

Forum:[Hack Forums](#)

Thread: [Stopping a ScheduledExecutorService](#)

Juice. - August 1, 2020, 10:13 am

So I have a discord bot where I am trying to send a message every hour. I got that part of it working using the scheduled executor service. It starts when I say !futureson

I want to be able to stop it when I send !futuresoff

However, I cannot get this to work properly. Here is my code:

```
Code ScheduledExecutorService exec = Executors.newScheduledThreadPool(1);
String[] args = event.getMessage().getContentRaw().split(" ");
if (args[0].equalsIgnoreCase("!futuresOn")) {
    Color red = new Color(255, 0, 0);
    Color green = new Color(0, 204, 0);
    Runnable futRun = new Runnable() {
        public void run() {
            EmbedBuilder futuresBot = new EmbedBuilder();
            futuresBot.setAuthor("Futures Bot", null, event.getAuthor().getAvatarUrl());

            try {
                futuresBot.addField("***S&P 500***", getSPY(), false);
                futuresBot.addField("***NASDAQ***", getNASDAQ(), false);
                futuresBot.addField("***DOW***", getDOW(), false);
                if (getSPY().contains("+")) {
                    futuresBot.setColor(green);
                }
                else {
                    futuresBot.setColor(red);
                }
            }
            catch (IOException e) {
                // TODO Auto-generated catch block
                e.printStackTrace();
            }

            futuresBot.setFooter("Cantina Capital", event.getGuild().getIconUrl());
            event.getChannel().sendMessage(futuresBot.build()).queue();

        }
    };

    exec.scheduleAtFixedRate(futRun, 0, 1, TimeUnit.HOURS);
```

```
}  
if (args[0].equalsIgnoreCase("!futuresoff")) {  
    event.getChannel().sendMessage("Futures bot off.").queue();  
    exec.shutdown();  
}
```

When I send !futuresoff it recognizes it because it sends my "Futures bot off." message. The shutdown thing isn't working like I want it to though. Could someone point me in the right direction?

GenericSkid - August 1, 2020, 1:40 pm

Shutdown() only closes off the service from taking in new runnables.

You want to call shutdownNow()

[https://docs.oracle.com/javase/7/docs/api/java/util/concurrent/ExecutorService.html#shutdownNow\(\)](https://docs.oracle.com/javase/7/docs/api/java/util/concurrent/ExecutorService.html#shutdownNow())

Juice. - August 1, 2020, 2:08 pm

(Aug 1, 2020 01:40 PM)GenericSkid Wrote: [\[CITE\]](#) Shutdown() only closes off the service from taking in new runnables.

You want to call shutdownNow()

[https://docs.oracle.com/javase/7/docs/api/java/util/concurrent/ExecutorService.html#shutdownNow\(\)](https://docs.oracle.com/javase/7/docs/api/java/util/concurrent/ExecutorService.html#shutdownNow())

```
Code if (args[0].equalsIgnoreCase("!futuresoff")) {  
    exec.shutdownNow();  
    event.getChannel().sendMessage("Attempting to shut down futures bot.").queue();  
}
```

I tried this and it does not stop it from running unfortunately :(I also set the scheduled thread pool to 0 instead of 1 at the start. Any other ideas? Or did I not implement that correctly?

knox - August 1, 2020, 8:17 pm

(Aug 1, 2020 02:08 PM)Juice. Wrote: [\[CITE\]](#) Code if (args[0].equalsIgnoreCase("!futuresoff")) {
 exec.shutdownNow();
 event.getChannel().sendMessage("Attempting to shut down futures bot.").queue();
}

I tried this and it does not stop it from running unfortunately :(I also set the scheduled thread pool to 0 instead of 1 at the start. Any other ideas? Or did I not implement that correctly?

"Does not stop it from running" means that your runnable continues to be called, or that your application does not terminate?

It would help to be more explicit about your expectations. Exactly what are you expecting to happen, and what is happening instead?

Hopefully this doesn't come off as rude, but from an outsiders perspective, we don't know what "The shutdown thing isn't working like I want it to though" means, because we don't know what your goals are.