Composants et Modules

Un composant

Un composant, c'est une classe Typescript avec des attributs, des méthodes, un constructeur et aussi des métadonnées injectées via la directive @Component.

Le composant sert d'unité de découpage visuel dans le projet front.

On définit un composant dans Angular par un ensemble de ressources : une portion de HTML.

Création du composant

```
ng generate component say-hello
```

> le composant est injecter dans le fichier `app.module.ts`

Analyse du composant

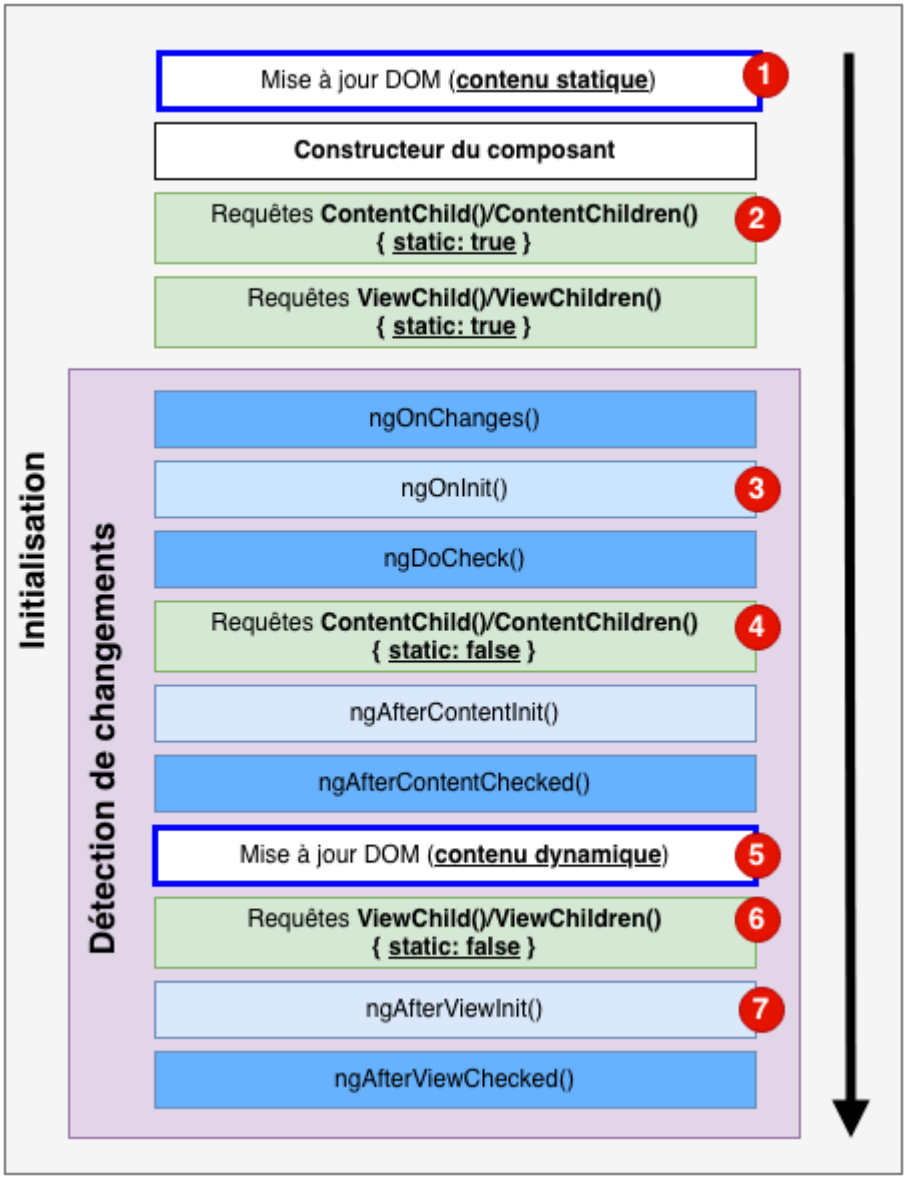
- say-hello.component.ts
- say-hello.component.html
- say-hello.component.css
- say-hello.component.spec.ts

Déclenchement du composant

`app.component.html`

```
<app-say-hello></app-say-hello>
```

Cycles de vie du composant



1. A l'initialisation, le DOM est mis à jour avec le contenu statique de la vue.
2. Les requêtes avec le paramètre `{ static: true }` sur le contenu projeté (avec `@ContentChild()` ou `@ContentChildren()`) et sur la vue (avec `@ViewChild()` ou `@ViewChildren()`) sont exécutées sur le contenu statique de la vue.
3. A l'exécution de la *callback* `ngOnInit()`, les requêtes sur le contenu statique ont été exécutées.
4. Les requêtes avec le paramètre `{ static: false }` sur le contenu projeté (avec `@ContentChild()` ou `@ContentChildren()`) sont exécutées. Le lien avec le composant parent n'apparaît pas sur ce schéma toutefois à ce state, le contenu dynamique du DOM du composant parent a été mise à jour.
5. A chaque détection de changements, le contenu dynamique de la vue est mis à jour.
6. Les requêtes avec le paramètre `{ static: false }` sur la vue (avec `@ViewChild()` ou `@ViewChildren()`) sont exécutées sur le contenu dynamique de la vue.
7. A l'exécution de la *callback* `ngAfterViewInit()`, les requêtes sur le contenu dynamique ont été exécutées.

Création d'un module

```
ng generate component my-first-module
```

Création d'un module avec router interne