17/7

Change Detection:  là một cơ chế theo dõi sự thay đổi, cho phép nội dung ở phần giao diện luôn được đồng bộ với sự thay đổi trong model tương ứng. Sự thay đổi này có thể được phát sinh bởi những trường hợp sau:

* DOM events (click, hover, …)
* AJAX requests
* Timers (setTimers (), setInterval ())

View State: Mỗi view có một trạng thái nắm giữ vai trò rất quan trọng để Angular xác định xem có thực hiện change detection trên view và child view hay không. Một số state view:

1. FirstCheck
2. ChecksEnabled
3. Errored
4. Destroyed

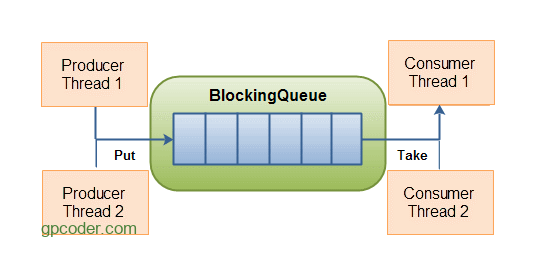
Promise: Được thiết kế để xử lý 1 tác vụ bất đồng bộ và trả về 1 giá trị, vì thế không thể sử dụng cho các bài toán để trả về nhiều giá trị, nếu muốn xử lý bài toán đó ta cần tạo ra nhiều object.

Directive:

Observable: Để xử lý được những tác vụ tác đồng bộ trả về nhiều giá trị trong tương lai.

* Producer: Công việc của producer là tạo data, đưa nó vào bộ đệm và bắt đầu lại.
* Customer: Công việc của customer là tiêu thụ dữ liệu (loại bỏ nó khỏi bộ đệm), từng phần một và xử lý nó. Customer và producer hoạt động song song với nhau.

Vấn đề là đảm bảo rằng nhà sản xuất không thể thêm dữ liệu vào bộ đệm nếu nó đầy và người tiêu dùng không thể xóa dữ liệu khỏi bộ đệm trống, đồng thời đảm bảo an toàn cho luồng (thread-safe).



Giai pháp để giải quyết vấn đề Producer/Customer

* Giải pháp cho nhà sản xuất là đi ngủ (wait) nếu bộ đệm đầy. Lần tiếp theo người tiêu dùng xóa một mục khỏi bộ đệm, nó đánh thức (notify) nhà sản xuất, bắt đầu đưa dữ liệu vào bộ đệm.
* Theo cách tương tự, người tiêu dùng có thể đi ngủ (wait) nếu thấy bộ đệm trống. Lần tiếp theo nhà sản xuất đưa dữ liệu vào bộ đệm, nó đánh thức (notify) người tiêu dùng đang ngủ (wait).
* Trong khi làm tất cả điều này, phải đảm bảo an toàn cho luồng (thread safe).