API Testing: From Entry Level to PhD in 40 minutes

with API Fortress

What is an API, anyway?

API Stands for: Application Programming Interface

Web Development: Content Delivery Method

Mechanism for moving formatted data from place to place.

Server to server, server to client, microservice to microservice

Exposes business logic to the outside world

What does an API response look like?

```
1  {
2      "this" : "is",
3      "one" : "example",
4      "of" : "JSON",
5      "formatting" : "!"
6  }
```

REST/JSON

```
1  <main>
2      <body>
3          <text>"This is what an xml response might look like!"</text>
4      </body>
5  </main>
```

SOAP/XML

How are API Responses formatted?

Most responses will be either JSON or XML format.

JSON: Javascript Object Notation

XML: eXtensible Markup Language

Other potential formats:

HTML: Hypertext Markup Language

Plaintext: Just text!

Still JSON but very worth mentioning: GraphQL

Benefits of REST

Browser clients are more supportive of REST, which makes up 70% of the API environment.

REST APIs are often more performative.

More allowable data formats

JSON is a bit more friendly than XML

Benefits of SOAP

Less required coding in the application layer for security, trust, etc.

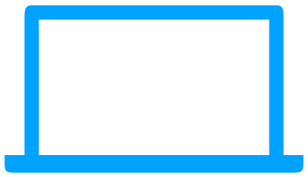
Greater transactional reliability (ACID compliance)

Simpler across firewalls or proxies without modifications to the protocol itself

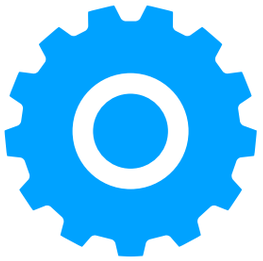
SOAP is highly extensible

THE TECHNICAL BITS!

What does an API look like under the hood?



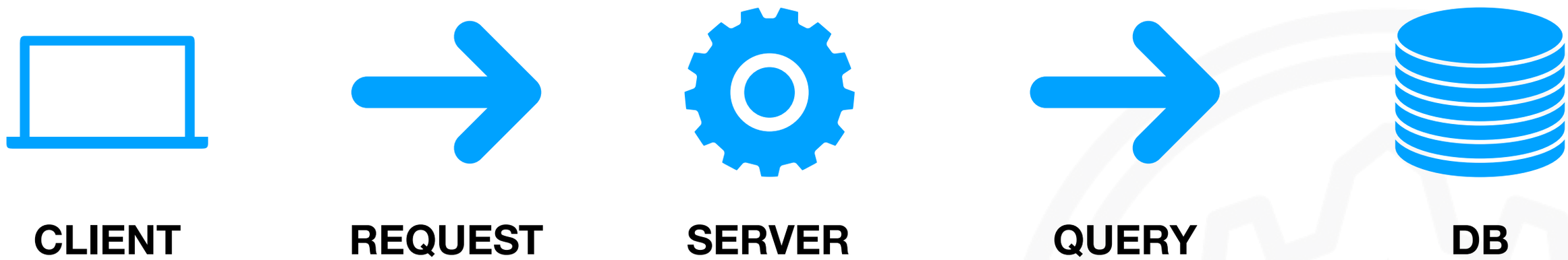
CLIENT



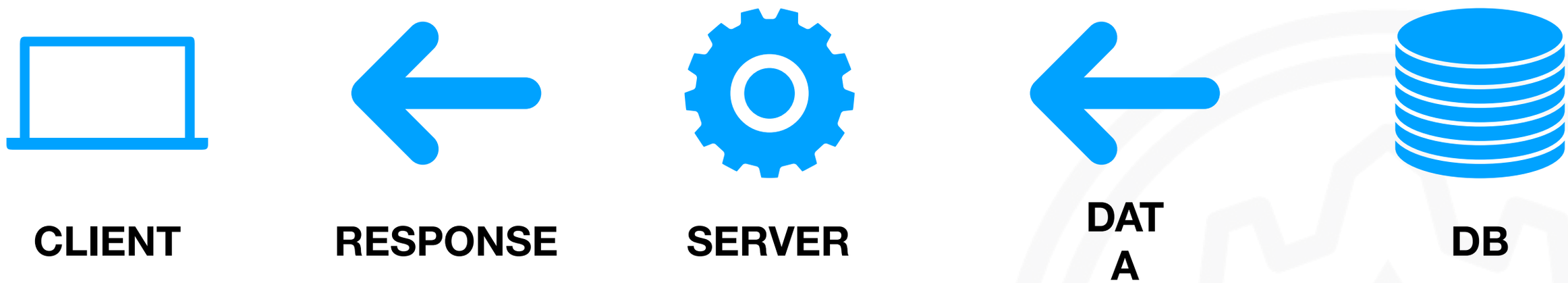
SERVER

DB

What does an API look like under the hood?



What does an API look like under the hood?



No, no. UNDER the hood.

```
1   const router = require('express').Router()
2
3   router.get('/', (req, res, next) => {
4       res.send("HI")
5   })
6
7   module.exports = router
8
```

HI

When the Machine Breaks Down

Case 1: The Case of the Missing Categories

Company is a retail organization that has over 50000 items in inventory

Never saw errors during UI testing

API Testing threw 3000 errors on *every single run* of the Products endpoint

Revenue — —

Case 2: The API that Always Mounted

Company was a retail organization with a very diverse set of products

Every single product was returning with “Mounting Instructions”

Current API testing tool could not account for different schemas in one API endpoint

Introduced conditional logic which led to… less mounting.

Case 3: François and the CharSet

APIs typically have a defined CharSet

This specific company did *not* account for the “cédille” character

Any time it appeared, it was replaced with “?” on a “write” operation

François became Fran?ois

Anything that referenced the users name against a DB entry… blew up.

Case 4: No, *you* broke it.

Photo sharing and printing platform

2 siloed teams - Image Processing and Print Order Processing

Image Processing changed their API schema

Print Order Processing was no longer able to utilize the API delivered data from Image Processing

Rule 1: Keep it DRY

- DRY - Don’t Repeat Yourself

- Parameterize data and code wherever you can

- Increase the modularity of the work you’re doing

- Create reusable tests

Rule 2: Make Your Intentions Clear

How am I going to remember what this test is for?

How is the next developer or tester going to know what this test is for?

We need to do increasing amounts of work if our tests can't be reused

Collaboration is very difficult when the intent behind a test is obscured

Rule 3: Act Like the Consumer Would!

Stop creating test cases in a vacuum

Many APIs exist in the scope of a system

Testing individual parts is important, but testing full workflows is more important

Simulate user workflows to find the holes

Rule 4: Eliminate Static Data Sources!

Static data is rarely your friend

Caveat: Some endpoints rely on static data. That's okay!

Use live data wherever possible

Where live data is unavailable, use mock responses that match your expected response

What *is* API Fortress?

About API Fortress API Fortress is a complete performance and quality solution for companies that care about their APIs.

A web-based platform to help teams evaluate API accuracy, monitor performance, and simulate load. Reduce costs with automated test generation, save time with an intuitive interface, and validate deployments to catch problems before your customers or partners.

Contact Jason at support@apifortress.com if you'd like to learn more about API Fortress