1. Location 基础知识：

1.1 概念：

我们可以通过配置Location指令块，来决定客户端发过来的请求URI如何处理。

1.2 语法：

```
Syntax: location [ = | ~ | ~* | ^~ ] uri { ... }
location @name { ... }
Default:    —
Context:    server, location
```

location 配置可以有两种配置方法，可以在server指令块和location指令块配置。

1、修饰符 + uri（资源路径）

2、@ + name

1.2.1 修饰符：

```
 =   ：精确匹配（必须全部相等）
 ~   ：大小写敏感（正则表达式）
 ~*  ：忽略大小写（正则表达式），这里要注意忽略大小写的意思是请求的字符大小写都可以，
 但是不会进行大小转换，请求的大小写对应的文件必须存在。
 ^~  ：只需匹配uri部分
 @   ：内部服务跳转
```

2. Location 配置实例：

2.1 精准匹配：

=，精准匹配，一般是匹配某个具体文件。

```
location = /index.html {
    [ configuration ]
}
# 则匹配到`http://www.lutixia.com/index.html`这种请求。
```

2.2 大小写敏感匹配：

~，大小写敏感（正则表达式）。

```
    location ~ /LUTIXIA/ {
         [ configuration ]
    }
    #请求示例
    #http://www.lutixia.com/LUTIXIA/  [成功]
    #http://www.lutixia.com/lutixia/  [失败]
```

2.3 大小写不敏感匹配：

~*，大小写忽略（正则表达式）。

```
location ~* /lutixia.html {
           [ configuration ]
}
# 则会忽略 uri 部分的大小写
#http://www.lutixia.com/lutixia.html  [成功] 可以成功匹配，但是目录中要lutixia.html文件
#http://www.lutixia.com/LUTIXIA.html  [成功] 可以成功匹配，但是目录中要LUTIXIA.html文件
```

2.4 指定后缀匹配：

匹配以gif、jpg、jpeg结尾的文件

```
location ~* \.(gif|jpg|jpeg)$ {
    [ configuration ]
}
#http://www.lutixia.com/img/lutixia.jpg [成功]
```

2.5 忽略正则匹配：

^~，只匹配以 uri 开头，匹配成功以后，会停止搜索后面的正则表达式匹配。

```
location ^~ /img/ {
        [ configuration ]
}
#以 /img/ 开头的请求，都会匹配上
#http://www.lutixia.com/img/lutixia.jpg    [成功]
#http://www.lutixia.com/img/lutixia.png [成功]
```

如果配置了2.5，那么所有请求 /img/ 下的图片会被上面的处理，因为 ^~ 指令匹配到了，则不检查正则表达式。对比这两个location，可以**设置不同目录，相同文件进行实验。**

3. Location优先级：

完整范例：

```
这里有一简短的localtion配置：
location / {
            echo "this is $request_uri";
}
location ~* \.(jpg|png) {
            echo "this is ~* \.(jpg|png)";
}
location ~ \.(jpg|png) {
             echo "this is ~ \.(jpg|png)";
}

location ^~ /img/jfedu.jpg {
       echo  "this is ^~ /img/jfedu.jpg";
}
location = /img/jfedu.jpg {
       echo  "this is = /img/jfedu.jpg";
}
```

如果客户端的请求是：

```
curl 192.168.75.131/img/jfedu.jpg
```

那么按照匹配规则顺序应该是这样的：

第一步：取出uri：/img/jfedu.jpg

第二步：去匹配localtion规则，查找有没有 = /img/jfedu.jpg 的规则，有则停止匹配。

```
[root@localhost ~]# curl 192.168.75.131/img/jfedu.jpg
this is = /img/jfedu.jpg
```

第三步：将location = /img/jfedu.jpg 规则注释，继续查找有没有 ^~ /img/ 的规则，

```
[root@www ~]# curl 192.168.0.116/img/jfedu.jpg
this is ^~ /img/jfedu.jpg
```

第四步：将 location ^~ /img/注释，这是它会去查找有没有正则匹配规则。

```
location / {
            echo "this is $request_uri";
}
location ~* \.(jpg|png)$ {
            echo "this is ~* \.(jpg|png)";
}
location ~ \.(jpg|png)$ {
             echo "this is ~ \.(jpg|png)";
}

#location ^~ /img/jfedu.jpg {
#       echo  "this is ^~ /img/jfedu.jpg";
#}
#location = /img/jfedu.jpg {
#       echo  "this is = /img/jfedu.jpg";
#}

其中，第二个和的第三个规则都是正则，这时会按照至上而下的顺序匹配。
[root@localhost ~]# curl 192.168.75.131/img/jfedu.jpg
this is ~* \.(jpg|png)
```

第五步：其他的都注释后，因为优先匹配规则都没有找到，最后匹配到/img/规则。

```
[root@localhost ~]# curl 192.168.75.131/img/jfedu.jpg
this is /img/jfedu.jpg
```

4. rewrite规则：

Nginx的rewrite功能需要pcre软件的支持，即通过perl兼容正则表达式语句进行规则匹配的。

默认参数编译nginx就会支持rewrite的模块，但是也必须要pcre的支持。

rewrite是实现URL重写的关键指令，根据regex（正则表达式）部分内容，重定向到replacement，结尾是flag标记。

4.1 rewrite语法：

```
rewrite     <regex>     <replacement>     [flag];

            正则          替代内容           flag标记

正则：perl兼容正则表达式语句进行规则匹配

替代内容：将正则匹配的内容替换成replacement

flag标记：rewrite支持的flag标记

# flag标记说明：

last   #本条规则匹配完成终止当前location的规则，继续向下匹配新的location URI规则

break   #本条规则匹配完成即终止，不再匹配后面的任何规则

redirect   #返回302临时重定向，浏览器地址会显示跳转后的URL地址，关闭服务，无法重定向。

permanent   #返回301永久重定向，浏览器地址栏会显示跳转后的URL地址，关闭服务，依然可以重定向，清除缓存失效。
```

5. rewrite实例：

5.1 实现域名跳转：

```
方案一：
server {
        listen 80;
        server_name lutixia.com;
        rewrite ^/(.*)$ http://www.lutixia.com/$1
permanent;
}


server {
        listen 80;
        server_name www.lutixia.com;
        location / {
                root /data/www/;
                index index.html index.htm;
        }
}

方案二：
server {
        listen 80;
        server_name www.lutixia.com lutixia.com;
        if ( $host != 'www.lutixia.com' ) {
                rewrite ^/(.*)$ http://www.lutixia.com/$1
permanent;
                }
        location / {
                root /data/www/;
                index index.html index.htm;
        }
}

ps：本地需要做hosts解析
192.168.75.130   www.lutixia.com  lutixia.com
```

5.2 实现不同终端跳转：

```
server {
                listen 80;
```

```
                server_name  lutixia.com www.lutixia.com;
                root /usr/share/nginx/html/test;
                if ( $http_user_agent ~* "iphone|android"
) {
                        rewrite ^/(.*)$
http://m.lutixia.com/$1;
                }
                index index.html;
}
server {
                listen 80;
                server_name m.lutixia.com;
                root /data/www/m;
                index index.html;
                location / {
                        default_type text/html;
                        return 200 "this is iphone|android
html";
                }
}
```

5.3 实现浏览器的语言跳转：

```
# 根据浏览器的语言跳转到指定url:
server {
                listen 80;
                server_name  lutixia.com www.lutixia.com;
                root /usr/share/nginx/html/test;
                index index.html;

              location / {
                if ( $http_accept_language ~ "^zh-CN" ) {
                        rewrite ^/(.*)  /zh/$1 ;
                        }
                if ( $http_accept_language ~ "^en" ) {
                        rewrite ^/(.*)  /en/$1 ;
                        }
                root   html;
                index  index.html index.htm;
        }
        location ^~ /zh/ {
```

```
                root   html/;
                index index.html;
                }
        location ^~ /en/ {
                root html/;
                index index.html;
                }
        }

mkdir -p /usr/share/nginx/html/test/zh
mkdir -p /usr/share/nginx/html/test/en

echo "this is 中文  " >
/usr/share/nginx/html/test/zh/index.html
echo "this is English " >
/usr/share/nginx/html/test/en/index.html
```

5.4 实现错误页面返回首页：

```
error_page 404 =200 /index.html;
```

5.5 实现错误页面返回腾讯公益页面：

```
error_page 404 =200 /404.html;

vim /usr/local/nginx/html/404.html

<!DOCTYPE HTML>
<html>
<head>
   <meta charset="UTF-8" />
   <title>公益404</title>
</head>
<body>
<!--<script type="text/javascript"
src="http://www.qq.com/404/search_children.js"></script>--
>

<script type="text/javascript"
src="//qzonestyle.gtimg.cn/qzone/hybrid/app/404/search_chi
ldren.js" charset="utf-8"></script>
```

```
</body>
</html>
```