

API fundamentals

5 whys??

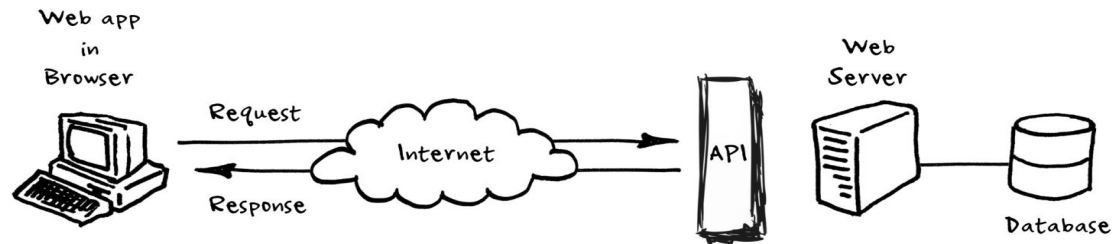


What is API

- Acronym for “Application programming interface”
- API is an intermediary which the software application accesses the server / resources

How does API work?

- APIs acts as middleman between the systems
- APIs communicate through set of definitions and protocols that allow software components to talk to each other



Why APIs??

- APIs are technology agnostic
- APIs provide the abstraction of the functionality between the two systems
- APIs can add an additional layer of security
- APIs can be reused

Who & where APIs are used??

- APIs are very versatile and can be used on web-based systems, operating systems, database systems and computer hardware.
- All social media APPs - Twitter, instagram, facebook
- All content streaming platforms - Netflix, Amazon, Hotstar
- All cloud platforms - AWS, Azure, PCF
- Selenium uses API to process the selenese commands
- JIRA/ Service now

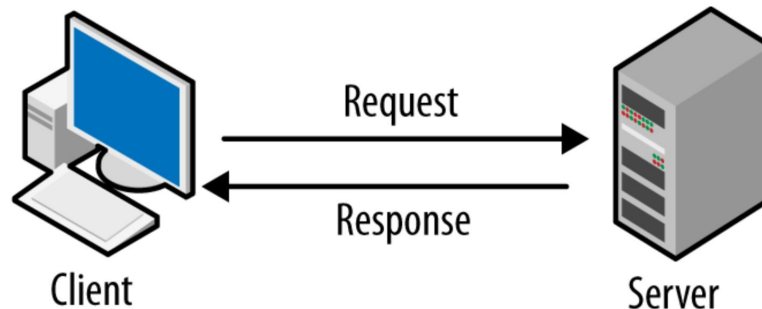
Types of APIs

- Request driven / sync APIs
- Event driven / Async APIs

Sync / request driven APIs

- An API that makes an explicit call to another service and awaits on response then process

Eg:- A payment gateway application calls the check balance API on the bank side



Async / Event driven APIs

- These works based on events, these systems subscribes to a messaging channel and keep on receiving the data that the publisher publishes
- These systems never explicitly call a service for data

Eg: While booking a OLA cab, all the drivers within the radius are listening to the topic and get notified on a ride

All the followers on social media get notified upon their follower tweet / post

Types of request driven APIs

- Simple object Access protocol (SOAP)
- Representational state transfer (REST)
- GraphQL
- GRPC

SOAP - simple object access protocol

- A messaging protocol which exchanges structured info in the implementation of web services
- It only supports XML for data exchange
- It supports only one HTTP verb, POST
- It works purely based on WSDL (web service descriptive language)
- It carries heavier payload
- It can't be cached as its stateful

REST- Representational state transfer

- is an architectural style to create webservices, built over HTTP protocol
- It supports JSON, XML, HTML, Multipart
- It supports many HTTP methods - POST, PUT, PATCH, GET, DELETE
- It is stateless and can be cached
- Carries lesser payload

GraphQL

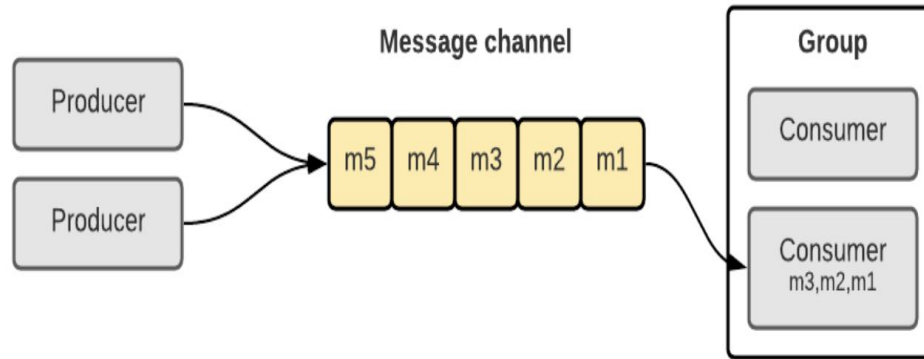
- It's a query language for API, implemented by Facebook
- It accesses multiple resources in single request
- It's very faster and gets the predictable results
- It uses types and fields
- It uses only one endpoint to fetch all the information

GRPC- google remote procedure call

- gRPC is a technology for implementing RPC APIs that uses HTTP 2.0 as its underlying transport protocol
- It works based on protocol buffers, googles serialized technique
- Unlike REST gRPCS addresses the procedures and hides the data

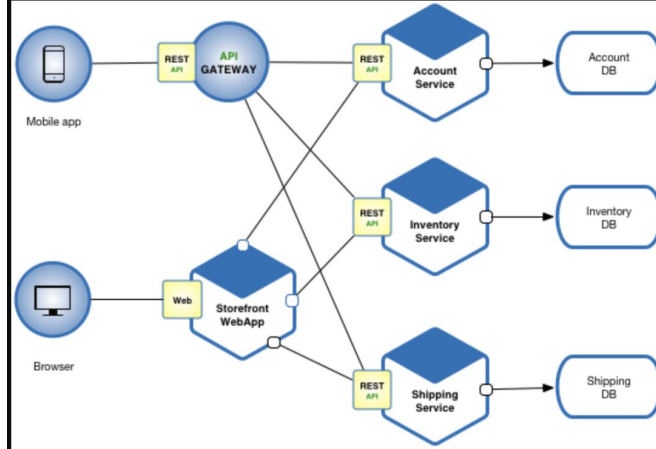
Types of Async APIs

- Kafka
- Pubsub
- Webhooks
- Web socket



What is microservice?

- Microservice is an architectural style that structures an application as a small logical subsystems
- Microservices can have one or a few APIS exposed



Difference between API, Web service & Microservice

API

An intermediary between systems, could be a webservice, a utility, .jar, library , driver

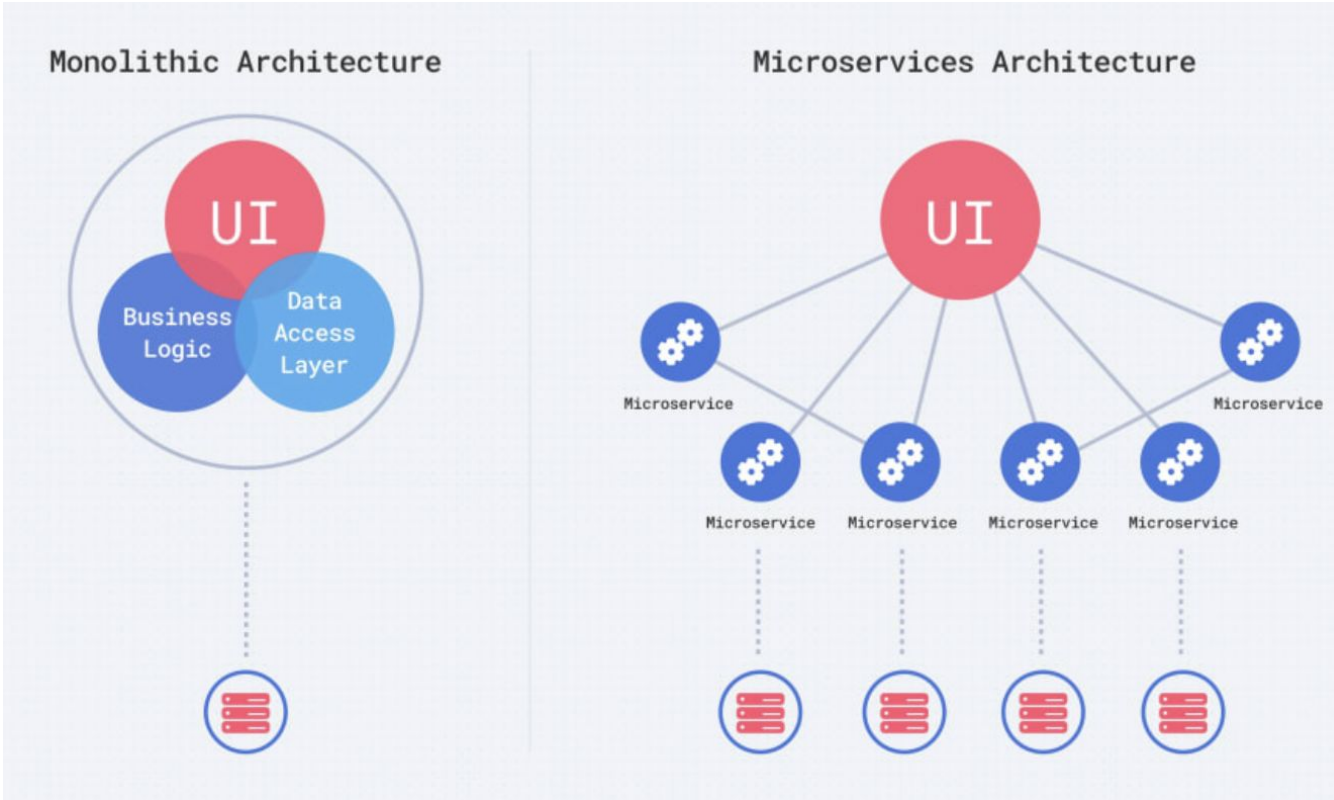
Web service

Webservice is an API that works over web (HTTP& HTTPS)

Microservice

Its an architecture that breaks the bigger apps into smaller cohesive subsystems, it consists of web services

Difference between Monolith and Microservice



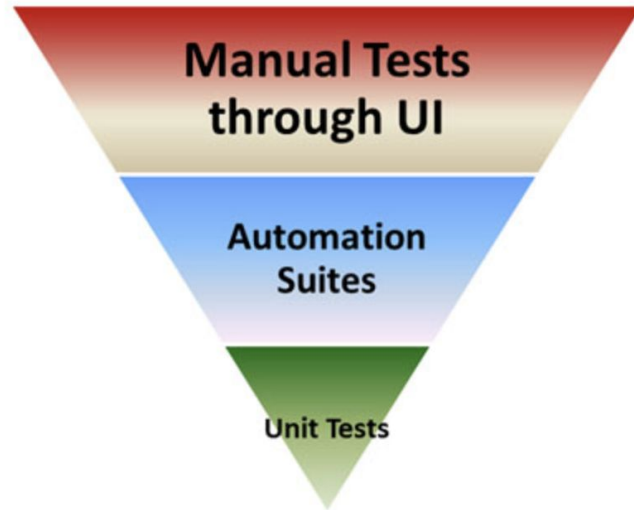
Why Microservices??

- Highly Highly maintainable and testable
- Loosely coupled
- Independently deployable
- Organized around business capabilities
- Owned by a small team

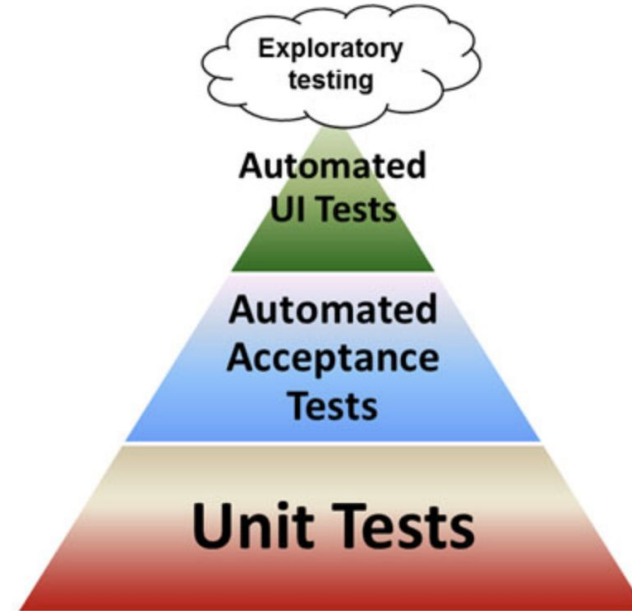
API first approach

- It's the modern application development strategy
- This principle states that identify the APIs and build first

Mike Cohn's Testing pyramid



Traditional
(find bugs)



Agile
(prevent bugs)

Test driven development

- A development principle that says not even a single line of production code w/o a test
- The development is done to make sure the tests are passed
- This strategy ensures that there are no big leaks to higher stages

Contract based testing

- An efficient testing technique to prevent the bugs that are of changes at contract level in a typical microservice environment
- Intelligent way of stubbing systems
- It is mainly of two types
 - Producer driven contract based testing
 - Consumer driven contract based testing

What is contract

- Contract is an agreement between the two microservices to negotiate with each other
- Contract consists
 - Metadata,
 - Request
 - Response

```
org.springframework.cloud.contract.spec.Contract.make {  
    request {  
        description("""  
Represents a successful scenario of getting a beer  
  
given:  
    client is old enough  
when:  
    he applies for a beer  
then:  
    we'll grant him the beer  
""")  
        method 'POST'  
        url '/check'  
        body(  
            age: value(consumer(regex('[2-9][0-9]')))  
        )  
        headers {  
            header 'Content-Type', 'application/json'  
        }  
    }  
    response {  
        status 200  
        body( """  
        {  
            "status": "OK"  
        }  
        """)  
        headers {  
            header(  
                'Content-Type', value(consumer('application/json'),producer(regex('application/json.*')))  
            )  
        }  
    }  
}
```


What is provider / publisher?

- The service who provides the functionality / exposes the endpoint acts as provider
- It's the service that is being called

What is consumer / subscriber ?

- The service that consumes the functionality of the provider service is the consumer or subscriber
- It's the caller service

API Testing process

- Document the API specifications
- Build request as per contract
- Get details of endpoint
- Make a request
- Validate status code and other response fields

Why should we learn API

- Faster feedback
- Easier and isolated
- Less failures
- Growing API based apps - API first approach