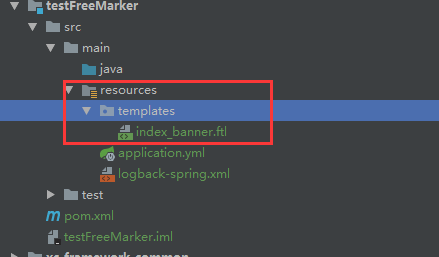
1：页面静态化，模板需要放在resources的templates目录下

没有该目录需要建立，因为在静态化时，会自动找该模板文件，当然如果你使用的是内存模板，那就不需要了



1： 页面静态化，在内存生成模板文件，直接页面静态化，不需要像一那样，需要先获取磁盘上的模板文件

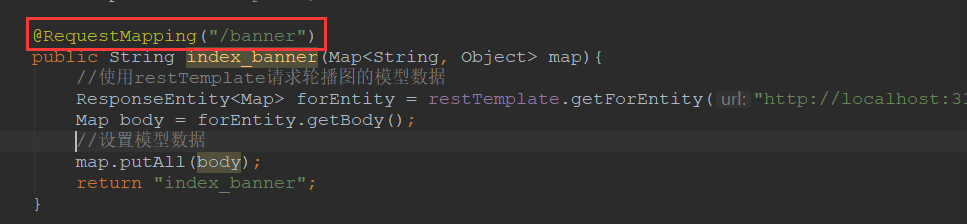
/\*使用在内存生成模板方式来执行页面静态化，不需要在磁盘生成一个模板，这是最常用的方法\*/  
/\*就是根据一个段字符串，在程序执行时，在内存先生成一个模板，然后填充数据模型，返回静态化的页面（也就是html代码）\*/  
private String generateHtml(String temPlateContent, Map model) {  
 try {  
 //生成配置类  
 Configuration configuration = new Configuration(Configuration.*getVersion*());  
 //模板加载器，根据temPlateContent生成模板文件  
 StringTemplateLoader stringTemplateLoader = new StringTemplateLoader();  
 stringTemplateLoader.putTemplate("template", temPlateContent);  
 configuration.setTemplateLoader(stringTemplateLoader);  
 //获取模板文件  
 Template template = configuration.getTemplate("template");  
 //填充数据模型，执行页面静态化  
 String html = FreeMarkerTemplateUtils.*processTemplateIntoString*(template, model);  
 return html;  
 } catch (Exception e) {  
 e.printStackTrace();  
 }  
 return null;  
  
}

这里会根据字符串temPlateContent生成对应的模板文件，其实temPlateContent 里面就是模板文件内容字符串, temPlateContent就是以下内容，注意加红色是Freemarker语法，

就是根据temPlateContent在内存生成了一个模板文件，然后获取该模板，填充数据，返回一个html内容，执行页面静态化完成

<!DOCTYPE html>  
<html lang="en">  
<head>  
 <meta charset="UTF-8">  
 <title>Title</title>  
 <link rel="stylesheet" href="http://localhost:80/plugins/normalize-css/normalize.css"/>  
 <link rel="stylesheet" href="http://localhost:80/plugins/bootstrap/dist/css/bootstrap.css"/>  
 <link rel="stylesheet" href="http://localhost:80/css/page-learing-index.css"/>  
 <link rel="stylesheet" href="http://localhost:80/css/page-header.css"/>  
</head>  
<body>  
<div class="banner-roll">  
 <div class="banner-item">  
 **<#if model??>  
 <#list model as item>  
 <div class="item" style="background-image: url(${item.value});"></div>  
 </#list>  
 </#if>** <#--<div class="item" style="background-image: url(http://localhost:80/img/widget-bannerB.jpg);"></div>-->  
 <#--<div class="item" style="background-image: url(http://localhost:80/img/widget-bannerA.jpg);"></div>-->  
 <#--<div class="item" style="background-image: url(http://localhost:80/widget-banner3.png);"></div>-->  
 <#--<div class="item" style="background-image: url(http://localhost:80/img/widget-bannerB.jpg);"></div>-->  
 <#--<div class="item" style="background-image: url(http://localhost:80/img/widget-bannerA.jpg);"></div>-->  
 <#--<div class="item" style="background-image: url(http://localhost:80/img/widget-banner3.png);"></div>-->  
 </div>  
 <div class="indicators"></div>  
</div>  
<script type="text/javascript" src="http://localhost:80/plugins/jquery/dist/jquery.js"></script>  
<script type="text/javascript" src="http://localhost:80/plugins/bootstrap/dist/js/bootstrap.js"></script>  
<script type="text/javascript">  
 **var** tg = $('.banner-item .item');  
 **var** num = 0;  
 **for** (i = 0; i < tg.length; i++) {  
 $('.indicators').append('<span></span>');  
 $('.indicators').find('span').eq(num).addClass('active');  
 }  
  
 **function** roll() {  
 tg.eq(num).animate({  
 'opacity': '1',  
 'z-index': num  
 }, 1000).siblings().animate({  
 'opacity': '0',  
 'z-index': 0  
 }, 1000);  
 $('.indicators').find('span').eq(num).addClass('active').siblings().removeClass('active');  
 **if** (num >= tg.length - 1) {  
 num = 0;  
 } **else** {  
 num++;  
 }  
 }  
  
 $('.indicators').find('span').click(**function** () {  
 num = $(**this**).index();  
 roll();  
 });  
 **var** timer = setInterval(roll, 3000);  
 $('.banner-item').mouseover(**function** () {  
 clearInterval(timer)  
 });  
 $('.banner-item').mouseout(**function** () {  
 timer = setInterval(roll, 3000)  
 });  
</script>  
</body>  
</html>

2：注意，最原始是返回到视图层的，当没有注解@RestController或者@ResponseBody，因为加这些注解表示返回json字符串，没有加就是返回一个视图层，就是html，那么sprinboot中，返回视图层，默认回去静态目录找资源，资源目录默认命名是templates



3：流程：

就是用户在客户端点击用户预览图片，就开始页面静态化，配置nginx交给nginx去访问后台预览图方便以后集群访问使用，负载均衡