

# 深入浅出 FreeMarker(二)-高级教程

Peter Wei

2012-3-6

搜狐技术部-我的搜狐

i.sohu.com

## Freemarker 的功能

- MVC 框架中的 View 层组件
- Html 页面静态化
- 代码生成工具
- CMS 模板引擎
- 页面栏目动态定制

## 基础教程

深入浅出 FreeMarker(一)-基础教程

<http://dev.nrcp.sohu.com/wiki/pages/viewpage.action?pageId=2720811>

- FreeMarker 介绍
- 开发指南-5 分钟入门
- 开发进阶
- 实例应用讲解(我的搜狐 marco 宏以及 layout 布局的使用)
- 常用 FreeMarker 资源

# 分享目标

掌握 macro 宏的使用

掌握 freemarker 在 MVC 框架 View 层中的使用细节

掌握 freemarker 结合 springmvc 的异常处理

深入了解 freemarker 在页面静态化以及 cms 中的使用，达到综合应用的能力。

## Macro 宏的使用

个人展示页 url 应用地址生成，有个性域名及没有的规则不一样。

### 宏 showUrl.ftl

```
<!--
    Description:根据domain或者upt、接入方式、appname动态生成url
    Author: Peter Wei (mailto:guangbowei@sohu-inc.com)
-->
<!--
拼装个人展示地址:因为会存在没有domain的用户
    domain:个性域名
    upt:经加密后的url passport
    host:主机
    accessType:接入方式
    appName:如blog,mblog,video等
-->
<#macro showUrl domain=show_domain!"" upt=show_upt!"" host="i.sohu.com"
accessType="index" appName="index">
<#escape x as (x!)?html>
<#if domain!="">
    <#if accessType=="proxy">
        http://${domain}.${host}/${appName}/<#t>
    <#elseif accessType=="appview">
        http://${domain}.${host}/app/${appName}/<#t>
    <#else>
        http://${domain}.${host}/<#t>
    </#if>
```

```

<#elseif upt!="">
    <#if accessType=="proxy">
        http://${host}/p/${upt}/${appName}/<#t>
    <#elseif accessType=="appview">
        http://${host}/p/${upt}/app/${appName}/<#t>
    <#else>
        http://${host}/p/${upt}/<#t>
    </#if>
</#if>
</#escape>
</#macro>

```

## Head.ftl

```

<#import "component/showUrl.ftl" as c>
<div class="main-nav-wrapper fr">
    <ul class="main-nav">
        <li><a href="@c.showUrl/">首页</a></li>
        <li><a href="@c.showUrl accessType="proxy"
appName="mblog" />index.htm">微博</a></li>
        <li><a href="@c.showUrl accessType="proxy"
appName="blog" />index.htm">博客</a></li>
        <li><a href="@c.showUrl accessType="proxy"
appName="album" />index.htm">相册</a></li>
        <li><a href="@c.showUrl accessType="proxy"
appName="video" />index.htm">播客</a></li>
        <li><a href="@c.showUrl accessType="proxy"
appName="scomment" />index.htm">我来说两句</a></li>
        <li class="main-nav-extend"><a href="@c.showUrl
accessType="appview" appName="baby" />">育儿</a></li>
        <li class="main-nav-extend"><a href="@c.showUrl
accessType="appview" appName="wenda" />">问答</a></li>
        <li class="main-nav-extend"><a href="@c.showUrl
accessType="proxy" appName="group" />index.htm">圈子</a></li>
        <li class="more-btn
more-left"><a href="javascript:void(0);">更多</a></li>
    </ul>
</div>

```

## 生成界面

```
<div class="main-nav-wrapper fr">
  <ul class="main-nav">
    <li><a href="http://runcodeman.i.sohu.com/">首
页</a></li>
    <li><a
href="http://runcodeman.i.sohu.com/mblog/index.htm">微博</a></li>
    <li><a
href="http://runcodeman.i.sohu.com/blog/index.htm">博客</a></li>
    <li><a
href="http://runcodeman.i.sohu.com/album/index.htm">相册</a></li>
    <li><a
href="http://runcodeman.i.sohu.com/video/index.htm">播客</a></li>
    <li><a
href="http://runcodeman.i.sohu.com/scomment/index.htm">我来说两句
</a></li>
    <li class="main-nav-extend"><a
href="http://runcodeman.i.sohu.com/app/baby/">育儿</a></li>
    <li class="main-nav-extend"><a
href="http://runcodeman.i.sohu.com/app/wenda/">问答</a></li>
    <li class="more-btn more-left"><a
href="javascript:void(0);">更多</a></li>
  </ul>
</div>
```

## Freemarker 使用细节

### Html 转义

```
<#escape x as (x!)?html>
  <div>...</div>
</#escape>
```

### 变量及全局变量

```

<#assign soml_follow="<soml.widget name='follow'/'>">
<#assign soml_grade="<soml.widget name='grade'/'>">
<#assign soml_visitor="<soml.widget name='visitor'/'>">
<#assign soml_interested="<soml.widget name='interested'/'>">
<#assign soml_follow_user="<soml.widget name='follow_user'/'>">

<#assign follow>
    <div class="rtbar-wrapper rtbar-follow">
        <ul class="rtbar-com">
            <li class="rtbar-shadow"><span><a href="<@s.showUrl
accessType="appview" appName="friend" />">跟随</a> <a
id="ms_num_follow" href="<@s.showUrl accessType="appview"
appName="friend" />">?</a></span></li>
            <li class="trbar-trailer"><span><a href="<@s.showUrl
accessType="appview" appName="friend"
/>!!/a/app/friend/fans/fansshow.do">跟随者</a> <a
id="ms_num_follower" href="<@s.showUrl accessType="appview"
appName="friend"
/>!!/a/app/friend/fans/fansshow.do">?</a></span></li>
            <li class="rtbar-news"><span><a href="<@s.showUrl/>">新鲜事</a>
<a id="ms_num_feeds" href="<@s.showUrl/>">?</a></span></li>
        </ul>
    </div>
</#assign>

<!--JS 版本号-->
<#global version="201202022018">

```

## <#t>标签

源代码清除空白不换行标签

```

<#if accessType=="proxy">
    http://${host}/p/${upt}/${appName}/<#t>
<#elseif accessType=="appview">
    http://${host}/p/${upt}/app/${appName}/<#t>
<#else>
    http://${host}/p/${upt}/<#t>
</#if>

```

## List 索引\_index

```

                <div
class="img-list1"><span></span>

                <ul><#if
root.subList[0]?exists&&root.subList[0].productList?exists>
                    <#list root.subList[0].productList as product>
                        <#if (product_index)=6>
                            <li>

<a href="${ctx}/product/${product.sid}.html">${product.productName?if
_exists}</a>

                            </li>
                        <#if (product_index)=9> <#break> </#if>
                    </#if>
                </#list>
            </#if>
        </ul>
    </div>

```

## Java 常量及枚举类型的使用

### 常量

```

/**
 * 布局类型
 */
public static final String LAYOUT_TYPE = "layout_type";

/**
 * 默认布局,内容区通栏
 */
public static final String DEFAULT_LAYOUT = "2";

/**
 * 两列布局,内容区分左右
 */
public static final String TWO_COLUMN_LAYOUT = "2";

```

## ConfigInterceptor

```
public void postHandle(HttpServletRequest request, HttpServletResponse
response, Object handler,
        ModelAndView modelAndView) throws Exception
{
    ModelAndView modelAndView) throws Exception {
        BeansWrapper bw = SimpleObjectWrapper.getDefaultInstance();
        TemplateHashModel statics = bw.getStaticModels();
        modelAndView.addObject("statics", statics);
    }
}
```

## ftl 页面调用

```
<#assign
PersonConstant=statics?if_exists["com.sohu.suc.person.constants.Perso
nConstant"]>

%{PersonConstant.TWO_COLUMN_LAYOUT}
```

## SpringMVC 中的配置

```
<bean id="freemarkerConfig"

    class="org.springframework.web.servlet.view.freemarker.FreeMarker
Configurer">
    <!-- templateLoaderPath和 freemarker ViewResolver中的prefix只能只
定一个，要不然会报找不着ftl的错误 -->
    <property name="templateLoaderPaths">
        <list>
            <value>/WEB-INF/pages/</value>
        </list>
    </property>
    <property name="freemarkerSettings">
        <props>
            <!-- 处理变量中的null值，自动转为空字符串 -->
            <prop key="classic_compatible">true</prop>
            <!--默认编码 -->
            <prop key="defaultEncoding">GBK</prop>
            <!--异常处理 -->
```

```

        <prop key="template_exception_handler">ignore</prop>
        <!-- <prop
key="template_exception_handler">com.sohu.suc.plaza.web.util.Freemark
erExceptionHandler
        </prop> -->

        <!-- 正式环境改为更大: 如1800(半小时) -->
        <prop key="template_update_delay">300</prop>
        <prop key="datetime_format">yyyy-MM-dd HH:mm:ss</prop>
        <prop key="date_format">yyyy-MM-dd</prop>
        <prop key="time_format">HH:mm:ss</prop>
        <!-- 防止出现1,234这样的数字格式 -->
        <prop key="number_format">#</prop>
        <prop key="boolean_format">true,false</prop>
        <!-- 清除空白 -->
        <prop key="whitespace_stripping">true</prop>

        <!-- 可以输出false -->
        <!-- <prop
key="auto_import">/common/index.ftl</prop><prop
key="tag_syntax">auto_detect</prop>
        <prop key="url_escaping_charset">GBK</prop> -->
    </props>
</property>
<property name="freemarkerVariables">
    <map>
        <entry key="webRoot" value="${webRoot}" />
        <entry key="cssRoot" value="${cssRoot}" />
        <entry key="jsRoot" value="${jsRoot}" />
    </map>
</property>

</bean>

```

## 异常处理

freemarker 提供了一个支持其错误处理机制的接口 `TemplateExceptionHandler`，需要自己去扩展实现，构造自己的处理 freemarker 模板错误的规范。

主要分为两个部分



## 实现接口

```
public class FreemarkerExceptionHandler implements
TemplateExceptionHandler {

    private static final Logger log =
Logger.getLogger(FreemarkerExceptionHandler.class);

    /* (non-Javadoc)TemplateException
    * @see
freemarker.template.TemplateExceptionHandler#handleTemplateException(
freemarker.template.TemplateException, freemarker.core.Environment,
java.io.Writer)
    */
    @Override
    public void handleTemplateException(TemplateException te,
Environment env, Writer out) throws TemplateException {
        try {
            out.write("[Freemarker Error: " + te.getMessage() + "]);
            log.warn("[Freemarker Error: " + te.getMessage() + "]);
        } catch (IOException e) {
            log.warn(e.getMessage());
            throw new TemplateException("Failed to print error message.
Cause: " + e, env);
        }

    }

}
```

## 接口加入 spring freemarker 配置中

```
<property name="freemarkerSettings">
<prop
key="template_exception_handler">com.sohu.suc.plaza.web.util.Freemark
erExceptionHandler
</prop>
</property>
```

# 页面静态化

## 为什么要静态化

互联网，特别是门户网站，用户量和请求量都特别大。  
如何解决高并发高负载的问题

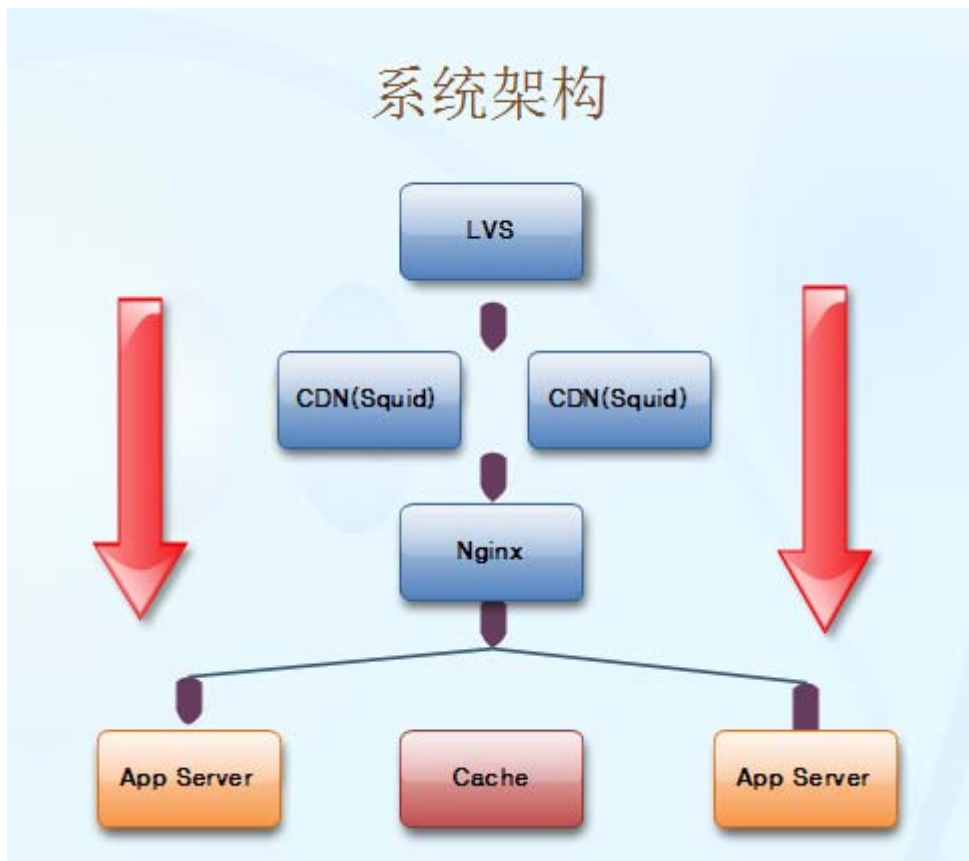
常规的手段

- CDN
- 负载均衡(nginx)
- 应用缓存(memcached,redis)
- 分库分表
- 图片服务器分离
- HTML 静态化

应用 Cache 再快，也快不过 html 静态化页面

## 静态内容 CDN 系统

图片、CSS、JS 脚本、html 静态化页面



用 squid 做前端 cache，而 nginx 在 squid 的后面提供真正的 web 服务。这样的架构要求大部分都是静态页面。

在这样的前提下，特别是新闻、广场等对实时性要求不高的系统，一般都会用 CMS 做动态内容静态化。

## 实例讲解

我的搜狐广场系统的设计与实现

我的搜狐

个人中心 我的展示 应用

金币 登录 注册 搜狗搜索

娱乐 ENTERTAINMENT

更多频道

[首页](#) | 
 [正在发生](#) | 
 [名人堂](#) | 
 [我来说两句](#) | 
 [博客](#) | 
 [图片](#) | 
 [问答](#)

香港苹果店迎客 全天候应对

### 热点新闻 TOP10 >>

1	第751期：导演呼吁票房降价：这才是真民意	35
2	董洁三岁儿子可亲近照曝光 与妈妈开心玩耍	20
3	付辛博韩国“万人”庆生 张佑赫陈翔惊喜共贺	15
4	张歆艺曾承认与王志飞分手：那七年是最美好时	11
5	《向上吧！少年》沈阳站海选即将启动	11
6	湖南卫视公益慈善节目凸显“雷锋”气质	6
7	陈思成因“穷”遭弃 梅新女友与李晨张译闹	6
8	白静和周成海追悼会时间未定 两人确定不合葬	5
9	杨幂助力《梦想成真》帮助贫困女孩圆梦	5
10	《匹夫》黄晓明惨遭扒衣 张歆艺承认七年情断	3

立即加入

登录名   
 密码

☐ 自动登录  
 忘记密码?

- 卡内资源丰富 会员权益试用
- 测试压力指数 赢取礼物
- 微博试用周周 大奖等你来
- HOLD好状态，大奖等你赢

### 博客 BLOG >>

	作者	阅读数	评论数
宋祖英穿皮草献24万名表贵气现身两会(组图)	头条八卦	1	0
周润发黄晓明两代许文强共演《大上海》值得期待	一条路旁	3	0
赵本山海南代言养生楼盘	长卷国策	3	0
孙俪增讯周润八卦造谣缺少智商	落目忘花	1	0
马原归来，文学得救了.....	兰瑟发	1	0
黄磊陈数“夫妻”写真照曝光	天长小月	6	0
Ella婚前半裸“玩火”欲疯狂到底	雨落花间	2	0
宋祖英两会穿华丽皮草引非议(图)	敬辞流风	0	

### 大家正在说

**田歌**  
 上传视频《大张伟精彩三分钟.mp4》：“光荣绽放 田歌访谈 大张伟”  
<http://t.itc.cn/YysAV> <http://t.itc.cn/Y2MbQ>  
 2012-03-06 12:46

---

**落目忘花**  
 这关系太复杂了，让人一头雾水，你能看懂了吗？ <http://t.itc.cn/Y6eMb>



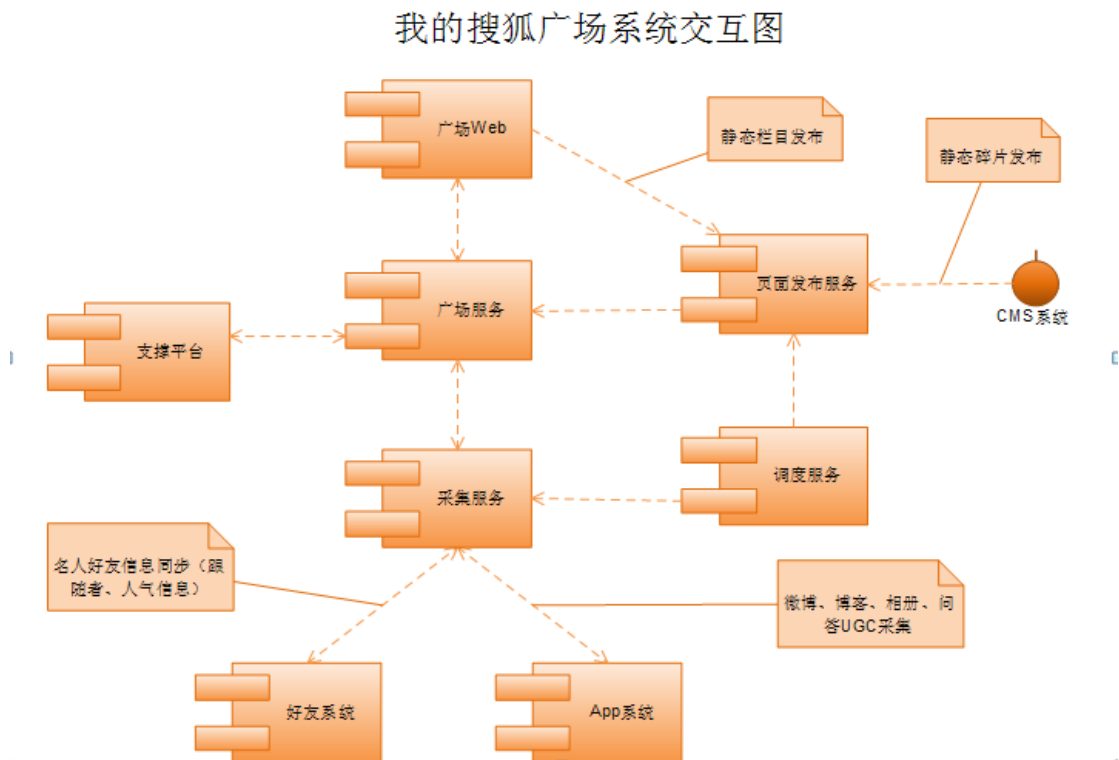
## 更新策略

不同类型的应用更新的周期都不一样

- 说两句新闻排行榜：每半小时自动更新一次，每次显示 15 条；
- 博客内容推荐：前端每 6 小时自动更新一次，每次显示 20 条；
- 相册内容推荐：前端每 12 小时自动更新一次，每次显示 20 条；

- 问答内容推荐：前端每 24 小时自动更新一次，每次显示 10 条；
- 微博内容推荐：前端每 10 分钟自动更新一次，每次显示 15 条，总计读取 30 条

## 总体设计



### 页面静态化：

各频道新闻首页及列表页采用页面栏目静态化的技术实现动态碎片的生成。

### 定时生成策略：

使用 quartz 调度任务根据特定模板和数据定时生成静态化页面

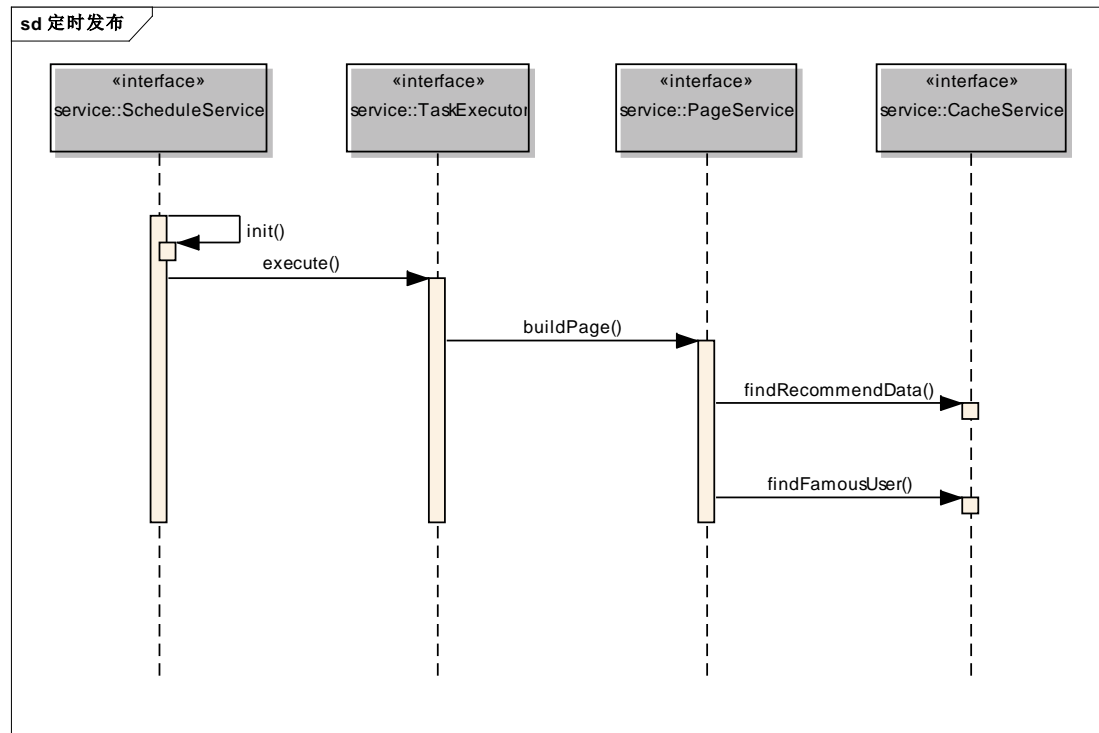
### 同步 Cache

采用 rsync 定时(每几分钟)同步各种静态化碎片页面到 CMS 的服务器做 CDN Cache.

## CMS 管理



## 定时发布



```
/**
 * 页面碎片生成Job.
 *
 * @author Peter Wei (mailto:guangbowei@sohu-inc.com)
 * @since 2011
 * @version 1.0.0
 */
public class PageJob implements Job {

    private static Logger log = Logger.getLogger(PageJob.class);

    @Override
    public void execute(JobExecutionContext context) throws
    JobExecutionException {
        PageService pageService = (PageService)
        ApplicationContext.getBean("pageService");

        Map<String, Object> pageContext = new HashMap<String, Object>();
        for (Object key : context.getMergedJobDataMap().keySet()) {
            pageContext.put((String) key,
            context.getMergedJobDataMap().get(key));
        }
    }
}
```



```

        pageService.buildPage(pageContext);
    }
}

```

id	task_name	task_gro...	t...	execute_class	s...	config	s.	end_time	r.	cron
1	微博定时采集	plaza	3	com.sohu.suc.plaza.quartz.job.MblogGath	11		197	1970-07-01 00:00:00	0	0/5 * * * ?
2	博客定时采集	plaza	3	com.sohu.suc.plaza.quartz.job.BlogGathe	11		197	1970-07-01 00:00:00	0	0/5 * * * ?
3	相册定时采集	plaza	3	com.sohu.suc.plaza.quartz.job.AlbumGath	11		197	1970-07-01 00:00:00	0	0/5 * * * ?
4	好友跟随数采集	plaza	3	com.sohu.suc.plaza.quartz.job.FansGathe	12		197	1970-07-01 00:00:00	0	0/5 * * * ?
5	微博推荐页生成	plaza	3	com.sohu.suc.plaza.quartz.job.PageJob	10	template=mblog_recommend.ftl;appType=1;pageType=1;itemN	197	1970-07-01 00:00:00	0	0/3 * * * ?
6	微博二级页生成	plaza	3	com.sohu.suc.plaza.quartz.job.PageJob	10	template=blog_list.ftl;appType=1;pageType=2;itemNum=100;	197	1970-07-01 00:00:00	0	0/3 * * * ?
7	博客推荐页生成	plaza	3	com.sohu.suc.plaza.quartz.job.PageJob	10	template=blog_recommend.ftl;appType=2;pageType=1;itemNui	197	1970-07-01 00:00:00	0	0/3 * * * ?
8	博客二级页生成	plaza	3	com.sohu.suc.plaza.quartz.job.PageJob	10	template=blog_list.ftl;appType=2;pageType=2;itemNum=200;re	197	1970-07-01 00:00:00	0	0/3 * * * ?
9	相册推荐页生成	plaza	3	com.sohu.suc.plaza.quartz.job.PageJob	10	template=album_recommend.ftl;appType=3;pageType=1;itemN	197	1970-07-01 00:00:00	0	0/3 * * * ?
10	相册二级页生成	plaza	3	com.sohu.suc.plaza.quartz.job.PageJob	10	template=album_list.ftl;appType=3;pageType=2;itemNum=80;re	197	1970-07-01 00:00:00	0	0/3 * * * ?
11	名人推荐页生成	plaza	3	com.sohu.suc.plaza.quartz.job.PageJob	10	template=fame_recommend.ftl;appType=7;pageType=1;itemN	197	1970-07-01 00:00:00	0	0/3 * * * ?
12	名人二级页生成	plaza	3	com.sohu.suc.plaza.quartz.job.PageJob	10	template=fame_list.ftl;appType=7;pageType=2;itemNum=280;x	197	1970-07-01 00:00:00	0	0/3 * * * ?

## Freemarker 配置

这个可以单独配在 spring 中，也可以和 springmvc 结合使用

模板加载路径可以是文件系统，也可以是类路径，也可以是 web 路径

```

<bean id="freemarkerConfiguration"

    class="org.springframework.ui.freemarker.FreeMarkerConfigurationF
actoryBean">
    <property name="templateLoaderPaths">
        <list>
            <value>/WEB-INF/pages/template/</value>
            <value>file:/opt/template/plaza</value>
            <value>classpath:/template/plaza</value>

        </list>
    </property>
    <property name="freemarkerSettings">
        <props>
            ...
        </props>
    </property>

```

和 web 工程中的稍不一样，这个是 Springmvc 中用的.本质上是一样的，最后调的都是同一个东西。

```
org.springframework.web.servlet.view.freemarker.FreeMarkerConfigurer
```

# 代码实现

## PageService

```
/**
 * 广场页面服务:生成静态化页面.
 *
 * @author Peter Wei (mailto:guangbowei@sohu-inc.com)
 * @since 2011
 * @version 1.0.0
 */
public interface PageService {

    /**
     * 根据上下文生成页面文件
     *
     * @param context 上下文数据, 如从调度任务传来的一些参数
     * @return
     */
    public int buildPage(Map<?, ?> context);

    /**
     * 根据上下文和数据生成页面文件
     *
     * @param context 上下文数据, 如从调度任务传来的一些参数
     * @param data 生成数据
     * @return
     */
    public int buildPage(Map<?, ?> context, Map<?, ?> data);
}
```

## TemplateService

```
/**
 * 模板服务.
 *
 * @author Peter Wei (mailto:guangbowei@sohu-inc.com)
```

```

* @since 2011
* @version 1.0.0
*/
public interface TemplateService {

    /**
     * get Template获取模板
     *
     * @param name
     * @return
     */
    public Template getTemplate(String name);

    /**
     * get Template String Content获取模板(字符串格式)
     *
     * @param name
     * @return
     */
    public String getTemplateContent(String name);

    /**
     * 根据模版名称和对应的数据渲染模版
     *
     * @param name 模板名称
     * @param data 数据
     * @return 生成的字符串内容
     */
    public String buildPage(String name, Map<?, ?> data);

    /**
     * 根据模版名称和对应的数据渲染模版,并生成文件
     *
     * @param name 模板名称
     * @param data 数据
     * @param fileFullName 包括路径的文件名
     * @return
     */
    public int buildPage(String name, Map<?, ?> data, String
fileFullName);

    /**
     * 根据模版名称和对应的数据渲染模版,并生成文件
     *

```

```

    * @param name 模板名称
    * @param data 数据
    * @param filePath 文件路径
    * @param fileName 文件名
    * @return
    */
    public int buildPage(String name, Map<?, ?> data, String filePath,
String fileName);

    /**
     * 根据模版名称和对应的数据渲染模版,并生成文件
     *
     * @param name 模板名称
     * @param data 数据
     * @param file 文件
     * @return
     */
    public int buildPage(String name, Map<?, ?> data, File file);

    /**
     * 初始化上下文
     */
    public void initialize();
}

```

## TemplateServiceImpl

```

@Component("templateService")
public class TemplateServiceImpl implements TemplateService {

    private static final Logger log =
Logger.getLogger(TemplateServiceImpl.class);

    /**
     * 如果不是web环境,可以设成静态单例,现为在spring中配置:
     *
     * <pre>
     * <bean id="freemarkerConfiguration"
     *
     * class="org.springframework.ui.freemarker.FreeMarkerConfigurationFacto
ryBean">
     * </bean>
     * </pre>

```

```

    */
    @Autowired
    @Qualifier("freemarkerConfiguration")
    Configuration cfg;

    /**
     * 由于我们是web工程，所以可以重用freemarker中的配置
     */
    // @Autowired
    // @Qualifier("freemarkerConfig")
    FreeMarkerConfig freemarkerConfig;

    /**
     * 是否初始化
     */
    private static boolean init = false;

    @Override
    public Template getTemplate(String name) {
        if (!init) {
            initialize();
        }
        Template template = null;
        try {
            template = cfg.getTemplate(name,
PlazaConstant.DEFAULT_ENCODING);
        } catch (IOException e) {
            log.warn("get template error,name:" + name + "." +
e.getMessage());
        }
        return template;
    }

    @Override
    public String getTemplateContent(String name) {
        return getTemplate(name).toString();
    }

    @Override
    public String buildPage(String name, Map<?, ?> data) {
        StringWriter result = new StringWriter();
        Template template = getTemplate(name);
        try {
            template.process(data, result);

```

```

    } catch (Exception e) {
        log.error(e.getMessage(), e);
    }
    return result.getBuffer().toString();
}

@Override
public int buildPage(String name, Map<?, ?> data, String fileFullName)
{
    Writer result;
    try {
        result = new BufferedWriter(new OutputStreamWriter(new
FileOutputStream(fileFullName),
        PlazaConstant.DEFAULT_ENCODING));
        Template template = getTemplate(name);
        template.process(data, result);
    } catch (Exception e) {
        log.error(e.getMessage(), e);
        throw new BizException(e.getMessage(), e);
    }
    return 0;
}
}

```

## 数据库加载模板

当然也可以把模板内容存入数据库，在应用启动时，批量初始化模板。是字符串格式的，这是和上面文件格式的区别。

## StringTemplateServiceImpl

```

//@Component("stringTemplateService")
public class StringTemplateServiceImpl implements TemplateService {

    private static final Logger log =
Logger.getLogger(StringTemplateServiceImpl.class);

    //也可以像上面的例子一样，从spring注入。
    private static Configuration cfg = new Configuration();

```

```

private static boolean init = false;

//如果template以字符串形式存在数据库中，则以此种方式加载。
public void initialize() {
    StringTemplateLoader stringLoader = new StringTemplateLoader();
    //从数据库中加载模板
    String templateName = "";
    String template = "";
    stringLoader.putTemplate(templateName, template);
    cfg.setTemplateLoader(stringLoader);
    cfg.setDefaultEncoding(PlazaConstant.DEFAULT_ENCODING);
    init = false;
}

public void reload() {
    init = false;
}

@Override
public Template getTemplate(String name) {
    // TODO Auto-generated method stub
    return null;
}

@Override
public String getTemplateContent(String name) {
    // TODO Auto-generated method stub
    return null;
}

@Override
public String buildPage(String name, Map<?, ?> data) {
    if (!init) {
        initialize();
    }
    StringWriter result = new StringWriter();
    Template template = null;
    try {
        template = cfg.getTemplate(name,
PlazaConstant.DEFAULT_ENCODING);
        template.process(data, result);
    } catch (Exception e) {
        log.error(e.getMessage(), e);
    }
}

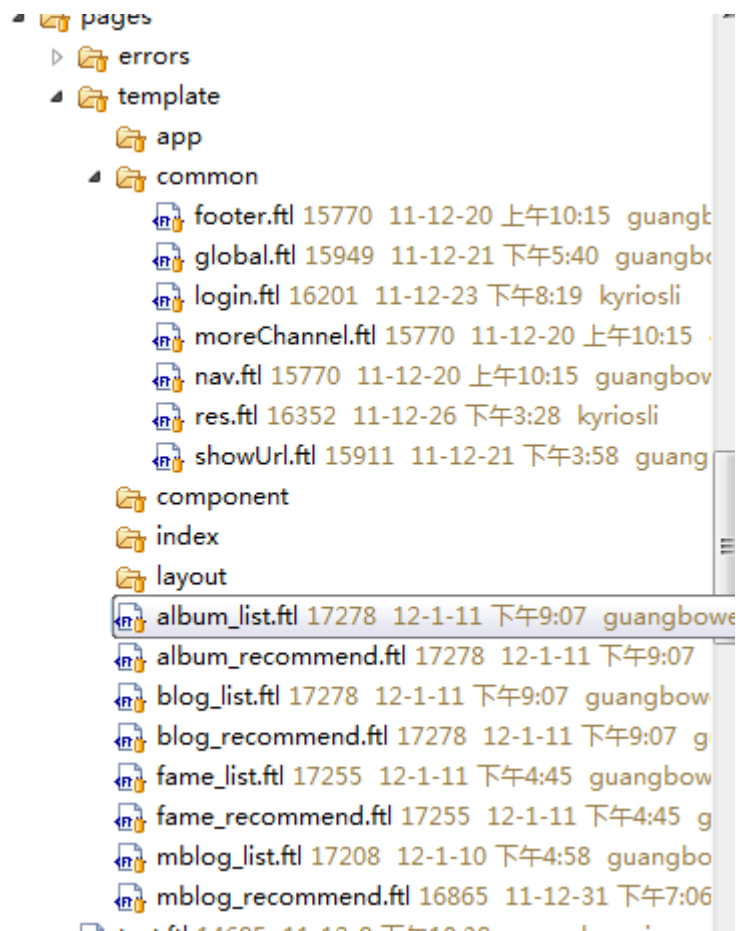
```

```

        throw new BizException(e.getMessage(), e);
    }
    return result.getBuffer().toString();
}

```

## 动态碎片模板



某个具体的碎片模板

```

<div class="blogList">
    <ul>
    <#if contentList?exists>
        <#list contentList as content>
            <li>
                <span
class="commentNum">${content.commentCount?if_exists}</span>
                <span class="radeNum">${content.count?if_exists}</span>
                <span
class="writer"><a href="${content.publishUrl?if_exists}"
target="_blank"

```



```
title="${content.publisher?if_exists?html}">${content.publisher?if_exists}</a></span>
    <span
class="txt"><a href="${content.contentUrl?if_exists}" target="_blank"
title="${content.content?if_exists}">${content.content?if_exists}</a>
</span>
    </li>
</#list>
</#if>

</ul>
</div>
```

## 工程代码

详见 svn 工程

<http://dev.nrcp.sohu.com/svn/suc/trunk/suc-plaza>

suc-plaza-service

suc-plaza-schedule

suc-plaza-web

## 关于我

- 邮箱地址: [weigbo@163.com](mailto:weigbo@163.com)
- MSN: [weigbo@163.com](mailto:weigbo@163.com)
- 我的搜狐: <http://fastwei.i.sohu.com>
- 新浪微博: <http://weibo.com/fastwei>
- 个人网站: <http://www.peterwei.net>