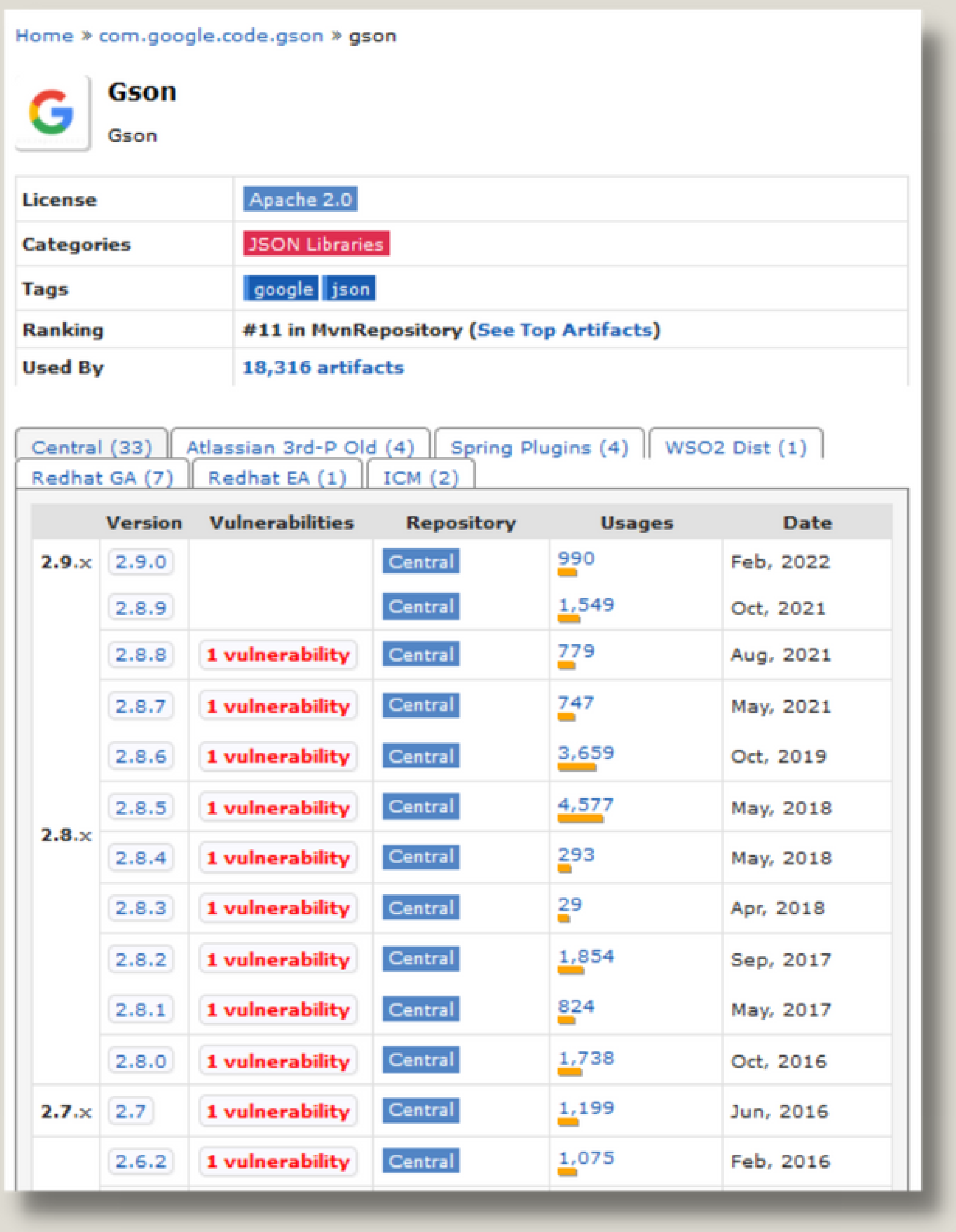
SEMANTIC VERSIONING ON MAVEN CENTRAL

Analyzing compatibility across versions and whether developers adhere to semantic versioning: can you trust a library to remain compatible within the same major version?

AUTHOR
Simcha Vos
a.s.j.vos@student.tudelft.nl

AFFILIATION
TU Delft - Faculty of Electrical Engineering, Mathematics and Computer Science

RESPONSIBLE STAFF
Supervisor: Mehdi Keshani
Responsible professor: Sebastian Proksch


Home » com.google.code.gson » gson

Gson
Gson

License	Apache 2.0
Categories	JSON Libraries
Tags	google json
Ranking	#11 in MvnRepository (See Top Artifacts)
Used By	18,316 artifacts

Central (33) | Atlassian 3rd-P Old (4) | Spring Plugins (4) | WSO2 Dist (1) | Redhat GA (7) | Redhat EA (1) | ICM (2)

	Version	Vulnerabilities	Repository	Usages	Date
2.9.x	2.9.0		Central	990	Feb, 2022
	2.8.9		Central	1,549	Oct, 2021
	2.8.8	1 vulnerability	Central	779	Aug, 2021
	2.8.7	1 vulnerability	Central	747	May, 2021
	2.8.6	1 vulnerability	Central	3,659	Oct, 2019
	2.8.5	1 vulnerability	Central	4,577	May, 2018
2.8.x	2.8.4	1 vulnerability	Central	293	May, 2018
	2.8.3	1 vulnerability	Central	29	Apr, 2018
	2.8.2	1 vulnerability	Central	1,854	Sep, 2017
	2.8.1	1 vulnerability	Central	824	May, 2017
	2.8.0	1 vulnerability	Central	1,738	Oct, 2016
2.7.x	2.7	1 vulnerability	Central	1,199	Jun, 2016
	2.6.2	1 vulnerability	Central	1,075	Feb, 2016

INTRODUCTION 1

Maven Central is the most popular repository that distributes JVM-based artifacts. It accumulates all versions of published libraries, which allows us to analyze the dependencies between those libraries.

We are interested in the evolution of APIs over time. We investigate compatibility through versions and whether developers adhere to semantic versioning.

SEMANTIC VERSIONING 2

version 1.0.3

Major version upgrades can contain a complete change to the API;

Minor version upgrades cannot remove methods of the API, so-called breaking changes;

Patch version upgrades cannot add any new methods to the API.

We analyze these two!

RESEARCH QUESTIONS 3

RQ1: How often do breaking changes occur during upgrades of minor and patch versions, and to what extent do they happen?

RQ2: How often do extensions of API occur during patch version upgrades, and to what extent do they happen?

RQ3: Does popularity of a method influence the presence of breaking changes?

METHODOLOGY 4

RQ1: Analyze the method signatures of two versions and check whether the newer version does not have one of the signatures.

RQ2: Analyze the method signatures of two versions and check whether the older version does not have one of the signatures.

RQ3: Find method signatures with and without breaking changes and find the respective popularity of these methods.

RESULTS 5

28% of artifacts feature at least one of the violations.

- A lot of artifacts do not fully adhere to semantic versioning.
  - While 76% of artifacts do not feature breaking changes and 76% do not feature illegal API extensions, the remaining artifacts feature on average up to 10% semantic versioning violations, compared to the total number of methods;
- Illegal API extensions occur as frequently as breaking changes (see Fig. 1);
- We can not conclude that popularity is related to whether a method will be involved in a breaking change or not (see Fig. 2).

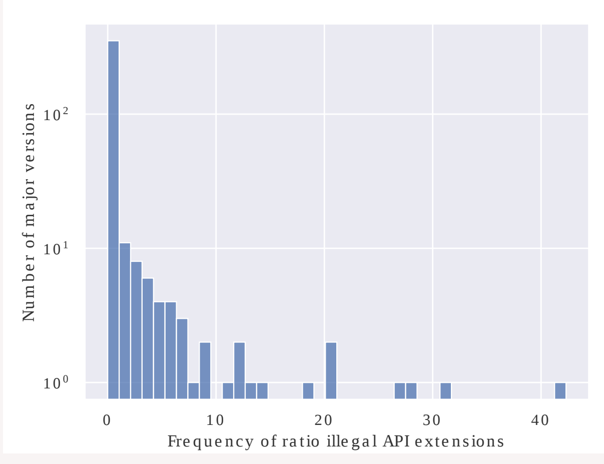


Fig. 1: Frequency of ratio of illegal API extensions over respective number of methods

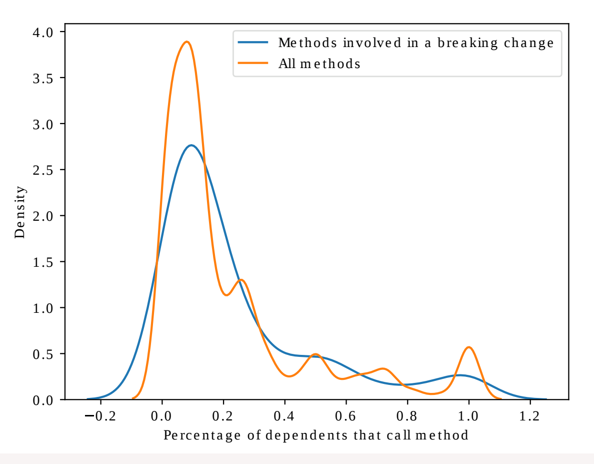


Fig. 2: Kernel Density Estimates of methods involved in a breaking change and all methods

DISCUSSION 6

Violating semantic versioning is dangerous, as developers rely on the guarantee that upgrading minor or patch versions do not break compatibility.

Maintainers introduce breaking changes regardless of the popularity of these methods. There might be some factors at play, but their effect is apparently small.

This study serves as an encouragement and reminder to uphold the principles of semantic versioning.

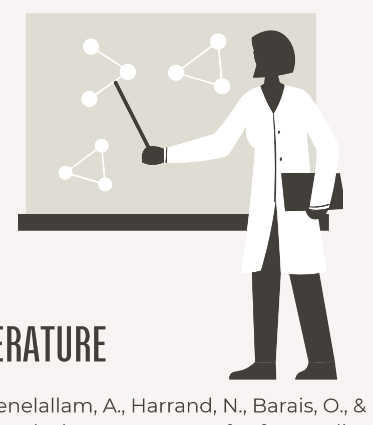
There is a need for more tools - client-side or platform-side. IDEs could offer warnings when changes to the set of method signatures are detected. Platforms like Maven Central could verify whether the versioning is appropriate or inform developers of the popularity of parts of their API.

We should aim for a safer and more trustworthy environment within the ecosystem of artifacts, where semantic versioning can guarantee compatibility.

CONCLUSION 7

As 28% of artifacts feature breaking changes or API extensions, too many developers violate semantic versioning.

As the chance of being affected by breaking changes is not large (as not all methods will have breaking changes), the impact is not too large within the ecosystem. But, as breaking changes will have a huge impact in case the dependent makes use of the method, it is clear that more focus should be put at adhering to semantic versioning.



RELATED LITERATURE

Soto-Valero, C., Benelallam, A., Harrand, N., Barais, O., & Baudry, B. (2019, May). The emergence of software diversity in maven central. In 2019 IEEE/ACM 16th International Conference on Mining Software Repositories (MSR) (pp. 333-343). IEEE.