What's the Optional Type?

Optional is a generic class, whose purpose is to be a container for a value which may or may not be null.

It was created by Java's engineers, to address the problem of the NullPointerException, which is one of the most common errors in Java.

This type is **primarily intended** for use as a **method return type**, under specific conditions.

No Result is valid vs. No Result is a problem

Optional tries to solve the problem of when no result, or no data, is a perfectly valid situation, vs. when no result might really be an error.

You can think of many situations where no data makes sense.

Not everyone has a middle initial in their name, or even a last name for that matter.

No data for a birthdate may or may not be an exception.

New inventory may not yet have a sales price.

Optional is a way of telling you that a value may not be present, therefore you can ignore the value in processing.

Creating an instance of Optional

Optional is just another generic class, so you declare it like any other type, with type arguments.

But you don't construct an Optional.

Instead you use one of the static factory methods, I'm showing here.

Factory Method	When to Use	Best Practice Notes
`Optional<T> empty()`	Use this method to create an Optional that you know has no value.	Never return null from a method that has Optional as a return type.
`Optional<T> of(T value)`	Use this method to create an Optional that you know has a value.	Passing null to this method raises a NullPointerException. Use ofNullable instead, if a possible value might be null.
`Optional<T> ofNullable(T value)`	Use this method to create an Optional when you are uncertain if the value is null or not.	

These methods are empty, of, and of Nullable.

The downside of Optional

- Wrapping elements in Optional will consume more memory and has the possibility of slowing down execution.
- Wrapping elements in Optional adds complexity, and reduces readability of your code.
- Optional is not serializable.
- Using Optional for fields or method parameters is not recommended.

I'll point you to a StackExchange message, from the author of Optional, if you want further reading.

https://stackoverflow.com/questions/26327957/should-java-8-getters-return-optional-type/26328555#26328555