Le pourquoi du comment du type Optional.

Fabrice Sznajderman

Qui vous parle?

Fabrice Sznajderman

Développeur Java / Scala / Web

Twitter : @fsznajderman
Github : fabszn

```
/**
* I promise the following service will never return a NULL reference
*/
Contact c = cs.getContactById(1);

System.out.println("Age =" + c.getAge());
```

```
/**
* I promise the following service will never return a NULL reference
*/
Contact c = cs.getContactById(1);

System.out.println("Age =" + c.getAge());
…
```

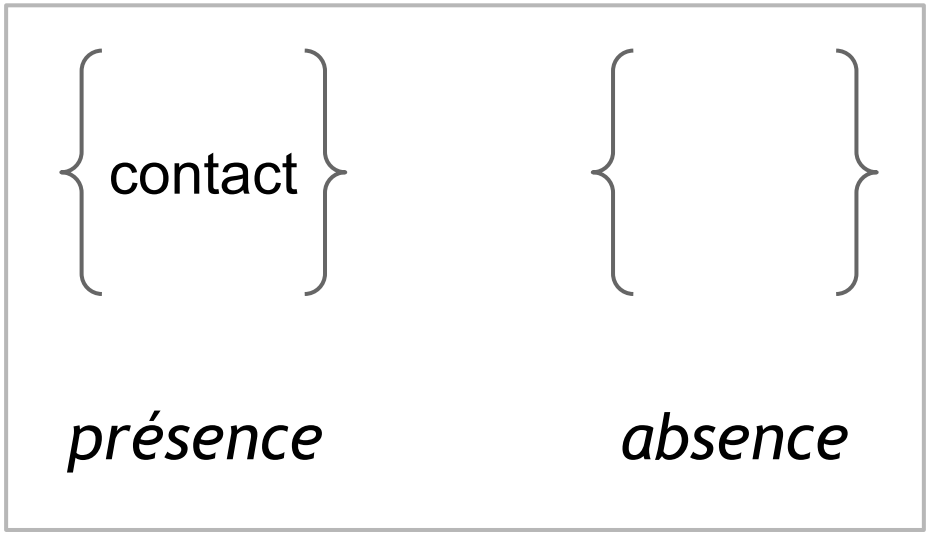
NullPointerException

```
/**
* I promise the following service will never return a NULL reference
*/
Contact c = cs.getContactById(1);

if (c != null) {
   System.out.println("Age =" + c.getAge());
}
…
```

```
/**
* I promise the following service will never return a NULL reference
*/
Contact c = cs.getContactById(1);

if (c != null) {
   System.out.println("Age =" + c.getAge());
}
...
```

Type Optional: Un peu de théorie...

*'**Optional** est un type permettant de caractériser la présence ou l'absence de valeur'*



Type Optional: Un peu de théorie...

```
Contact c1 = null;
Contact c2 = new Contact("Hari","kovair",34);

Optional<Contact> o1 = Optional.ofNullable(c1);
Optional<Contact> o2 = Optional.of(c2);

System.out.println("o1 = " + o1);
System.out.println("o2 = " + o2);
```

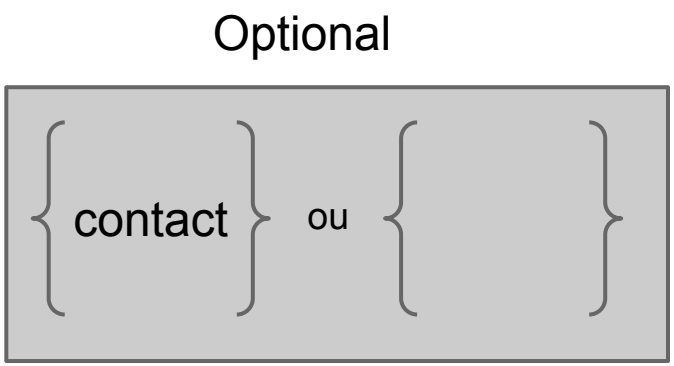
Type Optional: Un peu de théorie...

```
Contact c1 = null;
Contact c2 = new Contact("Hari","kovair",34);

Optional<Contact> o1 = Optional.ofNullable(c1);
Optional<Contact> o2 = Optional.of(c2);

System.out.println("o1 = " + o1);
System.out.println("o2 = " + o2);
```

```
o1 = Optional.empty
o2 = Optional[model.Contact@54bedef2]
```

Type Optional: Un peu de théorie...

*'**Une monade** est comme une boite, vide ou ayant un contenu, qui nous fournit des opérations au dessus de la valeur éventuellement encapsulée.'*



.map()
.orElse()
.get()
.filter()
...

Type Optional : Cas pratiques

Live coding...

Type Optional : Conclusion

- C’est bon pour l’environnement!
- Disparition du pattern if-null
- Documentation automatique
- Disponible dans d’autres langages

```
final Optional<Questions> qs = getQuestions();
```

```
qs.map( q → ”Answers : ” + q.answers ).orElse(“Merci de votre attention!”)
```