

Reactive Forms

- Explicit management of data flowing between an non-UI data model and a UI-oriented form model
 - Create a tree of Angular form control objects in the component class
 - Bind them to the native form control elements in the template
 - Component class has immediate access to both data model and the form control structure
 - push data model values to form controls and pull user-changed values back out
- Reactive patterns, testing and validation

Reactive Forms

- + Values and Validity updates are always synchronous
- + Easier to unit test

Reactive Form Classes

- FormControl: powers individual form control, tracks the value and validation status
- FormGroup: group of FormControl instances
- AbstractControl: abstract base class for the form control classes
- FormArray: numerically indexed array of AbstractControl instances

Form Builder

- Import FormBuilder from @angular/forms
- Use FormBuilder to construct the reactive form
 - reduce repetition and clutter

```
this.feedbackForm = this.fb.group({  
  firstname: "",  
  lastname: "",  
  telnum: "",  
  email: "",  
  agree: false,  
  contacttype: 'None',  
  message: ""  
});
```

Populate Form Model

- Form model and Data Model are separate
- User changes flow from the DOM elements to the form model
- Two methods to populate the form model from the data model:
 - setValue(): assign every form control value at once
 - patchValue(): update specific form control value