

Guava Release 11.0.2: Release Notes

Release 11.0 was released on December 18, 2011. Release 11.0.1 was released on January 9, 2012. Release 11.0.2 was released on February 22, 2012. (See [ReleaseHistory](#).)

[Full API Documentation](#)

Using Guava in your project

This release will be identified in the Maven Central repository as [com.google.guava:guava:11.0.2](#) and [com.google.guava:guava-gwt:11.0.2](#)

See [UseGuavaInYourBuild](#) for help integrating Guava into your build environment.

If you don't use managed dependencies, you can also just manually download JARs of the classes, sources and documentation from:

- [guava-11.0.2.jar](#)
- [guava-gwt-11.0.2.jar](#) (for GWT users)
- [guava-11.0.2-javadoc.jar](#) (Javadoc)
- [guava-11.0.2-sources.jar](#) (Source)

API Changes

[Full JDiff Report](#) of changes since release 10.0.1

To build a combined report of the API changes between release 11.0.2 and any older release, check out our docs tree and run `jdiff/jdiff.sh` with the previous release number as argument (example: `jdiff.sh 5.0`).

Changes between 11.0.1 and 11.0.2

- Fix BloomFilter performance bug, breaking backwards compatibility in the process.
- Fix GWT compilation of CacheBuilder.

Changes between 11.0 and 11.0.1

- Fix bug in `Optional#presentInstances`.
- Make `MessageDigestHashFunction` thread-safe.
- Fix Cache clear in the absence of a removal listener.
- Cache documentation improvements.

LoadingCache split out of Cache

The [Cache](#) methods related to automatic cache loading have been split out into the [LoadingCache](#) interface. [CacheBuilder.build\(CacheLoader\)](#) now returns a `LoadingCache`, and [CacheBuilder.build\(\)](#) was introduced, which returns a `Cache`.

When upgrading to version 11.0.2, all `Cache` references should be changed to `LoadingCache`.

Issues resolved

[Issues fixed](#)

Additions

New packages

- [common.hash](#)
- [common.math](#)

New classes

common.cache

- [LoadingCache](#)
- [AbstractLoadingCache](#)
- [ForwardingLoadingCache](#)
- [Weigher](#)

common.collect

- [ImmutableTable](#)
- [Queues](#)
- [SortedMultiset](#)

common.net

- [HttpHeaders](#)

common.primitives

- [UnsignedInteger](#)
- [UnsignedInts](#)
- [UnsignedLong](#)

common.util.concurrent

- [AbstractScheduledService](#)
- [AsyncFunction](#)
- [AtomicDouble](#)
- [AtomicDoubleArray](#)
- [AtomicLongMap](#)

New members

common.base

- Objects.ToStringHelper.add/addValue primitive overloads
- Optional.asSet, presentInstances
- Strings.commonPrefix, commonSuffix

common.cache

- Cache.get, getIfPresent, getAllPresent, invalidateAll, put
- CacheBuilder.build, maximumWeight, refreshAfterWrite, weigher
- CacheLoader.loadAll, reload
- CacheStats.plus

common.collect

- ImmutableMultimap.inverse
- Iterables.mergeSorted, tryFind
- Iterators.mergeSorted, tryFind
- Maps.difference, filterEntries, filterKeys, filterValues, transformEntries, transformValues

- `Multimaps.filterEntries`, `filterKeys`, `filterValues`
- `Multisets.copyHighestCountFirst`, `unmodifiableSortedMultiset`
- `Ordering.max`, `min` (Iterator overload)
- `Sets.filter` (`SortedSet` overload), `newLinkedHashSetWithExpectedSize`
- `Tables.unmodifiableRowSortedTable`, `unmodifiableTable`

`common.io`

- `Files.getFileExtension`, `simplifyPath`

`common.primitives`

- `Ints.tryParse`

`common.util.concurrent`

- `Futures.transform(..., AsyncFunction)` (to replace `Futures.chain`)