

AI Found a Bug in My Code

I was playing around with the [SalesForce CodeGen](#) language model. It is set up to generate new code from a prompt. But I wanted to see how it would analyze existing code.

I had the model look at some existing code and rank the probability of each token appearing given the previous tokens. I also had it suggest its own token and compared the probability of my token to the probability of the model's token.

I found the smallest file in my codebase and fed it in. Brightness represents unlikelihood of each token. How red it is is how much more likely the model's token would be. Hover over the tokens to see what the model would have suggested.

```
export class JoelEvent<T = void> {
  private listeners = new Set<(arg: T) => void>();
  constructor(
    public current: T
  ) {}
  dispatch(arg: T) {
    this.current = arg;
    for (const listener of [...this.listeners]) {
      if (!this.listeners.has(listener))
        return;
      listener(arg);
    }
  }
  on(listener: (arg: T) => void) {
    this.listeners.add(listener);
  }
}
```

```

off(listener: (arg: T) => void) {
  this.listeners.delete(listener);
}
once(): Promise<T> {
  return new Promise<T>(x => {
    const cb = (arg: T) => {
      this.listeners.delete(cb);
      x(arg);
    };
    this.listeners.add(cb);
  });
}
}
}

```

- **Green** — The AI is confused but doesn't have a good alternative suggestion.
- **Red** — The AI has a suggestion that it likes better.
- **Gray** — No notes. The AI could have written this itself.

I tried to make the code more readable for the AI by adding some comments.

```

dispatch(arg: T) {
  this.current = arg;
  // Iterate over a copy of listeners in case one is added during dispatch
  for (const listener of [...this.listeners]) {
    // Check that the listener is still there in case one is removed during dispatch
    if (!this.listeners.has(listener))
      return;
    listener(arg);
  }
}
}

```

While the comments themselves are a surprise for the AI, the redness around them is diminished. Except for the `return` statement which gets brighter red.

I did not plan this, but it turns out there is a bug in my code. When an event listener is removed during dispatch, I return from the function. Hovering over the suspicious code, the model correctly suggests `continue`.

I am going to use this for code review from now on.

