

OA-Bootstrap

新一代的OA将采用基于Bootstrap的前端开发框架，Bootstrap是基于Node.js模式开发最优秀前端框架之一。

简介

Bootstrap提供了基于jekyll的静态服务器的演示，jekyll相当于apache一样的提供WEB服务的本地或者远程服务器，在我们定制版OA产品中，并不需要此功能。本产品基于目前的Bootstrap 3.3.6版本为基础，对编译环境配置进行了定制化改造，并对其JS、CSS进行适当改造以兼容IE8浏览器。

优点

Bootstrap主要有三大特性，更多的制定了一套严格前台、样式及JS代码的规范，并从源头节省了前端开发的时间成本，详细文档请前往Bootstrap官网进一步了解。基主要特性表现以下：

基于预处理脚本的方式

虽然我们可以直接使用 Bootstrap 提供的 CSS 样式表，但是默认的UI样式并不能满足产品的需求，而Bootstrap样式源码是基于最流行的 CSS 预处理脚本 - Less 和 Sass 开发的。目前我们的前端开发模式采用的是Less预编辑处理方式，简言之就是像写程序一样的去写样式，避免逐行编写几千几万行的CSS样式代码。

一个框架、多种设备

在理论的全新架构下，我们可以在产品设计阶段斟酌设备适配，例如基于手机端设备向上兼容所有大屏幕设备响应，并且基于一套前端代码的表现有效的适配手机、平板、PC设备，旨在从前端阶段减少开发成本。

丰富的组件插件

我们不必再为产品功能需求而烦恼，不再为表单、控件件的UI不统一而纠缠，对于前端开发和程序开发人员来说，逻辑和交互的剥离带来的是更多的便捷开发，Bootstrap内置了大量的常用HTML 元素、CSS 组件、JS插件，满足了产品的UI基本需求。重要的是Bootstrap是基于jQuery的二次开发，它完美的兼容jQuery。

外框架

换肤

三种样式，18种颜色。

加载样式表：

```
<!-- 默认 -->
<link rel="stylesheet" href="dist/css/template_default/themes/2016/template.theme.before.css" />
<link rel="stylesheet" href="dist/css/template_default/themes/2016/template.theme.after.css" />

<!-- 定宽 -->
<link rel="stylesheet" href="dist/css/template_default/themes/2016/template.theme.width.css" />
```

```
<!-- 纯色 -->
<link rel="stylesheet" href="dist/css/template_default/themes/2016/template.theme.newpure.css" />
```

ie8下bug: 默认情况下, 后五种样式加载不了。分开写解决此bug

使用方法:

```
<html lang="zh-cn" class="wh-gray-bg theme-gray theme-width-gray size-big">
```

表示 调用 `gray` `width` 皮肤。

详情页栅格修改

弹出详情页的栅格 与 bootstrap 原生的栅格有所不同。

```
.col-wh-1,
.col-wh-2,
.col-wh-3,
.col-wh-4,
.col-wh-5,
.col-wh-6,
.col-wh-7,
.col-wh-8,
.col-wh-9,
.col-wh-10,
.col-wh-11,
.col-wh-12,
```

用法:

```
<div class="row">
  <div class="col-wh-9">
    ...
  </div>
  <div class="col-wh-3">
    ...
  </div>
</div>
```

弹出框

使用layer插件。文档参见: <http://layer.layui.com/>

整合到产品中的使用方法:

1. 弹出页包含在父页面中:

```
<div id="detail-selectuser" class="layer-modal modal-selectuser-set">
  ...
</div>
```

`id`: modal 的唯一标示;

`layer-modal` : 必要的类名, 初始状态下, 样式为 `display: none;` ;

`modal-selectuser-set` : 用于该 `modal` 的样式;

调用js:

```
$('#detail-selectuser').on('click', function(argument) {
    layer.open({
        type: 1,
        title: '参会人号码',
        shadeClose: false,
        shade: 0.8,
        area: ['480px', 'auto'],
        content: $('#detail-selectuser')
    });
})
```

参数配置参见layer说明文档。

2. 弹出页为iframe , 可在多个页面调用同一个弹出框

```
<div id="meeting-start" class="meeting-start-set layui-iframe-set">
    ...
</div>
```

`id` : modal 的唯一标示;

`modal-selectuser-set` : 用于该 `modal` 的样式;

`layui-iframe-set` 表示弹出框为iframe 的形式。必要类名。

调用js:

```
$(".meeting-handle").on("click", function(argument) {
    layer.open({
        type: 2,
        title: '视频会议',
        shadeClose: false,
        shade: 0.8,
        area: ['620px', '350px'],
        content: "include/meeting-iframe-test.html"
    });
});
```

参数配置参见layer说明文档。

需要注意的是, 弹出框为iframe中的样式需要重新加载。

下拉框

下拉框我们可以使用bs 原生的下拉菜单, 在设计不能满足的情况下, 可以进行定制。一般情况下采用添加 `class` 类名的方式去操作动画效果。

样式源码编译目录

```

source/
<!--源码所在目录-->
├─ bspjs/ <!--源码JS所在目录-->
│   └─ bootstrap/ <!--bootstrap源码JS所在目录-->
│       ├── affix.js
│       ├── alert.js
│       ├── tap.js
│       ├── dropdown.js
│       └─ .....
│   └─ bootstrap-wh/ <!--修改版bootstrap源码JS所在目录-->
│       └─ dropdown-wh.js <!--这是基于原版的定制版-->
├─ extend/ <!--这是按照bootstrap方式封装的扩展插件-->
│   ├── autogrow-textarea.js <!--textarea自适应: https://github.com/jevin/Autogrow-Textarea-->
│   ├── chosen.js <!--多功能select重绘: https://github.com/harvesthq/chosen-->
│   ├── laydate.js <!--lay日期控件: https://github.com/sentsin/laydate -->
│   ├── layer.js <!--lay弹层控件: https://github.com/sentsin/layer-->
│   ├── pagination.js <!--分页控件: https://github.com/mricle/Mricode.Pagination-->
│   ├── select.js <!--轻量级select重绘http://silviomoreto.github.io/bootstrap-select-->
│   └─ ...
├─ custom/ <!--这是按照bootstrap方式封装的扩展插件-->
│   ├── active-status.js <!--点击增加过渡类选择器, 查看USEAGE.md阅读使用说明-->
│   ├── scroll-top.js <!--返回顶部, 查看USEAGE.md阅读使用说明-->
│   └─ device.js <!--识别设备userAgent, 可根据性能差异将其移除-->
├─ template/ <!--静态资源所在目录-->
│   ├── fonts/ <!--矢量图标文件所在目录-->
│   ├── images/ <!--图片资源所在目录-->
│   └─ less/ <!--图片less样式所在目录-->
│       ├── bootstrap/ <!--Bootstrap原生less文件-->
│       │   └─ ...
│       ├── fontless/ <!--矢量图标样式所在目录-->
│       ├── global/ <!--公共样式-->
│       │   ├── template.global.less <!--每个编译出的less都需要它-->
│       │   ├── template.keyframe.less <!--11.4的时候统一规划的动画less库-->
│       │   └─ template.ztree.less <!--ztree主样式库-->
│       ├── media/ <!--公共组件所在的样式-->
│       │   ├── template.attachment.less <!--附件列表样式-->
│       │   ├── template.btn.less <!--按钮的样式-->
│       │   ├── template.modal.less <!--弹框类样式-->
│       │   ├── template.chosen.less <!--对应多功能select重绘样式-->
│       │   ├── template.colorpicker.less <!--对应控制面板拾色器样式-->
│       │   ├── template.gotop.less <!--对应返回顶部样式-->
│       │   ├── template.laydate.less <!--对应日期控件样式-->
│       │   ├── template.pagination.less <!--对应分页控件样式-->
│       │   ├── template.select.less <!--对应轻量级select重绘样式-->
│       │   ├── template.laydate.less <!--对应日期控件样式-->
│       │   └─ template.skincolor.less <!--对应公共换肤样式-->
│       ├── others/ <!--原有的ie8、ios补丁样式, 计划移除-->
│       ├── size/ <!--管理继承11.4的门户和portal的字号大小样式-->
│       │   ├── template.portal.big.less <!--portal中14px字号变量库-->
│       │   ├── template.portal.small.less <!--portal中12px字号变量库-->
│       │   ├── template.style.big.less <!--门户中14px字号变量库-->
│       │   ├── template.style.small.less <!--门户中12px字号变量库-->
│       │   └─ template.style.big16.less <!--门户中16px字号变量库-->
│       └─ style/ <!--主样式-->
│           ├── template.detail.less <!--详情页的样式库-->
│           ├── template.frame.less <!--门户的样式库-->
│           ├── template.list.less <!--列表样式库-->
│           └─ template.table.less <!--表格样式库-->

```

```

└─ svg/ <!--计划未来对一些svg素材进行管理-->
└─ system/ <!--规划的系统样式-->
    └─ template.login.ie8.less <!--登录页的ie8补丁样式，未来仅在登录样式中展现-->
    └─ template.login.less <!--登录页样式库-->
    └─ template.system.less <!--系统样式库-->
    └─ template.login_12oa.less <!--登录页样式表12.0+版本更新-->
└─ themes/ <!--管理换肤主题样式-->
    └─ 2015/ <!--特定文件夹-->
        └─ color/ <!--管理颜色的样式库-->
            └─ var-default <!--默认皮肤类型对应的颜色变量库-->
            └─ var-global <!--所有皮肤的公共颜色变量库-->
            └─ var-line <!--线性皮肤类型对应的颜色变量库-->
            └─ var-pure <!--纯色皮肤类型对应的颜色变量库-->
        └─ template.theme.detail.less <!--详情页的颜色样式-->
        └─ template.theme.less <!--主样式的颜色样式-->
        └─ template.theme.line.less <!--线性皮肤的补充样式-->
        └─ template.theme.newpure.less <!--标准皮肤样式-->
        └─ template.theme.media.less <!--公共组件的颜色样式-->
        └─ template.theme.width.less <!--定宽版式基础样式-->
└─ template.bootstrap.less <!--bootstrap主样式-->
└─ template.bootstrap.theme.less <!--bootstrap自定义变量库-->
└─ template.bootstrap.wh.less <!--bootstrap-wh版重置样式-->
└─ <!-- ***** -->
└─ <!--下述样式均依赖global/template.global.less-->
└─ <!-- ***** -->
└─ template.detail.less <!--详情页编译样式，依赖、style/template.detail.less-->
└─ template.detail.size.less <!--详情页的字号大小编译样式-->
└─ template.fa.less <!--字体图标的编译样式，依赖于 fontless/下的所有样式库-->
└─ template.lists.less <!--列表页编译样式，依赖style/template.table.less和template.list.less-->
└─ template.media.less <!--公共组件编译样式，依赖 media/下所有样式库-->
└─ template.page.less <!--11.4的portal编译补充样式，包含知识文库、知道-->
└─ template.portal.less <!--11.4的portal编译主样式-->
└─ template.portal.size.less <!--Portal的字号大小编译样式，依赖 size/下所有样式库-->
└─ template.print.less <!--打印页编译样式-->
└─ template.reset.less <!--公共重置编译样式-->
└─ template.style.less <!--门户编译主样式-->
└─ template.style.size.less <!--门户编译字号大小样式-->

```