



09

Para saber mais: JsonAlias e JsonProperty

@JsonAlias e @JsonProperty são anotações em Jackson, uma biblioteca Java para processar JSON, que ajudam a mapear propriedades de classe para campos JSON.

No entanto, eles possuem funções diferentes. Vamos conhecer melhor cada uma dessas anotações?

@JsonProperty

Essa anotação é usada para definir o nome da propriedade JSON que está associada ao campo Java.

Quando o JSON é serializado, isto é, convertido de objetos Java para JSON, o nome especificado em @JsonProperty será usado como a chave para o campo no JSON de saída. Da mesma forma, quando o JSON é

desserializado (convertido de JSON para objetos Java), a biblioteca procura pelo nome especificado em `@JsonProperty` para mapear o valor JSON para o campo Java.

Então, se você quiser usar um nome diferente no JSON, pode usar o `@JsonProperty("nomeNoJson")`. Isso é útil, por exemplo, quando você precisa trabalhar com um JSON que segue uma convenção de nomenclatura diferente da sua classe Java.

Por exemplo, se você tiver uma classe Java com a propriedade "nomeCompleto" e quiser que ela seja mapeada para o JSON como "nome", você pode usar o `@JsonProperty("nome")` para especificar o nome correto no JSON:

```
public class Pessoa {  
    @JsonProperty("nome")  
    private String nomeC  
}
```

[COPIAR CÓDIGO](#)

@JsonAlias

O `@JsonAlias` é usado para definir um ou mais apelidos para o nome da propriedade JSON associada ao campo Java.

Ao desserializar, o `@JsonAlias` permite que a biblioteca encontre o valor JSON correspondente, mesmo que o nome da propriedade no JSON não corresponda exatamente ao nome do campo Java.

Isso é útil quando você está trabalhando com diferentes versões de um JSON ou quando deseja permitir que uma propriedade seja referenciada por nomes diferentes.

Por exemplo, se você tiver uma classe Java com a propriedade `"nomeCompleto"` e o JSON usa `"nome"` em vez disso, você pode usar o `@JsonAlias("nome")` para mapear corretamente a propriedade. Dessa forma, tanto `"nomeCompleto"` quanto `"nome"` serão aceitos ao fazer o mapeamento:

```
public class Pessoa {  
    @JsonAlias({"nomeCom  
    private String nomeC  
}
```

[COPIAR CÓDIGO](#)

Para mais informações sobre as anotações do Jackson, você pode consultar a documentação oficial aqui: [Documentação do Jackson Annotations](https://github.com/FasterXML/jackson) (<https://github.com/FasterXML/jackson>).