**也来学习写一下nginx的自定义模块**

[February 19, 2012](http://zhwen.org/?p=554)  [helight](http://zhwen.org/?author=2)  [linux 应用](http://zhwen.org/?cat=7)

这几天在家里，学习了一下nginx的一些东西，首先当然就是用编译安装来配置简单的服务了。上一篇也算是学习配置的一点心得，在网上看了nginx的可扩展性后，我也简单研究了一下nginx的自定义模块。并且在晚上代码的参考下，也编译了两个模块来玩了一下，这里先展示一下一个最简单的模块。

这个模块主要是这样一个目的，在输入一个指定的url后，会返回自定义的内容：

url：http://127.0.0.1/hello

输出内容为我自定义的字符串，比如：”I try to fuck Nginx!!!”

这里介绍以下步骤：

第一步：

建立任意一个目录，比如：ngx\_hello

第二步：

在这个文件夹中建立一个config文件，内容如下：

ngx\_addon\_name=ngx\_xtest  #这里指定模块名称  
HTTP\_MODULES=”$HTTP\_MODULES ngx\_xtest” #指定编译后文件名称  
NGX\_ADDON\_SRCS=”$NGX\_ADDON\_SRCS $ngx\_addon\_dir/ngx\_xtest.c”  #指定源文件，这里我只有一个文件多个文件就要全部写上去

CORE\_LIBS=”$CORE\_LIBS ” #这里指定编译的nginx库，这变量是由编译nginx的时候Makefile传递过来的，上面的几个变量也是一样

第三步：

建立ngx\_xtest.c文件，内容如下，也写了简单的注释  
//引用nginx头文件，我们这个测试文件就一个没有其它的测试文件，所以也没有引用其它的头文件。  
#include <ngx\_config.h>  
#include <ngx\_core.h>  
#include <ngx\_http.h>

char\* ngx\_xtest\_setup(ngx\_conf\_t \*cf, ngx\_command\_t \*cmd, void \*conf);

u\_char ngx\_xtest\_string[] = “I try to FUCK Nginx !!!”;  
//这个事命令组，这里我们可以写多个命令，以数组的形式存在  
static ngx\_command\_t ngx\_xtest\_commands[] =  
{  
{   ngx\_string(“hello”),//这是命令字段，其实也就是我们在url中访问时写的目标文件  
NGX\_HTTP\_LOC\_CONF|NGX\_CONF\_NOARGS,  
ngx\_xtest\_setup,      //这个函数来响应hello这个命令，hello命令触发这个函数  
0,  
0,  
NULL  
},  
ngx\_null\_command  
};//这个数组必须以ngx\_null\_command结尾

//这个是用来处理配置文件的，这里我们使用系统的默认配置文件处理  
static ngx\_http\_module\_t ngx\_xtest\_ctx = {  
NULL,                          /\* preconfiguration \*/  
NULL,                          /\* postconfiguration \*/  
NULL,                          /\* create main configuration \*/  
NULL,                          /\* init main configuration \*/  
NULL,                          /\* create server configuration \*/  
NULL,                          /\* merge server configuration \*/  
NULL,                          /\* create location configuration \*/  
NULL                           /\* merge location configuration \*/  
};

//这个才是真正的模块。  
ngx\_module\_t ngx\_xtest = {  
NGX\_MODULE\_V1,  
&ngx\_xtest\_ctx,       /\* module context \*/  
ngx\_xtest\_commands,   /\* module directives \*/  
NGX\_HTTP\_MODULE,      /\* module type \*/  
NULL,                 /\* init master \*/  
NULL,                 /\* init module \*/  
NULL,                 /\* init process \*/  
NULL,                 /\* init thread \*/  
NULL,                 /\* exit thread \*/  
NULL,                 /\* exit process \*/  
NULL,                 /\* exit master \*/  
NGX\_MODULE\_V1\_PADDING  
};//实例化模块对象

//hello命令的真正处理函数，参数是ngx\_http\_request\_t即一个请求  
//此函数实现的也是基本处理流程  
static ngx\_int\_t ngx\_xtest\_handler(ngx\_http\_request\_t \*r){  
ngx\_int\_t    rc;  
ngx\_buf\_t   \*b;  
ngx\_chain\_t  out;

//填充HTTP头  
/\* set the ‘Content-type’ header \*/  
r->headers\_out.content\_type.len = sizeof(“text/html”) – 1;  
r->headers\_out.content\_type.data = (u\_char \*) “text/html”;

//分配输出内存空间  
/\* allocate a buffer \*/  
b = ngx\_pcalloc(r->pool, sizeof(ngx\_buf\_t));  
if (b == NULL) {  
return NGX\_HTTP\_INTERNAL\_SERVER\_ERROR;  
}  
//输出缓存附加到输出链表上  
/\* attach buffer to the buffer chain \*/  
out.buf = b;  
out.next = NULL;

//填写输出缓存内容  
/\* adjust the pointers of the buffer \*/  
b->pos = ngx\_xtest\_string;  /\* the begin offset of the buffer \*/  
b->last = ngx\_xtest\_string + sizeof(ngx\_xtest\_string) – 1; /\* the end offset of the buffer \*/  
b->memory = 1;    /\* this buffer is in memory \*/  
b->last\_buf = 1;  /\* this is the last buffer in the buffer chain \*/

/\* 设置http返回码 \*/  
r->headers\_out.status = NGX\_HTTP\_OK;  
r->headers\_out.content\_length\_n = sizeof(ngx\_xtest\_string) – 1;

//发送HTTP报头  
/\* send the headers of your response \*/  
rc = ngx\_http\_send\_header(r);

if (rc == NGX\_ERROR || rc > NGX\_OK || r->header\_only) {  
return rc;  
}  
//输出内容  
/\* send the buffer chain of your response \*/  
return ngx\_http\_output\_filter(r, &out);  
}

//实现hello命令的初始化函数，此函数指定命令的真正处理函数为ngx\_xtest\_handler  
char\* ngx\_xtest\_setup(ngx\_conf\_t \*cf, ngx\_command\_t \*cmd, void \*conf)  
{  
ngx\_http\_core\_loc\_conf\_t \*clcf;

clcf = ngx\_http\_conf\_get\_module\_loc\_conf(cf, ngx\_http\_core\_module);  
clcf->handler = ngx\_xtest\_handler; /\* handler to process the ‘hello’ directive \*/

return NGX\_CONF\_OK;  
}

第四步：编译安转

在编译时需要加入–add-module=/path

$./configure –prefix=/usr/local/nginx –add-module=/data/babycode/nginx/ngx\_xmodule/ –with-debug

$make

#make install

第五步：修改配置  
配置文件—nginx.conf

location /hello{

hello;

}  
第六步：运行测试  
#/usr/local/nginx/sbin/nginx  
测试

在浏览器中输入：http://127.0.0.1/hello

浏览器中会出现”I try to FUCK Nginx !!!”，则表示运行成功拉。。。

OK! 今天就先学习到这里，打完收工，睡觉拉。。。

No related posts.