

กรณีการใช้งานจริง สำหรับ mod_rewrite ของ Apache - ฉบับที่ 165 สิงหาคม 2009

โดย [Anderson Silva](#) [1]

แปลโดย [Sake](#) [2]

เทคโนโลยีเป็นสิ่งที่น่าขบขัน. บางครั้ง คุณต้องการที่จะเขียนถึงมันเพียงบางส่วนโดยเฉพาะ. บางครั้ง, คุณต้องการจะแบ่งปันความรู้กับบางคน, แต่ในการทำ, และทำให้ดี, คุณรู้สึกถึงความจำเป็นที่จะอธิบายเทคโนโลยีทั้งหมด ที่ต้องใช้ในการสร้างส่วนที่เฉพาะเจาะจงนั้นให้สำเร็จ.

บทความนี้ ไม่ได้กล่าวถึงว่า *mod_rewrite* ทำงานจริง ๆ ได้อย่างไร. ถ้าต้องการทราบ ผมอาจจะต้องเขียนเกี่ยวกับสิ่งต่าง ๆ อย่าง : โพรโตคอล HTTP [3], [Apache](#) [4] HTTP Server, [นิพจน์ปกติ \(Regular Expression\)](#) [5], และอื่น ๆ อีกเล็กน้อย.

คน ๆ หนึ่งไม่ได้จำเป็นต้องทราบว่ารถยนต์ทำงานอย่างไร, จากทฤษฎีของฟิสิกส์ทั้งหมดที่จะสร้างเครื่องยนต์กลไกมันขึ้นมา, เพื่อที่จะสามารถขับชี้ได้, ถูกต้องไหม้? เพราะฉะนั้น, บทความนี้จะไม่ใช่เกี่ยวกับการทำงานข้างใต้ เมื่อทำการจัดการกับ *mod_rewrite*. แต่จะแสดงเพียงว่าเปิดมันอย่างไร, และทำงานกับมันอย่างไร.

ถ้าอย่างนั้น, *mod_rewrite* คืออย่างไร? มันทำงานรวดเร็ว, และยังมีดหยุ่นพอสมควรและมีศักยภาพในเชิงซับซ้อน ที่จะจัดการกับ URL ในด้านเครื่องแม่ข่ายโดยการใช้กฎของนิพจน์ปกติ. คุณสามารถจับความสัมพันธ์ของการร้องขอ HTTP ด้วยเงื่อนไขหลากหลาย อย่างตัวแปรเครื่องแม่ข่าย, ส่วนหัว HTTP และอื่น ๆ.

ผมไม่แน่ใจเกี่ยวกับตัวแจกจ่ายลินุกซ์อื่น, แต่บน Fedora ที่ผมเลือกใช้, Apache HTTP Server ถูกลงแยกจากเครื่อง พร้อมกับ *mod_rewrite* ทำงาน, แต่ถูกปิดไว้.

ในการเปิดใช้ เพียงแค่เพิ่ม:

RewriteEngine On

ใน httpd.conf ของคุณ, หรือถ้าคุณทำการใช้งานโฮสต์เสมือนบนเครื่องแม่ข่ายของคุณ, คุณสามารถเปิดใช้งาน *mod_rewrite* ในแต่ละโฮสต์เสมือนของคุณ.

ตอนนี้, ถ้าคุณใช้งานกับนิพจน์ปกติแล้ว, และคุณไม่ได้สะดวกสบายกับมัน, มันง่ายมากที่จะถูกทำให้เป็นผู้พ่ายแพ้โดยมัน. ในการที่จะทำให้สิ่งต่าง ๆ ง่ายขึ้น, *mod_rewrite* ได้สร้างตัวเก็บบันทึกระบบมาด้วย เพื่อที่จะช่วยผู้ดูแลในการแก้ปัญหาต่าง ๆ.

การเปิดใช้บันทึกการทำงาน *mod_rewrite* ของคุณ:

```
RewriteLog /var/log/httpd/rewrite.log
RewriteLogLevel 5
```

อย่างน้อย, นี่เป็นวิธีที่คุณจะเริ่มทำงานกับ และพร้อมที่จะแก้ปัญหาของมัน.

RewriteCond - นี่เป็นคำสั่งที่อนุญาตให้คุณทดสอบสถานะบางอย่างสำหรับกฎที่จะนำมาใช้. คิดว่ามันเป็นถ้อยแถลง(statement) if ของภาษาในการเขียนโปรแกรมที่คุณใช้อยู่ทุกวัน, สองสถานะหรือมากกว่าสามารถถูกเขียนอย่างเป็นลำดับอย่างตรรกะ AND, หรือโดยการเพิ่ม [OR] ที่ท้ายของบรรทัดสำหรับตรรกะ [OR]. คุณจะเห็นได้ว่า RewriteCond เป็นสิ่งที่ยืดหยุ่น และอนุญาตให้คุณที่จะเขียนการทดสอบสำหรับตัวแปรเครื่องแม่ข่ายอย่างส่วนหัว HTTP, การเชื่อมต่อและการร้องขอ, ส่วนภายในเครื่องแม่ข่าย, หรือแม้แต่ข้อมูลระบบ.

RewriteRule - เป็นคำสั่งที่สำคัญที่สุดที่คุณใช้. มันเป็นสิ่งที่ตามเอกสารของ Apache เรียกใช้งานมัน, เป็น 'ม้านางในการเขียนใหม่จริง ๆ' ของโมดูล *mod_rewrite*. มันมักจะใช้ 3 พารามิเตอร์: รูปแบบในการจับคู่, สายอักขระที่จะแทนที่, และรายการของเครื่องหมาย. นี่คือการของเครื่องหมายที่ผมเพิ่งใช้บนตัวอย่างข้างต้น:

R - บอก RewriteRule ว่าคุณกำลังทำการเปลี่ยนทิศทางใหม่, และ, นอกจากนี้ใส่รหัส 301, มันจะมีค่ามาตรฐานเป็น 302, ซึ่งหมายถึงย้ายชั่วคราว.

L - บอก RewriteRule ให้ออกจากสายโซ่ของกฎ และไม่ตามกฎใด ๆ หลังจาก RewriteRule ล่าสุด.

NC - ทำให้รูปแบบการจับคู่เป็นแบบอักขระเล็กใหญ่ไม่สำคัญ.

NE - บอก RewriteRule ไม่ต้องหลีกเลี่ยง (escape) ผลของ URI กับสิ่งที่มีลักษณะคล้ายกับ %20 สำหรับที่ว่าง.

สรุปท้ายสุด

mod_rewrite ของ Apache เป็นเครื่องมือที่ความยืดหยุ่นอย่างไม่น่าเชื่อ ในการอนุญาตให้ผู้ดูแลระบบ ที่จะทำงานอย่างรวดเร็ว ในการแก้ไขประเด็นปัญหาเกี่ยวกับเครื่องแม่ข่ายเว็บ. การแก้ไขบางอย่างอาจจะเป็นแบบลักษณะชั่วคราวจนกว่าคำตอบถาวรที่เหมาะสมนำมาใช้, และ, ถึงแม้ว่ามันอาจจะมีดีกว่าเมื่อ *mod_rewrite* อาจจะเป็นส่วนหนึ่งของการแก้ปัญหาถาวร, อนาคตมันมากเกินไป, เพราะกฎของ *mod_rewrite* อาจจะสามารถอย่างรวดเร็ว และเป็นสิ่งที่ยากต่อการดูแลมัน. คุณเคยที่ดูแลโค้ดภาษา Perl code ที่มี regexes อยู่ทุก ๆ ที่มัย? ถ้าเคย, คุณก็อาจจะรู้ว่าผมพูดอะไรอยู่.

ท้ายสุด, ถ้าคุณต้องการที่จะรู้มากขึ้นว่า ข้างใต้ของ *mod_rewrite* มีอะไรอยู่, ให้แน่ใจว่า คุณอ่าน [เอกสารคู่มือ](#) [11]ของ Apache แล้ว, และเมื่อสงสัย ใช้บันทึกการทำงานของ *mod_rewrite* เพื่อช่วยคุณในการแก้ไขปัญหา.

แหล่งข้อมูลภายนอก

1. <http://www.w3.org/Protocols/rfc2616/rfc2616.html> [3]
2. <http://httpd.apache.org> [4]
3. http://en.wikipedia.org/wiki/Regular_expression [5]
4. http://groups.google.com/group/Google_Webmaster_Help/web/faqs-for-crawling-indexing-and-ranking-2?pli=1 [10]
5. http://httpd.apache.org/docs/2.2/mod/mod_rewrite.html [11]

Talkback: [สนทนาเกี่ยวกับบทความนี้กับ The Answer Gang](#) [12]



แอนดรูว์ ซิลวา (Anderson Silva) ทำงานเป็นวิศวกรการออกวางจำหน่ายสารสนเทศที่ Red Hat, Inc.

เขาจบปริญญาตรีด้านวิทยาการคอมพิวเตอร์จากมหาวิทยาลัยลิเวอร์พูล, ปริญญาโทด้วยระบบข้อมูลจากมหาวิทยาลัยมาอิน.

เขาเป็นวิศวกรรับรองของหมวกแดง (Red Hat Certified Engineer), และ เป็นผู้เขียนบทความด้านลินุกซ์หลายแห่งเพื่อการเผยแพร่: Linux Gazette, Revista do Linux, และ Red Hat Magazine. แอนดรูว์ได้แต่งงานกับทวานใจสมัยมัธยมปลายมาแล้ว 11 ปี, มีลูก 3 คน. เมื่อเขาไม่ได้ทำงานเขียน, เขามีความสุขกับการใช้เวลากับครอบครัว, ดูรถสูตร 1 และแข่งรถอิสระ, และพูดคุยเรื่องรถโกคาร์ทกับลูกชาย.

สงวนลิขสิทธิ์ ปี 2009, Anderson Silva. ออกวางภายใต้สัญญาอนุญาต Open Publication license เว้นแต่บันทึกภายในบทความบอกเป็นอย่างอื่น. Linux Gazette ไม่ได้ถูกสร้างขึ้น, ได้รับการสนับสนุน, หรือได้รับการรับรอง จากผู้ให้โฮสต์, SSC, Inc.

ตีพิมพ์ในเล่มที่ 16 ของ of Linux Gazette, สิงหาคม 2009



-
- อ่าน 9492 ครั้ง

Source URL (modified on 2009-08-16 16:49): <https://sake.in.th/node/85>

Links

[1] <http://linuxgazette.net/authors/silva.html>

[2] <http://sake.in.th>

[3] <http://www.w3.org/Protocols/rfc2616/rfc2616.html>

[4] <http://httpd.apache.org/>

[5] http://en.wikipedia.org/wiki/Regular_expression

[6] http://www.yourcompany.com/ask_me_how/

[7] <http://www.yourcompany.com/ask-me-how/>

[8] <http://www.yourcompany.com>

[9] <http://www.yourcompany.net>

[10] http://groups.google.com/group/Google_Webmaster_Help/web/faqs-for-crawling-indexing-and-ranking-2?pli=1

[11] http://httpd.apache.org/docs/2.2/mod/mod_rewrite.html

[12] <mailto:tag@lists.linuxgazette.net?subject=Talkback:165/silva.html>