**站点目录结构**

业务功能模块化，针对常见的业务模型，抽象出以下层级关系：

* 站点(site)：一般指能独立提供服务，具有单独二级域名的产品线。如旅游产品线或者特大站点的子站点（lv.baidu.com）。
* 模块(module)：具有较清晰业务逻辑关系的功能业务集合，一般也叫系统子模块，多个子系统构成一个站点。
* 页面(page): 具有独立URL的输出内容，多个页面一般可组成子系统。
* 组件(widget)：能独立提供功能且能够复用的独立资源，它可以是独立的JS、CSS或者是由JS、CSS和页面组成的页面碎片。
* 静态资源(static)：非组件资源目录，包括模板页面引用的静态资源和其他静态资源（favicon,crossdomain.xml等）。
* 插件(plugin): 模板插件目录(可选，对于特殊需要的产品线，可以独立维护)。
* 测试数据(test): 页面对应的测试数据目录。

FIS规范定义了两类模块： **common模块**与 **业务模块**。

* common模块: 为其他业务模块提供规范、资源复用的通用模块。
* 业务模块: 根据业务、URI等将站点进行划分的子系统站点模块。

站点整体目录结构示意：

|---site

| |---common //通用子系统

| | |---config //smarty配置文件

| | |---page //模板页面文件目录，也包含用于继承的模板页面

| | └── layout.tpl

| | |---widget //组件的资源目录，包括模板组件,JS组件,CSS组件等

| | | └── menu //widget模板组件

| | | | └── menu.tpl

| | | | └── menu.js

| | | | └── menu.css

| | | └── ui //UI组件

| | | └── dialog //JS组件

| | | | └──dialog.js

| | | | └──dialog.css

| | | └── reset //CSS组件

| | | └── reset.css

| | |---static //非组件静态资源目录，包括模板页面引用的静态资源和其他静态资源

| | |---plugin //模板插件目录**(**可选，对于特殊需要的产品线，可以独立维护**)**

| | |---fis-conf.js //fis配置文件

| |---module1 //module1子系统

| | |---test

| | |---config

| | |---page

| | └── index.tpl

| | |---widget

| | |---static

| | | └── index //index.tpl模板对应的静态资源

| | | └── index.js

| | | └── index.css

| | |---fis-conf.js //fis配置文件