**What is Pact test?**

* Contract Testing tool to ensure that services (such as an API provider and a client) can communicate with each other
* Consumer-driven -Consumer takes part in the creation of the API of the provider
* Connect a set of unit tests, one set for service consumer and one set for service provider.
* TDD for microservices
* Effectively, the tests are done against the contract.

Benefits:

* Help address the pain points with microservices (distributed) architectural system.
* Eliminate integration hell
* Enforce compatibility testing.
* Not necessary to deploy all microservices to perform end to end testing
* If we have breaking changes, we will know who is affected
* 

How it works?

* The consumer tests contain the specifications (usually the mobile frontend) about the HTTP request and the shape of the response
* When the test is run, the Pact framework convert the specifications into a contract and upload to a repository (called pact broker)
* Pact broker setup at <https://pact.shared.frigg.li/>
* The broker is a web service that contains a collection of contracts and association between consumers and producers

Pact workflow



Resources:

<https://docs.pact.io/>

<https://github.com/pact-foundation>

<https://github.com/pact-foundation/pact-python>

<https://github.com/Mattersight/pact-net-messages>

<https://www.baeldung.com/pact-junit-consumer-driven-contracts>

<https://github.com/christian-draeger/pact-example>

<https://github.com/pact-foundation/pact-js/tree/feat/message-pact#asynchronous-api-testing>