Angular Forms教程

Angular表单中的StatusChanges

<https://www.tektutorialshub.com/angular/statuschanges-in-angular-forms/>

StatusChanges是当Angular计算FormControl、FormGroup或FormArray的验证状态时，由Angular表单引发的一个事件。它返回一个可观察对象，这样你就可以订阅它了。可观察对象获得控件的最新状态。Angular会对控件所做的每一个更改运行验证检查。它还生成一个验证错误列表，在这种情况下状态变为INVALID。如果没有错误，则状态变为VALID

**如何使用StatusChanges**Angular Forms有三个构建块。FormControl, FormGroup和FormArray。所有这些控件都扩展了AbstractControl基类。AbstractControl基类实现了StatusChanges事件

我们可以通过获取控件的引用并订阅它来订阅StatusChanges，如下所示

this.reactiveForm.get("firstname").statusChanges.subscribe(newStaus => {

console.log('firstname status changed')

console.log(newStaus)

})

您还可以订阅顶级表单，如下所示。

this.reactiveForm.statusChanges.subscribe(newStaus => {

console.log('form Status changed event')

console.log(newStaus)

})

**StatusChanges例子**

创建一个响应式表单，如下所示

reactiveForm = new FormGroup({

firstname: new FormControl('', [Validators.required]),

lastname: new FormControl(),

address: new FormGroup({

city: new FormControl(),

street: new FormControl(),

pincode: new FormControl()

})

})

**StatusChanges of FormControl**

你可以订阅一个FormControl的StatusChanges，如下所示。在newStatus变量中，我们将获得名字的最新状态。你也可以使用this.reactiveForm.get("firstname").status获取名字的最新状态

this.reactiveForm.get("firstname").statusChanges.subscribe(newStatus => {

console.log('firstname status changed')

console.log(newStatus) //latest status

console.log(this.reactiveForm.get("firstname").status) //latest status

})

但是，此时顶级表单还没有更新，因此this. reactiveform .status仍然显示名字和表单的旧状态。

firstname的statusChanges事件在新状态更新之后，但在更改被发泡到其父状态之前立即触发。因此this.reactiveForm.status仍然显示旧的状态。

this.reactiveForm.get("firstname").statusChanges.subscribe(newStatus=> {

console.log('firstname status changed')

console.log(newStatus) //latest status

console.log(this.reactiveForm.get("firstname").status) //latest status

console.log(this.reactiveForm.status) //Previous status

})

您可以通过使用setTimeout等待下一个滴答来解决这个问题，如下所示。

this.reactiveForm.get("firstname").statusChanges.subscribe(newStatus=> {

console.log('firstname status changed')

console.log(newStatus) //latest status

console.log(this.reactiveForm.get("firstname").status) //latest status

console.log(this.reactiveForm.status) //Previous status

setTimeout(() => {

console.log(this.reactiveForm.status) //latest status

})

})

**StatusChanges of FormGroup**

每当计算出FormGroup或FormArray的任何子控件的状态时，就会触发StatusChanges事件。例如，以下的StatusChanges将被触发，即使当城市，状态和pin码的状态被计算。

this.reactiveForm.get("address").statusChanges.subscribe(newStaus => {

console.log('address status changed')

console.log(newStaus)

})

**StatusChanges of Form**

下面的示例显示我们可以订阅对整个表单所做的更改。

this.reactiveForm.statusChanges.subscribe(newStaus => {

console.log('form status changed')

console.log(newStaus)

})

**emitEvent & StatusChanges**

即使当angular通过UI或编程方式计算控件的状态时，也会触发statusChanges事件。在某些情况下，您可能不希望引发statusChanges事件。为此，我们可以使用emitEvent: false

在下例中，statusChanges事件根本没有被触发，即使更改了名字的值使其和表单INVALID。

this.reactiveForm.get("firstname").setValue("", { emitEvent: false });

你可以在setValue, patchValue, markAsPending, disable, enable, updateValueAndValidity和setErrors方法中使用emitEvent: false。

**onlySelf & StatusChanges**

当onlySelf: true时，更改只会影响这个FormControl，而不会影响到它的父控件。因此父FormGroup的StatusChanges事件不会被触发。

例如，下面的代码将导致名字的StatusChanges。但不是它的父元素(即顶层形式)

this.reactiveForm.get("firstname").setValue("", { onlySelf: true });

你可以在setValue, patchValue, markasunchanged, markAsPristine, markAsPending, disable, enable和updateValueAndValidity方法中使用onlySelf: true