**blacklist**

从用户输入的值中删除字符。

如果你想删除一些特定的文本，然后在黑名单中应用。每当用户在相应的HTML控件中输入值时，它就会清除该值并在FormControl/Model属性中设置该值。

export class User {

@prop()

@blacklist('abc' )

freeText: string;

}

export class BlacklistAddComponent implements OnInit {

userFormGroup: FormGroup

constructor( private formBuilder: RxFormBuilder ) { }

ngOnInit() {

let user = new User();

this.userFormGroup = this.formBuilder.formGroup(user);

}

}

<div>

<form [formGroup]="userFormGroup">

<div class="form-group">

<label>Free Text</label>

<input type="text" formControlName="freeText" class="form-control" />

Transformed Value : {{userFormGroup["modelInstance"]["freeText"]}}

</div>

<button [disabled]="!userFormGroup.valid" class="btn btn-primary">Submit</button>

</form>

</div>

**custom**

您可以根据您的业务逻辑清除这些值。

export class User {

@prop()

@sanitize({custom:(x)=> x == '1K' ? 1000 : x })

amount: string;

}

**escape**

当用户输入输入时转义字符串。

export class User {

@prop()

@escape()

freeText: string;

}

**prefix**

Allows you to add prefix to the input.

export class User {

@prop()

@prefix('Dr' )

doctorName: string;

}

**ltrim**

修剪输入左侧的字符。

**rtrim**

修剪输入右侧的字符。

**suffix**

允许您向输入添加后缀。

**toBoolean**

将输入转换为布尔值。

对活动字段应用布尔消毒

对strictActive应用布尔消毒。如果输入值是布尔值，则严格应用于此。

严格toBoolean例子

让我们创建一个用户模型，并在模型中定义一个属性'strictActive'。

export class User {

@prop()

@toBoolean(true )

strictActive: boolean;

}

**完整的toBoolean示例**

这个完整因子示例包括所有严格的和不严格的属性，将满足场景1和2的要求。

export class User {

@prop()

@toBoolean()

active: boolean;

@prop()

@toBoolean(true )

strictActive: boolean;

}

**toDate**

将输入转换为日期，如果输入不是日期则为空。

export class User {

@prop()

@toDate()

dob: Date;

}

**toDouble**

将输入转换为浮点数，如果输入不是浮点数，则转换为NAN。

**toFloat**

将输入转换为浮点数，如果输入不是浮点数，则转换为NAN。

**toInt**

将输入转换为整数，如果输入不是整数，则转换为NAN。

**toString**

将输入转换为字符串

**trim**

从输入中修剪字符。

**whitelist**

删除没有出现在白名单中的字符。