

## Signals: What's To Come?

As mentioned, the "Angular Signals" feature is **not finished** yet!

Whilst the foundation (creating signals, reading signals, updating signals) is fully implemented - as a developer preview, though - some more advanced features are still **missing**.

**Signal-based components**, for example.

In the future, you'll be able to mark components as signal-based (presumably by adding the `signals: true` flag to `@Component`) to let Angular know that it should not use its regular change detection algorithm in such components.

And that's just one feature, of course. In signal-based components, you'll also be able to get rid of many decorators like `@Input()` or `@Output()` (and others). You would use special functions instead:

```
1. @Input() title: string;
```

would become

```
1. title = input<string>();
```

for example.

But, again, this part is still missing! It's likely to be added with Angular 17.

You can learn more about the complete planned (!) Signals API in the official RFCs: <https://github.com/angular/angular/discussions/49685>

Please note that this RFC is split up into four "sub-RFCs" that dive deeper into specific aspects of Angular.

For example, [this sub-RFC](#) takes a closer look at the signal-based components mentioned above.