



FLOW

Production: Designing for Testability

Michael Bryzek

mike@flow.io / @mbryzek

Cofounder / CTO Flow

Cofounder / ex-CTO Gilt

QCon
NEW YORK

**Think about something you deployed
recently to production...**



Is It Working?



Right now in production?



How do you know?

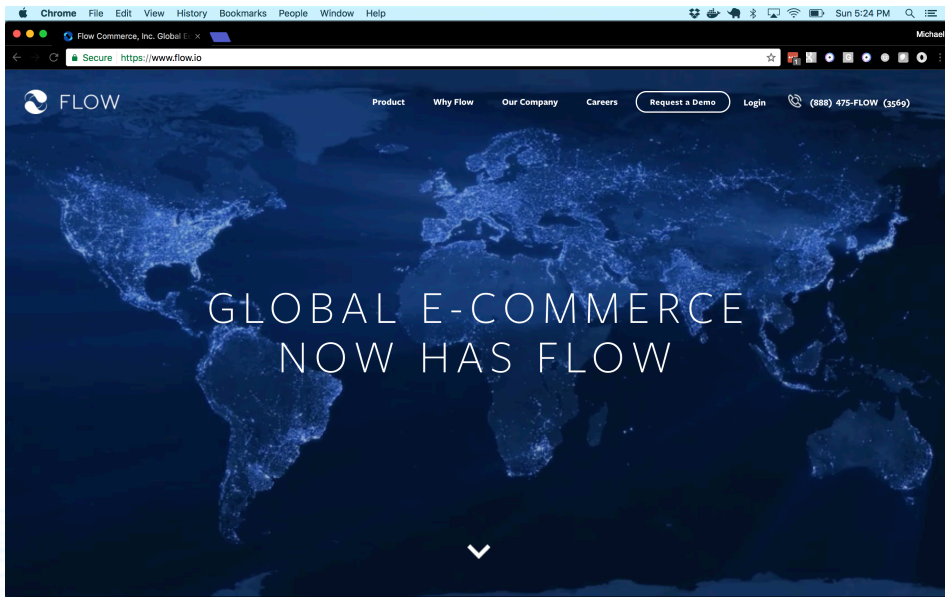


Feeling anxious?



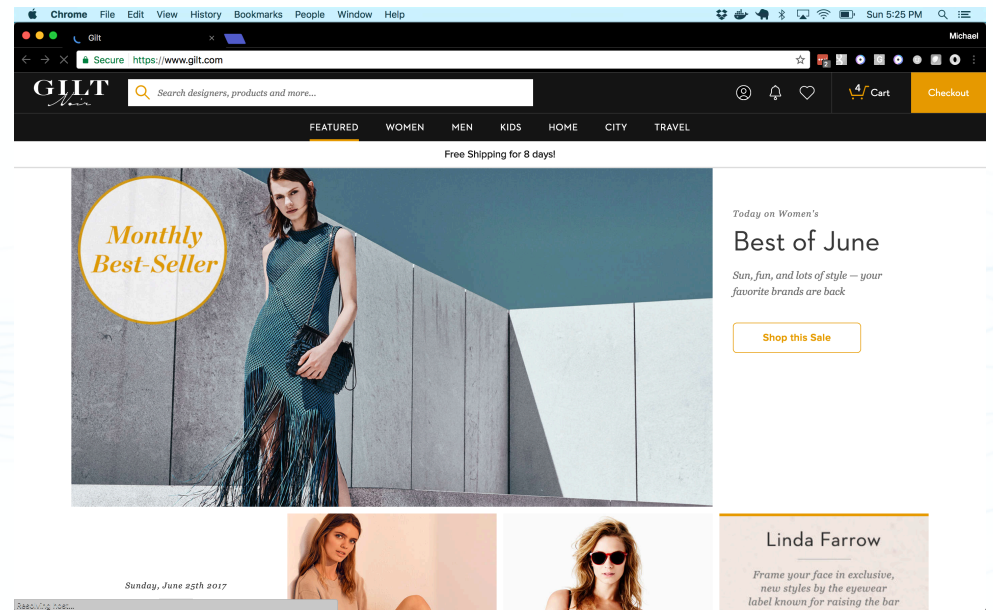
Let's remove that anxiety

About Me



www.flow.io

www.gilt.com






Software Quality is Hard

Think end to end for entire lifecycle of code

Verification in Production is a powerful technique
to help us build quality software

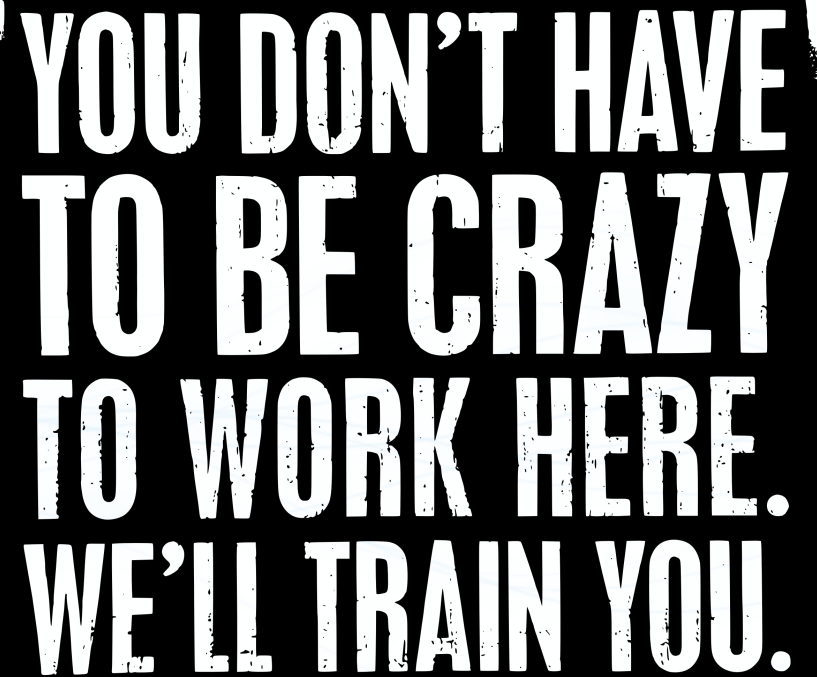


Life at Flow: Delivering Quality Software

True continuous delivery

No staging environments

Don't run code locally



YOU DON'T HAVE
TO BE CRAZY
TO WORK HERE.
WE'LL TRAIN YOU.

True Continuous Delivery

Automated tests / No safeguards

1 way to do something

Assume Continuous Delivery in Design Process

Desired State
0.3.15
2 instances
<i>Updated 2 hours ago</i>

Last State
0.3.15
2 instances
<i>Updated seconds ago</i>

“I love my staging environment”

Said Nobody Ever

A decorative graphic at the bottom of the slide consisting of several overlapping, wavy, light blue lines that create a sense of motion and depth.

■ No Staging Environments

Bottlenecks

Fragile

Difficult to understand failure

Expensive (30-40% of budget common)

Create the wrong incentives

■ Don't run code locally

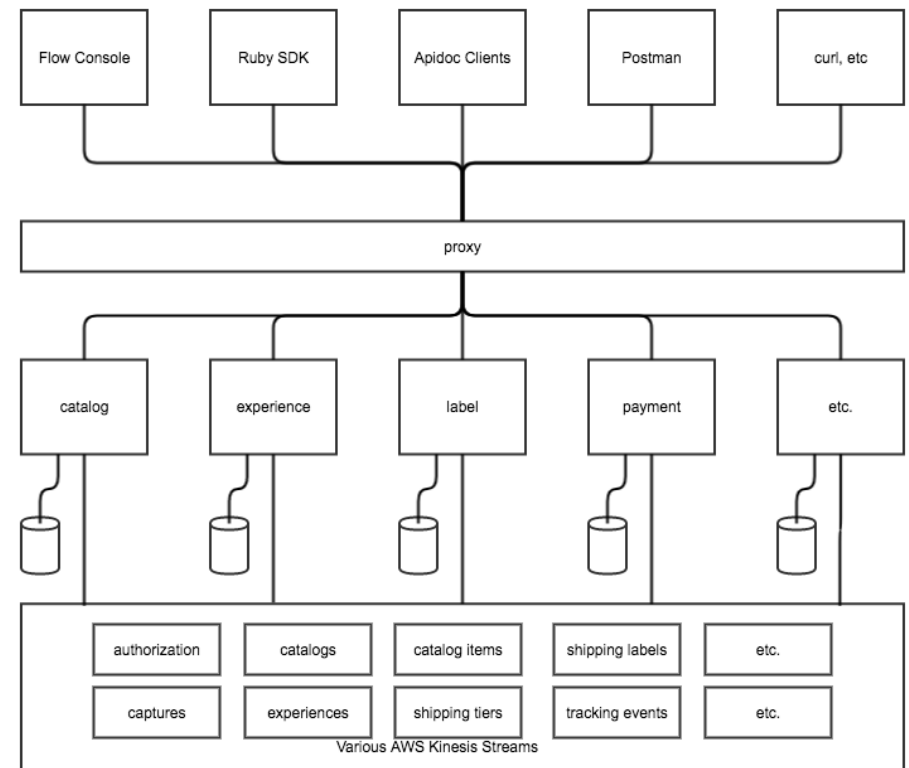
If unsure, write the test!

Learn to trust your tests

Quality Through Architecture

Extreme Isolation

- Rich event streams
- Own DNS, load balancer
- Private database
- No consul/zk/shared state
- Stop cascading failures
- “Delay” not “Outage”



Let's look at real examples

Successfully “testing in production”

Example: Know That Checkout Works

GILT
Hair

Checkout [Continue Shopping](#)

Ship To Michael Bryzek
[Redacted]
Glen Rock, NJ 07452-2209 [Change](#)
United States
9174941197

Payment Michael Bryzek
VISA Visa xxxx-[Redacted] [Change](#)
01/2019
United States

Submit Your Order

Subtotal	\$41.30
Shipping ?	\$0.00
Total:	\$41.30

You're saving: \$183.70
Insider Point Value: 206

Bot places an order every few minutes

Identify test orders and immediately cancel

Example: Support “Sandbox” Accounts

“SaaS” – even for internal accounts

Mark individual accounts as sandbox

One API Key for all sandbox accounts

“every service is a third party”

Example: End to End Integration Tests

Create Sandbox Org

Run tests

Delete Sandbox Org

“Safe and Repeatable”

Example: Using Sandbox Account for Test Orders

```
order = api_wrapper.create_order(experience.key, country, currency, items, first, last)
puts " - order: #{order.number}: #{order.total.label}"
auth = api_wrapper.create_auth(order, card.token)
puts " - auth: #{auth.id}"
capture = api_wrapper.create_capture(auth.id)
puts " - capture: #{capture.id}"
```

✓ master Michael Bryzek	CRON Randomize credit cards used in demo orders	👤 #34 passed e7ef4a9	🕒 6 min 13 sec 📅 5 days ago	⌵
✓ master Michael Bryzek	CRON Randomize credit cards used in demo orders	👤 #33 passed e7ef4a9	🕒 1 min 2 sec 📅 6 days ago	⌵
✓ master Michael Bryzek	CRON Randomize credit cards used in demo orders	👤 #32 passed e7ef4a9	🕒 2 min 3 sec 📅 7 days ago	⌵
✓ master Michael Bryzek	CRON Randomize credit cards used in demo orders	👤 #31 passed e7ef4a9	🕒 2 min 16 sec 📅 8 days ago	⌵
✓ master Michael Bryzek	CRON Randomize credit cards used in demo orders	👤 #30 passed e7ef4a9	🕒 5 min 13 sec 📅 9 days ago	⌵
✓ master Michael Bryzek	CRON Randomize credit cards used in demo orders	👤 #29 passed e7ef4a9	🕒 3 min 10 sec 📅 10 days ago	⌵

Example: Verifying Proxy Server Works as Expected

```
$ eval $(AWS_PROFILE=flowvault dev sbt --env production)
```

```
API Proxy Tests against http://localhost:7000  
All tests Passed
```

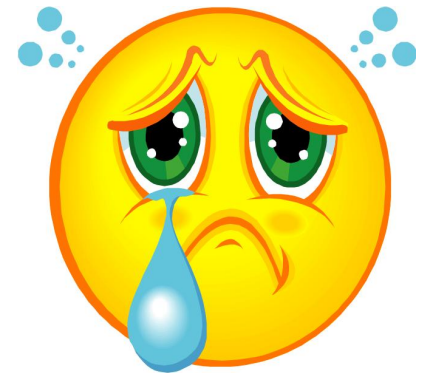
<https://github.com/flowvault/proxy>



Operating As Expected

But sometimes things go wrong

Even to the best of teams





Considerations

Make production access explicit (not the default)

Use defined paths (e.g. API calls)

Restrict sensitive data

Design for side effects



Unexpected Benefits



Perfect Documentation

The screenshot shows a web browser window with the URL `https://docs.flow.io/getting-started#how-to-authenticate`. The page title is "API Documentation - Flow Commerce". The left sidebar contains a navigation menu with sections: "GETTING STARTED" (with sub-items like "Register with Flow Commerce", "Obtain your API key", "How to authenticate", "Types", "Field Naming Conventions", "Operations and Responses", "Native Libraries", "JSONP", "Envelopes for request/response", "Geolocation proxy tools", "Getting Help"), "INTEGRATION OVERVIEW", "PARTNER API", "SHOPIFY", "SOLIDUS", "TESTING", "GEOLOCATION" (with sub-items "Addresses", "Country Defaults", "Timezones"), and "LOCALIZATION" (with sub-items "Allocations", "Attributes").

The main content area is titled "How to authenticate" and contains the following text:

All authentication to the Flow Commerce API is via HTTP Basic Authorization. Your API Key is your username, and there is no password.

Note: when using curl, if you append a ":" after your API Key, curl will not prompt you for your password.

For example:

```
curl -u <your api key>: https://api.flow.io/organizations
```

You should see a response similar to:

```
[
  {
    "id": "flow",
    "name": "Flow Commerce",
    "environment": "production",
    "discriminator": "organization"
  },
  {
    "id": "flow-sandbox",
    "name": "Flow Sandbox",
    "environment": "sandbox",
    "parent": {
      "id": "flow"
    },
    "discriminator": "organization"
  }
]
```

Note that you will also see which environment your organization belongs to (production or sandbox).

Types

The entire Flow API is strongly typed and friendly to use from the command line. We design our API by declaring types for every field and object with no exception, and then use this type information to validate inbound JSON.

Internally, we use [apidoc](#) to create our API - providing documentation and great native language client libraries.

Primitive Types:

Type	Description	Example
------	-------------	---------

Capture request/response of API Calls

```
curl -u <your api key>: https://api.flow.io/organizations
```

```
[
  {
    "id": "flow",
    "name": "Flow Commerce",
    "environment": "production",
    "discriminator": "organization"
  },
  {
    "id": "flow-sandbox",
    "name": "Flow Sanbdox",
    "environment": "sandbox",
    "parent": {
      "id": "flow"
    },
    "discriminator": "organization"
  }
]
```

```
items
items/get
items/get/scrapbook-demo
items/get/scrapbook-demo/catalog
items/get/scrapbook-demo/catalog/items
items/get/scrapbook-demo/catalog/items/sku-1
items/get/scrapbook-demo/catalog/items/sku-1/simple.response.json
items/get/scrapbook-demo/catalog/items/sku-2
items/get/scrapbook-demo/catalog/items/sku-2/simple.response.json
items/get/scrapbook-demo/catalog/targetings
items/get/scrapbook-demo/catalog/targetings/scrapbook-targetings-example
items/get/scrapbook-demo/catalog/targetings/scrapbook-targetings-example/simple.response.json
items/post
items/post/:organization
items/post/:organization/catalog
items/post/:organization/catalog/items
items/post/:organization/catalog/items/advanced.request.json
items/post/:organization/catalog/items/advanced.response.json
items/post/:organization/catalog/items/simple.request.json
items/post/:organization/catalog/items/simple.response.json
```

✓ master

 Michael Bryzek

CRON Wait 90 seconds for metric

🟢 #299 passed

 443707a

🕒 1 min 23 sec

📅 27 about a minute ago

Tooling: API | Builder
(formerly known as apidoc)

www.apibuilder.io

The screenshot shows the landing page for API Builder. At the top left, it says 'API | BUILDER'. In the top right, there is a 'Log In with Github' button and a 'Fork on Github' link. The main heading is 'Simple, Comprehensive Tooling for Modern APIs' with a 'Get Started' button below it. Below the heading are four feature cards: 'Amazing native libraries and code generators', 'Beautiful documentation for modern APIs', 'Complete API lifecycle management', and 'Open Source and Compatible'.

Version control for APIs
Backwards Compatibility
High Quality Mocks

High Quality Mocks – From Specs

```
@javax.inject.Singleton
class MockLocationClient @javax.inject.Inject() (
  override val addresses: MockAddresses
) extends io.flow.location.v0.mock.Client

@javax.inject.Singleton
class MockAddresses() extends io.flow.location.v0.mock.MockAddresses {
```

Full Mock Generated

Implement Only What
You Need To Test

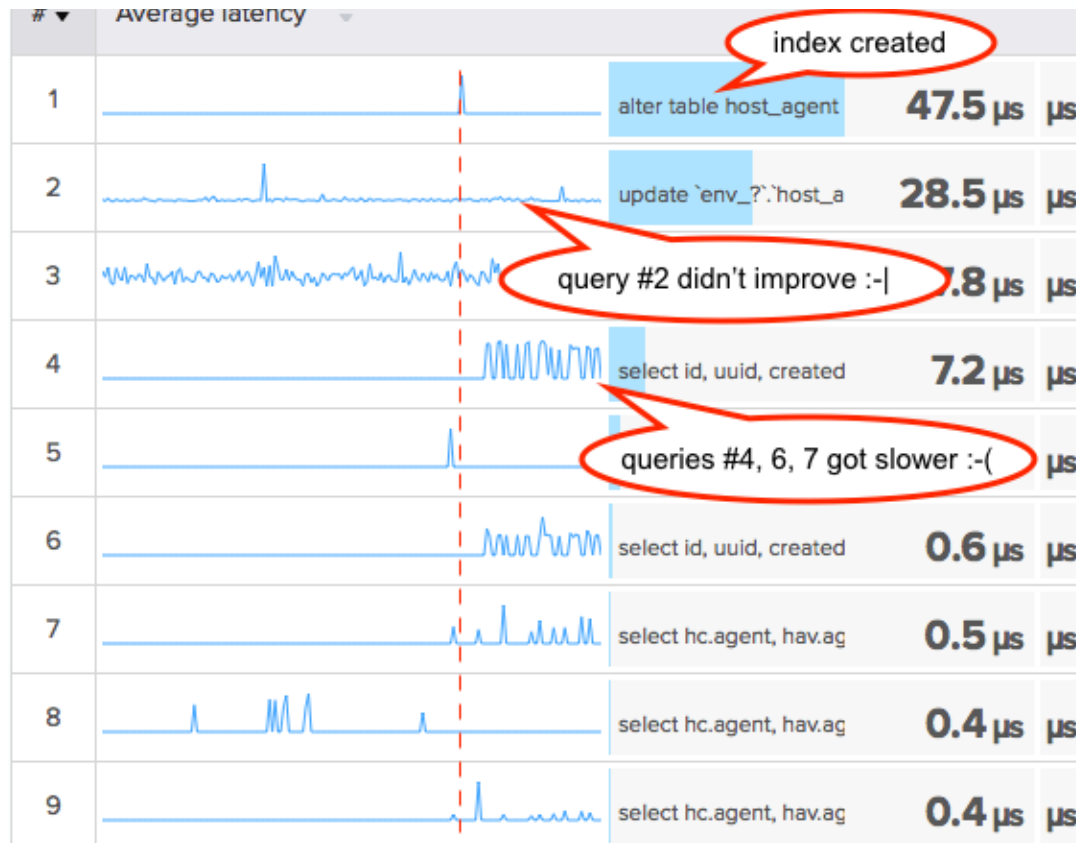
```
def setAddress(ip: String, address: Address) {
  ips += (ip -> address)
}

override def get(
  address: _root_.scala.Option[String] = None,
  ip: _root_.scala.Option[String] = None,
  requestHeaders: Seq[(String, String)] = Nil
)(implicit ec: scala.concurrent.ExecutionContext) = Future {
  ip match {
    case None => Nil
    case Some(ipAddress) => Seq(ips.get(ipAddress)).flatten
  }
}
```

www.apibuilder.io



Tooling: Real Time DB Monitoring



<https://www.vividcortex.com/blog/2014/04/22/visualizing-impact-index-change/>

Tooling: Super Simple Alerts from Log



Log a prefix Schedule a real time alert



Sumo Logic

to tech ▾

Jun 17 (8 days ago)

Saved Search	FlowAlertError - ReconcileSubmittedOrdersInShopify
Search String	"FlowAlertError [ReconcileSubmittedOrdersInShopify]"
Time Range	06/17/2017 12:00:00 PM EDT to 06/17/2017 12:15:00 PM EDT
Run Frequency	Every 15 minutes
Notification Threshold	Greater than 0
Run At	06/17/2017 12:18:14 PM EDT

```
{  
  "log": "FlowAlertError [ReconcileSubmittedOrdersInShopify] - ...",  
  "stream": "stdout",  
  "time": "2017-06-17T18:30:36.131267324Z"  
}
```

■ Key Takeaways – Design Production to be Testable

Trust your tests, run subset in production

Invest in continuous delivery

Sandbox accounts are powerful

High quality, trustworthy mocks

Real-time feedback from production



FLOW

Thank You!

We're hiring: <https://www.flow.io/careers>

QCon
NEW YORK

Michael Bryzek

mike@flow.io / @mbryzek

Cofounder / CTO Flow

Cofounder / ex-CTO Gilt