

What's New in Angular



Joe Eames

Google Developer Expert

@josepheames | joeames.me



Overview



Standalone components

Signals

Miscellaneous





Standalone Components



Standalone Component

```
@Component({  
  standalone: true,  
  selector: 'new-user',  
})  
Export class NewUserComponent {  
  ...  
}
```

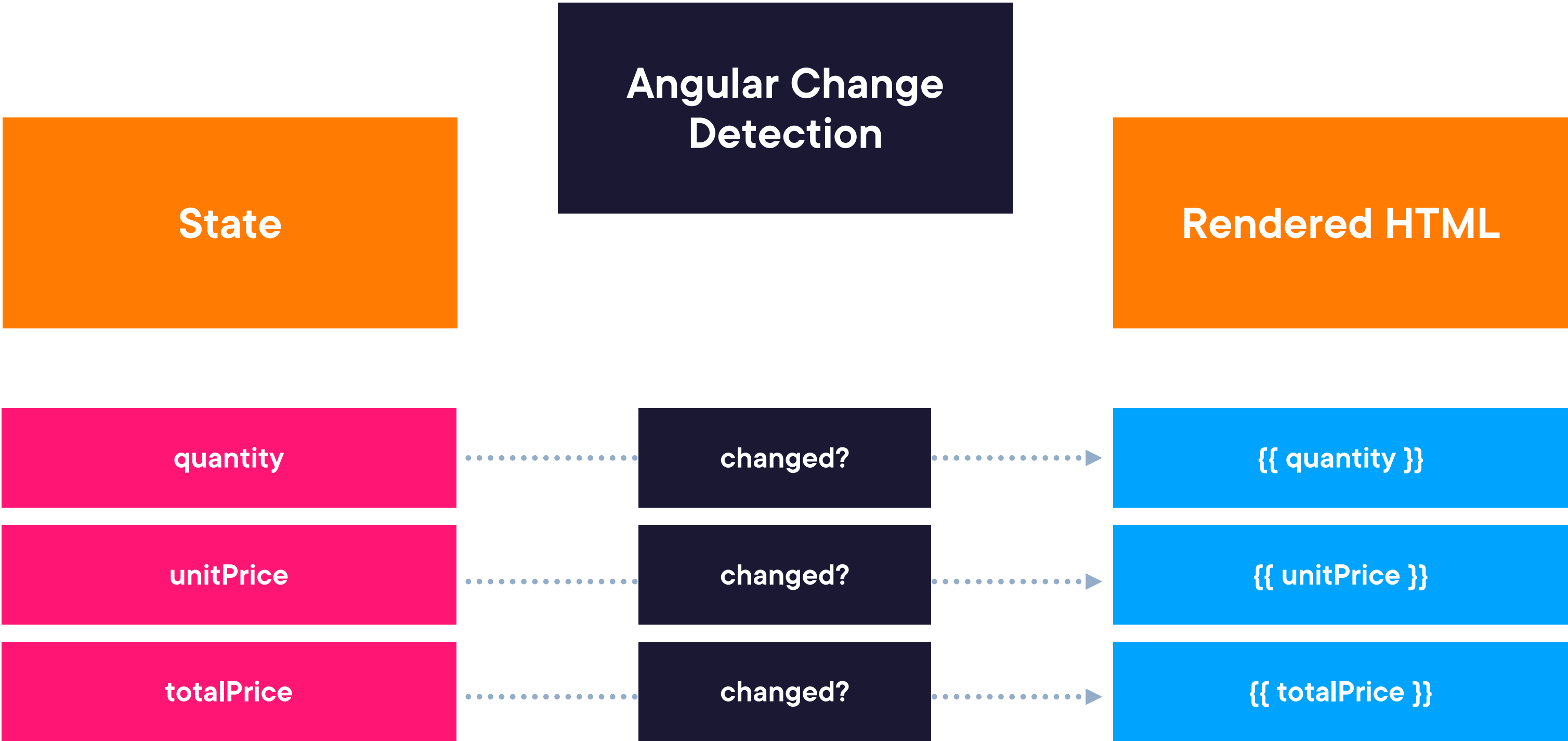




Signals



Signals



Signals

```
quantity = signal(1);
```

```
quantity()
```

```
quantity.set(2);
```

```
quantity.update(qty => qty + 1);
```

```
totalPrice = computed(() => quantity() * unitPrice())
```





Miscellaneous



Stack Trace Improvements

```
ERROR Error: Uncaught (in promise): Error
Error
    at app.component.ts:18:11
    at Generator.next (<anonymous>)
    at asyncGeneratorStep (asyncToGenerator.js:3:1)
    at _next (asyncToGenerator.js:25:1)
    at _ZoneDelegate.invoke (zone.js:372:26)
    at Object.onInvoke (core.mjs:26378:33)
    at _ZoneDelegate.invoke (zone.js:371:52)
    at Zone.run (zone.js:134:43)
    at zone.js:1275:36
    at _ZoneDelegate.invokeTask (zone.js:406:31)
    at resolvePromise (zone.js:1211:31)
    at zone.js:1118:17
    at zone.js:1134:33
```

Old Style
Overly verbose

```
ERROR Error: Uncaught (in promise): Error
Error
    at app.component.ts:18:11
    at fetch (async)
    at (anonymous) (app.component.ts:4)
    at request (app.component.ts:4)
    at (anonymous) (app.component.ts:17)
    at submit (app.component.ts:15)
    at AppComponent_click_3_listener (app.component.html:4)
```

New Traces
More concise and useful



Router Guards

```
@Injectable({ providedIn: 'root' })  
export class MyGuard implements CanActivate {  
  constructor(private loginService: LoginService) {}  
  canActivate() {  
    return this.loginService.isLoggedIn();  
  }  
}
```

```
route = {  
  path: '/userSettings',  
  canActivate: [MyGuard]  
};
```



Functional Router Guards

```
const route = {  
  path: 'userSettings',  
  canActivate: [() => inject(LoginService).isLoggedIn()]  
};
```



Summary



Standalone components

Signals

Other

