In Reactive Forms Approach,

1. Create formGroupName, like for e.g. [formGroupName]=”form”,
2. Each form controls, should be defined using ‘formControlName’
3. On OnInit lifecycle hook, create the validators for all the form controls, for e.g., this.form = new FormGroup ({
   ```
               name: new FormControl(‘’, ..optional validators array);
               email: new FormControl(‘’, ..optional validators array);
    });
   ```
4. FormGroup, FormControl, Validators are need to be imported from ‘@angular/forms’
5. Validators can be verified using the existing methods/members like required, minLength, maxLength, pattern, email etc. for e.g.,

   ```
   name: new FormControl('', [Validators.required, Validators.minLength(4)]),
   email: new FormControl('',[Validators.required, Validators.email] )
   ```

6. For displaying interactive messages, in the mark up can use <formName>.get(‘formControlName’).hasError(‘<validator>’), for e.g.
   ```
    <div *ngIf="form.get('name').hasError('minlength')">
        Should be atleast 4 characters!
    </div>
    <div *ngIf="form.get('name').hasError('required')">
      Name is required!
    </div>
   ```

7. Note that, ‘minlength’ here is all small letters, unlike in component (which is camelCase), same applies for ‘maxlength’
8. ReactiveForms doesn’t need ‘name’ attribute as we are using formControlName attribute.
9. Every Form control has zero or more validators, zero or more custom validators,
10. Alternate syntax for one or more validators is passing an array to Validt
11. Combo validations for a group of controls (formGroup) is impossible in TDF, only achievable in RF (Reactive Forms)

- All http get, post return <Observable>

For multiple custom validations:
https://stackoverflow.com/questions/48322092/angular-4-two-or-more-custom-validation

1. Async Validators are third optional parameter for each form control, is used usually for detecting the duplicates on every keystroke, syntax would look like this,
   ```
   email: new FormControl(
          '',
           [
              Validators.required,
   ```

```
              Validators.email,
              this.customValidator1ForEmail,
              this.customValidator2ForEmail
            ],
            [
              //Async Validators
            ]
        )
```

2.