

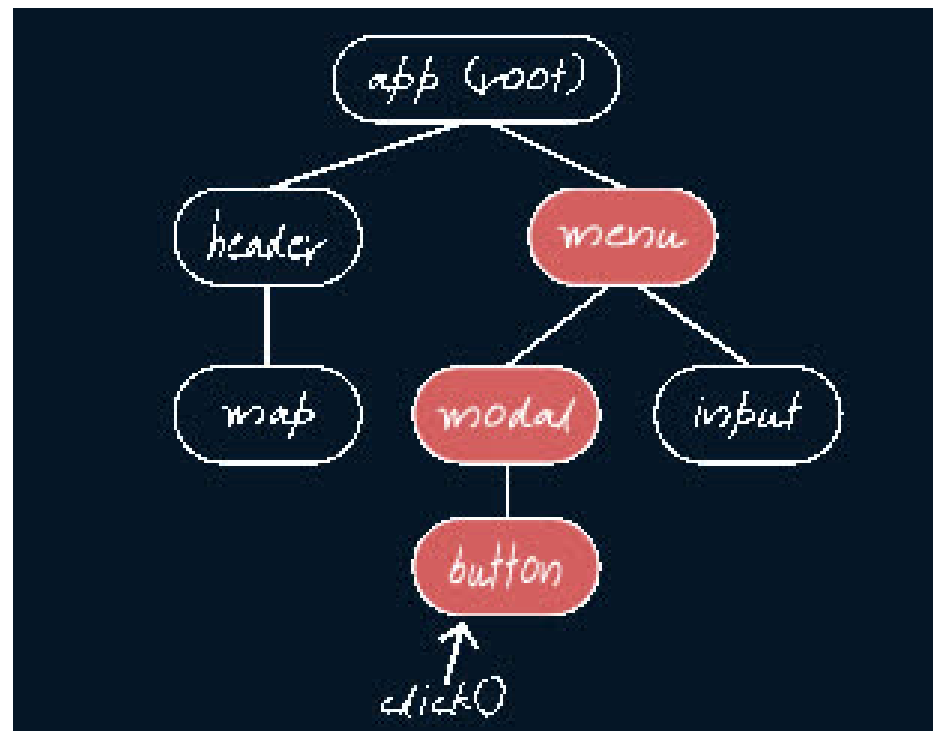
Angular Signals e Reatividade

Ândrei de Medeiros e Ítalo Costa



Introdução à Angular Signals

Otimização de atualizações de interface no Angular com Signals



O problema das atualizações ineficientes

O que são Signals e como funcionam

Como Signals Resolvem o Problema?

Comparação: Sem Signals vs. Com Signals

Demonstração prática e laboratório

O Problema : Atualizações Ineficientes na Interface de Usuário

*Como o Ciclo de Detecção de
Mudanças do Angular Pode
Gerar Atualizações
Desnecessárias*

Renderizações Desnecessárias

O Angular re-renderiza componentes que não foram alterados, ao verificar o estado de toda a árvore de componentes. Isso pode gerar atualizações de interface desnecessárias.

Desempenho Comprometido em Aplicações Grandes

Em aplicações complexas, como dashboards ou sistemas com muitos componentes, pequenas mudanças no estado podem desencadear grandes re-renderizações, afetando o desempenho.

Experiência do Usuário Prejudicada

Renderizações excessivas podem resultar em interfaces mais lentas e menos responsivas, prejudicando a experiência do usuário, especialmente em dispositivos com recursos limitados.

O Que São Signals?

Signals são uma nova API introduzida no Angular 16 que permite rastrear e reagir a mudanças de estado de forma eficiente, controlando precisamente quais partes da UI precisam ser re-renderizadas.



Funcionamento de Signals

Signals monitoram o estado e, quando ocorre uma mudança, atualizam apenas os componentes que dependem diretamente dos dados alterados, evitando renderizações desnecessárias.



Diferença em Relação à Abordagem Tradicional

Diferente do ciclo de detecção de mudanças tradicional, Signals permitem uma reatividade mais controlada e previsível, sem a necessidade de observables ou gerenciamento de assinaturas.

Como Signals Resolvem o Problema?

*Controlando Precisamente Quais Partes da UI São Atualizadas
com Base nas Mudanças de Estado*





Atualizações Granulares e Precisão na Reatividade

Signals permitem que apenas os componentes que dependem dos dados modificados sejam re-renderizados, eliminando a necessidade de re-renderizar toda a árvore de componentes.



Eliminação de Renderizações Desnecessárias

Ao invés de re-renderizar grandes partes da UI desnecessariamente, Signals garantem que apenas as áreas afetadas pela mudança de estado sejam atualizadas.



Simplicidade e Controle

Signals trazem um controle mais simples e direto sobre a reatividade, sem a complexidade de observar ou assinar fluxos de dados como nos observables, tornando o código mais fácil de entender e manter.



Comparação : Sem Signals vs Com Signals

*Como Signals Otimizam a Atualização da
Interface em Relação à Abordagem
Tradicional*

Comparação - Sem Signals vs Com Signals



Sem Signals - Renderizações Desnecessárias

Na abordagem tradicional sem Signals, o Angular re-renderiza a árvore de componentes inteira, mesmo quando apenas uma pequena parte do estado é alterada.



Com Signals - Atualizações Granulares e Precisão

Com Signals, apenas os componentes que dependem diretamente do dado modificado são atualizados, resultando em uma interface mais eficiente e responsiva.



Comparação de Código - Simplicidade e Controle

O código com Signals é mais simples, eliminando a necessidade de assinaturas manuais e permitindo maior controle sobre quais partes da UI precisam ser atualizadas.

Quais as Alternativas, e Por Que Signals São Melhores?

Comparando RxJS com Signals: Simplicidade e Eficiência em Reatividade

Alternativa - RxJS

O Angular tradicionalmente utiliza observables com RxJS para gerenciamento reativo de mudanças, mas isso pode exigir muito gerenciamento manual de assinaturas e lógica adicional.

```
import { Component } from '@angular/core';
import { BehaviorSubject } from 'rxjs';

@Component({
  selector: 'app-example',
  template: `<p>{{ value$ | async }}</p>`
})
export class ExampleComponent {
  value$ = new BehaviorSubject<number>(0);

  updateValue() {
    this.value$.next(this.value$.value + 1);
  }
}
```

Quais as Alternativas, e Por Que Signals São Melhores?

Comparando RxJS com Signals: Simplicidade e Eficiência em Reatividade

Por que Signals São Melhores?

Signals são mais simples, mais intuitivos e integram-se diretamente com o sistema de detecção de mudanças do Angular, tornando a reatividade mais eficiente e o código mais limpo.

```
import { Component, signal } from '@angular/core';

@Component({
  selector: 'app-example',
  template: `<p>{{ value() }}</p>`
})
export class ExampleComponent {
  value = signal(0);

  updateValue() {
    this.value.set(this.value() + 1);
  }
}
```

OBRIGADO!

"A persistência é o caminho do êxito." — Charles Chaplin