

Hosting WWW

Bezpieczeństwo hostingu WWW

Dr Michał Tanaś (<http://mtanas.home.amu.edu.pl/>)

Apache2 – moduły

- Moduły – fragmenty kodu, które dodają nowe funkcje do Apache2
- ... czyli odpowiednik pluginów po stronie serwera.
- Moduł może mieć swoje własne dyrektywy konfiguracyjne
- ... których można używać we wszystkich plikach konfiguracyjnych Apache
- ... pod warunkiem że moduł jest załadowany.
- Moduły są ładowane w momencie startu Apache.

Apache2 – mod_alias

- Moduł odpowiedzialny za zmiany adresów URL po stronie serwera
- ... przy użyciu mechanizmów protokołu HTTP (w odróżnieniu od mod_rewrite)
- Przeglądarka wie że serwer prosi o przekierowanie
- ... natomiast decyzja o przejściu na stronę docelową należy do przeglądarki

Apache2 – mod_alias

- Dyrektywa Alias
 - Ogólna postać
Alias URL ścieżka
 - Powoduje że odwołania do danego URL'a będą w rzeczywistości kierowane do ścieżki
 - Ścieżka może być plikiem albo katalogiem
 - Natomiast sam URL w ogóle nie musi istnieć na serwerze
 - Przykład
Alias /download /opt/ftp/

Apache2 – mod_alias

- Dyrektywa AliasMatch
 - Ogólna postać
AliasMatch regex ścieżka
 - Wersja Alias w której można stosować wyrażenia regularne (regexp)
 - Przykład
AliasMatch (.*)\.zip /opt/ftp/

Apache2 – mod_alias

- Uwaga! Wyglądające tak samo Alias i AliasMatch mogą działać inaczej!
- Przykład
 - Alias /download/ /opt/ftp/
 - Przekierowuje URL'e typu `http://server/download/cośtam` do `/opt/ftp/`
 - AliasMatch /download/ /opt/ftp/
 - Przekierowuje URL'e zawierające w dowolnym miejscu tekst „/download/” do `/opt/ftp/`
 - np. takie `http://server/katalog1/download/katalog2/`

Apache2 – mod_alias

- Dyrektywa ScriptAlias
 - Ogólna postać
ScriptAlias URL ścieżka
 - Powoduje że odwołania do danego URL'a będą w rzeczywistości kierowane do ścieżki
 - ... a jednocześnie odwołanie zostanie potraktowane jako odwołanie do skryptu CGI a nie do zwykłej strony
 - Przykład
ScriptAlias /cgi-bin/ /opt/cgi/

Apache2 – mod_alias

- Dyrektywa ScriptAlias
 - ScriptAlias /cgi-bin/ /opt/cgi/ jest odpowiednikiem
Alias /cgi-bin/ /opt/cgi/
<Directory /opt/cgi/>
SetHandler cgi-script
Options +ExecCGI
</Directory>

Apache2 – mod_alias

- Dyrektywa ScriptAliasMatch
 - Ogólna postać
ScriptAliasMatch regex ścieżka
- Wersja ScriptAlias w której można stosować wyrażenia regularne (regexp)

Apache2 – mod_alias

- Dyrektywa Redirect
 - Ogólna postać
Redirect staryURL nowyURL
Redirect kod staryURL nowyURL
 - W przypadku gdy przeglądarka zażąda dostępu do adresu staryURL
 - ... serwer wysyła odpowiedź HTTP z prośbą o przekierowanie
 - ... opcjonalnie z podanym kodem odpowiedzi HTTP
 - Domyślny kod to 302 „Moved temporarily”

Apache2 – mod_alias

- Dyrektywa Redirect
 - Przeglądarka wie że została przekierowana
 - Decyzja czy zostanie wczytany nowy URL należy do przeglądarki
 - Nowy URL pojawi się w pasku adresu przeglądarki

Apache2 – mod_alias

- Dyrektywa Redirect
 - Wyspecjalizowane wersje
 - RedirectMatch – wersja z wyrażeniami regularnymi (regexp)
 - RedirectPermanent – zgłasza kod odpowiedzi 301 „Moved permanently”
 - RedirectTemp – równoważna Redirect
 -

Apache2 – mod_alias

- Dyrektywa Redirect
 - Jako kod odpowiedzi można podać dowolny kod odpowiedzi HTTP
 - ... niekoniecznie 3xx (związany z przekierowaniami)
 - ... ale wtedy nie podaje się nowego URL'a!
 - Przykład
Redirect 410 http://server/skasowanyplik

Apache2 – mod_rewrite

- Moduł odpowiedzialny za zmiany adresów URL po stronie serwera
- ... poprzez bezpośrednie zastępowanie fragmentów tekstu w adresie URL po stronie serwera
- Decyzja o przekierowaniu należy do serwera
- ... a przeglądarka nawet nie wie że została przekierowana
- ... ponieważ serwer staje się niewidzialnym proxy.

Apache2 – mod_rewrite

- RewriteEngine on|off
 - Włącza/wyłącza możliwość przepisywania URL'i dla danego VirtualHost'a lub katalogu
 - Musi być użyta jako pierwsza dyrektywa z mod_rewrite w danym bloku
 - W przypadku wyłączenia RewriteEngine inne dyrektywy z mod_rewrite są ignorowane
 - ... tzn. nie działają ale nie powodują błędu „500 Internal Server Error”

Apache2 – mod_rewrite

- RewriteRule staryURL nowyURL [flagi]
 - Jeżeli URL pasuje do wzorca staryURL
 - ... to serwer zastąpi go przez nowyURL
 - staryURL musi być w formacie wyrażeń regularnych (regex)

Apache2 – mod_rewrite

- RewriteRule staryURL nowyURL [flagi]
 - nowyURL może być:
 - Ścieżką absolutną w systemie plików serwera
 - Ścieżką względem DocumentRoot
 - Innym URL'em (może być względem VirtualHost'a)
 - Znakiem „-” jeżeli przepisywanie URL'a nie ma sensu (bo np. rezultatem RewriteRule ma być sztucznie wygenerowany błąd)

Apache2 – mod_rewrite

- RewriteRule staryURL nowyURL [flagi]
 - Przykład
 - RewriteRule „^/games” „/usr/local/games/www”
 - Spowoduje że adres `http://serwer/games/saper.php`
 - ... będzie prowadził do pliku `/usr/local/games/www/saper.php`

Apache2 – mod_rewrite

- RewriteRule staryURL nowyURL [flagi]
 - nowyURL może dodatkowo zawierać wyrażenia
 - \$n zostanie zastąpione przez n-ty nawias () we wzorcu staryURL
 - %n zostanie zastąpione przez n-ty spełniony warunek w poprzedzających RewriteCond
 - %{var} zostanie zastąpione przez zmienną serwera

Apache2 – mod_rewrite

- RewriteRule staryURL nowyURL [flagi]
 - Przykład
 - RewriteRule „^/page/([^\^.]+)/?\$” „index.php?page=\$1”
 - Spowoduje że adres `http://serwer/page/1`
 - ... zostanie zastąpiony przez adres `http://serwer/index.php?page=1`
 - Również adres `http://serwer/page/1/`
 - ... zostanie zastąpiony przez adres `http://serwer/index.php?page=1`

Apache2 – mod_rewrite

- RewriteRule staryURL nowyURL [flagi]
 - RewriteRule może być użyte więcej niż jeden raz w tym samym bloku Directory albo VirtualHost
 - ... wtedy sprawdzane są wszystkie po kolei
 - ... a każde kolejne RewriteRule otrzymuje na wejściu:
 - wynik ostatniego poprzedniego RewriteRule które zadziałało (regex pasował)
 - wejściowy URL jeżeli żadne z poprzednich RewriteRule nie zadziałało

Apache2 – mod_rewrite

- RewriteRule staryURL nowyURL [flagi]
- Najważniejsze flagi:
 - C – „chain”, kolejne RewriteRule zostaną wykonane tylko jeżeli RewriteRule z flagą C została wykonana (regex pasował)
 - F – „forbidden”, jeżeli regex pasował serwer zwraca błąd „403 Forbidden”
 - G – „gone”, jeżeli regex pasował serwer zwraca błąd „410 Gone”

Apache2 – mod_rewrite

- RewriteRule staryURL nowyURL [flagi]
 - Najważniejsze flagi:
 - L - „last”, jeżeli regex pasował to żadne kolejne RewriteRule nie zostaną wykonane
 - N - „next”, rozpoczyna przetwarzanie przepisywanie URLa od pierwszej RewriteRule, ale jako tekst wejściowy przyjmując wynik bieżącej RewriteRule
 - PT - „passthrough”, rozpoczyna przetwarzanie konfiguracji VirtualHosta od początku, ale jako wejściowy URL przyjmując wynik bieżącej RewriteRule. Używany gdy dyrektywa Alias ma działać z wynikiem przepisywania URLa

Apache2 – mod_rewrite

- RewriteRule staryURL nowyURL [flagi]
- Najważniejsze flagi:
 - S=n - „skip”, jeżeli regex pasuje serwer pominie kolejnych n RewriteRule
 - NC - „nocase”, przy porównaniu wzorców nie rozróżnia małych i dużych liter
 - NE - „noescape”, zapobiega przekształceniu znaków specjalnych (np. # czy ?) do postaci %kod

Apache2 – mod_rewrite

- RewriteRule staryURL nowyURL [flagi]
- Najważniejsze flagi:
 - P - „proxy”, przekierowuje do innego URLa a serwer pełni funkcję niewidzialnego proxy. Uwaga, [P] automatycznie implikuje [P,L], tzn. żadne dalsze RewriteRule nie będą przetwarzane

Apache2 – mod_rewrite

- RewriteRule staryURL nowyURL [flagi]
- Najważniejsze flagi:
 - R=kod - „redirect”, wysyła do przeglądarki podany kod oraz
 - Jeżeli kod był postaci 3xx (redirect) wysyła URL będący wynikiem RewriteRule
 - Jeżeli kod nie zaczynał się od 3 wysyła tylko kod, bez URLa
 - Jeżeli nie poda się kodu domyślnie wysyła 302 „Found”
 - Uwaga! Nie implikuje [R,L], domyślnie po przekierowaniu serwer przetwarza kolejną RewriteRule, co zazwyczaj nie ma sensu.

Apache2 – mod_rewrite

- RewriteRule staryURL nowyURL [flagi]
- Najważniejsze flagi:
 - QSA - „query string append”, jeżeli zarówno staryURL i nowyURL zawierają znak „?” (parametry skryptu), to parametry i ze staryURL i z nowyURL zostaną połączone w wynikowym URLu
 - Domyślnie wynikowy URL będzie zawierać wyłącznie parametry z nowyURL
 - ... a parametry ze staryURL zostaną usunięte

Apache2 – mod_rewrite

- RewriteRule staryURL nowyURL [flagi]
- Najważniejsze flagi:
 - QSD - „query string discard”, jeżeli staryURL zawiera znak „?” (parametry skryptu) a nowyURL nie zawiera parametrów to wynikowy URL nie będzie zawierać parametrów skryptu
 - ... parametry ze staryURL zostaną usunięte
 - Domyślnie wynikowy URL będzie zawierać parametry ze staryURL

Apache2 – mod_rewrite

- RewriteRule staryURL nowyURL [flagi]
- Najważniejsze flagi:
 - Uwaga, QSA i QSD nie są swoimi odwrotnościami
 - ... ponieważ warunki ich zadziałania są inne
 - W pewnych sytuacjach ma sens stosowanie i QSA i QSD w jednej RewriteRule
 - ... jeżeli z góry nie wiadomo czy staryURL i nowyURL będą zawierać parametry czy nie

Apache2 – mod_rewrite

- RewriteCond wzorzec warunek [flagi]
 - Pozwala na warunkowe wykonanie dyrektywy RewriteRule
 - Jedna lub więcej dyrektywa RewriteCond może poprzedzać dyrektywę RewriteRule
 - RewriteRule zostanie wykonana tylko wtedy jeżeli wszystkie poprzedzające ją RewriteCond są spełnione

Apache2 – mod_rewrite

- RewriteCond wzorzec warunek [flagi]
 - Wzorzec jest dowolnym tekstem
 - ... który może zawierać następujące polecenia
 - \$n – n-ty nawias w ostatniej spełnionej RewriteRule
 - %n – n-ty nawias w ostatniej spełnionej RewriteCond
 - %{zmienna} – zmienna serwera

Apache2 – mod_rewrite

- RewriteCond wzorzec warunek [flagi]
- Najważniejsze zmienne serwera
 - `%{HTTP_USER_AGENT}` – przeglądarka użyta do wyświetlenia strony
 - `%{HTTP_REFERER}` – pierwotny URL (ten którego zażądała przeglądarka)
 - `%{HTTP_HOST}` – fragment `%{HTTP_REFERER}` zawierający nazwę lub IP serwera
 - `%{QUERY_STRING}` – fragment `%{HTTP_REFERER}` zawierający parametry skryptu

Apache2 – mod_rewrite

- RewriteCond wzorzec warunek [flagi]
- Najważniejsze zmienne serwera
 - `%{REQUEST_URI}` – fragment `%{HTTP_REFERER}` zawierający nazwę pliku (HTML lub innego)
 - `%{REQUEST_FILENAME}` – absolutna lokalna ścieżka do pliku do którego odnosi się URL, o ile serwer był wcześniej w stanie ten plik zidentyfikować. W przeciwnym wypadku równoważne `%{REQUEST_URI}`
 - `%{SCRIPT_FILENAME}` – jak `%{REQUEST_FILENAME}`, ale ścieżka jest względem DocumentRoot

Apache2 – mod_rewrite

- RewriteCond wzorzec warunek [flagi]
- Najważniejsze zmienne serwera
 - `%{REMOTE_ADDR}` – adres IP komputera na którym działa przeglądarka
 - `%{AUTH_TYPE}` – użyty sposób autoryzacji użytkownika
 - `%{THE_REQUEST}` – pełne pierwotne żądanie HTTP, łącznie z metodą i z wersją HTTP
 - `%{REQUEST_METHOD}` – fragment `%{THE_REQUEST}` zawierający metodę

Apache2 – mod_rewrite

- RewriteCond wzorzec warunek [flagi]
- Najważniejsze zmienne serwera
 - %{HTTPS} – ma wartość:
 - „on” jeżeli połączenie nawiązano przy pomocy https://
 - „off” jeżeli połączenie nawiązano przy pomocy http://

Apache2 – mod_rewrite

- RewriteCond wzorzec warunek [flagi]
- Warunek - wyrażenie regularne (regex), z następującymi niestandardowymi konstrukcjami
 - ! - RewriteCond będzie spełnione jeżeli wzorzec nie będzie spełniał warunku
 - =tekst – przetworzony wzorzec jest równy podanemu tekstowi
 - -d – przetworzony wzorzec jest ścieżką do istniejącego katalogu
 - -f – przetworzony wzorzec jest ścieżką do istniejącego pliku

Apache2 – mod_rewrite

- RewriteCond wzorzec warunek [flagi]
- Warunek - wyrażenie regularne (regex), z następującymi niestandardowymi konstrukcjami
 - -l – przetworzony wzorzec jest ścieżką do istniejącego dowiązania symbolicznego (symlinku)
 - -x – przetworzony wzorzec jest ścieżką do istniejącego pliku wykonywalnego
 - -F – przetworzony wzorzec jest ścieżką do istniejącego pliku, a konfiguracja serwera pozwala na dostęp do tego pliku

Apache2 – mod_rewrite

- RewriteCond wzorzec warunek [flagi]
 - Warunek - wyrażenie regularne (regex), z następującymi niestandardowymi konstrukcjami
 - -U – przetworzony wzorzec jest URL-em, a konfiguracja serwera pozwala na dostęp do tego URL-a

Apache2 – mod_rewrite

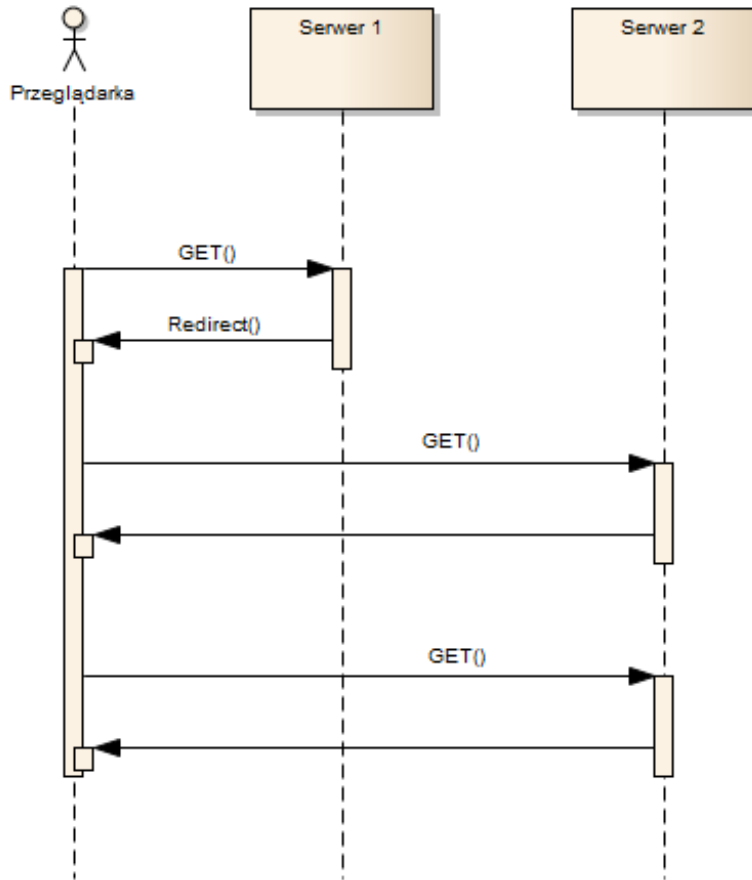
- RewriteCond wzorzec warunek [flagi]
- Flagi
 - NC - „nocase”, we wzorcu nie będą rozróżniane duże i małe litery
 - OR - „ornext”, RewriteRule zostanie wykonana jeżeli ta lub następna RewriteCond będą spełnione (domyślnie muszą być spełnione obie RewriteCond na raz)

Apache2 – mod_rewrite

- RewriteCond wzorzec warunek [flagi]
- Przykład
 - RewriteCond %{HTTP_USER_AGENT} „android” [NC,OR]
 - RewriteCond %{HTTP_USER_AGENT} „iphone” [NC]
 - RewriteRule ^/index.html\$ /mobile.html [L]
 - RewriteRule ^/\$ /mobile.html

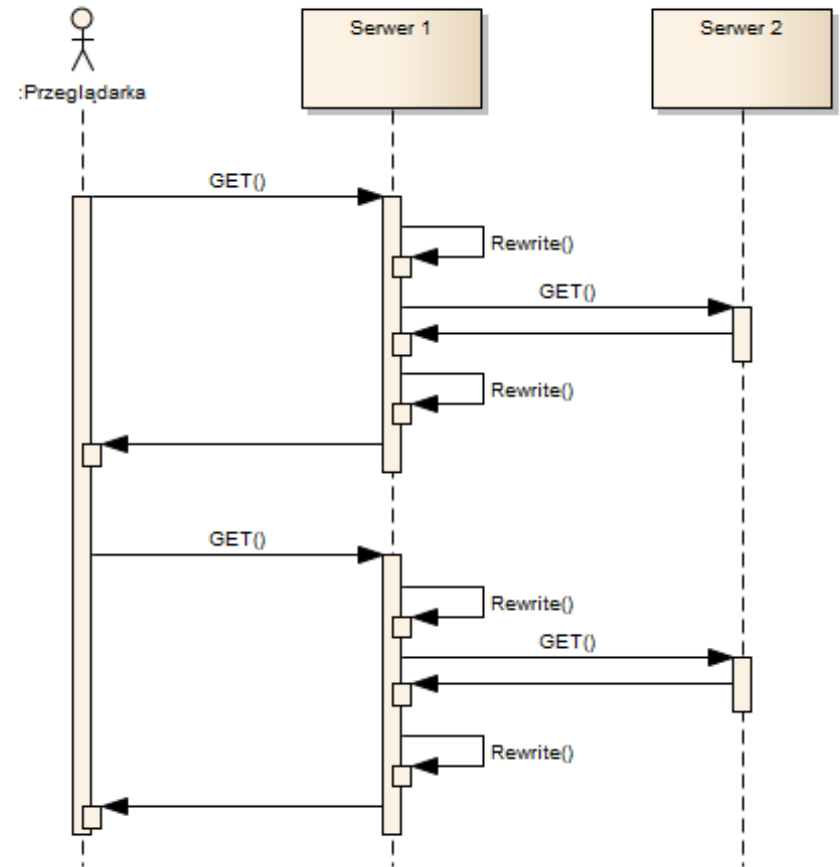
Apache2 – alias i rewrite – różnice w działaniu

sd mod_alias



mod_alias

sd Rewrite



mod_rewrite