1配置类（记得加@EnableAsync注解）

**import** org.springframework.context.annotation.Bean;  
**import** org.springframework.context.annotation.Configuration;  
**import** org.springframework.scheduling.annotation.EnableAsync;  
**import** org.springframework.scheduling.concurrent.ThreadPoolTaskExecutor;  
**import** java.util.concurrent.Executor;  
**import** java.util.concurrent.ThreadPoolExecutor;  
  
*/\*\*  
 \* Created by zxl on 2019/5/1.  
 \*/*@Configuration  
@EnableAsync  
**public class** TaskExecutePool {  
  
 **private int corePoolSize** = 10;*//线程池维护线程的最少数量* **private int maxPoolSize** = 30;*//线程池维护线程的最大数量* **private int queueCapacity** = 8; *//缓存队列* **private int keepAlive** = 60;*//允许的空闲时间* @Bean  
 **public** Executor myTaskAsyncPool() {  
 ThreadPoolTaskExecutor executor = **new** ThreadPoolTaskExecutor();  
 executor.setCorePoolSize(**corePoolSize**);  
 executor.setMaxPoolSize(**maxPoolSize**);  
 executor.setQueueCapacity(**queueCapacity**);  
 executor.setThreadNamePrefix(**"mqExecutor-"**);  
 *// rejection-policy：当pool已经达到max size的时候，如何处理新任务  
 // CALLER\_RUNS：不在新线程中执行任务，而是由调用者所在的线程来执行* executor.setRejectedExecutionHandler(**new** ThreadPoolExecutor.CallerRunsPolicy()); *//对拒绝task的处理策略* executor.setKeepAliveSeconds(**keepAlive**);  
 executor.initialize();  
 **return** executor;  
 }  
}

2在任务方法上加上注解，并加上自定义的线程池名（@Async("myTaskAsyncPool")）

**import** org.springframework.scheduling.annotation.Async;  
**import** org.springframework.stereotype.Component;  
**import** org.springframework.stereotype.Service;  
  
*/\*\*  
 \* Created by zxl on 2019/5/1.  
 \*/*@Component  
**public class** AsynTask {  
  
 @Async(**"myTaskAsyncPool"**)  
 **public void** test(String name) {  
 System.***out***.println(name);  
 }  
}

3启动类加上开启异步注解（@EnableAsync）

@EnableAsync  
@SpringBootApplication  
**public class** SpringbootPoolAsyncApplication {  
 **public static void** main(String[] args) {  
 SpringApplication.*run*(SpringbootPoolAsyncApplication.**class**, args);  
 }  
}

4controller调用

*/\*\*  
 \* Created by zxl on 2019/5/1.  
 \*/*@RestController  
**public class** TestController {  
  
 @Autowired  
 **private** AsynTask **asynTask**;  
  
 @GetMapping(**"/testAsynTask"**)  
 **public** String test(@RequestParam(**"name"**)String name){  
 *//异步并在线程池里开启一个子线程* **asynTask**.test(name);  
 **return** name;  
 }  
}