Rest API:

https://www.softwaretestingmaterial.com/api-testing-interview-questions/

In API Testing our main focus will be on Business logic layer of the software architecture. API testing can be performed on any software system which contains multiple APIs.

Using POST request, our intent is to create a new object on the server whereas with PUT request, our intent is to replace an object by another object.

GET request only allows read only rights. It enables you to retrieve data from a server but not create a resource. PUT or POST methods should be used to create a resource.

Also, another difference is that when you want to update a resource with PUT request, you have to send the full payload as the request whereas with PATCH, you only send the parameters which you want to update.

Web Services can be implemented in different ways, but the following two are the popular implementations approaches.

1. SOAP (Simple Object Access Protocol)
2. REST (Representational State Transfer architecture) representation from client -> http method (post,get,put,delete,patch) -> server (state transfer)

   https://reqres.in/

What are the common tests that are performed on APIs?

Some of the common tests we perform on APIs are as follows.

1. Verify whether the return value is based on input condition. Response of the APIs should be verified based on the request.
2. Verify whether the system is authenticating the outcome when the API is updating any data structure
3. Verify whether the API triggers some other event or request another API
4. Verify the behaviour of the API when there is no return value

What exactly needs to be verified in API Testing?

Basically, on API Testing, we send a request to the API with the known data and we analyse the response.

1. Data accuracy

2. HTTP status codes

3. Response time

3. Error codes in case API returns any errors

4. Authorization checks

5. Non-functional testing such as performance testing, security testing

How is UI testing is not similar to API testing?

UI (User Interface) testing is to test the graphical interface part of the application. Its main focus is to test the look and feel of an application. On the other hand, API testing enables communication between two different software systems. Its main focus is in business layer of the application.

Can you use GET request instead of PUT to create a resource?

No, GET request only allows read only rights. It enables you to retrieve data from a server but not create a resource. PUT or POST methods should be used to create a resource.

What is the difference between PUT and POST methods?

PUT and POST methods are sometimes confused in regards to when each should be used. Using POST request, our intent is to create a new object on the server whereas with PUT request, our intent is to replace an object by another object.

POST should be used when the client sends the page to the server and then the server lets the client know where it put it. PUT should be used when the client specifies the location of the page

What is Soap?

SOAP stands for Simple Object Access Protocol. It is an XML based messaging protocol. It helps in exchanging information among computers.

7. What is Rest API?

REST stands for Representational State Transfer. It is a set of functions helping developers in performing requests and receive responses. Interaction is made through HTTP Protocol in REST API.

Difference between SOAP and REST?

SOAP:

1. SOAP is a protocol through which two computers communicate by sharing XML document

2. SOAP supports only XML format

3. SOAP does not support caching

4. SOAP is slower than REST

5. SOAP is like custom desktop application, closely conneccted to the server

6. SOAP runs on HTTP but envelopes the message

REST:

1. REST is a service architecture and design for network-based software architecture

2. REST supports different data formats

3. REST supports caching

4. REST is faster than SOAP

5. REST client is just like a browser and uses standard methods An application has to fit inside it

6. REST uses the HTTP headers to hold meta information

SOAP	REST
SOAP is a protocol.	REST is an architectural style.
SOAP can't use REST because it is a protocol.	REST can use SOAP web services because it is a concept and can use any protocol like HTTP, SOAP.
SOAP only permits XML.	REST permits many different data formats including plain text, HTML, XML, and JSON...
SOAP requires more bandwidth and more resources.	REST requires less bandwidth and less resources.
SOAP supports both SMTP and HTTP protocols.	REST requires use of HTTP only.
SOAP is more reliable than REST.	REST is less secure than SOAP.
In most cases, SOAP is faster than REST.	REST is slower than SOAP.
SOAP defines its own security.	RESTful web services inherits security measures from the underlying transport.

SOAP UI SoapUI allows you to test REST and SOAP APIs with ease – as it has been built specifically for API testing.	**Postman** Postman is a plugin in Google Chrome, and it can be used for testing API services. It is a powerful HTTP client to test web services. For manual or exploratory testing, Postman is a good choice for testing API.
➢ Quick and Easy Test Creation: Point-and-click, drag-and-drop, functionality makes complicated tasks (like working with JSON and XML) simple ➢ Powerful data-driven testing: Load data from Excel, files, and databases to simulate the way consumers interact with your APIs ➢ Seamless Integrations: Integrates with 13 API management platforms, supports REST, SOAP, JMS, and IoT	➢ With Postman, almost all modern web API data can be extracted ➢ You can create a collection of REST calls and save each call as part of a collection for execution in future ➢ For transmitting and receiving REST information, Postman is more reliable ➢ Unlike CURL, it is not a command line based tool, which makes this tool hassle free of pasting text into command line window

WEB SERVICE	API
All web services are APIs.	All APIs are not web services.
It can only be hosted on IIS.	It can be hosted within an application or IIS.
It is not open source but can be used by any client that understands XML.	It is open source and it can be used by any client that understands JSON or XML.
It requires a SOAP protocol to receive and send data over the network, so it is not a light-weight architecture.	It is light-weight architectured and good for devices which have limited bandwidth, like mobile devices.
A Web service uses only three styles of use: SOAP, REST and XML-RPC for communication.	API may use any style of communication.
It only supports the HTTP protocol.	It supports the HTTP protocol: URL, Request/Response Headers, caching, versioning, content formats.

5 top open-source API testing tools: How to choose

https://techbeacon.com/app-dev-testing/5-top-open-source-api-testing-tools-how-choose

API testing questions:

https://www.youtube.com/watch?v=ggK_MiUkKNc

Test Plan:

Develop Test Strategy

Test strategy is a guideline to be followed to achieve the test objective and execution of test types mentioned in the testing plan. It deals with test objective, test environment, test approach, automation tools and strategy, contingency plan, and risk analysis

In the Test Plan, test focus and project scope are defined. It deals with test coverage, scheduling, features to be tested, features not to be tested, estimation and resource management.

https://reqtest.com/testing-blog/how-to-write-a-test-plan-2/

HTTP code error:

https://restfulapi.net/http-status-codes/

400 (Bad Request)

401 (Unauthorized) un authorized

403 (Forbidden) F b den

404 (Not Found)

405 (Method Not Allowed)

500 (Internal Server Error)

501 (Not Implemented)

HttpCode

https://www.cnblogs.com/DeasonGuan/articles/Hanami.html

```
'success' => [
        'error_code' => 0,
        'http_code' => 200, //服务器成功返回网页
        'message' => 'Success'
    ],
    'player exists' => [
            'error_code' => 4002,
            'http_code' => 400,//服务器不理解请求的语法。
            'message' => 'Player already exists'
        ],
    'token mismatch' => [
            'error_code' => 4031,
            'http_code' => 403, //  服务器拒绝请求。
            'message' => 'Token mismatch'
        ],
    'api not found' => [
            'error_code' => 4040,
            'http_code' => 404,//   服务器找不到请求的网页。
            'message' => 'Api not found'
        ],
    HTTP Status 401 （未授权）
    ->请求要求身份验证。 对于需要登录的网页，服务器可能返回此响应。
    'api not implemented' => [
            'error_code' => 5001,
            'http_code' => 500,// HTTP Status 500 （服务器内部错误）
            'message' => 'Api not implemented'
        ],
```

UI automation:

Implicit Wait: In Implicit wait, if WebDriver cannot find an element in the Document Object Model (DOM), it will wait for a defined amount of time for the element to appear in the DOM. The Implicit wait may slow down your tests, because once set, the implicit wait is set for the life of the WebDriver object's instance.

Explicit waits: are better than implicit wait. Unlike an implicit wait, you can write custom code or conditions for wait before proceeding further in the code. An explicit wait can be used where synchronization is needed, for example the page is loaded but we are still waiting for a call to complete and an element to appear.

//xpath:

//a[contains(@href,test1)]

//*[@type='submit' or @name='login'] //And

//lable [starts-with(@name,'btn')]

//*[@id='home']/child::li ////*[@id='home']/child::li[1]

Automation structure

Src : test script

Lib : methods for invoking

Class : test class

Log: output the log file/report

Element files -> elements file

Test files -> excel or other file to save the test data

Which test cases to automate:

Repetitive tests that run for multiple builds

Tests that use multiple data values for the same actions (data driven tests)

Identical tests that need to be executed using different browsers

Tests that run on several different hardware or software platforms and configurations

OOP:

Abstraction

Encapsulation in cap sulation

Inheritance.

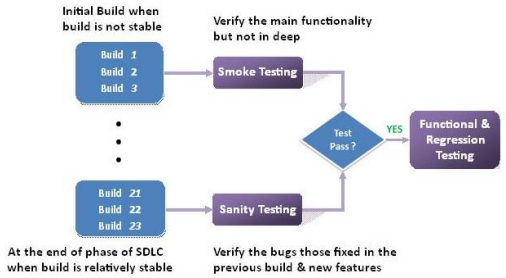
Polymorphism pao li mao fei zm

Function testing

- Unit Testing
- Smoke testing
- Sanity testing
- Integration Testing
- Interface Testing
- System Testing
- Regression Testing
- UAT

non functional testing types:

- Stress testing
- Usability testing
- Localization testing and Internationalization testing



Activies in Agile

https://www.softwaretestinghelp.com/test-automation-interview-questions/

PI planning -> grooming feature, initial backlog

Sprint meeting for a one Month or four- week, for two week sprint , breakdown use story ->write down the acceptance criteria , define the work

In sprint

Daily scrum meeting about 15 mins, talk about risk and problem.

Velocity

CI/CD

https://mindmajix.com/jenkins-interview-questions-answers

Performance Testing Tutorial

https://www.guru99.com/performance-testing.html#3

Type of performance

Load testing - checks the application's ability to perform under anticipated user loads. The objective is to identify performance bottlenecks before the software application goes live.

Stress testing - involves testing an application under extreme workloads to see how it handles high traffic or data processing. The objective is to identify the breaking point of an application.

Example Performance Test Cases (Long Load time /Poor response time /Poor scalability /Bottlenecking (Disk usage,Network utilization))

Verify response time is not more than 4 secs when 1000 users access the website simultaneously.

Verify response time of the Application Under Load is within an acceptable range when the network connectivity is slow

Verify response time of the application under low, normal, moderate and heavy load conditions.

Check the maximum number of users that the application can handle before it crashes.

Check database execution time when 500 records are read/written simultaneously.

Check CPU and memory usage of the application and the database server under peak load conditions

Login page test

Verify if a user will be able to login with a valid username and valid password. ->Positive

Verify if a user cannot login with a valid username and an invalid password. -> Negative

Verify if the font, text color, and color coding of the Login page is as per the standard.-> UI

Verify the time taken to log in with a valid username and password. ->Performance

Verify the login page and all its controls in different browsers -> browser

Non-functional:

Verify if a user cannot enter the characters more than the specified range in each field (Username and Password). Negative(security)

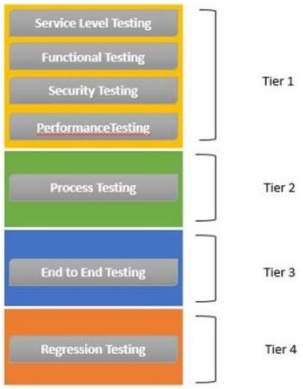
Verify the timeout functionality of the login session. (positive security)

Verify the Login page against SQL injection attack. (positive security) DDOS attack

Verify the implementation of SSL certificate. (positive security)

SOA Testing Tutorial

https://www.softwaretestinghelp.com/soa-testing/



Tier1

1) Service Level Testing:

Each service involved in the system is tested individually based on a Request and response method.

This test is mandatory and very important to proceed with other testing processes.

2) Functional Testing:

The test is conducted for services on their business needs to find if the response that is received is correct.

The business needs are first converted into the test cases and the request statements are formed.

Then the request statements are processed to see if the obtained responses are correct.

In case of invalid input data, proper error code should be thrown or proper error message should be triggered.

The formats of the response, as well as the negative scenarios, have to be executed.

3) Security Testing:

Whenever it comes to a web service, Security testing plays a key role in the success of the Testing process.

Authentication gateways, Payment gateways etc. should be encrypted when the data is parsed.

When it comes to XML, vulnerabilities such as CSRF, SQL injection should be verified.

4) Performance Testing:

Services used in the architecture are hosted so that a lot of other applications can make use of it. Performance testing makes sure the credibility of those services.

The testing of the services should be done to find out the following sets of result;

To determine the stability of the services.

To validate the scalability of the services.

Service behavior under Peak load conditions

To find the response times across services

Tier #2

1) Process Testing:

- This process involves the testing of various business processes.
- This should comprise of the integration scenarios of the Web services and application covering the business requirements
- Usage of simulators should be done to generate sample input data and validation should be done for the respective outputs.
- Data flow from different layers should be performed to prove smooth functioning of the system when it is integrated.

Tier #3

- 1) End to End Testing:
- This phase is meant to validate the business requirements both functionally and non-functionally.
- UI of the application is validated.
- The business process involved is tested.
- The end-to-end data flow is validated in this phase.
- Working with all services when the services are integrated to each other is validated.

Tier #4

1) Regression Testing:

- The stability of the system in incremental build releases is validated by this testing.
- This can be achieved either by Manual Testing/Automation Testing.

Verification and validation:

https://www.geeksforgeeks.org/differences-between-verification-and-validation/

Verification is the process of checking that software achieves its goal without any bugs. Static testing

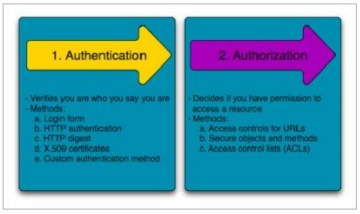
Validation is the process of checking whether the software product is up to the mark or in other words product has high level requirements. It is the process of checking the validation of product i.e. it checks what we are developing is the right product. Dynamic test

Levels of Testing

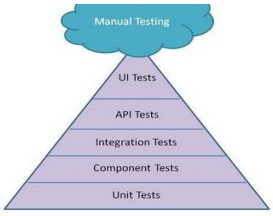
- Unit Testing
- Integration Testing
- Functional Testing
- System Testing
- End to end testing
- Sanity Testing
- Acceptance Testing
- Regression Testing
- Load Testing
- Stress Testing
- Performance Testing
- Volume Testing
- Usability Testing
- Installation Testing
- Recovery Testing
- Security Testing
- Compatibility Testing

HTTP stands for Hypertext Transfer Protocol.

HTTPS stands for Hypertext Transfer Protocol Secure

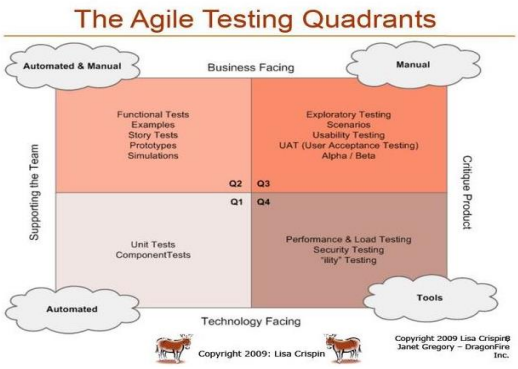


he terms are often used in conjunction with each other in terms of security,

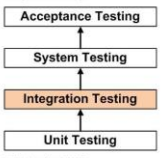


ıe test pyramid

nd the agile testing quadrants



INTEGRATION TESTING is a level of software testing where individual units are combined and tested as a group. The purpose of this level of testing is to expose faults in the interaction between integrated units. Test drivers and test stubs are used to assist in Integration Testing.



Definition by ISTQB

- **integration testing:** Testing performed to expose defects in the interfaces and in the interactions between integrated components or systems. See also *component integration testing, system integration testing*.
- **component integration testing:** Testing performed to expose defects in the interfaces and interaction between integrated components.
- **system integration testing:** Testing the integration of systems and packages; testing interfaces to external organizations (e.g. Electronic Data Interchange, Internet).