### 项目前端文件结构规范

* 该文档仅针对项目的前端js、jsp文件
* 目的：

统一、规范文件项目前端文件结构及依赖关系。

重点阅读“**基本文件结构**”，“**项目依赖、引用规则**”，“**版本管理**”

* 文件/文件夹命名规范
  + 各单词小写，用短横线(-)分隔
  + 单词保持短小，精炼，尽量使用常用缩写。
* **基本文件结构：**
* js结构

webapp

static

js

common --- 项目内公共文件夹

service -- 服务文件夹

mod1 -- 服务按模块划分的文件夹

DemoService.js -- 服务文件

......

mod1 -- 功能按模块划分的文件夹

fun1 -- 功能界面文件夹（单页面为例）

view.js -- 功能界面入口文件

biz.js -- 功能界面业务逻辑文件

components.js -- 功能界面控件配置文件

func2 -- 功能界面文件夹（内部有 外链、弹出 等页面为例）

main -- (主界面文件夹)

view.js -- 功能界面入口文件

biz.js -- 功能界面业务逻辑文件

components.js -- 功能界面控件配置文件

win (内部打开、外链窗口界面文件夹)

view.js -- 界面入口文件

biz.js -- 界面业务逻辑文件

components.js -- 界面控件配置文件

......

......

......

require-config.js -- 项目require配置，项目内部公共模块定义的地方

* jsp结构

webapp

WEB-INF

views

mod1 -- 功能按模块划分的文件夹

fun1.jsp -- 功能界面jsp（单页面为例）

fun2 -- 功能界面文件夹（内部有 外链、弹出 等页面为例）

main.jsp -- 主界面jsp

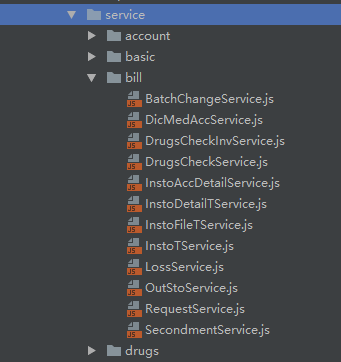
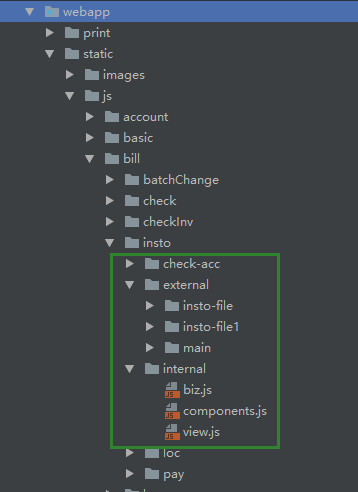
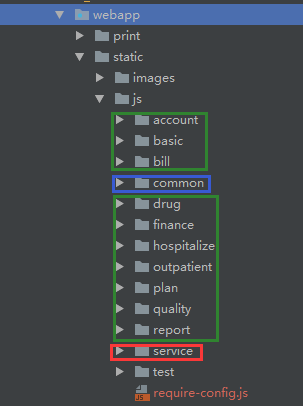
win.jsp -- 打开/外链界面jsp

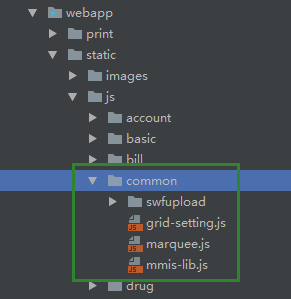
......

......

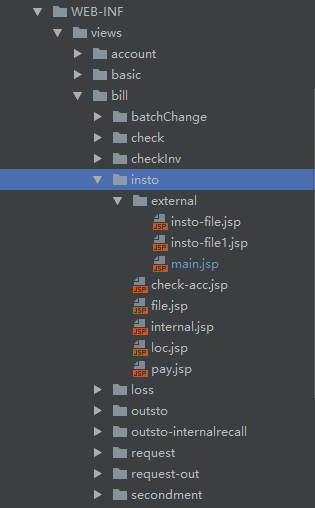
......

* js 结构实例截图

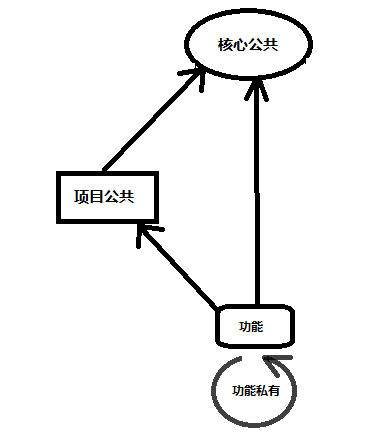




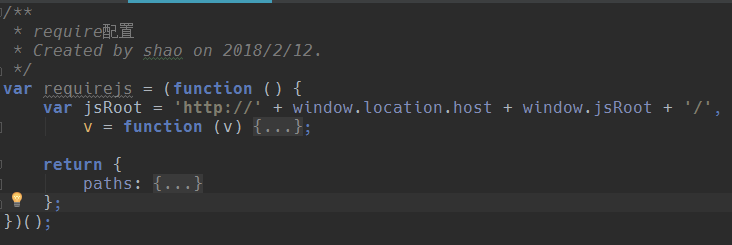
* jsp结构实例截图



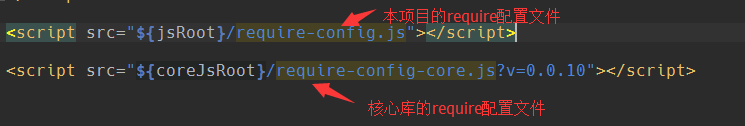
* **项目依赖、引用规则**
* 各项目（包括核心项目）整体的依赖关系用如下图表示：



* 项目内部公共模块定义在各自项目的require-config.js 中，如：



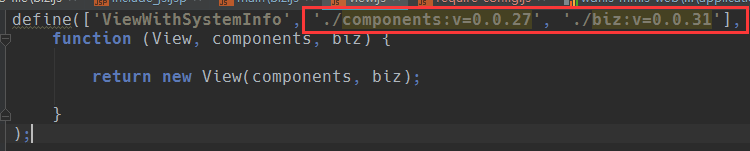
* 项目需引入核心项目的公共require配置文件，这样当前项目才使用核心库，如：



* （可选）项目需定义全局变量window.jsRoot，用于指定本项目的js文件夹根路径，

如：window.jsRoot = window.ctx + ‘/static/js’

* 功能私有的模块放到各自功能文件夹下面，不在reqiure-config.js 里配置，以相对路径的方式引入，如：



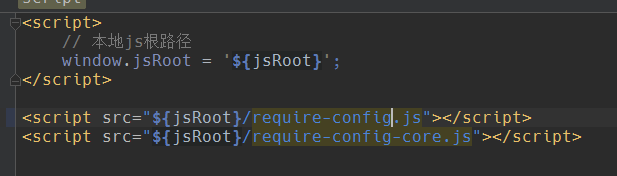
注意：

* 先前核心项目只引入一个配置文件，这个配置文件包含了项目本身及对其他项目开放的公共模块，为了不相互影响，现在把这两部分分开，分别是：

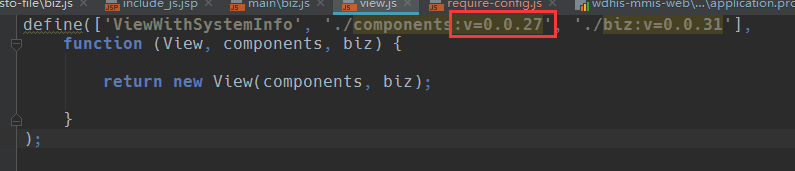
require-config.js：用于定义核心项目本身的配置

require-config-core: 用于定义所有项目的迫人配置，以及每个项目都会引用的核心模块。

* 项目自身的配置文件的变量名必须是：requirejs；
* 项目自身的配置文件必须在核心的配置文件之前引入，因为核心的配置文件会自动合并项目自身的配置（指定相同配置的情况下，核心的配置会被覆盖）；
* 核心的配置文件包含了核心库的配置，因此所有项目都需要引用，包括核心项目本身，下面是核心项目本身的引入方式：



* **版本管理**
* 功能私有的模块通过如下方式指定：



在引用的模块路径后面加上：“:v=具体版本号”，当文件内容改变的时候，引用的地方变更相应版本号

* 公共模块的版本通过在reqiure-config.js 中变更相应版本号，下面是参考做法：

