

## **ACC and Guava**

- Apache Common Collections
- Google Guava

## Apache Commons Collections

<https://github.com/apache/commons-collections>

The purpose of the Commons is to provide reusable, open source Java software.

```
<dependency>  
  <groupId>org.apache.commons</groupId>  
  <artifactId>commons-collections4</artifactId>  
  <version>4.1</version>  
</dependency>
```

**org.apache.commons.collections4.CollectionUtils**

**Returns a Collection containing the union of the given Iterables**

static <O> Collection<O> union(final Iterable<? extends O> a, final Iterable<? extends O> b)

**Returns a Collection containing the intersection of the given Iterables.**

static <O> Collection<O> intersection(final Iterable<? extends O> a, final Iterable<? extends O> b)

**Returns a new Collection containing a - b.**

static <O> Collection<O> subtract(final Iterable<? extends O> a, final Iterable<? extends O> b)

**Merges two sorted Collections, a and b, into a single, sorted List**

static <O extends Comparable<? super O>> List<O> collate(Iterable<? extends O> a, Iterable<? extends O> b)

**Returns a Collection of all the permutations of the input collection.**

static <E> Collection<List<E>> permutations(final Collection<E> collection)

## Google Guava

- com.google.common.collect.Collections2
- com.google.common.collect.Lists
- com.google.common.collect.Sets

<https://github.com/google/guava/wiki/CollectionUtilitiesExplained>

Google Guava is an open-source set of common libraries for Java, mainly developed by Google engineers.

Google Guava contains basic utilities to reduce menial labors to implement common methods and behaviors.

```
<dependency>  
  <groupId>com.google.guava</groupId>  
  <artifactId>guava</artifactId>  
  <version>21.0</version>  
</dependency>
```

## Collections2

**Returns the elements of unfiltered that satisfy a predicate.**

```
static <E> Collection<E> filter(Collection<E> unfiltered, Predicate<? super E> predicate)
```

**Returns a collection that applies function to each element of fromCollection.**

```
static <F,T> Collection<T> transform(Collection<F> fromCollection, Function<? super F,T> function)
```

**Returns a Collection of all the permutations of the specified Collection.**

```
static <E> Collection<List<E>> permutations(Collection<E> elements)
```

## Lists

**Returns every possible list that can be formed by choosing one element from each of the given lists in order; the "n-ary Cartesian product" of the lists.**

```
static <B> List<List<B>> cartesianProduct(List<? extends B>... lists)
```

**Returns consecutive sublists of a list, each of the same size (the final list may be smaller).**

```
static <T> List<List<T>> partition(List<T> list, int size)
```

## Sets

**Returns an unmodifiable view of the difference of two sets.**

```
static <E> Sets.SetView<E> difference(Set<E> set1, Set<?> set2)
```

**Returns an unmodifiable view of the intersection of two sets.**

```
static <E> Sets.SetView<E> intersection(Set<E> set1, Set<?> set2)
```

**Returns an unmodifiable view of the union of two sets.**

```
static <E> Sets.SetView<E> union(Set<? extends E> set1, Set<? extends E> set2)
```