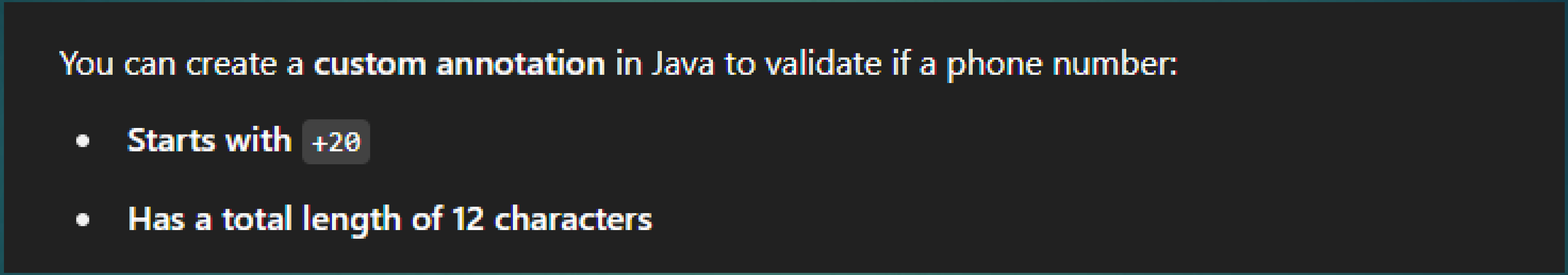
custom annotation in Java

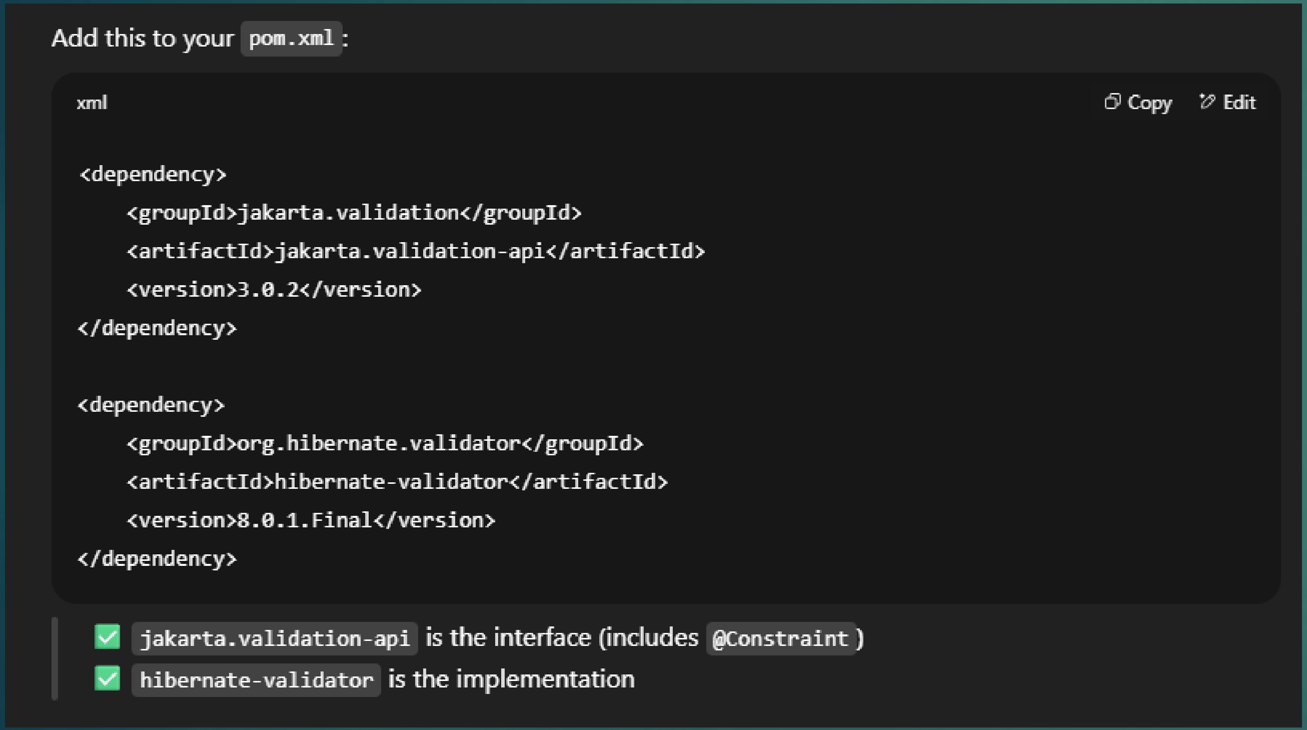


You can create a **custom annotation** in Java to validate if a phone number:

- **Starts with** `+20`
- **Has a total length of 12 characters**

By Eslam Khder

custom annotation in Java


Add this to your `pom.xml`:

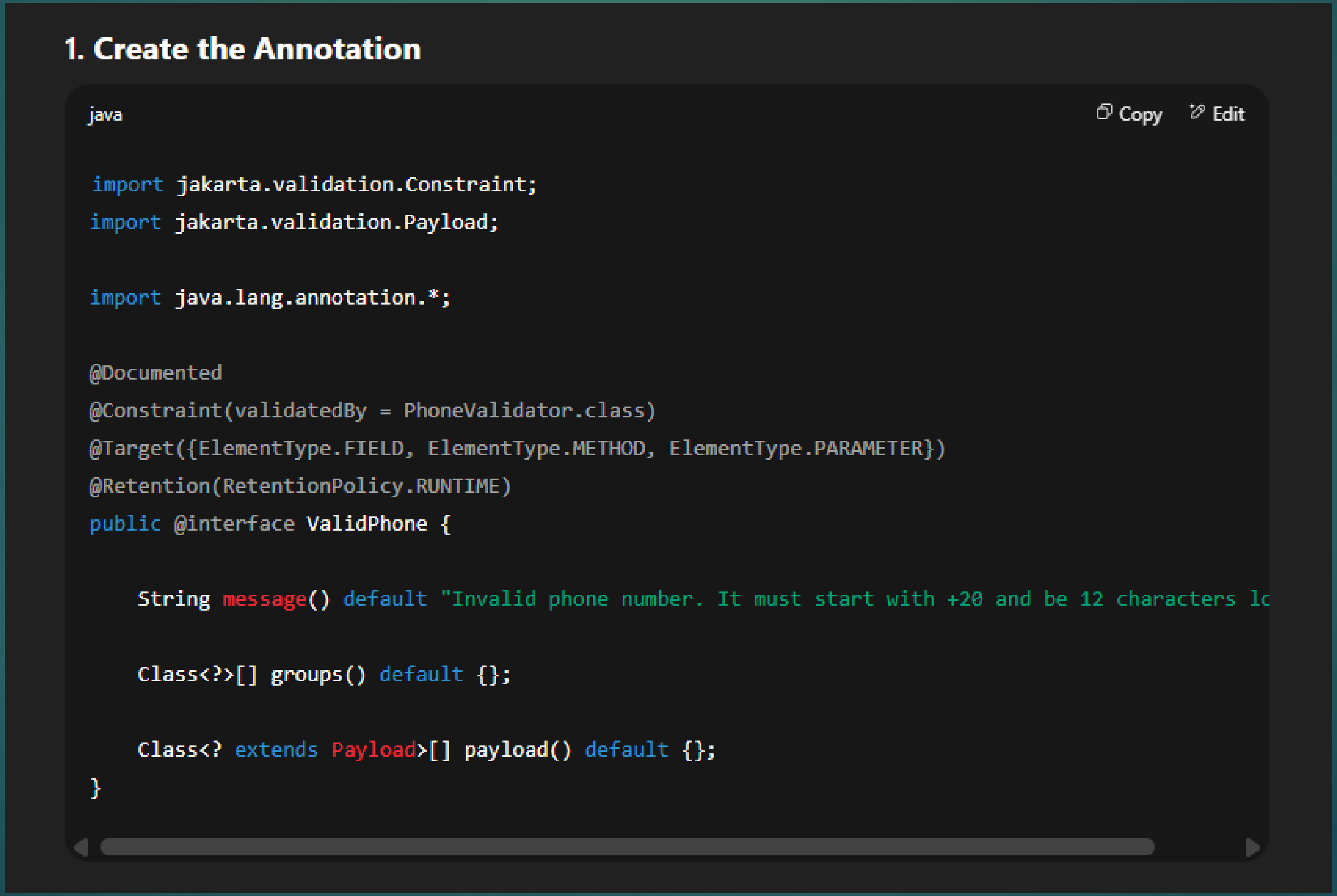
```xml
<dependency>
    <groupId>jakarta.validation</groupId>
    <artifactId>jakarta.validation-api</artifactId>
    <version>3.0.2</version>
</dependency>

<dependency>
    <groupId>org.hibernate.validator</groupId>
    <artifactId>hibernate-validator</artifactId>
    <version>8.0.1.Final</version>
</dependency>
```

> ✅ `jakarta.validation-api` is the interface (includes `@Constraint`)
> ✅ `hibernate-validator` is the implementation

```
<dependency>
<groupId>jakarta.validation</groupId>
<artifactId>jakarta.validation-api</artifactId>
<version>3.0.2</version>
</dependency>

<dependency>
<groupId>org.hibernate.validator</groupId>
<artifactId>hibernate-validator</artifactId>
<version>8.0.1.Final</version>
</dependency>
```

By Eslam Khder

custom annotation in Java



1. Create the Annotation

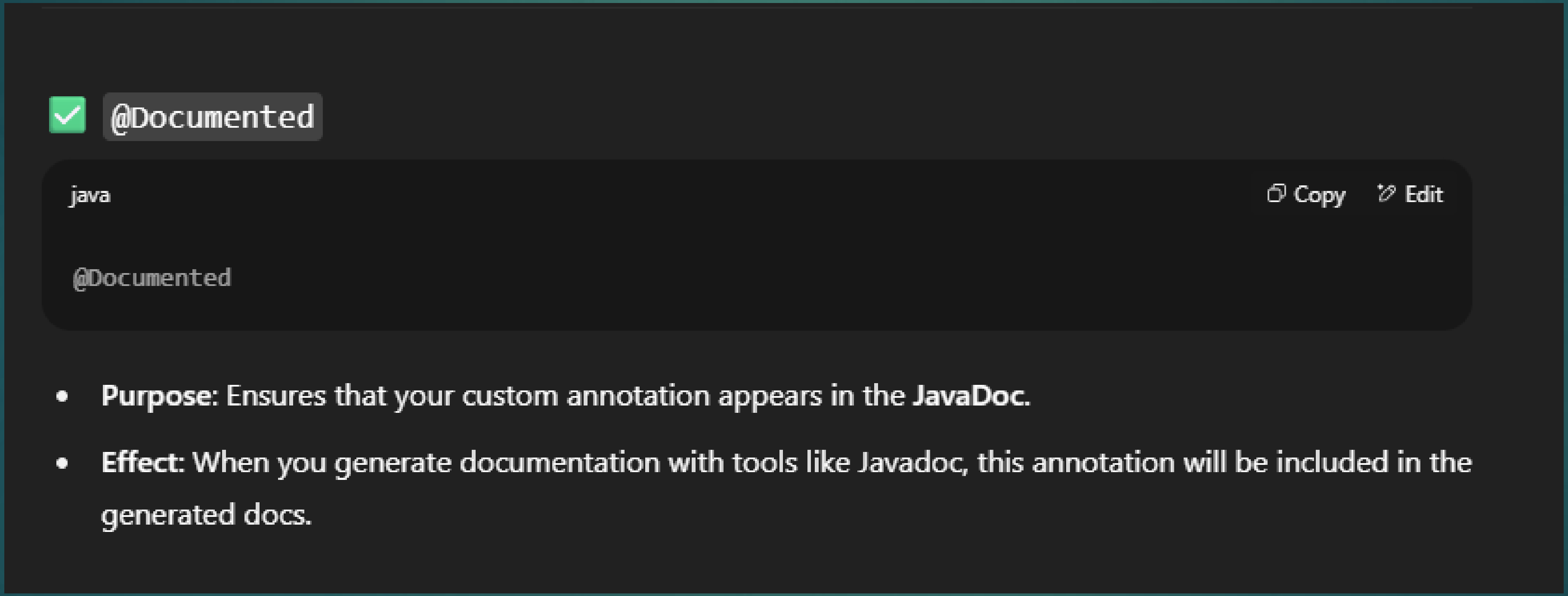
```java
import jakarta.validation.Constraint;
import jakarta.validation.Payload;

import java.lang.annotation.*;

@Documented
@Constraint(validatedBy = PhoneValidator.class)
@Target({ElementType.FIELD, ElementType.METHOD, ElementType.PARAMETER})
@Retention(RetentionPolicy.RUNTIME)
public @interface ValidPhone {

    String message() default "Invalid phone number. It must start with +20 and be 12 characters lo

    Class<?>[] groups() default {};

    Class<? extends Payload>[] payload() default {};
}
```

By Eslam Khder

spring

EraaSoft

custom annotation in Java



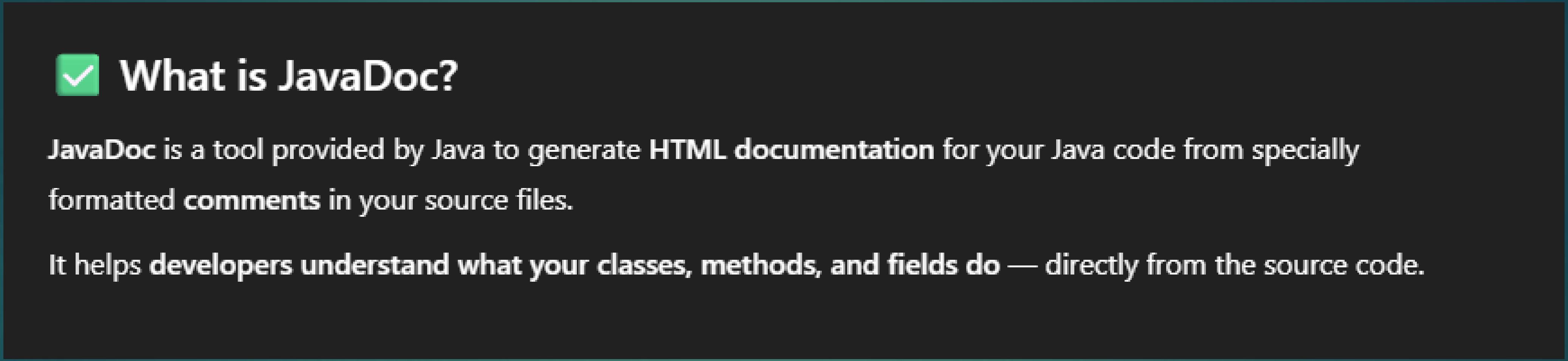
✅ `@Documented`

```java
@Documented
```

- **Purpose:** Ensures that your custom annotation appears in the **JavaDoc.**
- **Effect:** When you generate documentation with tools like Javadoc, this annotation will be included in the generated docs.

By Eslam Khder

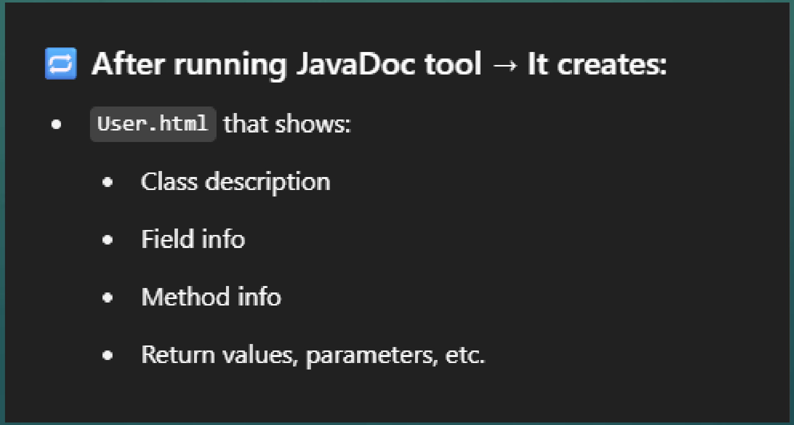
custom annotation in Java



✅ What is JavaDoc?

JavaDoc is a tool provided by Java to generate **HTML documentation** for your Java code from specially formatted **comments** in your source files.

It helps **developers understand what your classes, methods, and fields do** — directly from the source code.



🔁 After running JavaDoc tool → It creates:

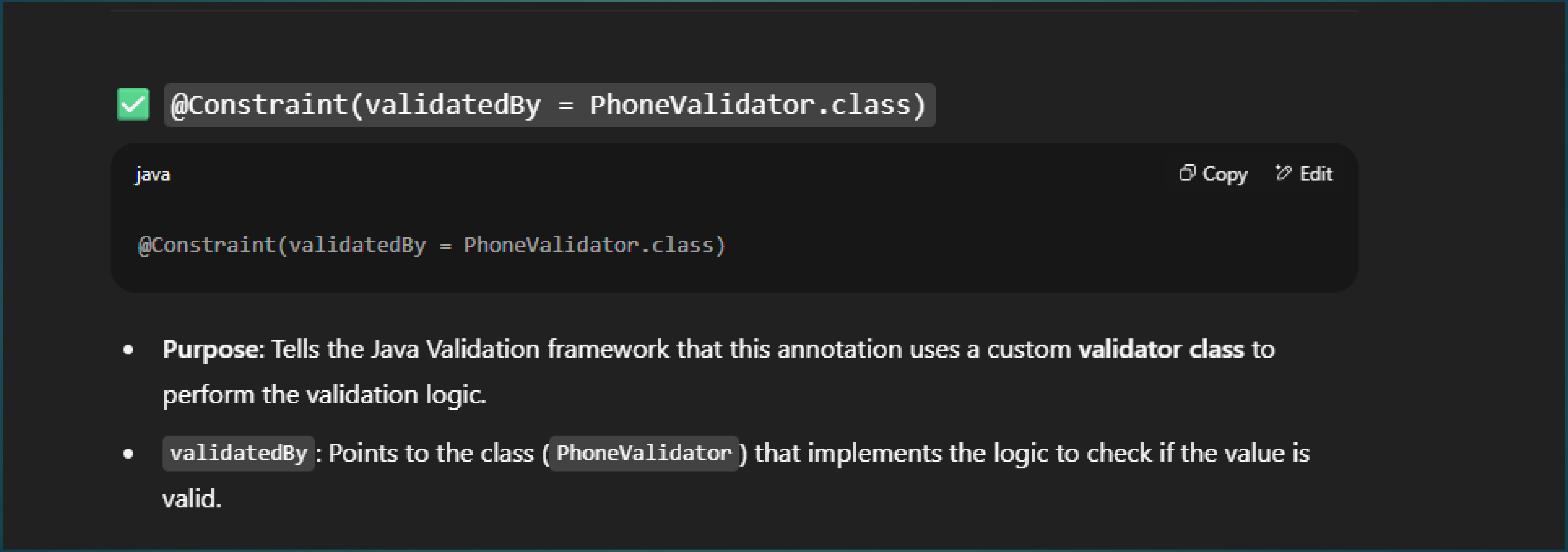
- `User.html` that shows:
    - Class description
    - Field info
    - Method info
    - Return values, parameters, etc.

By Eslam Khder

spring

EraaSoft

custom annotation in Java



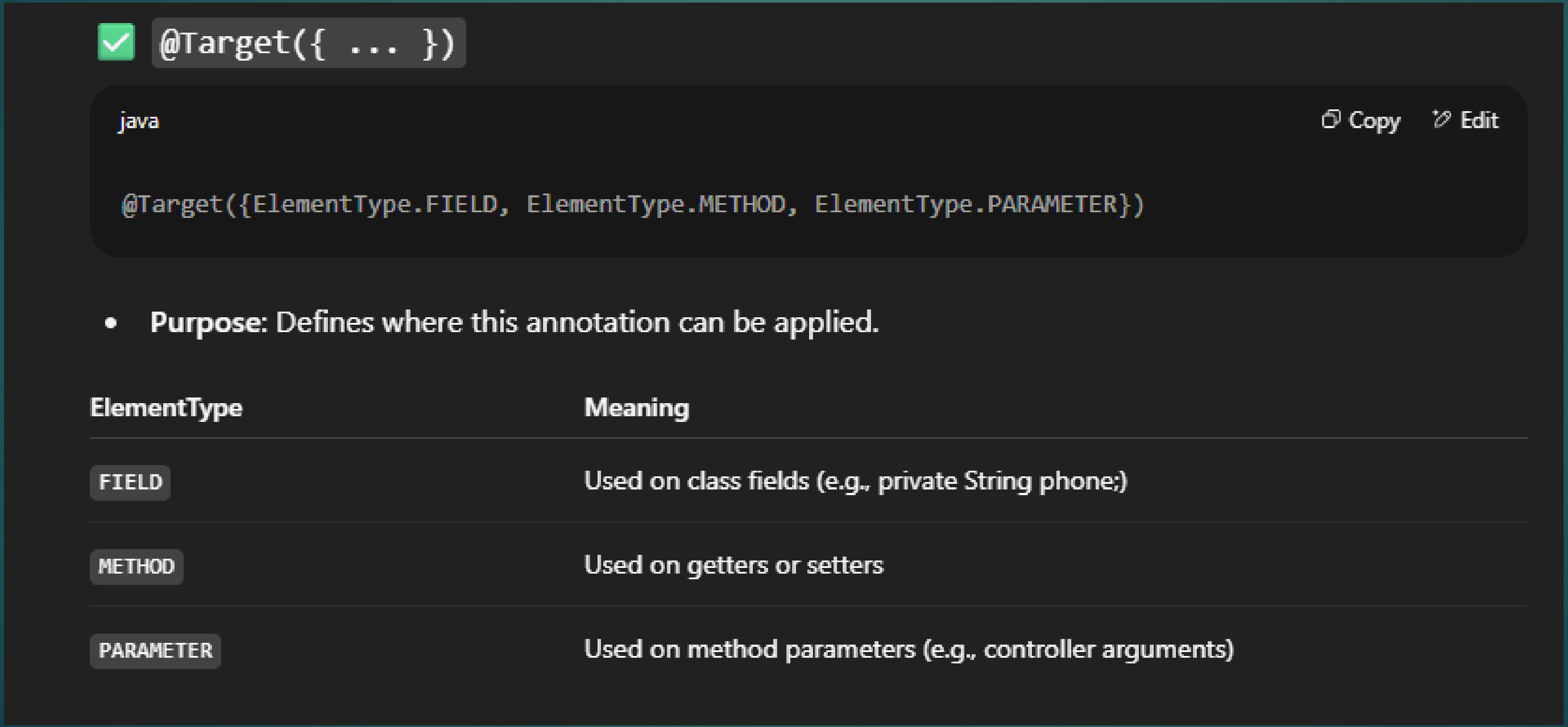
✅ `@Constraint(validatedBy = PhoneValidator.class)`

```java
@Constraint(validatedBy = PhoneValidator.class)
```

- **Purpose:** Tells the Java Validation framework that this annotation uses a custom **validator class** to perform the validation logic.
- `validatedBy`: Points to the class (`PhoneValidator`) that implements the logic to check if the value is valid.

By Eslam Khder

custom annotation in Java



✅ `@Target({ ... })`

java

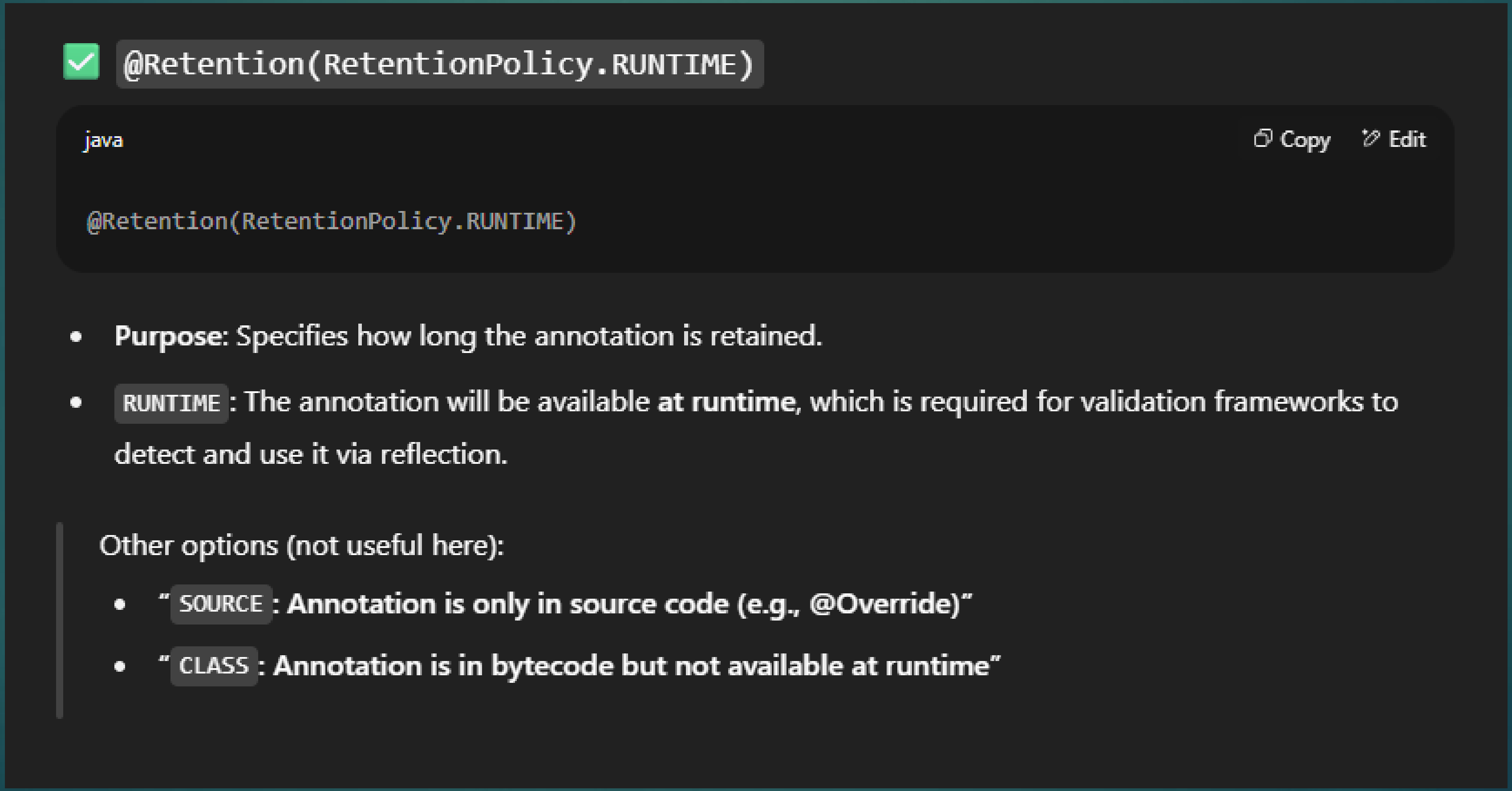
```java
@Target({ElementType.FIELD, ElementType.METHOD, ElementType.PARAMETER})
```

- **Purpose:** Defines where this annotation can be applied.

ElementType	Meaning
`FIELD`	Used on class fields (e.g., private String phone;)
`METHOD`	Used on getters or setters
`PARAMETER`	Used on method parameters (e.g., controller arguments)

By Eslam Khder

custom annotation in Java



✅ `@Retention(RetentionPolicy.RUNTIME)`

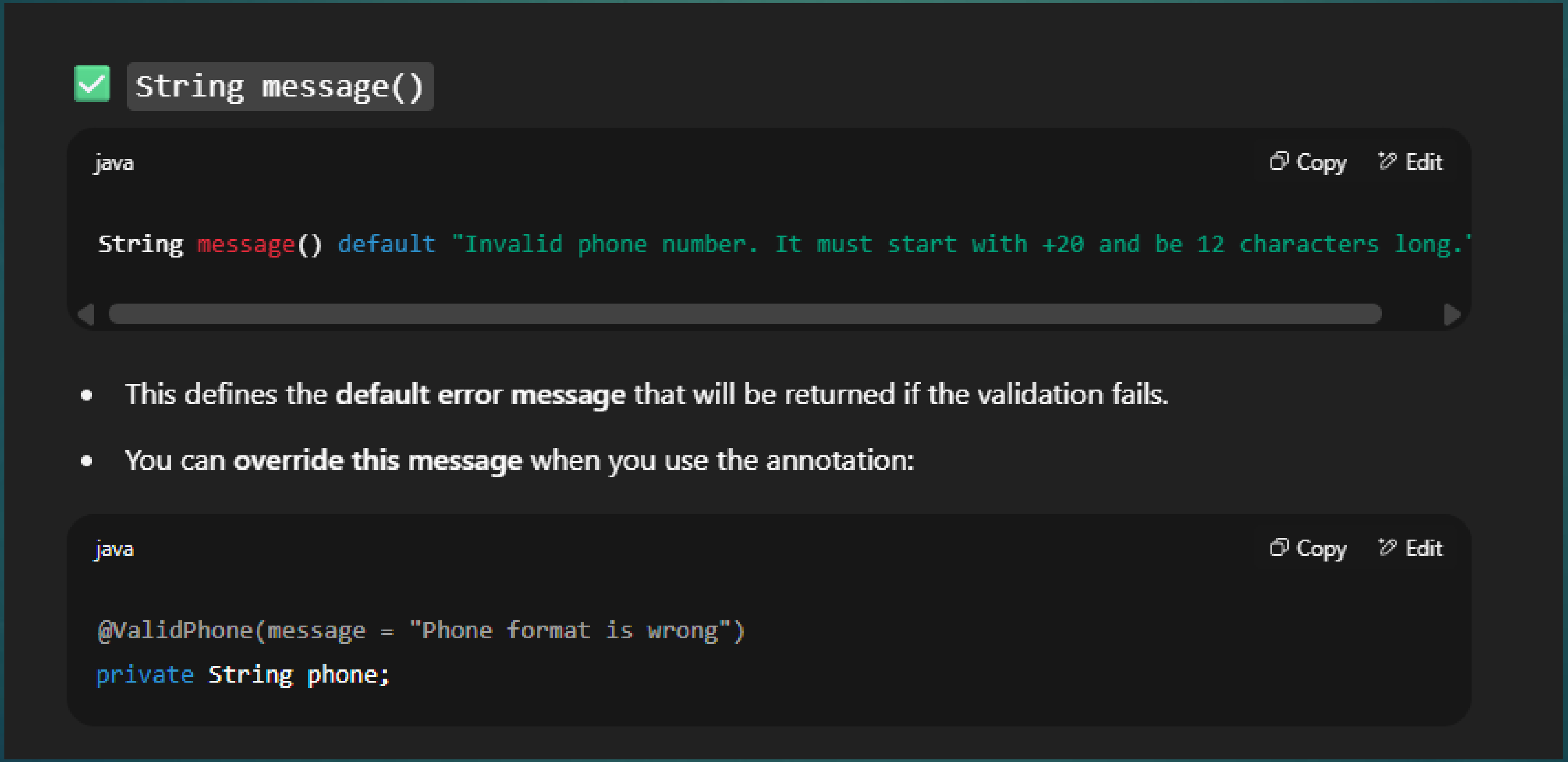
```java
@Retention(RetentionPolicy.RUNTIME)
```

- **Purpose:** Specifies how long the annotation is retained.
- `RUNTIME`: The annotation will be available **at runtime**, which is required for validation frameworks to detect and use it via reflection.

> Other options (not useful here):
> - "`SOURCE`: **Annotation is only in source code (e.g., @Override)**"
> - "`CLASS`: **Annotation is in bytecode but not available at runtime**"

By Eslam Khder

custom annotation in Java



✓ `String message()`

java

Copy Edit

```java
String message() default "Invalid phone number. It must start with +20 and be 12 characters long.'
```

- This defines the **default error message** that will be returned if the validation fails.
- You can **override this message** when you use the annotation:

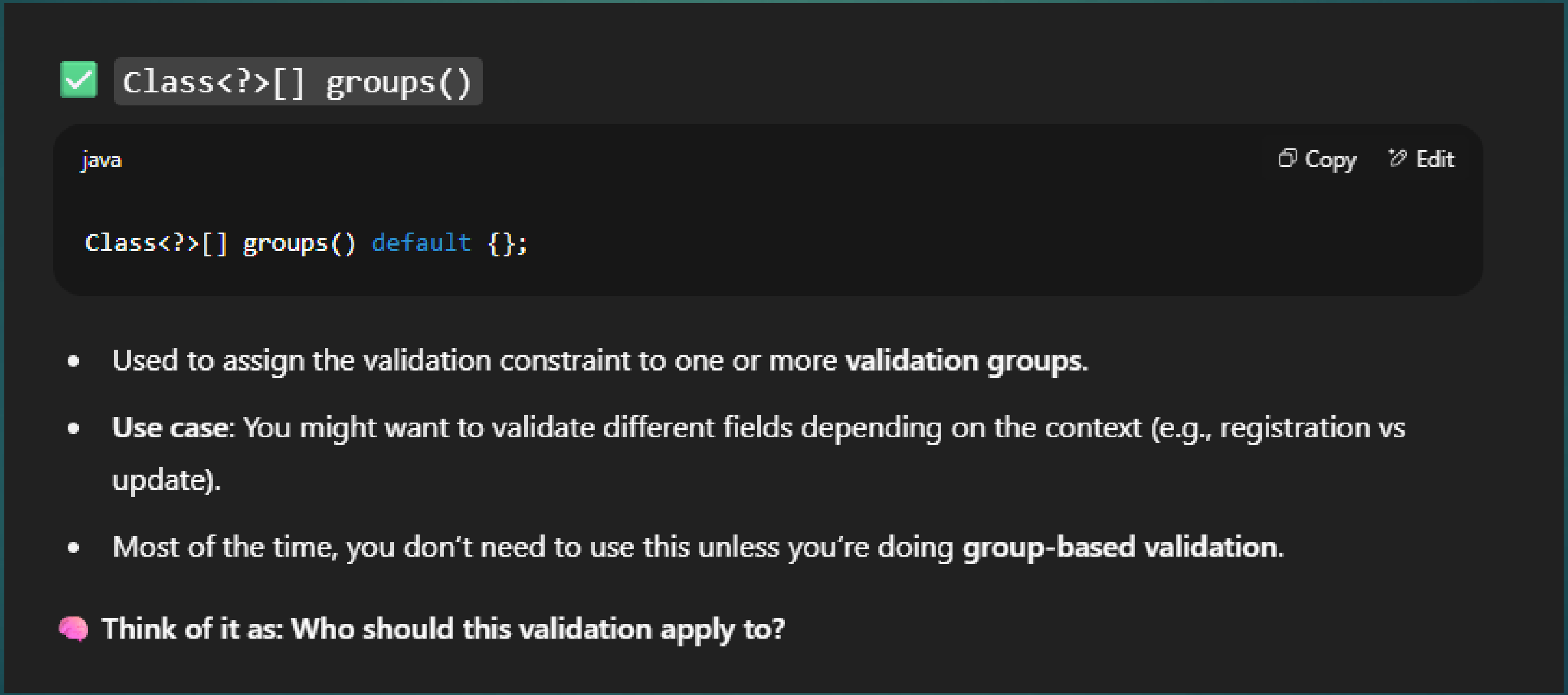
java

Copy Edit

```java
@ValidPhone(message = "Phone format is wrong")
private String phone;
```

By Eslam Khder

custom annotation in Java



✅ `Class<?>[] groups()`

java Copy Edit

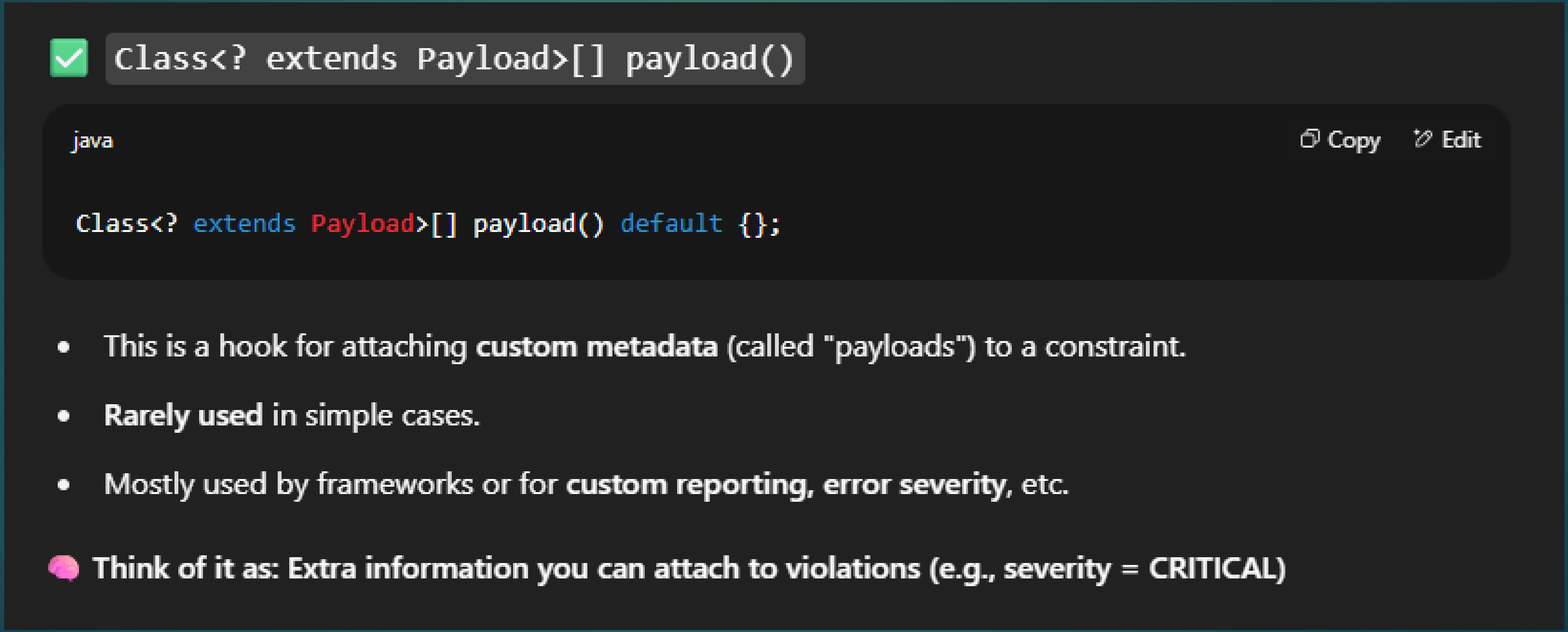
```java
Class<?>[] groups() default {};
```

- Used to assign the validation constraint to one or more **validation groups.**
- **Use case:** You might want to validate different fields depending on the context (e.g., registration vs update).
- Most of the time, you don't need to use this unless you're doing **group-based validation.**

🧠 **Think of it as: Who should this validation apply to?**

By Eslam Khder

custom annotation in Java



✅ `Class<? extends Payload>[] payload()`

```java
Class<? extends Payload>[] payload() default {};
```

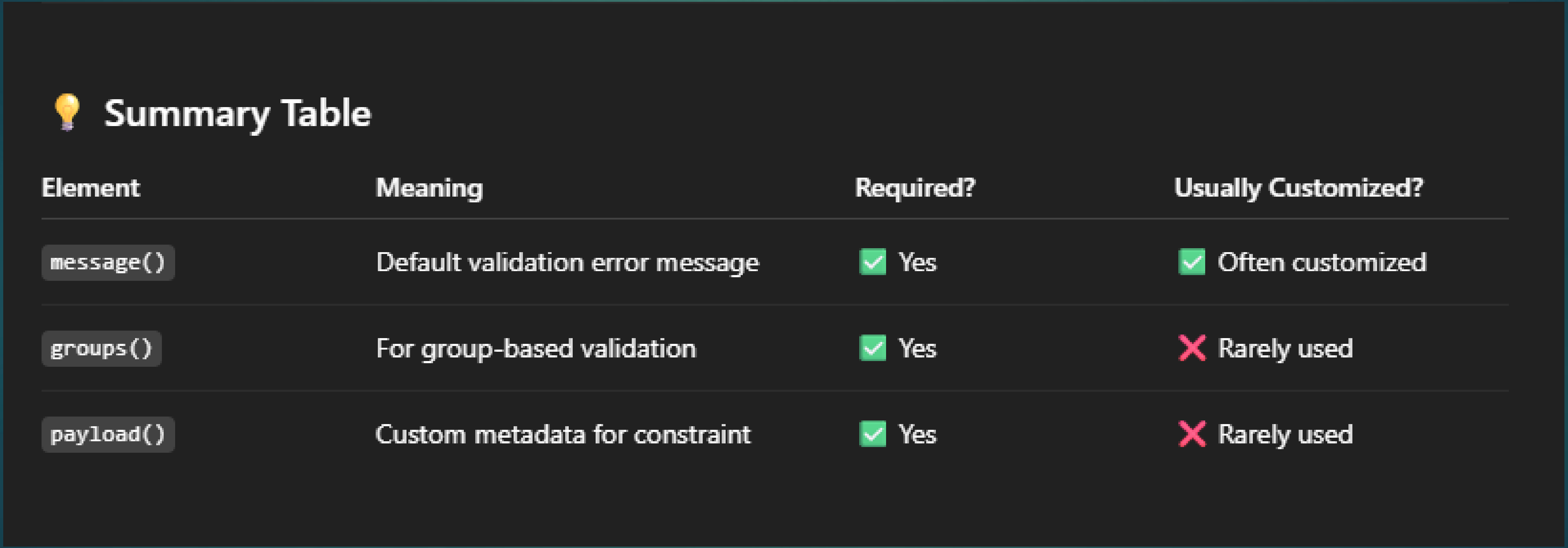
- This is a hook for attaching **custom metadata** (called "payloads") to a constraint.
- **Rarely used** in simple cases.
- Mostly used by frameworks or for **custom reporting, error severity**, etc.

🧠 **Think of it as: Extra information you can attach to violations (e.g., severity = CRITICAL)**

By Eslam Khder

spring

custom annotation in Java



💡 Summary Table

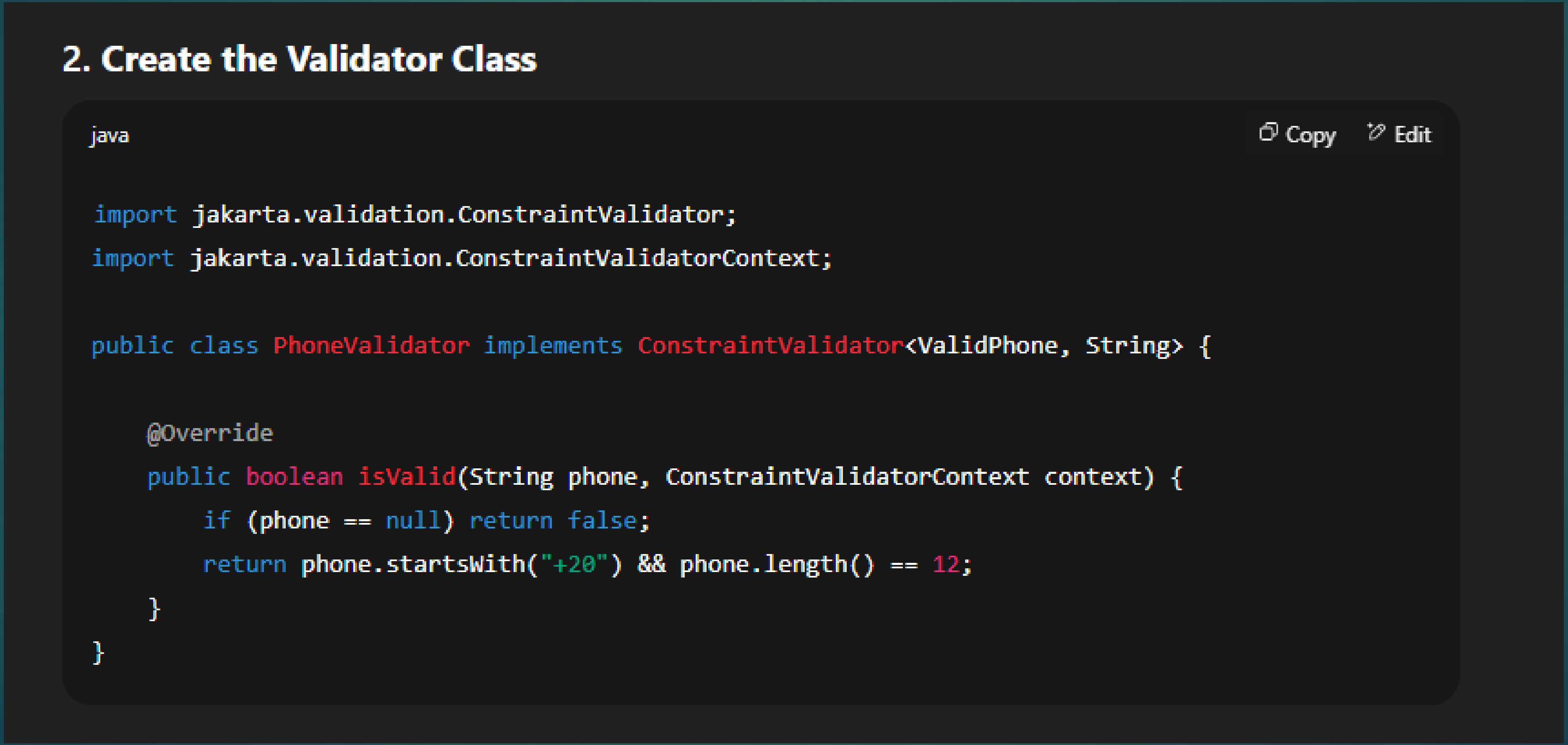
Element	Meaning	Required?	Usually Customized?
`message()`	Default validation error message	✅ Yes	✅ Often customized
`groups()`	For group-based validation	✅ Yes	❌ Rarely used
`payload()`	Custom metadata for constraint	✅ Yes	❌ Rarely used

By Eslam Khder

spring

EraaSoft

custom annotation in Java



2. Create the Validator Class

java

Copy Edit

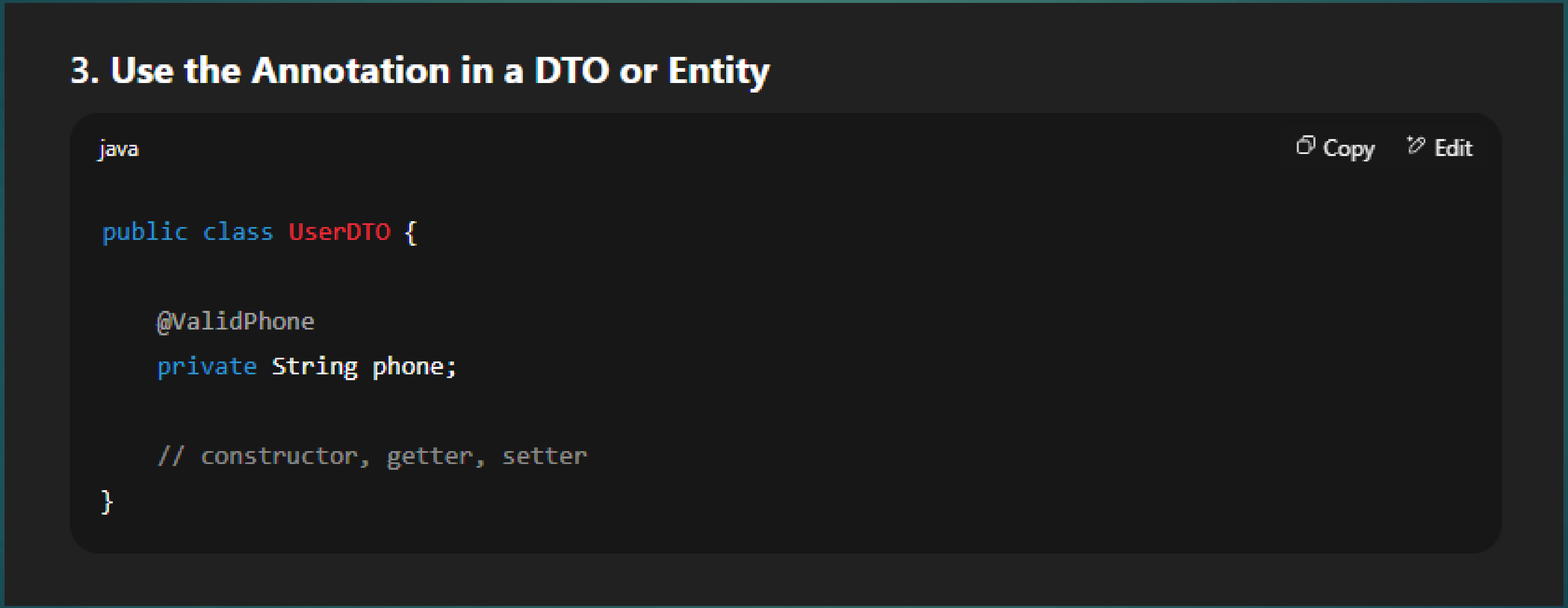
```java
import jakarta.validation.ConstraintValidator;
import jakarta.validation.ConstraintValidatorContext;

public class PhoneValidator implements ConstraintValidator<ValidPhone, String> {

    @Override
    public boolean isValid(String phone, ConstraintValidatorContext context) {
        if (phone == null) return false;
        return phone.startsWith("+20") && phone.length() == 12;
    }
}
```

By Eslam Khder

spring®

EraaSoft

custom annotation in Java



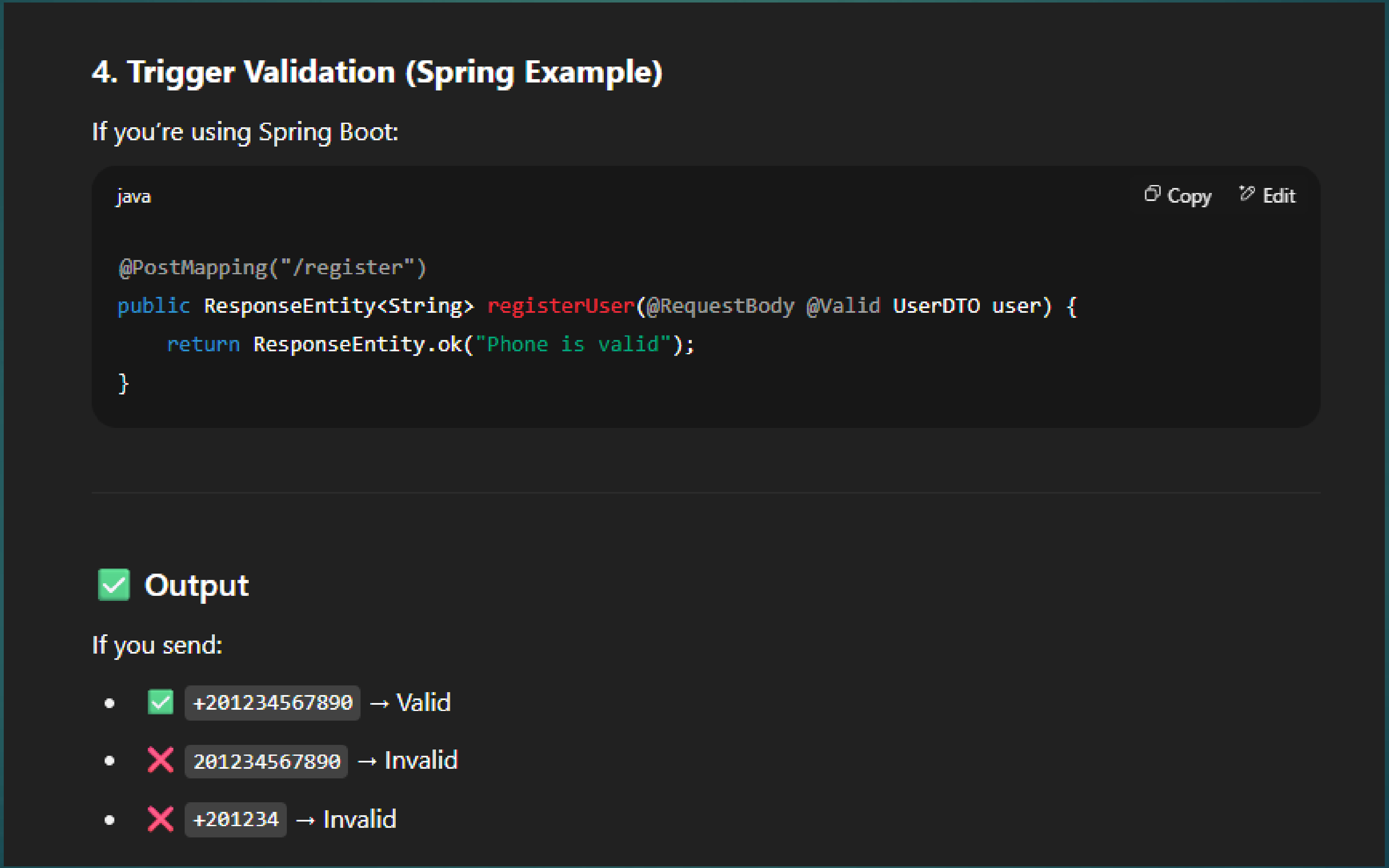
3. Use the Annotation in a DTO or Entity

java

```java
public class UserDTO {

    @ValidPhone
    private String phone;

    // constructor, getter, setter
}
```

By Eslam Khder

custom annotation in Java



4. Trigger Validation (Spring Example)

If you're using Spring Boot:

java

```java
@PostMapping("/register")
public ResponseEntity<String> registerUser(@RequestBody @Valid UserDTO user) {
    return ResponseEntity.ok("Phone is valid");
}
```

✅ Output

If you send:

- ✅ `+201234567890` → Valid
- ❌ `201234567890` → Invalid
- ❌ `+201234` → Invalid

By Eslam Khder

custom annotation in Java

By Eslam Khder

custom annotation in Java

By Eslam Khder