

Header Already Sent!?!

เฮ!... ก็ไม่ได้ใส่คำสั่งเอาต์พุตอะไรก่อนหน้าเลยนี่ แล้วทำไม...

คนเขียน PHP ร้อยละ 99.999% ต้องเคยเจอข้อความ Error อันนี้ โดยเฉพาะตอนที่ใช้คำสั่งแก้ Header ของ HTTP ด้วยฟังก์ชัน `header()` (เช่น Redirect หรือเปลี่ยน File Type) หรือโดยเฉพาะการสร้าง Session ID ด้วยฟังก์ชัน `session_start()` เช่น การให้กรอก User/Pass ก่อนเข้าใช้บริการภายในเว็บไซต์ และร้อยละ 99.998% ก็เชื่อว่าปัญหานี้แก้ตาม “คำฟ่อง” มันเลย คือ เอาฟังก์ชันหรือคำสั่งพวกนี้ย้ายไปอยู่บนสุดของไฟล์ พร้อมทั้งลบอักขระเว้นว่างก่อนหน้า “<?” หรือ “<?php” ช่วงต้นไฟล์ไปให้หมด (เช่น Backspace ถี่ๆๆๆ) ให้แน่ใจว่าไม่ได้ส่ง Output อะไรออกไปก่อนหน้า Command พวกนี้

แถมเหตุการณ์นี้เป็นเฉพาะกับบางโฮสต์ดั้งเดิมด้วย ซึ่งทำให้เราสันนิษฐานได้ว่าน่าจะมาจากการตั้งค่าที่เกี่ยวข้องกับการประมวลผลไฟล์ .php เช่น การตั้งค่าออกพชั่นต่างๆ ใน “php.ini” แน่แน่นอนว่าเราที่เป็นลูกค้าไม่มีสิทธิ์จะเข้าไปแก้ได้ ตามก็มีวิธีที่ใช้ตั้งค่าทับ (Override) เฉพาะไฟล์เดอร์ที่ต้องการ เช่น การสร้างไฟล์ “.htaccess”

ความรู้เพิ่มเติม: ไฟล์ .htaccess มีไว้สำหรับแก้พชั่นที่เซตไว้ใน php.ini ซึ่งเป็นไฟล์คอนฟิกหลักของตัวคอมไพล์ PHP บนเซิร์ฟเวอร์ ซึ่งมีผลเฉพาะไฟล์ .php ในไฟล์เดอร์ที่มีไฟล์ตัวนี้อยู่ และไฟล์เดอร์ลูกของไฟล์เดอร์หลักนี้

ดังนั้น เราจึงสร้างไฟล์ “.htaccess” บนไฟล์เดอร์ของเว็บไซต์หลัก (เช่น public_html) แล้วในไฟล์ก็ใส่คำสั่งแก้ค่าออกพชั่นด้วย โดยคำสั่งที่จะใช้กรณีนี้คือ `PHP_FLAG output_buffering on` เนื่องจากสันนิษฐานว่า ที่มันยังฟ่องว่า “header already sent” แสดงว่าต้องมีเอาต์พุตอะไรบางอย่างส่งออกมาก่อนเริ่มคอมไพล์ไฟล์ .php ของเราแน่นอน เลยเปิด Output Buffering ให้ PHP บัฟเฟอร์ข้อมูลที่จะเอาต์พุตไว้ก่อนประมวลผลเสร็จทั้งไฟล์ดีกว่า (มานั่งให้เราบ้าจี้กด backspace ไร้จุดหมายต่อทำไม ใช่มั้ย)

ความรู้เพิ่มเติม: ปกติ ค่า `output_buffering` ใน `php.ini` จะเซตเป็น 0 (off) ทำให้เวลาคอมไพล์ไฟล์ .php จะส่งเอาต์พุตออกมาทันทีที่อ่านแต่ละบรรทัดจบ (จบด้วยเครื่องหมาย “;”) การเปิด Output Buffering ทำให้ PHP บัฟเฟอร์เอาต์พุตไว้ก่อน รอให้ประมวลผลจบทั้งหมด แล้วค่อยส่งเอาต์พุตออกมา ทำให้ไม่เกิดกรณี Error “header already sent” เวลาใช้คำสั่งดัดแปลง header อีก

ความรู้เพิ่มเติม: ศึกษารายละเอียดการแก้ไข .htaccess เพิ่มเติมได้ที่

<http://php.net/manual/en/configuration.changes.php>

แก้ได้จริงๆ ด้วย เพียงแต่ก็มีข้อสังเกตว่าการเปิดบัฟเฟอร์เอาต์พุตนี้จะทำให้เวลา Debug ยากอยู่ เพราะ PHP จะระบุไม่ได้ว่าความผิดพลาดที่ทำให้ต้องหยุดคอมไพล์เกิดที่บรรทัดไหน (และระวังด้วยว่า ถ้าได้มี Error หรือต้องรอ Request ที่บรรทัดไหน มันจะไม่ยอมส่งเอาต์พุตออกมาสักบรรทัดเลย เพราะต้องรอให้เสร็จทั้งไฟล์ก่อน บางทีก็อาจรอเก้ออยู่อย่างนั้น!)

ที่นี้ ปัญหาต่อมาคือ การสร้าง Session ด้วยคำสั่ง “`session_start()`” ซึ่งจะสร้าง Session ID ขึ้นเป็น Cookie เก็บไว้บนเครื่องผู้ใช้ สำหรับจดจำค่าตัวแปรที่เราสร้างตระกูล `$_SESSION[]` ได้ เช่น จำค่า User/Pass สำหรับล็อกอินไว้ ซึ่งเราต้องการให้มันจำค่าชุดตัวแปรเหล่านี้ตลอดไม่ว่า Navigate ไปหน้าเว็บไหนภายในโดเมนของเว็บไซต์เรา ถูกไหมครับ แต่ถ้าเกิดในบางโฮสต์ตั้งค่าตัวแปร SESSION หลุดออกทุกครั้งที่ Navigate ไปหน้าอื่น (เช่นต้องล็อกอินใหม่ตลอด) จะทำอย่างไรดี

เป็นไปได้ว่า Session ID เดิมถูกเปลี่ยนให้หมดอายุตลอดทุกครั้งที่โหลดสคริปต์ .php ใหม่ ดังนั้นมุขที่ใช้แก้ปัญหาเฉพาะหน้าเบื้องต้นคือ การรีเฟรช Session ID ทุกครั้งที่ Start Session โดยที่ไม่ได้เปลี่ยนหรือลบค่าตัวแปร Session เดิม ซึ่งตรงกับหน้าที่ของฟังก์ชัน “`session_regenerate_id()`” พอดี

ความรู้เพิ่มเติม: ฟังก์ชัน “`session_regenerate_id()`” ใช้สร้าง Session ID ใหม่แบบ “ทับของ Session เดิม”

นั่นคือ ค่าตัวแปรตระกูล `$_SESSION[]` ที่เรจิสเตอร์ไว้ก่อนหน้านี้จะไม่หายไปด้วย นิยมใช้คู่กับ “`session_start()`” เวลาสร้าง Session ที่ต้องการความปลอดภัยสูง เช่น การล็อกอินด้วย User/Pass

หลังจากใส่ “`session_regenerate_id()`” คู่ไปกับ “`session_start()`” ทุกหน้า ก็ทำให้ยังคงตัวแปรเซสชันอยู่ได้แม้จะ Navigate ไปหน้าอื่น แล้วเวลาอยากปิดเซสชันค่อยสั่ง “`session_unset()`” หรือ “`session_close()`” เคา (หรือการปิดหน้าต่างเบราว์เซอร์ก็เป็นการออกจากเซสชันไปโดยปริยายอยู่แล้ว)

ปัญหาหลายอย่างที่เกี่ยวกับเทคโนโลยี เวมักเจอในแบบที่ต้องแก้ปัญหาลเฉพาะหน้า คนที่คลุกคลี นานจนเชี่ยวชาญ หรือมีประสบการณ์ก็เชื่อว่าอยู่รอด เพราะเทคโนโลยีพัฒนาเปลี่ยนแปลงอย่างรวดเร็ว การ คิดแก้ไขตามหลักเหตุผลโดยมีทฤษฎีที่ถูกต้องเป็นพื้นฐานไว้จะช่วยร่นระยะเวลาแก้ปัญหาลเหล่านี้ได้มากขึ้น

