

Differences between ngForm and FormControl

Asked 1 year, 9 months ago Active 9 months ago Viewed 8k times



I'm working with forms on angular4 and I don't recognize what is the differences using `NgForm` and `FormGroup` handling errors and inputs validators. is there any big differences handling forms??

6

Thanks!!



6

asked Jan 8 '18 at 4:20

 [Kenry Sanchez](#)
883 1 10 19

2 Answers



23

NgForm does something handy but non-obvious: it includes the **form** tag in its selector (instead of requiring you to explicitly add **ngForm** as an attribute). What this means is that if you import **FormsModule**, **NgForm** will get automatically attached to any tags you have in your view. This is really useful but potentially confusing because it happens behind the scenes.



There are two important pieces of functionality that **NgForm** gives us:



1. A **FormGroup** named **ngForm**
2. A (**ngSubmit**) output

You can see that we use both of these in the tag in our view:

```
<form #f="ngForm" (ngSubmit)="onSubmit(f.value)"
```

First we have **#f="ngForm"**. The **#v=thing** syntax says that we want to create a local variable for this view. Here we're creating an alias

By using our site, you acknowledge that you have read and understand our [Cookie Policy](#), [Privacy Policy](#), and our [Terms of Service](#).

And what type of object is **ngForm**? It is a **FormGroup**. That means we can use **f** as a **FormGroup** in our view. And that's exactly what we do in the **(ngSubmit)** output.

We bind to the **ngSubmit** action of our form by using the syntax: **(ngSubmit)="onSubmit(f.value)"**.

- **(ngSubmit)** - comes from **NgForm**
- **onSubmit()** - will be implemented in our component definition class
- **f.value** - **f** is the **FormGroup** that we specified above. And **.value** will return the key/value pairs of this **FormGroup**

For more detail please go through this [link](#).

edited Jan 4 at 6:20



Derek 朕會功夫

65.5k 38 145 200

answered Jan 8 '18 at 4:59



Dipak Delvadiya

2,478 2 15 32

Thanks man, this is the answer I was looking for. Thanks a lot!! – [Kenry Sanchez](#) Jan 8 '18 at 14:13

"And what type of object is ngForm? It is a FormGroup." - This is not true. The variable "f" here will be of type NgForm (and not FormGroup). To get the FormGroup object use f.form. – [RajV](#) Mar 28 '18 at 1:15

▲ If I am reading this right, you are bumping up against the difference between Template Driven Forms and Reactive Forms.

7

Template Driven Forms are as it sounds; Template Driven. Most if not all of the logic lies in the template. Template driven forms are great for simple forms that don't involve a lot of complicated logic. [Template Driven Forms](#)

▼ **Reactive Forms** take a different approach by defining the form logic within the model. Reactive Forms are typically a little bit more advanced and allow quite a bit of customization in terms of functionality and flexibility. [Reactive Forms](#)

answered Jan 8 '18 at 4:44



josh Rathke

4,261 5 16 32

It's a good answer but it's not what I'm looking for :D thanks!!! I know that differences – [Kenry Sanchez](#) Jan 8 '18 at 14:12

1 hmmm...I think this answer is better. – [Sifena](#) Mar 21 '18 at 10:15

By using our site, you acknowledge that you have read and understand our [Cookie Policy](#), [Privacy Policy](#), and our [Terms of Service](#).

By using our site, you acknowledge that you have read and understand our [Cookie Policy](#), [Privacy Policy](#), and our [Terms of Service](#).