# 后台文档使用说明

## 搭建模板编译环境

1.1安装nodejs运行环境

参照：<http://jingyan.baidu.com/article/a948d6515d4c850a2dcd2e18.html>

* 1. 安装tomdjs

在cmd 窗口执行命令：npm install -g tmodjs

## 在项目中使用模板进行开发

1.在 web项目中的 src/main/resources/static/tpl/路径下面新建um/post目录

2.在post目录中新建index.html文件(注意:新建的html文件必须在tpl目录或它的子目录下面)

3.编写index.html文件，具体样式可以参照bootstrap.

也可以参照模板的demo

Demo的路径在 src/main/resources/static/demo目录

index.html源码

<div class="container">  
 *<!--页面导航插件 ，具体使用方式 请参照 ui-api.doc-->* {{include '../../public/breadcrumbs'}}  
  
 <div class="card">  
 <div class="card-header">  
 <h2>用户职位管理</h2>  
 <ul class="actions">  
 <li>  
 *<!--新增按钮 ，这个url的意思是，先请求后台路径 um/post/init 后台SpringMvc Controller init方法,将返回的json数据用js绑定到 um/post/save 模板上。具体使用方式,请参照 ui-api.doc-->* <a href="#um/post/init?tpl=um/post/save" class="btn-modal">  
 <i class="zmdi zmdi-plus"></i>  
 </a>  
 </li>  
 </ul>  
 </div>  
 *<!--表格上面的搜索框和按钮-->* <div class="bootgrid-header container-fluid">  
 *<!--搜索用的form表单 tpl=表示模板的路径 ，具体使用方式 请参照 ui-api.doc-->* <form id="frmSearch" action="#um/post/list?tpl=um/post/save" role="form">  
 <div class="row">  
 <div class="col-sm-3">  
 <div class="input-group">  
 <span class="input-group-addon"><i class="zmdi zmdi-search"></i></span>  
  
 <div class="fg-line">  
 *<!--这个是搜索的输入框，支持自动补全，需添加样式auto-complete 和data-url属性，这个地方输入一个返回json的地址 ，具体使用方式 请参照 ui-api.doc-->* <input type="text" class="form-control auto-complete" name="code"  
 data-url="\_data/typeahead/all.json" value="{{query.code}}" hidden="true"  
 placeholder="名称和编码进行搜索">  
 </div>  
 </div>  
 </div>  
 <div class="col-sm-2">  
 *<!--搜索按钮，通过配置文件读取的按钮的名称，主要用于国际化 ，具体使用方式 请参照 ui-api.doc-->* <button type="button" class="btn btn-primary btn-sm m-t-5 waves-effect btn-search">  
 {{lang.btn.search}}  
 </button>  
 </div>  
 </div>  
 </form>  
 </div>  
 <div class="table-responsive">  
 *<!--显示数据的表格-->* <table class="table table-striped table-hover table-ellipsis">  
 <thead>  
 <tr>  
 <th width="5%">#</th>  
 <th width="10%">职位名称</th>  
 <th width="10%">编号</th>  
 <th width="13%">备注</th>  
 <th width="15%">插入时间</th>  
 <th width="15%">更新时间</th>  
 <th width="\*">操作</th>  
 </tr>  
 </thead>  
 <tbody>  
 *<!--循环遍历list,返回的json格式请严格参照这个模块已开发的格式 ，具体使用方式 请参照 ui-api.doc-->* {{each list}}  
 <tr>  
 <td>{{$index+1}}</td>  
 <td title="{{$value.name}}">{{$value.name}}</td>  
 <td title="{{$value.code}}">  
 {{$value.code}}  
 </td>  
 <td title="{{$value.note}}">{{$value.note}}</td>  
 <td title="">{{$value.insertTime | dateFormat}}</td>  
 <td title="">{{$value.updateTime | dateFormat}}</td>  
 <td>  
 *<!--修改按钮，具体使用方式 请参照 ui-api.doc-->* <button type="button" class="btn btn-xs btn-default btn-icon btn-modal waves-effect waves-circle waves-float waves-effect waves-circle waves-float"  
 data-href="um/post/get/{{$value.id}}?tpl=um/post/save">  
 <span class="zmdi zmdi-edit"></span> </button>  
 *<!--<button type="button" class="btn btn-xs btn-default btn-icon btn-modal waves-effect waves-circle waves-float waves-effect waves-circle waves-float"  
 data-href="sm/role/menu/{{$value.id}}?tpl=sm/role/tree">  
 <span class="zmdi zmdi-account"></span> </button>-->  
 <!--删除按钮，具体使用方式 请参照 ui-api.doc-->* <button type="button" class="btn btn-xs btn-danger btn-icon btn-del"  
 data-url="um/post/del/{{$value.id}}">  
 <span class="zmdi zmdi-delete"></span> </button>  
 </td>  
 </tr>  
 *<!--循环结束-->* {{/each}}  
 </tbody>  
 </table>  
 </div>  
 *<!--分页插件注入，需要分页的话，原样写就行了-->* {{#pagination}}  
 </div>  
</div>

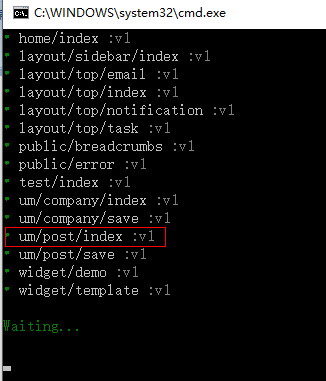
后台方法请自行看我写的模块

## 运行项目

1. 先编译模板html

运行 src/main/resources/static/tpl/ 目录下面的 build.bat

（注意：如果出现打开闪退的话，请重复步骤 1）



可以看到，我们写的html已经编译了，这个编译的窗口可以不用关，当我们修改了tpl目录下面的html文件后，会自动编译模板。

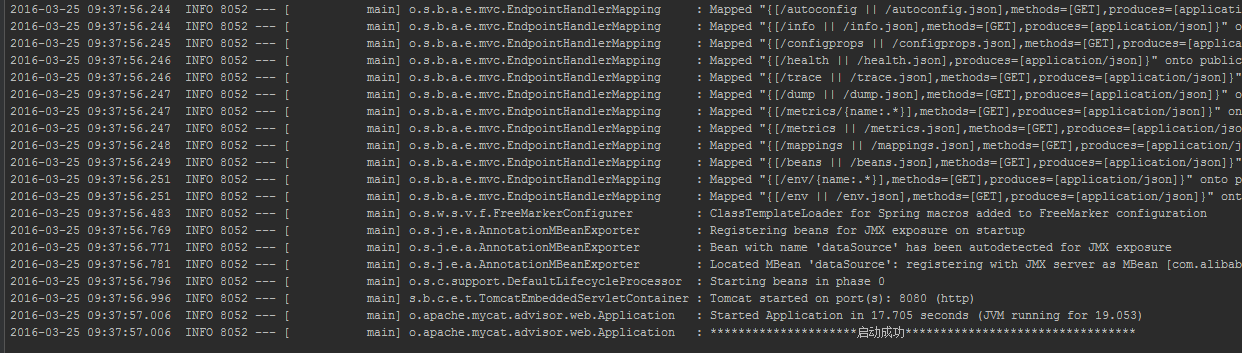
例如，我们在刚才写的index.html文件中加个空格



发现代码已经被编译了

然后运行web/src/main/java 下面的Application class

打印如下

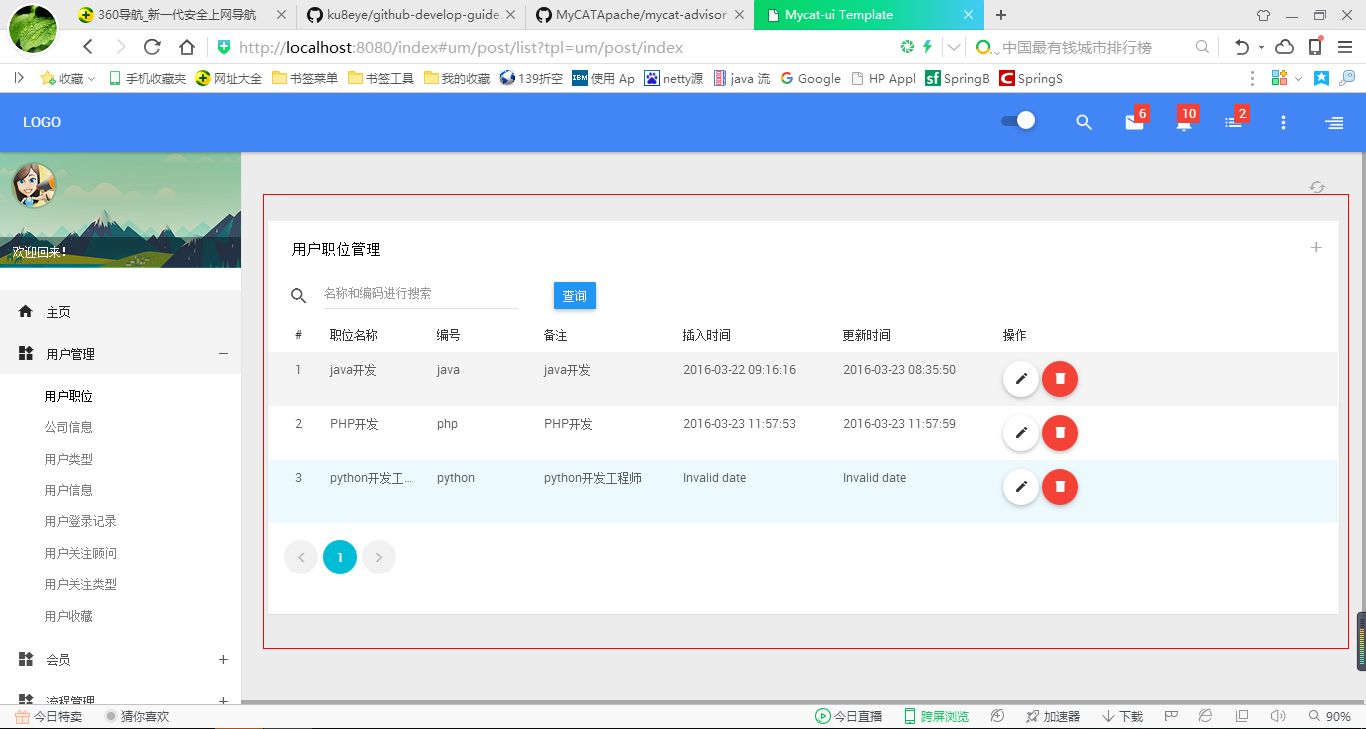


项目启动成功

打开浏览器输入：<http://localhost:8080/>

输入 admin/admin

进入系统，点击用户用户职位，我们刚才写的模块就好了



红框框选的部分就是刚才编写的html代码