

@ConfigurationProperties

`@ConfigurationProperties` is a Spring Boot annotation used to bind and validate external configuration properties to a Java object. It allows you to map properties defined in external configuration files (such as `application.properties` or `application.yml`) to fields in a Java bean.

Here's how it works:

1. Define a Java class with fields that correspond to the properties you want to bind:

```
```java
@ConfigurationProperties(prefix = "myapp")
public class MyAppProperties {
 private String name;
 private int port;

 // Getters and setters
}
```
```

2. Annotate the class with `@ConfigurationProperties` and specify a prefix that matches the prefix used in your properties file.

3. Define the properties in your `application.properties` or `application.yml` file:

```
```properties
myapp.name=My Application
myapp.port=8080
```
```

4. Spring Boot will automatically bind the properties to the corresponding fields in your `MyAppProperties` class.

You can then inject the `MyAppProperties` bean into your Spring components using `@Autowired` or constructor injection, and access the properties through the bean's getter methods.

Additionally, `@ConfigurationProperties` supports features like validation, default values, and nested properties, making it a powerful tool for managing configuration in Spring Boot applications.

<https://www.baeldung.com/configuration-properties-in-spring-boot>

<https://github.com/spring-projects/spring-boot/issues/28592>