

What is **null safety**?

Null safety was introduced in Dart 2.12 and Flutter 2.0.

The main reason for introducing null safety was to **prevent errors** that result from unintentional access of variables set to null.

For example, if a method expects an integer but receives null, your app causes a **runtime error**. This type of error, a null dereference error, can be **difficult to debug**.

Before null safety

```
factory User.fromJson(Map<String, dynamic> json) {  
  return User(  
    id: json["id"] == null ? null : json["id"],  
    name: json["name"] == null ? '' :  
    json["name"]  
  );  
}
```

After null safety

```
name: json["name"] ??  
''
```

 [Leave a comment](#)

 [Save for later](#)



hello@somniosoftware.com

