**COOLDOWNS**

* Trong phần này sẽ hướng dẫn cách tạo “Cooldowns” trong plugins, ở một số trường hợp không phải lúc nào nó sẽ thực hiện liên tục mà phải có khoảng ngắt quãng. Đây cũng là cách để giảm gánh nặng server khi thực hiện khối lượng công việc.
* Trong phần này sẽ ví dụ thông qua việc người chơi thực hiện liên tục câu lệnh:

|  |
| --- |
| Public class FartCommand implements CommandExecutor {  Private final HashMap<UUID, Long> coolDowns;  Public FartCommand() {  This.coolDowns = new HashMap<>();  }  @Override  Public boolean onCommand(CommandSender sender, Command command, String label, String[] args) {  If(sender instanceof Player) {  Player p = (Player) sender;  If(!this.coolDowns.containsKey(player.getUniqueId()) || timeElapsed >= 10000) {  this.coolDowns.put(player.getUniqueId(), System.currentTimeMillis());  player.sendMessage(“Run command successfully!”);  } else {  player.sendMessage(“Please waitting for ” + (10000 – System.currentTimeMillis() – coolDowns.get(p.getUniqueId())) + “ to execute the next command!”);  }  }  }  } |

* Giải thích:
  + UUID là mã UUID của người chơi để nhắm tới.
  + Long là khoảng thời gian ngắt quãng
  + Khi người chơi thực thi một câu lệnh (giả sử là “abc”) thì nó sẽ kiểm tra xem trong HashMap nó có mã UUID của người chơi đó hay không hoặc là thời gian phản hồi nó lớn hơn 10000 ticks (tương đương 10s) thì nó sẽ cho phép người chơi thực thi câu lệnh và đồng thời thêm mã UUID của người chơi đó vào HashMap.
  + Nếu trường hợp mà người chơi đó đã có mã UUID trong HashMap rùi thì nó sẽ tiến hành chặn câu lệnh thực thi của người chơi đó và in ra thông báo cho người chơi biết là mình không được phép thực thi câu lệnh liên tục.
  + Trong trường hợp người chơi đã có mã UUID trong HashMap rùi thì sẽ tính trường hợp thứ hai nếu thời gian phản hồi >= 10s coi như cho phép thực thi và đặt lại thời gian phản hồi lại cho người chơi.