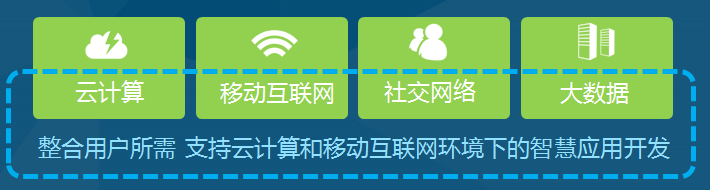
# Smart Cloud 开发指南

## 一、概述

#### 1.1什么是智慧云平台



（图 1-1 智慧云平台）

如图1-1所示：智慧云平台支持基于云计算、移动互联网、社交网络和大数据环境下的智慧应用开发，提供web或移动app应用所需的应用框架、常见界面组件、应用组件和数据分析等功能。

#### 1.2智慧云平台的技术架构



（图 1-2 平台架构）

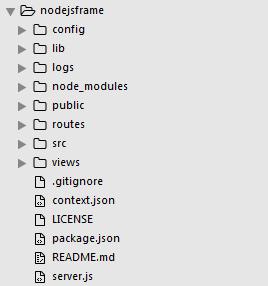
如图1-2所示：智慧云平台架构一共分为五层，分别是界面组建、分析功能、应用组件、应用框架以及数据库，其中界面组件是应用UI端，其它四层都是应用服务端。

本项目以nodejs为技术平台，提供应用服务，可利用浏览器或移动客户端等多种方式构建应用界面，方便用户使用，并提供对bootstrap、jquery等常用组件的支持，其访问过程如图1-3所示：



（图 1-3 访问过程）

## 二、工程目录结构



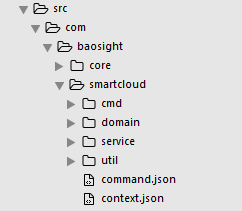
（图 2-1 总体目录结构）

如上图2-1所示：Smart Cloud系统主要具有如下目录文件：

* config目录：主要用来存放系统相关配置文件，包括系统初始化配置文件（config.json）、应用服务配置文件（server.json）以及日志配置文件（log4js.json）等。
* node\_modules目录：存放所有的项目依赖库；
* public目录：用来存放网站所有的静态资源和公共资源，包含images、javascripts、css、前端配置文件等；
* routes目录：用来存放地址访问控制文件；
* src目录：用来存放应用服务，所有的服务必须在src目录下；
* views目录：视图文件目录，用来存放模板引擎文件；
* context.json:基于bearcat框架的配置文件；
* server.js文件：项目入口文件，程序就是从这里开始运行；
* package.json文件：用来声明项目中使用的模块及相关信息，主要方便在新环境部署时，直接使用npm install进行项目的发布安装。

## 三、应用服务端

#### 3.1 目录结构



（图 3-1 应用服务端目录结构）

如上图3-1所示，smartcloud是一个应用服务：

* 为区分不同的应用服务，在src目录下应建立唯一地址（规范：公司域名+应用服务名，形如：com/baosight/employeemanage，其中com/baosight为公司的域名，smartcloud为应用服务名）；
* 在应用服务下，必须具有command.json以及context.json配置文件，分别是后台地址访问配置文件以及bearcat配置文件；
* 在应用服务下，应该具有cmd、domain以及service文件夹。cmd文件夹存放接口文件，即后台服务入口文件，domain文件夹用来存放数据模型，service文件夹用来存放业务处理文件；

#### 3.2 调用过程

应用服务端的调用过程如下图3-2所示：



（图3-2 应用服务端调用过程）

其详细过程如下所示：

1. http Server通过route请求数据，在route中匹配进行路由控制；
2. 在route中根据配置文件对url地址进行解析，解析过程如下所示：



（图3-3 路由解析）

1. route在url解析完成后，调用相应的command，处理页面请求中的逻辑；
2. command调用相应的业务服务处理业务逻辑；

4.1、command也可直接通过数据模型获取数据；

1. 业务处理通过数据模型获取数据；
2. command将服务处理返回的数据，据填充到ejs模板文件；
3. command将数据返回到route；
4. route将数据返回到http Server。

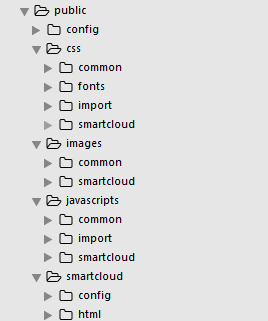
#### 3.3 对象容器

应用服务端的对象管理通过bearcat框架中的轻量级容器实现。bearcat是一个基于POJOs进行开发的应用层框架,提供了一个轻量级的容器来编写简单，可维护的[nodejs](http://nodejs.org/). Bearcat 提供了一个基础的底层来管理应用逻辑对象，使得开发者可以将精力放在应用层的逻辑编写上。核心容器包含核心，Beans，以及Context模块. 核心和Beans模块提供容器的基础部分，包含IoC容器和依赖注入. BeanFactory 是一个复杂的factory工厂模式实现. 它去除了手动去编写单例，允许你的实际程序逻辑从配置和依赖的管理中解耦。

bearcat框架地址：<https://github.com/bearcatnode/bearcat>

## 四、应用界面端

#### 4.1 目录结构



（图 4-1 应用界面端目录结构）

public文件夹下存在config、css、images、javascripts、smartcloud文件夹，其中config存放前端配置文件（包括：菜单、组件等配置文件）；css存放css样式文件（包括：common通用的自定义样式；import导入外部的样式，以及smartcloud（项目名）的自身样式）；同理images以及javascripts的结构与css结构类似；smartcloud表示一个项目名，此文件下存放两个文件夹，分别是config以及html，config 下存放了前端组件的相关配置信息，html存放了静态的html文件。

|  |
| --- |
| 项目下config中的前端配置信息如下所示：  {  "name":"智慧云平台",//项目名  "path":"smartcloud",//项目对应public下的路径  "code":"component\_smartcloudServer",//项目code 必须“component\_”+server "pCode":"component\_root",//父级目录，由于是前端组件 所以以component\_打头  "index":"user.list",//默认访问的页面  "modules":[  {"module":"user","caption":"用户管理","desc":"用户管理","js":"/javascripts/smartcloud/userController.js","pages":[  {"id":"list","caption":"用户列表","desc":"用户列表","url":"/html/user/userList.html","method":"init","components":[  {"component":"addUser","caption":"新增用户按钮","desc":"新增用户按钮","limit":false},//limit表示是否加入权限控制  {"component":"queryUser","caption":"查询用户按钮","desc":"查询用户按钮","limit":true}  ]},    ]}  ]  } |

|  |
| --- |
| 静态html页面注意点如下所示：   1. html的最顶层必须加入ng-page属性； 2. 所有的点击事件必须以如下命名：ng-click=”函数名” id=”函数名” |

#### 4.2 页面调用过程

页面调用过程如下图4-2所示：



（图4-2 页面调用过程）

其详细过程如下所示：

1. 浏览器输入url进行页面访问，如http://127.0.0.1:18080/#user.list；
2. App监听器中监听url地址变化，并对url地址进行解析，获取#后面的字符串（user.list）；
3. App实例读取AppConfig中的相关配置信息；
4. App实例利用第2步获取到的字符串与AppConfig中的配置信息pageId进行匹配，加载相匹配的templateUrl属性对应的html文件；
5. App实例显示加载的html文件；
6. App实例调用html对应的method属性的方法；
7. 方法至服务端获取数据；
8. 方法将获取到的数据填充至加载的html文件。