

# Day 12 - Angular & Asp in 21 days Handouts

▼ Status

## Day 12: Input Validation

```
API Input Validation
public record class BookDto(
    [Required] Guid FlightId ,
    [Required][StringLength(320, MinimumLength = 3)][EmailAddress] string PassengerEmail ,
    [Required][Range(1, 853)] byte NumberOfSeats
);

UI Input Validation
form = this.fb.group({
    number: [1, Validators.compose([Validators.required, Validators.min(1), Validators.max(853)])]
})
```

To validate API input use validation attributes from `System.ComponentModel.DataAnnotations` on your Data Transfer Objects.

```
using System.ComponentModel.DataAnnotations;
public record CustomerRegistrationDto(
    [Required][MinLength(2)][MaxLength(35)] string FullName
    , [Required][MinLength(3)][MaxLength(320)][EmailAddress] string Email);
```

Remember to add the `ApiController` attribute on top of the `Controller` .

To validate Reactive Forms in Angular, add the validation rules to the array that describes form group properties. If there are more than a single validation rule for a property, compose them as a single rule:

```
form = this.fb.group({
    fullName: ['', Validators.compose([Validators.required, Validators.minLength(2), Validators.maxLength(35)])],
    email: ['', Validators.compose([Validators.email, Validators.required, Validators.minLength(3), Validators.maxLength(320)])]
})
```

To display error messages whenever a user violates a validation rule, aside from checking the validity of the property, check to see if the user has changed anything or is leaving it without fixing the errors.

```
<input formControlName="email" type="text" />

<div *ngIf="form.controls.email.invalid && (form.controls.email.dirty || form.controls.email.touched)">
  <div *ngIf="form.controls.email.errors">Email is invalid</div>
</div>
```

You can also show a specific message per rule violation:

```
<div *ngIf="form.controls.fullName.invalid && (form.controls.fullName.dirty || form.controls.fullName.touched)">

  <div *ngIf="form.controls.fullName.errors?.['required']">
    Full Name is required.
  </div>
  <div *ngIf="form.controls.fullName.errors?.['minlength']">
    Full Name must be at least 2 characters long.
  </div>
  <div *ngIf="form.controls.fullName.errors?.['maxlength']">
```

```
    Full Name must be more than 35 characters long.  
  </div>  
  
</div>
```