

Java ListIterator interface

📅 Last Updated: August 30, 2020 👤 By: Lokesh Gupta 📁 Java Collections 🔖 Java Iterator

Java ListIterator interface is **bi-directional iterator** which is used to iterate over the elements of [list](#) in either direction *previous* or *next*.

We can obtain the reference to list iterator for any given list using **list.listIterator()** method call. Follow given **ListIterator syntax**.

ListIterator Syntax

```
ListIterator<T> listIterator = list.listIterator();
```

1. Java ListIterator Features

Following is a list of features provided by **ListIterator in Java**.

1. ListIterator is available since Java 1.2.
2. ListIterator **extends Iterator** interface.
3. ListIterator is applicable only for List implemented classes.
4. Unlike Iterator, ListIterator supports all CRUD operations (CREATE, READ, UPDATE and DELETE) over a list of elements.
5. Unlike Iterator, ListIterator is **bi-directional**. It supports both forward direction and backward direction iterations.

6. It has no current element; its cursor position always lies between the element that would be returned by a call to `previous()` and the element that would be returned by a call to `next()`.

2. Java ListIterator Example

Let's start with a simple Java example of ListIterator to iterate over elements of an ArrayList. Please note that array indices start with '0'.

ListIterator example

```
ArrayList<String> list = new ArrayList<>();

list.add("A");
list.add("B");
list.add("C");
list.add("D");
list.add("E");
list.add("F");

ListIterator<String> listIterator = list.listIterator();

System.out.println("Forward iteration");

//Forward iterator
while(listIterator.hasNext()) {
    System.out.print(listIterator.next() + ",");
}

System.out.println("Backward iteration");

//Backward iterator
while(listIterator.hasPrevious()) {
    System.out.print(listIterator.previous() + ",");
}

System.out.println("Iteration from specified position");

//Start iterating from index 2
listIterator = list.listIterator(2);

while(listIterator.hasNext()) {
    System.out.print(listIterator.next() + ",");
}
```

Program Output.

Console

Forward iteration

A,B,C,D,E,F,

Backward iteration

F,E,D,C,B,A,

Iteration from specified position

C,D,E,F,

3. Java ListIterator Methods

1. **void add(Object o)** : Inserts the specified element into the list (optional operation).
2. **boolean hasNext()** : Returns **true** if this list iterator has more elements when traversing the list in the forward direction.
3. **boolean hasPrevious()** : Returns **true** if this list iterator has more elements when traversing the list in the reverse direction.
4. **Object next()** : Returns the next element in the list and advances the cursor position.
5. **int nextIndex()** : Returns the index of the element that would be returned by a subsequent call to **next()**.
6. **Object previous()** : Returns the previous element in the list and moves the cursor position backwards.
7. **int previousIndex()** : Returns the index of the element that would be returned by a subsequent call to **previous()**.
8. **void remove()** : Removes from the list the last element that was returned by **next()** or **previous()** (optional operation).
9. **void set(Object o)** : Replaces the last element returned by **next()** or **previous()** with the specified element (optional operation).

4. Conclusion

In this tutorial, we learned the Java ListIterator interface. We learned the ListIterator methods and simple examples to iterate over list elements in forward and backward direction.

Drop me your questions in comments section.

Happy Learning !!

References:

[ListIterator Interface Java Docs](#)

Was this post helpful?

Let us know if you liked the post. That's the only way we can improve.

Yes

No

Recommended Reading:

1. [ArrayList listIterator\(\) method example](#)
2. [Java Iterator interface example](#)
3. [Java Spliterator interface](#)
4. [Java Comparable Interface](#)
5. [Java Comparator Interface](#)
6. [Interface vs Abstract Class in Java](#)
7. [Java Cloneable interface – Is it broken?](#)

8. [Private Methods in Interface – Java 9](#)
9. [Spring boot – CommandLineRunner interface example](#)
0. [Spring Boot Async Rest Controller with Callable Interface](#)

Join 7000+ Awesome Developers

Get the latest updates from industry, awesome resources, blog updates and much more.

Email Address

Subscribe

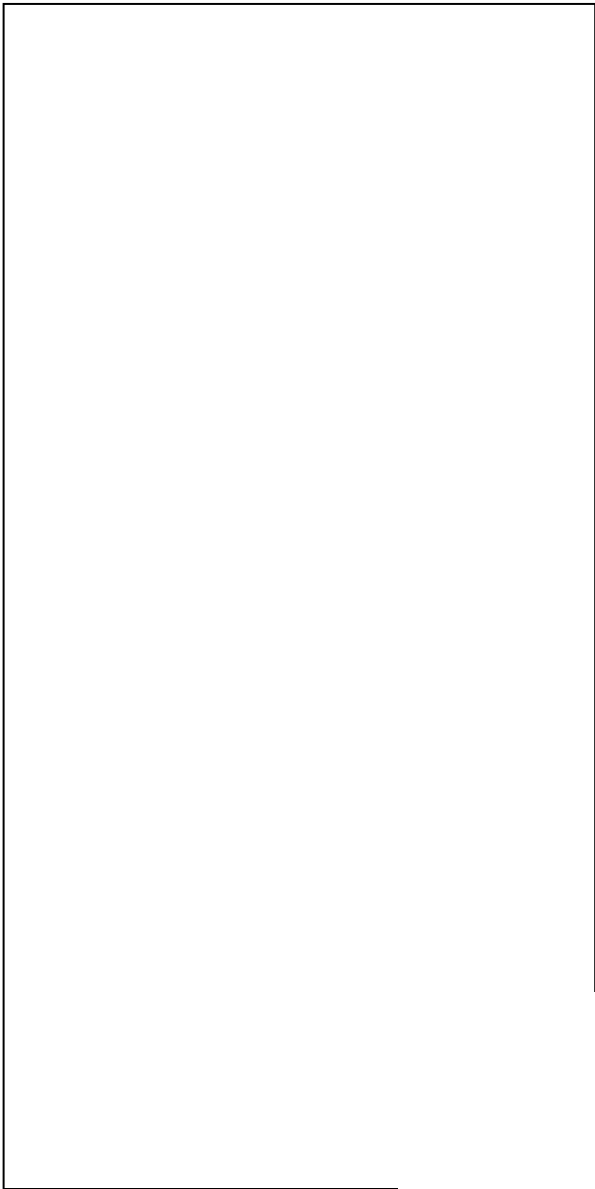
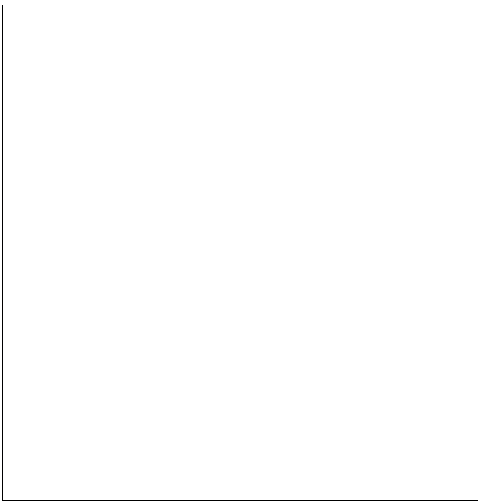
** We do not spam !!*

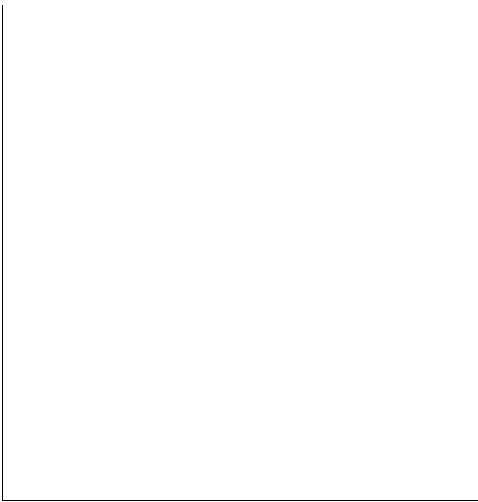
Leave a Comment

☐ Add me to your newsletter and keep me updated whenever you publish new blog posts

Post Comment







HowToDoInJava

A blog about Java and related technologies, the best practices, algorithms, and interview questions.

Meta Links

- [About Me](#)
- [Contact Us](#)
- [Privacy policy](#)
- [Advertise](#)
- [Guest Posts](#)

Blogs

[REST API Tutorial](#)



Copyright © 2022 · Hosted on [Cloudways](#) · [Sitemap](#)