Import and Component Scan

https://www.baeldung.com/spring-import-annotation#import-vs-componentscan

In the context of Spring framework configuration, `@Import` annotation is used to import other configuration classes into the current configuration class. This allows you to group related configuration settings and beans across multiple classes.

When it comes to managing a growing number of configuration classes, manually adding `@Import` statements for each new component or configuration class can become cumbersome and prone to human error.

Consider a scenario where you have multiple configuration classes related to different aspects of your application, such as `DatabaseConfig`, `SecurityConfig`, `MessagingConfig`, etc. As your application grows, you may need to add more configuration classes for new features or functionalities. If you use `@Import` to include these configuration classes in your main configuration class, you'll need to remember to update the `@Import` statements every time you add a new configuration class.

To address this issue, the statement "On the other hand, we don’t want to write an @Import for each new component because doing so is counterproductive" suggests that manually importing each configuration class using `@Import` can be tedious and inefficient, especially as your application grows.

Instead, using `@ComponentScan` annotation along with specifying base packages allows Spring to automatically detect and import all configuration classes within those packages. This approach eliminates the need for manually adding `@Import` statements for each new configuration class, making the configuration management process more scalable and less error-prone.

In summary, while `@Import` can be useful for explicitly importing specific configuration classes, relying solely on it to manage a large number of configuration classes can become unwieldy. Using `@ComponentScan` provides a more automated and scalable way to import configuration classes without the need for manual intervention.