👩‍💻

Take Home Exercise - Event Emitter

- **Duration:** 30 - 45 mins
- Feel free to implement in JavaScript or TypeScript
- Add your solution to a private Github repo and send your username to your recruiter.

Here is a simple event emitter implementation.

```
const handlers = {}

const trigger = (type) => {
  handlers[type]()
}

const on = (type, handler) => {
  handlers[type] = handler
}

// Usage

on('foo', () => {
  console.log('Do foo');
});

trigger('foo');
// log: Do foo
```

1. Check if the current implementation has any issues

Validate that both `trigger` and `on` functions don't have any bugs.

2. How do we let you add more than 1 handler for the same type?

With the current implementation if I call `on` with the same type twice, the second handler will override the first one. So that when I `trigger` the event, only the second handler will be called.

eg.

```
on('foo', () => {
  console.log('Do foo');
});
on('foo', () => {
  console.log('Do some other foo');
});

trigger('foo');
// log: Do some other foo
```

How would I update my EventEmitter code above so that I can have multiple event handlers for the same event type and both handlers are executed?

3. How do we add an off method?

Usually with an event emitter you want to be able to remove a handler, this is usually in the form of an `off` or `removeListener` method. How would we add an `off` method to our EventEmitter that takes a type and a handler as arguments and lets you remove that specific handler. The method signature will look like:

```
const off = (type, handler) => {
 // ...
}
```

Usage would be:

```
const onFoo = () => {
  console.log('Do foo');
}
on('foo', onFoo);
off('foo', onFoo);
```

```
trigger('foo');
// No output
```

4. How do I allow you to pass arbitrary arguments?

Usually with an event emitter you want to be able to pass some data with your events. How would we extend our event emitter to allow you to pass **any number** of arguments along with your event? For example:

```
on('foo', (arg1, arg2) => {
  console.log('here are my args', arg1, arg2);
})
trigger('foo', arg1, arg2)
```