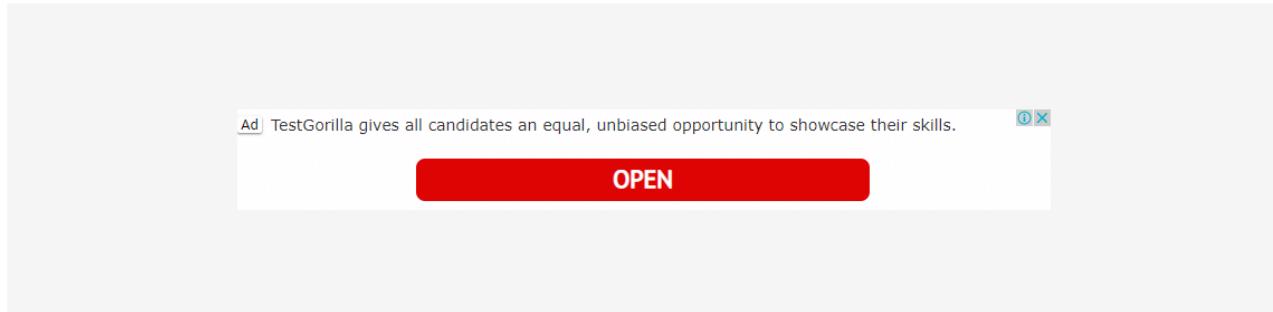




PUBLISHED: NOVEMBER 07 2018

## Angular 7 - Reactive Forms Validation Example



Example built with Angular 7.0.1

Other versions available:

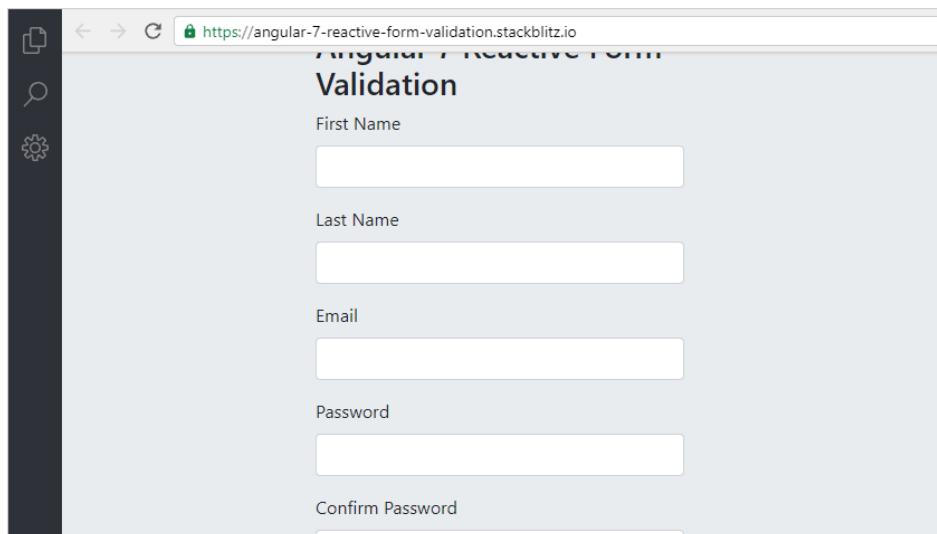
- [Angular Reactive Forms: Angular 10, 9, 8, 6](#)
- [Angular Template-Driven Forms: Angular 10, 9, 8, 7, 6](#)
- [Blazor: Blazor WebAssembly](#)
- [Next.js: Next.js](#)
- [React + Formik: Formik 2, 1](#)
- [React Hook Form: React Hook Form 7, 6](#)
- [Vue + VeeValidate: Vue 3, 2](#)
- [Vue + Vuelidate: Vue 2](#)

This is a quick example of how to setup form validation in Angular 7 using Reactive Forms. The example is a simple registration form with pretty standard fields for first name, last name, email, password and confirm password. All fields are required, the email field must be a valid email address and the password field must have a min length of 6. There's also a custom validator called `MustMatch` which is used to validate that the confirm password and password fields match.

I've setup the form to validate on submit rather than as soon as each field is changed, this is implemented with a 'submitted' field in the app component that is set to true when the form is submitted for the first time.

Styling of the example is all done with Bootstrap 4 CSS.

Here it is in action:



SUBSCRIBE FOR UPDATES



Twitter



YouTube



Github



RSS

HIRE A ANGULAR 7 EXPERT

fiverr.

Find a Freelancer

What service are you looking for?

Try "Angular 7"

Search for Freelancers

LEVEL UP YOUR SKILLS

Udemy™

Top online courses in  
Web Development



Angular 7 - Reactive Forms Validation Example

JasonWatmore.com

Console

Edit on [StackBlitz](#)

(See on StackBlitz at <https://stackblitz.com/edit/angular-7-reactive-form-validation>)

## Reactive Forms Validation App Component

The app component defines the form fields and validators for our registration form using an Angular FormBuilder to create an instance of a FormGroup that is stored in the registerForm property. The registerForm is then bound to the form in the template below using the [FormGroup] directive.

I also added a getter 'f' as a convenience property to make it easier to access form controls from the template. So for example you can access the confirmPassword field in the template using

`f.confirmPassword` instead of `registerForm.controls.confirmPassword`.

```
import { Component, OnInit } from '@angular/core';
import { FormBuilder, FormGroup, Validators } from '@angular/forms';

// import custom validator to validate that password and confirm password fields match
import { MustMatch } from './helpers/must-match.validator';

@Component({
  selector: 'app',
  templateUrl: 'app.component.html'
})

export class AppComponent implements OnInit {
  registerForm: FormGroup;
  submitted = false;

  constructor(private formBuilder: FormBuilder) { }

  ngOnInit() {
    this.registerForm = this.formBuilder.group({
      firstName: ['', Validators.required],
      lastName: ['', Validators.required],
      email: ['', [Validators.required, Validators.email]],
      password: ['', [Validators.required, Validators.minLength(6)]],
      confirmPassword: ['', Validators.required]
    }, {
      validator: MustMatch('password', 'confirmPassword')
    });
  }

  // convenience getter for easy access to form fields
  get f() { return this.registerForm.controls; }

  onSubmit() {
    this.submitted = true;

    // stop here if form is invalid
    if (this.registerForm.invalid) {
      return;
    }

    alert('SUCCESS!! :-)\n\n' + JSON.stringify(this.registerForm.value))
  }
}
```

Get started



DESIGN MADE EASY

Canva

Empowering the world to design.

FREE SITE AUDIT

SEMrush

Run a Full Website Scan in Minutes

SUPPORT MY WORK

BECOME A PATRON

Donate with

## Reactive Forms Validation App Template

The app component template contains all the html markup for displaying the example registration form in your browser. The form element uses the `[formGroup]` directive to bind to the `registerForm` FormGroup in the app component above.

The form binds the form submit event to the `onSubmit()` handler in the app component using the Angular event binding `(ngSubmit)="onSubmit()"`. Validation messages are displayed only after the user attempts to submit the form for the first time, this is controlled with the `submitted` property of the app component.

```
<!-- main app container -->
<div class="jumbotron">
  <div class="container">
    <div class="row">
      <div class="col-md-6 offset-md-3">
        <h3>Angular 7 Reactive Form Validation</h3>
        <form [formGroup]="registerForm" (ngSubmit)="onSubmit()">
          <div class="form-group">
            <label>First Name</label>
            <input type="text" formControlName="firstName" class="form-control" [ngClass]="{{ 'is-invalid' if f.firstName.errors && f.firstName.submitted }}>
              <div *ngIf="f.firstName.errors.required">First Name is required</div>
            </div>
          </div>
          <div class="form-group">
            <label>Last Name</label>
            <input type="text" formControlName="lastName" class="form-control" [ngClass]="{{ 'is-invalid' if f.lastName.errors && f.lastName.submitted }}>
              <div *ngIf="f.lastName.errors.required">Last Name is required</div>
            </div>
          </div>
          <div class="form-group">
            <label>Email</label>
            <input type="text" formControlName="email" class="form-control" [ngClass]="{{ 'is-invalid' if f.email.errors && f.email.submitted }}>
              <div *ngIf="f.email.errors.required">Email is required</div>
              <div *ngIf="f.email.errors.email">Email must be a valid email address</div>
            </div>
          </div>
          <div class="form-group">
            <label>Password</label>
            <input type="password" formControlName="password" class="form-control" [ngClass]="{{ 'is-invalid' if f.password.errors && f.password.submitted }}>
              <div *ngIf="f.password.errors.required">Password is required</div>
              <div *ngIf="f.password.errors.minLength">Password must be at least 6 characters long</div>
            </div>
          </div>
          <div class="form-group">
            <label>Confirm Password</label>
            <input type="password" formControlName="confirmPassword" class="form-control" [ngClass]="{{ 'is-invalid' if f.confirmPassword.errors && f.confirmPassword.submitted }}>
              <div *ngIf="f.confirmPassword.errors.required">Confirm Password is required</div>
              <div *ngIf="f.confirmPassword.errors.mustMatch">Passwords must match</div>
            </div>
          </div>
          <div class="form-group">
            <button class="btn btn-primary">Register</button>
          </div>
        </form>
      </div>
    </div>
  </div>
</div>
```

## Reactive Forms Custom "Must Match" Validator

The custom `MustMatch` validator is used in this example to validate that both of the password fields - `password` and `confirmPassword` - are matching. However it can be used to validate that any pair of fields

is matching (e.g. email and confirm email fields).

It works slightly differently than a typical custom validator because I'm setting the error on the second field instead of returning it to be set on the `FormGroup`. I did it this way because I think it makes the template a bit cleaner and more intuitive, the `mustMatch` validation error is displayed below the `confirmPassword` field so I think it makes sense that the error is attached to the `confirmPassword` form control.

```
import { FormGroup } from '@angular/forms';

// custom validator to check that two fields match
export function MustMatch(controlName: string, matchingControlName: string) {
  return (formGroup: FormGroup) => {
    const control = formGroup.controls[controlName];
    const matchingControl = formGroup.controls[matchingControlName];

    if (matchingControl.errors && !matchingControl.errors.mustMatch) {
      // return if another validator has already found an error on the matchingControl
      return;
    }

    // set error on matchingControl if validation fails
    if (control.value !== matchingControl.value) {
      matchingControl.setErrors({ mustMatch: true });
    } else {
      matchingControl.setErrors(null);
    }
  }
}
```



## Reactive Forms Validation App Module

There isn't much going on in the app module other than the standard stuff, the main thing you need to remember for using reactive forms in Angular is to import the `ReactiveFormsModule` from `'@angular/forms'` and include it in the imports array of the `@NgModule` decorator.

```
import { NgModule } from '@angular/core';
import { BrowserModule } from '@angular/platform-browser';
import { ReactiveFormsModule } from '@angular/forms';

import { AppComponent } from './app.component';

@NgModule({
  imports: [
    BrowserModule,
    ReactiveFormsModule
  ],
  declarations: [
    AppComponent
  ],
  bootstrap: [AppComponent]
})

export class AppModule { }
```





## Subscribe or Follow Me For Updates

Subscribe to my YouTube channel or follow me on Twitter, Facebook or GitHub to be notified when I post new content.

- Subscribe on YouTube at <https://www.youtube.com/JasonWatmore>
- Follow me on Twitter at [https://twitter.com/jason\\_watmore](https://twitter.com/jason_watmore)
- Follow me on Facebook at <https://www.facebook.com/Jason-Watmores-Blog>
- Follow me on GitHub at <https://github.com/cornflourblue>
- Feed formats available: [RSS](#), [Atom](#), [JSON](#)

### Other than coding...

I'm currently attempting to travel around Australia by motorcycle with my wife Tina on a pair of Royal Enfield Himalayans. You can follow our adventures on YouTube, Instagram and Facebook.



- Subscribe on YouTube at <https://www.youtube.com/TinaAndJason>
- Follow us on Instagram at <https://www.instagram.com/tinaandjason>
- Follow us on Facebook at <https://www.facebook.com/TinaAndJasonVlog>

## Need Some Angular 7 Help?

Search [fiverr](#) to find help quickly from experienced Angular 7 developers.

Tags: [Angular 7](#), [Angular 2](#), [TypeScript](#), [Validation](#)

Share:



## More Angular 7 Posts

- [Angular - HTTP POST Request Examples](#)
- [Angular - HTTP GET Request Examples](#)
- [Angular + Webpack - How to add global CSS styles to Angular with webpack](#)
- [Angular 7 Tutorial Part 7 - Migrating to an Angular CLI Project](#)
- [Angular 7 Tutorial Part 6 - Home Page & Alert Component](#)
- [Angular 7 Tutorial Part 5 - Registration Form & User Service](#)
- [Angular 7 Tutorial Part 4 - Login Form, Authentication Service & Route Guard](#)
- [Angular 7 - Mock Backend Example for Backendless Development](#)
- [Angular 7 Tutorial Part 3 - Add Routing & Multiple Pages](#)
- [Angular 7 Tutorial Part 2 - Create Base Project Structure & Webpack Config](#)

[Show more...](#)



Adobe Creative Cloud  
Take your ideas to new places.

Join now



What do you think?

148 Responses

 Upvote

 Funny

 Love

 Surprised

 Angry

 Sad

23 Comments

Jason Watmore's Blog

 Disqus' Privacy Policy

 Login

 Recommend 3

 Tweet

 Share

Sort by Best



Join the discussion...

LOG IN WITH



OR SIGN UP WITH DISQUS 

Name



**ArnaudPirates** • 3 years ago

Great tutorial ! Thanks

7 ^ | v · Reply · Share >



**Manpreet Singh** • 3 years ago

super and easily away to learn thanku so much

2 ^ | v · Reply · Share >



**sonam** • 3 years ago

For to mustMatchValidator, it is not showing any error but Whether I type same password or two different password still then it is successfully submitting the form.

2 ^ | v 1 · Reply · Share >



**Daya Choudekar** • 2 years ago

its help me a log sir. thanks

1 ^ | v · Reply · Share >



**Satyam Thakare** • 2 years ago

Hi Jason, Thanks for the tutorial.

I have two form in single component and I have created two form builder to apply validation, But when I click first form the validation is activate for second form as well. How to deal with this? any suggestions?

1 ^ | v · Reply · Share >



**sffasfdfafa** • 2 years ago

wwwooooooooooooowwwwwwwwwwwwwwwwwww

1 ^ | v · Reply · Share >



**Julian Shikui** • 3 years ago

Thanks for the great tutorial!

1 ^ | v · Reply · Share >



**shyam panchal** • 3 years ago

Thanks a lot ! Saved me with my assignment !

1 ^ | v · Reply · Share >



**iamjayfer** • 3 years ago

heart heart .. thanks for this <3

1 ^ | v · Reply · Share >



**Subhashi** • a month ago

Thank you. It worked for me!

^ | v · Reply · Share >



**d** • 6 months ago

cool!

^ | v · Reply · Share >



**Eshwar** • a year ago

I was trying to follow the exact same steps on mine but whenever I enter the same password on both the boxes, I get the error cannot read property must match of null. Any suggestions on how to solve this?

^ | v · Reply · Share >



**Eric Osman** • 2 years ago

I needed to change the

selector: 'app'

to

selector: 'app-root'

is it just me, or is this an error in your example?

Thanks.

^ | v · Reply · Share >



**onozor** • 2 years ago

Well done!!



Let your `this.submitted = true;` come below the the

```
if (this.registerForm.invalid) {
  return;
}
```

Otherwise you might end up having an endless loader when there's an error.

[^](#) | [v](#) - Reply · Share >



**Jason Watmore** → onozor 2 years ago

Hi Onozor, the `this.submitted = true;` needs to be above for the error messages to display when the form is invalid.

For a loader I would recommend setting another property `this.loading = true;` below the `registerForm.invalid` check so the loader only appears if the form is valid.

Cheers,  
Jason

[^](#) | [v](#) - Reply · Share >



**donyy gunawan** 2 years ago · edited

Hi Jason, for email validation when type test@test it's assigned as valid, why?  
edited - oh it's seems not related to angular it self <https://stackoverflow.com/q...>

[^](#) | [v](#) - Reply · Share >



**sffasfdfafa** 2 years ago

wooooooooooooowww!!!!!!  
(^\_^)  
[^](#) | [v](#) - Reply · Share >



**Parth Savaliya** 3 years ago

Great tutorial.It helped me a lot but how to compare multiple dates using your way? For example there are four fields registration start date, registration end date, tournament start date, tournament end date and They should be ascending way. any suggestion?

[^](#) | [v](#) - Reply · Share >



**codesquery** 3 years ago

getting this error

formGroup expects a FormGroup instance. Please pass one in.

[^](#) | [v](#) - Reply · Share >



**Bujar Deari** 3 years ago

Identifier 'firstName' is not defined. '\_\_type' does not contain such a member

[^](#) | [v](#) - Reply · Share >



**sonam** → Bujar Deari 3 years ago

use `*ngIf="['[firstName'].errors.required"`

[1](#) [^](#) | [v](#) - Reply · Share >



**suckerp** 3 years ago

Thanks for the tutorial. I have an additional question.

I know I can disable the register button until everything is valid by adding `[disabled] = "IregisterForm.valid` into the button-tag.  
But is it also possible to track the status of valid and auto-submit the form when valid is true?

[^](#) | [v](#) - Reply · Share >



**Subramanian P** 3 years ago

Good and simple.

one small error on mustMatchValidator, trows error when password matches, solved by matchingControl.setErrors({mustMatch: false});

[^](#) | [v](#) - Reply · Share >



Subscribe

Add Disqus to your site

Do Not Sell My Data

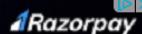
**DISQUS**



## Accept payments & settle them with a click

Grow your business with Payment Buttons

[Get started](#)




**ABOUT**

**MONTHS**

**TAGS**

I'm a web developer by day, a father by night and co-founder of [Point Blank Development](#), I've been building websites and web applications in Sydney since 1998.

Other than coding, I'm currently attempting to travel around Australia by motorcycle with my wife Tina, you can follow our adventure on [YouTube](#) and [Instagram](#).

Support me on: [Patreon](#)

Find me on:

Subscribe to Feed: [RSS](#), [Atom](#), [JSON](#)

[Code Snippets](#)

October (8)  
September (30)  
August (12)  
July (4)  
June (7)  
May (8)  
April (6)  
  
2020  
2019  
2018  
2017  
2016  
2015  
2014  
2013  
2012  
2011

Angular 5, Angular 6, Angular 7, Angular 8, Angular 9, Angular Directive, Angular UI Router, AngularJS, Animation, ASP.NET, ASP.NET Core, ASP.NET Web API, Authentication and Authorization, AWS, Axios, Azure, Basic Authentication, Blazor, Bootstrap, C#, Chai, CKEditor, CSS3, DDD, Deployment, Design Patterns, Dynamic LINQ, EF Core, ELMAH, ES6, Exchange, Facebook, Fetch, Fluent NHibernate, Formik, Google Analytics, Google API, Google Maps API, Google Plus, Heroku, HTML5, HTTP, IIS, Insecure Content, Instagram API, Ionic Framework, iOS, iPhone, JavaScript, jQuery, JWT, LinkedIn, LINQ, Login, MEAN Stack, MERN Stack, MEVN Stack, Mocha, Modal, MongoDB, Moq, MVC, MVC5, MySQL, .NET, NextJS, NGINX, ngMock, NHibernate, Ninject, NodeJS, npm, Pagination, Pinterest, Razor Pages, React, React Hook Form, React Hooks, Recoil, Redmine, Redux, Registration, Repository, RxJS, Security, Sequelize, Shell Scripting, Sinon, SinonJS, SSH, TDD, Terraform, Twitter, TypeScript, Ubuntu, Umbraco, Unit of Work, Unit Testing, URL Rewrite, Validation, Vanilla JS, VS Code, Vue, Vue 3, Vuex, Webpack, Windows Server 2008

Powered by [MEANie](#) | Hosted on [DigitalOcean](#) | Optimized with [Semrush SEO Tools](#)

© 2021 JasonWatmore.com