new spike

JAVA INTERVIEW QUESTIONS

COMMON FUNCTIONAL INTERFACES

Posted By

Andrei Saizu

INTRO

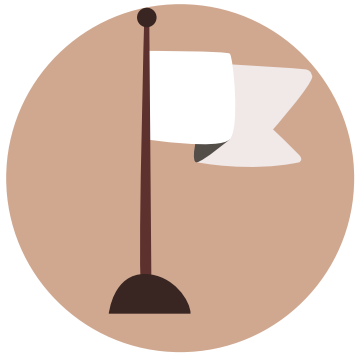
In a previous post, we spoke about the **@FunctionalInterface** feature coming with Java 8.

INTRO

Java developers also created some
out of the box functional interfaces to
make our lives easier.

INTRO

Let's go through the **most common ones**,
see what are they used for and
how to use them on your particular
use-case.

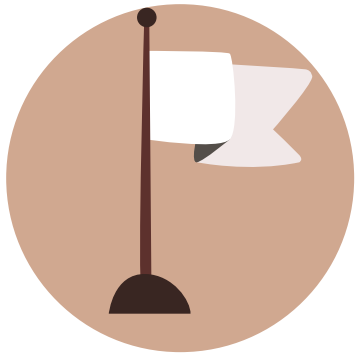


PREDICATE

```
@FunctionalInterface
public interface Predicate<T> {
    boolean test(T input);
}
```

Example:

```
Predicate<Integer> isEvenPredicate = input -> input%2 == 0;
```

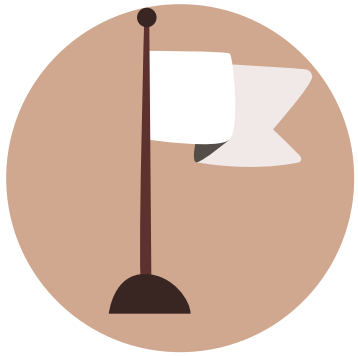


CONSUMER

```
@FunctionalInterface
public interface Consumer<T> {
    void accept(T input);
}
```

Example:

```
Consumer<Integer> loggingConsumer =
                            input -> System.out.println(input);
```

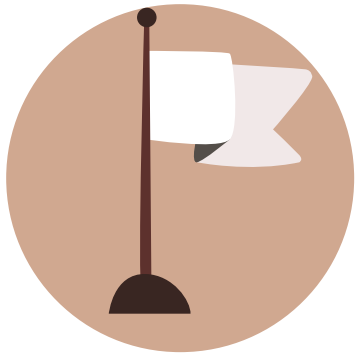


SUPPLIER

```
@FunctionalInterface
public interface Supplier<T> {
    T get();
}
```

Example:

```
Supplier<Double> randomSupplier = () -> Math.random();
```



FUNCTION

```
@FunctionalInterface
public interface Function<T, U> {
    U apply(T input);
}
```

Example:

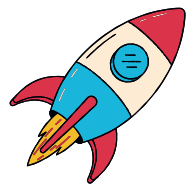
```
Function<String, User> userFunction = input -> new User(input);
```

BONUS

There are also a few derived interfaces to use when you need two inputs:

- BiFunction<T,U,R>
- BiPredicate<T,U>
- BiConsumer<T,U>

STAY UP TO DATE!



Check out the link in the comments

for a free in-depth breakdown on

how it works in the background!