Custom Validators

https://dzone.com/articles/how-to-create-custom-validators-in-angular

import {AbstractControl, ValidatorFn} from '@angular/forms';

export function ageRangeValidator(control: AbstractControl): { [key: string]: boolean } | null {

  if (control.value !== undefined && (isNaN(control.value) || control.value < 18 || control.value > 45)) {

      return { 'ageRange': true };

  }

  return null;

}

function ageRangeValidator(min: number, max: number): ValidatorFn {

return (control: AbstractControl): { [key: string]: boolean } | null => {

if (control.value !== undefined && (isNaN(control.value) || control.value < min || control.value > max)) {

return { 'ageRange': true };

}

return null;

};

}

min = 10;

max = 20;

ngOnInit() {

this.loginForm = new FormGroup({

email: new FormControl(null, [Validators.required]),

password: new FormControl(null, [Validators.required, Validators.maxLength(8)]),

age: new FormControl(null, [ageRangeValidator(this.min, this.max)])

});

}

import {AbstractControl, ValidatorFn , FormGroup} from '@angular/forms';

export function ageRangeValidator(control: AbstractControl): { [key: string]: boolean } | null {

  if (control.value !== undefined && (isNaN(control.value) || control.value < 18 || control.value > 45)) {

      return { 'ageRange': true };

  }

  return null;

}

export function patternValidator(): ValidatorFn {

  return (control: AbstractControl): { [key: string]: any } => {

    if (!control.value) {

      return null;

    }

    const regex = new RegExp('^(?=.\*?[A-Z])(?=.\*?[a-z])(?=.\*?[0-9]).{8,}$');

    const valid = regex.test(control.value);

    return valid ? null : { invalidPassword: true };

  };

}

export function MatchPassword(password: string, confirmPassword: string) {

  return (formGroup: FormGroup) => {

    const passwordControl = formGroup.controls[password];

    const confirmPasswordControl = formGroup.controls[confirmPassword];

    if (!passwordControl || !confirmPasswordControl) {

      return null;

    }

    if (confirmPasswordControl.errors && !confirmPasswordControl.errors.passwordMismatch) {

      return null;

    }

    if (passwordControl.value !== confirmPasswordControl.value) {

      confirmPasswordControl.setErrors({ passwordMismatch: true });

    } else {

      confirmPasswordControl.setErrors(null);

    }

  }

}

export function userNameValidator(userControl: AbstractControl) {

  return new Promise(resolve => {

    setTimeout(() => {

      if (this.validateUserName(userControl.value)) {

        resolve({ userNameNotAvailable: true });

      } else {

        resolve(null);

      }

    }, 1000);

  });

}

export function validateUserName(userName: string) {

  const UserList = ['ankit', 'admin', 'user', 'superuser'];

  return (UserList.indexOf(userName) > -1);

}

<https://www.google.com/amp/s/www.freecodecamp.org/news/how-to-validate-angular-reactive-forms/amp/>