# Web Service Testing Approach

This is a type of software testing that involves testing APIs directly and also as a part of integration testing to check whether the API meets expectations in terms of functionality, reliability, performance, and security of an application.

**Level of testing**

* Unit Testing
* Functional Testing
  + Mocking the service
    - Static response
    - Dynamic response
* Integration Testing
* Performance Testing

**Common challenge of Web Services testing**

* Use of external web services
* Implementation of using complex standard & protocol
* Headless nature of web services

**Validating Service**

* No Soap fault assertion
* XPath match assertion
* Response assertion
* Soap Fault response
* Contains – expected string token in the response
* Not contains – searches for non-existence of a string token
* XQuery match
* JSON path match

**Compliance, Status and Standard**

* Invalid HTTP status codes
* Schema compliance
* JSON schema validation
* XML schema validation against XSD schema
* Validate HTTP Status code

**Error messages /Bad error**

* Change unexpectedly
* Don’t match their status code (HTTP)
* Don’t tell user what they did wrong
* Don’t tell user what they need to do right
* Show clients stuff that could be misused

**Security Testing**

* Do you handle identity
* Do you maintain integrity
* Do you assess vulnerability (formatting, Data types values, dates & time zones, regional formatting, localized messages /error)
* Does the API align with client domain(process / workflow, nomenclature, related APIs)
* Execute complex authentication, encryption and access control tests scenarios
* Generating a broad range of penetration attack scenario involving parameter fuzzing, injections, large payloads etc
* Running penetration attack scenarios against your existing functional test scenarios
* Monitoring back end during test execution in order to determine where security is actually compromised

**Performance Testing**

* API responds within agreed SLA
* Does API perform consistently
* Does API scale – What if load increases from 100s request per minute to 1000 requests/second over the timespan of a couple of weeks or days?

**Things to think about when testing API**

1. Who needs it

Who are actual stakeholders for API quality? Who really needs it – and what is important to them

1. Is your API available
2. Is your API consistent
3. Is your API transparent
4. Is your API secure
   1. Do you handle identity
   2. Do you maintain integrity
   3. Do you assess vulnerability (formatting, Data types values, dates & time zones, regional formatting, localized messages /error)
   4. Does the API align with client domain(process / workflow, nomenclature, related APIs)
   5. Execute complex authentication, encryption and access control tests scenarios
   6. Generating a broad range of penetration attack scenario involving parameter fuzzing, injections, large payloads etc
   7. Running penetration attack scenarios against your existing functional test scenarios
   8. Monitoring back end during test execution in order to determine where security is actually compromised
5. Have you tested for expected and unexpected results (not only content of the results but also the format)