Consumer Driven Contract Testing with Pact and Docker

By Harry Winser
@Hazz223
OR... CONFIDENCE IN PRODUCTION

rightmove
find your happy
In the beginning, there were monoliths...

Image from: http://odino.org/on-monoliths-service-oriented-architectures-and-microservices/
... And then came Microservices...
... Which meant loads more communication...
Example

- Email Service
- Invoice Service
- Authentication Service
- Backend for frontend
Example API

http://email.service.com/user/125635

```json
[
  {
    "to": "testington@test.com",
    "from": "anotherTest@someone.com",
    "subject": "Help I'm stuck in a presentation",
    "body": "Dear Testing, it's happened again"
  },
  {
    "to": "testington@test.com",
    "from": "anotherTest@someone.com",
    "subject": "RE:Help I'm stuck in a presentation",
    "body": "Dear Testing, I'm still awaiting a reply"
  }
]
```
...And so chaos ensued

Image from: http://lrhslaughterhouse5.weebly.com/chaos.html
Consumer Driven Contracts
What are Consumer Driven Contracts (CDCs)?

“A contract between a consuming service and a providing service, stating what the consumer wants from a providing service, in a defined format”
Benefits of Consumer Driven Contracts

• Enable services to be deployed independently
• Enables teams to work independently from each other
• Enables verification of external endpoints – am I building what is wanted?
What they aren't

• A test of business logic
• An SLA between services
• A way of validating / verifying external services
Back to the API

http://email.service.com/user/125635

[{
  "to": "testington@test.com",
  "from": "anotherTest@someone.com",
  "subject": "Help I'm stuck in a presentation",
  "body": "Dear Testing, it's happened again"
},
{
  "to": "testington@test.com",
  "from": "anotherTest@someone.com",
  "subject": "RE: Help I'm stuck in a presentation",
  "body": "Dear Testing, I'm still awaiting a reply"
}]


How Rightmove use CDCs
CDC check list

- A consistent format to describe Request/Responses
- An easy way to create Consumer Driven Contracts
- A way of storing these Contracts
- An easy way to test CDCs in our CD pipeline
- A way of enabling us to Rollback an application
- An easy way to run tests locally
A consistent format

Loads of languages available:
- Java
- C#
- Go
- Python
- Ruby
- PHP
- JavaScript
- Plus more
Rightmove’s CI and CD pipeline

Jenkins

Images from: https://jenkins.io/
Rightmove’s Artifacts

Fitting CDC’s into the pipeline
Running the Pact Tests for a provider
Example: UK-Location

1. Download latest Property Web/Property Management pacts for UK-Location
2. Download the stubbed UK-Location
3. Start the UK-Location
4. Gradle runner reads the Pacts, and sends requests and checks the response
5. Shut down the UK-Location
6. Publish the results
Consumer tests
Example – Property Web

1. Download latest Pacts for a Property-Web, for UK-Location and Property-Search providers
2. For each provider
   a. Download stub
   b. Gradle runner reads the Pact between the Property Web and provider, and sends requests / checks response
   c. Shut down provider
3. Publish results
CDC check list

• A consistent format to describe request responses
• An easy way to create Consumer Driven Contracts
• An easy way to add these tests to our CI/CD pipelines
• A way of storing these contracts
• A way of enabling us to Rollback an application
• An easy way to run tests locally
Rollbacks
Provider tests – Latest, Live, Rollback

[Diagram showing the process]

1. Download Pacts for each Consumer
2. Download Provider Stub
3. Start Provider
4. Run tests against Provider
5. Shut down provider
6. Publish Results

Consumers Latest, Live, and Rollback pact

Artifact repository (Nexus)

Pact Broker

Deployed Versions
Example: UK-Location

1. Download Latest, Live, and Rollback Property Web/Property Management pacts for the latest UK-Location
2. Download the latest stubbed UK-Location
3. Start the latest UK-Location
4. Gradle runner reads the Pacts, and sends requests and checks the response
5. Shut down the UK-Location
6. Publish the results
Consumer tests – Latest, Live, Rollback
Example: Property-Web

1. Download latest Pacts for Property Web, for UK-Location and Property-Search
1. For each provider (latest, live, and rollback)
   2. Download
   3. Gradle runner reads the Pact between the consumer and provider, and sends requests / checks response
   4. Shut down provider
2. Publish results
Are we there yet?

• A consistent format to describe request responses
• An easy way to create Consumer Driven Contracts
• An easy way to add these tests to our CI/CD pipelines
• A way of storing these contracts
• A way of enabling us to Rollback an application
• An easy way to run tests locally
Testing Locally

• Faster to spin up and use
• Way less custom code / logic
• Easier to run different project types – not just java
• Platform-agnostic

Images from: docker.com
Provider Tests

Images from: docker.com
Consumer Tests

Images from: docker.com

Download Pacts

Download Provider Information

Find Live and Rollback

Deployed Versions

Docker Trusted Registry

Latest Consumer Pact

Download Provider Stub

Start provider

Run tests against provider

Shut down provider

Publish Results

Images from: docker.com
Are we there yet?

- A consistent format to describe request responses
- An easy way to create Consumer Driven Contracts
- An easy way to add these tests to our CI/CD pipelines
- A way of storing these contracts
- A way of enabling us to Rollback an application
- An easy way to run tests locally
The Future

• Parallel Tests
• Greater insights into our services, using Pacts
• Easier Pact version upgrading
• Open sourcing what we’ve done
Questions?

Feed back and comments: @Hazz223

We’re also hiring.
Come speak to me after!
Or email me:
harry.winser@rightmove.co.uk
Or head here:
http://plc.rightmove.co.uk/careers/current-vacancies.aspx

References
• Consumer Driven Contracts: A Service Evolution Pattern
  https://martinfowler.com/articles/consumerDrivenContracts.html
• The Pact Foundation: https://github.com/pact-foundation
• Pact JVM library: https://github.com/DiUS/pact-jvm
• Docker: https://www.docker.com/what-docker