# Part 10 - First workshop feedback

Dimitri Merejkowsky

EFREI 2022-2023

## About code feedback

I re-use the code *you* wrote to teach you things, but don't worry - I keep them anonymous :)

## Show me the source!

Every team should send me *two* links by mail by the end of the day

Don't worry if you did not finish, you're here to learn!

## General feedback

- Make sure to keep test code in `src/test`
- Delete the `//TODO` comments when you're done!

## General feedback

- One language per team, please
- Re-use set ups when you can

## Git

- Don't mix bumping the java version and refactoring a class
- Add a .gitignore right away
- Don't put `target/` under version control

## Assertions (1)

Don't use `equal(true)`

```
let outcome = //  ... // ;

// Instead of:
t.equal(outcome, true);
// Use:
t.ok(outcome);
```

## Assertions (2)

Don't put useless assertion messages

```
// No
t.equal(request.name, 'John', 'should be the name')
```

## Assertions (3)

Do put assertion messages when it clarifies the test

```
const numberOfCalls = 2;

// Instead of
t.equal(numberOfCalls, 2)
// Use
t.equal(
  numberOfCalls,
  2,
  '<function> should have been called twice'
)
```

## Asserting lists

Regroup assertions when checking a list:

```python
# Instead of
assert result[0] == 'alice'
assert result[1] == 'bob'
# use
assert result == ['alice', 'bob']
```

Failure message will print both lists

```
AssertionError:
  assert ['alice', 'alice'] == ['alice', 'bob']
  At index 1 diff: 'alice' != 'bob'
```

## Know your test framework

Cool use of `assertAll`

```
assertAll("retry requests",
  () -> firStep()
  () -> secondStep()
);
```

## Avoid assertions in test doubles

```
class HttpClient {
  constructor() {
    this.numberOfCalls = 0
  }

  post(url, request) {
    // Not the right place
    t.equal(
        request instanceof Request, true,
        'should be a request'
    )
    this.numberOfCalls++;
  }
}
```

## Building a log

```java
class FakeNotifier {
  private List<String> logs;

  public void notify(User user, String message) {
      logs.add(user.name() + " got " + message);
   }
}
```

Building a list of strings describing what happened can make the test failures really clear

And humans are good at comparing texts

## Vocabulary

Heard a lot of “stubs”, “mock”, and so on.

Let’s clarify !