目录

[第1章 基本使用 1](#_Toc6170)

[1.1编写配置文件 1](#_Toc15290)

[1.2 RewriteCond 1](#_Toc11215)

[1.3 RewriteRule 4](#_Toc17568)

* 1. 基本使用
     1. 编写配置文件

采用配置文件的方式，将事先编写好的重写规则写入文件中。放置A10安装目录中。位于 **${ApusicHome}/aas/domains/mydomain/config 目录下**

文件名为：**rewrite.config**

示例文件如下(鼠标右键保存文件至任意目录查看)：

****

* + 1. RewriteCond

格式：RewriteCond TestString CondPattern [flag]

1.RewriteCond 指令定义了一个规则条件。一个或多个 RewriteCond 指令可以优先于 RewriteRule 指令执行。如果 URI 当前状态匹配它的模式，并且满足了这些条件，才会使用下列规则。

2.TestString 是一种字符串，除了简单的文本之外，它还可以含有下列扩展结构。

 %{ NAME\_OF\_VARIABLE } 。%{ NAME\_OF\_VARIABLE } 的 *NAME\_OF\_VARIABLE* 是一种取自下面列表的字符串：

****HTTP 报头:****

HTTP\_USER\_AGENT

HTTP\_REFERER

HTTP\_COOKIE

HTTP\_FORWARDED

HTTP\_HOST

HTTP\_PROXY\_CONNECTION

HTTP\_ACCEPT

****连接与请求:****

REMOTE\_ADDR

REMOTE\_HOST

REMOTE\_PORT

REMOTE\_USER

REMOTE\_IDENT

REQUEST\_METHOD

SCRIPT\_FILENAME

REQUEST\_PATH

CONTEXT\_PATH

SERVLET\_PATH

PATH\_INFO

QUERY\_STRING

AUTH\_TYPE

****服务器内部:****

DOCUMENT\_ROOT

SERVER\_NAME

SERVER\_ADDR

SERVER\_PORT

SERVER\_PROTOCOL

SERVER\_SOFTWARE

****日期与时间:****

TIME\_YEAR

TIME\_MON

TIME\_DAY

TIME\_HOUR

TIME\_MIN

TIME\_SEC

TIME\_WDAY

****特殊字符串:****

THE\_REQUEST

REQUEST\_URI

REQUEST\_FILENAME

HTTPS

* REQUEST\_PATH（在示例中，应为 /find）。  
  对应用于映射的完整路径。
* CONTEXT\_PATH （在示例中，应为 /demo）。  
  对应映射的上下文的路径。
* SERVLET\_PATH （在示例中，应为 /find）。  
  对应 Servlet 路径。
* THE\_REQUEST  
  由浏览器发送给服务器的完整 HTTP 请求代码行（比如，GET /demo/find HTTP/1.1）。这并不包括任何由浏览器发送的额外报头。
* REQUEST\_URI  
  HTTP 请求代码行中所请求的资源（在上例中，应为为 /demo/find）。
* REQUEST\_FILENAME  
  与请求相匹配的文件或脚本的完整本地文件系统路径。
* HTTPS  
  当连接使用 SSL/TLS 时，含有文本 "on"，否则含有 "off"。

另外还需要注意的是:

1. %{ENV:variable}。其中的 variable 可以是任何 Java 系统属性。目前可以使用。
2. %{HTTP:header}。其中的 header 可以是任意的 HTTP MIME 报头名称。该变量常用来获取发送到 HTTP 请求的报头值。范例： %{HTTP:Proxy-Connection} 是 HTTP 报头 Proxy-Connection: 的值。

3.CondPattern即条件模式，是一种应用于 *TestString* 当前实例的正则表达式。TestString在匹配 *CondPattern* 之前，会首先求值。CondPattern是一种兼容 perl 并带有一些扩展的正则表达式。

4.可以将 [flag] 做为 RewriteCond 指令的第三个参数，为 CondPattern 设定标记。这里的 flag 是一个包含下列任一标记，且标记间由逗号分隔的列表：

* nocase|NC（不区分大小写）无论是扩展的 TestString 还是在 *CondPattern* 中，都不区分大小写（A-Z 和 a-z 是等同的）。该标记只有在比较 *TestString* 和 *CondPattern* 时才是有效的。对于文件系统和子请求（HTTP 请求）是无效的。
* ornext|OR（或者下一个条件）利用本地 OR（而不是隐式的 AND）来组合规则条件。典型范例为:

**RewriteCond %{REMOTE\_HOST} ^host1.\* [OR]**

**RewriteCond %{REMOTE\_HOST} ^host2.\* [OR]**

**RewriteCond %{REMOTE\_HOST} ^host3.\***

**RewriteRule ...some special stuff for any of these hosts...**

* + 1. RewriteRule

格式：RewriteRule Pattern Substitution [flag]

RewriteRule 指令是重写机制的核心。此指令可以多次使用，每个实例都定义一个单独的重写规则。这些规则的定义顺序尤为重要，因为这是在运行时应用它们的顺序。

模式是一个作用于当前 URL 的兼容 perl 的正则表达式，这里的“当前”是指该规则生效时的 URL，它可能与被请求的 URL 不同，因为其他规则可能在此之前已经发生匹配并对它做了改动。

Pattern是一个perl兼容的正则表达式。这个模式的比较对象取决于RewriteRule指令定义的位置。

Substitution重写规则的替换是替换Pattern匹配的原始url路径的字符串。

可以将Pattern替换为文件系统路径、url路径、绝对URL或者一个 **-** (表示应用URL路径不进行更改。)

[flags] 做为RewriteRule  的第三个参数，从而为替代字符串设置特殊标记。

*flags*是一个包含下列标记且标记间以逗号分隔的列表：

* chain|C （与下一个规则相链接）

此标记使当前规则与下一个规则（它又可以与其后规则相链接，如此反复）相链接。 它产生如下效果：如果某个规则被匹配，通常会继续处理其后继规则，这个标记就不起作用；如果某规则不被匹配，则其后继链接规将会被忽略。

* cookie|CO = ***NAME:VAL:domain[:lifetime[:path]]*** （设置 cookie）

在客户端浏览器上设置一个 cookie。cookie 的名称是 ***NAME***，其值是 ***VAL***。 *domain* 字段是该 cookie 的域，比如 .apache.org，可选的*lifetime* 是 cookie 生命周期（以分钟计），可选的 *path* 是 cookie 的路径。

* env|E = *VAR:VAL* （设置环境变量）

强制一个名为 *VAR* 的请求变量值为 *VAL*

* forbidden|F（强制禁止访问该 URL ）

强制禁止访问当前的 URL——立即反馈一个 HTTP 响应代码 403。

* gone|G（强制 URL 为已失效）

强制当前 URL 为已失效——立即反馈一个 HTTP 响应代码 410（请求资源已被删除）。

* host|H=Host（重写虚拟主机）

重写虚拟主机，而不是重写 URL。

* last|L（最后一个规则）

立即停止重写操作，并不再应用其他重写规则。它对应于Java 语言中的 break 命令。使用该标记可以防止当前已被重写的 URL 被后续规则所继续重写。

* next|N（重新执行）

重新执行重写操作（从第一个重写规则重新开始）。这时，要匹配的 URL 已不是原始 URL 了，而是经最后一个重写规则处理过的 URL。它对应于 Java 语言中的 continue 命令。此标记可以重新开始重写操作，立即回到循环的头部。[****但是要小心，不要制造死循环！****]

* nocase|NC（不区分大小写）

使模式不区分大小写。当模式与当前 URL 匹配时，A-Z 和 a-z 没有区别。

* noescape|NE（在输出中不对 URI 进行转义）

此标记阻止对重写结果应用常规的 URI 转义规则。一般情况下，特殊字符（如%、$、;等）都会被转义为十六进制值。此标记可以阻止这种转义，允许百分号等符号出现在输出中，如：

RewriteRule /foo/(.\*) /bar?arg=P1\%3d$1 [R,NE]

将 /foo/zed 转化为 /bar?arg=P1=zed 的安全请求。

* qsappend|QSA（附加查询串）

该标记会强制重写引擎将替代字符串中的一个查询字符串部分添加到已有字符串上，而不是简单地替换已有字符串。当你想要通过重写规则为查询字符串添加更多数据时，可以使用该标记。

* redirect|R****[=code]****（强制重定向）

强制执行一个外部重定向。如果没有指定 code，则产生一个HTTP 响应代码 302（暂时移动）。如果需要使用在 300-400 范围内的其他响应代码，只需在此指定这个数值即可。

* skip|S****=num****（忽略后续规则）

如果当前规则匹配，此标记会强制重写引擎跳过当前匹配规则后面的 num 个规则。

* type|T****=MIME-type****（强制指定 MIME 类型）

强制指定目标文件的 MIME 类型为 ****MIME-type****，可基于一些条件设置内容类型。

RewriteRule ^(.+\.php)s$ $1 [T=application/x-httpd-php-source]

