

ROSEAU  (REED)

Fast, Accurate, Source-based
API Breaking Change Analysis in Java

LaBRI TU/e



Corentin Latappy



Thomas Degueule



Jean-Rémy Falleri



Romain Robbes



Lina Ochoa Venegas

Software Libraries

google / **guava** Public

Watch 2.5k Star 42.3k Fork 9.4k

Code Issues 640 Pull requests 77 Actions Projects Wiki Security Insights

master 14 branches 93 tags Go to file Add file Code

About

Google core libraries for Java

java guava

Readme

Apache-2.0 License

eamonmcmannus and Google Java Core Libraries ... cd3b419 2 days ago 5,625 commits

.github	Bump styfle/cancel-workflow-action from 0.9.0 to 0.9.1	last month
android	Add <code>ImmutableMap.ofEntries</code> .	2 days ago

JUnit 5 JUnit 4

The 5th major version of the programmer-friendly testing framework for Java and the JVM

User Guide Javadoc Code & Issues Q & A Support JUnit

 **Apache Commons**
<http://commons.apache.org/>

Apache Commons Collections™ Last Published: 09 July 2019 | Version: 4.4

Collections

Overview
Download
Security Reports
Users guide
History
Release Notes 4.4
Release Notes 4.3

Commons Collections

The [Java Collections Framework](#) was a major addition in JDK 1.2. It added many powerful data structures. Commons-Collections seek to build upon the JDK classes by providing new interfaces, implementations, and utilities.

- Bag interface for collections that have a number of copies of each object
- BiMap interface for maps that can be looked up from value to key as well as key to value
- MapIterator interface to provide simple and quick iteration over maps

Software Libraries Evolution

google / guava Public

Watch 2.5k Star 42.3k Fork 9.4k

<> Code Issues 640 Pull requests 77 Actions Projects Wiki Security Insights

master 14 branches 93 tags Go to file Add file Code

eamonnmcmanus and Google Java Core Libraries ... cd3b419 2 days ago 5,625 commits

.github	Bump styfle/cancel-workflow-action from 0.9.0 to 0.9.1	last month
android	Add <code>ImmutableMap.ofEntries</code> .	2 days ago

About

Google core libraries for Java

java guava

Readme

Apache-2.0 License

Software Libraries Evolution

google / **guava** Public

Watch 2.5k Star 42.3k Fork 9.4k

Code Issues 640 Pull requests 77 Actions Projects Wiki Security Insights

master 14 branches 93 tags

Go to file Add file Code

About

Google core libraries for Java

java guava

Readme

Apache-2.0 License

eamonnmcmanus and Google Java Core Libraries ... cd3b419 2 days ago 5,625 commits

.github	Bump styfle/cancel-workflow-action from 0.9.0 to 0.9.1	last month
android	Add ImmutableMap.ofEntries.	2 days ago

Repositories that depend on com.google.guava:guava			Package: com.google.guava:guava
275,122 Repositories 0 Packages			
aws	aws / amazon-documentdb-jdbc-driver	☆ 6 🍴 5	
kamilinw	Lift-managemennt	☆ 0 🍴 0	
chahrazedbb	OPTIMA	☆ 0 🍴 0	
ryanrousseau	AppBuilder-EKS	☆ 0 🍴 0	

Software Libraries Evolution

google / **guava** Public

Watch 2.5k Star 42.3k Fork 9.4k

Code Issues 640 Pull requests 77 Actions Projects Wiki Security Insights

master 14 branches 93 tags

Go to file Add file Code

About

Google core libraries for Java

java guava

Readme

Apache-2.0 License

eamonnmcmanus and Google Java Core Libraries ... cd3b419 2 days ago 5,625 commits

- .github Bump styfle/cancel-workflow-action from 0.9.0 to 0.9.1 last month
- android Add ImmutableMap.ofEntries. 2 days ago

Repositories that depend on com.google.guava:guava		Package: com.google.guava:guava
275,122 Repositories 0 Packages		
aws / amazon-documentdb-jdbc-driver	☆ 6 🐞 5	
kamilinw / Lift-managemennt	☆ 0 🐞 0	
chahrazedbb / OPTIMA	☆ 0 🐞 0	
ryanrousseau / AppBuilder-EKS	☆ 0 🐞 0	

```
guava/src/com/google/common/collect/TopKSelector.java

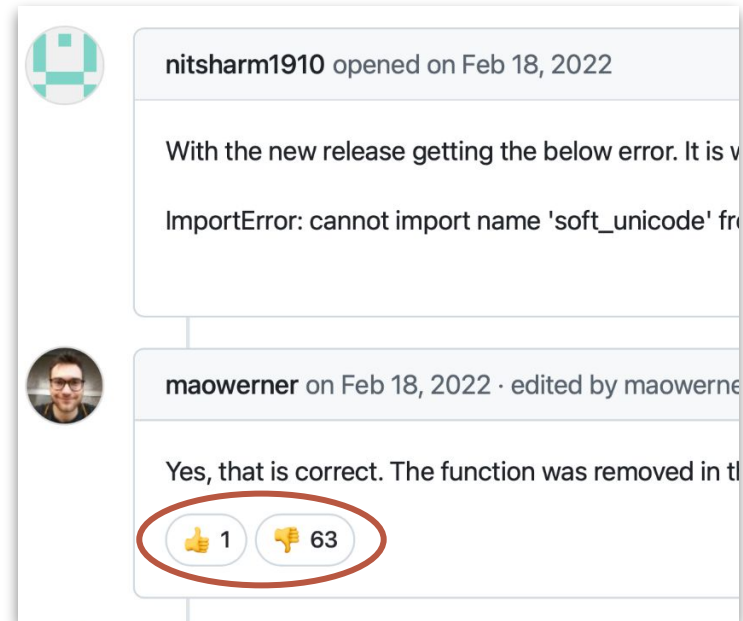
@@ -186,7 +186,7 @@ private void trim() {
186 186     iterations++;
187 187     if (iterations >= maxIterations) {
188 188         // We've already taken O(k log k), let's make sure we don't take longer than O(k log k).
189 - Arrays.sort(buffer, left, right, comparator);
189 + Arrays.sort(buffer, left, right + 1, comparator);
190 190     break;
191 191 }
192 192 }
```

Breaking Changes in Practice

- Why?

Breaking Changes in Practice

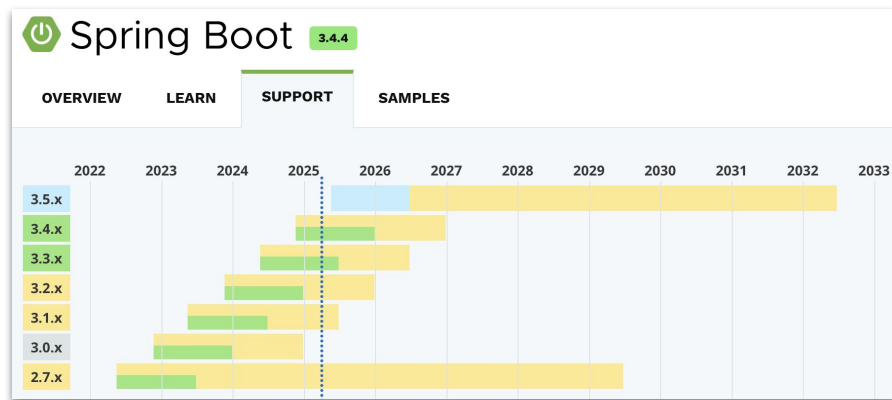
- Why?
 - Keep users confidence



<https://github.com/pallets/markupsafe/issues/284>

Breaking Changes in Practice

- Why?
 - Keep users confidence
 - Avoid multiple versions support



<https://spring.io/projects/spring-boot#support>

Breaking Changes in Practice

- Why?
 - Keep users confidence
 - Avoid multiple versions support
- How?
 - Policies

guava / README.md

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:

<https://github.com/google/guava/blob/master/README.md#important-warnings>

Breaking Changes in Practice

- Why?
 - Keep users confidence
 - Avoid multiple versions support
- How?
 - Policies
 - Tools



[cargo-semver-checks](#)



[gocompat](#)
[gorelease](#)



[AexPy](#)



[JApiCmp](#)
[Revapi](#)

Breaking Changes in SE Research

- Studies: *why, how, when* do libraries break, how does it affect clients?
 - Xavier et al.: 317 libraries, 9k releases¹
 - Ochoa et al.: 14k libraries, 120k releases²
 - Jayasuriya et al.: 7k libraries, 18k releases³

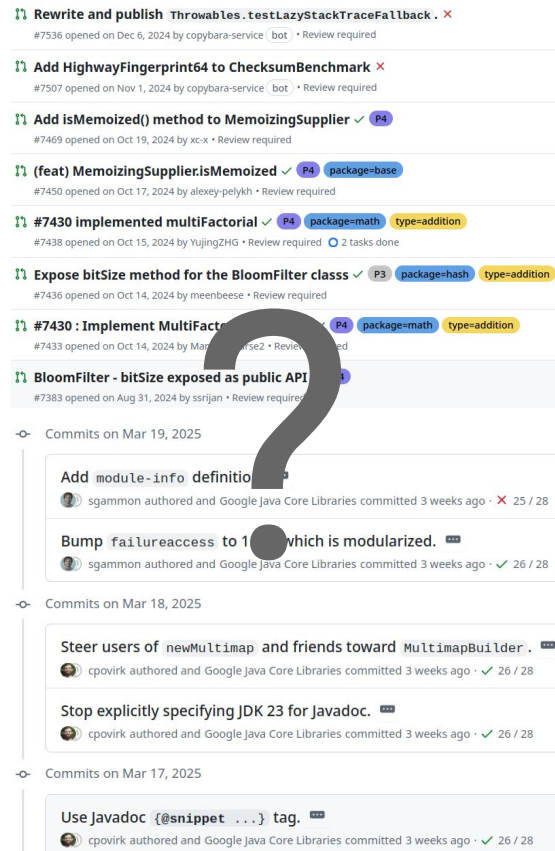
[1] "Historical and impact analysis of API breaking changes: A large-scale study". L. Xavier; A. Brito; A. Hora, M. Valente. SANER'17

[2] "Breaking Bad? Semantic Versioning and Impact of Breaking Changes in Maven Central". L. Ochoa, T. Degueule, JR. Falleri, J. Vinju. EMSE'22

[3] "An extended study of syntactic breaking changes in the wild". D. Jayasuriya, S. Ou, S. Hegde, V. Terragni, J. Dietrich, K. Blincoe. EMSE'25

Breaking Changes in SE Research

- Studies: *why, how, when* do libraries break, how does it affect clients?
 - Xavier *et al.*: 317 libraries, 9k releases¹
 - Ochoa *et al.*: 14k libraries, 120k releases²
 - Jayasuriya *et al.*: 7k libraries, 18k releases³
- Researchers are limited in their ability to study library evolution
 - Japicmp is the *de facto* standard, but requires compiled JAR files
 - Only about 50% of arbitrary versions on GitHub can be built automatically⁴
 - Even when possible, compilation costs are impractical for large-scale studies
 - Empirical research is limited to analyzing *pre-built releases*
- Analyzing commits, pull requests, etc. would be valuable but is impractical⁵



[1] "Historical and impact analysis of API breaking changes: A large-scale study". L. Xavier; A. Brito; A. Hora, M. Valente. SANER'17
[2] "Breaking Bad? Semantic Versioning and Impact of Breaking Changes in Maven Central". L. Ochoa, T. Degueule, JR. Falleri, J. Vinju. EMSE'22
[3] "An extended study of syntactic breaking changes in the wild". D. Jayasuriya, S. Ou, S. Hegde, V. Terragni, J. Dietrich, K. Blincoe. EMSE'25
[4] "Automatic building of java projects in software repositories: A study on feasibility and challenges". F. Hassan, M. Shaikh, E. Lam, X. Wang. ESEM'17
[5] "BreakBot: Analyzing the Impact of Breaking Changes to Assist Library Evolution". L. Ochoa, T. Degueule, JR. Falleri. ICSE'22 (NIER)



```
V1 public final class List<E> {  
    private int size;  
    public E add(int pos, E elem) {  
        ...  
    }  
}
```

```
V2 public final class List<E T> {  
    private int double size listSize;  
    public E void add(int pos, E T elem) {  
        ...  
    }  
}
```



V1

```
public final class List<E> {  
    private int size;  
    public E add(int pos, E elem) {  
        ...  
    }  
}
```



Breaking Changes:

- METHOD_RETURN_TYPE_CHANGED

V2

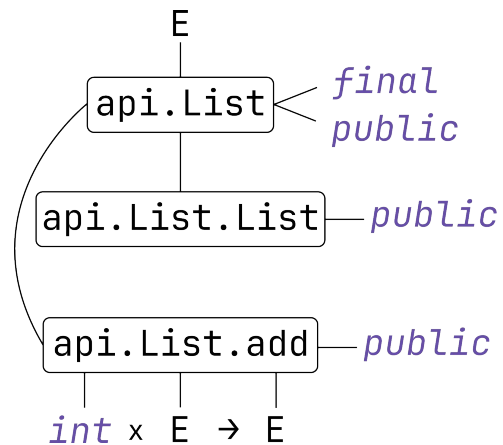
```
public final class List<E T> {  
    private int double size listSize;  
    public E void add(int pos, E T elem) {  
        ...  
    }  
}
```

ROSEAU – API Model

```
public final class List<E> {  
    private int size;  
    public E add(int pos, E elem) {  
        ...  
    }  
}
```

ROSEAU - API Model

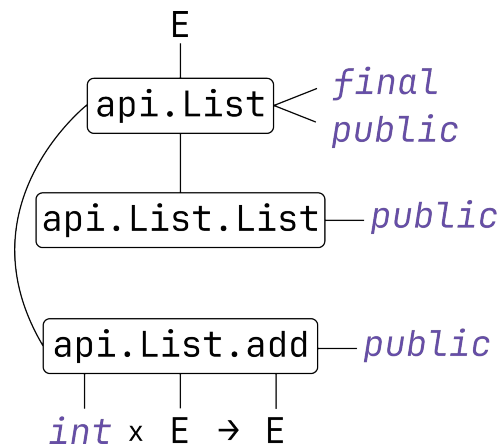
```
public final class List<E> {  
    private int size;  
    public E add(int pos, E elem) {  
        ...  
    }  
}
```



ROSEAU – API Model

```
public final class List<E> {  
    private int size;  
    public E add(int pos, E elem) {  
        ...  
    }  
}
```

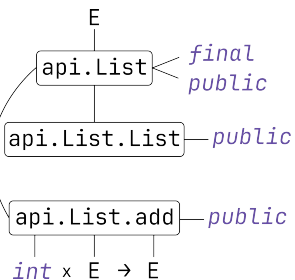
```
public final class  
api.List<E> {  
    public api.List();  
        0: aload_0  
        1: invokespecial #1  
        4: return  
  
    public E add(int, E);  
        0: aconst_null  
        1: areturn  
}
```



```
public final class List<E> {
    private int size;
    public E add(int pos, E elem) {
        ...
    }
}
```

```
public final class api.List<E> {
    public api.List();
        0: aload_0
        1: invokespecial #1
        2: return

    public E add(int, E);
        0: aconst_null
        1: areturn
}
```

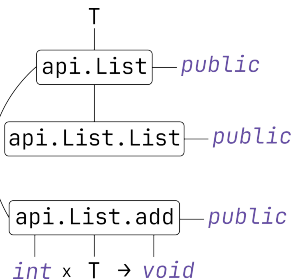


```
public final class List<E T> {
    private int double size listSize;
    public E void add(int pos, E T elem) {
        ...
    }
}
```

```
public class api.List<T> {
    public api.List();
        0: aload_0
        1: invokespecial #1
        4: return

    public void add(int, T);
        0: aconst_null
}

```



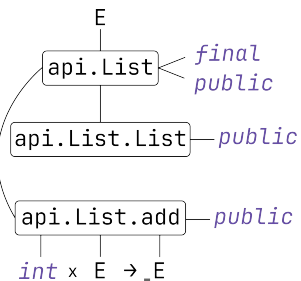


V1

```
public final class List<E> {
    private int size;
    public E add(int pos, E elem) {
        ...
    }
}
```

```
public final class api.List<E> {
    public api.List();
        0: aload_0
        1: invokespecial #1
        4: return

    public E add(int, E);
        0: aconst_null
        1: areturn
}
```

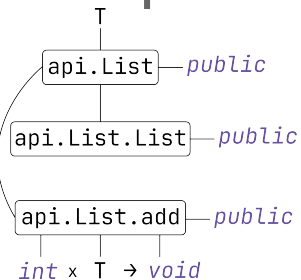


V2

```
public final class List<E T> {
    private int double size listSize;
    public E void add(int pos, E T elem) {
        ...
    }
}
```

```
public class api.List<T> {
    public api.List();
        0: aload_0
        1: invokespecial #1
        4: return

    public void add(int, T);
        0: aconst_null
}
```



Diff

ROSEAU – BCs as Diffs of API Models

- Translating JLS rules as detection rules on API models
- Expressing these rules at the API level is much easier than at the AST/classfile level

Kind	Detection Rule	Binary	Source
Type removed	$T \in A \wedge \nexists T' \in A' \mid T \sim T'$	✗	✗
Type now protected	$T \sim T' \wedge \text{public}(T) \wedge \text{protected}(T')$	✗	✗
Type kind changed	$T \sim T' \wedge \text{typeKind}(T) \neq \text{typeKind}(T')$	✗	✗
Supertype removed	$T \sim T' \wedge \exists S \in \text{superTypes}(T) \mid \text{exported}(S) \wedge T' \not\sqsubseteq S$	✗	✗
Type variable removed	$T \sim T' \wedge \text{typeVars}(T) > \text{typeVars}(T') $	✓	✗
Type variable added	$T \sim T' \wedge \text{typeVars}(T') > \text{typeVars}(T) > 0$	✓	✗
Type variable changed	$T \sim T' \wedge \exists i \in [1, \text{typeVars}(T)], \exists b' \in \text{bounds}(\text{typeVar}_i(T')), \forall b \in \text{bounds}(\text{typeVar}_i(T)) \mid b \not\sqsubseteq b'$	✓	✗
Type now final	$T \sim T' \wedge \neg \text{effectivelyFinal}(T) \wedge \text{effectivelyFinal}(T')$	✗	✗
Nested type now static	$T \sim T' \wedge \neg \text{static}(T) \wedge \text{static}(T')$	✗	✗
Nested type no longer static	$T \sim T' \wedge \text{static}(T) \wedge \neg \text{static}(T')$	✗	✗
Class now checked exception	$T \sim T' \wedge \text{uncheckedException}(T) \wedge \text{checkedException}(T')$	✓	✗
Abstract method added to type	$T \sim T' \wedge \exists m' \in \text{allMethods}(T') \mid \text{abstract}(m') \wedge \nexists m \in \text{allMethods}(T) \mid m \sim m'$	✓	✗
Executable removed	$e \in \text{executables}(T) \wedge \nexists e' \in \text{executables}(T') \mid e \sim e'$	✗	✗
Executable now protected	$e \sim e' \wedge \text{public}(e) \wedge \text{protected}(e')$	✗	✗
Executable checked exception removed	$e \sim e' \wedge \exists t \in \text{thrown}(e) \mid \nexists t' \in \text{thrown}(e'), t' \sqsubseteq t$	✓	✗
Executable checked exception added	$e \sim e' \wedge \exists t' \in \text{thrown}(e') \mid \nexists t \in \text{thrown}(e), t' \sqsubseteq t$	✓	✗
Executable parameter generics changed	$e \sim e' \wedge \exists i \in [1, \text{params}(e)], \text{typeArgs}(\text{param}_i(e)) \neq \text{typeArgs}(\text{param}_i(e')) $	✓	✗
Executable type variable removed	$e \sim e' \wedge \text{typeVars}(e) > \text{typeVars}(e') \wedge (\text{isMethod}(e) \vee \text{typeVars}(e) > 1)$	✓	✗
Executable type variable added	$e \sim e' \wedge \text{typeVars}(e') > \text{typeVars}(e) > 0$	✓	✗
Executable type variable changed	$e \sim e' \wedge \exists i \in [1, \text{typeVars}(e)], \exists b' \in \text{bounds}(\text{typeVar}_i(e')), \forall b \in \text{bounds}(\text{typeVar}_i(e)) \mid b \not\sqsubseteq b'$	✓	✗
Method now final	$m \sim m' \wedge \neg \text{effectivelyFinal}(m) \wedge \text{effectivelyFinal}(m')$	✗	✗
Method now static	$m \sim m' \wedge \neg \text{static}(m) \wedge \text{static}(m')$	✗	✓
Method no longer static	$m \sim m' \wedge \text{static}(m) \wedge \neg \text{static}(m')$	✗	✗
Method now abstract	$m \sim m' \wedge \neg \text{abstract}(m) \wedge \text{abstract}(m')$	✗	✗
Method return type changed	$m \sim m' \wedge \text{returnType}(m) \neq \text{returnType}(m')$	✗	✗
Field removed	$f \in \text{fields}(T) \wedge \nexists f' \in \text{fields}(T') \mid f \sim f'$	✗	✗
Field now protected	$f \sim f' \wedge \text{public}(f) \wedge \text{protected}(f')$	✗	✗
Field now final	$f \sim f' \wedge \neg \text{final}(f) \wedge \text{final}(f')$	✗	✗
Field now static	$f \sim f' \wedge \neg \text{static}(f) \wedge \text{static}(f')$	✗	✓
Field no longer static	$f \sim f' \wedge \text{static}(f) \wedge \neg \text{static}(f')$	✗	✗
Field type changed	$f \sim f' \wedge \text{type}(f) \neq \text{type}(f')$	✗	✗

CHAPTER 13

Binary Compatibility

13.4.6 Class Body and Member Declarations

No incompatibility with pre-existing binaries is caused by adding an instance (respectively `static`) member that has the same name and accessibility (for fields), or same name and accessibility and signature and return type (for methods), as an instance (respectively `static`) member of a superclass or subclass. No error occurs even if the set of classes being linked would encounter a compile-time error.

Deleting a class member or constructor that is not declared `private` may cause a linkage error if the member or constructor is used by a pre-existing binary.

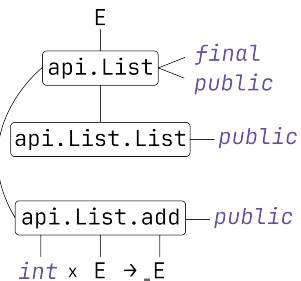


V1

```
public final class List<E> {
    private int size;
    public E add(int pos, E elem) {
        ...
    }
}
```

```
public final class api.List<E> {
    public api.List();
        0: aload_0
        1: invokespecial #1
        4: return

    public E add(int, E);
        0: aconst_null
        1: areturn
}
```



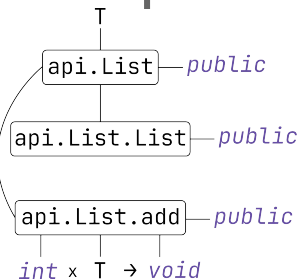
Diff

V2

```
public final class List<E T> {
    private int double size listSize;
    public E void add(int pos, E T elem) {
        ...
    }
}
```

```
public class api.List<T> {
    public api.List();
        0: aload_0
        1: invokespecial #1
        4: return

    public void add(int, T);
        0: aconst_null
}
```



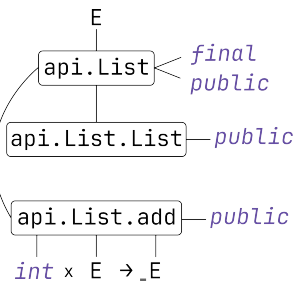


V1

```
public final class List<E> {
    private int size;
    public E add(int pos, E elem) {
        ...
    }
}
```

```
public final class api.List<E> {
    public api.List();
        0: aload_0
        1: invokespecial #1
        4: return

    public E add(int, E);
        0: aconst_null
        1: areturn
}
```



Diff

CHAPTER 13

Binary Compatibility

13.4.6. Class Body and Member Declarations

No incompatibility with pre-existing binaries is caused by adding an instance (respectively `static`) member that has the same name and accessibility (for fields), or same name and accessibility and signature and return type (for methods), as an instance (respectively `static`) member of a superclass or subclass. No error occurs even if the set of classes being linked would encounter a compile-time error.

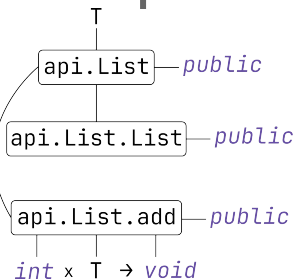
Deleting a class member or constructor that is not declared `private` may cause a linkage error if the member or constructor is used by a pre-existing binary.

V2

```
public final class List<E T> {
    private int double size listSize;
    public E void add(int pos, E T elem) {
        ...
    }
}
```

```
public class api.List<T> {
    public api.List();
        0: aload_0
        1: invokespecial #1
        4: return

    public void add(int, T);
        0: aconst_null
}
```



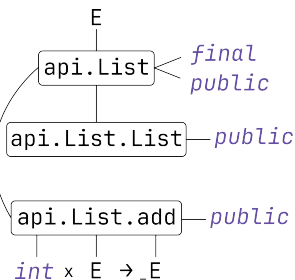


V1

```
public final class List<E> {
    private int size;
    public E add(int pos, E elem) {
        ...
    }
}
```

```
public final class api.List<E> {
    public api.List();
        0: aload_0
        1: invokespecial #1
        4: return

    public E add(int, E);
        0: aconst_null
        1: areturn
}
```



Diff

CHAPTER 13

Binary Compatibility

13.4.6. Class Body and Member Declarations

No incompatibility with pre-existing binaries is caused by adding an instance (respectively `static`) member that has the same name and accessibility (for fields), or same name and accessibility and signature and return type (for methods), as an instance (respectively `static`) member of a superclass or subclass. No error occurs even if the set of classes being linked would encounter a compile-time error.

Deleting a class member or constructor that is not declared `private` may cause a linkage error if the member or constructor is used by a pre-existing binary.

Breaking Changes:

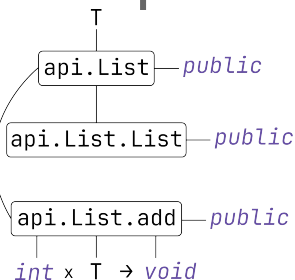
- METHOD_RETURN_TYPE_CHANGED

V2

```
public final class List<E T> {
    private int double size listSize;
    public E void add(int pos, E T elem) {
        ...
    }
}
```

```
public class api.List<T> {
    public api.List();
        0: aload_0
        1: invokespecial #1
        4: return

    public void add(int, T);
        0: aconst_null
}
```



Features / Optimizations

- Incremental Parsing

```

package com.google.common.base; import static com.google.common.base.Preconditions.checkNotNull; import static java.util.Objects.requireNonNull; import
com.google.common.annotations.GwtCompatible; import com.google.errorprone.annotations.CanIgnoreReturnValue; import java.io.IOException; import java.util.AbstractList; import
java.util.Arrays; import java.util.Iterator; import java.util.List; import java.util.Map; import java.util.Map.Entry; import java.util.concurrent.ConcurrentMap;
org.jspecify.annotations.Nullable; @GwtCompatible public class Joiner {    public static Joiner on(String separator) {    return new Joiner(separator);    }    public static
Joiner on(char separator) {    return new Joiner(String.valueOf(separator));    }    private final String separator;    private Joiner(String separator) {    this.separator =
checkNotNull(separator);    }    private Joiner(Joiner protot
Iterable<?> parts) throws IOException {    return appendTo(
throws IOException {    checkNotNull(appendable);    if (
appendable.append(separator);    appendable.append(toS
appendable, @Nullable Object[] parts)    throws IOExcept
appendTo(appendable, partsList);    }    public final <A
Object... rest)    throws IOException {    return append
Iterable<?> parts) {    return appendTo(builder, parts.ite
appendTo((Appendable) builder, parts);    }    catch (IOExcep
Appendable) appendable, @Nullable Object[] parts) {    return appendTo(
return appendTo(builder, partsList);    }    public final St
@Nullable Object... rest) {    return appendTo(builder, if
size == ((List<?>) parts).size());    if (size == 0) {
parts) {    if (i == toJoin.length) {    toJoin
toString(part);    }    if (i != toJoin.length)
String result = String.join(separator, toJoin);    return result;
appendTo(new StringBuilder(), parts).toString();    }    public final String join(@Nullable Object[] parts) {    @SuppressWarnings("nullness") List<?> partsList =
Arrays.<@Nullable Object>asList(parts);    return join(partsList);    }    public final String join(
@Nullable Object first, @Nullable Object second, @Nullable Object...
rest) {    return join(Iterable.of(first, second, rest));    }    public Joiner useNull(String nullText) {    checkNotNull(nullText);    return new Joiner(this) {
@Override    CharSequence toString(CharSequence part, @Nullable Object partNull) ? nullText : Joiner.this.toString(part);    }    @Override    public Joiner

```

Reference → [java.util.Iterator](#)

Reference → [myapi.Collection](#)

Reference → [java.lang.String](#)

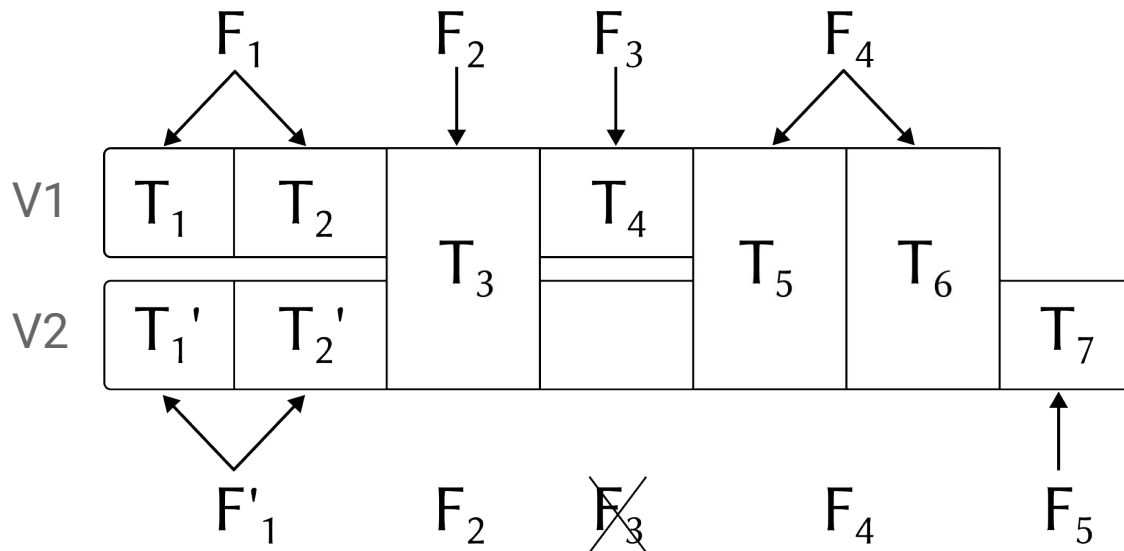
Features / Optimizations

- Incremental Parsing
- “Diet” Parsing

```
public class ArrayList {  
    public int indexOf(Object o) {  
        E[] a = this.a;  
        if (o == null) {  
            for (int i = 0; i < a.length; i++)  
                if (a[i] == null)  
                    return i;  
        } else {  
            for (int i = 0; i < a.length; i++)  
                if (o.equals(a[i]))  
                    return i;  
        }  
        return -1;  
    }  
}
```

Features / Optimizations

- Incremental Parsing
- “Diet” Parsing
- Partial Updates



Evaluation

A large, solid dark blue shape that starts from the bottom left corner and extends diagonally upwards towards the right, covering the bottom half of the slide.

Accuracy

API Evolution Data Corpus (*Jezek et al.*¹ and *Ochoa et al.*²)

[1] K. Jezek and J. Dietrich, "Api evolution and compatibility: A data corpus and tool evaluation." *J. Object Technol.*, vol. 16, no. 4, pp. 2–1, 2017.

[2] L. Ochoa, T. Degueule, J.-R. Falleri, and J. Vinju, "Breaking bad? semantic versioning and impact of breaking changes in maven central: An external and differentiated replication study," *Empirical Software Engineering*, vol. 27, no. 3, p. 61, 2022.

Accuracy

API Evolution Data Corpus (Jezek et al.¹ and Ochoa et al.²)

	All			Source			Binary		
	Prec.	Rec.	F ₁	Prec.	Rec.	F ₁	Prec.	Rec.	F ₁
JAPICMP	0.90	0.82	0.86	0.78	0.80	0.79	0.90	0.98	0.94
REVAPI	0.85	0.97	0.91	0.77	0.96	0.85	0.89	0.95	0.92
ROSEAU	0.98	0.99	0.99	0.88	1.00	0.94	0.91	0.98	0.94

[1] K. Jezek and J. Dietrich, "Api evolution and compatibility: A data corpus and tool evaluation." *J. Object Technol.*, vol. 16, no. 4, pp. 2–1, 2017.

[2] L. Ochoa, T. Degueule, J.-R. Falleri, and J. Vinju, "Breaking bad? semantic versioning and impact of breaking changes in maven central: An external and differentiated replication study," *Empirical Software Engineering*, vol. 27, no. 3, p. 61, 2022.

Performance

Dataset w/ Top 100 Artifacts from Maven

 Artifact last stable version on itself

Java Microbenchmark Harness (JMH)

→ **50** measures by artifact and tool

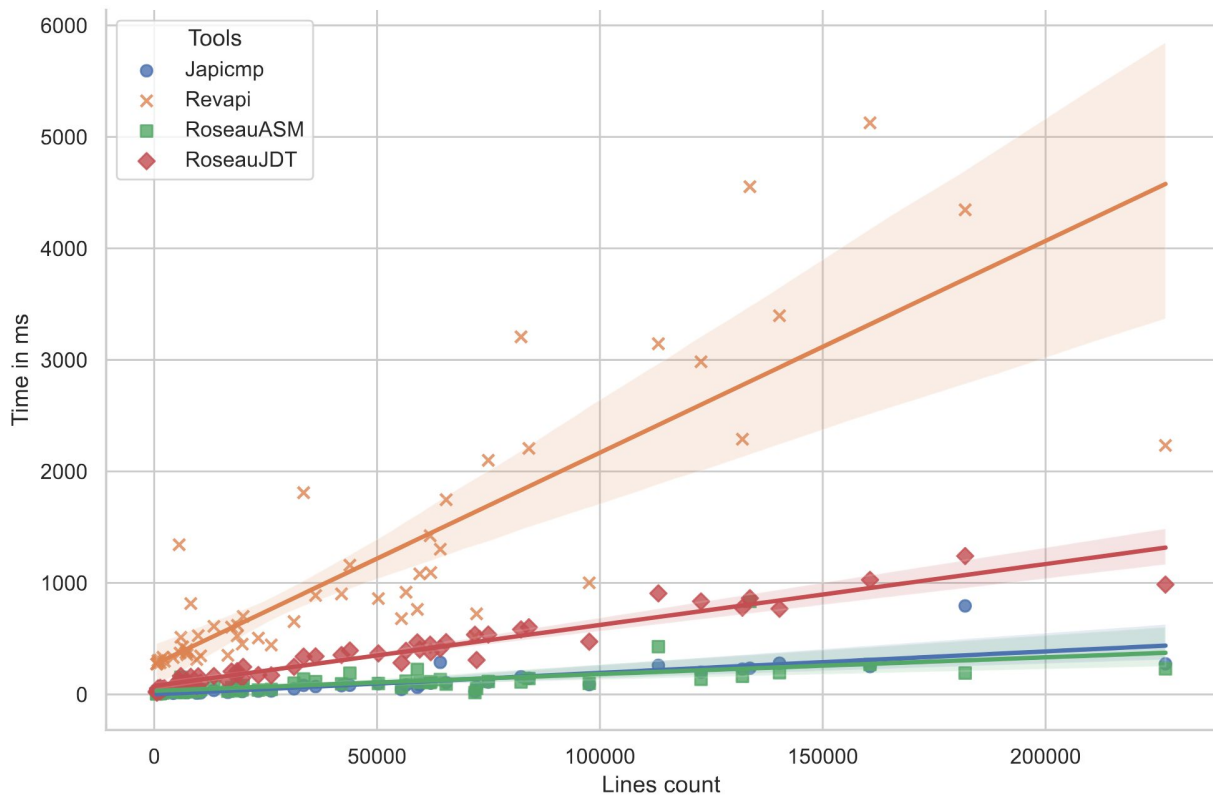
Performance

Dataset w/ Top 100 Artifacts from Maven

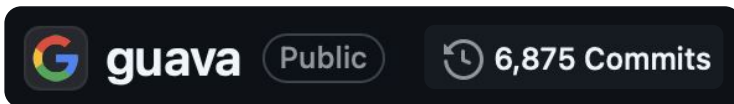
🕒 Artifact last stable version on itself

Java Microbenchmark Harness (JMH)

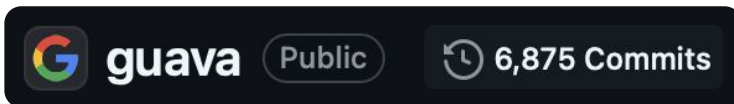
→ **50** measures by artifact and tool



Large-scale Longitudinal Studies



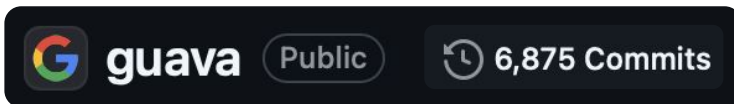
Large-scale Longitudinal Studies



Preliminary study: 100 latest commits of Google's Guava

Method	Mean commit time (ms)	Total time (minutes)
Packaging JAR	68,750	7,836
ROSEAU w/o incremental	1,065	121
ROSEAU w/ incremental	81	9

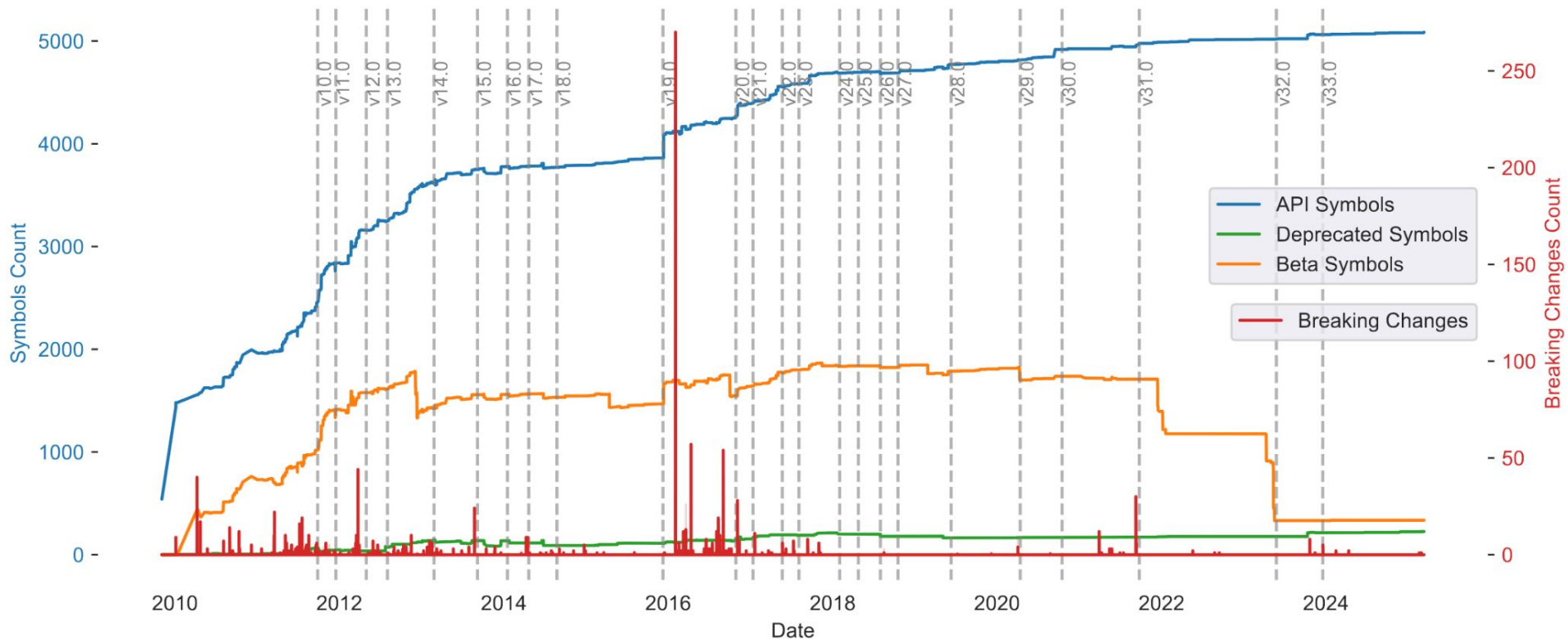
Large-scale Longitudinal Studies



Preliminary study: 100 latest commits of Google's Guava

Method	Mean commit time (ms)	Total time (minutes)
Packaging JAR	68,750	7,836
ROSEAU w/o incremental	1,065	121
ROSEAU w/ incremental	81	9

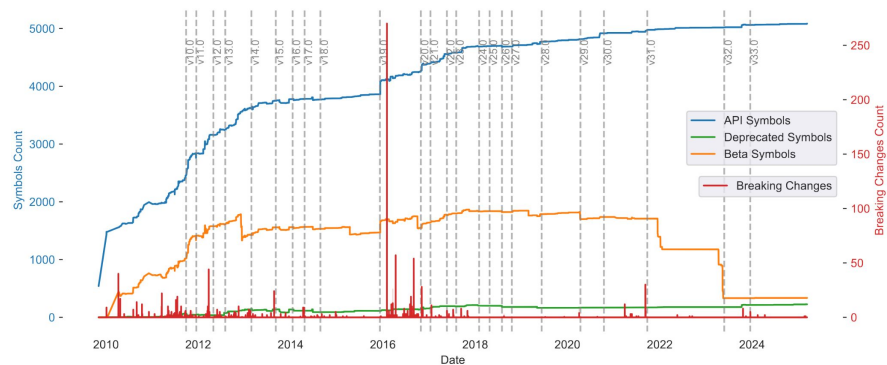
5,5 days vs 9 minutes



Evolution of Guava API and introduction of Breaking Changes over time

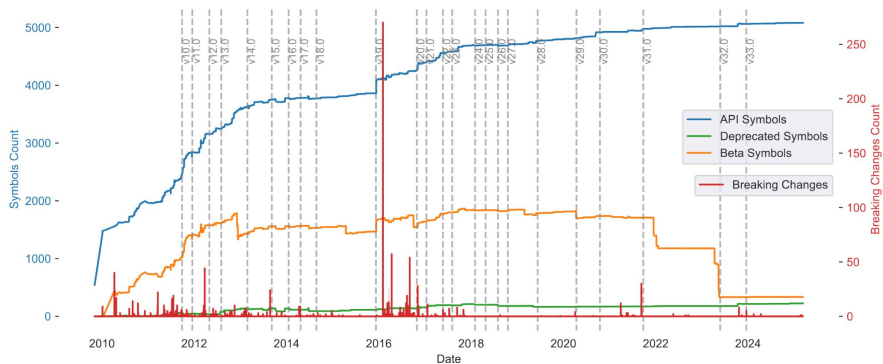
Future Work

Qualitative case studies of some popular libraries



Future Work

Qualitative case studies of some popular libraries



Maintain and improve Roseau



junit-team / junit-framework

Commit 465a06f



marcphilipp committed on Jul 11 Verified

Try out roseau to analyze API changes between versions

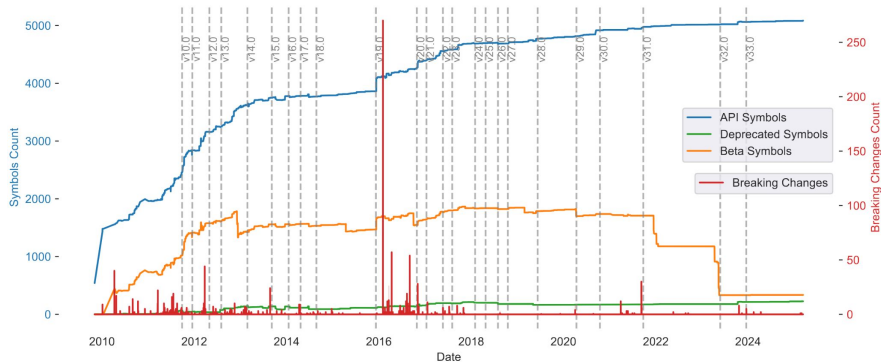


[marc/roseau](#)

1 parent [eaf5ae9](#) commit 465a06f

Future Work

Qualitative case studies of some popular libraries



Combinatorial benchmark

V1 API :

288 types

47K members



1,5M V2

~80K clients

Maintain and improve Roseau



junit-team / junit-framework

Commit 465a06f



marcphilipp committed on Jul 11

Verified

Try out roseau to analyze API changes between versions

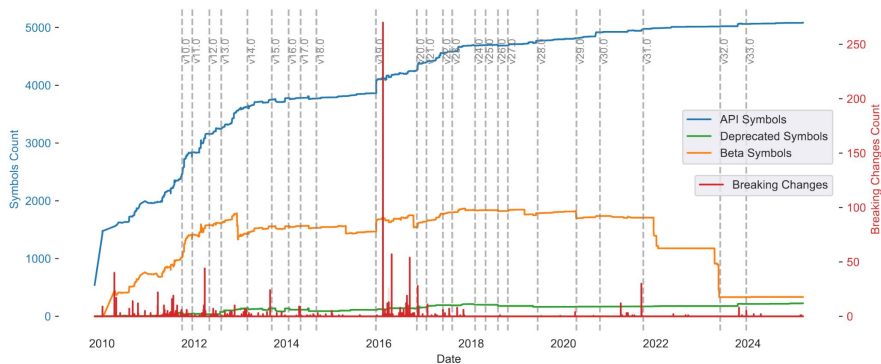


marc/roseau

1 parent eaf5ae9 commit 465a06f

Future Work

Qualitative case studies of some popular libraries



Combinatorial benchmark

V1 API :

288 types
47K members

~80K clients



1,5M V2

Maintain and improve Roseau



junit-team / junit-framework

Commit 465a06f



marcphilipp committed on Jul 11

Verified

Try out roseau to analyze API changes between versions



marc/roseau

1 parent eaf5ae9 commit 465a06f

Live feedback in IDEs

```
boolean isEffectivelyFinal() { 6 clients 1 broken Revert
    return isFinal()
        || isNative()
        || containingType.isEffectivelyFinal();
}
```

ROSEAU (REED)

Accuracy

API Evolution Data Corpus (Jezek et al.¹ and Ochoa et al.²)

	All			Source
	Prec.	Rec.	F ₁	
JAPICMP	0.90	0.82	0.86	0.78
REVAPI	0.85	0.97	0.91	0.77
ROSEAU	0.98	0.99	0.99	0.88

Large-scale Longitudinal Studies

 **guava** Public  **6,875 Commits**

Preliminary study: 100 latest commits of Google's Guava

Method	Mean commit time (ms)	Total time (minutes)
Packaging JAR	68,750	7,836
ROSEAU w/o incremental	1,065	121
ROSEAU w/ incremental	81	9

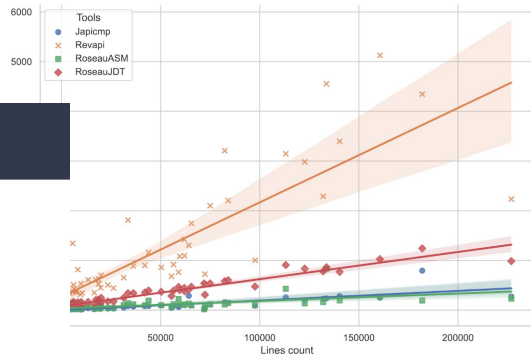
5,5 days vs **9 minutes**

Performance

Dataset w/ Top 100 Artifacts from Maven

 Artifact last stable version on itself

Java Microbenchmark Harness (JMH)



[1] K. Jezek and J. Dietrich, "Api evolution and compatibility: A data corpus and tool evaluation," *J. Object Technol.*, vol. 16, no. 4, pp. 3-1, 2017.
[2] L. Ochoa, T. Depaepe, J.-R. Faller, and J. Vinju, "Breaking bad? semantic versioning and impact of breaking changes in maven central," *A*

ROSEAU (REED)

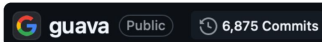


Accuracy

API Evolution Data Corpus (Jezek et al.¹ and Ochoa et al.²)

	All			Source
	Prec.	Rec.	F ₁	
JAPICMP	0.90	0.82	0.86	0.78
REVAPI	0.85	0.97	0.91	0.77
ROSEAU	0.98	0.99	0.99	0.88

Large-scale Longitudinal Studies



Preliminary study: 100 latest commits of Google's Guava

Method	Mean commit time (ms)	Total time (minutes)
Packaging JAR	68,750	7,836
ROSEAU w/o incremental	1,065	121
ROSEAU w/ incremental	81	9

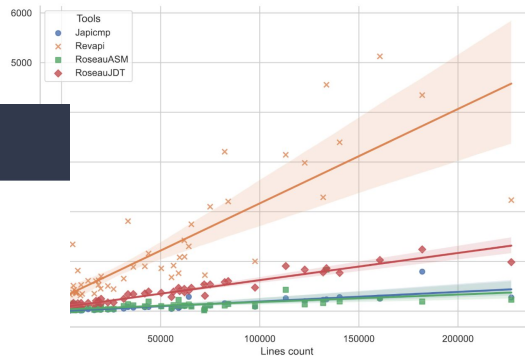
5,5 days vs 9 minutes

Performance

Dataset w/ Top 100 Artifacts from Maven

Artifact last stable version on itself

Java Microbenchmark Harness (JMH)



[1] K. Jezek and J. Dietrich, "Api evolution and compatibility: A data corpus and tool evaluation," *J. Object Technol.*, vol. 16, no. 4, pp. 3-1, 2017.
[2] L. Ochoa, T. Depaepe, J.-R. Faller, and J. Vrijs, "Breaking bad? semantic versioning and impact of breaking changes in maven central: A

"I bend and do not break."

The Oak and the Reed
by Jean de La Fontaine

