





Change Detection from Angular 2 (zone.js) to Angular 18 (Zoneless)

#angular #webdev #frontend #javascript

Change detection is a fundamental aspect of Angular, responsible for identifying and updating parts of the DOM that have changed in response to data modifications or user interactions. This process ensures that the UI remains consistent with the underlying data, enhancing user experience and application performance.

The Role of Zone.js

Historically, Angular has relied on Zone.js for its change detection mechanism. Zone.js is a JavaScript library that intercepts asynchronous operations, allowing Angular to monitor changes and trigger updates accordingly. However, the inclusion of Zone.js can add overhead to the application, particularly in scenarios with frequent asynchronous activities.

Change Detection Strategies

Angular provides two primary change detection strategies:

• **Default:** Change detection is triggered after every lifecycle hook, such as ngonInit or ngAfterViewInit. This strategy is straightforward but can lead to unnecessary DOM updates, especially in large applications.

This strategy is easier to implement, as Angular handles most of the change detection logic automatically.

The biggest challenge with this change detection strategy was that it led to unnecessary DOM updates which gets critical for large applications.

• **OnPush:** Change detection is triggered only when input properties or asynchronous observables change. This strategy is more performant for complex components with frequent data updates but requires more manual management.

While onPush change detection strategy minimized unnecessary DOM manipulations, the biggest challenge with onPush change strategy was that the developers had to do more manual management and trigger changes manually. They had to handle change

detection explicitly with <code>changeDetectorRef.detectChanges()</code> in certain scenarios, such as when data changes indirectly or when using mutable objects.

Angular 18 and Hybrid Change Detection: A Zone-less Approach

To address the challenges associated with <code>zone.js</code>, Angular 18 introduced an experimental feature known as Hybrid Change Detection. This approach aims to eliminate <code>zone.js</code> entirely, using a new change detection mechanism to detect changes and manipulate the DOM.

Enabling Hybrid Change Detection:

To enable Hybrid Change Detection, you can use the following code:

```
bootstrapApplication(RootCmp,
{ providers: [provideExperimentalZonelessChangeDetection()]
}
);
```

This will trigger change detection in the following scenarios:

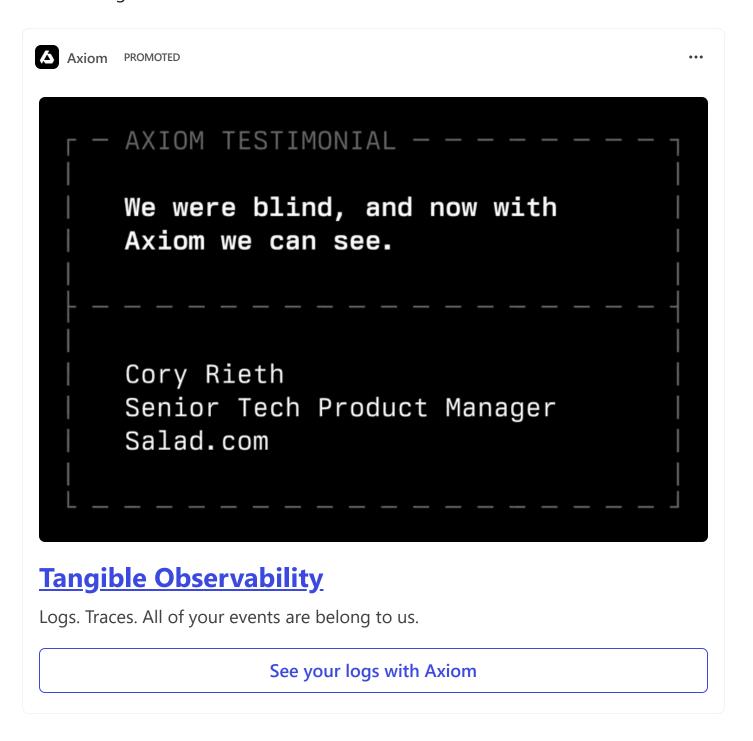
- A signal is updated.
- changeDetectorRef.markForCheck() is called.
- An observable subscribed with the AsyncPipe receives a new value.
- A component gets attached or detached.
- An input is set.

Once Hybrid Change Detection is enabled, you can safely remove Zone.js from your application's polyfills.

Benefits of Hybrid Change Detection

• **Improved performance:** Eliminating Zone.js reduces overhead, leading to better performance, especially in applications with numerous asynchronous operations.

- **Enhanced developer experience:** The removal of manual change detection simplifies development and reduces the likelihood of errors.
- **Smaller application size:** Zone.js typically adds around 13KB to the application size. Removing it can lead to a smaller bundle.



Top comments (0)

Code of Conduct • Report abuse