Maven API study
Looking at changes in popularity in the Maven ecosystem

1 Background

- Developers tend to reuse open source code. Using dependency built tools developers can more easily reuse open source code.
- One of the popular build tools is Maven, Maven imposes an immutability policy
- As an artifact is immutable, it is interesting to research because it cannot change.
- We look into the Maven ecosystem using the FASTEN project, Which is a collaborative project with the intention to enhance robustness and security in software ecosystems.
- Studies have pointed out that only a small amount of APIs are being used. We hope to find trends in popularity. That can help library maintainers focus on a smaller set of APIs.

2 Research Questions

RQ1: How does popularity of packages change?
RQ2: How does the popularity of methods change?

3 Data Selection

- The sample we decided upon is a sample size of 384 unique packages as this will result into a confidence level of 95% and a maximum margin of error of 5%.
- The sampling technique used is weighted random sampling where the weight is the number of dependents.
- We derive all the versions of the 384 unique packages which are within our time frame
- Our time frame being 6 months starting in October.
- For method analysis we randomly sample 60 packages

4 Methodology

For package analysis:

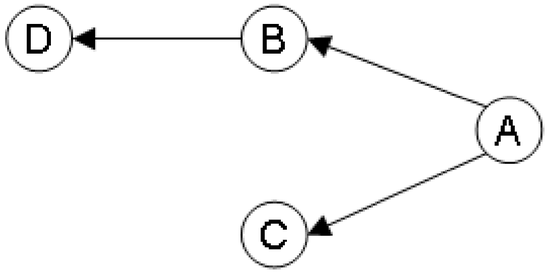
- We define popularity amongst packages by the number of dependents an artifact has.
- To compare versions we use a relative dependent metric
- We want to see how the popularity changes over time and over versions
- We also want to do linear regression to see if there is any relation between time and sustaining popularity

For callgraph generation:

- We need to do call graph generation for each artifact that we want to analyze
- We do so by providing a list with all the dependents of an artifact
- To confirm an artifact is indeed an dependent of the target artifact, we need to resolve the dependencies of the dependents.
- We also filter edges to result into an acyclic graph

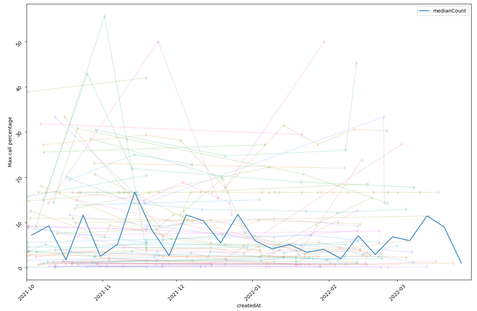
For Method analysis:

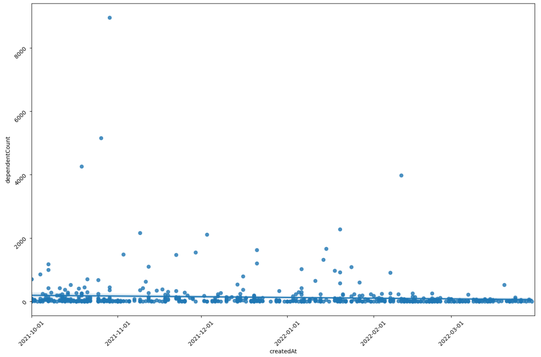
- We define popularity of methods by the number of unique dependents that call a method.
- To compare between different artifacts we decide to look at the relative calls a method receives
- To see trends in method popularity we take the max value of relative calls and then take the median between all the packages.
- We then test the data using the Mann Kendall Test to see if there are any trends over time



5 Observations

- Older versions sometimes get re-released
- Low usage in Alpha, Beta releases
- Popular packages seem to have less releases
- Tightly coupled packages
- Tightly coupled methods





6 Future Work

- Categorizing similar types of packages
- It can be interesting to look at larger timeframes.
- Look at look at popular packages right now and how they ascended to popularity.
- Why popular packages from a few years ago seized in being popular

7 Conclusion

- We looked at the package level and did not find a correlation between time and popularity. (p=0.18)
- We also did not see a trend over time at the method level. The concentration of methods used seems to remain the same, which on the median is 10%.
- As the popularity of methods does not change much over time, implies that developers do not have to check the popularity of method frequently. This research shows that it might take a long time for popularity to change, but more future research on a larger time frame is necessary to conclude this.

Author - Michel Bulten
Supervisor - Mehdi Keshani
Responsible Teacher - Sebastian Proksch
CSE3000