

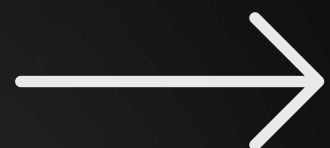
MAÎTRISE DEBOUNCETIME AVEC RXJS ET ANGULAR

Contrôle les événements rapides.
Améliore l'expérience utilisateur.



Elodie TURLIER

Angular, IA, agilité &
storytelling : construire des apps
prêtes à survivre à l'apocalypse
digitale



COMPRENDRE DEBOUNCETIME

debounceTime(ms) est un opérateur RxJS qui permet de **contrôler la fréquence d'émission** d'un Observable.

Il attend qu'un certain délai se soit écoulé sans nouvelle émission avant de transmettre la dernière valeur.

Si une nouvelle valeur est émise avant la fin du délai
→ compteur remis à zéro.

Idéal pour :

recherche, autocomplétion, ou tout événement déclenché trop souvent.



Elodie TURLIER

Angular, IA, agilité & storytelling



POURQUOI L'UTILISER ?

- **Optimisation des performances** : Évite d'envoyer trop de requêtes (ex : API) lors de la saisie utilisateur.
- **Expérience utilisateur améliorée** : Ne déclenche l'action qu'une fois l'utilisateur arrêté de taper.
- **Réduction de la charge serveur** : Limite le nombre d'appels réseau ou traitements coûteux.

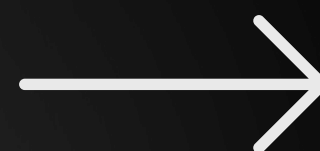
Cas d'usage typiques :

- Champs de recherche
- Auto-suggestions
- Saisie clavier rapide



Elodie TURLIER

Angular, IA, agilité & storytelling



SYNTAXE

```
import { debounceTime } from 'rxjs';  
  
observable$.pipe(  
  debounceTime(300)  
)
```

Ici, **300 ms** sans nouvelle émission
→ la dernière valeur est transmise.



Elodie TURLIER

Angular, IA, agilité & storytelling



EXEMPLE AVEC REACTIVE FORMS



```
this.searchControl.valueChanges.pipe(  
  debounceTime(300),           // ⌚ attend 300 ms après la dernière touche  
  distinctUntilChanged()      // 🚫 ignore les valeurs répétées  
)  
.subscribe(value => {  
  this.search(value);         // ✅ exécute la recherche  
});
```

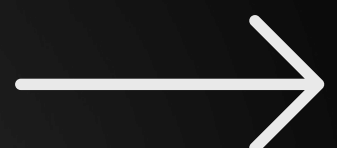
Cet exemple évite de lancer une recherche à chaque frappe, n'agissant que lorsque l'utilisateur a fini de taper

Parfait pour les champs de recherche dans Angular.



Elodie TURLIER

Angular, IA, agilité & storytelling



EXEMPLE AVEC FROMEVENT SUR UN INPUT NATIF



```
const input = document.querySelector('input');  
  
fromEvent(input, 'keyup').pipe(  
  debounceTime(1000)  
)  
.subscribe(() => {  
  console.log('Valeur saisie :', input.value);  
});
```

Ici, la console n'affiche la valeur qu'une seconde après la dernière frappe (et non chaque touche).



Elodie TURLIER

Angular, IA, agilité & storytelling



EXEMPLE AVEC SUBJECT PERSONNALISÉ

```
searchSubject = new Subject<string>();

ngOnInit() {
  this.searchSubject.pipe(
    debounceTime(300)
  ).subscribe(value => this.search(value));
}

onSearch(value: string) {
  this.searchSubject.next(value);
}
```

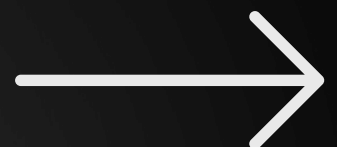
Utile pour contrôler manuellement l'émission des valeurs.

Ex : *via (input)="onSearch(\$event.target.value)"*



Elodie TURLIER

Angular, IA, agilité & storytelling



BONNES PRATIQUES

- Combinez souvent ***debounceTime*** avec ***distinctUntilChanged*** pour éviter les traitements sur des valeurs identiques successives.
- Utilisez ***takeUntil(this.destroy\$)*** pour désabonner proprement lors de la destruction du composant.
- **Adaptez le délai** selon le contexte utilisateur (200–500 ms pour la plupart des recherches).



Elodie TURLIER

Angular, IA, agilité & storytelling



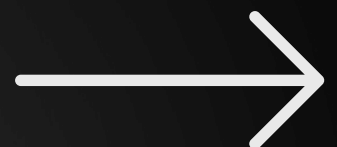
RÉSUMÉ

- ***debounceTime*** attend un **délai** sans nouvelle émission **avant de transmettre** la dernière valeur.
- **Idéal** pour les **champs de recherche**, l'autocomplétion, ou tout traitement déclenché par des événements fréquents.
- Permet d'**optimiser les performances** et d'améliorer l'expérience utilisateur dans Angular



Elodie TURLIER

Angular, IA, agilité & storytelling



CONCLUSION

En bref : Utilisez ***debounceTime*** dans vos applications Angular chaque fois que vous souhaitez éviter des traitements ou requêtes inutiles lors d'entrées utilisateur rapides. C'est un outil incontournable pour des interfaces réactives et performantes.

Pour en savoir plus, rendez-vous sur la **documentation officielle**:

<https://rxjs.dev/api/operators/debounceTime>



Elodie TURLIER

Angular, IA, agilité & storytelling



ABONNE TOI **POUR** **PLUS DE CONTENU**



Elodie TURLIER

Angular, IA, agilité & storytelling : construire des apps prêtes à survivre à l'apocalypse digitale



Enregistre ou partage si ça t'a été utile