Guava: Google Core Libraries for Java

release v31.1 CI passing

Guava is a set of core Java libraries from Google that includes new collection types (such as multimap and multiset), immutable collections, a graph library, and utilities for concurrency, I/O, hashing, caching, primitives, strings, and more! It is widely used on most Java projects within Google, and widely used by many other companies as well.

Guava comes in two flavors:

- The JRE flavor requires JDK 1.8 or higher.
- If you need support for Android, use the Android flavor. You can find the Android Guava source in the `android` directory.

Adding Guava to your build

Guava's Maven group ID is `com.google.guava`, and its artifact ID is `guava`. Guava provides two different "flavors": one for use on a (Java 8+) JRE and one for use on Android or by any library that wants to be compatible with Android. These flavors are specified in the Maven version field as either `31.1-jre` or `31.1-android`. For more about depending on Guava, see using Guava in your build.

To add a dependency on Guava using Maven, use the following:

```
<dependency>
  <groupId>com.google.guava</groupId>
  <artifactId>guava</artifactId>
  <version>31.1-jre</version>
  <!-- or, for Android: -->
  <version>31.1-android</version>
</dependency>
```

To add a dependency using Gradle:

```
dependencies {
  // Pick one:

  // 1. Use Guava in your implementation only:
  implementation("com.google.guava:guava:31.1-jre")

  // 2. Use Guava types in your public API:
  api("com.google.guava:guava:31.1-jre")

  // 3. Android - Use Guava in your implementation only:
  implementation("com.google.guava:guava:31.1-android")

  // 4. Android - Use Guava types in your public API:
  api("com.google.guava:guava:31.1-android")
}
```

For more information on when to use `api` and when to use `implementation`, consult the Gradle documentation on API and implementation separation.

Snapshots and Documentation

Snapshots of Guava built from the `master` branch are available through Maven using version `HEAD-jre-SNAPSHOT`, or `HEAD-android-SNAPSHOT` for the Android flavor.

- Snapshot API Docs: guava
- Snapshot API Diffs: guava

Learn about Guava

- Our users' guide, Guava Explained
- A nice collection of other helpful links

Links

- GitHub project
- Issue tracker: Report a defect or feature request
- StackOverflow: Ask "how-to" and "why-didn't-it-work" questions
- guava-announce: Announcements of releases and upcoming significant changes
- guava-discuss: For open-ended questions and discussion

IMPORTANT WARNINGS

1. APIs marked with the `@Beta` annotation at the class or method level are subject to change. They can be modified in any way, or even removed, at any time. If your code is a library itself (i.e., it is used on the CLASSPATH of users outside your own control), you should not use beta APIs unless you repackage them. **If your code is a library, we strongly recommend using the Guava Beta Checker to ensure that you do not use any `@Beta` APIs!**

2. APIs without `@Beta` will remain binary-compatible for the indefinite future. (Previously, we sometimes removed such APIs after a deprecation period. The last release to remove non-`@Beta` APIs was Guava 21.0.) Even `@Deprecated` APIs will remain (again, unless they are `@Beta`). We have no plans to start removing things again, but officially, we're leaving our options open in case of surprises (like, say, a serious security problem).

3. Guava has one dependency that is needed for linkage at runtime: `com.google.guava:failureaccess:1.0.1`. It also has some annotation-only dependencies, which we discuss in more detail at that link.

4. Serialized forms of ALL objects are subject to change unless noted otherwise. Do not persist these and assume they can be read by a future version of the library.

5. Our classes are not designed to protect against a malicious caller. You should not use them for communication between trusted and untrusted code.

6. For the mainline flavor, we test the libraries using only OpenJDK 8 and OpenJDK 11 on Linux. Some features, especially in `com.google.common.io`, may not work correctly in other environments. For the Android flavor, our unit tests also run on API level 15 (Ice Cream Sandwich).