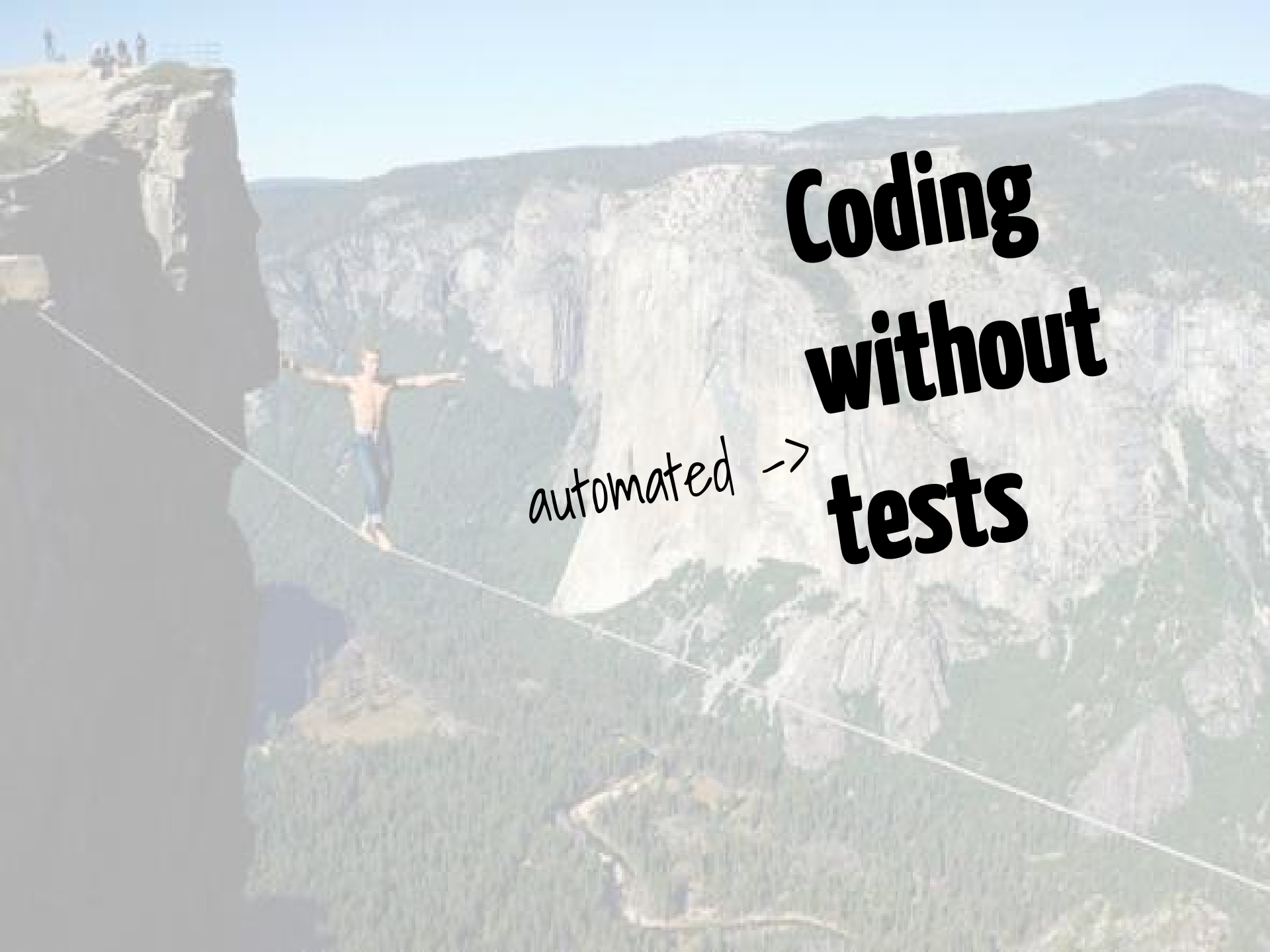
A person is walking a tightrope across a deep, rocky canyon. The person is shirtless and wearing blue pants, with arms outstretched for balance. The canyon walls are steep and rocky, with some greenery at the bottom. In the background, there are more hills and a clear blue sky. The text "Coding without tests" is overlaid on the right side of the image in a large, bold, black font.

**Coding
without
tests**



**Coding
without
tests**

automated ->

Press ? for help

.. (up a dir)

/home/calum/mindvalley/

► apposphere/

▼ daily_v2/

► app/

► bin/

► config/

► db/

► doc/

► lib/

► log/

► public/

▼ spec/

▼ acceptance/

businesses_spec.rb

organizations_spec.rb

► controllers/

► factories/

► jobs/

► models/

► queries/

spec_helper.rb

► vendor/

Capfile

config.ru

Gemfile

Gemfile.lock

Rakefile

README.md

► Ignite/

► locale-maketext-lexicon/

```
1 require 'spec_helper'
2 require 'rspec_api_documentation/dsl'
3
4 resource "Businesses" do
5   get "/api/v1/businesses" do
6
7     before :each do
8       (1..2).each { FactoryGirl.create(:business) }
9       (1..3).each { FactoryGirl.create(:bundle, {business: Business.all.first}) }
10    end
11
12    example "Listing businesses" do
13      do_request
14
15      expect(status).to eq(200)
16
17      body = JSON.parse(response_body)
18      expect(body['response'].length).to eq(2)
19    end
20  end
21
22  get "/api/v1/businesses/:id" do
23    let(:id) { FactoryGirl.create(:business).id }
24
25    example "Get a business" do
26      path.should == "/api/v1/businesses/#{id}"
27
28      do_request
29
30      expect(status).to eq(200)
31
32      body = JSON.parse(response_body)
33      expect(body['response']['id']).to eq(id)
34    end
35  end
36
37  context 'with some payments' do
38    let(:business_1) { FactoryGirl.create(:business) }
39    let(:business_2) { FactoryGirl.create(:business) }
40    let(:bundle) { FactoryGirl.create(:bundle, {business: business_1, created_at: 1.month.ago}) }
41    let(:customer) { FactoryGirl.create(:customer, {created_at: 1.month.ago}) }
42    let(:order) { FactoryGirl.create(:order, {customer: customer, business: business_2, state: 'complete', created_at: 2.day.ago}) }
43    let(:payment) { FactoryGirl.create(:payment,
```

```
self.assertEqual(meta['total_count'], 1)
66
67
68 def test_category_detail(self):
69     """
70     Retrieve a single category via the API.
71     """
72     client = Client()
73     response = client.get(
74         '/api/v1/category/1/',
75         ACCEPT='application/json',
76         HTTP_AUTHORIZATION=self.create_api_key_http_authorization(
77             self.john_doe.api_access))
78     self.assertEqual(response.status_code, 200)
79     item = json.loads(response.content)
80     self.check_category_item(item)
81
82 def test_category_product_listing(self):
83     """
84     Retrieve product listing of a category via the API.
85     """
86     client = Client()
87     response = client.get(
88         '/api/v1/category/1/product/',
89         ACCEPT='application/json',
90         HTTP_AUTHORIZATION=self.create_api_key_http_authorization(
91             self.john_doe.api_access))
92     self.assertEqual(response.status_code, 200)
93     resp_json = json.loads(response.content)
94     objects = resp_json['objects']
95     self.assertEqual(meta['total_count'], len(objects))
96     self.assertEqual(meta['total_count'], 1)
97
98     for item in objects:
99         check_product_detail(self, item)
100
101 def test_category_product_listing_with_nonselling(self):
102     """
103     Retrieve product listing of a category via the API, including the
104     non-selling items.
105     """
106     client = Client()
107     response = client.get(
108         '/api/v1/category/1/product/',
109         {'include_nonselling': 'true'},
110         ACCEPT='application/json',
111         HTTP_AUTHORIZATION=self.create_api_key_http_authorization(
112             self.john_doe.api_access))
```

What are automated tests anyway?

AINT NOBODY GOT TIME

FO DAT

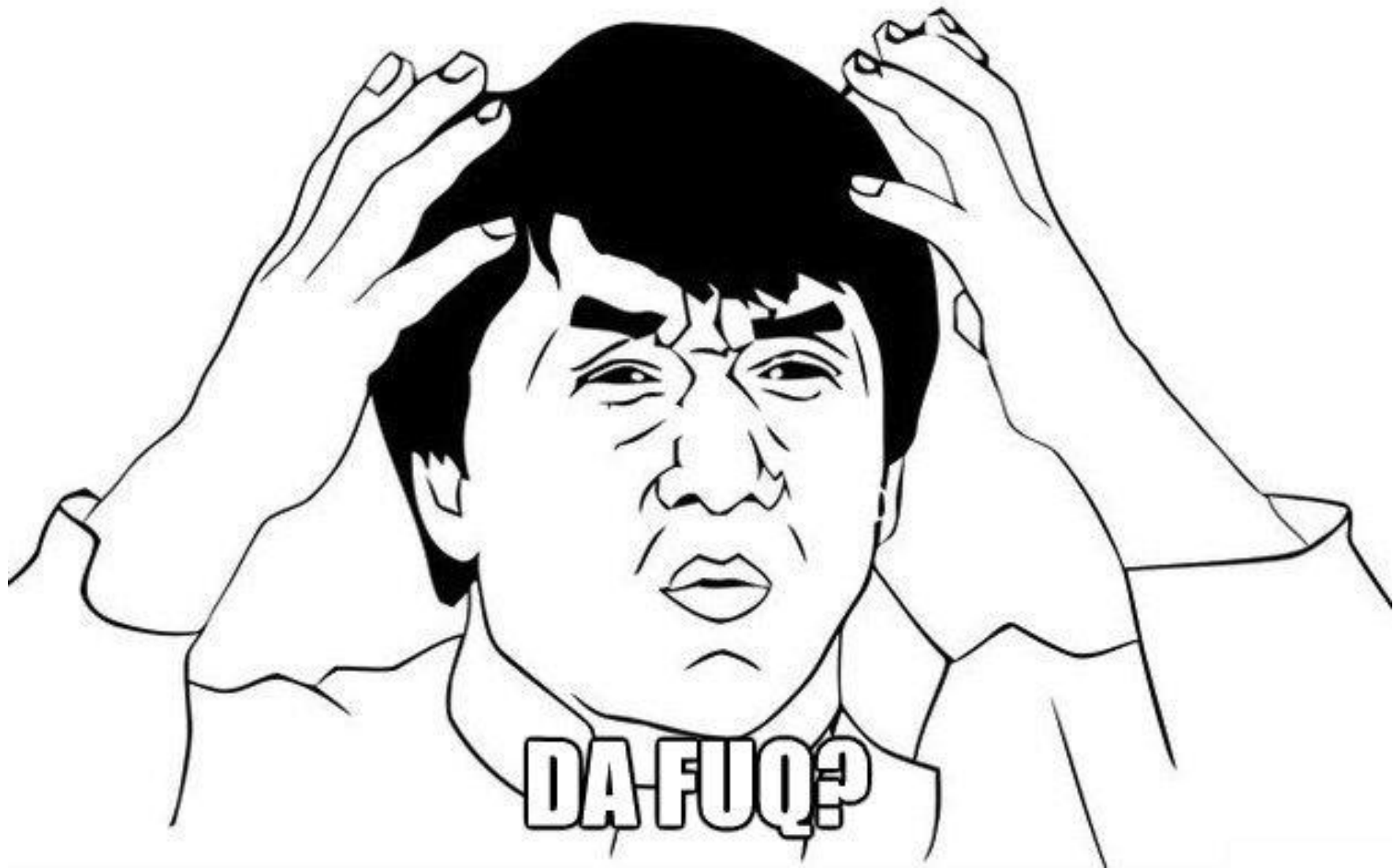
**faster
faster!**







Helpful
Tips

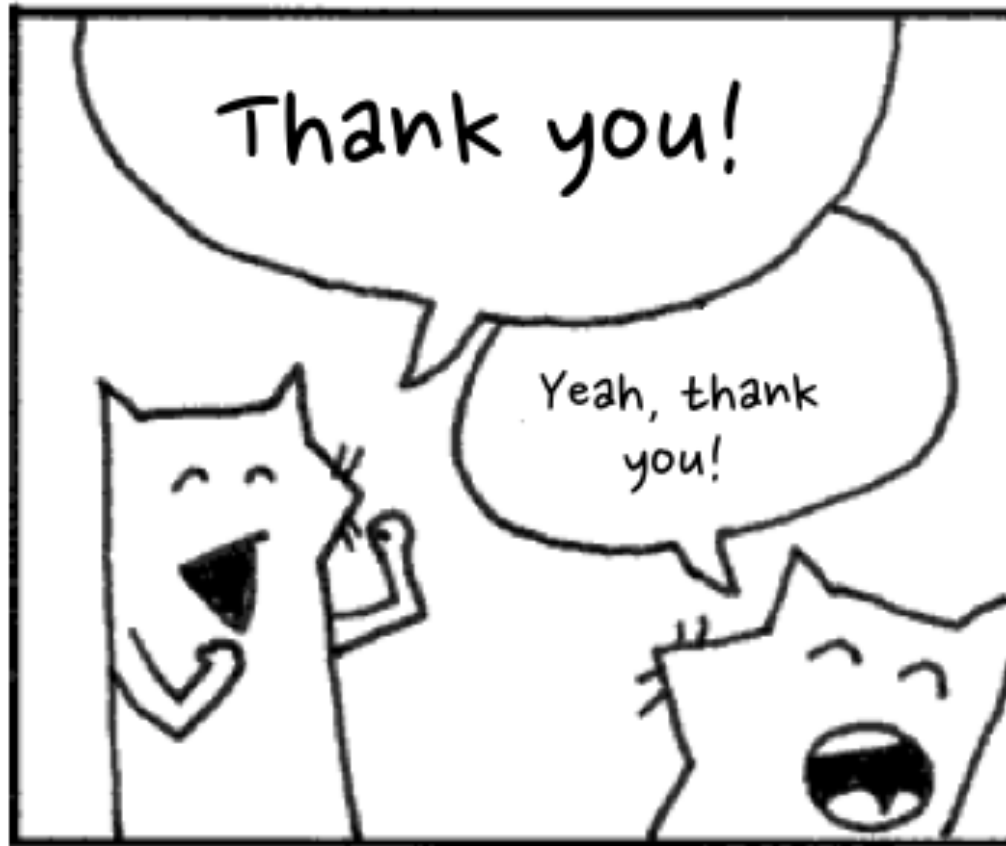


How do we fix this?



Learning is fun!





calumhalcrow