

所有的迷茫都是想得太多，做的太少，共勉！

5

收藏

评论

微信

微博

QQ

原

谈谈渲染，玩玩nginx——前后端分离，转发请求到Tomcat的尝试

2016年08月07日 21:42:51

阅读数：20189

一、谈谈“渲染”

相信好多人都挺听过“渲染”这个词，但不清楚它是什么意思？前端开发以为这是后端的活儿，后端开发以为是前端的事儿，推着推着就不了了之。其实渲染很简单，不说概念，直接举例：

1、后端渲染：以JSP为例，可以分成三步

- a、编写标签或Java代码（可以称之为模板）
- b、在JSP编译阶段被转换成Servlet编译为Servlet Class
- c、执行编译后的代码，将响应（模板执行结果）返回给页面

优势：减少前端工作，前端只需要设计纯页面，其他的都由后端来做；

缺点：依赖于服务器端，增大服务器压力，前后端职责分工不明确；

应用场景：在页面不太多、渲染压力不大、服务器端能够承受范围内可以使用后端渲染。

2、前端渲染：以基于JS的模板引擎为例

- a、编写模板代码
- b、通过模板引擎将模板转化为脚本语言，拼接在JS中（第一次拼接，以后使用缓存）
- c、页面加载执行JS

优势：减少服务器压力，前后端职责可以很好地分开，后端只做Json数据接口，前端进行渲染；

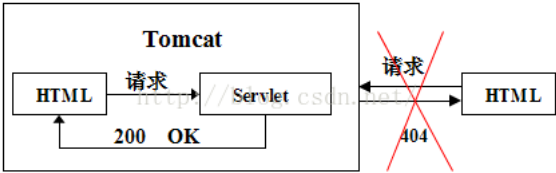
缺点：前端渲染依赖于客户端，增大的前端压力，需要代理服务器、未班渲染引擎的支持；

应用场景：在前端页面较多，前端开发人员能力较强，需要前后端分离的场景可以使用前端渲染（前端渲染是趋势）。

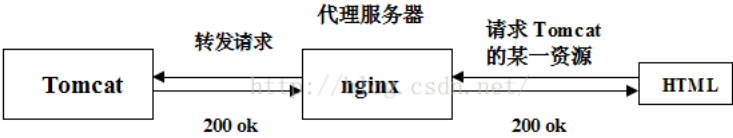
二、谈谈nginx

1、谈谈为什么会用到nginx？

首先明确一件事，浏览器可以发出请求吗？可以！那我们为什么要用到服务器呢？因为我们的前端如果不依赖服务器，页面就只能访问本地资源而不能访问服务器上的资源，而我们的后台一定是写在服务器上的。所以举个例子，我们在Tomcat服务器时，就必须把前端资源架在Tomcat上，才能访问后台的servlet。如下图所示：



所以当我们希望前后端分离时，前端的资源就不能放在Tomcat上面，那如何获得Tomcat的资源呢？这就用到了nginx，如下图所示：



2、谈谈nginx的反向代理

有反向代理必有正向代理，先谈谈**正向代理**：一般默认的代理都是正向代理，用户访问不了一个资源，然后通过代理服务器去访问这个资源，将响应带回给用户。关键在于用户知道自己访问的是其他服务器的资源，代理服务器不会掩饰URL。

而**反向代理**是，代理服务器也是在中间层，但是用户不知道自己访问的资源是其他服务器的资源，代理服务器会掩饰URL。

3、谈谈如何使用nginx反向代理tomcat

(1) 首先打开nginx，两种方式，一种是直接点击nginx.exe，一种是使用命令行，cd到nginx目录下，start nginx，无报错即启动成功

```
D:\>cd nginx
D:\nginx>cd nginx-1.10.1
D:\nginx\nginx-1.10.1>start nginx://blog.csdn.net/
D:\nginx\nginx-1.10.1>
```

(2) 启动成功后，如何验证，因为nginx.conf核心配置文件默认配置监听80端口，所以浏览器打开localhost，看到如下显示：



(3) 下一步就是配置反向代理Tomcat，打开conf目录下的nginx.conf文件，主要看35行左右开始的代码，下面是我修改过的代码：

主要修改location属性，使所有的请求都被转发到http://localhost:8080的Tomcat服务器下处理：

```
35     server {
36         listen      80;
37         server_name tomcat;
38
39         #charset koi8-r;
40
41         #access_log logs/host.access.log main;
42
43         location / {
44             proxy_pass http://localhost:8080;
45         }
```

listen：是监听的端口，即用户访问nginx服务的端口

server_name：服务名，经过测试并不会影响到什么

location：定义资源类型与服务器中资源地址url的映射关系，可在/后面定义资源类型，可设置多个location

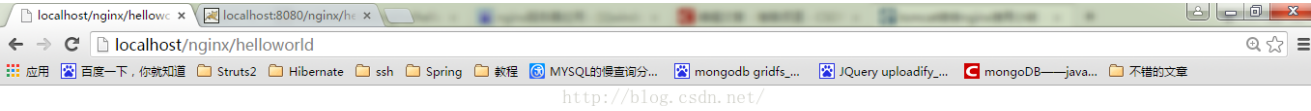
其中**proxy_pass**代表要反向代理的服务器资源url，只要资源类型匹配，在这个url下的子路径资源都可以访问到，

一个location中只能配置一个root或proxy_pass。

(4) 修改后nginx.conf文件后，使用nginx -s reload指令，重启nginx，如果没有报错即重启成功

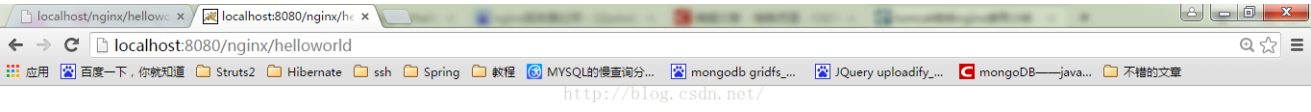
```
D:\nginx\nginx-1.10.1>nginx -s reload
D:\nginx\nginx-1.10.1> http://blog.csdn.net/
```

(5) 发出请求，获得Json，url显示依然是80端口的资源，即我们说的反向代理的特点，掩饰url，效果如下图所示：



{"text":"HelloWorld"}

事实上，nginx是将请求转发到Tomcat服务器，是8080端口下的资源，如下图所示：



{"text":"HelloWorld"}

(6) 如果不光有Tomcat服务器的资源，那么就需要定义多个location，比如，jsp资源请求就转发到Tomcat服务器下，php、html、js、css等资源资源可以转到Apache服务器目录下，如下图配置示例：

```
[html]
1. location ~ /\.jsp$ {
2.     proxy_pass http://localhost:8080;
3. }
4.
5. location ~ \.(html|js|css|png|gif)$ {
6.     root D:/software/developerTools/server/apache-tomcat-7.0.8/webapps/ROOT;
7. }
```

nginx的功能远不止于此，它的高效也同样被称道，有兴趣可以更深入了解！

版权声明：本文为博主原创文章，未经博主允许不得转载。https://blog.csdn.net/zhangliangzi/article/details/52143358

文章标签： nginx 前端开发 模板引擎

个人分类： Nginx

查看更多>>

想对作者说点什么？ 我来说一句

Xanthuim 2018-01-22 10:43:23 #2楼
看了你的文章好笑，完全是瞎扯