Angular, the concepts of "pure" and "impure" refer to whether a pipe or change detection strategy has side effects or not. These concepts are especially relevant when working with pipes and change detection mechanisms in Angular applications.

**Pure Pipe:**

A pure pipe is a type of pipe in Angular that only produces output based on its input. It does not have any internal state or dependencies that could lead to side effects. Pure pipes are designed to be idempotent, meaning that given the same input, they will always produce the same output. Pure pipes are highly efficient and are automatically cached by Angular's change detection mechanism.

**Example:**

**@Pipe({**

**name: 'uppercase',**

**pure: true // This is the default value**

**})**

**export class UppercasePipe implements PipeTransform {**

**transform(value: string): string {**

**return value.toUpperCase();**

**}**

**}**

**Impure Pipe:**

An impure pipe, on the other hand, may have side effects or dependencies that can change over time, leading to different outputs for the same input. Impure pipes are re-evaluated frequently by Angular's change detection, which can impact performance if not used carefully. You can create an impure pipe by setting the pure property to false in the @Pipe decorator.

**Example:**

**@Pipe({**

**name: 'randomNumber',**

**pure: false**

**})**

**export class RandomNumberPipe implements PipeTransform {**

**transform(): number {**

**return Math.random();**

**}**

**}**

In general, it's recommended to use pure pipes whenever possible, as they are more efficient and predictable. Impure pipes should be used sparingly and only when necessary, as they can lead to unnecessary re-evaluations during change detection.

It's important to note that the concepts of pure and impure are also applicable to change detection strategies for components. Components can have either the default "OnPush" (pure) change detection strategy or the "Default" (impure) change detection strategy, which affects how changes are detected and propagated through the application.

Regenerate