



Estados de un Control Durante la Validación

La clase `AbstractControlDirective` contiene la definición de estados y operaciones básicas durante la validación de un control:

```
class AbstractControlDirective {  
  get control: AbstractControl | null  
  get value: any  
  get valid: boolean | null  
  get invalid: boolean | null  
  get pending: boolean | null  
  get disabled: boolean | null  
  get enabled: boolean | null  
  get errors: ValidationErrors | null  
  get pristine: boolean | null  
  get dirty: boolean | null  
  get touched: boolean | null  
  get status: string | null  
  get untouched: boolean | null  
  get statusChanges: Observable<any> | null  
  get valueChanges: Observable<any> | null  
  get path: string[] | null  
  reset(value: any = undefined): void  
  hasError(errorCode: string, path?: string[]): boolean  
  getError(errorCode: string, path?: string[]): any  
}
```

Miembros

- `get control: AbstractControl | null`

`FormControl`, `FormGroup` o `FormArray` que respalda esta directiva. La mayoría de las propiedades caen en esa instancia.

- `get value: any`

El valor almacenado en el control.

- `get valid: boolean | null`

Un control es `valid` cuando su `status === VALID`.

Para tener este estado, el control debe haber pasado todas sus verificaciones de validación.

- `get invalid: boolean | null`

Un control `invalid` es `invalid` cuando su `status === INVALID`.

Para tener este estado, el control debe haber fallado al menos una de sus verificaciones de validación.

- `get pending: boolean | null`

Un control está `pending` cuando su `status === PENDING`.
Para tener este estado, el control debe estar realizando una verificación de validación.

- `get disabled: boolean | null`

Un control está `disabled` cuando su `status === DISABLED`.
Los controles deshabilitados están exentos de las comprobaciones de validación y no están incluidos en el valor agregado de sus controles ancestrales.

- `get enabled: boolean | null`

¡Un control está `enabled` mientras su `status !== DISABLED`.
En otras palabras, tiene un estado de `VALID`, `INVALID` o `PENDING`.

- `get errors: ValidationErrors | null`

Devuelve cualquier error generado por la validación fallida. Si no hay errores, devolverá nulo.

- `get pristine: boolean | null`

Un control es `pristine` si el usuario no ha cambiado el valor en la interfaz de usuario desde que se desplegó.
Tenga en cuenta que los cambios programáticos en el valor de un control *no lo* marcarán como "sucio".

- `get dirty: boolean | null`

Un control está `dirty` si el usuario ha cambiado el valor en la interfaz de usuario.
Tenga en cuenta que los cambios programáticos en el valor de un control *no lo* marcarán como "sucio".

- `get touched: boolean | null`

Un control se marca como `touched` una vez que el usuario ha tocado este control.

- `get status: string | null`

- `get untouched: boolean | null`

Un control posee el estado `untouched` si el usuario aún no lo ha tocado.

- `get statusChanges: Observable<any> | null`

Emite un evento cada vez que se vuelve a calcular el estado de validación del control.

- `get valueChanges: Observable<any> | null`

Emite un evento cada vez que cambia el valor del control, en la interfaz de usuario o programáticamente.

- `get path: string[] | null`

Devuelve una matriz que representa la ruta del formulario de nivel superior a este control. Cada índice es el nombre de la cadena del control en ese nivel.

- `reset(value: any = undefined): void`

Restablece el control de formulario. Esto significa por defecto:

- está marcado como `pristine`
- está marcado como `untouched`
- el valor se establece en nulo

Para obtener más información, vea [AbstractControl](#).

- `hasError(errorCode: string, path?: string[]): boolean`

Devuelve verdadero si el control con la ruta dada tiene el error especificado. De lo contrario, devuelve falso.

Si no se proporciona ninguna ruta, verifica el error en el control actual.

- `getError(errorCode: string, path?: string[]): any`

Devuelve datos de error si el control con la ruta dada tiene el error especificado. De lo contrario, devuelve null o undefined.

Si no se proporciona ninguna ruta, verifica el error en el control actual.