

# Augmenting Stack Overflow with API Usage Patterns Mined from GitHub

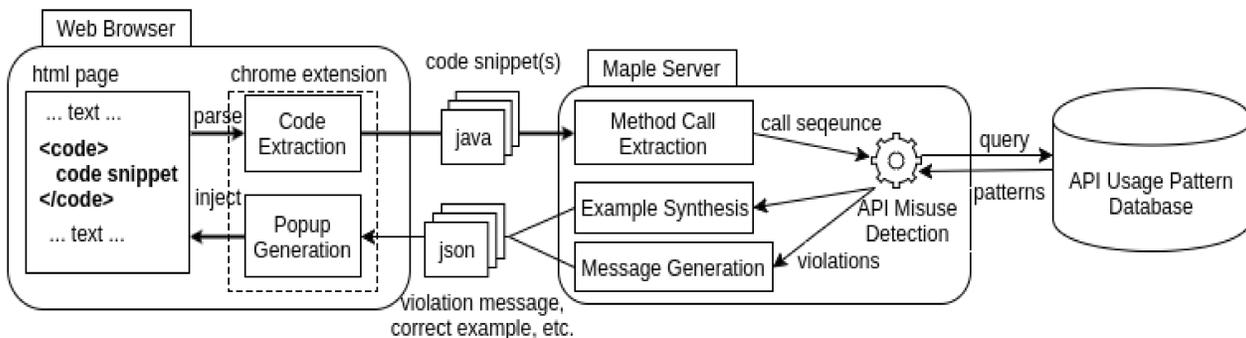
Anastasia Reinhardt<sup>1</sup> Tianyi Zhang<sup>2</sup> Miryung Kim<sup>2</sup>

<sup>1</sup> George Fox University, research done through CRA-W DREU    <sup>2</sup>University of California, Los Angeles

## Problem Statement

- Programmers often search for online code examples to learn new APIs, and Stack Overflow (SO) is a popular Q&A website that programmers consult.
- However, SO examples are not always complete or reliable, which can be misleading and potentially dangerous when programmers follow the same example in a client program.
- Our previous study shows that over 31% of 217,818 SO posts contain API misuse that could produce symptoms of program crashes and resource leaks if reused in a target system.

## Solution: Soap (Stack Overflow Augmentation using mined Patterns)



- Commonly practiced idioms in massive code corpora may represent a desirable API usage pattern that a programmer can use to trust and enhance code examples on Stack Overflow.
- Soap is an interactive approach that augments Stack Overflow with code idioms learned from GitHub and alerts programmers about the potential API usage violations in a code example.
- Soap leverages a scalable API usage mining technique to learn three types of API-related idioms---temporal ordering, guard conditions, and exception handling of API calls---from over 380K GitHub projects.
- To help developers build confidence on a code idiom, Soap shows how many GitHub developers also follow the idiom as well as how many other users like or dislike this idiom.

## Soap Chrome Extension Features

- A user can open a pop-up window on a highlighted API call, which contains a potential API misuse detected by the chrome extension.
- The pop-up window includes a descriptive API usage violation message and a synthesized code example that demonstrates the correct API usage pattern.
- The user can click on the GitHub links to explore concrete, supporting examples in an open-source project that follow the correct API usage pattern.
- The user can also upvote or downvote this pattern based on its helpfulness or applicability.

## Example

This Stack Overflow answer does not use the `JsonElement.getAsString()` method correctly, so `getAsString` is highlighted:

```
import com.google.gson.JsonElement;
import com.google.gson.JsonObject;
import com.google.gson.JsonPrimitive;

public class GsonTest {

    public static void main(String[] args) {
        JsonElement jsonElement = new JsonPrimitive("foo");

        System.out.println(jsonElement.toString());
        System.out.println(jsonElement.getAsString());
    }
}
```

(1) Pop-up window

Potential API Misuse

You may want to check whether a condition is true before calling `getAsString()`. 113 GitHub code examples also do this. Make sure the `JsonElement` is not null to avoid `NullPointerException`.

This pop-up window appears when the user clicks on the highlighted text.

(2) Descriptive API-usage message

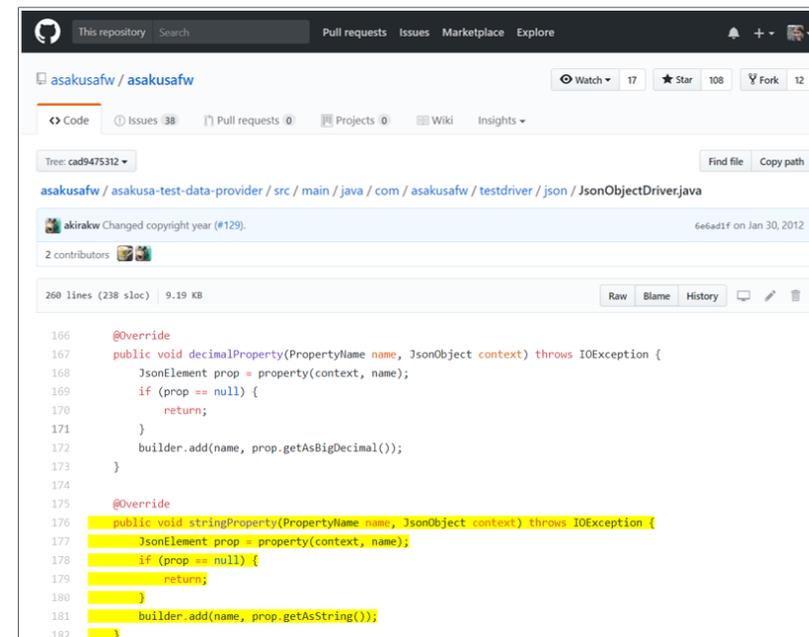
(3) Correct code example

See this in a GitHub example:

Example 1  
Example 2  
Example 3

(4) Concrete GitHub examples

(5) Pagination for multiple violations



By clicking on a GitHub link in the pop-up window, the user is redirected to a highlighted concrete example that follows the same API usage pattern.