Optional

Jesper de Jong
Software Architect

@jesperdj www.jesperdj.com

NullPointerException

**Occurs when you dereference a null value,
for example when you call an instance method
or access a field via a variable that has the value null.**

“I call it my billion-dollar mistake. It was the invention of the null reference in 1965.”

Sir Tony Hoare – 2009

Optional as a Safe Alternative to Null

An Optional is a container that can either contain a value or be empty

```
public Customer getCustomer(long id)
```

Optional as a Safe Alternative to Null

An Optional is a container that can either contain a value or be empty

```
public Optional<Customer> getCustomer(long id)
```

Optional as a Safe Alternative to Null

Optional is an immutable value class

Optional as a Safe Alternative to Null

Optional is an immutable value class

Should not be used for synchronization

Optional as a Safe Alternative to Null

Optional is an immutable value class
Should not be used for synchronization

Optional was primarily designed
to be used as a return type for methods

Summary

Class java.util.Optional

- Immutable value class
- Return type for methods

Methods for working with Optional

- Basic methods
- Functional style methods

Up Next: Try-with-resources and AutoCloseable