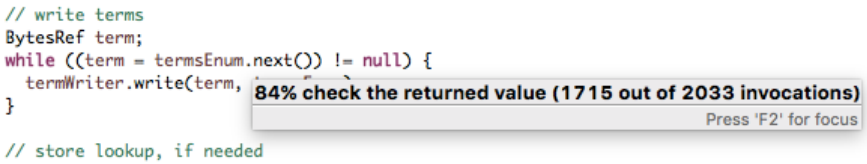

Figure 4.12: Tooltip showing that `termsEnum.next()` is checked for null in 84% of the dereferenced cases.

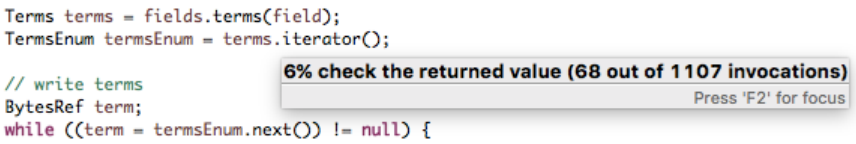

Figure 4.13: Tooltip showing that `terms.iterator()` is checked for null in 6% of the dereferenced cases.

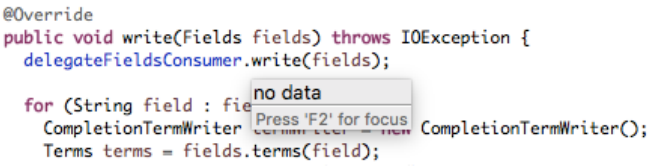

Figure 4.14: Tooltip showing that there is no data for `delegateFieldsConsumer.write(fields)`.

4.3 Summary

We automatically analyze over 45'638 artifacts clients of Apache APIs within two days. Client usage accounts for 7.7% of all detected invocations tracked to 10.3% of all invoked methods. Of 30'739'729 detected conditional expressions, 31.4% are null checks. We find 45.5% of null checks to be associated with the return value of an invoked method, identifying null returning methods as a major driver of cyclomatic complexity. In a manual inspection to verify the correctness of the analysis we find some samples of temporary live variables. The detected null checks in those cases are not installed to check if a method returned null, but to determine the state of an iterative procedure. We compute the nullability of the invoked methods and find that 65.0% of them are never checked for null, 33.5% are sometimes checked and 1.5% are always checked. There are many methods that are rarely checked, and we find many superfluous null checks among them. In the JRE API documentation, method nullability is incompletely documented, although it is an import part of a method's contract. To fill this gap, we create an IDE plugin that shows the nullability of a hovered method.