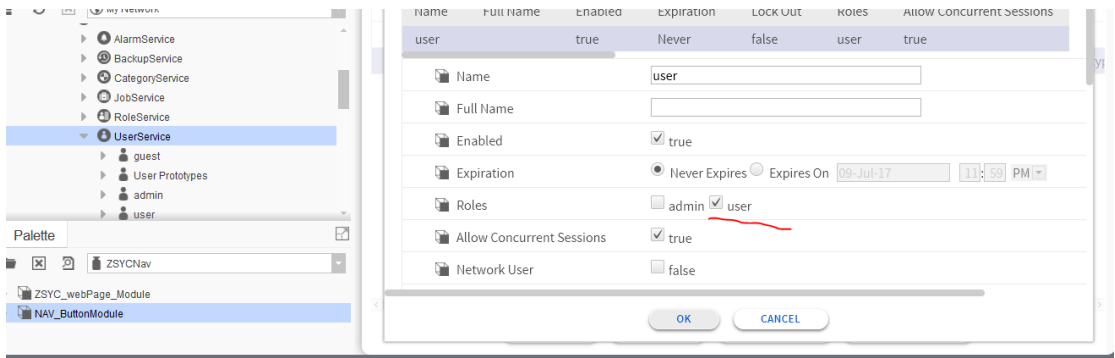


注意：本模块最低只设定有 Operator 的 r 和 Admin 的 R 权限即可使用。

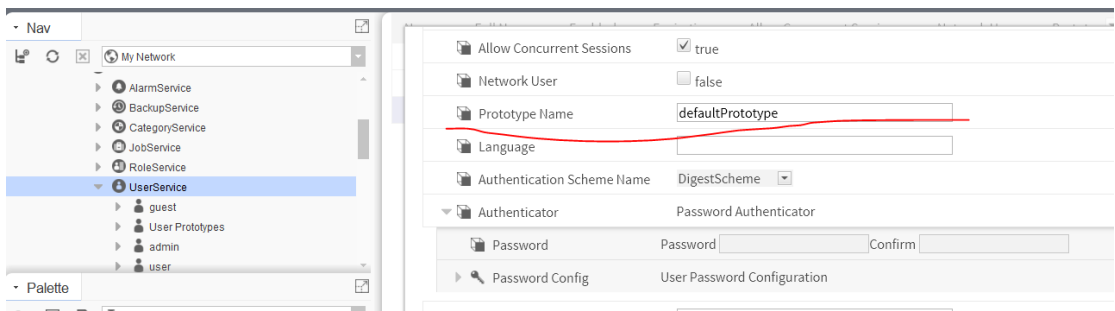
3.6 设置用户

3.6.1 新建用户，注意最好不要给用户用默认的 admin 用户，因为该用户为默认的管理员用户，拥有所有操纵站点的权限

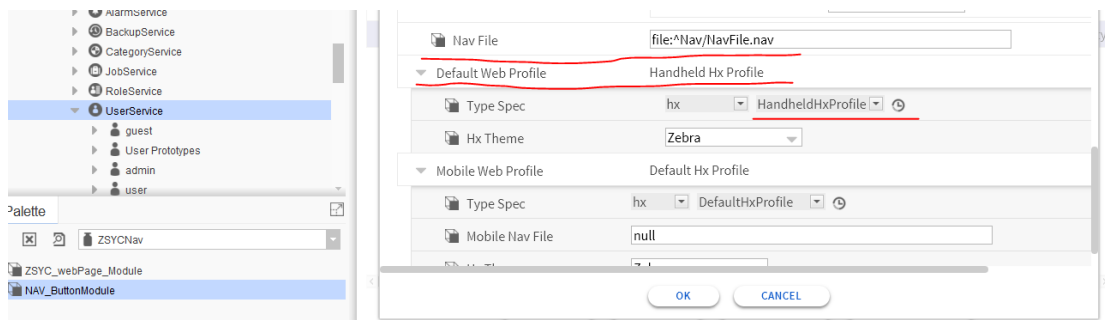
3.6.2 用户 role 的选择



3.6.3 用户 protoType 的设置



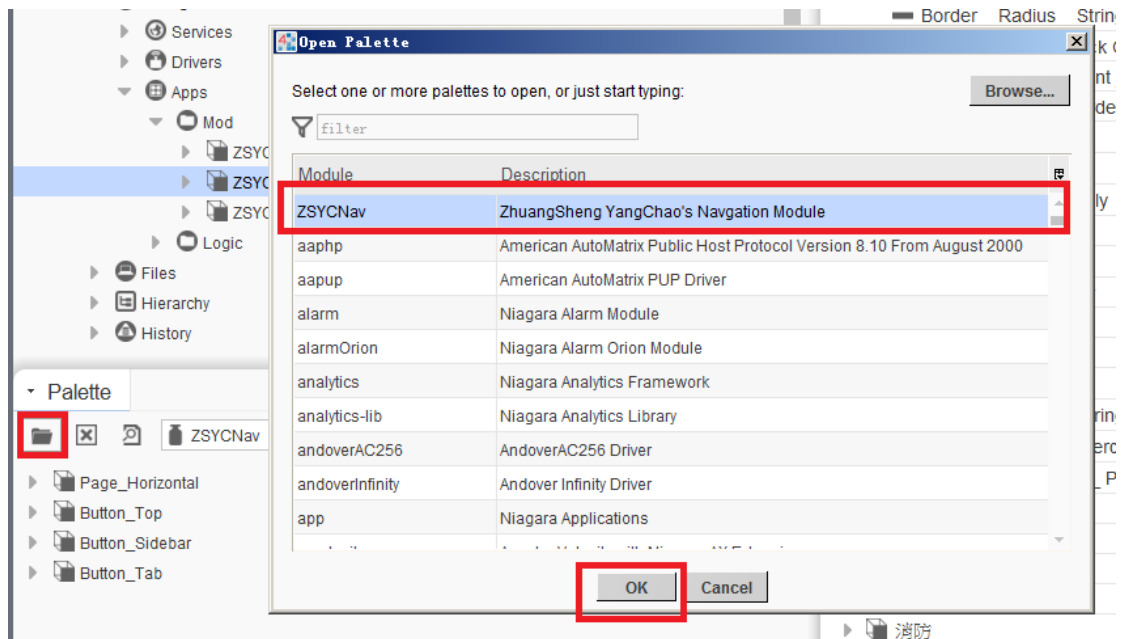
3.6.4 用户 NAV 文件和网页类型的选择



有多用户管理需求时，可建立配置多个安全组，建立配置多个 Role，建立配置多个 NAV 文件，建立配置多个 Page_Horizontal 模块对象用以对应多个用户。

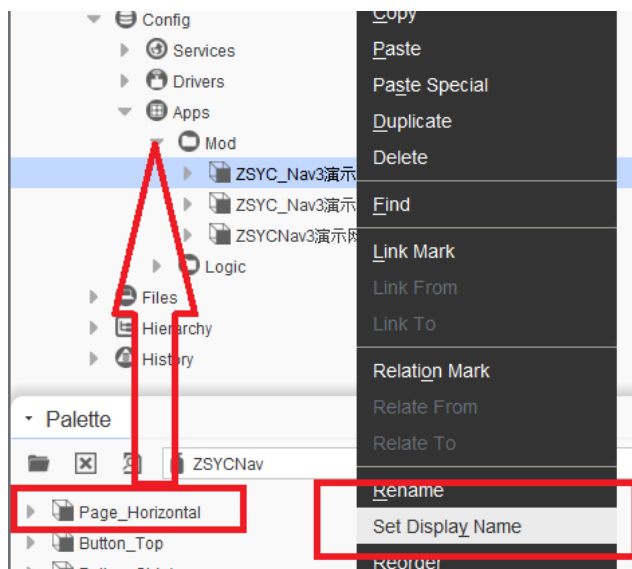
4 框架模块的简单使用

4.1 打开 ZSYCNav-ux 模块的 palate

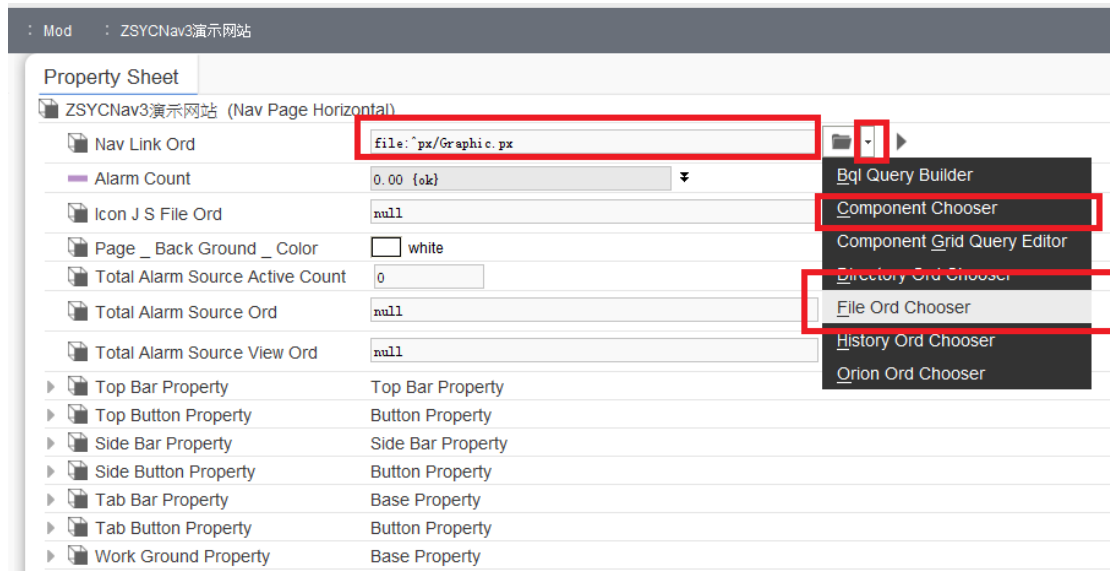


4.2 将 Page_Horizontal 拖入文件夹中

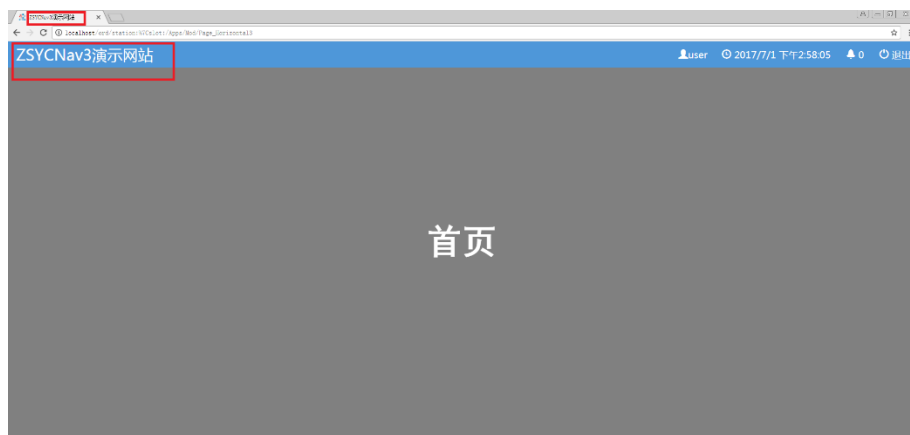
设置它的 displayName，对象 displayName 将为网站的标题。



4.3 Page_Horizontal 模块的 Nav_Link_Ord 可设置网站登陆后的默认视图



此时用新建的用户名密码登录 N4 网站，将会看到如下首页

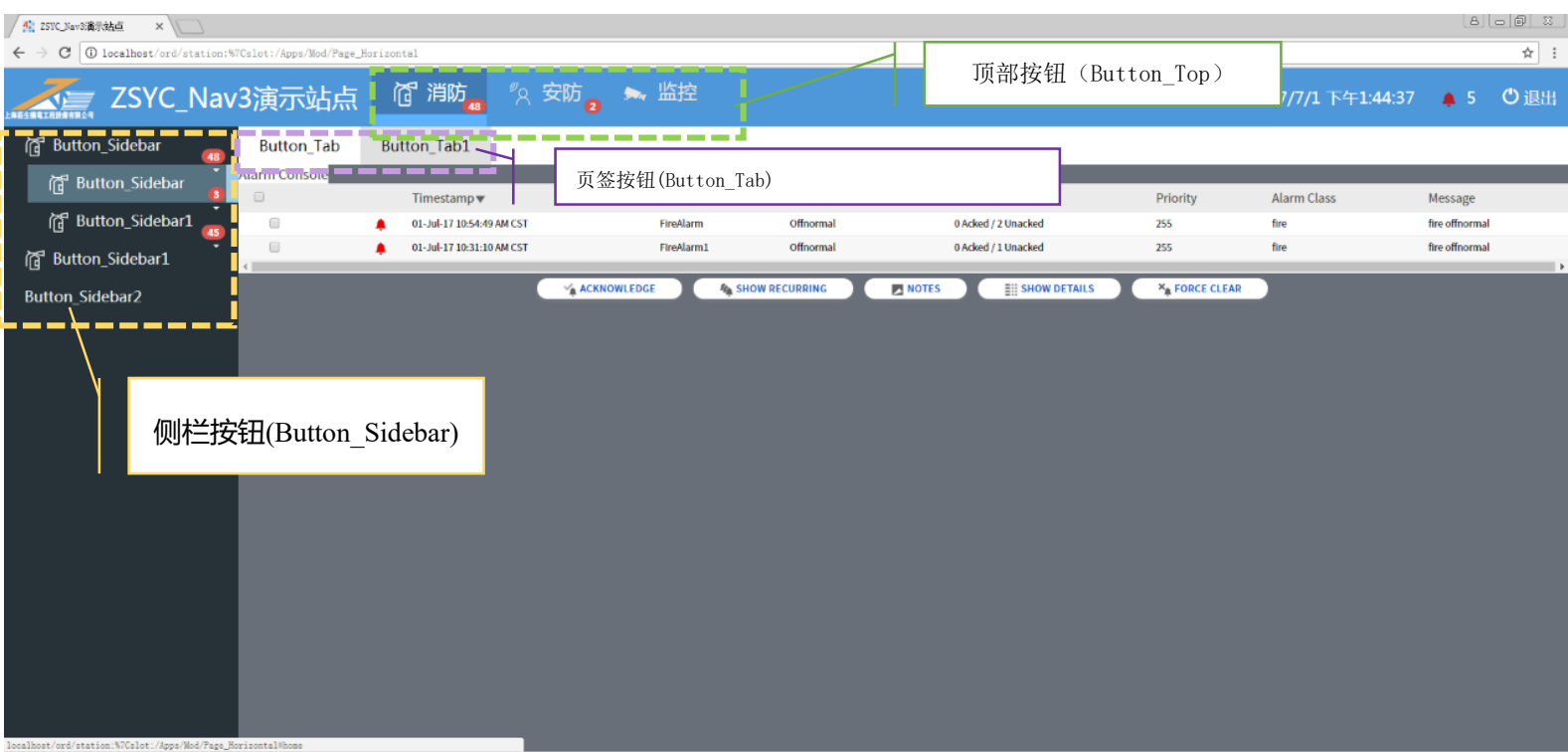


5 导航目录结构

将 Button_Top(顶栏按钮)或 Button_Sidebar (侧边栏按钮) 或 Button_Tab (页签) 拖入

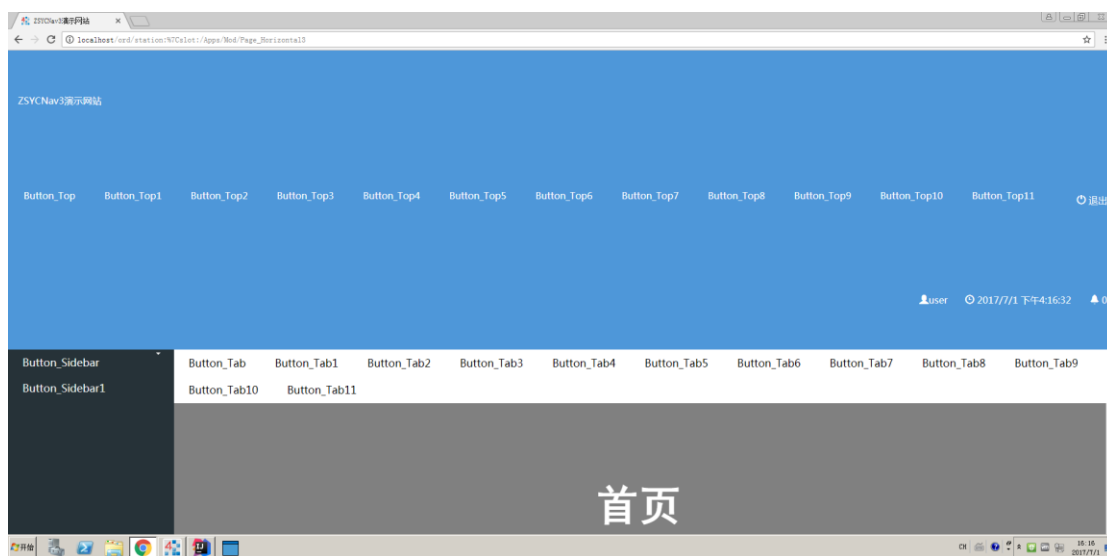
Page_Horizontal 对象下

各个对象所处区域如下图所示



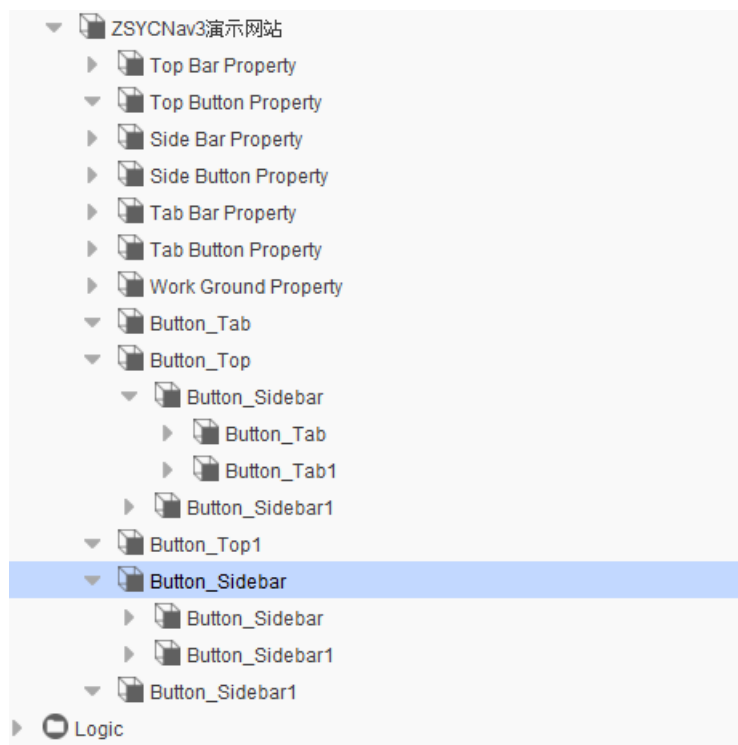
- 直接处于 Page_Horizontal 对象下的 Button_Top(顶栏按钮)或 Button_Sidebar (侧边栏按钮) 或 Button_Tab (页签) 将在首页直接显示，点击网站标题也将跳回首页。
- Button_Top(顶栏按钮)下可包含有 Button_Sidebar (侧边栏按钮) 或 Button_Tab (页签)，这些按钮将在点击该 Button_Top 的时候显示在相应区域。
- Button_Sidebar (侧边栏按钮)下可包含 Button_Sidebar (侧边栏按钮) 或 Button_Tab (页签)，最多嵌套 25 层。
- Button_Sidebar (侧边栏按钮)再包含 Button_Sidebar (侧边栏按钮)将表现为侧边栏按钮的多层结构。
- Button_Tab (页签)下不可再包含其他按钮。

注意：Button_Top(顶栏按钮)和 Button_Tab（页签）请不要过多，过多会使顶栏或页签栏变多层，不好看，侧边栏按钮无此限制



5.1 例子

设置按钮

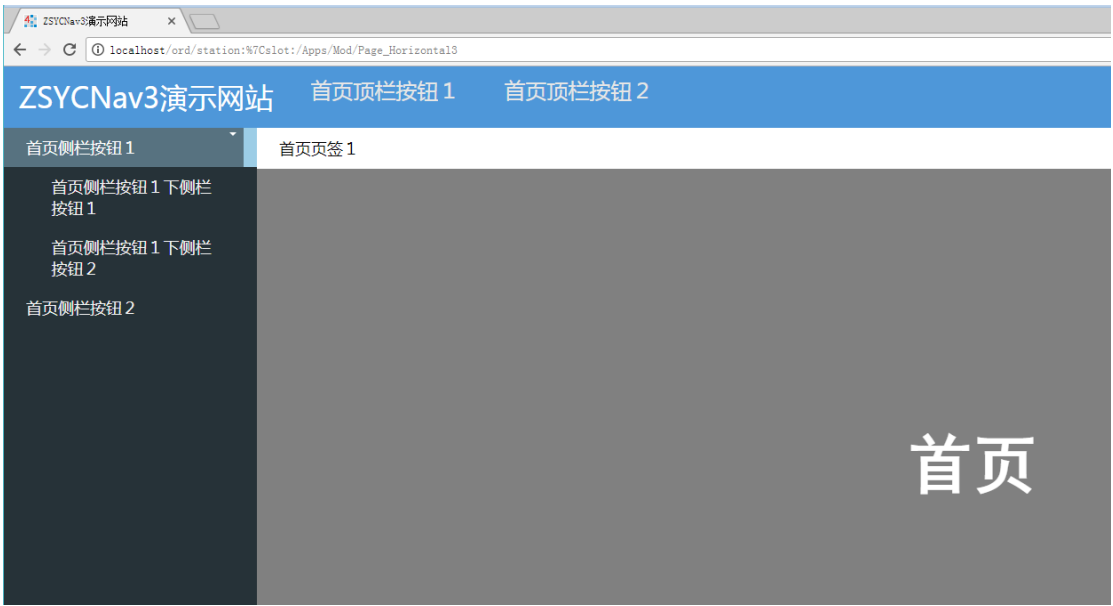


设置各个按钮的 displayName



5.2 上例效果

配置完按钮后首页



按下顶栏按钮 1 及其下的侧栏按钮 1



5.3基本界面布局

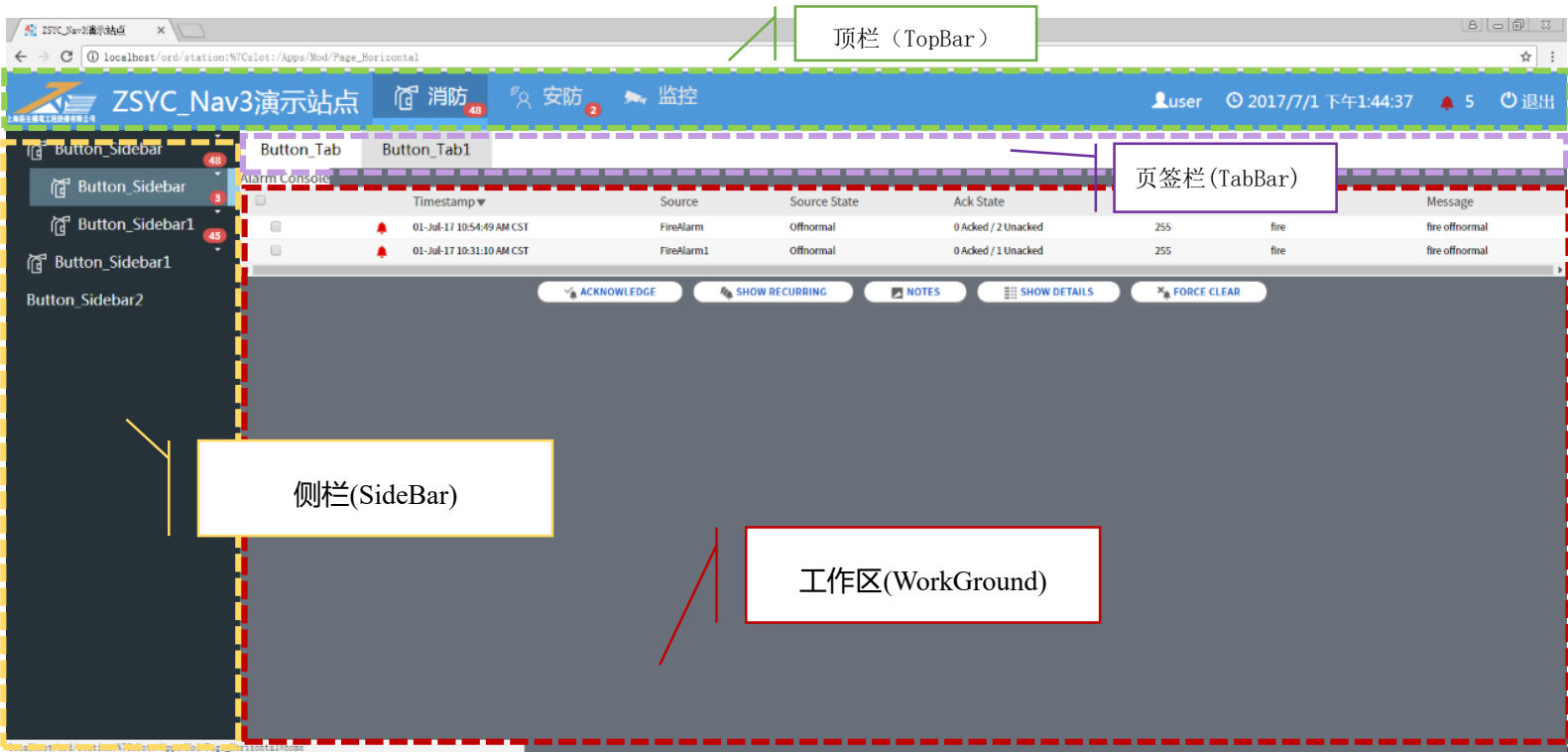
默认界面包括顶栏、侧栏、页签栏、工作区这四个大区域。

在 Page_Horizontal 对象的属性中，如下属性集合分别对应上述四大区域的显示属性：

- TopbarProperty
- SidebarProperty
- TabBarProperty
- WorkGroundProperty

ZSYCNav3演示网站 (Nav Page Horizontal)	
Nav Link Ord	file:~/px/Graphic.c.px
Alarm Count	0.00 {ok}
Icon J S File Ord	null
Page _ Back Ground _ Color	white
Total Alarm Source Active Count	0
Total Alarm Source Ord	null
Total Alarm Source View Ord	null
Top Bar Property	Top Bar Property
Top Button Property	Button Property
Side Bar Property	Side Bar Property
Side Button Property	Button Property
Tab Bar Property	Base Property
Tab Button Property	Button Property
Work Ground Property	Base Property
Button_Tab	Nav Button Tab
Button_Top	Nav Button Top
Button_Top1	Nav Button Top

其对应界面区域如下图所示



界面元素包括 Logo 图片、网站标题栏、顶部按钮、用户名信息、时间信息、总报警信息、退出登录按钮、侧栏按钮、页签按钮、工作区视图等元素。

在 Page_Horizontal 对象的属性中 Logo、标题、信息栏各个部件的显示属性包含在 TopbarProperty 里。

在 Page_Horizontal 对象的属性中如下属性集合分别对应上述顶部按钮、侧栏按钮、页签按钮的显示属性：

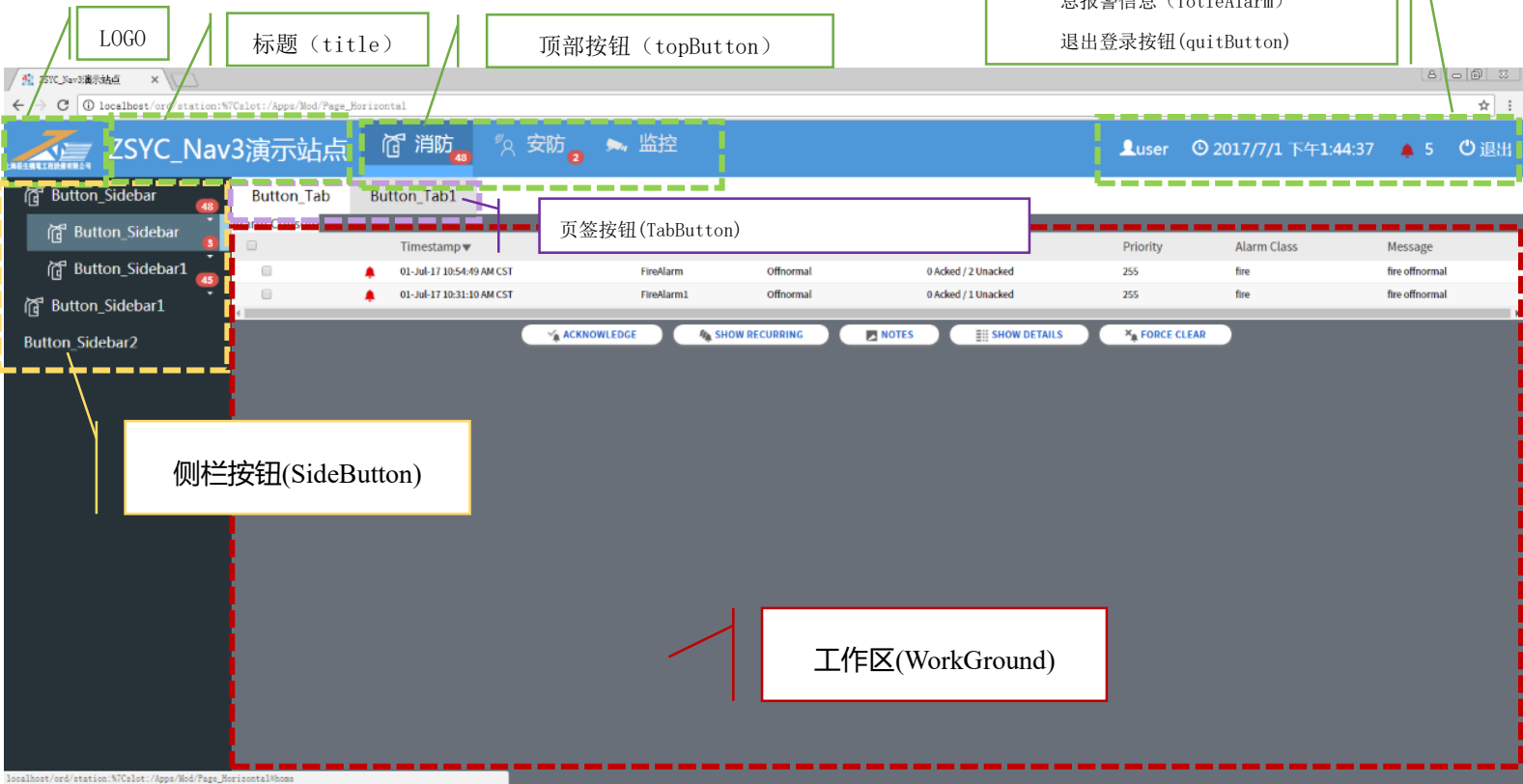
TopButtonProperty
SideButtonProperty
TabButtonProperty

ZSYCNav3演示网站 (Nav Page Horizontal)

Nav Link Ord	file:"px/Graphic.c.px"
Alarm Count	0.00 {ok}
Icon J S File Ord	null
Page _ Back Ground _ Color	white
Total Alarm Source Active Count	0
Total Alarm Source Ord	null
Total Alarm Source View Ord	null
Top Bar Property	Top Bar Property
Top Button Property	Button Property
Side Bar Property	Side Bar Property
Side Button Property	Button Property
Tab Bar Property	Base Property
Tab Button Property	Button Property
Work Ground Property	Base Property
Button_Tab	Nav Button Tab
Button_Top	Nav Button Top
Button_Top1	Nav Button Top

信息栏 (infoBar) , 包括:
用户名信息 (userinfo)
时间信息 (dateTime)
总报警信息 (TotleAlarm)
退出登录按钮 (quitButton)

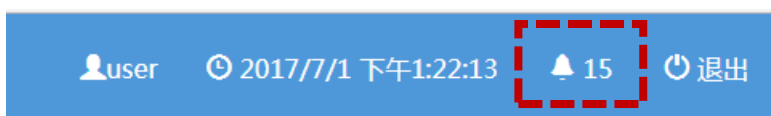
其分别对应的页面区域如下图所示标：



6 参数详解

6.1 Page_Horizontal 模块下属性

- 6.1.1 navLinkOrd : 该属性设定导航栏首页工作区的视图链接
- 6.1.2 alarmCount : 该属性为本人用的一个标志位属性, 会自动变化, 也不能设置, 请不要在意它。
- 6.1.3 iconJSFileOrd : 该属性设定图标所使用的图标的 JS 文件, 如何生成及下载图标 JS 文件请参考本文后面的第 10 章 “附录 4”。
- 6.1.4 page_BackGround_Color : 该属性设定整个页面的底色, 默认风格时是看不到的, 当区域设置外边距时会漏出底色。
- 6.1.5 totalAlarmSourceActiveCount : 该属性为标志位, 显示下面的总报警点属性 totalAlarmSourceOrd 所指向的对象里的 unackedAlarmCount 属性值, 界面上的总报警值会随该属性的变化而变化。此属性为一个标志位, 会自动变化, 不可设置, 请不要在意它。



- 6.1.6 totalAlarmSourceOrd : 该属性设定总报警点地址, 一般来说请指向某个报警类 alarmClass 对象, 该属性会取得目标点下的名字为 unackedAlarmCount 类型为 baja : Interg 的属性值并传送给 totalAlarmSourceActiveCount。如果取得总报警数有困难, 请参考本文后文第 9 章 “附录 3”。
- 6.1.7 totalAlarmSourceViewOrd : 该属性设定总报警点的视图, 当点击总报警的图标时, 工作区会自动切换至该属性指向的视图。如有困难请参考文后文第 9 章 “附录 2”。
- 6.1.8 topBarProperty : 该属性设置顶栏各个属性, 关于值的具体涵义及设定格式请见后文第 7 章 “附录 1”, 该属性下有如下属性:
 - 6.1.8.1 margin_String : 顶栏区域外边距设置, 默认 0px 即无外边距。
 - 6.1.8.2 backGround_Color : 顶栏背景色, 默认 #ff4e97d9
 - 6.1.8.3 front_Color : 顶栏内元素前景颜色, 默认 #ffffff 即白色, 该属性会影响到网站标题、用户信息、总警报信息、时间、退出按钮的文字颜色及其图标颜色。
 - 6.1.8.4 border_Color : 顶栏区域的边框颜色, 默认 null 即透明
 - 6.1.8.5 border_Width : 顶栏区域的边框宽度, 默认 0px 即无边框

- 6.1.8.6 border_Radius_String : 顶栏区域的圆角设定
- 6.1.8.7 isDisplay_Logo : 是否显示 LOGO
- 6.1.8.8 logo_ImgFileOrd : LOGO 的图片文件链接
- 6.1.8.9 isDisplay_Title : 是否显示网站标题
- 6.1.8.10 title_Font_Size : 网站标题字号, 单位 px
- 6.1.8.11 title_Font_Family : 字体, 默认 windows 系统是微软雅黑
- 6.1.8.12 infobar_Font_Size : 信息栏内用户名、日期、总警报数、退出按钮几个元素的字体大小
- 6.1.8.13 infobar_Font_Family : 信息栏内字体
- 6.1.8.14 infobar_Icon_Height : 信息栏内各个元素的图标大小, 注意如果图标是默认的未自定义, 则图标大小等于字体大小
- 6.1.8.15 isDisplay_User_Info : 是否显示用户名
- 6.1.8.16 isDisplay_DateTime_Info : 是否显示时间。(注意该时间是浏览器所在计算机的系统时间, 不是 N4 服务器上的时间)
- 6.1.8.17 isDisplay_TotleAlarmCount : 是否显示系统总报警数
- 6.1.8.18 isDisplay_Quit : 是否显示退出按钮
- 6.1.8.19 iconString_User : 用户名图标, 默认为空, 为空时显示 , 如需自定义, 则参考附录 4, 同按钮的图标设定。
- 6.1.8.20 iconString_DateTime : 时间图标, 默认为空, 为空时显示 
- 6.1.8.21 iconString_TotleAlarmCount : 警报图标, 默认为空时显示 
- 6.1.8.22 iconString_Quit : 退出按钮图标, 默认为空时显示 
- 6.1.9 topButtonProperty : 所有顶栏按钮的共同属性, 包括:
 - 6.1.9.1 margin_String : 顶栏按钮外边距设置。
 - 6.1.9.2 backGround_Color : 顶栏按钮背景色
 - 6.1.9.3 front_Color : 顶栏按钮内元素前景颜色。
 - 6.1.9.4 border_Color : 顶栏按钮的边框颜色
 - 6.1.9.5 border_Width : 顶栏按钮的边框宽度
 - 6.1.9.6 border_Radius_String : 顶栏按钮的圆角设定
 - 6.1.9.7 button_Active_BackGround_Color : 顶栏按钮按下后背景色。当鼠标于按钮上悬停时按钮背景色也会变为这个颜色
 - 6.1.9.8 button_Active_Front_Color : 顶栏按钮按下后字体颜色, 当鼠标于按钮上悬停时按钮字体颜色也会变为这个颜色
 - 6.1.9.9 button_Active_Border_Color : 顶栏按钮按下后边框颜色。
 - 6.1.9.10 icon_Height : 顶栏按钮上图标的 height。
 - 6.1.9.11 button_Font_Size : 顶栏按钮上的字号大小, 单位 px
 - 6.1.9.12 button_Font_Family : 顶栏按钮上的文字的字体。
- 6.1.10 sideBarProperty : 侧边栏区域的属性, 包括:
 - 6.1.10.1 margin_String : 侧边栏区域外边距设置。

6.1.10.2 backGround_Color : 侧边栏区域背景色

6.1.10.3 front_Color : 侧边栏区域内元素前景颜色。

6.1.10.4 border_Color : 侧边栏区域的边框颜色

6.1.10.5 border_Width : 侧边栏区域的边框宽度

6.1.10.6 border_Radius_String : 侧边栏区域的圆角设定

6.1.10.7 sideBar_Width_Percent : 侧边栏宽度占页面宽度的百分比, 单位为%, 默认值为 15, 即侧边栏宽度占当前页面宽度的 15%。

6.1.10.8 sideBar_Collapse_Padding_Left_Percent : 侧边栏多级子菜单按钮缩进量占按钮宽度的百分比, 默认值为 15, 即下级子菜单按钮会向右边缩进其父菜单按钮宽度的 15%。



6.1.11 sideButtonProperty : 所有侧边栏按钮的共同属性, 包括 :

6.1.11.1 margin_String : 侧边栏按钮外边距设置

6.1.11.2 front_Color : 侧边栏按钮前景颜色

6.1.11.3 border_Color : 侧边栏按钮的边框颜色

6.1.11.4 border_Width : 侧边栏按钮的边框宽度

6.1.11.5 border_Radius_String : 侧边栏按钮的圆角设定

6.1.11.6 button_Active_BackGround_Color : 侧边栏按钮按下后背景色。当鼠标于按钮上悬停时按钮背景色也会变为这个颜色

6.1.11.7 button_Active_Front_Color : 侧边栏按钮按下后字体颜色, 当鼠标于按钮上悬停时按钮字体颜色也会变为这个颜色

6.1.11.8 button_Active_Border_Color : 侧边栏按钮按下后边框颜色。

6.1.11.9 icon_Height : 侧边栏按钮上图标的高度。

6.1.11.10 button_Font_Size : 侧边栏按钮上的字号大小, 单位 px

6.1.11.11 button_Font_Family:侧边栏按钮上的文字的字体的。

6.1.12 tabBarProperty : 页签栏区域的属性, 包括 :

6.1.12.1 margin_String : 页签栏区域外边距设置

6.1.12.2 front_Color : 页签栏区域前景颜色

6.1.12.3 border_Color : 页签栏区域的边框颜色

6.1.12.4 border_Width : 页签栏区域的边框宽度

6.1.12.5 border_Radius_String : 页签栏区域的圆角设定

6.1.13 tabButtonProperty : 所有页签按钮的共同属性, 包括 :

6.1.13.1 margin_String : 页签按钮外边距设置。

6.1.13.2 backGround_Color : 页签按钮背景色

- 6.1.13.3 front_Color : 页签按钮前景颜色。
- 6.1.13.4 border_Color : 页签按钮的边框颜色
- 6.1.13.5 border_Width : 页签按钮的边框宽度
- 6.1.13.6 border_Radius_String : 页签按钮的圆角设定
- 6.1.13.7 button_Active_BackGround_Color : 页签按钮按下后背景色。当鼠标于按钮上悬停时按钮背景色也会变为这个颜色
- 6.1.13.8 button_Active_Front_Color : 页签按钮按下后字体颜色，当鼠标于按钮上悬停时按钮字体颜色也会变为这个颜色
- 6.1.13.9 button_Active_Border_Color : 页签按钮按下后边框颜色。
- 6.1.13.10 icon_Height : 页签按钮上图标的 height。
- 6.1.13.11 button_Font_Size : 页签按钮上的字号大小，单位 px
- 6.1.13.12 button_Font_Family: 页签按钮上的文字的字体。
- 6.1.14 workGroundProperty : 工作区域属性，包括：
 - 6.1.14.1 margin_String : 工作区域外边距设置。
 - 6.1.14.2 backGround_Color : 顶栏背景色
 - 6.1.14.3 front_Color : 顶栏内元素前景颜色。
 - 6.1.14.4 border_Color : 工作区域的边框颜色
 - 6.1.14.5 border_Width : 工作区域的边框宽度
 - 6.1.14.6 border_Radius_String : 工作区域的圆角设定

6.2 按钮属性

除了在 Page_Horizontal 模块下的 topButtonProperty、sideButtonProperty、tabButtonProperty 规定的各自区域下的按钮的共通外观属性外，每个按钮还有自己独属的一些专用属性，这些属性包括：

Property Sheet	
Button_Tab (Nav Button Tab)	
Nav Http U R I String	{ok}
Nav Link Ord	null
Alarm Count	0.00 {ok}
Alarm Source Active Count	0
Alarm Source Ord	null
Icon String	{ok}
Button_Back Ground_Img_Ord	null
Button_Active_Back Ground_Img_Ord	null

6.2.1 NavHttpURISting：该按钮链接到的外部网址属性。

6.2.2 Nav_Link_Ord：该按钮链接到的站点内点的属性。

这两个属性指定了点击该按钮后工作区显示哪个页面。

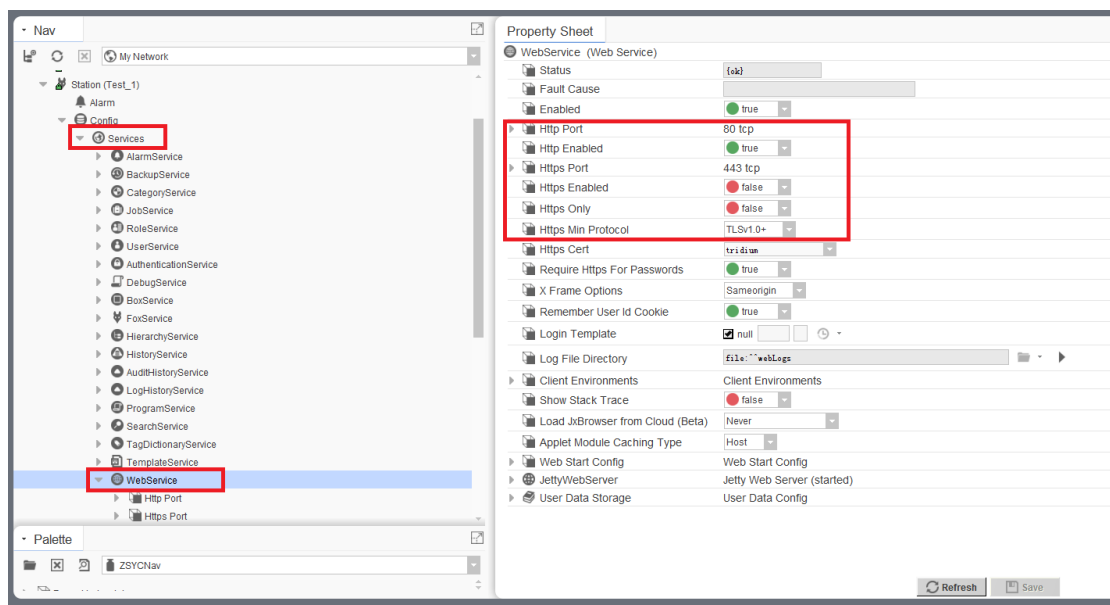
Property Sheet	
Button_Top总顶栏按钮 (Nav Button Top)	
Nav Http U R I String	http://www.sh-zs.cn {ok}
Nav Link Ord	null
Alarm Count	0.00 {ok}
Alarm Source Active Count	0
Alarm Source Ord	null
Icon String	{ok}
Button_Back Ground_Img_Ord	null
Button_Active_Back Ground_Img_Ord	null

注意，NavHttpURISting 必须包含其 Http 或 Https 的地址头部，比如如果想将 www.sh-zs.cn 整合至系统中，则必须在 NavHttpURISting 中输入 [http:// www.sh-zs.cn](http://www.sh-zs.cn)

6.2.3 在 NavHttpURISTing 中输入 http:// www.sh-zs.cn 的效果



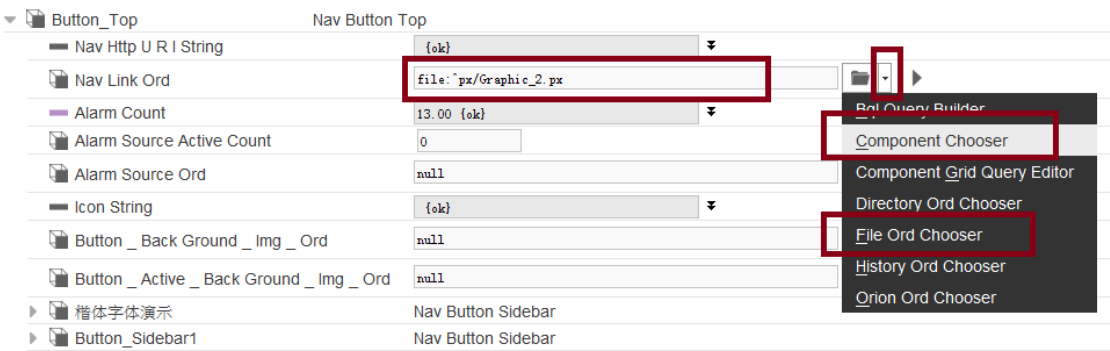
注意，如果引入的是网络安全协议的 [Https://](https://) 开头网址，无需特殊设定，但如果需要在 NavHttpURISTing 中整合进 <http://> 开头网址，则同时必须将 Niagara4 站点设置为 http 网站，设置方法如下：



注意，若这两个同时有值时，工作区将显示 NavHttpURISTing 所指向网页，建议这两个值只有一个有值。

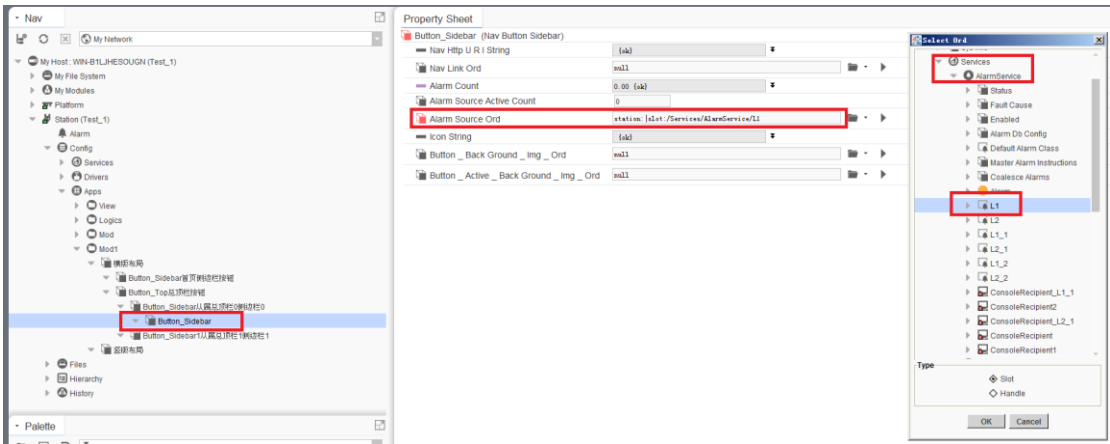
6.2.4 Nav_Link_Ord：此属性指定该按钮所链接的站内视图，既可以用 Component 选择器选择站点里的点,也可以用 File 选择器选择站点里的 px 文件。

注意：当 NavHttpURISting 不为空时，按钮将优先链接至 NavHttpURISting 目标网站。



6.2.5 按钮报警图章显示：

Alarmsourceord 属性可以绑定站点的 Services/AlarmService/下的某个 AlarmService 类，将指定的 alarmService 里的未确认报警条数绑定到该按钮的右下角小印章中。



注意，底层按钮的报警条数将自动汇总到其上层按钮中，前提是它上层的按钮的 Alarmsourceord 属性为空。

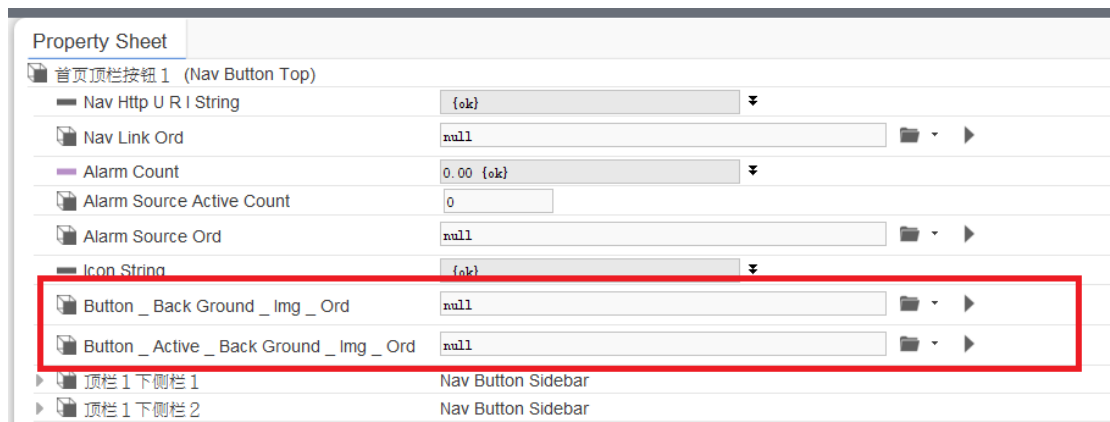
注意，若某个按钮的 Alarmsourceord 指向了某个 alarmService，则其报警值就是该 AlarmService 的报警数，而不会统计其下子模块的报警数目。

6.2.6 Icon_String 设置该按钮的自定义图标，具体设置过程请参见后文章节 10 “附录 4”

6.2.7 设置按钮自定义背景图片：按钮可具体设置某个按钮的背景图片，具体有如

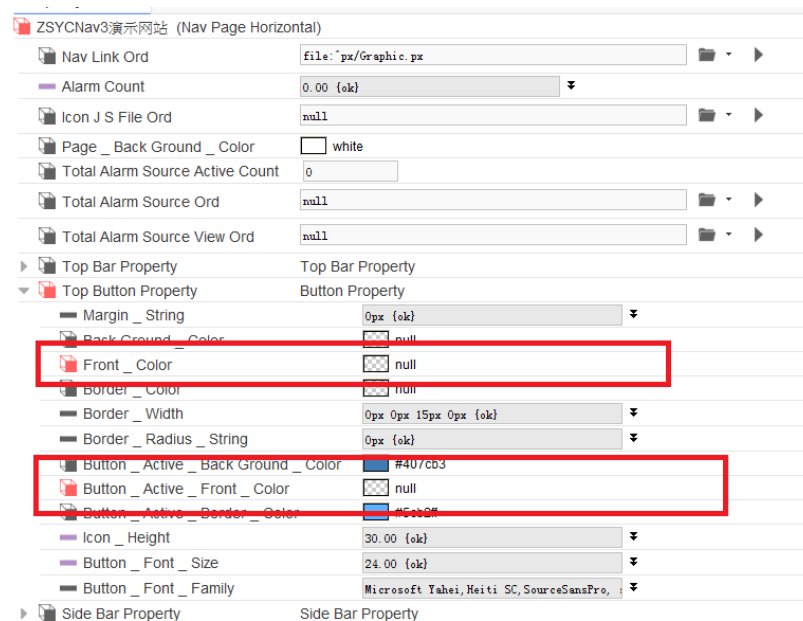
下两个属性：

- Button _ Back Ground _ Img _ Ord 属性：设置按钮未按下时的背景图片。
- Button _ Active _ Back Ground _ Img _ Ord 属性：设置按钮按下后的背景图片。

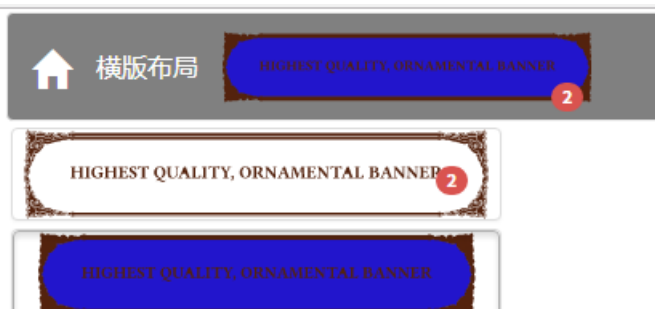


注意：

如需隐藏按钮本身的文字，使用图片里的文字替代，可设置具体按钮类型里的字体颜色为 null，这样这些文字会变透明。



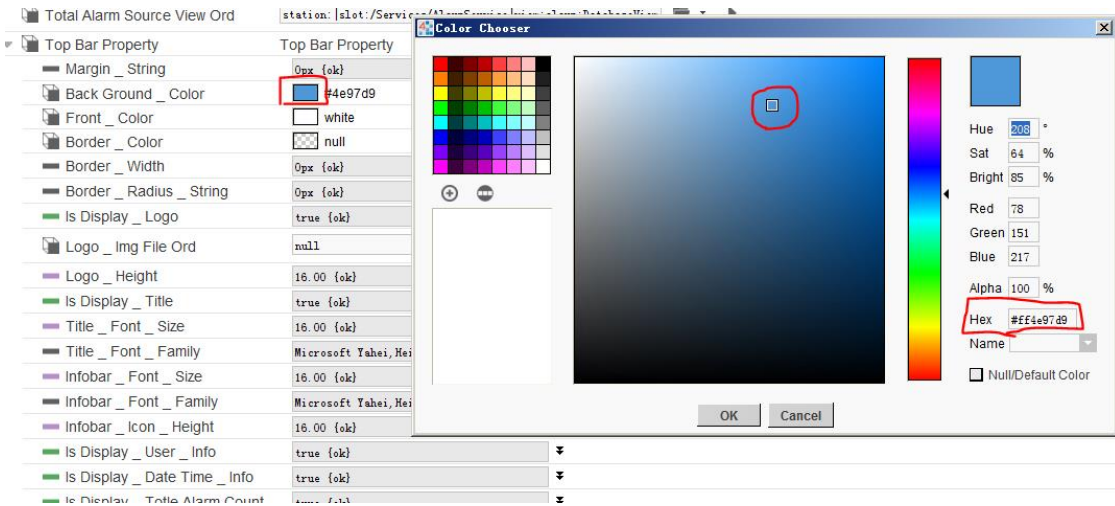
您可以每一个按钮设置图片，用以制作一些特殊按钮（比如说把按钮做成魔兽世界风格的）。该方法适用于您需要定制化个性化的按钮以及按钮上的字体。



7 附录 1：几种属性的涵义、格式

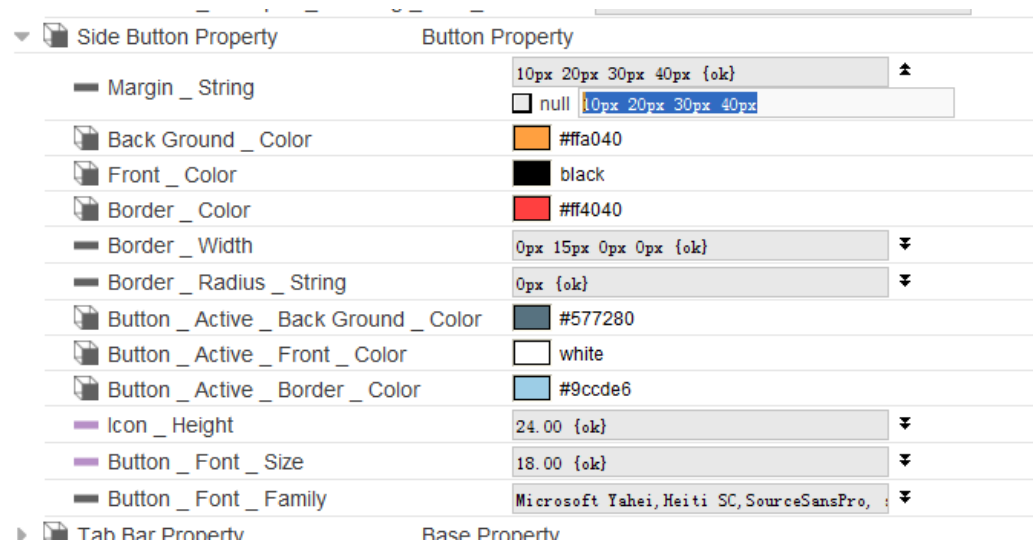
7.1 颜色：

可点击属性旁边的颜色方框，进入 N4 标准的颜色设置界面，选取颜色，或输入具体的颜色代码来设置颜色属性。

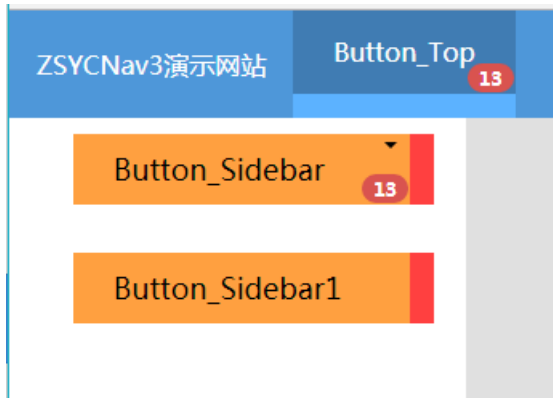


7.2 外边距 margin_String

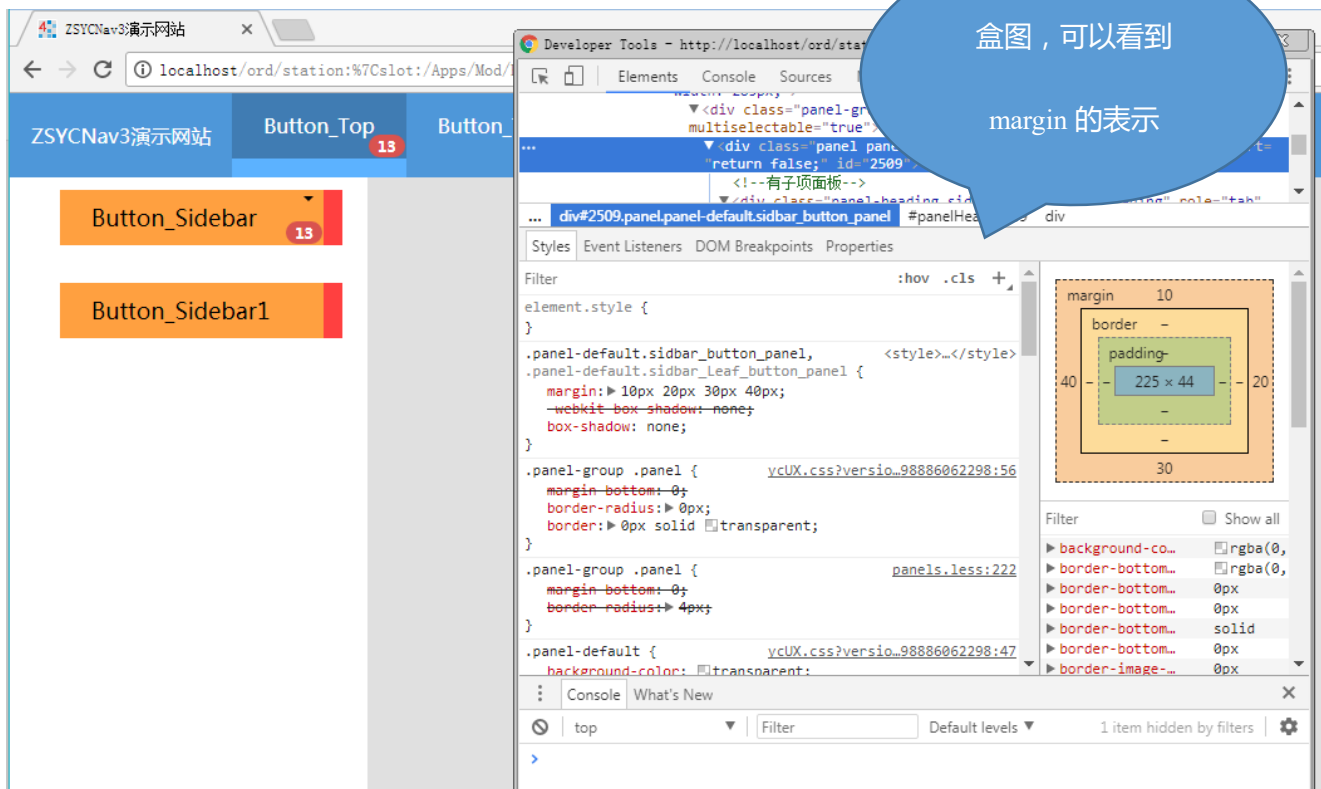
设定相应对象与其他对象的距离，有两种格式，一种为直接写一个数值单位，比如直接写“15px”，则意味着该对象的上、右、下、左四个方向的外边距都是 15px，第二种方式为写形如“10px 20px 30px 40px”的字符串，有 4 个数值单位，中间有空格隔开，这样的形式分别指定了该对象的上、右、下、左四个方向的外边距，具体如“10px 20px 30px 40px”指定的就是该对象上外边距 10px，右外边距 20px，下外边距 30px，左外边距 40px。例如，将 sideButtonProperty 里的 margin_String 属性改为“10px 20px 30px 40px”



此处为了让结果看起来更明显，将按钮颜色改为明显的黄色，将侧边栏底色变为白色。得到的结果为：



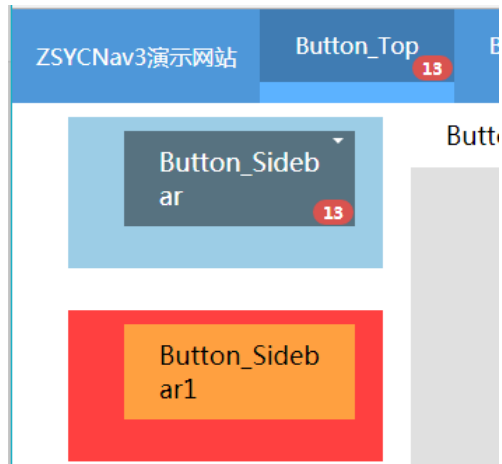
使用 chrome 的调试工具可以更清楚的了解外边距的概念



7.3 边框宽度 border_Width:

设定相应对象的边框宽度，有两种格式，一种为直接写一个数值单位，比如直接写“15px”，则意味着该对象的上、右、下、左四个方向的边框宽度都是 15px，第二种方式为写形如“10px 20px 30px 40px”的字符串，有 4 个数值单位，中间有空格隔开，这样的形式分别指定了该对象的上、右、下、左四个方向的边框宽度，具体如“10px 20px 30px 40px”指定的就是该对象上边框宽度 10px，右边框宽度 20px，下边框宽度 30px，左边框宽度 40px。

例如，将上例中的 sideButtonProperty 里的 border_Width 属性改为 “10px 20px 30px 40px” 得到的结果为：

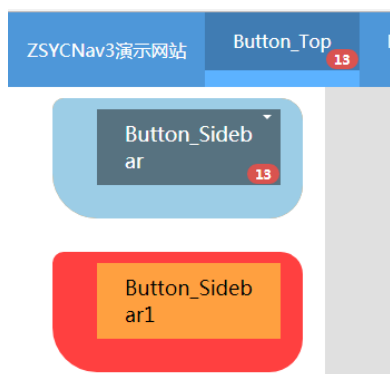


7.4 圆角 border_Radius_String

设定相应对象的圆角大小，有两种格式，一种为直接写一个数值单位，比如直接写 “15px”，则意味着该对象的上、右、下、左四个方向的圆角大小都是 15px，第二种方式为写形如 “10px 20px 30px 40px” 的字符串，有 4 个数值单位，中间有空格隔开，这样的形式分别指定了该对象的上、右、下、左四个方向的圆角大小，具体如 “10px 20px 30px 40px” 指定的就是该对象上圆角大小 10px，右圆角大小 20px，下圆角大小 30px，左圆角大小 40px。

例如，将上例中的 sideButtonProperty 里的 border_Radius_String 属性改为 “10px 20px 30px 40px”

得到的结果为：



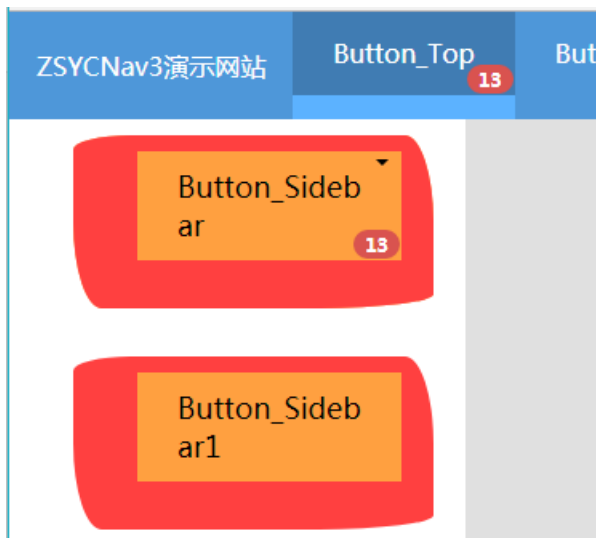
也可以将圆角设置为一个很大的数字，则这时就可以得到椭圆形的按钮，例如将上例中的 sideButtonProperty 里的 border_Radius_String 属性改为 “100px”，结果为：



注意这里面的不均匀是因为边框厚度的不统一。

圆角可以有更多更复杂的用法，请自行百度 border-Radius，在此我演示一个从网上看到的例子。

例如将上例中的 sideButtonProperty 里的 border_Radius_String 属性改为 “36px 18px 72px / 9px 54px”，结果为：



7.5 字体 Font_Family:

font-family 规定元素的字体系列。

font-family 可以把多个字体名称作为一个“回退”系统来保存。如果浏览器不支持第一个字体，则会尝试下一个。也就是说，font-family 属性的值是用于某个元素的字体族名称或/及类族名称的一个优先表。浏览器会使用它可识别的第一个值。

有两种类型的字体系列名称：

指定的系列名称：具体字体的名称，比如："times"、"courier"、"arial"。

通常字体系列名称：比如："serif"、"sans-serif"、"cursive"、"fantasy"、"monospace"

提示：使用逗号分割每个值，并始终提供一个类族名称作为最后的选择。

注意：使用某种特定的字体系列（Geneva）完全取决于用户机器上该字体系列是否可用；这个属性没有指示任何字体下载。因此，强烈推荐使用一个通用字体系列名作为后路。

本系统的字体的默认值都为“Microsoft Yahei,Heiti SC,SourceSansPro, sans-serif”，即最优先微软雅黑，苹果系统上的黑体次之，linux 系统下的 SourceSansPro 次之，sans-serif（基本都有）最后。

如需自定义，请百度 font-family，并将查到的字体名加在这个字符串第一位并添加英文输入法的逗号。

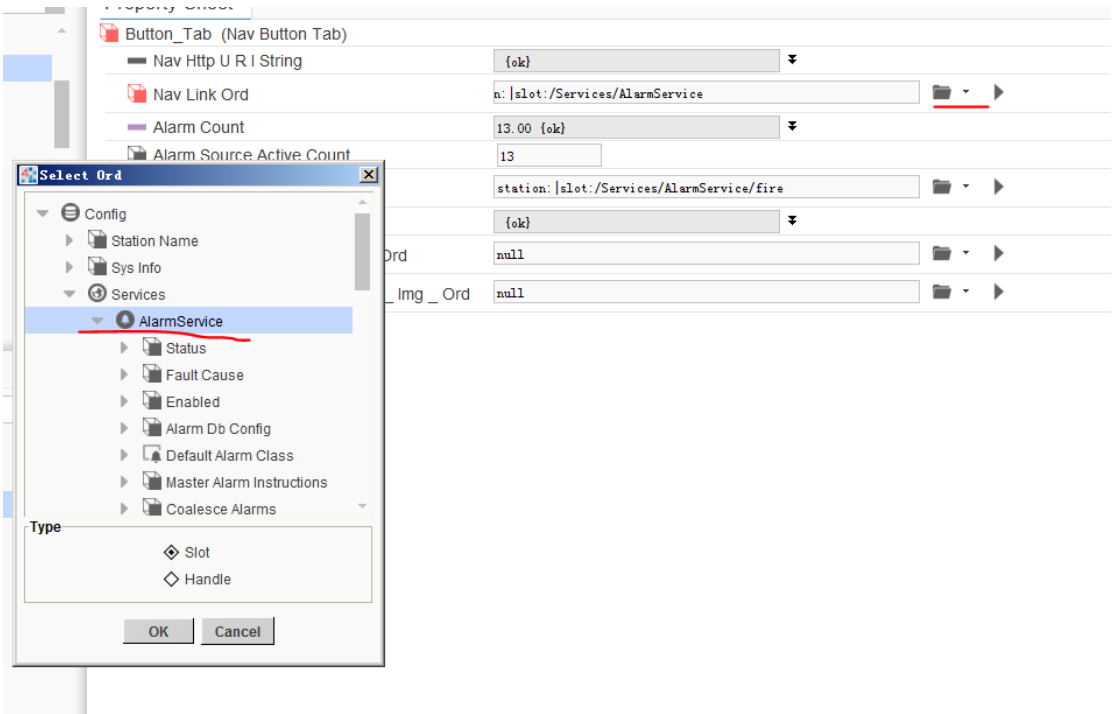
比如说想添加楷体，则将字体属性改为“KaiTi,Microsoft Yahei,Heiti SC,SourceSansPro, sans-serif”，改后其结果显示为：



8 附录 2：特殊视图的链接

有些特殊视图，直接试用 NavLinkOrd 属性的 Component 选择器不好选中，则可以使用下述方式输入到 NavLinkOrd 里去，下文以链接 AlarmService 的 DatabaseView 视图为例：

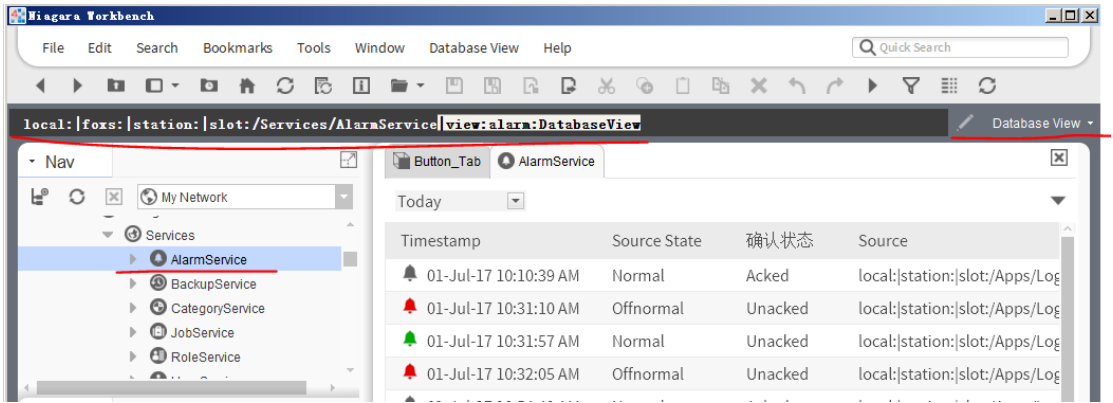
8.1 在按钮的 NavLinkOrd 属性用 Component 选择器选中 AlarmService 并保存



8.2 在 workBench 中直接打开 AlarmService，并选择到 DatabaseView 中，然后点击

地址栏，选择 “local:|foxs:|station:|slot:/Services/AlarmService” 后的

“|view:alarm:DatabaseView” 段落并复制。



8.3 将刚才的段落添加到 NavLinkOrd 值的末尾，并保存。

Button_TabAlarmService

Property Sheet

Button_Tab (Nav Button Tab)

Nav Http U R I String

{ok}

Nav Link Ord

station:|slot:/Services/AlarmService|view:alarm.DatabaseView

Alarm Count

13.00 {ok}

Alarm Source Active Count

13

Alarm Source Ord

station:|slot:/Services/AlarmService/fire

Icon String

{ok}

Button _ Back Ground _ Img _ Ord

null

Button _ Active _ Back Ground _ Img _ Ord

null

8.4 结果：

2310Nav3演示网站

Button_Top15Button_Top1

Button_Tab15

Today

Timestamp	Source State	Ack State	Source	Alarm Class	Priority	Message Text
01-Jul-17 10:10:39 AM	Normal	Acked	local: station: slot:/Apps/Logic/FireAlarm/BooleanChangeOfStateAlarmExt	fire	255	fire to normal
01-Jul-17 10:31:10 AM	Offnormal	Unacked	local: station: slot:/Apps/Logic/FireAlarm1/BooleanChangeOfStateAlarmExt	fire	255	fire offnormal
01-Jul-17 10:31:57 AM	Normal	Unacked	local: station: slot:/Apps/Logic/Raido/BooleanChangeOfStateAlarmExt	Raido	255	Raido to normal
01-Jul-17 10:32:05 AM	Offnormal	Unacked	local: station: slot:/Apps/Logic/Raido/BooleanChangeOfStateAlarmExt	Raido	255	Raido offnormal
01-Jul-17 10:54:49 AM	Normal	Acked	local: station: slot:/Apps/Logic/FireAlarm/BooleanChangeOfStateAlarmExt	fire	255	fire to normal
01-Jul-17 11:52:47 AM	Normal	Unacked	local: station: slot:/Apps/Logic/FireAlarm/BooleanChangeOfStateAlarmExt	fire	255	fire to normal
01-Jul-17 11:54:53 AM	Normal	Unacked	local: station: slot:/Apps/Logic/FireAlarm/BooleanChangeOfStateAlarmExt	fire	255	fire to normal
01-Jul-17 11:55:47 AM	Normal	Unacked	local: station: slot:/Apps/Logic/FireAlarm/BooleanChangeOfStateAlarmExt	fire	255	fire to normal
01-Jul-17 11:59:56 AM	Normal	Unacked	local: station: slot:/Apps/Logic/FireAlarm/BooleanChangeOfStateAlarmExt	fire	255	fire to normal
01-Jul-17 12:00:41 PM	Normal	Unacked	local: station: slot:/Apps/Logic/FireAlarm/BooleanChangeOfStateAlarmExt	fire	255	fire to normal
01-Jul-17 12:02:30 PM	Normal	Unacked	local: station: slot:/Apps/Logic/FireAlarm/BooleanChangeOfStateAlarmExt	fire	255	fire to normal
01-Jul-17 12:03:12 PM	Normal	Unacked	local: station: slot:/Apps/Logic/FireAlarm/BooleanChangeOfStateAlarmExt	fire	255	fire to normal
01-Jul-17 12:17:13 PM	Normal	Unacked	local: station: slot:/Apps/Logic/FireAlarm/BooleanChangeOfStateAlarmExt	fire	255	fire to normal
01-Jul-17 1:17:52 PM	Normal	Unacked	local: station: slot:/Apps/Logic/FireAlarm/BooleanChangeOfStateAlarmExt	fire	255	fire to normal
01-Jul-17 1:19:43 PM	Normal	Unacked	local: station: slot:/Apps/Logic/FireAlarm/BooleanChangeOfStateAlarmExt	fire	255	fire to normal
01-Jul-17 1:20:12 PM	Offnormal	Unacked	local: station: slot:/Apps/Logic/FireAlarm/BooleanChangeOfStateAlarmExt	fire	255	fire offnormal
01-Jul-17 5:22:40 PM	Normal	Acked	local: station: slot:/Apps/Logic/FireAlarm/BooleanChangeOfStateAlarmExt	fire	255	fire to normal

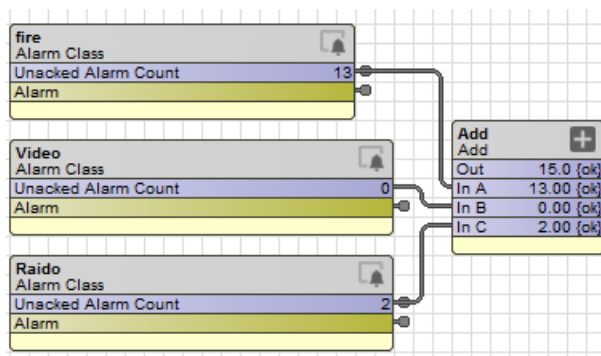
PREVIOUSPage 1 of 1NEXTResults: 100 per page

9 附录 3：总报警数的取得

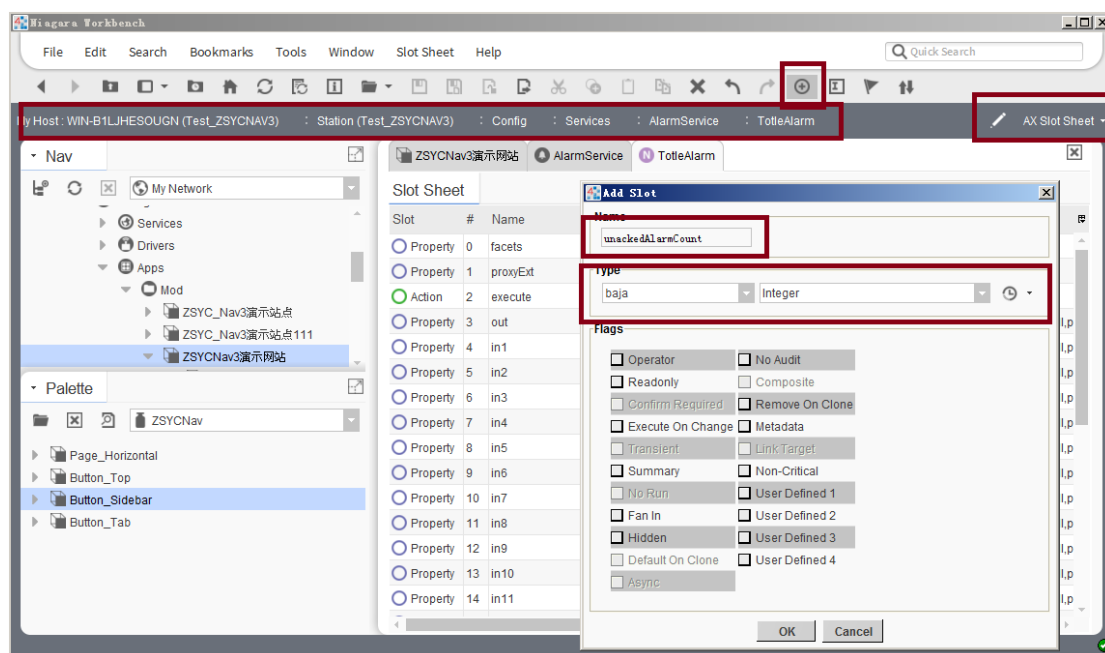
Paga_Horizontal 对象的 totalAlarmSourceOrd 属性设定系统的总报警点地址，一般来说请指向某个报警类 alarmClass 对象，该属性会取得目标点下的名字为 unackedAlarmCount 类型为 baja : Interg 的属性值并传送给 totalAlarmSourceActiveCount。

在 NiagaraN4 里，多个 AlarmClass 汇总到一个 AlarmClass 里的结果并不总是正确，这让更多 AlarmClass 的系统中获得整个系统的总报警数成为了一个问题，本系统内可以这么解决：

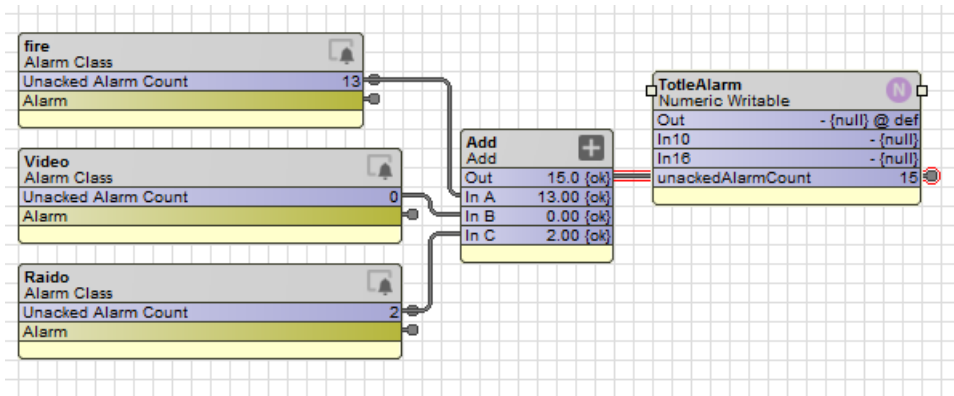
9.1 将各个 alarmClass 的 unackedAlarmCount 值取出并相加。



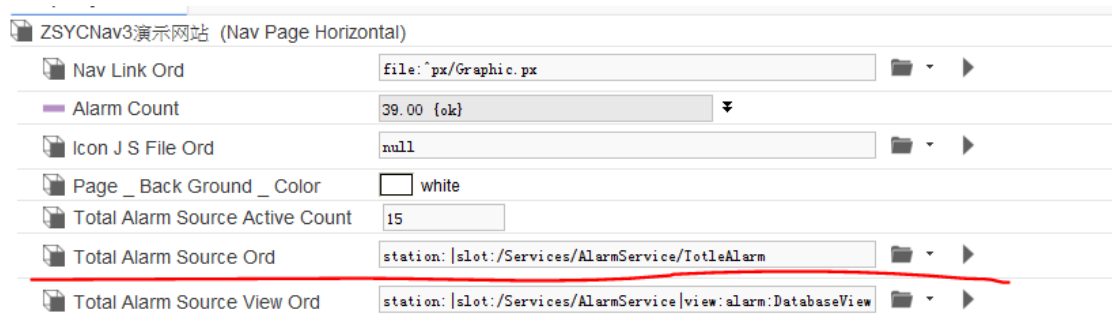
9.2 新建一个 NumericWritable 点，进入它的 AX Slot Sheet 视图，建立一个名为 unackedAlarmCount 类型为 baja : Interg 的 Slot。



9.3 将相加的值传给新建立的这个 Slot。



9.4 然后在 Paga_Hrizontal 的 totalAlarmSourceOrd 属性里添加这个点地址。



如此可取到这几个类的报警总数了。

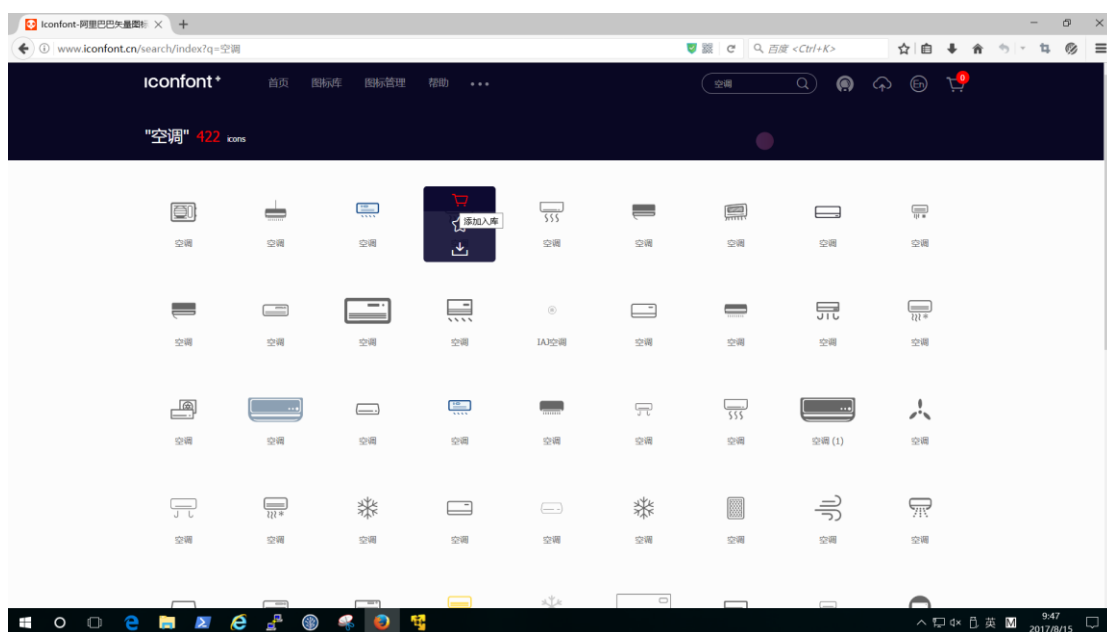
10 附录 4：自定义图标

10.1 下载图标文件

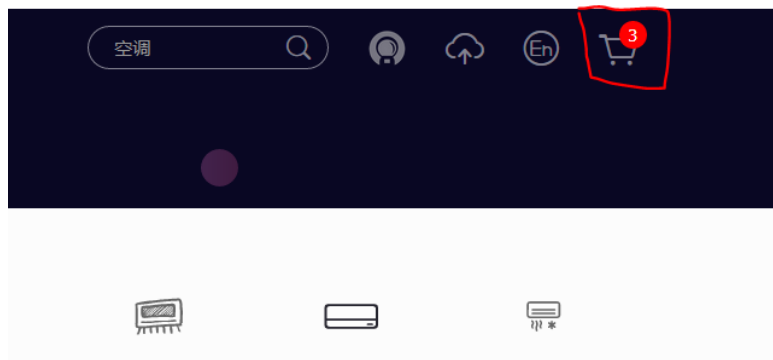
10.1.1 至 <http://www.iconfont.cn/> 上下载图标，登陆 <http://www.iconfont.cn/> 并使用注册使用账号登陆

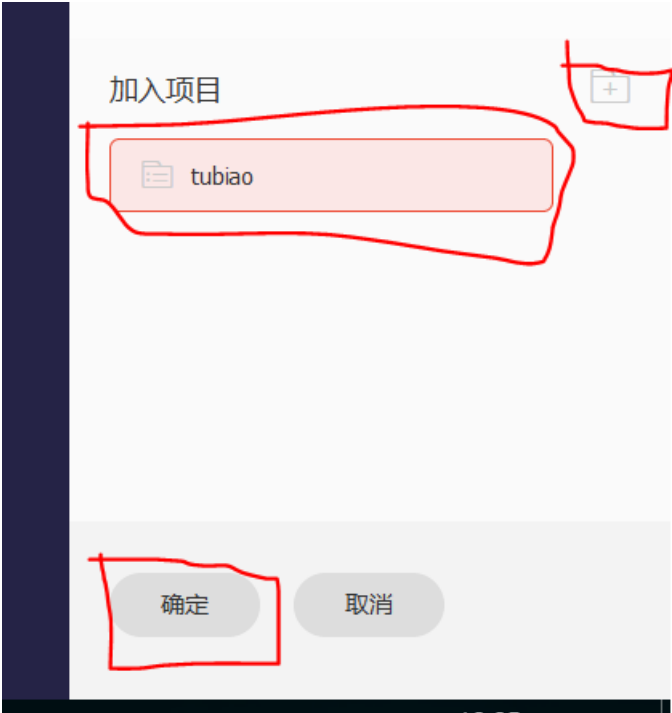
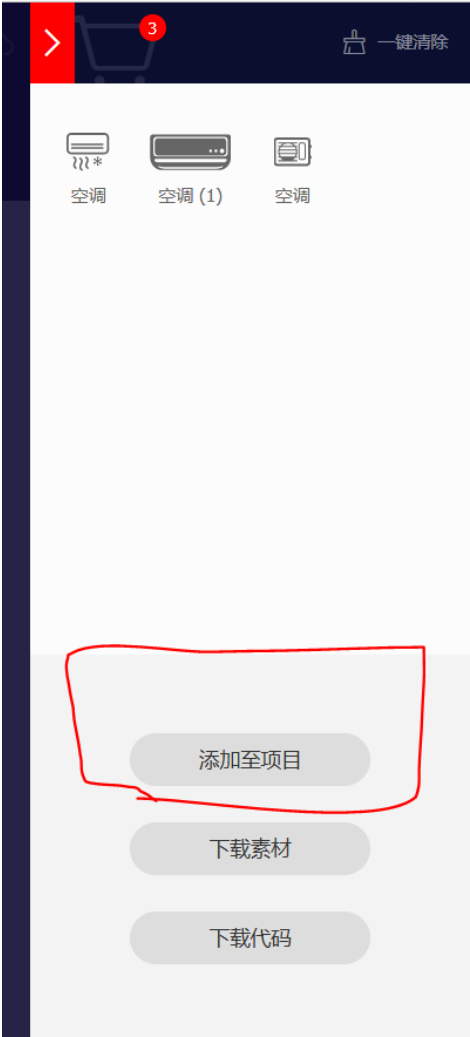
10.1.2 在 <http://www.iconfont.cn/> 搜索要使用的图标

10.1.3 将中意的图标添加至购物车里（基本都是免费）

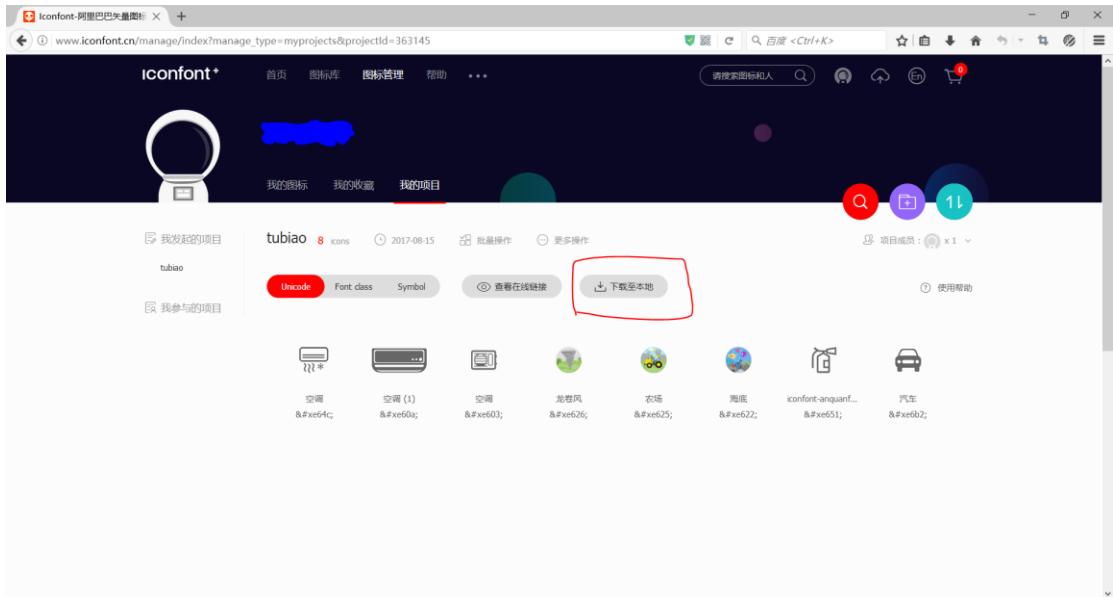


10.1.4 进入购物车，将选择好的几个图标添加至项目。

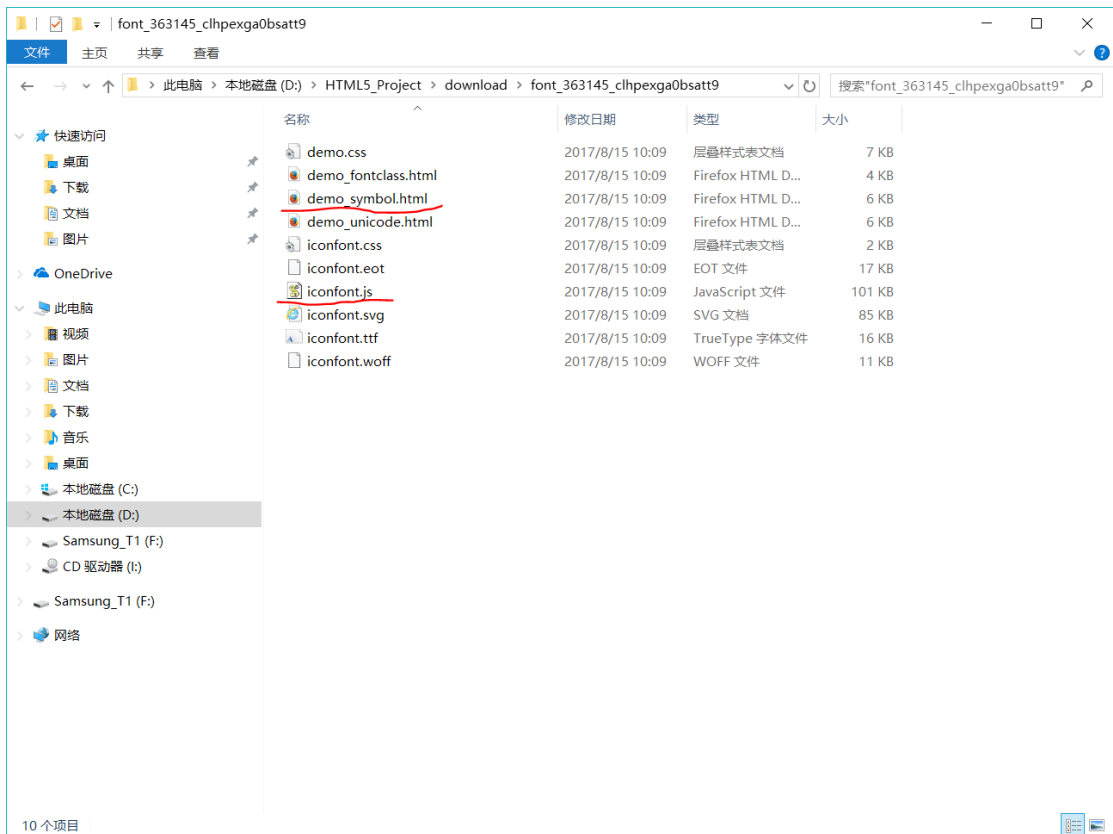




10.1.5 下载项目。



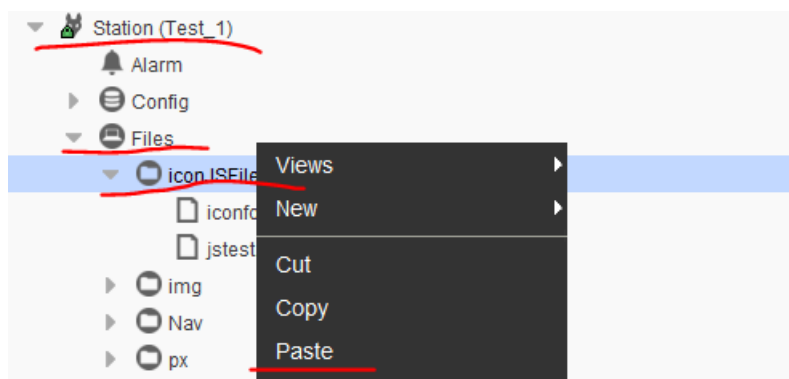
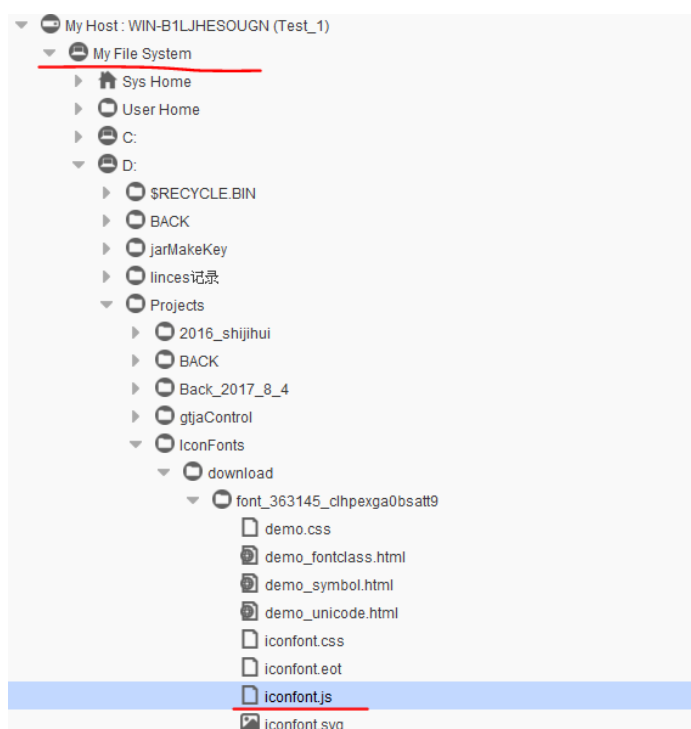
10.1.6 解压下载好的项目，浏览其中的 demo_symbol.html 网页，建议将各个图标下的文字标识都拷贝出来放到一个文本文件中，原因是因为有些图标的名称太长导致网页上两个图标的文字会重叠在一起，放到文本文件里就分开了。



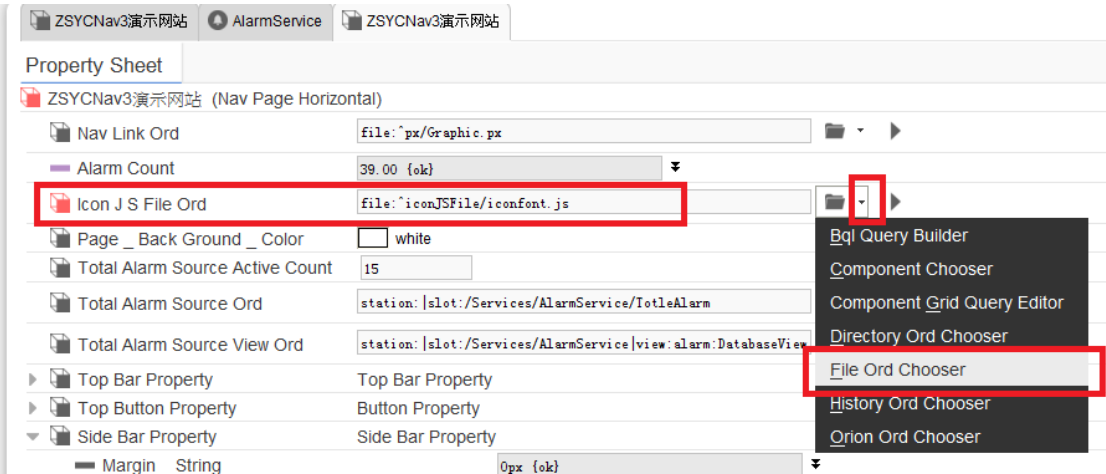


10.2 引入图标

10.2.1 将 iconfont.js 拷贝至 station 的 files 下。

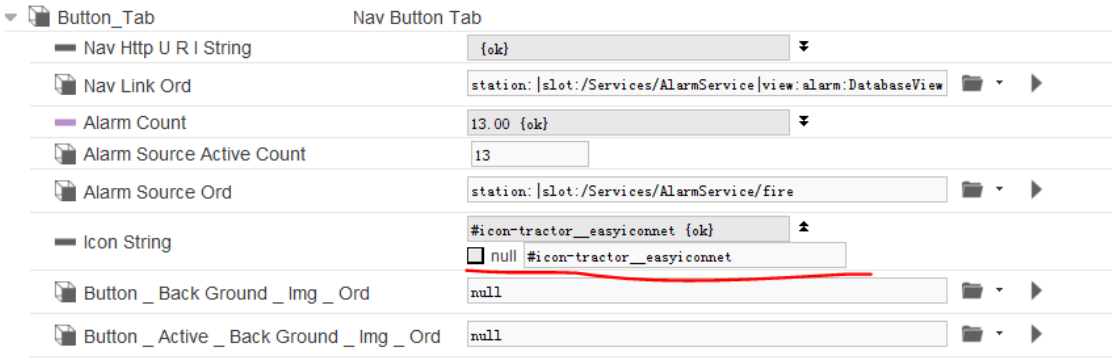


10.2.2 设置 Page_Horizontal 的 iconjsfileord 属性，引入 iconfont.js 文件。

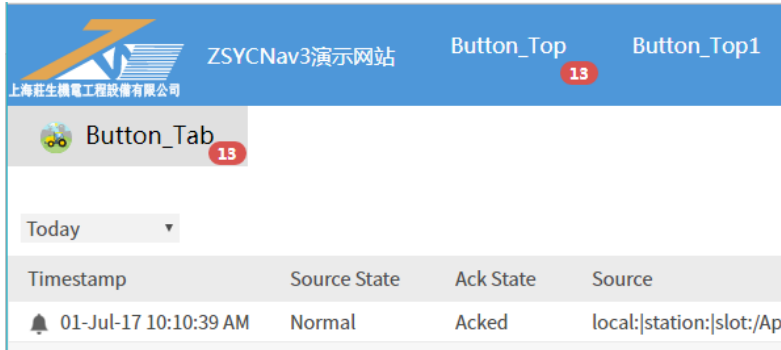


10.2.3 设置对应的 IconHeight 属性，该属性设置图标大小。

10.2.4 设置需要图标的 Button_Sidebar 或 Button_Top 或 Button_Tab 的 iconString 属性，填入需要的图标的具体代号，即刚才拷入文本文件中的相应图标的以#号开头的字符串，例如本例中#icon-iconfontanquanfanghu 对应的是个消防灭火器图标。



10.2.5 结果如下。

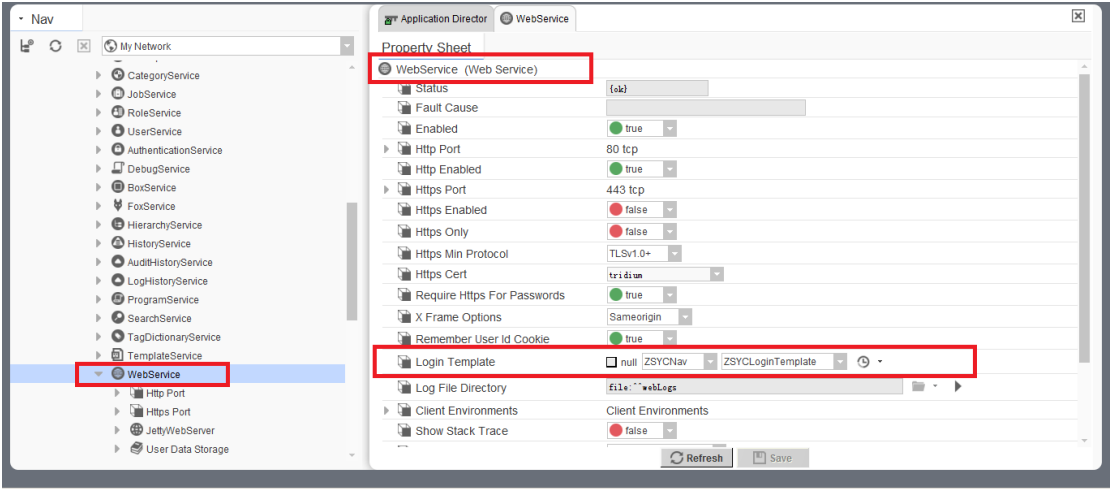


需要注意的是，该方法既可以引入纯色图标，也可以引入彩色图标，纯色图标将和按钮文字一起变色，而彩色图标则只有透明色部分会随文字变色。

11 附录 5：自定义登陆页面

11.1 设置登录模板

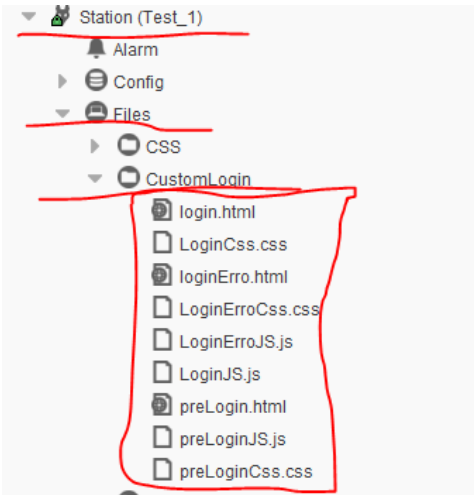
在站点的 webService 里设置该站点的登陆模板。



11.2 设置登陆首页相关文件夹

11.2.1 在站点的 files 下新建一个叫 CustomLogin 的文件夹，注意文件夹名必须为 CustomLogin

11.2.2 将自定义文件放入该文件中，自定义文件必须为如下文件名的几个文件：



- preLogin.html : 登陆时的第一个页面, 输入用户名。
- preLoginCss.css : preLogin.html 网页的 Css 文件, 内容会被插入至 preLogin.html 的 ID 为 preLginCSS 的 Dom 块中, 可以为空。
- preLoginJS.js : preLogin.html 网页的 javascript 文件, 内容会被插入至 preLogin.html 的 ID 为 preLginJS 的 Dom 块中, 可以为空。
- login.html : 登陆时的第二个页面, 输入密码。
- LoginCss.css : login.html 网页的 Css 文件, 内容会被插入至 login.html 的 ID 为 LginCSS 的 Dom 块中, 可以为空。
- LoginJS.js : login.html 网页的 javascript 文件, 内容会被插入至 login.html 的 ID 为 LginJS 的 Dom 块中
- loginErro.html : 如登陆错误, 则显示该网页, 如果是用户名没输入对, 请按切换用户链接回到 preLogin 界面重新输入用户名。
- LoginErroCss.css : loginErro.html 网页的 Css 文件, 内容会被插入至 loginErro.html 的 ID 为 loginErro CSS 的 Dom 块中
- LoginErroJS.js : loginErro.html 网页的 JavaScript 文件, 内容会被插入至 loginErro.html 的 ID 为 loginErroJS 的 Dom 块中

11.2.3 这三个网页最少必须包括如下代码（可添加 DOM 以及各种格式和属性，可

以修改如下示例中中文写明的可修改的文字的地方，但如下代码不可删减）：

11.2.3.1 预登陆网页 preLogin.html

```
<!DOCTYPE html>
<html>
<head>
  <meta name="viewport" content="width=device-width initial-scale=1.0
maximum-scale=1.0 target-densityDpi=medium-dpi">
  <title>登陆标题请在此修改</title>
  <script type="text/javascript" src="login/loginN4.js"></script>
  <style type="text/css" id="preLoginCSS">$cssContent</style>
  <script type="text/javascript" id="preLoginJS">$jsContent</script>
</head>
<body onload="checkFail()">
  <script type="text/javascript">
    if ('ontouchstart' in window) {
      document.body.className += ' touch-enabled';
    }
  </script>
  <div>
    <noscript>JavaScript must be enabled to login</noscript>
  </div>
  <script type="text/javascript" src="login/digestLoginN4.js"></script>
  <script type="text/javascript" src="login/core/auth.min.js"></script>
  <form method='POST' action='prelogin'>
    <input id="username" class="login-input" type="text"
name="j_username" autofocus="">
    <input type='submit' name='submit' value="登陆按钮文字请在此修改"
onclick="return doLogin();" />
  </form>
</body>
</html>
```

● 登陆网页 login.html

```

<!DOCTYPE html>
<html>
<head>
  <meta name="viewport" content="width=device-width initial-scale=1.0
maximum-scale=1.0 target-densityDpi=medium-dpi">
  <title>登陆标题文字请在此修改</title>
  <script type="text/javascript" src="login/loginN4.js"></script>
  <style type="text/css" id="loginCSS">$cssContent</style>
  <script type="text/javascript" id="loginJS">$jsContent</script>
</head>
<body onload="checkFail()">
  <script type='text/javascript'>
    if ('ontouchstart' in window) {
      document.body.className += ' touch-enabled';
    }
  </script>
  <div>
    <noscript>JavaScript must be enabled to login</noscript>
  </div>
  <script type="text/javascript" src="login/digestLoginN4.js"></script>
  <script type="text/javascript" src="login/core/auth.min.js"></script>
  <form method='POST' action='j_security_check'>
    <input id="username" class="login-input" type="text"
name="j_username" value="$initUsername" readonly disabled>
    <input id="password" class="login-input" type="password"
name="j_password" autocomplete="off" autofocus="">
    <input type='submit' name='submit' value="登陆按钮文字请在此修
改" onclick="return doLogin();" />
  </form>
  <div class='login-change-user'>
    <a href="/prelogin?clear=true">更换用户文字请在此修改</a>
  </div>
</body>
</html>

```

- 登陆错误网页 loginErro.heml

```

<!DOCTYPE html>
<html>
<head>
  <meta name="viewport" content="width=device-width initial-
scale=1.0 maximum-scale=1.0 target-densityDpi=medium-dpi">
  <title>登陆标题文字请在此修改</title>
  <script type="text/javascript" src="login/loginN4.js"></script>
  <style type="text/css" id="loginErroCSS">$cssContent</style>
  <script type="text/javascript" id="loginErroJS">$jsContent</script>
</head>
<body onload="checkFail()">
<script type='text/javascript'>
  if ('ontouchstart' in window) {
    document.body.className += ' touch-enabled';
  }
</script>
<div>
  <noscript>JavaScript must be enabled to login</noscript>
</div>
<div id="login-failed">
  登陆错误文字请在此修改
</div>
<script type="text/javascript" src="login/digestLoginN4.js"></script>
<script type="text/javascript" src="login/core/auth.min.js"></script>
<form method='POST' action='j_security_check'>
  <input id="username" class="login-input" type="text"
name="j_username" value="$initUsername" readonly disabled>
  <input id="password" class="login-input" type="password"
name="j_password" autocomplete="off" autofocus="">
  <input type='submit' name='submit' value="登陆按钮文字请在此修
改" onclick="return doLogin();" />
</form>
<div class='login-change-user'>
  <a href="/prelogin?clear=true">更换用户链接文字请在此修改
</a>
</div>
</body>
</html>

```

11.3 关于网页定制的相关说明

11.3.1 必须使用预登陆网页先登陆用户名，也必须使用“更换用户”的链接才能回到预登陆页面，更换用户必须使用预登陆网页，这个是 Niagara4 的新机制，没得商量。

11.3.2 登陆网页无法引入站点内 files 下的动态文件资源，这是 Niagara4 里的安全机制规定的，这意味着您或者使用 Internet 网上的线上资源作为图片来源，或者以 Data URI scheme 形式直接将图片转换成的 BASE64 字符串写入 HTML 网页中或 CSS 网页中。

11.3.3 将图片转换为 BASE64 字符串的方法如下：

11.3.3.1 进入 <http://www.css-js.com/tools/base64.html> 网站

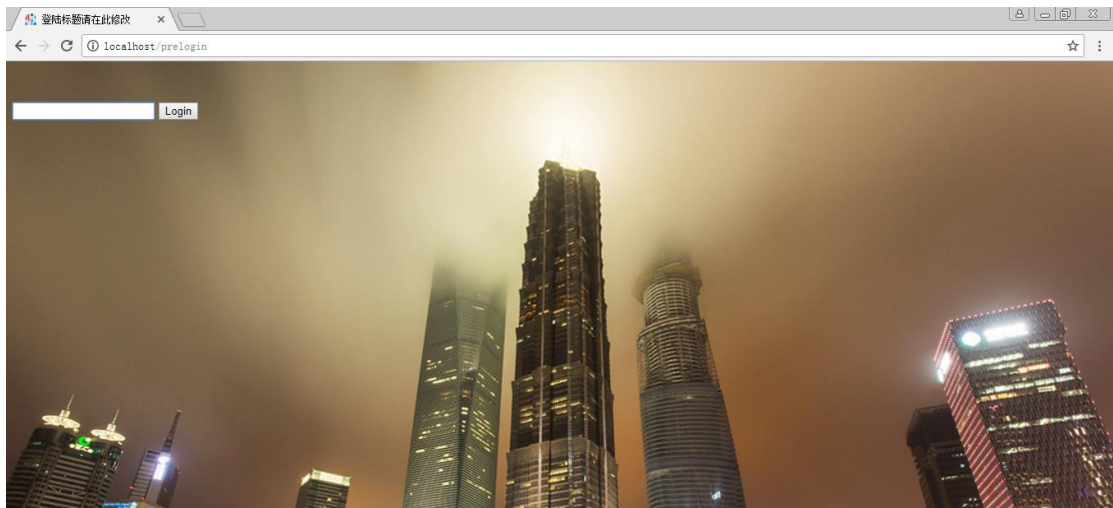
11.3.3.2 将转换得到的代码拷贝到需要使用的 CSS 文件或 HTML 文件的相应位置中



11.3.4 例如，我可以在 preLoginCss.css 里添加如下代码

```
body {
    padding-top: 40px;
    padding-bottom: 40px;
    background-repeat: no-repeat;
    background-size: 100%, 100%;
    /*background-position: center;*/
    background-image: url("data:image/jpg;base64,/9j/4AAQSkZJRg...
上例所得该图片的BASE64代码...");
}
```

11.3.5 则该站点的预登陆页面如下所示



更多如何美化网页的知识请参考 HTML 以及 CSS 以及 JavaScript 等网页相关内容，或者干脆外包找专门的网页制作人员来完成，只要网页内容包含了上文所说的那些必要的 DOM，其他怎么改本文就不赘述了。当然，也可参考本人提供的“自定义模板登陆网页修改示范案例”，该案例将上文中的几个必备 Form 和 Input 部件进行了组织，并添加了一些说明性文字，添加 CSS 格式，并在 CSS 文件中添加了背景图，具体可见案例中文件。