Date

What is Change detection In Angular

Detecting changes in any component of your app and re-render view, so it displays updated value to end users.

Different events on which change occurs

→ Data received from network requests or component events
→ Mouse clicks, scrolling, mouseover, keyboard navigation
→ AJax calls
→ Use of timer fn such as setTimeout

How does change detection Work?

→ Whenever something triggers a change
→ To detect and update DOM with changed data, the framework provides its own change detector to each component

→ The change detector make sure data model and DOM are in sync.

→ When Angular change detection detects the triggered change and run change detection to all component tree from top to bottom.

Page
Date

Angular Change Detection Strategies

→ Default Strategy
→ OnPush Strategy

Default Change Detection Strategy

If Angular Change Detector is set to default then for any change in any model property, Angular will run change detection traversing component tree to update DOM.

Change detector Ref class provides few built in methods, we can use to manipulate change detection tree:

- markForCheck() → marks that it changed
- detach() → exclude view from change detection tree
- detectChanges() → check view and its child component
- checkNoChanges() → check view and its child, throw error if change detected
- reattach() → reattach a view that was previously detached

Angular OnPush Change Detection

If Angular change detector is set to onPush then Angular will run change detector only when new reference is being passed to component.

If observable is passed to onPush, then change Detector must be called manually to update DOM

```
@Component ({
 Selector : 'app-root'
templateUrl : './app. component. html'
change Detection: change Detection Strategy. OnPush,
Style Urls : [:/ app. component. scss]
})

export class App Component {
} }
```

Benefits

→ Unnecessary check in child not performed if parent element updating values which are not passed as @ Input.