预备--nginx 安装 echo 模块

本堂课将要使用第三方模块 ngx_echo 的功能，请重新配置添加到 nginx 插件中

##下载第三方模块
wget https://github.com/openresty/echo-nginx-module/archive/v0.61.tar.gz
tar -zxvf v0.61.tar.gz　　##解压
cd nginx-1.15.8　　##进入 nginx 源码目录，准备重新配置 nginx

##配置，--add-module 指向模块目录即会安装插件到 nginx 中
./configure --add-module=/usr/local/src/echo-nginx-module-0.61/

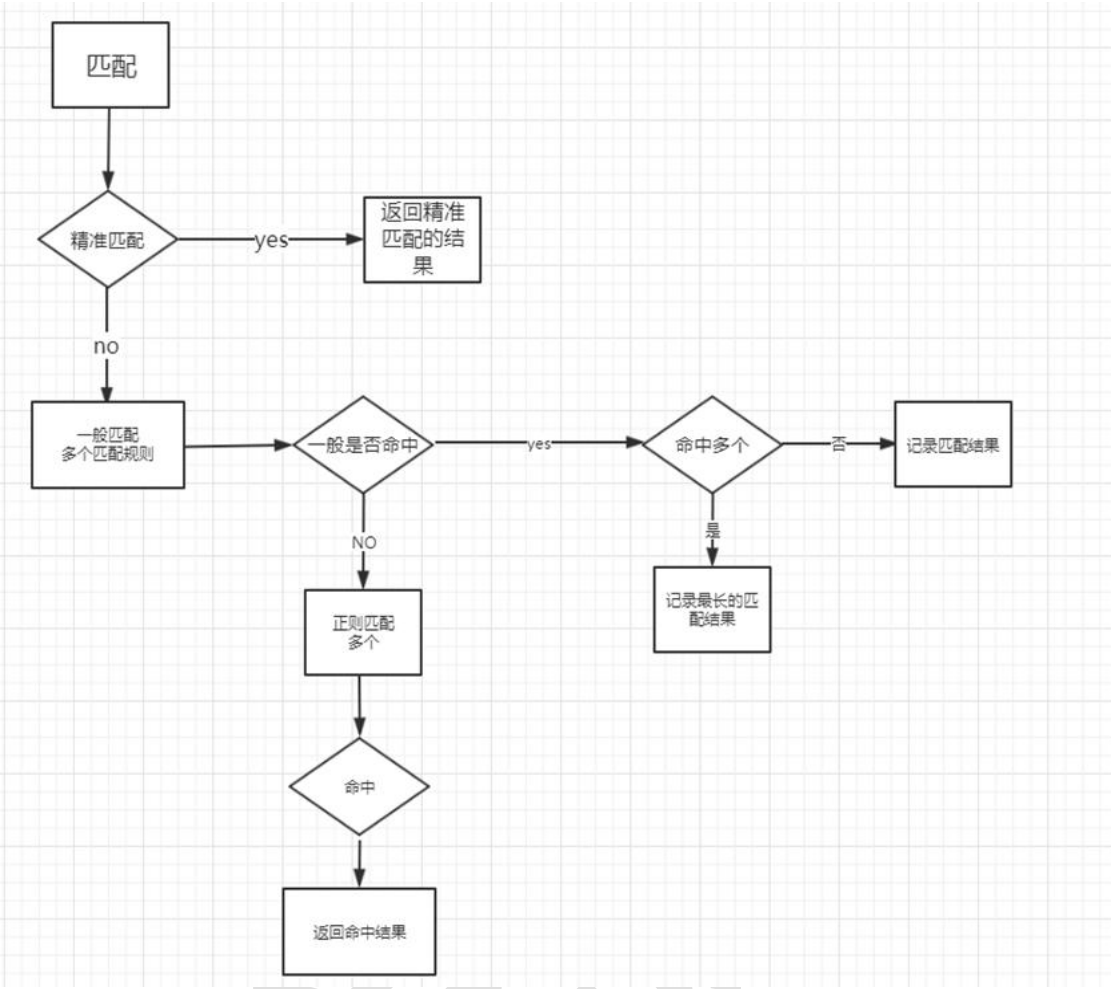
一、路由--Location 的使用

1、Location 语法规则

语法规则： location [=|~|~*|^~] /uri/ {… }
首先匹配 =，其次匹配^~, 其次是按文件中顺序的正则匹配，最后是交给 /通用匹配。当有匹配成功时候，停止匹配，按当前匹配规则处理请求。

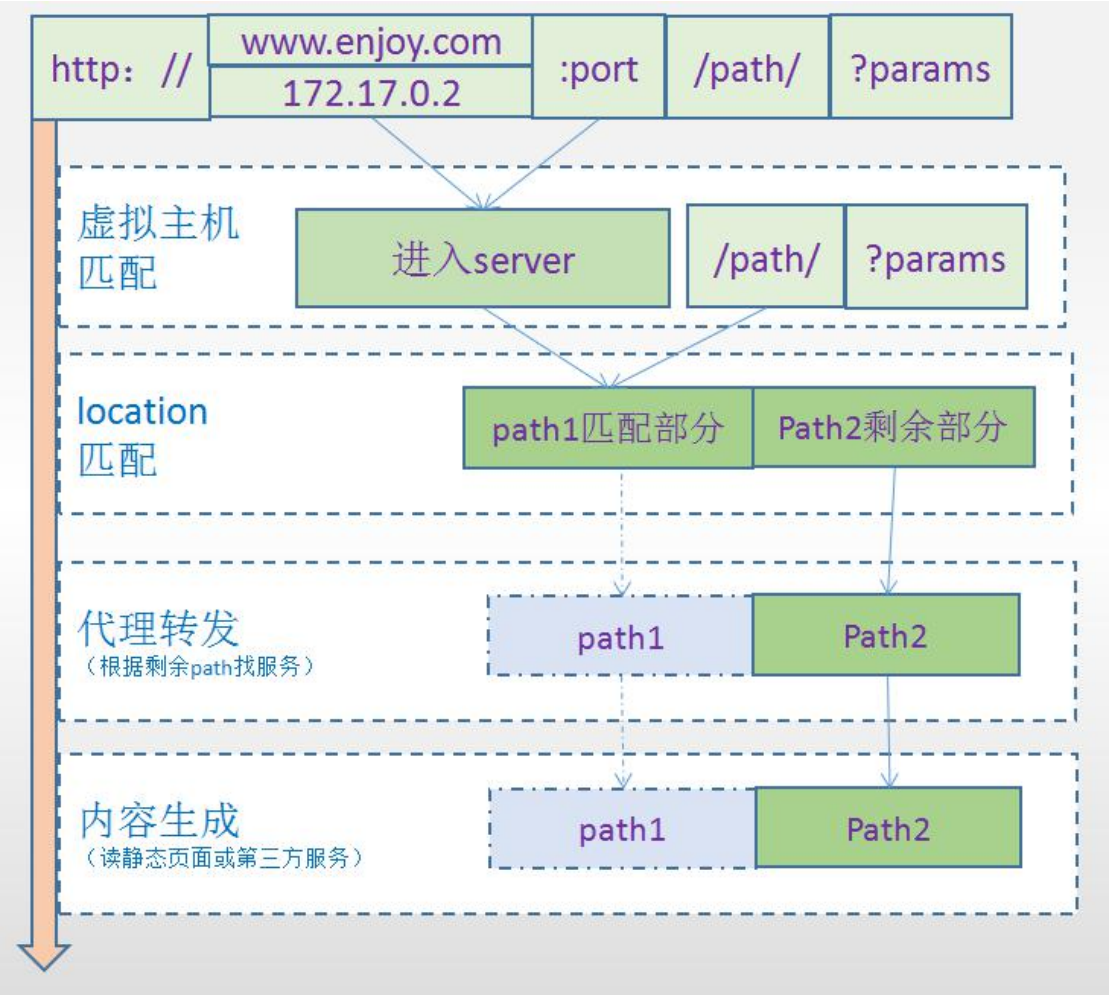
符号	含义
=	= 开头表示精确匹配
^~	^~开头表示 uri 以某个常规字符串开头，理解为匹配 url 路径即可（禁止正则匹配）。
~	~ 开头表示区分大小写的正则匹配
~*	~* 开头表示不区分大小写的正则匹配
!~和!~*	!~和!~*分别为区分大小写不匹配及不区分大小写不匹配的正则
/	用户所使用的代理（一般为浏览器）

匹配规则优先级如下：



- =精准匹配命中时，停止 location 动作，直接走精准匹配，
- 一般匹配（含非正则）命中时，先收集所有的普通匹配，最后对比出最长的那一条
- 如果最长的那一条普通匹配声明为非正则，直接此条匹配，停止 location
- 如果最长的那一条普通匹配不是非正则，继续往下走正则 location
- 按代码顺序执行正则匹配，当第一条正则 location 命中时，停止 location

2、path 匹配过程



假设 http 请求路径为
http://192.168.0.132:8088/mvc/index?id=2 ，匹配过程如下：

- 将整个 url 拆解为域名/端口/path/params
- 先由域名/端口，对应到目标 server 虚拟主机
- path 部分参与 location 匹配，path = path1 匹配部分 + path2 剩余部分
- 进入 location 方法体内部流程。
- 若是静态文件处理，则进入目标目录查找文件：root 指令时找 path1+path2 对应的文件;alias 指令时找 path2 对应的文件
- 若是 proxy 代理，则形如 proxy_pass=ip:port 时转发 path1+path2 路径到 tomcat;形如 proxy_pass=ip:port/xxx 时转发 path2 路径到 tomcat。params 始终跟随转发。

二、rewrite 使用：

rewrite regex replacement [flag];

flag=【break/last/redirect/permanent】

- ✧ regex 是正则表达式
- ✧ replacement 是替换值，新值
- ✧ flag -- 后续处理标识

1、flag=break

发生 nginx 内部重定向，path 值被更新，rewrite 层面的命令会中断。原控制流程逻辑不变往下走

2、flag=last

发生 nginx 内部重定向，path 值被更新，rewrite 层面的命令会中断。控制流程刷新，重新进行整个 location 层的逻辑流程。

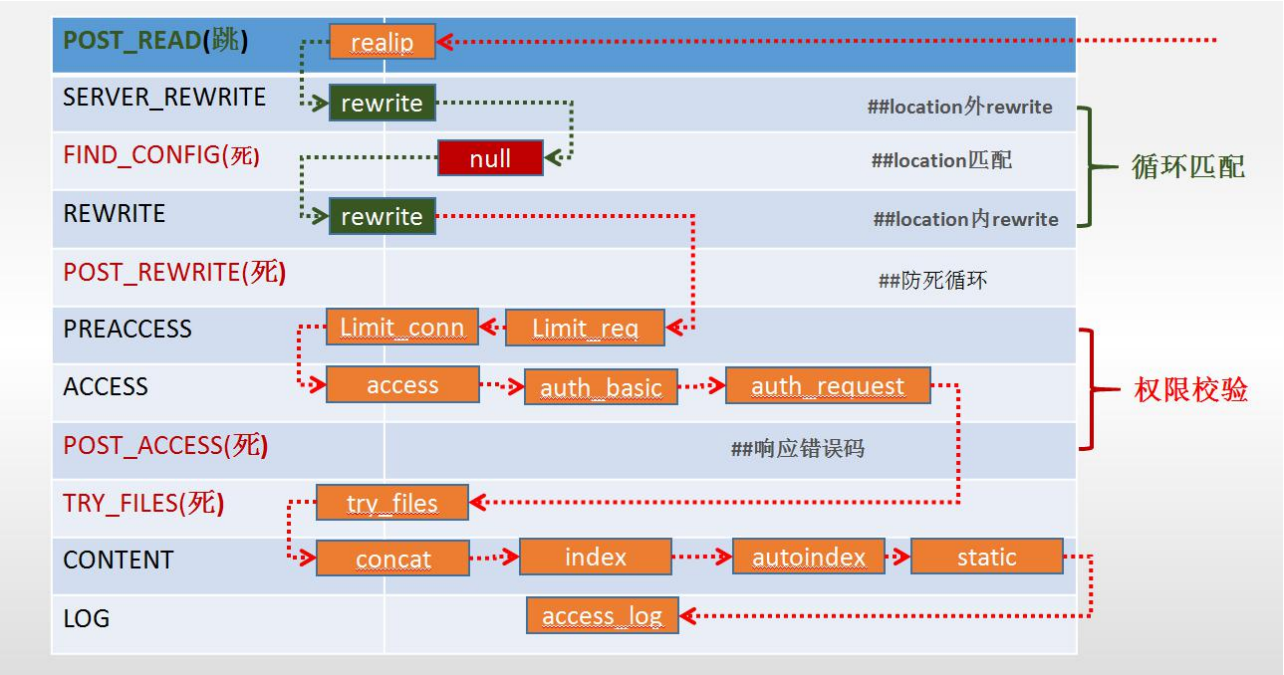
3、flag= redirect/permanent

发生页面重定向（301 永久重定向/302 临时重定向），nginx 流程结束，返回 http 响应到浏览器，页面 url 更新

4、flag 为空

发生 nginx 内部重定向，path 值被更新，rewrite 层面的命令继续。最后一个 rewrite 完毕，刷新控制流程，重新进行 location 重匹配

三、Nginx 处理请求的 11 个阶段



Nginx 处理请求的全过程一共划分为 11 个阶段（如图），按阶段由上到下依次执行 （上一阶段的所有指令执行完毕，才进入下一阶段）

各阶段的含义如下：

- post-read: 接收到完整的 http 头部后处理的阶段，在 uri 重写之前。一般跳过
- server-rewrite: location 匹配前，修改 uri 的阶段，用于重定向，location 块外的重写指令（**多次执行**）
- find-config: uri 寻找匹配的 location 块配置项（**多次执行**）
- rewrite: 找到 location 块后再修改 uri，location 级别的 uri 重写阶段（**多次执行**）
- post-rewrite: 防死循环，跳转到对应阶段
- preaccess: 权限预处理
- access: 判断是否允许这个请求进入
- post-access: 向用户发送拒绝服务的错误码，用来响应上一阶段的拒绝
- try-files: 访问静态文件资源
- content : 内容生成阶段，该阶段产生响应，并发送到客户端
- log: 记录访问日志