

背景介绍

最近在个人的多个项目部署中遇到这样一个问题，一个域名如何实现多个项目的访问。因为不想自己单独去申请域名证书和域名配置，便想到了这个方案，结合Nginx的location功能实现了自己的需求，便记录下来。示例中是以PHP的项目演示，其他的语言类似同样的方式进行部署。例如node的项目，可以在location中做一个验证，然后使用 `proxy_pass` 反向代理模块实现。

location模块的匹配介绍

- 1."="前缀指令匹配，如果匹配成功，则停止其他匹配。
- 2.普通字符串指令匹配，顺序是从长到短，匹配成功的location如果使用`^~`，则停止其他匹配（正则匹配）。
- 3.正则表达式指令匹配，按照配置文件里的顺序，成功就停止其他匹配。
- 4.如果第三步中有匹配成功，则使用该结果，否则使用第二步结果。

注意点

- 1.匹配的顺序是先匹配普通字符串，然后再匹配正则表达式。另外普通字符串匹配顺序是根据配置中字符长度从长到短，也就是说使用普通字符串配置的location顺序是无关紧要的，反正最后nginx会根据配置的长短来进行匹配，但是需要注意的是正则表达式按照配置文件里的顺序测试。找到第一个匹配的正则表达式将停止搜索。
- 2.一般情况下，匹配成功了普通字符串location后还会进行正则表达式location匹配。有两种方法改变这种行为，其一就是使用“=”前缀，这时执行的是严格匹配，并且匹配成功后立即停止其他匹配，同时处理这个请求；另外一种就是使用“`^~`”前缀，如果把这个前缀用于一个常规字符串那么告诉nginx 如果路径匹配那么不测试正则表达式。

```
location = /uri
```

=开头表示精确匹配，只有完全匹配上才能生效。

```
location ^~ /uri
```

`^~` 开头对URL路径进行前缀匹配，并且在正则之前。

```
location ~ pattern
```

`~`开头表示区分大小写的正则匹配。

```
location ~* pattern
```

~*开头表示不区分大小写的正则匹配。

```
location /uri
```

不带任何修饰符，也表示前缀匹配，但是在正则匹配之后。

```
location /
```

通用匹配，任何未匹配到其它location的请求都会匹配到，相当于switch中的default。

配置实例

```

server {
    listen      80;
    server_name test.com;
    index index.html index.htm index.php;
    charset koi8-r;
    access_log /var/log/nginx/host.access.log main;

    # 域名+项目1名称
    location ^~ /a1/ {
        alias /usr/share/nginx/html/a1/public;
    }

    # 域名+项目2名称
    location ^~ /a2/ {
        alias /usr/share/nginx/html/a2/public;
    }

    error_page 404 /404.html;

    # redirect server error pages to the static page /50x.html

    error_page 500 502 503 504 /50x.html;
    location = /50x.html {
        root /usr/share/nginx/html/500.html;
    }

    #pass the PHP scripts to FastCGI server listening on 127.0.0.1:9000

    location ~ \.php$ {
        root html;
        fastcgi_pass 127.0.0.1:9000;
        fastcgi_index index.php;
        fastcgi_param SCRIPT_FILENAME /scripts$fastcgi_script_name;
        include fastcgi_params;
    }

    location ~ /\.ht {
        deny all;
    }
}

```

效果预览

1.访问a1项目



2.访问a2项目

