

Stream filter() in Java with examples

Difficulty Level : Medium Last Updated : 06 Dec, 2018

Stream filter(Predicate predicate) returns a stream consisting of the elements of this stream that match the given predicate. This is an *intermediate operation*. These operations are always lazy i.e, executing an intermediate operation such as filter() does not actually perform any filtering, but instead creates a new stream that, when traversed, contains the elements of the initial stream that match the given predicate.

Syntax :

```
Stream<T> filter(Predicate<? super T> predicate)
```

Where, Stream is an interface and T is the type of the input to the predicate.
The function returns the new stream.

Example 1 : filter() method with operation of filtering out the elements divisible by 5.

```
// Java code for Stream filter
// (Predicate predicate) to get a stream
// consisting of the elements of this
// stream that match the given predicate.
import java.util.*;

class GFG {

    // Driver code
    public static void main(String[] args)
    {

        // Creating a list of Integers
        List<Integer> list = Arrays.asList(3, 4, 6, 12, 20);

        // Using Stream filter(Predicate predicate)
        // to get a stream consisting of the
        // elements that are divisible by 5
        list.stream().filter(num -> num % 5 == 0).forEach(System.out::println)
    }
}
```

Output :

20

Example 2 : filter() method with operation of filtering out the elements with upperCase letter at index 1.

```
// Java code for Stream filter
// (Predicate predicate) to get a stream
// consisting of the elements of this
// stream that match the given predicate.
import java.util.stream.Stream;

class GFG {

    // Driver code
    public static void main(String[] args)
    {

        // Creating a Stream of Strings
        Stream<String> stream = Stream.of("Geeks", "fOr",
                                         "GEEKSQUIZ", "GeeksforGeeks");

        // Using Stream filter(Predicate predicate)
        // to get a stream consisting of the
        // elements having UpperCase Character
        // at index 1
        stream.filter(str -> Character.isUpperCase(str.charAt(1)))
              .forEach(System.out::println);
    }
}
```

Output :

fOr
GEEKSQUIZ

Example 3 : filter() method with operation of filtering out the elements ending with s.

```
// Java code for Stream filter
// (Predicate predicate) to get a stream
```

```
// consisting of the elements of this
// stream that match the given predicate.
import java.util.stream.Stream;

class GFG {

    // Driver code
    public static void main(String[] args)
    {

        // Creating a Stream of Strings
        Stream<String> stream = Stream.of("Geeks", "for",
                                         "GeEksQuiz", "GeeksforGeeks");

        // Using Stream filter(Predicate predicate)
        // to get a stream consisting of the
        // elements ending with s
        stream.filter(str -> str.endsWith("s"))
              .forEach(System.out::println);
    }
}
```

Output :

Geeks

GeeksforGeeks