

JAVA 8

Find Methods :
findFirst vs findAny



@techy.tacos

@techy.tacos

findAny method returns **any arbitrary element in the stream**. We use it when we're looking for an element without paying an attention to any specific order. This method returns an Optional instance, which is empty if the Stream is empty. In a non-parallel operation, **it will most likely return the first element** in the Stream, but there is no guarantee for this.

Example : Return any element from the current stream.

```
public class DemoFindAndMatch {  
    public static void main(String[] args) {  
        // Creating a Stream of Strings  
        Stream<String> stream = Stream.of("Albert", "ram", "Monica");  
        Optional<String> result = stream.filter(str -> Character.isUpperCase(str.charAt(0)))  
                                         .findAny();  
        System.out.println(result);  
    }  
}  
//output: Optional[Albert]
```

@techy.tacos

findFirst method element returns the first element in the stream. This is useful in a situation where stream has a specific *encounter order* and we need to get the first element from this sorted order. When there is no encounter order, it returns any element from the Stream.

Example : Return the first element from the current sorted stream.

```
public class DemoFindAndMatch {  
    public static void main(String[] args) {  
        // Creating a Stream of Strings  
        Stream<String> stream = Stream.of("ram", "Monica", "Albert");  
        Optional<String> result = stream.filter(str -> Character.isUpperCase(str.charAt(0)))  
                                       .sorted()  
                                       .findFirst();  
        System.out.println(result);  
    }  
}  
//output: Optional[Albert]
```