NGINX处理请求的几个阶段

bing edited this page on 20 Sep 2016 · [2 revisions](https://github.com/bingbo/blog/wiki/NGINX%E5%A4%84%E7%90%86%E8%AF%B7%E6%B1%82%E7%9A%84%E5%87%A0%E4%B8%AA%E9%98%B6%E6%AE%B5/_history)

**Pages 138**

[blog home](https://github.com/bingbo/blog)

**Clone this wiki locally**



[**Clone in Desktop**](https://desktop.github.com/)

读取完请求头后，nginx进入请求的处理阶段。简单的情况下，客户端发送过的统一资源定位符(url)对应服务器上某一路径上的资源，web服务器需要做的仅仅是将url映射到本地文件系统的路径，然后读取相应文件并返回给客户端。但这仅仅是最初的互联网的需求，而如今互联网出现了各种各样复杂的需求，要求web服务器能够处理诸如安全及权限控制，多媒体内容和动态网页等等问题。这些复杂的需求导致web服务器不再是一个短小的程序，而变成了一个必须经过仔细设计，模块化的系统。nginx良好的模块化特性体现在其对请求处理流程的多阶段划分当中，多阶段处理流程就好像一条流水线，一个nginx进程可以并发的处理处于不同阶段的多个请求。nginx允许开发者在处理流程的任意阶段注册模块，在启动阶段，nginx会把各个阶段注册的所有模块处理函数按序的组织成一条执行链。

nginx实际把请求处理流程划分为了11个阶段，这样划分的原因是将请求的执行逻辑细分，各阶段按照处理时机定义了清晰的执行语义，开发者可以很容易分辨自己需要开发的模块应该定义在什么阶段，其定义在http/ngx\_http\_core\_module.h中有定义：

**NGX\_HTTP\_POST\_READ\_PHASE:**

接收完请求头之后的第一个阶段，它位于uri重写之前，实际上很少有模块会注册在该阶段，默认的情况下，该阶段被跳过

**NGX\_HTTP\_SERVER\_REWRITE\_PHASE:**

server级别的uri重写阶段，也就是该阶段执行处于server块内，location块外的重写指令，在读取请求头的过程中nginx会根据host及端口找到对应的虚拟主机配置

**NGX\_HTTP\_FIND\_CONFIG\_PHASE:**

寻找location配置阶段，该阶段使用重写之后的uri来查找对应的location，值得注意的是该阶段可能会被执行多次，因为也可能有location级别的重写指令

**NGX\_HTTP\_REWRITE\_PHASE:**

location级别的uri重写阶段，该阶段执行location基本的重写指令，也可能会被执行多次

**NGX\_HTTP\_POST\_REWRITE\_PHASE:**

location级别重写的后一阶段，用来检查上阶段是否有uri重写，并根据结果跳转到合适的阶段

**NGX\_HTTP\_PREACCESS\_PHASE:**

访问权限控制的前一阶段，该阶段在权限控制阶段之前，一般也用于访问控制，比如限制访问频率，链接数等

**NGX\_HTTP\_ACCESS\_PHASE:**

访问权限控制阶段，比如基于ip黑白名单的权限控制，基于用户名密码的权限控制等

**NGX\_HTTP\_POST\_ACCESS\_PHASE:**

问权限控制的后一阶段，该阶段根据权限控制阶段的执行结果进行相应处理

**NGX\_HTTP\_TRY\_FILES\_PHASE:**

try\_files指令的处理阶段，如果没有配置try\_files指令，则该阶段被跳过

**NGX\_HTTP\_CONTENT\_PHASE:**

内容生成阶段，该阶段产生响应，并发送到客户端

**NGX\_HTTP\_LOG\_PHASE:**

日志记录阶段，该阶段记录访问日志

请求初始化处理是在http/ngx\_http.c里的ngx\_http\_block()函数里执行，然后会依次创建初始化各个阶段的配置(main\_conf,server\_conf,loc\_conf),初始化各个阶段的handler,ngx\_http\_init\_phase\_handlers()