

UI Testing using Mock Service Worker

Jeremy Liberman

@MrLeebo 🐦

@MrLeebo 🍄 🐱

Who Am I

- Senior Software Developer
- Worked at Zelis Network Analytics for 10 years
- Primarily front-end, working with TypeScript and React
- Two daughters, a dog, and three cats
- Like playing Tennis, breaking out of Escape Rooms, and Food Trucks
- I have a twin brother who looks just like me



What are we going to talk about

What is a Mock Service Worker?

MSW for unit testing

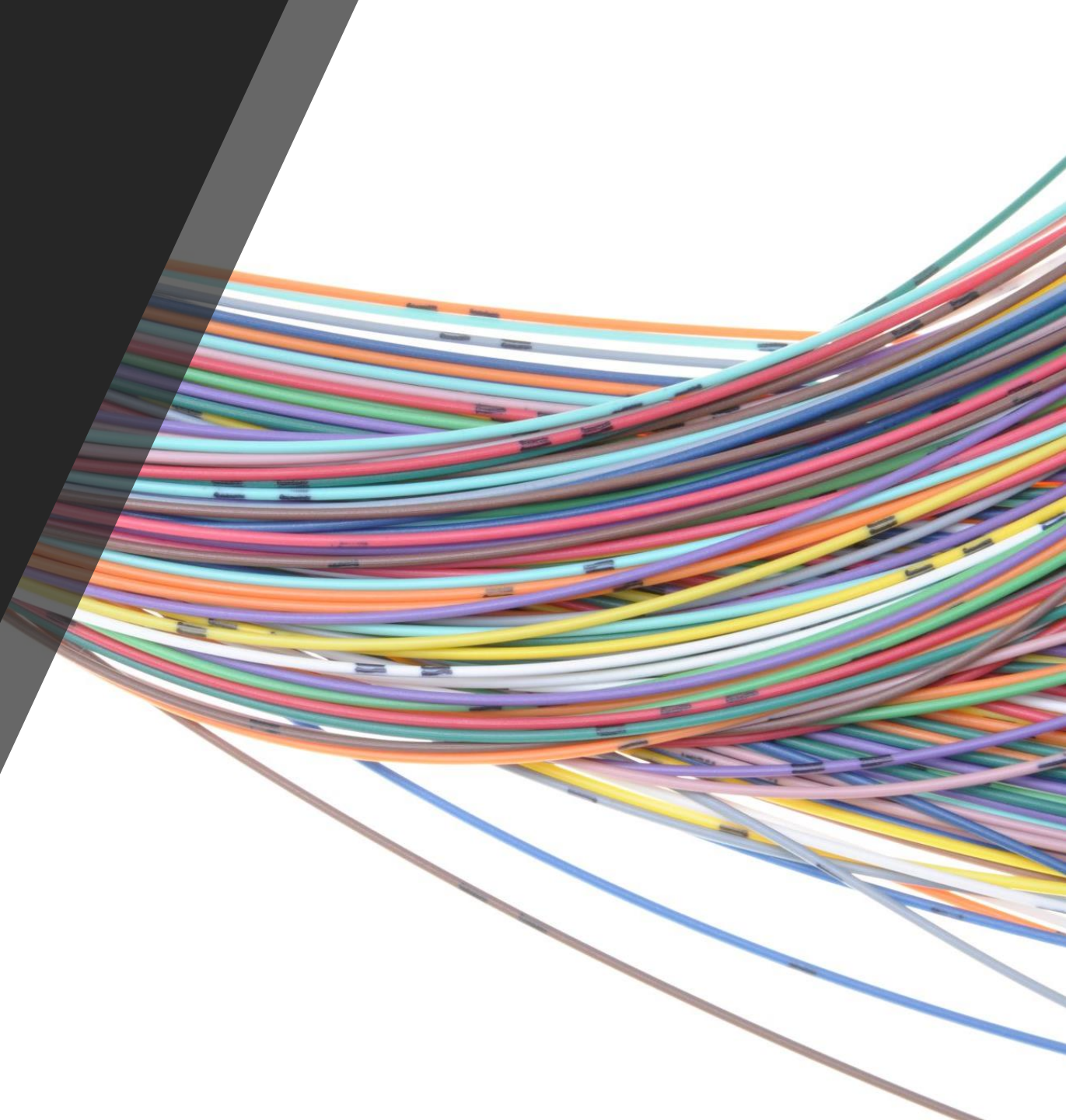
Go beyond "payload"

+=+=+=+=+=+=+=+ DEMO +=+=+=+=+=+=+=+

Review

What is a Mock Service Worker?

- Mock Service Worker (MSW) is a library that mocks HTTP requests made by your application.
- It intercepts outgoing network requests and returns mock responses instead of making real requests to a server.
- It intercepts at the networking layer, making it compatible with any framework, library, or tech stack.
- <https://mswjs.io/>



MSW for unit testing

- MSW has two forms of operation, Server for node, and Worker for browsers.
 - Use your testing frameworks before/after hooks to start/stop the listeners.
 - Set handlers anywhere; override them as you see fit.
-

```
// server.test.ts
import { rest } from 'msw';
import { setupServer } from 'msw/node';

const server = setupServer();

beforeAll(() => server.listen());
afterEach(() => server.resetHandlers());
afterAll(() => server.close());

test('fetches and displays a list of users', async () => {
  const payload = [{ id: 1, name: 'John' }];
  server.use(
    rest.get('https://api.com/users', (req, res, ctx) => {
      res(ctx.json(payload))
    })
  )

  const res = await fetch('https://api.com/users');
  const json = await res.json();

  expect(json).toEqual(payload);
})

// Photo by Constantin Panagopoulos on Unsplash
```

Go beyond "payload"

- Faster, more efficient testing
- Reliably mock response methods, status codes, and headers
- Simplify test setup with one mock that handles any and all HTTP clients
- Reusable mock handlers
- Test slow endpoints or race conditions by setting `ctx.delay()`
- Can produce images / binary content



DEMO



Review

- Mock in the browser too.
- Create handlers for APIs that haven't been built yet.
- URLSearchParams can be used to toggle mocks and are easy to control with Cypress.
- Mocked requests have a network latency of 0ms.
- Create "data profiles" to group mock handlers together.
- Use API contracts!



Thank You