## Sự khác nhau giữa **Template-Driven Forms  và  Reactive Forms trong Angular**

**Một số sự khác biệt chính**

|  |  |  |
| --- | --- | --- |
|  | **Template-Driven Forms** | **Reactive Forms** |
| 1.Thiết lập mô hình biểu mẫu | Rõ ràng được tạo trong lớp thành phần | Ngụ ý, được tạo bởi các lệnh |
| 2.Mô hình dữ liệu | Có cấu trúc | Không có cấu trúc |
| 3.Khả năng dự đoán | Đồng bộ | Không đồng bộ |
| 4.Xác thực biểu mẫu | Chức năng | Chỉ thị |
| 5.Khả năng mở rộng | Bất biến | Có thể thay đổi |
| 6.Khả năng thay đổi | Quyền truy cập API cấp thấp | Tóm tắt trên APIS |

|  |  |
| --- | --- |
| **Template-Driven Forms ( Biểu mẫu theo hướng mẫu )** | **Reactive Forms ( Biểu mẫu phản ứng )** |
| - Dễ sử dụng | - Linh hoạt hơn, nhưng cần thực hành nhiều |
| - Thích hợp cho các kịch bản đơn giản, không thích hợp cho các kịch bản phức tạp | - Xử lý mọi tình huống phức tạp |
| - Tương tự như AngularJS | - Không có ràng buộc dữ liệu nào được thực hiện |
| - Liên kế dữ liệu hai chiều | - Mã thành phần nhiều hơn và đánh dấu HTML ít hơn |
| - Mã thành phần tối thiểu | - Biến đổi phản ứng có thể được thực hiện như:  + Xử lý một sự kiện dựa trên thời gian phát hành  + Xử lý các sự kiện khi các thành phần khác biệt cho đến khi thay đổi  + Thêm các yếu tố linh hoạt |
| - Tự động theo dõ bản mẫu và dữ liệu của nó |  |
| - Kiểm tra đơn vị khó khăn | - Kiểm tra đơn vị dễ dàng hơn |
| - Ít khả năng được mở rộng | - Có thể tái sử dụng |
| - Được nhâp bằng FormsModel | - Được nhập bằng ReactiveFormsModel |