Ảnh có chứa văn bản

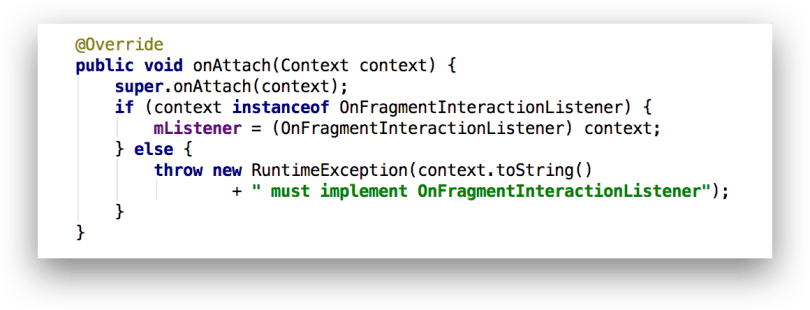
Mô tả được tạo tự động

Sau đây là ý nghĩa của từng callback

### onAttach()

Callback này được gọi khá sớm, ngay khi Activity chứa nó được kích hoạt. Hoặc ngay khi được gắn vào Activity.

Callback này được gọi một lần duy nhất trong vòng đời Fragment. Và ở giai đoạn này Fragment đã “nhận biết” được Activity chứa nó rồi, nên bạn có thể tận dụng để kiểm tra sớm một số điều kiện nào đó như các dòng code ở ***FirstFragment*** chúng ta đã từng làm.

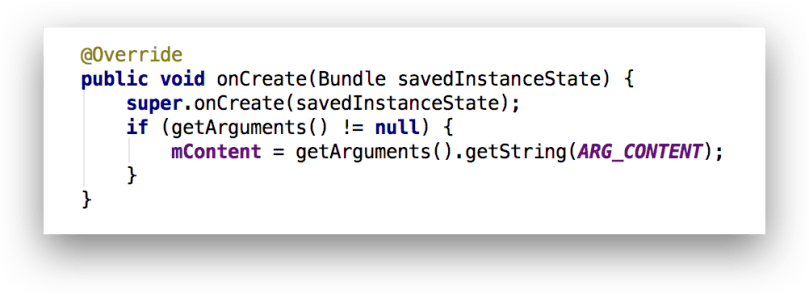
[](https://i0.wp.com/yellowcodebooks.com/wp-content/uploads/2018/03/Screen-Shot-2018-03-06-at-16.23.36-1.png?ssl=1)

### onCreate()

Callback này được gọi khi Fragment bắt đầu khởi tạo từ các dữ liệu đầu vào.

Khác với [onCreate() của Activity](https://yellowcodebooks.com/2017/10/19/android-bai-28-vong-doi-cua-mot-activity/" \l "onCreate" \t "_blank), rằng bạn có thể tạo giao diện cho màn hình ở callback này, thì với Fragment chúng ta còn phải đợi qua callback tiếp theo mới có thể tạo giao diện được.

Callback này cũng được gọi một lần trong đời sống Fragment. Nên thường tận dụng để lấy dữ liệu từ bên ngoài truyền vào như ở ***SecondFragment*** chúng ta có làm quen.

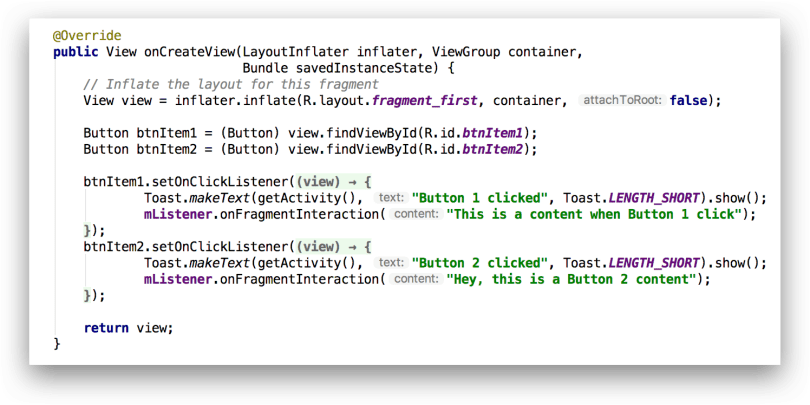
[](https://i0.wp.com/yellowcodebooks.com/wp-content/uploads/2018/03/Screen-Shot-2018-03-06-at-16.33.12.png?ssl=1)

### onCreateView()

Khi Fragment bắt đầu vẽ UI lên màn hình, callback này được gọi. Nên chúng ta sẽ tận dụng callback này cho các thiết lập về giao diện.

Bạn thấy rằng, theo như sơ đồ trên, thì callback này sẽ được gọi lại khi mà Fragment được gỡ ra khỏi Activity nhưng được đưa vào Back Stack, và được gọi lại hiển thị sau đó.

Khi kết thúc callback này, hãy nhớ return một View như những gì bạn đã thử nghiệm với ***FirstFragment*** và ***SecondFragment***. Lưu ý là chúng ta hoàn toàn có thể return ***null*** nếu Fragment không có UI.

[](https://i0.wp.com/yellowcodebooks.com/wp-content/uploads/2018/03/Screen-Shot-2018-03-06-at-16.43.44.png?ssl=1)

### onViewCreated()

Callback này được gọi ngay sau khi ***onCreateView()*** được gọi. Nó báo hiệu trạng thái Activity chứa nó được khởi tạo hoàn toàn. Tuy ít được sử dụng hơn các callback khác, nhưng bạn cũng có thể tận dụng nó để thay đổi giao diện hay các tương tác với Activity chứa Fragment này thoải mái được rồi.

### onStart()

Khi Fragment bắt đầu được nhìn thấy bởi người dùng và chuẩn bị nhận tương tác.

### onResume()

Người dùng hoàn toàn nhìn thấy và tương tác được với Fragment.

### onPause()

Callback này như một dấu hiệu cho thấy rằng người dùng đang rời khỏi Fragment hiện tại. Mặc dù không phải lúc nào ***onPause()*** được gọi là người dùng sẽ bái bai Fragment này. Nhưng bạn nên sao lưu các dữ liệu cần thiết của Fragment ở callback này, nhỡ đâu người dùng thực sự rời đi không quay lại thì sao.

### onStop()

Fragment chính thức không còn được nhìn thấy nữa.

### onDestroyView()

Chắc chắn là đối tượng View sẽ bị hủy ở callback này. Và do đó các khởi tạo view của bạn ở ***onCreateView()*** sẽ nhanh chóng không còn nữa.

Nếu như Fragment được đưa vào Back Stack, thì khi được lấy ra lại sau đó, callback ***onCreateView()*** sẽ được gọi lại.

### onDestroy()

Fragment đã sắp “chết”. Nhưng khác với Activity, khi [onDestroy() của Activity](https://yellowcodebooks.com/2017/10/19/android-bai-28-vong-doi-cua-mot-activity/" \l "onDestroy" \t "_blank) được gọi thì xem như Activity đã đến “cuối đời”. Còn với Fragment, callback này chỉ như một lời “nhắc nhở” về vận mệnh của Fragment mà thôi.

### onDetach()

Callback này gọi đến báo hiệu Fragment sẽ được gỡ khỏi Activity đang chứa nó. Kết thúc vòng đời của Fragment.

## **Các Trạng Thái Chính Trong Vòng Đời Fragment**

Bạn sẽ thấy các trạng thái chính này cũng không khác gì với Activity cả, nhưng hãy xem các tình huống cụ thể liên quan với nhau giữa Fragment và Activity như thế nào nhé.

### Hoạt Động (Active Hay Resume)

Khi Fragment được gắn vào Activity, được nhìn thấy và có thể tương tác được.

### Tạm Dừng (Pause)

Cũng khá giống với trạng thái [tạm dừng của Activity](https://yellowcodebooks.com/2017/10/19/android-bai-28-vong-doi-cua-mot-activity/" \l "Tam_Dung_Pause" \t "_blank). Tức là nếu Activity có chứa Fragment này bị che lấp bởi Activity khác (nhưng không bị che hoàn toàn, người dùng vẫn nhìn thấy được Activity bị che lấp, chỉ là không tương tác được), thì cả Activity và Fragment đó đều vào trạng thái tạm dừng.

### Dừng (Stop)

Cũng giống với [Activity](https://yellowcodebooks.com/2017/10/19/android-bai-28-vong-doi-cua-mot-activity/#Dung_Stop), Fragment bị dừng khi bị thành phần nào đó che khuất hoàn toàn. Hay bị gỡ ra khỏi Activity.

Dừng chưa phải là chấm hết cho đời sống của Fragment. Cụ thể là các trạng thái của nó vẫn còn được lưu trữ, để phòng trường hợp Fragment này được trở lại hiển thị cho người dùng.

### Chết (Dead)

Nếu Fragment bị gỡ ra khỏi Activity, nhưng không được đưa vào Back Stack trước đó, thì nó sẽ kết thúc vòng đời. Hoặc khi Activity chứa Fragment này bị gỡ khỏi Back Stack, Fragment cũng sẽ chết theo.