

## 目录

1、模板层的目录文件规划 .....	2
2、如何根据 url 定位页面模板 .....	3
2.1 定位技巧 1——在文件夹中查找 .....	4
2.2 定位技巧 2——screen 下 vm 模板对应线上页面 .....	5
3、screen、layout、control 深化理解 .....	7

## 一、模板层的目录文件规划

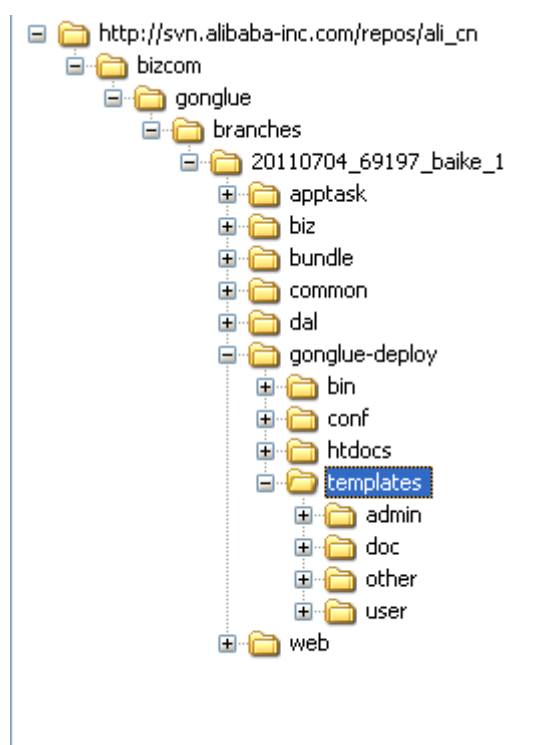
中文站使用的模板引擎是 velocity 模板 ,同时基于后端框架 webx 进行了进一步封装。

相对于基础的 velocity ,封装后的引擎给模板开发带来了更多的便利。

首先 , velocity 模板达到的第一点就是页面展示逻辑与后台业务逻辑的分离。

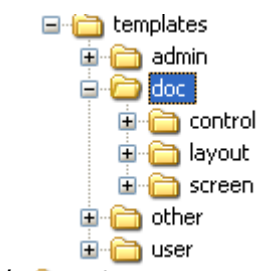
观察一下下面图中某个应用的 svn 分支 , **templates** 就是专门放置 vm 模板的位置 ,

即专门放置页面 html 代码片段的文件夹 ,自然是前端同学最关心的目录。



同时 , 它给我们提供了更符合前端页面科学的 3 个概念 : **layout**、**control**、**screen**。

以下是映射 templates 目录下的某个文件上的目录结构 :



3 个文件夹中的 vm 模板文件 ,都有其特殊的含义 ,基础语法篇中简单介绍过这 3 个概念。

### screen

- 页面主体内容区
- 线上动态页面会对应**唯一**一个screen下vm

### layout

- 公用的页面框架，产品级html载体
- 和产品级的js、css文件三位一体

### control

- 多个页面公用的代码功能片段
- 类似于widget的概念

以上是文件部署上的一些基础概念。下面介绍大家比较关心的实际开发中的技巧。

## 二、如何根据 url 定位页面模板

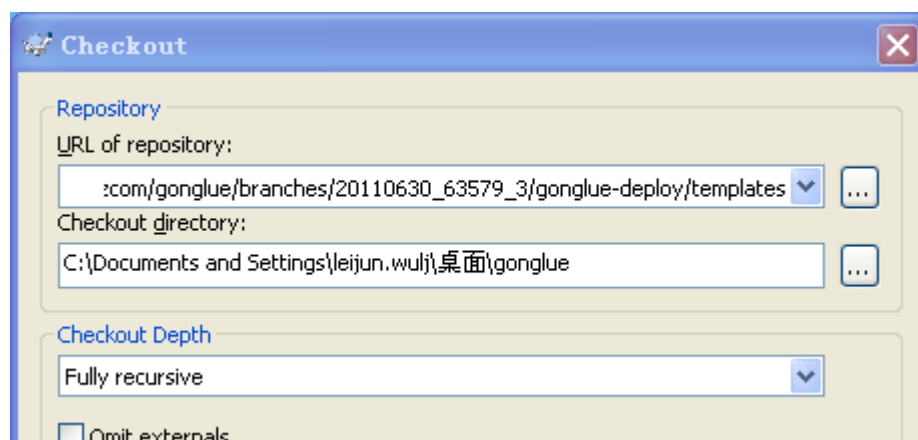


有这么一个**业务需求**：

修改生意经首页（<http://baike.china.alibaba.com/doc/index.html>）生意经分类这个文案，改成生意经类目，且修改[全部]的链接。[全部]链接原来的 url 是：  
[http://baike.china.alibaba.com/doc/list\\_all-1--.html](http://baike.china.alibaba.com/doc/list_all-1--.html)，  
需要修改成 <http://baike.china.alibaba.com/doc/category/index.html>。

这个需求 HTMLdemo 甚至不需要修改，直接到模板中修改一下就可以。那么就是如何找到它的代码所在模板的问题了。

baike.china.alibaba.com 是 gonglue 应用的域名，那么我们只需要有 gonglue 应用的 svn 代码分支中的 templates 目录，理论上就能找到需要修改的代码位置。



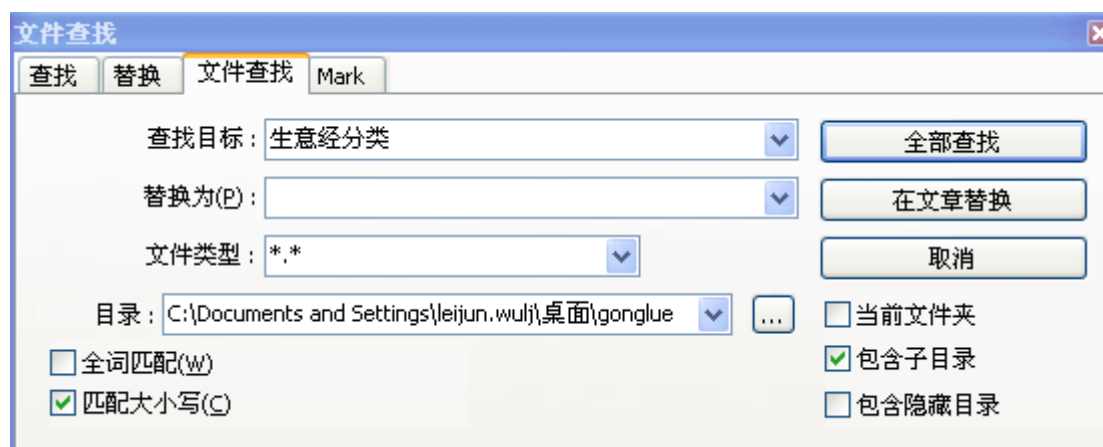
将分支 checkout 到本地（或者某台服务器上）

### 下面介绍**定位技巧 1——在文件夹中查找**

我们有几个比较关键的信息，要修改的是“生意经分类”这个文案，当然，我们也可以打开 firebug，看到 html 的代码结构。

```
<div class="module mod-index-graybox mod-index-category">
  <div class="cell-hd">
    <h3>生意经分类</h3>
    <a class="fd-right" title="全部" href="http://baike.china.alibaba.com/doc/list_all-1-.html">[全部]</a>
  </div>
  <div class="cell-bd">
```

比较巧妙的方法，很多编辑器等自带的功能——文件查找，来找出对应的代码。



在 checkout 后的本地文件夹 gonglue 中查找“生意经分类”，找出一个结果：

```
Search "生意经分类" (1 hits in 1 files)
C:\Documents and Settings\leijun.wulj\桌面\gonglue\doc\screen\index.vm (1 hits)
Line 48: <h3>生意经分类</h3>
```

哦，那就是在 index.vm 中了，打开确认后修改即可。

当然，有时候可能这个条件不一定能找到，如果生意经分类是后端数据库中读取出来的，自然在文件夹中找不到这个。可以适当根据其它个性化的特点来寻找，比如它的 html 中的 mod-index-category 这个 class。

这个技巧很实用，能在很多场景下使用，不仅仅局限在模板开发中，js 中也能通过类似方法，不过也有其局限性。

优点：

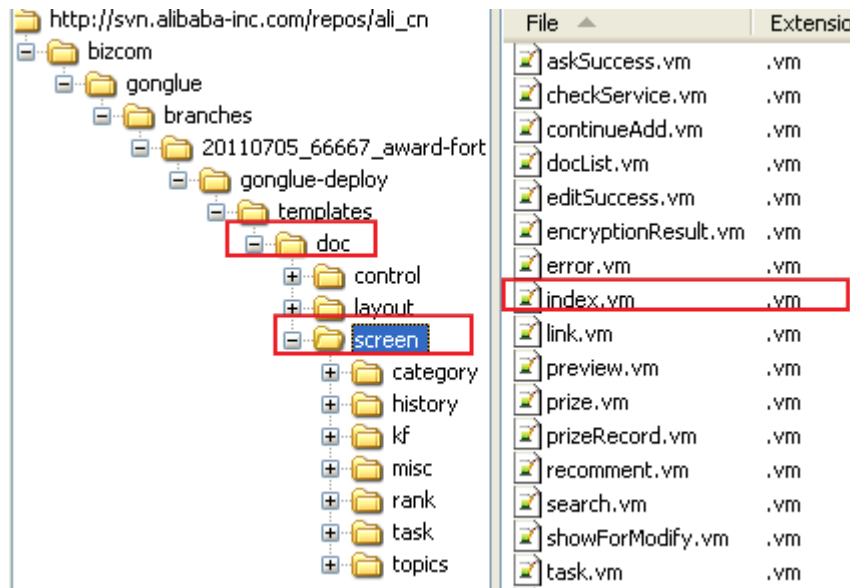
- ① 查找结果很有针对性
- ② 清晰定位出查找的条件涉及的相关文件，方便确认排查影响范围。

缺点：

- ① 条件必须相对特殊，否则查找命中率不高
- ② 查找条件如果是目标文件夹以外提供的话，就找不到结果

## 定位技巧 2——screen 下 vm 模板对应线上页面

还是修改生意经首页（<http://baike.china.alibaba.com/doc/index.html>）下修改文案的需求。



观察分支情况后，发现 templates 文件夹下面有个 doc 文件夹，doc 文件下面分成了 screen、layout、control，而 screen 下面，则有一个 index.vm。

有一个规律，screen 下面的 vm 会对应线上的一个页面。

而比较 **templates->doc->screen->index.vm** 和

**<http://baike.china.alibaba.com/doc/index.html>** 可以轻易发现 两者的 url 完全对应。

事实确认如此，index.vm 就是 index.html 这个页面对应的模板，要修改 index.html 中的东西，自然起点就是 index.vm 这个 screen。

还有一类特殊的页面，url 组装可能比较特殊，如 detail 页面、list 页面，从 url 中也能可见其个性化的一面，如

**<http://baike.china.alibaba.com/doc/view-d21278359.html>**。

数字这块代表这个 detail 的 id，会是变化的。

但其实它本质是一个页面，只是 id 的变化，从数据库中获取的内容造成差异，url 中也有迹可循，都是有同一个模板 **view.vm** 生成的。这也正是模板的魅力，通过一个模板，造成千变万化的效果。

还有一些特殊的 url，**<http://shhjuneng.cn.alibaba.com/>**，从 url 上看不出它是什

么模板生成的任何端倪，但是我们可以初步判断出，它其实是经过跳转的，它的真实页面地址肯定不是这个，而本产品线的同学一般知道 athena 应用中的页面。

这类 url 的话，如果不是对这个页面在应用中的情况有了足够了解，可能就需要后端同学的协助来定位模板了，或者其实可以巧用在文件夹中查找这个方法了，通过利用这个页面中的特殊片段，在 athena 应用的 templates 中查找。

如果有经验了，其实还可以知道，这个页面的经过跳转前的原始 url 是 <http://shhjuneng.cn.alibaba.com/athena/index.html>。

最终，我们都能定位到应用 templates 目录下 screen 中的某个 vm 来对应这个页面 url。

有些页面也可能更复杂，更需要我们去深入的了解应用、熟悉应用，才能更好的优化我们的产品。

优点：

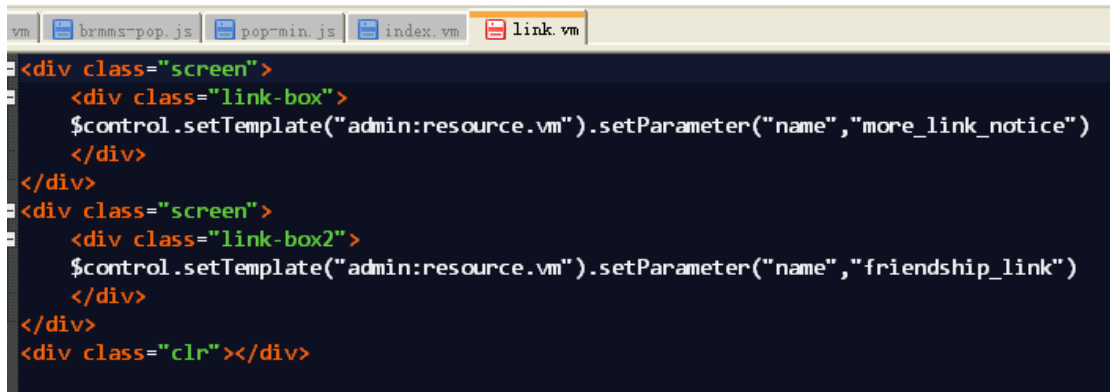
- ① 程序约定层面的规则，更严谨
- ② 定位页面中的架构能清晰掌握

缺点：

- ① 特殊 url 不容易找到对应的 vm
- ② 需要一定经验

### 三、screen、layout、control 深化理解

<http://baike.china.alibaba.com/doc/link.html> 我们轻易找到了 screen 下的 link.vm。



```
<div class="screen">
  <div class="link-box">
    $control.setTemplate("admin:resource.vm").setParameter("name","more_link_notice")
  </div>
</div>
<div class="screen">
  <div class="link-box2">
    $control.setTemplate("admin:resource.vm").setParameter("name","friendship_link")
  </div>
</div>
<div class="clr"></div>
```

这个 vm 文件中，有 control，这个较好理解，control 就是公用的代码片段。页面哪个部分需要使用，就在自己的位置调用下。而这个文件最特殊的地方是，它没有外层的页面框架，只有主体区域的代码，而 head 标签等部分，都没有在这个片段中。缺少了页面框架，却能生成一张页面，肯定有其独特之法。

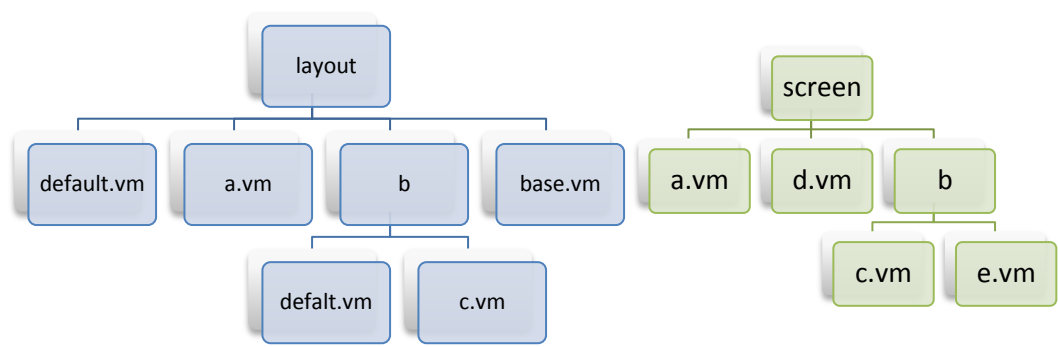
这独特之法，正是模板的奇妙之处，layout 的用武之处。页面框架，即页面风格布局，当然对一个产品来说，页面风格布局一般就只有几种，所以 layout 有其特殊的意义。

### 下面介绍下 screen 下 vm 自动查找页面 layout 的规则：

假如 screen 模板名字为/xxx/yyy/hello.vm，那么 webx 将依次查找下面的 layout 模板：

- 1./xxx/yyy/hello.vm （如果找不到，尝试下一个）
- 2./xxx/yyy/default.vm （如果找不到，尝试下一个）
- 3./xxx/default.vm （如果找不到，尝试下一个）
- 4./default.vm （如果找不到，尝试下一个）
- 5.（如果上面所有都找不到，就显示不带 layout 的 screen）





所以，上图中的对应关系为

screen	layout
b/e.vm	b/default.vm
b/c.vm	b/c.vm
d.vm	default.vm
a.vm	a.vm

自动查找的规则，符合程序语言的约定，但需要一定的理解成本（找 layout 的时候，需要重温一遍规则），而且容易造成布局代码的冗余。同时，很难排查出 default.vm 被使用的范围，需要一个个人为检查，不利于产品维护升级。

举个小例子，比如上图中，b 文件夹下面的 default.vm 是一套和/default.vm 页面布局风格不同的框架。b 文件夹下面，大部分子页面都会使用/b/default.vm 风格的布局，但是有一张子页面 c.vm 要使用/default.vm 的布局，按照自动查找的规则，显然做不到。除非新建一个/b/c.vm 的 layout，不过这样/b/c.vm 和/default.vm 其实又是同一套代码。

显然，这个并不是理想的方案。但是，值得庆幸的是，webx 提供了一个开发者人为指定 layout 的方式。

在 screen 中使用语法：

**\$rundata.getModuleInfo().setLayoutTemplate("base.vm")**

则页面组装时，会使用人为指定的 layout。而且，一些比较特殊的页面，比如接口页面，也会使用 \$rundata.getModuleInfo().setLayoutTemplate(null) 来保证返回纯粹的 json 结构。

**这个相较于自动寻找 layout 的机制，有几个优点：**

- ① 代码重用性很高，冗余较低
- ② 调用关系清晰，screen 中就可以快速判断出它的页面框架 layout
- ③ layout 的被调用范围清晰可查，方便以后 layout 升级
- ④ css、js 对应性更强，产品级合并 style 文件对应产品级 layout
- ⑤ 清晰的调用关系，更符合严谨开发的习惯

所以，**建议使用指定的 layout 的方式**，这样会给我们产品维护、应用中的模板架构、关联的 css、js 排查带来极大便利。

**对模板开发的理解：**

模板开发，是前端可以拓展的其中一个技能。基于前端的基础技能 HTML、JS、CSS，结合到应用开发中的一个专业纵深。

模板开发是前后端工作配合非常紧密的点。其中主要进行的也是 HTML 的拆解和程序动态数据的填充。HTML 拆解，相对来说前端同学更加擅长。而程序动态数据的填充，分成逻辑变量、内容变量，逻辑变量的话，类似是否有诚信通 icon 的渲染逻辑判断，内容变量的话，不过是在指定的 html 位置，填充数据。

掌握了模板开发，虽然对前端带来了更多的要求，但同时也极大提升前端开发的价值和

前端开发主动探索的可能。应用开发，可能是前端开发一个很好的职业纵深。前端，也是一个能获得技术底蕴支撑的职业。

模板开发，也是让我们接触后端技术，获得技术支撑最好的机遇。