1.rewrite的介绍

nginx的rewrite规则就是使用正则匹配请求的url，然后根据定义的规则进行重写和改变，需ngx_http_rewrite_module模块来支持url重写功能，该模块是标准模块，默认已经安装。

url和uri的区别：

URI：Universal Resource Identifier ，通用资源标识符，用于对网络中的各种资源进行标识，由存放资源的主机名、片段标志符和相对的URI三部分组成。存放资源的主机名一般由传输协议（Scheme）、主机和资源路径三部分组成；片段标识符指向资源内容的具体元素、相对URI表示资源在主机上的相对路径。一般格式为：Scheme://[用户名][:密码]@主机名[:端口号][/资源路径]

URL：Uniform Resource Location，统一资源定位符，是用于在Internet中描述资源的字符串，是URI的子集，主要包括传输协议（Scheme）、主机（IP、端口号或者域名）和资源集体地址（目录或文件名）等三部分，一般格式为：scheme://主机名[:端口号]/[资源路径]

2.rewrite涉及的指令

执行顺序：

1.执行server块的rewrite指令(这里的块指的是server关键字后{}包围的区域，其它xx块类似)

2.执行location匹配

3.执行选定的location中的rewrite指令

如果其中某步URI被重写，则重新循环执行1-3，直到找到真实存在的文件

如果循环超过10次，则返回500 Internal Server Error错误

1）if指令

语法：if(condition){...}

默认值：无

作用域：server,location

对给定的条件condition进行判断。如果为真，大括号内的rewrite指令将被执行。

if条件(conditon)可以是如下任何内容：

一个变量名；false如果这个变量是空字符串或者以0开始的字符串；
使用= ,!= 比较的一个变量和字符串，true/false
使用~， ~*与正则表达式匹配的变量，如果这个正则表达式中包含右花括号}或者分号;则必须给整个正则表达式加引号
使用-f ， !-f 检查一个文件是否存在
使用-d, !-d 检查一个目录是否存在
使用-e ， !-e 检查一个文件、目录、符号链接是否存在
使用-x ， !-x 检查一个文件是否可执行
if指令实例

```
 if ($http_user_agent ~ MSIE) {
    rewrite ^(.*)$ /msie/$1 break;
 }
 if ($http_cookie ~* "id=([^;]+)(?:;|$)") {
    set $id $1;
 }
 if ($request_method = POST) {
    return 405;
 }
 if ($slow) {
    limit_rate 10k;
 }
```

2）return指令
用于完成对请求的处理，直接给客户端返回状态码，改指令后所有的nginx配置都是无效的，
语法：return code;
return code URL;
return URL;
默认值：无
作用域：server,location,if
3）set指令

语法：set variable value;
默认值：none
作用域：server,location,if

定义一个变量并赋值，值可以是文本，变量或者文本变量混合体。

4）uninitialized_variable_warn指令
语法：uninitialized_variable_warn on | off;
默认值：uninitialized_variable_warn on
作用域：http,server,location,if
控制是否输出为初始化的变量到日志
5）rewrite指令
该指令通过正则来改变url，可以同时存在一个或者多个指令
语法：rewrite regex replacement [flag];
默认值：无
作用域：server,location,if
regex ： 用于匹配uri的正则表达式。使用括号（）标记要截取的内容
replacement 匹配成功后用于替换uri中被截取内容的字符串，默认情况下，如果该字符串是由http://或者https://开头的，则不会继续向下对uri进行其他处理，而是直接将重写后的uri返回给客户端
flag 用来设置rewrite对uri的处理行为，常用的有
last 停止处理后续rewrite指令集，然后对当前重写的新URI在rewrite指令集上重新查找。
break 停止处理后续rewrite指令集，并不在重新查找,但是当前location内剩余非rewrite语句和location外的的非rewrite语句可以执行。
redirect 如果replacement不是以http:// 或https://开始，返回302临时重定向
permant 返回301永久重定向
补充：last和break标记的区别在于，last标记在本条rewrite规则执行完后，会对其所在的server { … } 标签重新发起请求，而break标记则在本条规则匹配完成后，停止匹配，不再做后续的匹配。另外有些时候必须使用last，比如在使用alias指令时，而 使用proxy_pass指令时则必须使用break。
注意：rewrite 规则优先级要高于location，在nginx配置文件中，nginx会先用rewrite来处理url，最后再用处理后的url匹配location
6）常用的变量
$args ： #这个变量等于请求行中的参数，同$query_string
$content_length ： 请求头中的Content-length字段。
$content_type ： 请求头中的Content-Type字段。

$document_root： 当前请求在root指令中指定的值。
$host： 请求主机头字段，否则为服务器名称。
$http_user_agent： 客户端agent信息
$http_cookie： 客户端cookie信息
$limit_rate： 这个变量可以限制连接速率。
$request_method： 客户端请求的动作，通常为GET或POST。
$remote_addr： 客户端的IP地址。
$remote_port： 客户端的端口。
$remote_user： 已经经过Auth Basic Module验证的用户名。
$request_filename： 当前请求的文件路径，由root或alias指令与URI请求生成。
$scheme： HTTP方法（如http，https）。
$server_protocol： 请求使用的协议，通常是HTTP/1.0或HTTP/1.1。
$server_addr： 服务器地址，在完成一次系统调用后可以确定这个值。
$server_name： 服务器名称。
$server_port： 请求到达服务器的端口号。
$request_uri： 包含请求参数的原始URI，不包含主机名，
如：”/foo/bar.php?arg=baz”。
$uri： 不带请求参数的当前URI，$uri不包含主机名，如”/foo/bar.html”。
$document_uri： 与$uri相同。
7）常用正则：
.： 匹配除换行符以外的任意字符
?： 重复0次或1次
+： 重复1次或更多次
*： 重复0次或更多次
\d：匹配数字
^： 匹配字符串的开始
$： 匹配字符串的介绍
{n}： 重复n次
{n,}： 重复n次或更多次
[c]： 匹配单个字符c
[a-z]： 匹配a-z小写字母的任意一个
小括号()之间匹配的内容，可以在后面通过$1来引用，$2表示的是前面第二个()里的内容。正则里面容易让人困惑的是\转义特殊字符。