

Comparing Thread, Handler, Service, AsyncTask in Android

- **AsyncTask**(short-lived tasks): thực thi các task nhỏ yêu cầu giao tiếp với main thread(ui thread). AsyncTask cho phép thực thi tiến trình ở background sau đó hiển thị kết quả lên UI thread một cách dễ dàng. nên sử dụng AsyncTask khi caller thread là UI thread.

- **Thread**(standard java construction) và **Handler**: thực thi các task lớn hơn, có thể giao tiếp với các worker thread khác và main thread(caller thread).

- Thread được sử dụng để thực thi một tiến trình song song với main thread(caller thread), không thể giao tiếp với UI, nếu muốn giao tiếp với UI từ thread chúng ta phải cần thêm Handler, nhưng như vậy thì sẽ làm cho ứng dụng trở nên phức tạp hơn khi phải xử lý các responses từ Thread.

- Thread là một tiến trình thực thi ở background. Nó cũng có call stack. Có 2 cách thức để thực thi 1 thread trong ứng dụng.

+ sử dụng một class mới extends Thread và override phương thức run().

+ cung cấp một instance của lớp Runnable và override phương thức run().

- **Handler** cho phép gửi và xử lý các đối tượng message và runnable được liên kết với MessageQueue của Thread tạo ra nó. mỗi Thread sẽ có một message queue. Thread sẽ xử lý mỗi message cho đến khi message queue trống.

- **Service**(time consuming tasks): là một context giống như một Activity nhưng không có UI. Service cho phép thực thi tiến trình ở background một cách lâu dài, cung cấp chức năng cho các ứng dụng khác sử dụng. Service cũng giống như các thành phần khác của ứng dụng, chạy trên main thread(caller thread). do đó đối với những thực thi đòi hỏi nhiều tính toán CPU(playing music, networking) chúng ta nên tạo riêng một thread mới cho Service.

- IntentService là một Service chạy trên một Thread riêng.

bảng so sánh

	Service	Thread	Intentservice	AsyncTask	Handler
When to use?	- không cần UI, - không nên thực thi các tiến trình lớn. mà nên xử lý các tác vụ nhỏ nhưng những thời gian xử lý lâu. - nếu cần xử lý các tiến trình lớn, nên sử dụng các Thread và	- xử lý long Task - xử lý song song(multi threads)	- xử lý long task mà không cần giao tiếp với main thread - nếu cần giao tiếp với main thread, có thể sử dụng handler, hoặc broadcast intent.	- small task, cần giao tiếp với UI thread. - để xử lý song song sử dụng nhiều instances, hoặc executor.	- là thành phần giao tiếp giữa UI và background threads.

	Handler.				
Trigger	OnStartService())	Start()	Intent	Execute(), ExcuteOnExcutor(param1, param2)	
Triggered from(Thread)	bất kỳ thread	bất kỳ thread	main thread(intent được nhận trên main thread và sau đó sẽ thực thi trên worker thread)	-main thread.	main thread
Runs on(Thread)	main thread	thread riêng	thread riêng	- worker thread - có thể giao tiếp với UI thread thông qua một số methods được hỗ trợ(onPrePostExecute(), onPostExcute, onUpdateProgress()).	main thread
Limitations /drawbacks	có thể block UI	- phải quản lý Thread - code có thể khó đọc	- không thể chạy song song. - nếu nhiều intents sẽ được đưa lên hàng đợi của cùng 1 thread.	- từ HoneyComb thì Excute() mặc định sẽ được chạy tuần tự(serially). nếu muốn chạy song song (parallel) cần gọi executeOnExcutor(java.util.concurrent.Executor, Object[]) với THREAD_POOL_EXECUTOR. - một instance chỉ được thực thi một lần. - nên được khởi tạo và thực thi từ main thread.	