1. 表单异步验证 - 服务

private createRegistrationForm(): FormGroup {  
 return this.\_fb.group(  
 {  
 username: ['', Validators.*required*, this.\_registerService.usernameUnique.bind(this.\_registerService)],  
 }  
 );  
}

@Injectable({ providedIn: 'root' })  
export class RegisterService {  
 usernameUnique(field: AbstractControl): Observable<ValidationErrors | null> {  
 return of(field.value).pipe(  
 delay(20000),  
 map(value => value >= 0 ? null : {usernameUnique: " you're not authorized to work"}));  
 }

1. 表单异步验证 - 自定义验证器

private createRegistrationForm(): FormGroup {  
 return this.\_fb.group(  
 {  
 username: ['', Validators.*required*,  
 this.\_usernameUniqueValidator.validate.bind(this.\_usernameUniqueValidator)],  
 }  
 );  
}

@Injectable({ providedIn: 'root' })  
export class UsernameUniqueValidator implements AsyncValidator {  
 constructor() {}  
 validate(field: AbstractControl): Promise<ValidationErrors | null> | Observable<ValidationErrors | null> {  
 return of(field.value).pipe(  
 delay(20),  
 map(value => value >= 0 ? null : {usernameUnique: " you're not authorized to work"}));  
 }