页面静态化

1. 概述

动态页面处理方式：

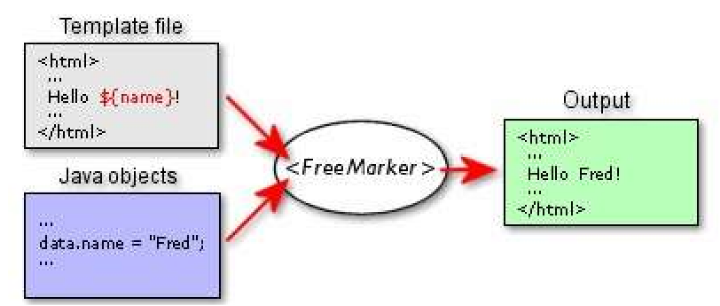
* 用户打开某个商品详情页面，前端将商品id等信息返回后端
* 后台根据商品id等信息查询数据库中商品信息，并返回前端
* 前端根据后台返回的商品信息动态渲染页面

如果系统某些页面（比如商品页面）流量很大，采用上面动态页面的方式，用户访问页面时都需要查询数据库，这样对系统带来很大负担的同时页面展示的速度也容易出现慢的情况。

所以为了减轻后台的负担，提升用户体验，我们采用页面静态化的处理。本文采用FreeMarker来实现页面静态化的处理。

1. FreeMarker原理

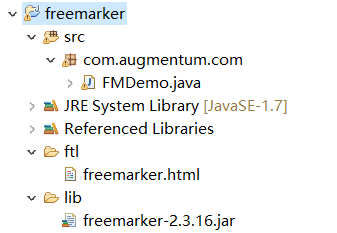
FreeMarker是一个基 于Java的开发包和类库的一种将模板和数据进行整合并输出文本的通用工具，FreeMarker实现页面静态化的原理是：将页面中所需要的样式写入到FreeMarker模板文件中，然后将页面所需要的数据进行动态绑定并放入到Map中，然后通过FreeMarker的模板解析类process()方法完成静态页面的生成。其工作原理如图所示。



模板 +  数据模型 = 输出

1. 示例演示FreeMarker

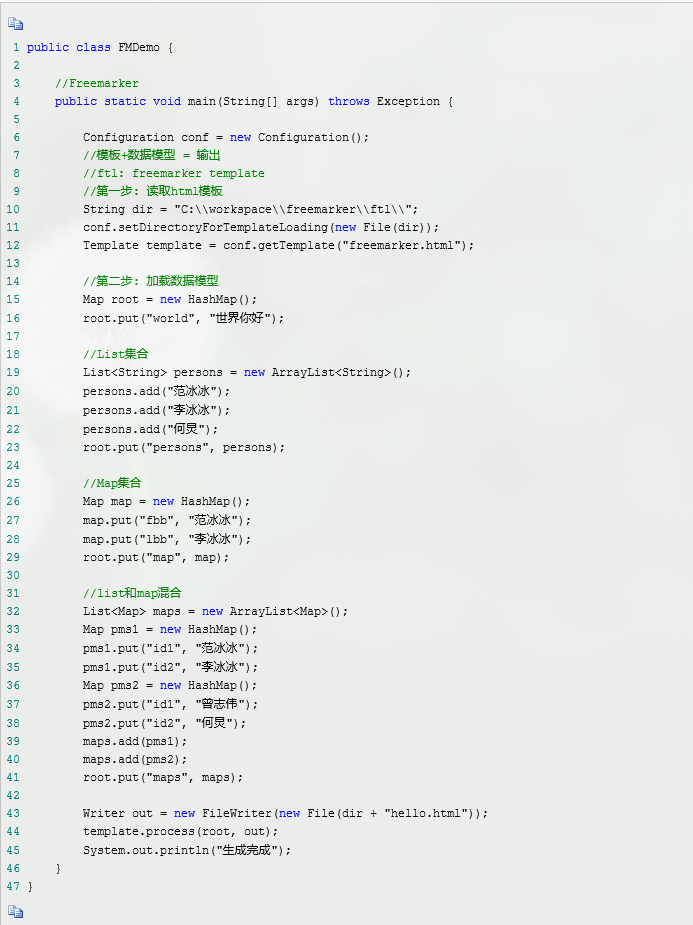
**先看一下Demo项目的整体结构:**



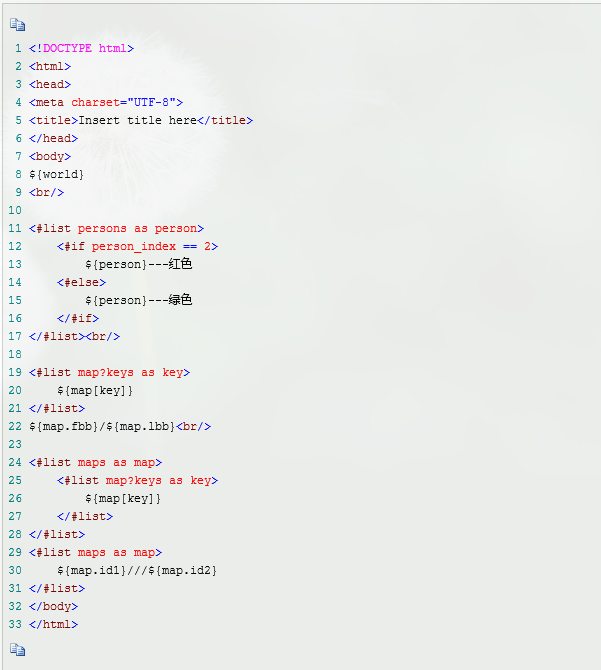
上面我们已经说了, 模板+数据模型=输出, 那么我们就一个个看模板和数据模型是什么样子的, 以及最后的输出是什么样子的.

注: 这里将省略freemarker的语法, 因为很多都是类似EL表达式的, 这里只提供几种情况的讲解, 其中包括: list, map, list和map混合

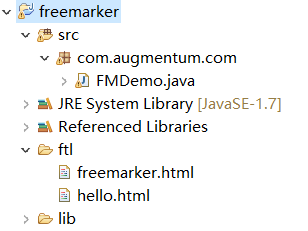
FMDemo.java:



freemarker.html: 模板文件



执行FMDemo.java中的Main方法,这会生成:



hello.html:



1. 总结

以上只是简单介绍了freemarker，通过freemarker生成静态html文件，用户就可以直接访问。如生成某个商品的静态文件id.html，用户浏览该商品详情显示（注意url的不同）如下图：



1. 静态化的优缺点

* **可以提高网站的打开速度**

页面静态减少了数据库查询工作，所以打开页面速度会加快。

* **有利于搜索引擎抓取和索引**

网站静态化之后，网页地址中没有参数，所以有利于搜索引擎抓取、分析、索引，从而对增加网站被搜索引擎收录的网页数量有非常大的帮助，对关键字在搜索引擎上的自然排名也有一定的帮助。

* **可以减轻服务器压力**

静态化之后的网页，在用户访问的时候，不需要再次查询数据库，所以对减轻网站服务器的压力非常有帮助。

* **内容更新不能即时显示**

对网站数据进行修改的时候，在网页前端不能即时显示出来，需要对网页进行静态化之后，修改的才能显示出来。因为静态生成的网页，是不能自动查询数据库中的数据，所以对数据发生的变化不能即时响应，需要重新运行静态化生成程序，根据新修改的数据重新生成静态页面，如果页面比较少的时候，这都没有关系。如果当网站规模很大，这里所产生的额外工作量将会非常大。

* **需要更大的服务器硬盘容量**

生成静态化网页之后，需要额外的服务器硬盘容量存放生生的静态化文件。