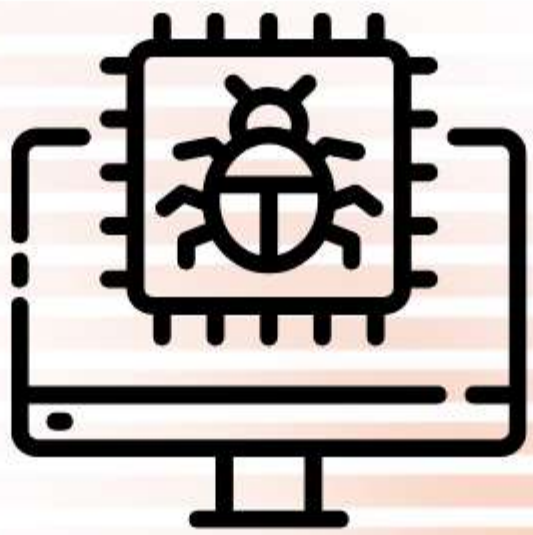
Dasari Vamsi

Avoid



NullPointerException

with

Java Optional

Write cleaner, safer code using
Java 8's Optional class

dasarivamsi.netlify.app

Dasari Vamsi

The Problem

NullPointerException 😬

It's the most common exception in Java — and the hardest to debug in large codebases.

```
if(user != null && user.getEmail() != null) { ... }
```

Messy, right?

dasarivamsi.netlify.app

Dasari Vamsi

The Solution

Meet Optional<T>

Optional is a container object that may or may not hold a non-null value.

```
Optional<String> email = Optional.of("abc@example.com");
```

No more manual null checks!

dasarivamsi.netlify.app

Dasari Vamsi

Creating Optional

How to Create Optional

```
Optional<String> name = Optional.of("Vamsi");

Optional<String> empty = Optional.empty();

Optional<String> nullable = Optional.ofNullable(null);
```

✓ Safe
✓ Clear intent

dasarivamsi.netlify.app

Dasari Vamsi

Accessing Values

Accessing Optional Values

```
optional.ifPresent(value -> System.out.println(value));

optional.orElse("Default");

optional.orElseThrow(() -> new RuntimeException("Missing!"));
```

No null checks needed!

dasarivamsi.netlify.app

Dasari Vamsi

Best Practices

Best Practices with Optional

✅ Use in return types (not method arguments)

✅ Use map, filter, flatMap to transform safely

❌ Don’t overuse it — avoid nested Optionals

dasarivamsi.netlify.app

Dasari Vamsi

Summary

Wrap Up 💬

- Avoids null and improves readability
- Makes APIs more expressive
- Encourages functional style in Java

Have you used Optional in your code yet?

dasarivamsi.netlify.app

Dasari Vamsi

Follow Me to get more Information and tips like this.

dasarivamsi.netlify.app