HowToDoInJava

Java CopyOnWriteArraySet class

Last Updated: August 30, 2020 By: Lokesh Gupta Java Collections Java Collections, Java HashSet

Java CopyOnWriteArraySet is a thread-safe variant of **HashSet** which uses a underlying `CopyOnWriteArrayList` for all of its operations.

Similar to CopyOnWriteArrayList, it's **immutable snapshot** style iterator method uses a reference to the state of the array (inside the backing list) at the point that the iterator was created. This helps in usecases when traversal operations vastly outnumber set update operations and we do not want to synchronize the traversals and still want thread safety while updating the set.

Table of Contents

1. CopyOnWriteArraySet Hierarchy

The `CopyOnWriteArraySet` class extends `AbstractSet` class and implements `Serializable` interface.

```
CopyOnWriteArraySet.java

public class CopyOnWriteArraySet<E>
    extends AbstractSet<E>
    implements Serializable

{
  private final CopyOnWriteArrayList<E> al;

  //implementation
}
```

2. CopyOnWriteArraySet Features

The important things to learn about Java CopyOnWriteArraySet class are:

- As normal set data structure, it does not allow duplicates.
- CopyOnWriteArraySet class implement `Serializable` interface and extends `AbstractSet` class.
- Using CopyOnWriteArraySet is costly for update operations, bacause each mutation creates a cloned copy of underlying array and add/update element to it.
- It is thread-safe version of HashSet. Each thread accessing the set sees its own version of snapshot of backing array created while initializing the iterator for this set.
- Because it gets snapshot of underlying array while creating iterator, it **does not throw ConcurrentModificationException**.
- Mutation operations on iterators are not supported. These methods throw `UnsupportedOperationException`.
- CopyOnWriteArraySet is a concurrent replacement for a **synchronized Set** and offers better concurrency when iterations outnumber mutations.

- It allows duplicate elements and heterogeneous Objects (use generics to get compile time errors).
- Because it creates a new copy of underlying array everytime iterator is created, **performance is slower** than HashSet.

3. Java CopyOnWriteArraySet Example

Java program to show how iterators created at different times sees through snapshot version of set in CopyOnWriteArraySet. In given example, we first created list and **itr1** when list had elements (1,2,3).

Then we added one more element to list and again created an iterator **itr2**.

Finally we verified the elements in both iterators.

```
CopyOnWriteArraySet Example

CopyOnWriteArraySet<Integer> set = new CopyOnWriteArraySet<>(Arrays.asList(1,2,3))

System.out.println(set);  //[1, 2, 3]

//Get iterator 1
Iterator<Integer> itr1 = set.iterator();

//Add one element and verify set is updated
set.add(4);
System.out.println(set);  //[1, 2, 3, 4]

//Get iterator 2
Iterator<Integer> itr2 = set.iterator();

System.out.println("====Verify Iterator 1 content====");

itr1.forEachRemaining(System.out :: println); //1,2,3

System.out.println("====Verify Iterator 2 content====");

itr2.forEachRemaining(System.out :: println); //1,2,3,4
```

Program Output.

```
Console

[1, 2, 3]
[1, 2, 3, 4]
====Verify Iterator 1 content====
1
2
3
====Verify Iterator 2 content====
1
2
3
4
```

4. CopyOnWriteArraySet Constructors

- **CopyOnWriteArraySet()** : Creates an empty set.
- **CopyOnWriteArraySet(Collection c)** : Creates a set containing the elements of the specified collection, in the order they are returned by the collection's iterator.

5. CopyOnWriteArraySet Methods

- **boolean add(object o)** : Adds the specified element to this set if it is not already present.
- **boolean addAll(collection c)** : Adds all of the elements in the specified collection to this set if they're not already present.
- **void clear()** : Removes all of the elements from this set.
- **boolean contains(Object o)** : Returns true if this set contains the specified element.
- **boolean isEmpty()** : Returns true if this set contains no elements.
- **Iterator iterator()** : Returns an iterator over the elements contained in this set in the order in which these elements were added.
- **boolean remove(Object o)** : Removes the specified element from this set if it is present.

- **int size()** : Returns the number of elements in this set.

6. Java CopyOnWriteArraySet Usecases

Use CopyOnWriteArraySet in applications in which set sizes generally stay small, read-only operations vastly outnumber mutative operations, and you need to prevent interference among threads during traversal.

CopyOnWriteArraySet helps in minimizing programmer controlled synchronization steps and move control to inbuilt, well tested APIs.

7. Java CopyOnWriteArraySet Performance

Due to added step of creating a new backing array everytime the set is updated, it performs worse than HashSet.
There is no performance overhead on read operations and both classes perform same.

8. Conclusion

In this Java Collection tutorial, we learned to use **CopyOnWriteArraySet** class, it's constrcutors, methods and usecases.

We learned the **CopyOnWriteArraySet internal working in java** as well as **CopyOnWriteArraySet vs CopyOnWriteArrayList**.

We gone through **Java CopyOnWriteArraySet example program** to demo how snapshot iterators works.

Drop me your questions in comments.

Happy Learning !!

Reference:

CopyOnWriteArraySet 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. [Solved]: javax.xml.bind.JAXBException: class java.util.ArrayList nor any of its super class is known to this context
2. Java TransferQueue – Java LinkedTransferQueue class
3. Java TreeMap class
4. Java HashSet class
5. Java LinkedHashSet class
6. Java PriorityBlockingQueue class
7. Java ArrayBlockingQueue class
8. Java CopyOnWriteArrayList class
9. Java TreeSet class
0. Java LinkedList class

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

Name *

Email *

Website

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

Post Comment

Search ...

VIDEOGAMER
NEW SOUTH PARK GAME WITHOUT UBISOFT COMING?!
» CLICK HERE «

VIDEOGAMER
NEW SOUTH PARK GAME WITHOUT UBISOFT COMING?!
» CLICK HERE «

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