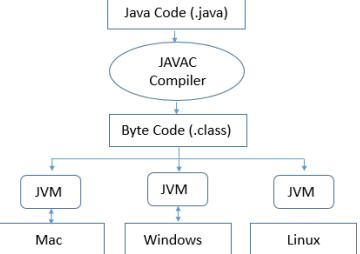
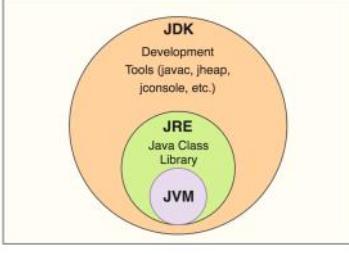


What is Java

29 July 2018 09:01

Introduction	<p>Java is "architecture-neutral, interpreted and portable". Java programs are compiled to bytecode that have no dependencies on a specific machine architecture. In order to run on a particular system, all you need is a Java Interpreter only which is JRE.</p>  <pre>graph TD; A[Java Code (.java)] --> B((JAVACompiler)); B --> C[Byte Code (.class)]; C --> D[JVM]; C --> E[JVM]; C --> F[JVM]; D --> G[Mac]; D --> H[Windows]; D --> I[Linux]</pre>
Platform Independent	<p>First, you write Java source file and compile it. When you compile Java source code, the result is a .class file that contains bytecode. The bytecode is the same no matter what platform you are on, provided that you are using Java Virtual Machine (JVM) which is platform dependent in nature. This JVM converts the byte code to machine code according to your original computer's machine architecture like x86,ARM etc. Java Virtual Machine (JVM) is of different type, according to computer system architecture, that means for x86 JVM will be different for ARM JVM.</p>
Components	<p>JVM – Java Virtual machine(JVM) is a very important part of both JDK and JRE because it is contained or inbuilt in both. Whatever Java program you run using JRE or JDK goes into JVM and JVM is responsible for executing the java program line by line hence it is also known as interpreter.</p> <p>JRE – JRE is an installation package which provides environment to only run(not develop) the java program(or application)onto your machine. JRE is only used by them who only wants to run the Java Programs i.e. end users of your system.</p> <p>JDK – JDK is Kit which provides the environment to develop and execute(run) the Java program.</p>  <pre>graph TD; A((JDK)) -.-> B((JRE)); B -.-> C((JVM))</pre>

Stack and Heap

30 July 2018 19:52

Stack	<p>Stack memory is a special region of the computer's memory that stores temporary variables created by each function. It contains method specific values that are short-lived and references to other objects in the heap that are getting referred from the method. Every time a function declares a new variable, it is "pushed" onto the stack. When a function exits, the block becomes unused and can be used the next time a function is called. The stack is always reserved in a last in first out (LIFO) order; the most recently reserved block is always the next block to be freed. This makes it really simple to keep track of the stack; freeing a block from the stack is nothing more than adjusting one pointer.</p> <p>Summary</p> <ul style="list-style-type: none">• It is used for static memory allocation and values in stack exist within the scope of the function. Once the function returns, values are discarded.• The stack grows and shrinks as functions push and pop local variables• There is no need to manage the memory yourself, variables are allocated and freed automatically• Stack has size limits• Stack variables only exist while the function that created them, is running• Every method has its own stack.
Heap	<p>Java Heap space is used by Java runtime to allocate memory to Objects and JRE classes. It is the runtime data area from which the Java Virtual Machine allocates memory for all class instances and arrays. Whenever we create any object, it's always created in the Heap space. Unlike the stack, there's no enforced pattern to the allocation and deallocation of blocks from the heap; you can allocate a block at any time and free it at any time. Any object created in the heap space has global access and can be referenced from anywhere of the program. Instance variables are created in heap. Java Garbage Collector is an automatic memory management system that reclaims heap memory for objects.</p> <p>Summary</p> <ul style="list-style-type: none">• It is used for dynamic memory and gets destroyed either by garbage collector or manually.• Variables can be accessed globally• No limit on memory size• Relatively slower access• No guaranteed efficient use of space, memory may become fragmented over time as blocks of memory are allocated, then freed

Access Modifier

29 July 2018 10:31

Modifier	public	protected	default	private
Same class	Yes	Yes	Yes	Yes
Same Package subclass	Yes	Yes	Yes	No
Same Package non-subclass	Yes	Yes	Yes	No
Different package subclass	Yes	Yes	No	No
Different package non-subclass	Yes	No	No	No

Exceptions

29 July 2018 12:28

Definition	<p>An exception is an unwanted or unexpected event, which occurs during the execution of a program i.e at run time, that disrupts the normal flow of the program's instructions.</p> <ul style="list-style-type: none"> Error: An Error indicates serious problem that a reasonable application should not try to catch. OutOfMemoryError, VirtualMachineError, AssertionError etc. Exception: Exception indicates conditions that a reasonable application might try to catch. 		
Exception Hierarchy	<p>All exception and errors types are sub classes of class Throwable, which is base class of hierarchy. One branch is headed by Exception. This class is used for exceptional conditions that user programs should catch. NullPointerException is an example of such an exception. Another branch, Error are used by the Java run-time system(JVM) to indicate errors having to do with the run-time environment itself(JRE). StackOverflowError is an example of such an error.</p> <pre> graph TD Object[Object] --> Throwable[Throwable] Throwable --> Exceptions[Exceptions] Throwable --> Error[Error] Exceptions --> Checked[Checked Exceptions Example: IO or Compile time Exception] Exceptions --> Unchecked[Unchecked Exceptions Example: Runtime or Null Pointer Exceptions] Error --> VM[Virtual Machine Error] Error --> Assertion[Assertion Error etc] </pre>		
How JVM Handles an Exception	<ul style="list-style-type: none"> A method creates an Exception Object containing the exception's name, description, and program state. This object is "thrown" and handed to the JVM. The JVM searches the Call Stack (the list of methods called) in reverse order for an appropriate exception handler (a method that can handle that type of exception). If a matching handler is found, the exception is passed to it. If no handler is found, the JVM's default exception handler prints the exception details (name, description, and call stack) and terminates the program abnormally. Exception in thread "xxx" Name of Exception : Description // Call Stack 		
Catch multiple exceptions	<p>A method can throw more than one exceptions. However that method needs to declare all the checked exceptions it can throw</p> <pre> try { //Do some processing which throws exceptions } catch(SQLException IOException e) { someCode(); } catch(Exception e) { someCode(); } </pre> <p>In this feature, you can catch multiple exceptions in single catch block. Before java 7, it was restricted to catch only one.</p> <p>The order of catch execution is whatever matches first, gets executed. If the first catch matches the exception, it executes, if it doesn't, the next one is tried and on and on until one is matched or none are. So, when catching exceptions you want to always catch the most specific first and then the most generic (as RuntimeException or Exception).</p>		
Checked vs Unchecked Exceptions	<p>1) Checked: are the exceptions that are checked at compile time. If some code within a method throws a checked exception, then the method must either handle the exception or it must specify the exception using throws keyword. For example, consider the following Java program that opens file at location "C:\test\a.txt" and prints first three lines of it. The program doesn't compile, because the function main() uses FileReader() and FileReader() throws a checked exception FileNotFoundException. It also uses readLine() and close() methods, and these methods also throw checked exception IOException</p> <p>2) Unchecked are the exceptions that are not checked at compiled time. In Java exceptions under Error and RuntimeException classes are unchecked exceptions, everything else under throwable is checked. Consider the following Java program. It compiles fine, but it throws ArithmeticException when run. The compiler allows it to compile, because ArithmeticException is an unchecked exception.</p>		
Custom Exception	<p>Create a custom class inheriting Exception and in the constructor accept the error message and pass it to super constructor.</p> <table border="1" style="width: 100%; border-collapse: collapse;"> <tr> <td style="padding: 5px; vertical-align: top;"> Define the Exception Class </td><td style="padding: 5px; vertical-align: top;"> Create a new class that extends one of the built-in exception classes: <ul style="list-style-type: none"> Exception: For general-purpose exceptions. RuntimeException: For unchecked exceptions. Other specific exception classes like IOException, SQLException, etc., for more specialized exceptions. <pre> public class MyCustomException extends Exception { // Constructor to initialize the exception message public MyCustomException(String message) { super(message); } </pre> </td></tr> </table>	Define the Exception Class	Create a new class that extends one of the built-in exception classes: <ul style="list-style-type: none"> Exception: For general-purpose exceptions. RuntimeException: For unchecked exceptions. Other specific exception classes like IOException, SQLException, etc., for more specialized exceptions. <pre> public class MyCustomException extends Exception { // Constructor to initialize the exception message public MyCustomException(String message) { super(message); } </pre>
Define the Exception Class	Create a new class that extends one of the built-in exception classes: <ul style="list-style-type: none"> Exception: For general-purpose exceptions. RuntimeException: For unchecked exceptions. Other specific exception classes like IOException, SQLException, etc., for more specialized exceptions. <pre> public class MyCustomException extends Exception { // Constructor to initialize the exception message public MyCustomException(String message) { super(message); } </pre>		

	<pre> } } }</pre>
Throw the Exception	<pre> public void myMethod() throws MyCustomException { // ... if /* some condition is not met */) { throw new MyCustomException("Custom exception message"); } // ... }</pre>
Catch and Handle the Exception	<pre> try { myMethod(); } catch (MyCustomException e) { System.out.println("Caught a custom exception: " + e.getMessage()); // Handle the exception, e.g., log the error, display an error message, etc. }</pre>
Example	<pre> public class InvalidInputException extends Exception { public InvalidInputException(String message) { super(message); } } public class MyClass { public void validateInput(int input) throws InvalidInputException { if (input < 0) { throw new InvalidInputException("Input cannot be negative"); } } }</pre>

Strings

08 June 2018 11:47

Architecture	String is a class in Java and is immutable														
<code>String str1="Hello"; String str2="World";</code>	Here hello and World are the objects and str1 and str2 points to the objects														
<code>String str1="Hello"; String str2="Hello";</code>	Here only one object Hello will be created in string constant pool and str1 and str2 will point to it. This is called String Literal Method. For this declaration objects are created in String Constant Pool .														
<code>String str1=new String("Welcome"); String str2=new String("Welcome");</code>	When new string function is called, it creates 2 objects , one in string constant pool(if not already present) as string literal and another in heap memory.														
Equality	We can use == operators for reference comparison (address comparison) and .equals() method for content comparison. In simple words, == checks if both objects point to the same memory location whereas .equals() evaluates to the comparison of values in the objects														
String Builder and Buffer(Thread Safe)	<ul style="list-style-type: none">If a string is going to remain constant throughout the program, then use String class object because a String object is immutable.If a string can change (example: lots of logic and operations in the construction of the string) and will only be accessed from a single thread, using a StringBuilder is good enough.If a string can change, and will be accessed from multiple threads, use a StringBuffer because StringBuffer is synchronous so you have thread-safety. <table border="1"><tr><td>append(String str)</td><td>sb.append("Hello");</td></tr><tr><td>insert(int offset, String str)</td><td>sb.insert(1, "Java");</td></tr><tr><td>delete(int start, int end):</td><td>sb.delete(1, 3);</td></tr><tr><td>deleteCharAt(int index):</td><td>sb.deleteCharAt(2);</td></tr><tr><td>reverse()</td><td>sb.reverse();</td></tr><tr><td>setCharAt(int index, char ch):</td><td>sb.setCharAt(1, 'J');</td></tr><tr><td>IndexOf, lastIndexOf</td><td>indexOf(String.valueOf('c'))</td></tr></table>	append(String str)	sb.append("Hello");	insert(int offset, String str)	sb.insert(1, "Java");	delete(int start, int end):	sb.delete(1, 3);	deleteCharAt(int index):	sb.deleteCharAt(2);	reverse()	sb.reverse();	setCharAt(int index, char ch):	sb.setCharAt(1, 'J');	IndexOf, lastIndexOf	indexOf(String.valueOf('c'))
append(String str)	sb.append("Hello");														
insert(int offset, String str)	sb.insert(1, "Java");														
delete(int start, int end):	sb.delete(1, 3);														
deleteCharAt(int index):	sb.deleteCharAt(2);														
reverse()	sb.reverse();														
setCharAt(int index, char ch):	sb.setCharAt(1, 'J');														
IndexOf, lastIndexOf	indexOf(String.valueOf('c'))														

Constructors

30 July 2018 11:15

Constructors are used to initialize the object's state. Like methods, a constructor also contains collection of statements(i.e. instructions) that are executed at time of Object creation. The purpose of the default constructor is to assign the default value to the objects. The java compiler creates a default constructor implicitly if there is no constructor in the class.

Car a=new Car();

Here **new Car()** is object and **a** is object reference variable of type **Car**.

- Every class has a constructor whether it's a normal class or an abstract class.
- Constructors are not methods and they don't have any return statement but **constructor returns current class instance**.
- Constructor can use any access specifier, they can be declared as private also. Private constructors are possible in java but their scope is within the class only.
- If you don't implement any constructor within the class, compiler will do it.
- **this() and super() should be the first statement in the constructor code.** If you don't mention them, compiler does it for you accordingly.
- Constructor overloading is possible but overriding is not possible.
- Constructors can not be inherited.
- **Interfaces do not have constructors.**
- A constructor can also invoke another constructor of the same class – By using **this()/super()**.
- **this(5)** inside another constructor will call a parameterized constructor and pass value 5 to it.
- A constructor in Java cannot be abstract, final, static and Synchronized.
- Types: Param, Non Param and Default Constructors

Abstraction

30 July 2018 17:45

Abstraction is a process of hiding the implementation details from the user, only the functionality will be provided to the user. In other words, the user will have the information on what the object does instead of how it does it.

Why WebDriver is an interface/ Meaning of Abstraction

The purpose of Selenium WebDriver to automate browsers but they do not know how each browser works internally. Every browser has their own logic to perform browser's actions such as Launching a browser, closing a browser, loading URL, handling different type of web elements. Same operations are performed in different ways by different browsers.

Without knowing internal working mechanisms of browsers, Selenium WebDriver developers can not write codes to perform actions. If they get to know the internal mechanism some how, it will be difficult to manage when internal mechanism changes and this should not break flow for other browsers. So to make it simple, a contract is placed between Selenium WebDriver developers and browsers and that contract is in form of WebDriver. A WebDriver consists of all related basic methods which could be performed on a browser.

Feature	Abstract Class	Interface
Definition	A class that may or may not contain abstract methods. It can provide a common base for related classes.	A blueprint of a class, defining a contract for what a class <i>must</i> do.
Methods	Can have both abstract and concrete (implemented) methods.	All methods are implicitly public and abstract (before Java 8).
Variables	Can have any type of variables (static, non-static, final, non-final).	All variables are implicitly public, static, and final (constants).
Constructors	Can have constructors (including parameterized and default).	Cannot have constructors.
Instantiation	Cannot be instantiated directly.	Cannot be instantiated directly.
Inheritance	Subclasses use the extends keyword to inherit.	Classes use the implements keyword to implement.
Multiple Inheritance	A class can inherit from only one abstract class.	A class can implement multiple interfaces.
Abstraction Level	Can provide partial abstraction (some methods implemented).	Provides total abstraction (all methods must be implemented).
Purpose	To provide a common base class and code reuse for related classes.	To define a contract or specification for classes to follow.
Use Case	When you have a hierarchy of related classes with some common behavior. Example: AbstractList class, it is inherited and implemented by ArrayList, LinkedList. Get is an abstract method from AbstractList class while ArrayList and LinkedList use Iterator<E> as it is.	When you want to define a role or capability that multiple unrelated classes can share.
Private Methods	Can have private methods.	Cannot have private methods.
Default Methods	Cannot have default methods (prior to Java 8).	Can have default methods (introduced in Java 8) to provide a default implementation.
Static Methods	Can have static methods.	Can have static methods (introduced in Java 8).
Fields	Can have instance fields that store the state of the object.	Cannot have instance fields; only constants.
Accessibility Modifiers	Methods and variables can have different access modifiers (public, protected, private).	All methods are implicitly public.

Encapsulation

30 July 2018 18:02

Definition:

- Wrapping data and methods into a single unit (a class).
- A protective shield preventing external access to data.

Technical Aspects:

- Data (variables) is hidden (private).
- Access to data is controlled through public methods (getters and setters).

Advantages:

- **Data Hiding:** Protects data from unintended changes.
- **Improved Maintenance:** Changes to internal implementation don't affect external code.
- **Modularity:** Breaks complex systems into manageable units.
- **Reusability:** Encapsulated code can be used in different parts of an application.
- **Flexibility:** Internal implementation can be changed without altering the external interface.
- **Simplified Testing & Debugging:** Individual components can be tested and debugged easily.
- **Increased Security:** Controls access to sensitive data.

Key takeaway: Encapsulation is about bundling data and methods together and controlling access to that data, leading to better code organization, maintainability, and security.

Inheritance

30 July 2018 15:46

Inheritance is an important pillar of OOP(Object Oriented Programming). It is the mechanism in java by which one class is allowed to inherit the features(fields and methods) of another class.

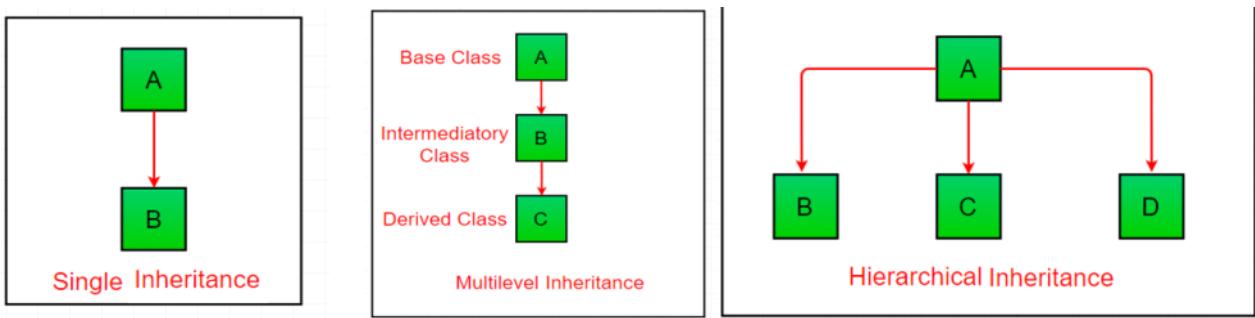
Important terminology:

- **Super Class:** The class whose features are inherited is known as super class(or a base class or a parent class).
- **Sub Class:** The class that inherits the other class is known as sub class(or a derived class, extended class, or child class). The subclass can add its own fields and methods in addition to the superclass fields and methods.

Reusability: Inheritance supports the concept of “reusability”, i.e. when we want to create a new class and there is already a class that includes some of the code that we want, we can derive our new class from the existing class. By doing this, we are reusing the fields and methods of the existing class.

Types of Inheritance in Java

- **Single Inheritance:** Cucumber Runner
- **Hierarchical:** Testbase and its sub classes



Java does not support multiple inheritance with classes. In java, we can achieve multiple inheritance only through Interfaces.

```
public class ClassD extends ClassA implements InterfaceB,InterfaceC
```

Overloading and Overriding

30 July 2018 18:09

Overloading	Method Overloading is a feature that allows a class to have more than one method having the same name, if their argument lists are different. It can't be achieved via different return type. Private, static and final can't be overridden. The access modifier of overriding class cannot be more restricting than overridden method. It's also called compile time polymorphism or static binding.
Example	<p>Function Overloading in Selenium:</p> <pre>driver.switchTo().frame("frameName/ID"); driver.switchTo().frame(1); driver.switchTo().frame(webElement); Action.click(); Action.click(WebElement);</pre> <p>Constructor Overloading in Selenium:</p> <pre>WebDriver driver = new ChromeDriver(); WebDriver driver = new ChromeDriver(options);</pre>
Overriding	Overriding is a feature that allows a subclass or child class to provide a specific implementation of a method that is already provided by one of its super-classes or parent classes. When a method in a subclass has the same name, same parameters or signature and same return type(or sub-type) as a method in its super-class, then the method in the subclass is said to override the method in the super-class. ChromeDriver class overrides get() of RemoteWebDriver class.
Example	<p>It is the type of the object being referred to (not the type of the reference variable) that determines which version of an overridden method will be executed.</p> <pre>class Animal { public void move() { System.out.println("Animals can move"); } } class Dog extends Animal { public void move() { System.out.println("Dogs can walk and run"); } public void bark() { System.out.println("Dogs can bark"); } } public class TestDog { public static void main(String args[]) { Animal a = new Animal(); // Animal reference and object Animal b = new Dog(); // Animal reference but Dog object (Top/Upcasting) a.move(); // runs the method in Animal class b.move(); // runs the method in Dog class (Dynamic Polymorphism) b.bark(); // Error } }</pre> <p>Compile time error as bark is missing in animal. In compile time, the check is made on the reference type. However, in the runtime, JVM figures out the object type and would run the method that belongs to that particular object.</p> <pre>TestDog.java:26: error: cannot find symbol b.bark(); ^ symbol: method bark() location: variable b of type Animal 1 error</pre>
Shadowing/Hiding of static functions	<p>This is not overloading.</p> <p>If a subclass defines a static method with the same signature as a static method in the super class then the method in the subclass hides the one in the super class. This happens because the static method is resolved at the compile time. Static methods bind during the compile time using the type of reference not a type of object.</p> <pre>// file name: Main.java</pre>

```
class A {  
    static void fun() {  
        System.out.println("A.fun()");  
    }  
}  
  
class B extends A {  
    static void fun() {  
        System.out.println("B.fun()");  
    }  
}  
  
public class Main {  
    public static void main(String args[]) {  
        A a = new B();  
        a.fun(); // prints A.fun()  
    }  
}
```

If we make both A.fun() and B.fun() as non-static then the above program would print "B.fun()".

Comparator & Comparable

03 December 2023 18:24

Comparator and Comparable are interfaces used for sorting objects in collections.

Comparator(Multi): It is an interface in java used to order the objects of user defined classes. It is useful when we need to provide multiple ordering options of the objects.

Comparable: It is an interface in java that is used to define the natural ordering of objects for a user defined class. it is part of Java.lang package and it provides a compareTo method to compare instance of the class. A class has to implement a Comparable interface to define its natural ordering.

Feature	Comparator	Comparable
Purpose	Defines custom ordering logic for objects.	Defines the natural ordering of objects.
Implementation	Implemented as a separate class or using lambda expressions.	Implemented by the class of objects being compared.
Method	compare(Object o1, Object o2): Compares two objects and returns a negative, zero, or positive integer.	compareTo(Object o): Compares the current object with another object and returns a negative, zero, or positive integer.
Sorting Behavior	Provides multiple sorting sequences based on different criteria.	Provides a single, inherent sorting sequence.
Class Modification	Does not modify the original class; ordering is external.	Modifies the original class to define its natural ordering.
Package	java.util	java.lang
Usage Scenario	When you need to sort objects based on criteria other than their natural ordering, or when the objects don't implement Comparable.	When you want to define the default, natural way objects of a class should be ordered.
Sorting Method Call	Collections.sort(List, Comparator) or Arrays.sort(array, Comparator)	Collections.sort(List) or Arrays.sort(array)
Flexibility	Highly flexible; allows for on-the-fly sorting logic changes.	Less flexible; requires modifying the class for any change in ordering.
Reusability	Can be reused for different collections of the same type or different types (if the logic applies).	Tied to the specific class that implements it.
Impact on Existing Code	Adding a new Comparator does not affect existing code that relies on the natural ordering.	Implementing Comparable affects all code that relies on the class's ordering.
Use with Anonymous Classes/Lambdas	Often used with anonymous classes or lambda expressions for concise, inline sorting logic.	Typically implemented directly within the class definition.
Null Handling	Can handle null values with custom logic within the compare() method.	compareTo() may throw NullPointerException if not handled within the implementation.
Example	<pre>@Data @AllArgsConstructor public class Student { private int rollNo; private String name; private Double averagePercentage; public static Comparator<Student> rollComparator=(s1,s2)-> Double.compare(s1.rollNo,s2.rollNo); public static Comparator<Student> nameComparator=(s1,s2)-> s1.name.compareTo(s2.name); public static void main(String[] args) { List<Student> students = new ArrayList<>(); students.add(new Student(3, "Alice", 85.5)); students.add(new Student(4, "Dick", 55.5)); students.add(new Student(1, "Bob", 91.2)); students.add(new Student(2, "Charlie", 78.9)); //students.sort(rollComparator); students.sort(nameComparator); for (Student student : students) { System.out.println(student); } } }</pre>	<pre>@Data @AllArgsConstructor public class Student implements Comparable<Student> { private int rollNo; private String name; private Double averagePercentage; @Override public int compareTo(Student otherStudent) { //return Integer.compare(this.rollNo, otherStudent.rollNo); //return this.name.compareTo(otherStudent.name); return Double.compare(this.averagePercentage, otherStudent.averagePercentage); } public static void main(String[] args) { List<Student> students = new ArrayList<>(); students.add(new Student(3, "Alice", 85.5)); students.add(new Student(1, "Bob", 91.2)); students.add(new Student(2, "Charlie", 78.9)); Collections.sort(students); for (Student student : students) { System.out.println(student); } } }</pre>
	Compare Strings public static Comparator<Student> nameComparator=(s1,s2)->	Compare Strings return this.name.compareTo(otherStudent.name)

	s1.name.compareTo(s2.name);		;
Compare Numbers	public static Comparator<Student> rollComparator=(s1,s2)-> Double.compare(s1.rollNo,s2.rollNo);	Compare Numbers	return Double.compare(this.averagePercentage, otherStudent.averagePercentage);
<ul style="list-style-type: none"> Achieve Sorting when section is same for 2 students: <code>studentList.sort(sectionComparator.thenComparing(nameComparator));</code> Sorting with reverse order: <code>students.stream().sorted(nameComparator.reversed()).forEach(System.out::println);</code> 			

Functional Programming

08 January 2021 10:16

Lambda	<p>Lambda provides a clear and concise way to represent one method interface using an expression. It is very useful in collection library. It helps to iterate, filter and extract data from collection.</p> <p>A lambda expression(another way to define a method) is a short block of code which takes in parameters and returns a value. Lambda expressions are similar to methods, but they do not need a name, no access modifier and they can be implemented right in the body of a method without explicitly stating return type.</p> <p>Lambda Restriction on Local Variables</p> <ul style="list-style-type: none">1. Local variable cannot be used as lambda parameter2. Local variables cannot be reassigned a value inside lambda(Local variables are effectively final inside lamdba)
Stream	<p>Java provides a new additional package in Java 8 called java.util.stream. This package consists of classes, interfaces and enum to allows functional-style operations on the elements. You can use stream by importing java.util.stream package.</p> <p>Stream provides following features:</p> <ul style="list-style-type: none">• Stream does not store elements. It simply conveys elements from a source such as a data structure, an array, or an I/O channel, through a pipeline of computational operations.• Stream is functional in nature. Operations performed on a stream does not modify its source. For example, filtering a Stream obtained from a collection produces a new Stream without the filtered elements, rather than removing elements from the source collection.• Stream is lazy and evaluates code only when required.• The elements of a stream are only visited once during the life of a stream. Like an Iterator, a new stream must be generated to revisit the same elements of the source.• You can use stream to filter, collect, print, and convert from one data structure to other etc.• parallelStream() enables parallel execution of stream operations. Instead of processing elements sequentially, tasks are divided into multiple smaller tasks that can run concurrently on different threads, leveraging the multi-core architecture of modern CPUs.
Method Reference	<p>Each time when you are using lambda expression to just referring a method, you can replace your lambda expression with method reference.</p> <p>System.out::println Objects::isNull String::toUpperCase String::concat String::replaceAll list::add Cat::new//constructor reference</p>
Intermediate Operations	<p>Steam can have multiple intermediate operations and only 1 terminal operation as intermediate returns new stream while terminal consumes stream.</p> <ul style="list-style-type: none">• Filter• Map• Limit• Distinct• Sorted
Terminal Operations	<ul style="list-style-type: none">• Foreach• Collect• Count: returns long• Min: returns optional• Max: returns optional• Findany: returns optional• Findfirst: returns optional• Anymatch: returns boolean• Nonematch: returns boolean• reduce
Flatmap	<p>The flatMap method is similar to the map function in Java Streams, but with a significant difference: while map transforms each element in the stream into another form (one-to-one mapping), flatMap transforms each element into a stream of elements and then flattens these streams into a single stream. This is particularly useful when each element in the stream represents a collection (like a list or an array).</p> <p>Flatmap returns stream of objects</p> <pre>List<String> studentActivities = StudentDataBase.getAllStudents() .stream() .map(Student::getActivities) // Stream<List<String>> .flatMap(List::stream) // <Stream<String> .collect(Collectors.toList());</pre>

	Here flatmap is required because map is returning List of String, map cannot capture that hence flatmap is required to flatten it to string.								
Optional	<p>Java introduced a new class Optional in jdk8. It is a public final class and used to deal with NullPointerException in Java application. You must import java.util package to use this class. It provides methods which are used to check the presence of value for particular variable.</p> <ul style="list-style-type: none"> • Optional.ofNullable(value): Use when the value might be null. Returns an empty Optional if value is null. • Optional.of(value): Use when you are sure the value is non-null. Throws NullPointerException if value is null. • Optional.empty(): Use to represent an absent value. • Optional.orElse(value): Returns the provided value if the Optional is empty. • Optional.orElseGet(Supplier): Returns the value supplied by the Supplier if the Optional is empty. • Optional.orElseThrow(Supplier): Throws an exception provided by the Supplier if the Optional is empty. • The Optional.isPresent() method checks if there is a value inside the Optional. It returns true if the Optional contains a value, otherwise false. • The Optional.ifPresent() method performs an action if a value is present. It takes a Consumer as an argument, which is a functional interface that accepts a value and returns no result. 								
Interface Changes	<ul style="list-style-type: none"> • Prior to Java 8, interfaces could only declare methods without providing implementations. • Any class implementing an interface had to provide concrete implementations for all the interface's methods. • Impact: Adding new methods to an interface would break existing implementations, making it challenging for library creators to evolve interfaces without causing issues. <p>Java 8 introduced default and static methods in interfaces to address these limitations</p> <table border="1"> <tr> <td>Default Methods</td><td>Interfaces can now have method implementations using the default keyword. Default methods allow new methods to be added to existing interfaces without breaking existing implementations. If all or most implementing classes will use a similar implementation, use a default method to avoid code duplication also they can be overridden by implementing classes. Example: The List interface includes a sort() method implemented with the default keyword.</td></tr> <tr> <td>Static Methods</td><td>Interfaces can include static methods. These methods belong to the interface itself and can be called using the interface name. Static methods in interfaces allow for utility methods/helper functions to be defined within interfaces, promoting code reuse but not tied to a specific instance. Unlike default methods, static methods in interfaces cannot be overridden by implementing classes Example: Collections.emptyList(); This static method from the Collections class in Java returns an unmodifiable empty list.</td></tr> <tr> <td>Functional Interfaces</td><td>These are interfaces with a single abstract method and serve as the basis for lambda expressions and method references. The @FunctionalInterface annotation can be used to mark an interface as functional</td></tr> <tr> <td>Method References</td><td> <ul style="list-style-type: none"> • This feature provides a concise way to refer to existing methods by their names, improving code readability and reuse • Optional Return Types: The Optional<T> class was introduced to handle nullability more expressively, reducing the risk of null pointer exceptions </td></tr> </table>	Default Methods	Interfaces can now have method implementations using the default keyword. Default methods allow new methods to be added to existing interfaces without breaking existing implementations. If all or most implementing classes will use a similar implementation, use a default method to avoid code duplication also they can be overridden by implementing classes. Example: The List interface includes a sort() method implemented with the default keyword.	Static Methods	Interfaces can include static methods. These methods belong to the interface itself and can be called using the interface name. Static methods in interfaces allow for utility methods/helper functions to be defined within interfaces, promoting code reuse but not tied to a specific instance. Unlike default methods, static methods in interfaces cannot be overridden by implementing classes Example: Collections.emptyList(); This static method from the Collections class in Java returns an unmodifiable empty list.	Functional Interfaces	These are interfaces with a single abstract method and serve as the basis for lambda expressions and method references. The @FunctionalInterface annotation can be used to mark an interface as functional	Method References	<ul style="list-style-type: none"> • This feature provides a concise way to refer to existing methods by their names, improving code readability and reuse • Optional Return Types: The Optional<T> class was introduced to handle nullability more expressively, reducing the risk of null pointer exceptions
Default Methods	Interfaces can now have method implementations using the default keyword. Default methods allow new methods to be added to existing interfaces without breaking existing implementations. If all or most implementing classes will use a similar implementation, use a default method to avoid code duplication also they can be overridden by implementing classes. Example: The List interface includes a sort() method implemented with the default keyword.								
Static Methods	Interfaces can include static methods. These methods belong to the interface itself and can be called using the interface name. Static methods in interfaces allow for utility methods/helper functions to be defined within interfaces, promoting code reuse but not tied to a specific instance. Unlike default methods, static methods in interfaces cannot be overridden by implementing classes Example: Collections.emptyList(); This static method from the Collections class in Java returns an unmodifiable empty list.								
Functional Interfaces	These are interfaces with a single abstract method and serve as the basis for lambda expressions and method references. The @FunctionalInterface annotation can be used to mark an interface as functional								
Method References	<ul style="list-style-type: none"> • This feature provides a concise way to refer to existing methods by their names, improving code readability and reuse • Optional Return Types: The Optional<T> class was introduced to handle nullability more expressively, reducing the risk of null pointer exceptions 								
Date/Time	<p>Java 8 introduced a new Date and Time API (java.time package) to address the limitations and design flaws of the old java.util.Date and java.util.Calendar classes</p> <pre> LocalDate today = LocalDate.now(); //2024-08-25 LocalDate birthday = LocalDate.of(1990, 8, 25); LocalTime now = LocalTime.now(); //14:30:00 LocalTime meetingTime = LocalTime.of(10, 30); LocalDateTime now = LocalDateTime.now(); //2024-08-25T14:30:00 LocalDateTime event = LocalDateTime.of(2024, 8, 25, 14, 30); </pre>								

Stream Notes

26 December 2023 11:50

Name	Example						
forEach	Performs an action for each element of the stream. <code>List<String> strings = Arrays.asList("apple", "banana", "orange"); strings.stream().forEach(System.out::println);</code>						
.filter	Returns a stream consisting of the elements that match the given predicate. <code>List<String> filteredList = strings.stream().filter(s -> s.startsWith("a")).collect(Collectors.toList());</code>						
.map	Transforms each element using the provided function and replaces it in the stream <code>List<Integer> lengths = strings.stream().map(String::length).collect(Collectors.toList());</code>						
.boxed()	Converts a primitive stream into a stream of wrapper objects. <code>Stream<Integer> boxedStream = IntStream.of(numArray).boxed();</code>						
mapToInt	Converts a stream of objects into an IntStream of primitive int values. <code>IntStream intStreamAgain = Stream.of(1, 2, 3, 4).mapToInt(Integer::intValue);</code>						
mapToObj	<code>mapToObj()</code> is used to convert a primitive stream (like IntStream, DoubleStream, LongStream) to a stream of objects (<code>Stream<U></code>). <code>IntStream intStream = IntStream.of(1, 2, 3, 4); Stream<String> stringStream = intStream.mapToObj(Integer::toString); // Converts IntStream to Stream<String> stringStream.forEach(System.out::println); // Prints "1" "2" "3" "4"</code>						
ArrayList to int Array	<code>arrayList.stream().mapToInt(i->i).toArray()</code>						
ArrayList to String Array	<code>al.toArray(new String[0]);</code>						
Integer Array to Array List	<code>ArrayList<Integer> arrayList = new ArrayList<>(Arrays.asList(intArray));</code>						
String array to ArrayList	<code>ArrayList<String> arrayList = new ArrayList<>(Arrays.asList(strArray));</code>						
.flatMap	Flattens the stream of collections into a single stream. <code>List<List<String>> nestedList = Arrays.asList(Arrays.asList("apple", "orange"), Arrays.asList("banana")); List<String> flatList = nestedList.stream().flatMap(Collection::stream).collect(Collectors.toList());</code>						
.distinct	Returns a stream consisting of distinct elements. <code>List<Integer> distinctNumbers = numbers.stream().distinct().collect(Collectors.toList());</code>						
.sorted	Returns a stream sorted according to the natural order or using a custom comparator. <code>strings.stream().sorted().collect(Collectors.toList()); strings.stream().sorted(Comparator.reverseOrder()).collect(Collectors.toList());</code> //Sort with String Length <code>strings.stream().sorted(Comparator.comparing(x->s.length())).collect(Collectors.toList());</code>						
.reduce	Performs a reduction on the elements of the stream using an associative accumulation function. <table border="1" style="width: 100%; border-collapse: collapse;"> <thead> <tr> <th style="text-align: center;">Return Type</th> <th style="text-align: center;">Code</th> </tr> </thead> <tbody> <tr> <td style="text-align: center;">Optional</td> <td><code>Optional<Integer> sum = numbers.stream().reduce(Integer::sum); Optional<Integer> optProduct=myList.stream().reduce((a,b)->a*b); System.out.println("Opt Product is "+optProduct.get());</code></td> </tr> <tr> <td style="text-align: center;">Integer</td> <td><code>int sum=Arrays.stream(array).reduce(0, (a, b) -> a + b); int sum = Arrays.stream(classPoints).sum(); Int product=Arrays.stream(array).reduce(1, (a, b) -> a * b); myList.stream().reduce((a, b) -> a > b ? a : b).orElse(Integer.MIN_VALUE);</code></td> </tr> </tbody> </table>	Return Type	Code	Optional	<code>Optional<Integer> sum = numbers.stream().reduce(Integer::sum); Optional<Integer> optProduct=myList.stream().reduce((a,b)->a*b); System.out.println("Opt Product is "+optProduct.get());</code>	Integer	<code>int sum=Arrays.stream(array).reduce(0, (a, b) -> a + b); int sum = Arrays.stream(classPoints).sum(); Int product=Arrays.stream(array).reduce(1, (a, b) -> a * b); myList.stream().reduce((a, b) -> a > b ? a : b).orElse(Integer.MIN_VALUE);</code>
Return Type	Code						
Optional	<code>Optional<Integer> sum = numbers.stream().reduce(Integer::sum); Optional<Integer> optProduct=myList.stream().reduce((a,b)->a*b); System.out.println("Opt Product is "+optProduct.get());</code>						
Integer	<code>int sum=Arrays.stream(array).reduce(0, (a, b) -> a + b); int sum = Arrays.stream(classPoints).sum(); Int product=Arrays.stream(array).reduce(1, (a, b) -> a * b); myList.stream().reduce((a, b) -> a > b ? a : b).orElse(Integer.MIN_VALUE);</code>						
.count	Returns the count of elements in the stream. <code>long count = strings.stream().count();</code>						
.allMatch	Checks whether all elements of the stream match the provided predicate. <code>boolean allEven = numbers.stream().allMatch(n -> n % 2 == 0);</code>						
.noneMatch	Checks whether no elements of the stream match the provided predicate. <code>boolean noneEven = numbers.stream().noneMatch(n -> n % 2 == 0);</code>						

.anyMatch	Checks whether any elements of the stream match the provided predicate. Returns true if at least one element matches the predicate; otherwise, returns false. boolean anyEven = numbers.stream().anyMatch(n -> n % 2 == 0);
.findAny	Returns an Optional containing any element of the stream. We also have findFirst List<String> names = Arrays.asList("Saurav", "John", "Doe", "Jane", "Smith"); Optional<String> anyName = names.stream().filter(name -> name.startsWith("J")) .findAny(); anyName.ifPresent(System.out::println);
.skip	Returns a stream with the first n elements skipped. List<Integer> afterSkip = numbers.stream().skip(2).collect(Collectors.toList());
.limit	Returns a stream truncated to be no longer than a specified size. List<Integer> limited = numbers.stream().limit(3).collect(Collectors.toList());
.peek	List<String> peekedList = strings.stream().peek(System.out::println).collect(Collectors.toList());
.min/.max	Finds the minimum and maximum element of the stream. Optional<Integer> min = numbers.stream().min(Integer::compareTo); int min = arrayList.stream().min(Integer::compareTo).orElse(Integer.MAX_VALUE); Optional<Integer> max = numbers.stream().max(Integer::compareTo); int min = Arrays.stream(arr).min().getAsInt(); int max = Arrays.stream(arr).max().getAsInt();
Collectors.toMap	Collects the elements of the stream into a Map. List<String> strings = Arrays.asList("apple", "banana", "orange"); Map<Integer, String> lengthToWordMap = strings.stream().collect(Collectors.toMap(String::length, Function.identity()));
.collect()	List<String> strings = Arrays.asList("apple", "banana", "orange"); LinkedList<String> linkedList = strings.stream().collect(Collectors.toCollection(LinkedList::new)); .collect(Collectors.toList()) .collect(Collectors.toSet()) .collect(Collectors.joining()) .collect(Collectors.joining(",")) .collect(counting()) Arrays.stream(phrase.split(" ")) .map(i -> i.substring(0, 1).toUpperCase() + i.substring(1)) .collect(Collectors.joining(" ")); ListToMap List<Integer> list=new ArrayList<>(Arrays.asList(1,2,3)); Map<Integer, Integer> listToMap=list.stream().collect(Collectors.toMap(val->val, val->val*val));
forEachOrdered	Performs an action for each element of the stream in the encounter order. strings.stream().forEachOrdered(System.out::println);
Switch	Until Java 7 only integers could be used in switch statement. Int value=5; Switch(value){ Case 1: sout; Break: Case 2: Default: } Java 8 added string and enum as case statements Java 12 introduced switch expression return switch (op) { case "+" -> v1 + v2; case "-" -> v1 - v2; case "*" -> v1 * v2; case "/" -> { if (v2 == 0) { throw new ArithmeticException("Division by zero is not allowed.");

```

        }
        yield v1 / v2;
    }
    default -> throw new IllegalArgumentException("Invalid operation: " + op);
};


```

Switch expression can have multiple case statements
 Return switch (day){
 Case "monday", "Tuesday" ->"weekdays";
};

AtomicInteger	AtomicInteger count = new AtomicInteger() This line initializes an AtomicInteger called count with an initial value of 0. AtomicInteger is used to ensure thread safety when incrementing the counter inside the stream. count.getAndIncrement(): retrieves the current value of the counter and then increments it
Stream of Characters	char[] charArray = {'a', 'b', 'c', 'd'}; IntStream charStream = IntStream.range(0, charArray.length).map(i -> charArray[i]); charStream.mapToObj(c -> (char)c).forEach(System.out::println);
Stream of String	Stream.of(s1.concat(s2).split("")) .sorted().distinct() .collect(Collectors.joining()); var chars = str.split(""); return IntStream.range(0, chars.length).mapToObj(i -> chars[i].toUpperCase() + chars[i].toLowerCase().repeat(i)) .collect(Collectors.joining("-")); String input = "sumitM28"; input.chars().filter(Character::isLowerCase).forEach(System.out::print); Chars() returns a stream of int
Stream of Array	int[] arr = {1, 23, 123, 45, 134}; Arrays.stream(arr).filter(num -> String.valueOf(num).startsWith("1")).forEach(System.out::println);
Stream of Map	Map<String, Integer> filteredMap = sampleMap.entrySet() .stream() .filter(entry -> entry.getValue() > 20) // Filter entries with value > 20 .collect(Collectors.toMap(Map.Entry::getKey, Map.Entry::getValue));
Print HashMap	map.forEach((key, value) -> System.out.println(key + " -> " + value)); for (Map.Entry<Integer, Integer> entry : countMap.entrySet()) { if (entry.getValue() % 2 != 0) { return entry.getKey(); } }
Intstream	• IntStream operates on primitive int values directly rather than boxed Integer objects. This reduces memory overhead because it avoids the cost of autoboxing (converting primitives to their wrapper types). • IntStream can be parallelized easily using the .parallel() method. • IntStream provides specialized reduction operations that are more efficient than their generic counterparts. Examples include sum(), average(), min(), and max(), which operate directly on int values without the overhead of boxing. • IntStream includes methods that are tailored specifically for dealing with int primitives, like range() and rangeClosed(), which generate streams of numbers over specified ranges. • Intstream.range(1,50)->1-49 • Intstream.rangeClosed(1,50)->1-50
getOrDefault	If the character is already in the map, increment its count, otherwise add it with a count of 1 charCountMap.put(c, charCountMap.getOrDefault(c, 0) + 1);
Parallel Stream	When to Use Parallel Stream <ul style="list-style-type: none"> • Large datasets: Parallelism shines with more elements to process. • CPU-intensive operations: Tasks that require computation benefit from multi-core processing. • Independent operations: Each task should be self-contained without shared mutable state. When Not to Use Parallel Stream <ul style="list-style-type: none"> • Small datasets: Thread overhead may result in slower execution. • I/O-bound operations: These don't benefit from CPU parallelism and may block threads. • Stateful operations: Operations like sorting that rely on shared or mutable state can yield incorrect or unpredictable results.

Solid Principles

18 February 2024 08:36

SRP	<p>Every software component(Class, method or module) should have only one Responsibility(Only 1 Reason to Change).</p> <ul style="list-style-type: none"> Cohesion: Degree to which various components of software are related. Bifurcate closely related method to increase cohesion. Coupling: Level of interdependency between software Components. Aim for High Cohesion in a component and loose coupling. Example: Page Classes <table border="1"> <thead> <tr> <th>Low Cohesion</th><th>High Cohesion</th></tr> </thead> <tbody> <tr> <td> <pre>public class Square { int side = 5; public int calculateArea() { return side * side; } public int calculatePerimeter() { return side * 4; } public void draw() { if (highResolutionMonitor) { // Render a high resolution image of a square } else { // Render a normal image of a square } } public void rotate(int degree) { // Rotate the image of the square clockwise to // the required degree and re-render } }</pre> </td><td> <pre>public class Square { int side = 5; public int calculateArea() { return side * side; } public int calculatePerimeter() { return side * 4; } public void draw() { if (highResolutionMonitor) { // Render a high resolution image of a square } else { // Render a normal image of a square } } public void rotate(int degree) { // Rotate the image of the square clockwise to // the required degree and re-render } }</pre> </td></tr> </tbody> </table>	Low Cohesion	High Cohesion	<pre>public class Square { int side = 5; public int calculateArea() { return side * side; } public int calculatePerimeter() { return side * 4; } public void draw() { if (highResolutionMonitor) { // Render a high resolution image of a square } else { // Render a normal image of a square } } public void rotate(int degree) { // Rotate the image of the square clockwise to // the required degree and re-render } }</pre>	<pre>public class Square { int side = 5; public int calculateArea() { return side * side; } public int calculatePerimeter() { return side * 4; } public void draw() { if (highResolutionMonitor) { // Render a high resolution image of a square } else { // Render a normal image of a square } } public void rotate(int degree) { // Rotate the image of the square clockwise to // the required degree and re-render } }</pre>	
Low Cohesion	High Cohesion					
<pre>public class Square { int side = 5; public int calculateArea() { return side * side; } public int calculatePerimeter() { return side * 4; } public void draw() { if (highResolutionMonitor) { // Render a high resolution image of a square } else { // Render a normal image of a square } } public void rotate(int degree) { // Rotate the image of the square clockwise to // the required degree and re-render } }</pre>	<pre>public class Square { int side = 5; public int calculateArea() { return side * side; } public int calculatePerimeter() { return side * 4; } public void draw() { if (highResolutionMonitor) { // Render a high resolution image of a square } else { // Render a normal image of a square } } public void rotate(int degree) { // Rotate the image of the square clockwise to // the required degree and re-render } }</pre>					
<table border="1"> <thead> <tr> <th>Tightly Coupled</th><th>Loosely Coupled</th></tr> </thead> <tbody> <tr> <td> <pre>public class Student { private String studentId; private Date studentDOB; private String address; public void save() { // Serialize object into a string representation String objectStr = MyUtils.serializeToString(this); Connection connection = null; Statement stmt = null; try { Class.forName("com.mysql.jdbc.Driver"); connection = DriverManager.getConnection("jdbc:mysql://localhost:3306/ MyDB", "root", "password"); stmt = connection.createStatement(); stmt.execute("INSERT INTO STUDENT VALUES (" + objectStr + ")"); } catch (Exception e) { e.printStackTrace(); } } public String getStudentId() { return studentId; } public void setStudentId(String studentId) { this.studentId = studentId; } }</pre> </td><td> <pre>public class Student { private String studentId; private Date studentDOB; private String address; public void save() { new StudentRepository().save(this); } public String getStudentId() { return studentId; } public void setStudentId(String studentId) { this.studentId = studentId; } } public class StudentRepository { public void save(Student student) { // Serialize object into a string representation String objectStr = MyUtils.serializeToString(student); Connection connection = null; Statement stmt = null; try { Class.forName("com.mysql.jdbc.Driver"); connection = DriverManager.getConnection("jdbc:mysql://localhost:3306/ MyDB", "root", "password"); stmt = connection.createStatement(); stmt.executeUpdate("INSERT INTO STUDENT VALUES (" + objectStr + ")"); } catch (Exception e) { e.printStackTrace(); } } }</pre> </td></tr> <tr> <td colspan="2"> <p>The Student class may have multiple reasons to change:</p> <ul style="list-style-type: none"> - Change in student ID format - Change in student name format - Change in database backend </td></tr> </tbody> </table>	Tightly Coupled	Loosely Coupled	<pre>public class Student { private String studentId; private Date studentDOB; private String address; public void save() { // Serialize object into a string representation String objectStr = MyUtils.serializeToString(this); Connection connection = null; Statement stmt = null; try { Class.forName("com.mysql.jdbc.Driver"); connection = DriverManager.getConnection("jdbc:mysql://localhost:3306/ MyDB", "root", "password"); stmt = connection.createStatement(); stmt.execute("INSERT INTO STUDENT VALUES (" + objectStr + ")"); } catch (Exception e) { e.printStackTrace(); } } public String getStudentId() { return studentId; } public void setStudentId(String studentId) { this.studentId = studentId; } }</pre>	<pre>public class Student { private String studentId; private Date studentDOB; private String address; public void save() { new StudentRepository().save(this); } public String getStudentId() { return studentId; } public void setStudentId(String studentId) { this.studentId = studentId; } } public class StudentRepository { public void save(Student student) { // Serialize object into a string representation String objectStr = MyUtils.serializeToString(student); Connection connection = null; Statement stmt = null; try { Class.forName("com.mysql.jdbc.Driver"); connection = DriverManager.getConnection("jdbc:mysql://localhost:3306/ MyDB", "root", "password"); stmt = connection.createStatement(); stmt.executeUpdate("INSERT INTO STUDENT VALUES (" + objectStr + ")"); } catch (Exception e) { e.printStackTrace(); } } }</pre>	<p>The Student class may have multiple reasons to change:</p> <ul style="list-style-type: none"> - Change in student ID format - Change in student name format - Change in database backend 	
Tightly Coupled	Loosely Coupled					
<pre>public class Student { private String studentId; private Date studentDOB; private String address; public void save() { // Serialize object into a string representation String objectStr = MyUtils.serializeToString(this); Connection connection = null; Statement stmt = null; try { Class.forName("com.mysql.jdbc.Driver"); connection = DriverManager.getConnection("jdbc:mysql://localhost:3306/ MyDB", "root", "password"); stmt = connection.createStatement(); stmt.execute("INSERT INTO STUDENT VALUES (" + objectStr + ")"); } catch (Exception e) { e.printStackTrace(); } } public String getStudentId() { return studentId; } public void setStudentId(String studentId) { this.studentId = studentId; } }</pre>	<pre>public class Student { private String studentId; private Date studentDOB; private String address; public void save() { new StudentRepository().save(this); } public String getStudentId() { return studentId; } public void setStudentId(String studentId) { this.studentId = studentId; } } public class StudentRepository { public void save(Student student) { // Serialize object into a string representation String objectStr = MyUtils.serializeToString(student); Connection connection = null; Statement stmt = null; try { Class.forName("com.mysql.jdbc.Driver"); connection = DriverManager.getConnection("jdbc:mysql://localhost:3306/ MyDB", "root", "password"); stmt = connection.createStatement(); stmt.executeUpdate("INSERT INTO STUDENT VALUES (" + objectStr + ")"); } catch (Exception e) { e.printStackTrace(); } } }</pre>					
<p>The Student class may have multiple reasons to change:</p> <ul style="list-style-type: none"> - Change in student ID format - Change in student name format - Change in database backend 						
OCP	<p>S/W Components should be closed for modification but open for extention. Playstation is closed for modification but can be extended by PS VR, headsets.</p> <ul style="list-style-type: none"> Classes should allow for extension through inheritance or composition. Existing code should not be modified when extending functionality. Promotes code reuse and minimizes the risk of introducing bugs in existing code. Example: Child class overriding parent class function <p>Open for Extension. Closed for Modification.</p> <p>i.e You should put new code in New Classes/Modules or Reuse existing code via Inheritance. Existing code should be modified only for Bug fixing.</p>					

If we want to add Vehicle Insurance product to our portfolio then InsurancePremiumDiscountCalculator class needs to be modified violating OCP principle.

```
public class InsurancePremiumDiscountCalculator {
    public int calculatePremiumDiscountPercent(HealthInsuranceCustomerProfile customer) {
        if (customer.isLoyalCustomer()) {
            return 20;
        }
        return 0;
    }
}
```

```
public class HealthInsuranceCustomerProfile {
    public boolean isLoyalCustomer() {
        return true; // or false
    }
}
```

Change to Accommodate Vehicle Insurance

```
public class InsurancePremiumDiscountCalculator {
    public int calculatePremiumDiscountPercent(HealthInsuranceCustomerProfile customer) {
        if (customer.isLoyalCustomer()) {
            return 20;
        }
        return 0;
    }

    public int calculatePremiumDiscountPercent(VehicleInsuranceCustomerProfile customer) {
        if (customer.isLoyalCustomer()) {
            return 20;
        }
        return 0;
    }
}
```

```
public class HealthInsuranceCustomerProfile {
    public boolean isLoyalCustomer() {
        return true; // or false
    }
}
```

```
public class VehicleInsuranceCustomerProfile {
    public boolean isLoyalCustomer() {
        return true; // or false
    }
}
```

OCP Design

```
public class InsurancePremiumDiscountCalculator {
    public int calculatePremiumDiscountPercent(CustomerProfile customer) {
        if (customer.isLoyalCustomer()) {
            return 20;
        }
        return 0;
    }
}
```

```
public interface CustomerProfile {
    public boolean isLoyalCustomer();
}
```

```
public class HealthInsuranceCustomerProfile
implements CustomerProfile {
    @Override
    public boolean isLoyalCustomer() {
        return true; // or false
    }
}
```

```
public class VehicleInsuranceCustomerProfile
implements CustomerProfile {
    @Override
    public boolean isLoyalCustomer() {
        return true; // or false
    }
}
```

```
// Without Open-Closed Principle
class Operation{
    double calculate(double a1, double a2, String operationType){
        switch(operationType){
            case "+":
                return a1 + a2;
            case "-":
                return a1 - a2;
            default:
                // Throw some exceptions
        }
        return 0;
    }
}
```

```
// With Open-Closed Principle
interface Operation{
    double calculate(double a1, double a2);
}
```

```
class AddOperation implements Operation{
    double calculate(double a1, double a2){
        return a1 + a2;
    }
}
```

```
class DivisionOperation implements Operation{
    double calculate(double a1, double a2){
        return a1/a2;
    }
}
```

Selenium Example: factory Design Pattern

```
public interface DriverManager {
    WebDriver getDriver();
}

public class ChromeDriverManager implements DriverManager {
    public WebDriver getDriver() {
        return new ChromeDriver();
    }
}

public class FirefoxDriverManager implements DriverManager {
```

```

public WebDriver getDriver() {
    return new FirefoxDriver();
}

// Add new browser support without modifying existing classes
public class EdgeDriverManager implements DriverManager {
    public WebDriver getDriver() {
        return new EdgeDriver();
    }
}

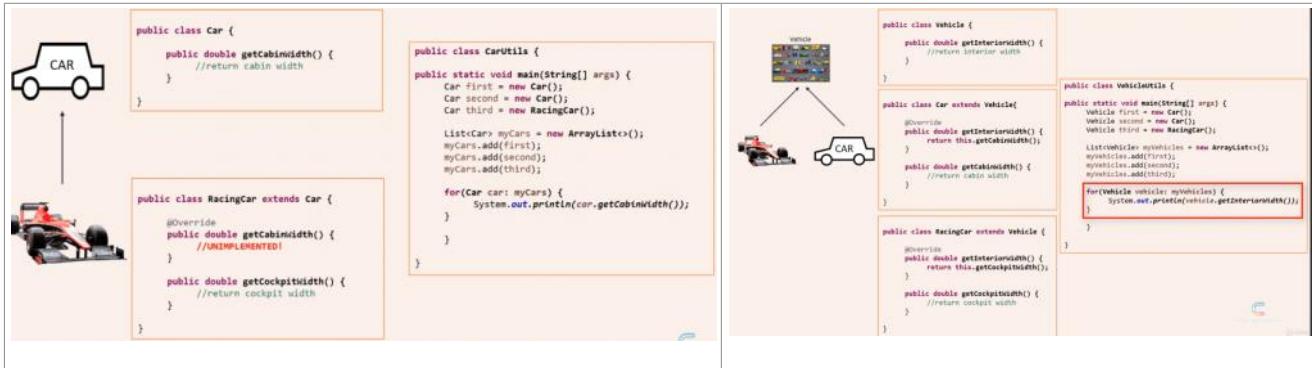
public class WebDriverFactory {
    public static DriverManager getManager(String browserType) {
        switch (browserType) {
            case "chrome":
                return new ChromeDriverManager();
            case "firefox":
                return new FirefoxDriverManager();
            case "edge":
                return new EdgeDriverManager(); // New functionality added
            default:
                throw new IllegalArgumentException("Invalid browser type");
        }
    }
}

```

- LSP** Objects of a superclass should be replaceable with objects of a subclass without affecting the correctness of the program.
- Subtypes must be substitutable for their base types.
 - The behavior of a subclass should not contradict the behavior of its superclass.
 - Example Strategy Pattern
 - In order to obey this law, the Child class should support all methods of Parents class.

Example: Ostrich **is-a** bird but is cannot fly so we cannot substitute ostrich's fly method with bird's fly method.

RacingCar **is-a** Car(inheritance) but getCabinWidth() of Car class cannot be called on RacingCar Object.

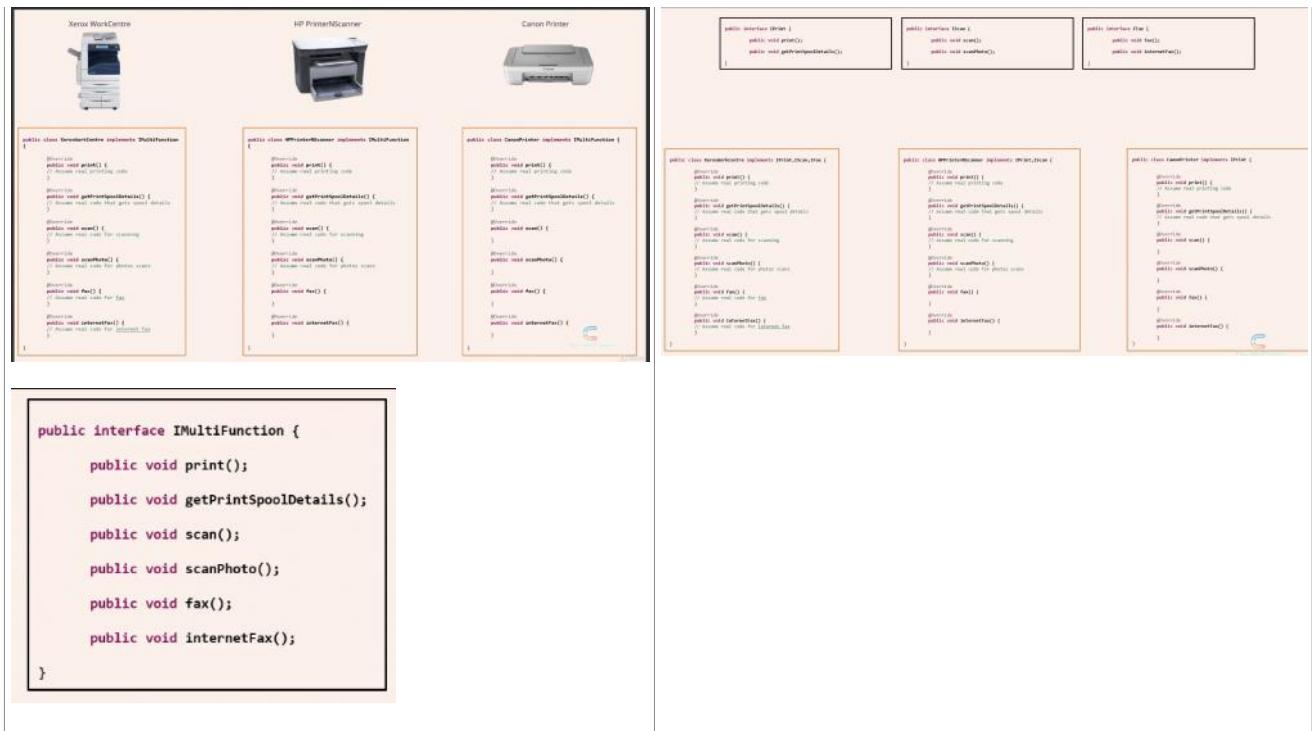


Selenium provides the WebDriver interface with multiple implementations like ChromeDriver, FirefoxDriver, EdgeDriver, etc. These implementations adhere to the LSP because they can replace the WebDriver reference without breaking functionality.

- ISP** The Interface Segregation Principle (ISP) states that no client should be forced to depend on methods it does not use.

In a real-world scenario, this principle can be illustrated with office equipment like printers, scanners, and fax machines. For instance, if we create an interface named IMultiFunction to represent a multi-function device, it may have methods for printing, scanning, and faxing. However, if we implement this interface for devices that cannot perform all these functions, we end up with blank method implementations, which violates the ISP.

This violation becomes problematic when other parts of the code, such as an employee portal application, rely on these interfaces. Developers may mistakenly assume that all methods are fully implemented, leading to potential errors and breaking of the code.



Selenium uses multiple interfaces to segregate responsibilities, allowing a class to implement only what it needs. For example:

- SearchContext: Defines methods for finding elements (findElement and findElements).
- WebDriver: Extends SearchContext and adds browser-related operations like get, navigate, and manage.
- TakesScreenshot: Used by drivers that support screenshots.
- JavascriptExecutor: Used by drivers that support JavaScript execution.

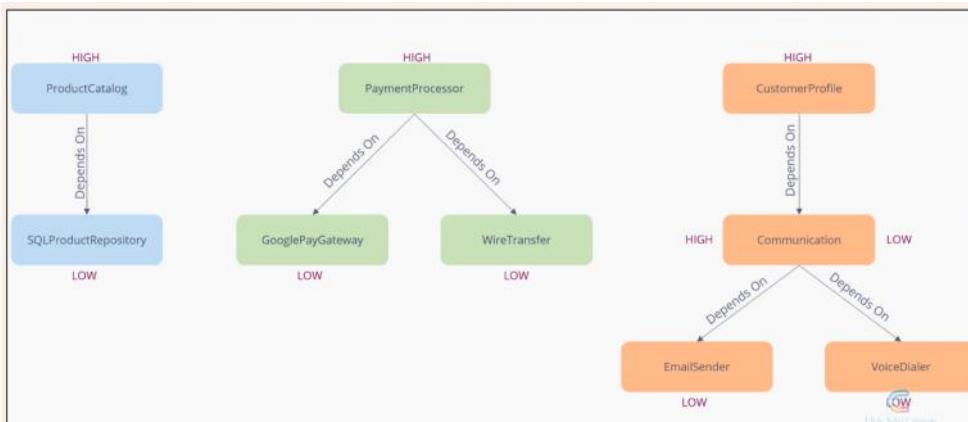
Not all drivers need to implement every interface.

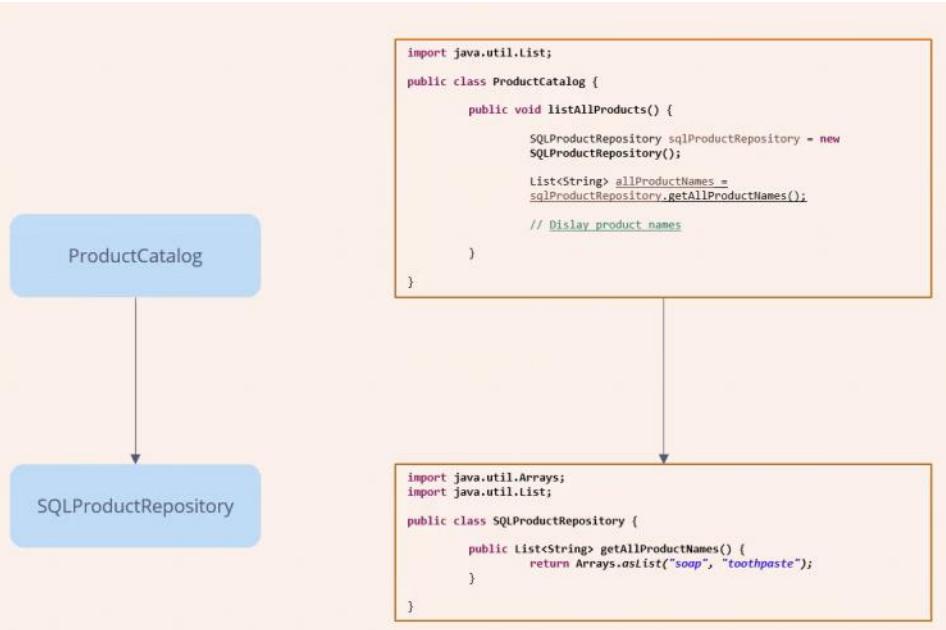
DIP The Dependency Inversion Principle emphasizes that our class should depend upon interfaces or abstract classes instead of Concrete class and functions.

Example: we use Webdriver driver =new chromedriver and not the other way around.

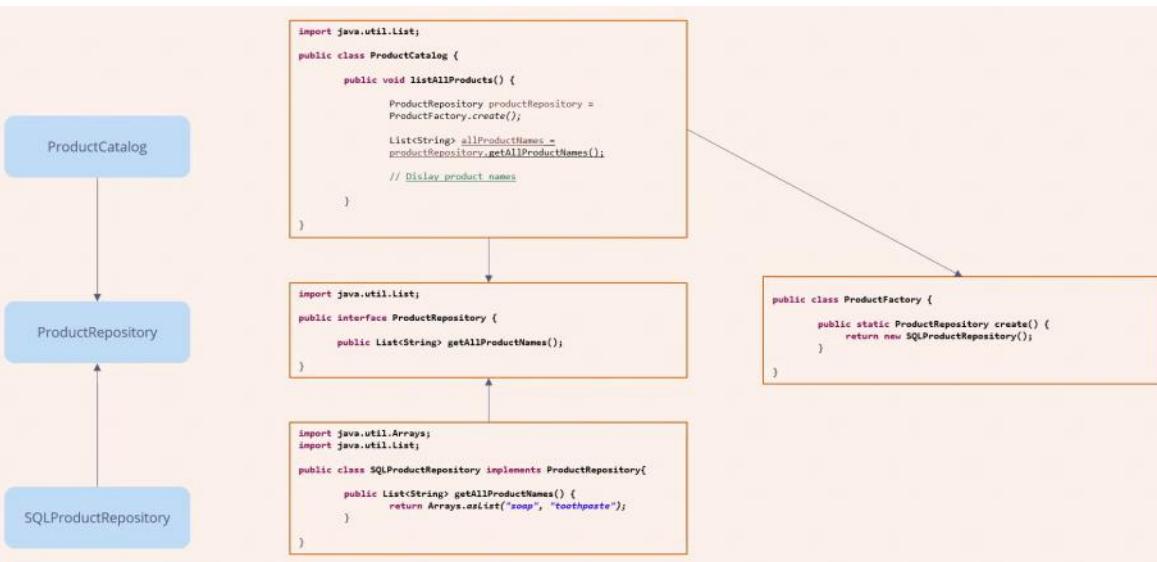
In a scenario where an eCommerce application comprises high-level modules like ProductCatalog, PaymentProcessor, and CustomerProfile, and low-level modules like SQLProductRepository and GooglePayGateway, dependencies must be carefully managed.

Initially, if high-level modules directly depend on low-level modules, it violates the DIP. For example, if ProductCatalog directly depends on SQLProductRepository, it creates tight coupling and violates the principle.





To address this violation, abstractions like interfaces are introduced. In the example, a `ProductRepository` interface is created, and `SQLProductRepository` implements it. A factory method is used to instantiate the concrete implementation, removing direct dependencies between high-level and low-level modules.



After restructuring, both high-level and low-level modules depend on abstractions, adhering to the DIP. Additionally, the principle requires that abstractions should not depend on details, leading to an inversion of dependency where details depend on abstractions.

Multi-Threading

30 March 2025 10:59

Concurrency	<ul style="list-style-type: none"> The illusion of multiple tasks running simultaneously (even on a single core). Achieved by rapidly switching between threads.
Parallelism	<ul style="list-style-type: none"> True simultaneous execution of tasks, possible only with multi-core CPUs. Boosts performance by distributing workload across multiple cores.
Process	<ul style="list-style-type: none"> An instance of a running application. <ul style="list-style-type: none"> Each process is isolated from others. Contains: <ul style="list-style-type: none"> Metadata – Process ID, file references. Code – Program instructions. Heap – Stores application data. Main Thread – Each process starts with at least one thread. Are isolated from each other, making context switches between processes slower.
Thread	<ul style="list-style-type: none"> A lightweight sub-unit of a process. <ul style="list-style-type: none"> Each thread has: <ul style="list-style-type: none"> Stack – Stores local variables and function calls. Instruction Pointer – Tracks the next instruction to execute. Threads: <ul style="list-style-type: none"> Consume fewer resources since they share code, heap, and file references. Context switches between threads in the same process are faster.
Context Switch(Thread Scheduler)	<p>Definition: The process of stopping one thread, saving its state, switching to another thread, and restoring its state.</p> <ul style="list-style-type: none"> Why is it Needed? <ul style="list-style-type: none"> Multiple threads compete for limited CPU cores. Even on multi-core CPUs, there are usually more threads than cores, requiring the OS to manage thread execution. <p>Why Are Context Switches Costly?</p> <ul style="list-style-type: none"> Threads rely on resources like: <ul style="list-style-type: none"> CPU registers CPU caches Kernel resources in memory During a context switch: <ul style="list-style-type: none"> The current thread's state (registers, cache data, etc.) is stored. The next thread's state is loaded back into the CPU. <p>Impact: Frequent context switches reduce productive work and increase overhead.</p>
Thrashing	Occurs when too many threads cause excessive context switching, leaving little CPU time for actual task execution.
Time Slicing	<ul style="list-style-type: none"> Time slicing is a CPU scheduling mechanism in which the CPU time is divided into small units called time slices (or time quanta). Each thread gets a fixed time slice to execute before the CPU switches to another thread. If a thread does not complete its execution within the allotted time slice, it is preempted and placed back in the queue. Java does not provide direct control over time slicing, as it depends on the OS and JVM implementation
Thread Lifecycle	<ul style="list-style-type: none"> NEW → Before start() Active <ul style="list-style-type: none"> RUNNABLE → Ready to run but waiting for CPU Running BLOCKED → Waiting for a lock <ul style="list-style-type: none"> WAITING → Indefinite waiting (needs notify()) TIMED_WAITING → Waiting for a fixed time (sleep(), wait(time)) TERMINATED → Completed execution or stopped
Basic Example	<pre>public class BasicExample { public static void main(String[] args) { Thread one = new Thread1(); Thread two = new Thread(new ThreadTwo()); Thread three = new Thread(() -> { for (int i = 0; i < 10; i++) { System.out.println("Thread Three : " + i); } }); one.start(); two.start(); three.start(); } } class Thread1 extends Thread { @Override public void run() { for (int i = 0; i < 10; i++) { System.out.println("T1 : " + i); } } }</pre>

	<pre> } } class ThreadTwo implements Runnable { @Override public void run() { for (int i = 0; i < 10; i++) { System.out.println("Thread Two : " + i); } } } </pre>																		
Join()	<p>The <code>join()</code> method in Java makes the calling thread wait until the specified thread completes its execution. It is used to ensure sequential execution when multiple threads are running. When a thread calls <code>t.join()</code>, it pauses execution until thread <code>t</code> completes.</p> <ul style="list-style-type: none"> • <code>join()</code> → Waits indefinitely until the thread finishes. • <code>join(millis)</code> → Waits for a maximum of <code>millis</code> milliseconds, then continues. 																		
User Vs Daemon Thread	<table border="1"> <thead> <tr> <th>Feature</th><th>User Thread</th><th>Daemon Thread</th></tr> </thead> <tbody> <tr> <td>Definition</td><td>A thread that executes user-defined tasks and keeps the JVM running until it completes.</td><td>A background thread that runs supportive tasks and automatically stops when all user threads finish.</td></tr> <tr> <td>JVM Behavior</td><td>JVM waits for all user threads to complete before shutting down.</td><td>JVM terminates daemon threads automatically when no user threads remain.</td></tr> <tr> <td>Purpose</td><td>Used for main application logic (e.g., handling requests, running tests).</td><td>Used for background tasks (e.g., garbage collection, logging, monitoring).</td></tr> <tr> <td>Lifecycle</td><td>Must complete its execution before JVM shuts down.</td><td>JVM does not wait for daemon threads to finish; they may be stopped abruptly.</td></tr> <tr> <td>Default Type</td><td>By default, all threads are User Threads unless explicitly set as Daemon.</td><td>Must be explicitly marked as Daemon using <code>setDaemon(true)</code>.</td></tr> </tbody> </table>	Feature	User Thread	Daemon Thread	Definition	A thread that executes user-defined tasks and keeps the JVM running until it completes.	A background thread that runs supportive tasks and automatically stops when all user threads finish.	JVM Behavior	JVM waits for all user threads to complete before shutting down.	JVM terminates daemon threads automatically when no user threads remain.	Purpose	Used for main application logic (e.g., handling requests, running tests).	Used for background tasks (e.g., garbage collection, logging, monitoring).	Lifecycle	Must complete its execution before JVM shuts down.	JVM does not wait for daemon threads to finish; they may be stopped abruptly.	Default Type	By default, all threads are User Threads unless explicitly set as Daemon.	Must be explicitly marked as Daemon using <code>setDaemon(true)</code> .
Feature	User Thread	Daemon Thread																	
Definition	A thread that executes user-defined tasks and keeps the JVM running until it completes.	A background thread that runs supportive tasks and automatically stops when all user threads finish.																	
JVM Behavior	JVM waits for all user threads to complete before shutting down.	JVM terminates daemon threads automatically when no user threads remain.																	
Purpose	Used for main application logic (e.g., handling requests, running tests).	Used for background tasks (e.g., garbage collection, logging, monitoring).																	
Lifecycle	Must complete its execution before JVM shuts down.	JVM does not wait for daemon threads to finish; they may be stopped abruptly.																	
Default Type	By default, all threads are User Threads unless explicitly set as Daemon.	Must be explicitly marked as Daemon using <code>setDaemon(true)</code> .																	
Daemon Example	<pre> public class DaemonUserThreadDemo { public static void main(String[] args) { Thread bgThread = new Thread(new DaemonHelper()); Thread usrThread = new Thread(new UserThreadHelper()); bgThread.setDaemon(true); bgThread.start(); usrThread.start(); } } class DaemonHelper implements Runnable { @Override public void run() { int count = 0; while (count < 500) { try { Thread.sleep(1000); } catch (InterruptedException e) { throw new RuntimeException(e); } count++; System.out.println("Daemon helper running ..."); } } } class UserThreadHelper implements Runnable { @Override public void run() { try { Thread.sleep(5000); } catch (InterruptedException e) { throw new RuntimeException(e); } System.out.println("User Thread Helper done with execution!"); } } </pre>																		
Thread Scheduler	<p>The Thread Scheduler in Java is part of the JVM and OS that decides which thread to execute next from the ready queue. It is non-deterministic, meaning we cannot guarantee the exact order of execution.</p> <p>It uses a mix of:</p> <ul style="list-style-type: none"> • Preemptive Scheduling (higher priority threads may interrupt lower ones) • Time-Sliced (Round-Robin) Scheduling (each thread gets a small time slice) 																		

Thread Priority	<p>Each thread in Java has a priority, which helps the scheduler decide which thread to run first.</p> <p>Thread priority values range from 1 to 10, where:</p> <ul style="list-style-type: none"> • Thread.MIN_PRIORITY = 1 • Thread.NORM_PRIORITY = 5 (default) • Thread.MAX_PRIORITY = 10 <p>However, priority is just a hint—it does not guarantee execution order. Thread Scheduler stores threads in a queue and same priority are FIFO.</p> <pre>t1.setPriority(Thread.MIN_PRIORITY); // 1 t2.setPriority(Thread.NORM_PRIORITY); // 5 t3.setPriority(Thread.MAX_PRIORITY); // 10 System.out.println(Thread.currentThread().getName()); System.out.println(Thread.currentThread().getPriority()); Thread.currentThread().setPriority(Thread.MAX_PRIORITY); System.out.println(Thread.currentThread().getPriority());</pre>
Synchronization	<pre>public class SynchronisationDemo { private static int counter = 0; public static void main(String[] args) { Thread one = new Thread(() -> { for (int i = 0; i < 10000; i++) { //counter++; increment(); } }); Thread two = new Thread(() -> { for (int i = 0; i < 10000; i++) { //counter++; increment(); } }); one.start(); two.start(); try { one.join(); two.join(); } catch (InterruptedException e) { throw new RuntimeException(e); } System.out.println("Counter is : " + counter); } private synchronized static void increment() { counter++; } }</pre> <p>Method level synchronized is discouraged as it locks entire method creating performance bottlenecks.</p>
LockWithCustomObjects	<pre>public class LockWithCustomObjects { private static int counter1 = 0; private static int counter2 = 0; private static final Object lock1 = new Object(); private static final Object lock2 = new Object(); public static void main(String[] args) { Thread one = new Thread(() -> { for (int i = 0; i < 10000; i++) { increment1(); } }); Thread two = new Thread(() -> { for (int i = 0; i < 10000; i++) { increment2(); } }); } private void increment1() { synchronized(lock1) { counter1++; } } private void increment2() { synchronized(lock2) { counter2++; } } }</pre>

```

        one.start();
        two.start();

        try {
            one.join();
            two.join();
        } catch (InterruptedException e) {
            throw new RuntimeException(e);
        }

        System.out.println("Counter values : Counter 1 " + counter1 + " Counter 2 " + counter2);
    }

    private static void increment1() {
        synchronized (lock1) {
            counter1++;
        }
    }

    private static void increment2() {
        synchronized (lock2) {
            counter2++;
        }
    }
}

```

	<p>wait() and notify()</p> <p>wait() and notify() are used for inter-thread communication, allowing threads to coordinate execution. They are methods of the Object class and must be called within a synchronized block.</p> <p>How wait() Works</p> <ul style="list-style-type: none"> • A thread calls wait() inside a synchronized block, releasing the lock and entering a waiting state. • The thread remains blocked until another thread calls notify() or notifyAll(). <p>How notify() Works</p> <ul style="list-style-type: none"> • A thread calls notify() to wake up one waiting thread. • The woken thread reacquires the lock and resumes execution after the notify thread has completed its execution <pre> public class WaitNotifyDemo { private static final Object LOCK = new Object(); public static void main(String[] args) { Thread one = new Thread(() -> { try { one(); } catch (InterruptedException e) { throw new RuntimeException(e); } }); Thread two = new Thread(() -> { try { two(); } catch (InterruptedException e) { throw new RuntimeException(e); } }); one.start(); two.start(); } private static void one() throws InterruptedException { synchronized (LOCK) { System.out.println("Hello from method one..."); LOCK.wait(); System.out.println("Back Again in the method one"); } } private static void two() throws InterruptedException { synchronized (LOCK) { System.out.println("Hello from method two..."); LOCK.notify(); // Remaining code lines in the block are executed System.out.println("Hello from method two even after notify..."); } } } </pre>
--	---

	<pre> } </pre>
Producer Consumer Example	<pre> import java.util.LinkedList; class SharedQueue { private final LinkedList<Integer> queue = new LinkedList<>(); private final int CAPACITY = 5; public synchronized void produce(int item) throws InterruptedException { while (queue.size() == CAPACITY) { System.out.println("Queue is full. Producer is waiting..."); wait(); // Release lock and wait } queue.add(item); System.out.println("Produced: " + item); notify(); // Notify consumer that data is available } public synchronized int consume() throws InterruptedException { while (queue.isEmpty()) { System.out.println("Queue is empty. Consumer is waiting..."); wait(); // Release lock and wait } int item = queue.removeFirst(); System.out.println("Consumed: " + item); notify(); // Notify producer that space is available return item; } } public class WaitNotifyExample { public static void main(String[] args) { SharedQueue sharedQueue = new SharedQueue(); // Producer Thread Thread producer = new Thread(() -> { try { for (int i = 1; i <= 10; i++) { sharedQueue.produce(i); Thread.sleep(500); // Simulate production delay } } catch (InterruptedException e) { e.printStackTrace(); } }); // Consumer Thread Thread consumer = new Thread(() -> { try { for (int i = 1; i <= 10; i++) { sharedQueue.consume(); Thread.sleep(1000); // Simulate consumption delay } } catch (InterruptedException e) { e.printStackTrace(); } }); producer.start(); consumer.start(); } } </pre>
ExecutorService	<p>ExecutorService is a framework in Java that manages and controls thread execution efficiently. It belongs to the <code>java.util.concurrent</code> package and provides a higher-level replacement for manually creating and managing threads.</p> <p>Instead of creating and starting threads manually, we use ExecutorService to handle thread pooling, task submission, and lifecycle management.</p> <p>Problems with Traditional Threads</p> <ul style="list-style-type: none"> • High Overhead → Creating a new thread for every task is expensive. • Difficult to Manage → No built-in way to control thread execution. • No Thread Reuse → Once a thread finishes execution, it cannot be reused. <p>How ExecutorService Solves These Problems</p> <ul style="list-style-type: none"> • Thread Pooling → Reuses a fixed number of threads instead of creating new ones. • Better Performance → Reduces thread creation overhead. • Task Queuing → Handles multiple tasks efficiently.

- **Graceful Shutdown** → Provides methods to manage lifecycle (shutdown(), awaitTermination()).

	<p>Fixed Thread Pool (Best for CPU-Intensive Tasks)</p>	<p>Best for scenarios where the number of threads is known in advance.</p> <pre>public static void main(String[] args) { ExecutorService executor = Executors.newFixedThreadPool(3); for (int i = 1; i <= 5; i++) { final int taskId = i; executor.submit(() -> { System.out.println("Executing Task " + taskId + " by " + Thread.currentThread().getName()); }); } executor.shutdown(); }</pre>
	<p>Cached Thread Pool (Best for Short-Lived Tasks)</p>	<ul style="list-style-type: none"> • Creates new threads as needed and reuses them when possible. • Best for tasks with unpredictable execution times. • Thread is killed if it is idle for more than 60 seconds. <pre>ExecutorService executor = Executors.newCachedThreadPool();</pre>
	<p>Single Thread Executor (Best for Sequential Execution)</p>	<ul style="list-style-type: none"> • Executes tasks one by one in a single thread. • Ensures task ordering. <pre>ExecutorService executor = Executors.newSingleThreadExecutor();</pre>
	<p>Scheduled Thread Pool (Best for Delayed or Repeated Tasks)</p>	<pre>ScheduledExecutorService scheduler = Executors.newScheduledThreadPool(2); // Schedule task with a delay scheduler.schedule(() -> System.out.println("Task executed after 2 seconds"), 2, TimeUnit.SECONDS); // Schedule a repeating task scheduler.scheduleAtFixedRate(() -> System.out.println("Repeating task"), 1, 3, TimeUnit.SECONDS);</pre> <p>Use Case: Periodic background tasks, Auto-save functionality, Monitoring.</p>
Internal Working	<p>When a task is submitted (submit() or execute()), the following steps occur:</p> <ol style="list-style-type: none"> 1. Task is added to a queue (if all worker threads are busy). 2. Worker threads pick tasks from the queue and execute them. 3. If queue(LinkedBlockingQueue) is full, a new thread is created (up to the max limit in ThreadPoolExecutor). 4. If max threads are reached, the task is either rejected or handled based on the rejection policy. 5. Threads are reused instead of creating new ones for every task. 6. When shutting down, ExecutorService ensures all running tasks complete before termination. 	
SingleThreadExecutor	<pre>public class SingleThreadExecutorDemo { public static void main(String[] args) { try (ExecutorService service = Executors.newSingleThreadExecutor()) { for (int i = 0; i < 5; i++) { service.submit(new Task(i)); } } } class Task implements Runnable { private final int taskId; public Task(int taskId) { this.taskId = taskId; } @Override public void run() { System.out.println("Task with ID : " + taskId + " being executed by thread : " + Thread.currentThread().getName()); try { Thread.sleep(500); } catch (InterruptedException e) { throw new RuntimeException(e); } } } }</pre>	
FixedThreadPool	<pre>public class FixedThreadPoolDemo { public static void main(String[] args) { try (ExecutorService executor = Executors.newFixedThreadPool(2)) {</pre>	

```

        for (int i = 0; i < 7; i++) {
            executor.execute(new Work(i + 1));
        }
    }
}

class Work implements Runnable {
    private final int workId;
    public Work(int workId) {
        this.workId = workId;
    }

    @Override
    public void run() {
        System.out.println("Task with ID : " + workId + " being executed by thread : " + Thread.currentThread().getName());
        try {
            Thread.sleep(700);
        } catch (InterruptedException e) {
            throw new RuntimeException(e);
        }
    }
}

```

	<p>ThreadPoolExecutor is the core implementation behind ExecutorService. It provides full control over:</p> <ol style="list-style-type: none"> 1. Number of threads (core and max) 2. Task queuing mechanism 3. Thread lifecycle management 4. Task rejection handling <p>It is part of java.util.concurrent and is used when built-in Executors (e.g., Executors.newFixedThreadPool()) don't provide enough customization.</p> <pre> static { int nCpu = Runtime.getRuntime().availableProcessors(); LOGGER.info("No. of CPU in the m/c: " + nCpu); TPEXECUTOR = new ThreadPoolExecutor(10, //Runtime.getRuntime().availableProcessors(), 10, 50000L, TimeUnit.MILLISECONDS, new LinkedBlockingQueue<Runnable>()); } </pre>
Callable	<ul style="list-style-type: none"> • Callable<T> is an interface in java.util.concurrent used to represent asynchronous tasks that return a result. • Unlike Runnable, Callable can return a value and throw checked exceptions <pre> import java.util.concurrent.*; public class CallableExample { public static void main(String[] args) { ExecutorService executor = Executors.newFixedThreadPool(2); // Submitting a Callable task Future<Integer> futureResult = executor.submit(() -> { System.out.println("Executing Callable Task..."); Thread.sleep(2000); return 42; // Returning a value }); try { System.out.println("Result: " + futureResult.get()); // Blocking call } catch (InterruptedException ExecutionException e) { e.printStackTrace(); } executor.shutdown(); } } </pre>

Miscellaneous

13 November 2024 09:22

str.split("\\s+")	Efficiently splits by any whitespace sequence and ignores multiple consecutive spaces
str.split(" ");	Splits exactly at single space characters, retaining empty strings for consecutive spaces Hello World" results in ["Hello", "", "", "World"]
str.split("")	Splits into individual characters, return type is still array of strings
Arrays.copyOf(arr, n)	Copies n elements to new array
Arrays.toString(str)	Converts an array to a readable string format
Character.isDigit	Checks if the specified character is a digit
Character.isUppercase Character.isSpaceChar Character.isLowercase Character.isLetterOrDigit Character.isWhitespace Character.isLetter(word.charAt(0)) Character.toUpperCase	
Character.getNumericValue	returns an integer that represents the numeric value of the character
Index of chars in String	String word="dd"; System.out.println(word.indexOf('d'));//0 System.out.println(word.lastIndexOf('d'));//1 If both are same then digit is not repeated in string, else repeated
SubString Function	String str = "Hello, World!"; String result = str.substring(7); System.out.println(result); // Output: "World!" String str = "Hello, World!"; String result = str.substring(7, 12); System.out.println(result); // Output: "World" beginIndex is the starting index (inclusive) of the substring. endIndex is the ending index (exclusive) of the substring
String.format	String name = "Alice"; int age = 30; double height = 5.8; String formattedString = String.format("Name: %s, Age: %d, Height: %.1f feet", name, age, height);
Arrays.sort(arrayA);	used to sort the elements of the array arrayA in ascending order import java.util.Arrays; import java.util.Collections; public class Main { public static void main(String[] args) { Integer[] arrayA = {5, 3, 8, 1, 2}; // Note: Changed to Integer[] Arrays.sort(arrayA, Collections.reverseOrder()); System.out.println(Arrays.toString(arrayA)); // Output: [8, 5, 3, 2, 1] } }
Arrays.equals(arrayA, arrayB);	Used to compare two arrays for equality. It checks if both arrays contain the same elements in the same order
Array Declaration	int[] a = new int[5]; a[0]=5; System.out.println(a[0]); int[] b = {1,2,3,4,5}; System.out.println(b[2]); int[][] c = new int[2][2]; c[0][0]=2; System.out.println(c[0][0]); int[][] d = {{1,2,3},{5,6,7}}; System.out.println(d[1][0]);
Print Array	Arrays.stream(beansArray).forEach(System.out::println);

System.out.printf	Specifier	Description
	%d	Decimal integer (int, long)
	%f	Floating-point number (float, double)
	%s	String
	%c	Character
	%b	Boolean
	%x	Hexadecimal integer
	%o	Octal integer
Char array to String	%n	Platform-specific newline character (equivalent to System.lineSeparator())
	<ul style="list-style-type: none"> • System.out.printf("The number is: %d%n", i);//42 • System.out.printf("The value of pi is: %.2f%n", pi);//3.14 • String name = "Saurav"; System.out.printf("Hello, %s!%n", name);//Hello, Saurav! • System.out.printf("Pi with width 3 and precision 2: %3.2f%n", pi);//3.14 	
Char array to String	str = new String(ch);	
Accepting user input	<pre>Scanner scanner = new Scanner(System.in); System.out.print("Enter an integer: "); int intValue = scanner.nextInt(); System.out.println("You entered: " + intValue); System.out.print("Enter a double: "); double doubleValue = scanner.nextDouble(); System.out.println("You entered: " + doubleValue); scanner.nextLine(); System.out.print("Enter a string: "); String stringValue = scanner.nextLine(); System.out.println("You entered: " + stringValue); System.out.print("Enter a boolean (true/false): "); boolean booleanValue = scanner.nextBoolean(); System.out.println("You entered: " + booleanValue); System.out.print("Enter a character: "); char charValue = scanner.next().charAt(0); System.out.println("You entered: " + charValue);</pre>	
Java.lang package	<p>The java.lang package is a core part of the Java programming language. It's automatically imported by the Java compiler, so you don't need to import it explicitly. This package contains fundamental classes that are essential for the Java language to function. Here are a few key classes:</p> <p>Object: The root class of the Java class hierarchy. Every class has Object as a superclass.</p> <p>String: Represents a sequence of characters. Immutable and widely used for text manipulation.</p> <p>Math: Provides methods for performing basic numeric operations like exponentiation, logarithms, and trigonometric functions.</p> <p>System: Contains several useful class fields and methods. It provides access to system resources such as standard input, output, and error streams.</p> <p>Thread: Represents a thread of execution in a program. Used for concurrent programming</p>	
Variable Arguments in Java	<ul style="list-style-type: none"> - Variable arguments, or varargs, allow a method to accept a variable number of parameters. - Introduced in Java 5, varargs simplify method invocation when the number of arguments is not known at compile time. <p>Declaration Syntax:</p> <ul style="list-style-type: none"> - returnType methodName(dataType... parameterName) - The ellipsis (...) denotes variable arguments and must be placed after the parameter type. - The variable arguments parameter acts like an array within the method. 	
javac	It produces java byte code from *.java files, an intermediate representation containing instructions.	
base class of all classes	Java.lang.Object	
Storage areas in JVM	<ul style="list-style-type: none"> - Method Area: Storage for compiled class files. - Heap: Storage for Objects. - Java Stack: Storage for local variables. - PC Register: Stores the address of the next instruction. - Native Method Stacks: Executes native methods. 	
Runtime Polymorphism by data members?	<p>Not possible because method overriding is used to achieve runtime polymorphism and data members cannot be overridden. We can override the member functions but not the data members. Consider the example given below.</p> <pre>class Bike{ int speedlimit=90; } class Honda3 extends Bike{ int speedlimit=150; public static void main(String args[]){ }</pre>	

	<pre>Bike obj=new Honda3(); System.out.println(obj.speedlimit);//90 }</pre> <p>Output:</p> <p>90</p>																											
String Pool	String pool is the space reserved in the heap memory that can be used to store the strings. The main advantage of using the String pool is whenever we create a string literal; the JVM checks the "string constant pool" first. If the string already exists in the pool, a reference to the pooled instance is returned. If the string doesn't exist in the pool, a new string instance is created and placed in the pool. Therefore, it saves the memory by avoiding the duplicacy.																											
Why are the objects immutable in java?	Because Java uses the concept of the string literal. Suppose there are five reference variables, all refer to one object "sac hin". If one reference variable changes the value of the object, it will be affected by all the reference variables. That is why string objects are immutable in java.																											
Final Keyword	final variable - A variable declared as final prevents the content of that variable being modified final method - A method declared as final prevents the user from overriding that method final class - A class declared as final cannot be extended thus prevents inheritance																											
Static Method	<ul style="list-style-type: none"> • A static method is a method which belongs to the class and not to the instance(object) • A static method can be invoked without the need for creating an instance of a class • A static method can call only other static methods and cannot call a non-static method from it • A static method can access static data member and can change the value of it • A static method cannot refer to this or super keywords in anyway • Static methods cannot access non static variables while non static functions can access static variables. • Static methods cannot be overridden because they are not part of the object's state. Rather, they belongs to the class static method can access static data member and can change the value of it. They can be overloaded. 																											
Static Block	Executed before Main and Constructors. Program with only static block will not run.																											
Static variable	A static variable is one that's associated with a class, not instance (object) of that class. They are initialized only once , at the start of the execution . A single copy to be shared by all instances of the class and it can be accessed directly by the class name and doesn't need any object. One common use of static is to create a constant value that's attached to a class. For example, if you have a variable: private static int stc = 0; and you increment it (stc++) in one instance, the change will be reflected in all instances. stc will now be 1 in all instances.																											
Memory Usage	<table border="1"> <thead> <tr> <th>Data Type</th> <th>Value Range</th> <th>Memory Usage (bits)</th> </tr> </thead> <tbody> <tr> <td>byte</td> <td>-128 to 127</td> <td>8</td> </tr> <tr> <td>short</td> <td>-32,768 to 32,767</td> <td>16</td> </tr> <tr> <td>int</td> <td>-2,147,483,648 to 2,147,483,647</td> <td>32</td> </tr> <tr> <td>long</td> <td>-9,223,372,036,854,775,808 to 9,223,372,036,854,775,807</td> <td>64</td> </tr> <tr> <td>float</td> <td>Approximately +-3.4028235e+38</td> <td>32</td> </tr> <tr> <td>double</td> <td>Approximately +-1.7976931348623157e+308</td> <td>64</td> </tr> <tr> <td>char</td> <td>Unicode characters (0 to 65,535)</td> <td>16</td> </tr> <tr> <td>boolean</td> <td>true or false</td> <td>1</td> </tr> </tbody> </table>	Data Type	Value Range	Memory Usage (bits)	byte	-128 to 127	8	short	-32,768 to 32,767	16	int	-2,147,483,648 to 2,147,483,647	32	long	-9,223,372,036,854,775,808 to 9,223,372,036,854,775,807	64	float	Approximately +-3.4028235e+38	32	double	Approximately +-1.7976931348623157e+308	64	char	Unicode characters (0 to 65,535)	16	boolean	true or false	1
Data Type	Value Range	Memory Usage (bits)																										
byte	-128 to 127	8																										
short	-32,768 to 32,767	16																										
int	-2,147,483,648 to 2,147,483,647	32																										
long	-9,223,372,036,854,775,808 to 9,223,372,036,854,775,807	64																										
float	Approximately +-3.4028235e+38	32																										
double	Approximately +-1.7976931348623157e+308	64																										
char	Unicode characters (0 to 65,535)	16																										
boolean	true or false	1																										
Pass-by-Value in Java	<p>When you pass a primitive data type to a method, Java passes a copy of the value. Any changes made to this parameter inside the method do not affect the original value.</p> <pre>public static void main(String[] args) { int original = 5; changePrimitive(original); System.out.println(original); // Outputs: 5 } static void changePrimitive(int num) { num = 10; }</pre> <p>Object References: When you pass an object to a method, Java still passes a copy of the value. The "value" here is the reference to the object in memory, not the actual object. This means you can modify the object's attributes within the method because both the original and the copy of the reference point to the same object.</p> <pre>public static void main(String[] args) { MyObject obj = new MyObject(); obj.value = 5; changeObject(obj); System.out.println(obj.value); // Outputs: 10 } static void changeObject(MyObject obj) { obj.value = 10;</pre>																											

	<pre> } static class MyObject { int value; } </pre>										
try-with-resources	<p>Introduced in Java 7: the try-with-resources statement. This construct simplifies resource management, particularly for objects that implement the AutoCloseable interface, such as FileInputStream.</p> <pre> try (FileInputStream fileInputStream = new FileInputStream("src/test/resources/data/secTypesDemo.xlsx")) { // ... } </pre> <p>The key advantage of using try-with-resources is that it automatically closes the resource (fileInputStream in this case) at the end of the try block or if an exception is thrown. This ensures proper resource cleanup, preventing potential resource leaks.</p>										
Replace vs replaceAll	<pre> String str = "Hello, World! 123"; String newStr = str.replaceAll("\\d", "*"); // Replace all digits with * System.out.println(newStr); // Output: Hello, World! *** </pre> <pre> String str = "Hello, World!"; String newStr = str.replace("o", "u"); System.out.println(newStr); // Output: Hellu, Wurld! </pre>										
Throw Vs Throws	<table border="1"> <thead> <tr> <th>throw keyword</th> <th>throws keyword</th> </tr> </thead> <tbody> <tr> <td>The throw keyword is used to throw an exception explicitly.</td> <td>The throws keyword is used to declare an exception.</td> </tr> <tr> <td>The checked exceptions cannot be propagated with throw.</td> <td>The checked exception can be propagated with throws</td> </tr> <tr> <td>The throw keyword is used within the method.</td> <td>The throws keyword is used with the method signature.</td> </tr> <tr> <td>You cannot throw multiple exceptions.</td> <td>You can declare multiple exceptions, e.g., public void method() throws IOException, SQLException.</td> </tr> </tbody> </table>	throw keyword	throws keyword	The throw keyword is used to throw an exception explicitly.	The throws keyword is used to declare an exception.	The checked exceptions cannot be propagated with throw.	The checked exception can be propagated with throws	The throw keyword is used within the method.	The throws keyword is used with the method signature.	You cannot throw multiple exceptions.	You can declare multiple exceptions, e.g., public void method() throws IOException, SQLException.
throw keyword	throws keyword										
The throw keyword is used to throw an exception explicitly.	The throws keyword is used to declare an exception.										
The checked exceptions cannot be propagated with throw.	The checked exception can be propagated with throws										
The throw keyword is used within the method.	The throws keyword is used with the method signature.										
You cannot throw multiple exceptions.	You can declare multiple exceptions, e.g., public void method() throws IOException, SQLException.										
HashCode and Equals	<p>If two objects are equal (equals() returns true), their hashCode() must be the same but Two objects with the same hashCode() are not necessarily equal.</p> <p>If equals() is overridden without hashCode(), the following issues arise:</p> <ul style="list-style-type: none"> Object may be considered equal but stored in different hash buckets in HashSet or HashMap. Leads to data inconsistency, duplicate entries, and unexpected behavior. Overriding hashCode() ensures objects that are equal are stored in the same bucket. 										
File Operations	<table border="1"> <tbody> <tr> <td>Reading</td> <td> <pre> String f1="src/test/resources/file1.txt"; try(Stream<String> lines= Files.lines(Paths.get(f1))){ lines.forEach(System.out::println); }catch (Exception e){ System.out.println("Unable to read file"); } </pre> </td></tr> <tr> <td>Reading</td> <td> <pre> try (BufferedReader reader = new BufferedReader(new FileReader(filePath))) { String currentLine; while ((currentLine = reader.readLine()) != null) { System.out.println(currentLine); } } catch (IOException e) { System.err.println("An error occurred while reading the file."); e.printStackTrace(); } </pre> </td></tr> <tr> <td>Writing</td> <td> <pre> public static void main(String[] args) { String text = "\nWelcome to GeeksforGeeks\nHappy Learning!"; String filePath = "src/test/resources/file1.txt"; try (BufferedWriter writer = new BufferedWriter(new FileWriter(filePath,true))) { writer.write(text); System.out.println("Successfully appended to the file: " + filePath); } catch (IOException e) { System.err.println("An error occurred while appending to the file."); e.printStackTrace(); } } </pre> </td></tr> <tr> <td>Line By Line Compare</td> <td> <pre> String file1 = "src/test/resources/file1.txt"; String file2 = "src/test/resources/file2.txt"; try (Stream<String> fileStream1 = Files.lines(Paths.get(file1)); ...) </pre> </td></tr> </tbody> </table>	Reading	<pre> String f1="src/test/resources/file1.txt"; try(Stream<String> lines= Files.lines(Paths.get(f1))){ lines.forEach(System.out::println); }catch (Exception e){ System.out.println("Unable to read file"); } </pre>	Reading	<pre> try (BufferedReader reader = new BufferedReader(new FileReader(filePath))) { String currentLine; while ((currentLine = reader.readLine()) != null) { System.out.println(currentLine); } } catch (IOException e) { System.err.println("An error occurred while reading the file."); e.printStackTrace(); } </pre>	Writing	<pre> public static void main(String[] args) { String text = "\nWelcome to GeeksforGeeks\nHappy Learning!"; String filePath = "src/test/resources/file1.txt"; try (BufferedWriter writer = new BufferedWriter(new FileWriter(filePath,true))) { writer.write(text); System.out.println("Successfully appended to the file: " + filePath); } catch (IOException e) { System.err.println("An error occurred while appending to the file."); e.printStackTrace(); } } </pre>	Line By Line Compare	<pre> String file1 = "src/test/resources/file1.txt"; String file2 = "src/test/resources/file2.txt"; try (Stream<String> fileStream1 = Files.lines(Paths.get(file1)); ...) </pre>		
Reading	<pre> String f1="src/test/resources/file1.txt"; try(Stream<String> lines= Files.lines(Paths.get(f1))){ lines.forEach(System.out::println); }catch (Exception e){ System.out.println("Unable to read file"); } </pre>										
Reading	<pre> try (BufferedReader reader = new BufferedReader(new FileReader(filePath))) { String currentLine; while ((currentLine = reader.readLine()) != null) { System.out.println(currentLine); } } catch (IOException e) { System.err.println("An error occurred while reading the file."); e.printStackTrace(); } </pre>										
Writing	<pre> public static void main(String[] args) { String text = "\nWelcome to GeeksforGeeks\nHappy Learning!"; String filePath = "src/test/resources/file1.txt"; try (BufferedWriter writer = new BufferedWriter(new FileWriter(filePath,true))) { writer.write(text); System.out.println("Successfully appended to the file: " + filePath); } catch (IOException e) { System.err.println("An error occurred while appending to the file."); e.printStackTrace(); } } </pre>										
Line By Line Compare	<pre> String file1 = "src/test/resources/file1.txt"; String file2 = "src/test/resources/file2.txt"; try (Stream<String> fileStream1 = Files.lines(Paths.get(file1)); ...) </pre>										

	<pre> Stream<String> fileStream2 = Files.lines(Paths.get(file2))) { Iterator<String> it1 = fileStream1.iterator(); Iterator<String> it2 = fileStream2.iterator(); int lineNum = 1; while (it1.hasNext() it2.hasNext()) { String line1 = it1.hasNext() ? it1.next() : "<no line>"; String line2 = it2.hasNext() ? it2.next() : "<no line>"; if (!line1.equals(line2)) { System.out.printf("Difference at line %d:\nFile1: %s\nFile2: %s\n\n", lineNum, line1, line2); } lineNum++; } } catch (IOException e) { e.printStackTrace(); } } </pre>
Read and Write	<pre> public static void main(String[] args) { String inputFile = "src/test/resources/application.log"; String outputFile = "src/test/resources/error.log"; try (BufferedReader reader = new BufferedReader(new FileReader(inputFile)); BufferedWriter writer = new BufferedWriter(new FileWriter(outputFile))) { String currentLine; while ((currentLine = reader.readLine()) != null) { if (currentLine.contains("ERROR")) { writer.write(currentLine); writer.newLine(); } } System.out.println("Successfully extracted error logs to: " + Paths.get(outputFile).toAbsolutePath()); } catch (IOException e) { System.err.println("An I/O error occurred: " + e.getMessage()); e.printStackTrace(); } } </pre>
Read Write with Stream	<pre> public static void main(String[] args) { Path inputFile = Paths.get("src/test/resources/application.log"); Path outputFile = Paths.get("src/test/resources/error.log"); try { List<String> errorLines = Files.lines(inputFile).filter(line -> line.contains("ERROR")).collect(Collectors.toList()); Files.write(outputFile, errorLines, StandardOpenOption.CREATE, StandardOpenOption.APPEND); System.out.println("Successfully extracted error logs to: " + outputFile.toAbsolutePath()); } catch (IOException e) { System.err.println("An I/O error occurred: " + e.getMessage()); e.printStackTrace(); } } </pre>

Regex is supported by the `java.util.regex` package, which provides two key classes:

Pattern	Matcher
Used for defining patterns	Used for performing match operations on text using patterns
<ul style="list-style-type: none"> • <code>Pattern.compile(String regex)</code>: Compiles a regular expression into a <code>Pattern</code> object. • <code>Pattern.matches(String regex, CharSequence input)</code>: Directly matches a regex pattern against an input string. 	<ul style="list-style-type: none"> • <code>Matcher.matcher(CharSequence input)</code>: Matches the pattern against the input string. • <code>matcher.find()</code>: Searches for the next subsequence that matches the pattern. • <code>matcher.group()</code>: Returns the matched subsequence.

Character Class	Description
[xyz]	Single Character from x,y or z
[^xyz]	Any characters other than x,y or z
[a-zA-Z]	characters from range a to z or A to Z.
[a-fm-t]]	Union of a to f and m to t.
[a-z && [^bc]]	a to z union with except b and c
[a-z && [^m-p]]	a to z union with except range m to p
X?	X appears once or not

X+	X appears once or more than once
X*	X appears zero or not once
X{n}	X appears n times
X{n,}	X appears n times or more than n
X{n,m}	X appears greater than equal to n times and less than m times.
.	Any character
\d	Any digits, [0-9]
\D	Any non-digit, [^0-9]
\s	Whitespace character, [\t\n\x0B\f\r]
\S	Non-whitespace character, [^\s]
\w	Word character, [a-zA-Z_0-9]
\W	Non-word character, [^\w]
\b	Word boundary
\B	Non -Word boundary

```
public boolean isValidEmail(String email) {
    String regex = "^[a-zA-Z0-9_+&*-]+(?:\\.[a-zA-Z0-9_+&*-]+)*@[?:[a-zA-Z0-9-]+\.\.]+[a-zA-Z]{2,7}$";
    Pattern pattern = Pattern.compile(regex);
    Matcher matcher = pattern.matcher(email);
    return matcher.matches();
}
```

^	Ensures that the pattern starts matching from the beginning of the string
\$	Ensures that the pattern ends matching at the end of the string
[a-zA-Z0-9_+&*-]+	Any uppercase or lowercase letter (a-z, A-Z) Any digit (0-9) Special characters: _, +, &, *, -. + after] denotes group of such valid chars
(?:\\.[a-zA-Z0-9_+&*-]+)*	\\.: Matches a literal dot (.). The backslash (\) escapes the dot, ensuring it is treated as a literal character rather than a regex wildcard. [a-zA-Z0-9_+&*-]+: After the dot, this part must have at least one valid character (same as defined above). *: Allows this group to appear zero or more times. This makes subdomains in the local part (e.g., user.name.part) optional. (?: ...): A non-capturing group, meaning it groups the content without creating a backreference.
@	Matches the @ character, separating the local part from the domain.
(?:[a-zA-Z0-9-]+\.\.)+	[a-zA-Z0-9-]: Matches a single character that can be: Any uppercase or lowercase letter, Any digit, The hyphen (-). +: Ensures at least one such character exists. \.: Matches a literal dot (.). (?: ...)+: The group repeats one or more times, ensuring there is at least one domain level (e.g., example. in example.com).

Lombok	Annotation	Description
	@Getter	Generates getter methods for all fields in the class.
	@Setter	Generates setter methods for all fields in the class.
	@ToString	Generates a toString method including all class fields by default. Customizable to exclude certain fields.
	@EqualsAndHashCode	Generates equals and hashCode methods based on the fields of the class. Customizable to exclude certain fields.
	@NoArgsConstructor	Generates a no-argument constructor for the class.
	@AllArgsConstructor	Generates a constructor with one parameter for each field in the class.
	@RequiredArgsConstructor	Generates a constructor for final fields and fields marked with @NonNull.
	@Data	Combines @Getter, @Setter, @ToString, @EqualsAndHashCode, and @RequiredArgsConstructor annotations.
	@Value	Used for immutable classes; generates getters, toString, equals, hashCode, and a constructor.

	<code>@Builder</code>	Generates the builder pattern for object creation. Customizable to include specific fields only.
	<code>@Singular</code>	Used with <code>@Builder</code> to generate methods that add a single element to a collection (e.g., List, Set).
	<code>@NonNull</code>	Generates a null-check for fields annotated with it. Commonly used in constructors and methods.
	<code>@Delegate</code>	Generates delegation methods for a field. The methods of the delegate type are added to the owning class.
	<code>@Slf4j</code>	Generates a Slf4j logger field in the class. Other logging frameworks are supported as well (e.g., <code>@Log4j</code> , <code>@Log</code>).
	<code>@Cleanup</code>	Ensures a resource (e.g., a stream) is cleaned up by closing it after usage.
	<code>@SneakyThrows</code>	Allows throwing checked exceptions without explicitly declaring them in the method signature.
	<code>@Accessors</code>	Customizes the naming conventions for generated getters and setters (e.g., fluent style).
	<code>@FieldDefaults</code>	Sets default visibility and other modifiers for fields (e.g., private, final).
	<code>@With</code>	Generates a "with" method for creating a copy of an immutable object with one field changed.
	<code>@SuperBuilder</code>	Similar to <code>@Builder</code> , but supports inheritance by generating builders for superclasses and subclasses.
	<code>@Getter(lazy = true)</code>	Generates a lazy-initialized getter for a field, initializing it only on the first access.
	<code>@NoArgsConstructor(force = true)</code>	Forces generation of a no-argument constructor, initializing final fields with default values.
	<code>@RequiredArgsConstructor(statue cName = "of")</code>	Generates a constructor for final fields and fields marked with <code>@NonNull</code> , and provides a static method named of to create instances.

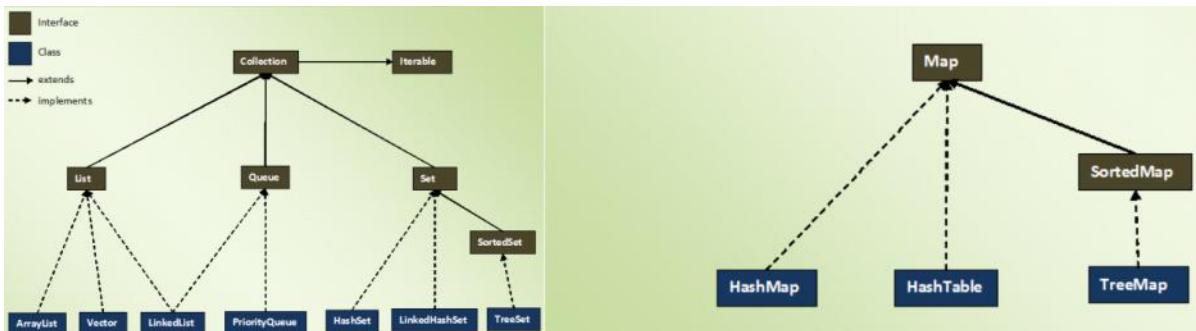
Clean Code	Naming	<ul style="list-style-type: none"> Give meaningful and short names to variable function and classes so that another person doesn't have to read the code to understand what is happening. <ul style="list-style-type: none"> Variables/Constants: Use nouns or short phrases with adjectives. e.g userData, isValid Functions: Verbs. e.g sendData(), getUser(), saveUser() Classes: Nouns. e.g UserBody, RequestBody Don't include redundant info in names like userWithNameAndAge Avoid Slangs like user.diePlease() Be Consistent with naming like if you use getUser then use getProduct, don't use fetchProduct() 			
	Ordering Functions	If one function calls another, the next function should be below the 1st function to improve readability.			
	Definition	<ul style="list-style-type: none"> Working and calling of the function should be easy/readable. What Matters <ul style="list-style-type: none"> Number and order of arguments Length of the function Rules <ul style="list-style-type: none"> Minimise number of parameters Try to use max 2 parameters Use Maps for more than 2 parameters so that order of parameters doesn't matter It should be small: A function should do one thing and outsource others 			
		<table border="1"> <tr> <td>This function takes 2 arguments</td> <td>Create another function</td> </tr> <tr> <td> <pre>function saveUser(email, password) { const user = { id: Math.random().toString(), email: email, password: password, }; db.insert('users', user); }</pre> </td> <td> Create A class <pre>class User { constructor(email, password) { this.email = email; this.password = password; this.id = Math.random().toString(); } save() { db.insert('users', this); } } const user = new User('test@test.com', 'testers'); user.save();</pre> </td> </tr> </table>	This function takes 2 arguments	Create another function	<pre>function saveUser(email, password) { const user = { id: Math.random().toString(), email: email, password: password, }; db.insert('users', user); }</pre>
This function takes 2 arguments	Create another function				
<pre>function saveUser(email, password) { const user = { id: Math.random().toString(), email: email, password: password, }; db.insert('users', user); }</pre>	Create A class <pre>class User { constructor(email, password) { this.email = email; this.password = password; this.id = Math.random().toString(); } save() { db.insert('users', this); } } const user = new User('test@test.com', 'testers'); user.save();</pre>				
Refactor	<ul style="list-style-type: none"> Extract codes which work on same functionality 				

	<ul style="list-style-type: none"> • Extract code which require more interpretation than surrounding code <p>Follow SRP and DRY</p>
Reflection	<p>Reflection in Java is a powerful feature that allows a program to inspect and manipulate the runtime behavior of classes, methods, and fields dynamically. It is part of the <code>java.lang.reflect</code> package and is commonly used for:</p> <ul style="list-style-type: none"> • Inspecting classes, methods, and fields at runtime • Accessing private members of a class • Dynamically creating objects and invoking methods • Framework development (e.g., dependency injection, serialization, ORMs, etc.)
Text Block	<p>Text Blocks in Java 15 provide a cleaner and more intuitive way to handle multi-line string literals, significantly enhancing the readability and maintainability of Java code that involves substantial textual content.</p> <pre>String multiLineString = """ This is a multi-line string literal. """; System.out.println(multiLineString);</pre> <p>Output: This is a multi-line string literal.</p>
CSV Reader	<pre>String file = "data.csv"; try (BufferedReader br = new BufferedReader(new FileReader(file))) { String line; while ((line = br.readLine()) != null) { String[] values = line.split(","); System.out.println(Arrays.toString(values)); } } catch (IOException e) { e.printStackTrace(); }</pre>
Inner Class Object	<p>Assume B is inner class of A</p> <pre>A obj=new A(); A.B obj1=obj.new B();</pre> <p>If Inner class is static: A.B obj1=new A.B();</p>
Other Points	<ul style="list-style-type: none"> • The <code>java.lang.System.exit()</code> method terminates the Java virtual machine, accepting a status code. Non-zero values typically indicate abnormal termination. • <code>System.out.print</code>: <code>System</code> is a final class of <code>java.lang</code>, <code>out</code> is a static variable of type <code>printstream</code>, and <code>print</code> is a method of <code>printstream</code> class. • Java automatically converts long to int if 'L' is not used in initialization. • When a package is imported, its sub-packages aren't imported; import them separately if needed. • Main method can be overloaded, and another class's main method can be called in a main method. Main method cannot be overridden • Every class has a constructor, whether it's a normal or abstract class. • When creating an object of a subclass, the parent class constructor is executed first, then the child class constructor. Compiler adds <code>super()</code> as the first statement in the child class constructor. • Default constructor cannot be called if a parameterized constructor is present unless explicitly declared. • Length is a property of array while a function in string • <code>Integer int1=new Integer(5)</code> will create new objects and == comparison won't work as in Strings. So create Integer variables with <code>Integer.valueOf(5)</code> • Equal and hashCode methods are overridden together in a class • JVM doesn't create an object of the Main class • <code>main()</code> is static for JVM to call it without creating an instance. • <code>throw</code>: Explicitly throws an exception. • <code>throws</code>: Specifies checked exceptions thrown by a method. • <code>Throwable</code>: Superclass of all errors and exceptions. • Use <code>BigDecimal</code> for financial calculations. Pass values as string to this constructor • For primitive variable we store its value in stack while we store address in case of objects. • We cannot overload methods that differ only by static keyword • We can overload <code>main()</code> in Java • <code>hashCode()</code> method: The <code>hashCode()</code> method returns a hash code value (an integer number). It returns the same integer number if two keys are identical using the <code>equals()</code> method. However, two hash code numbers can have different or the same keys. If two objects do not produce an equal result using the <code>equals()</code> method, then the <code>hashCode()</code> method will provide a different integer result for both objects. • Why override <code>equals()</code> method: The <code>equals</code> method is used to check whether two objects are the same or not. It needs to be overridden if we want to check the objects based on the property. • \n for new line • Var variable need value while declaration • Static blocks are executed first, followed by the main method. These are used to initialize static variables, loading native libs or

- doing complex one time setup i.e DB Connections.
- Static block is executed only once: the first time the class is loaded into memory. A class can have multiple static blocks, and they are executed in the order they appear in the source code
- Instance variable are stored in heap and automatically initialized.
- Local/Global variables are created in Stack.
- Class cannot be private or protected.

Introduction

29 July 2018 12:30



The Java Collections Framework is a class library designed to handle multiple objects, implemented in the `java.util` package. It stores references to other objects, representing a single unit of objects.

Iterable Interface:

- Root interface for all collection classes.
- Collection interface extends Iterable.
- Contains one abstract method: `Iterator<T> iterator()`, which returns an iterator over elements of type T.

Collection Interface:

- Implemented by all classes in the collection framework.
- Declares methods that every collection must have.
- Builds the foundation for the collection framework.
- Some methods: `Boolean add(Object obj)`, `Boolean addAll(Collection c)`, `void clear()`, etc.

Java Collections Cheat Sheet

Java Concept Of The Day

Basics

What is Java Collection Framework?

Java Collection Framework is a framework which provides some predefined classes and interfaces to store and manipulate the group of objects. Using Java collection framework, you can store the objects as a List or as a Set or as a Queue or as a Map and perform basic operations like adding, removing, updating, sorting, searching etc... with ease.

Why Java Collection Framework?

Earlier, arrays are used to store the group of objects. But, arrays are of fixed size. You can't change the size of an array once it is defined. It causes lots of difficulties while handling the group of objects. To overcome this drawback of arrays, Java Collection Framework is introduced from JDK 1.2.

Java Collections Hierarchy :

All the classes and interfaces related to Java collections are kept in `java.util` package. List, Set, Queue and Map are four top level interfaces of Java collection framework. All these interfaces (except Map) inherit from `java.util.Collection` interface which is the root interface in the Java collection framework.

List	Queue	Set	Map
<p>Intro :</p> <ul style="list-style-type: none"> • List is a sequential collection of objects. • Elements are positioned using zero-based index. • Elements can be inserted or removed or retrieved from any arbitrary position using an integer index. <p>Popular Implementations :</p> <ul style="list-style-type: none"> • <code>ArrayList</code>, <code>Vector</code> And <code>LinkedList</code> <p>Internal Structure :</p> <ul style="list-style-type: none"> • ArrayList : Internally uses re-sizeable array which grows or shrinks as we add or delete elements. • Vector : Same as <code>ArrayList</code> but it is synchronized. • LinkedList : Elements are stored as Nodes where each node consists of three parts – Reference To Previous Element, Value Of The Element and Reference To Next Element. <p>Null Elements :</p> <ul style="list-style-type: none"> • <code>ArrayList</code> : Yes • <code>Vector</code> : Yes • <code>LinkedList</code> : Yes <p>Duplicate Elements :</p> <ul style="list-style-type: none"> • <code>ArrayList</code> : Yes • <code>Vector</code> : Yes • <code>LinkedList</code> : Yes <p>Order Of Elements :</p> <ul style="list-style-type: none"> • <code>ArrayList</code> : Insertion Order • <code>Vector</code> : Insertion Order • <code>LinkedList</code> : Insertion Order <p>Synchronization :</p> <ul style="list-style-type: none"> • <code>ArrayList</code> : Not synchronized • <code>Vector</code> : Synchronized • <code>LinkedList</code> : Not synchronized <p>Performance :</p> <ul style="list-style-type: none"> • ArrayList : Insertion -> $O(1)$ (if insertion causes restructuring of internal array, it will be $O(n)$), Removal -> $O(1)$ (if removal causes restructuring of internal array, it will be $O(n)$), Retrieval -> $O(1)$ • Vector : Similar to <code>ArrayList</code> but little slower because of synchronization. • LinkedList : Insertion -> $O(1)$, Removal -> $O(1)$, Retrieval -> $O(n)$ <p>When to use?</p> <ul style="list-style-type: none"> • ArrayList : Use it when more search operations are needed than insertion and removal. • Vector : Use it when you need synchronized list. • LinkedList : Use it when insertion and removal are needed frequently. 	<p>Intro :</p> <ul style="list-style-type: none"> • Queue is a data structure where elements are added from one end called tail of the queue and elements are removed from another end called head of the queue. • Queue is typically FIFO (First-In-First-Out) type of data structure. <p>Popular Implementations :</p> <ul style="list-style-type: none"> • <code>PriorityQueue</code>, <code>ArrayDeque</code> and <code>LinkedList</code> (implements List also) <p>Internal Structure :</p> <ul style="list-style-type: none"> • PriorityQueue : It internally uses re-sizeable array to store the elements and a Comparator to place the elements in some specific order. • ArrayDeque : It internally uses re-sizeable array to store the elements. <p>Null Elements :</p> <ul style="list-style-type: none"> • <code>PriorityQueue</code> : Not allowed • <code>ArrayDeque</code> : Not allowed <p>Duplicate Elements :</p> <ul style="list-style-type: none"> • <code>PriorityQueue</code> : Yes • <code>ArrayDeque</code> : Yes <p>Order Of Elements :</p> <ul style="list-style-type: none"> • PriorityQueue : Elements are placed according to supplied Comparator or in natural order if no Comparator is supplied. • ArrayDeque : Supports both LIFO and FIFO <p>Synchronization :</p> <ul style="list-style-type: none"> • PriorityQueue : Not synchronized • ArrayDeque : Not synchronized <p>Performance :</p> <ul style="list-style-type: none"> • PriorityQueue : Insertion -> $O(\log(n))$, Removal -> $O(\log(n))$, Retrieval -> $O(1)$ • ArrayDeque : Insertion -> $O(1)$, Removal -> $O(n)$, Retrieval -> $O(1)$ <p>When to use?</p> <ul style="list-style-type: none"> • PriorityQueue : Use it when you want a queue of elements placed in some specific order. • ArrayDeque : You can use it as a queue OR as a stack. 	<p>Intro :</p> <ul style="list-style-type: none"> • Set is a linear collection of objects with no duplicates. • Set interface does not have its own methods. All its methods are inherited from <code>Collection</code> interface. It just applies restriction on methods so that duplicate elements are always avoided. <p>Popular Implementations :</p> <ul style="list-style-type: none"> • <code>HashSet</code>, <code>LinkedHashSet</code> and <code>TreeSet</code> <p>Internal Structure :</p> <ul style="list-style-type: none"> • HashSet : Internally uses <code>HashMap</code> to store the elements. • LinkedHashSet : Internally uses <code>LinkedHashMap</code> to store the elements. • TreeSet : Internally uses <code>TreeMap</code> to store the elements. <p>Null Elements :</p> <ul style="list-style-type: none"> • <code>HashSet</code> : Maximum one null element • <code>LinkedHashSet</code> : Maximum one null element. • <code>TreeSet</code> : Doesn't allow even a single null element <p>Duplicate Elements :</p> <ul style="list-style-type: none"> • <code>HashSet</code> : Not allowed • <code>LinkedHashSet</code> : Not allowed • <code>TreeSet</code> : Not allowed <p>Order Of Elements :</p> <ul style="list-style-type: none"> • <code>HashSet</code> : No order • <code>LinkedHashSet</code> : Insertion order • <code>TreeSet</code> : Elements are placed according to supplied Comparator or in natural order if no Comparator is supplied. <p>Synchronization :</p> <ul style="list-style-type: none"> • <code>HashSet</code> : Not synchronized • <code>LinkedHashSet</code> : Not synchronized • <code>TreeSet</code> : Not synchronized <p>Performance :</p> <ul style="list-style-type: none"> • <code>HashSet</code> : Insertion -> $O(1)$, Removal -> $O(1)$, Retrieval -> $O(1)$ • <code>LinkedHashSet</code> : Insertion -> $O(1)$, Removal -> $O(1)$, Retrieval -> $O(1)$ • <code>TreeSet</code> : Insertion -> $O(\log(n))$, Removal -> $O(\log(n))$, Retrieval -> $O(\log(n))$ <p>When to use?</p> <ul style="list-style-type: none"> • HashSet : Use it when you want only unique elements without any order. • LinkedHashSet : Use it when you want only unique elements in insertion order. • TreeSet : Use it when you want only unique elements in some specific order. 	<p>Intro :</p> <ul style="list-style-type: none"> • Map stores the data in the form of key-value pairs where each key is associated with a value. • Map interface is part of Java collection framework but it doesn't inherit <code>Collection</code> interface. <p>Popular Implementations :</p> <ul style="list-style-type: none"> • <code>HashMap</code>, <code>LinkedHashMap</code> And <code>TreeMap</code> <p>Internal Structure :</p> <ul style="list-style-type: none"> • HashMap : It internally uses an array of buckets where each bucket internally uses linked list to hold the elements. • LinkedHashMap : Same as <code>HashMap</code> but it additionally uses a doubly linked list to maintain insertion order of elements. • TreeMap : It internally uses Red-Black tree. <p>Null Elements :</p> <ul style="list-style-type: none"> • <code>HashMap</code> : Only one null key and can have multiple null values • <code>LinkedHashMap</code> : Only one null key and can have multiple null values. • <code>TreeMap</code> : Doesn't allow even a single null key but can have multiple null values. <p>Duplicate Elements :</p> <ul style="list-style-type: none"> • <code>HashMap</code> : Doesn't allow duplicate keys but can have duplicate values. • <code>LinkedHashMap</code> : Doesn't allow duplicate keys but can have duplicate values. • <code>TreeMap</code> : Doesn't allow duplicate keys but can have duplicate values. <p>Order Of Elements :</p> <ul style="list-style-type: none"> • <code>HashMap</code> : No Order • <code>LinkedHashMap</code> : Insertion Order • <code>TreeMap</code> : Elements are placed according to supplied Comparator or in natural order of keys if no Comparator is supplied. <p>Synchronization :</p> <ul style="list-style-type: none"> • <code>HashMap</code> : Not synchronized • <code>LinkedHashMap</code> : Not synchronized • <code>TreeMap</code> : Not Synchronized <p>Performance :</p> <ul style="list-style-type: none"> • <code>HashMap</code> : Insertion -> $O(1)$, Removal -> $O(1)$, Retrieval -> $O(1)$ • <code>LinkedHashMap</code> : Insertion -> $O(1)$, Removal -> $O(1)$, Retrieval -> $O(1)$ • <code>TreeMap</code> : Insertion -> $O(\log(n))$, Removal -> $O(\log(n))$, Retrieval -> $O(\log(n))$ <p>When to use?</p> <ul style="list-style-type: none"> • HashMap : Use it if you want only key-value pairs without any order. • LinkedHashMap : Use it if you want key-value pairs in insertion order. • TreeMap : Use it when you want key-value pairs sorted in some specific order.

Methods in Collection Interface

<<interface>>
Java.util.Collection

```
int size();
boolean isEmpty();
boolean contains(Object o);
Iterator<E> iterator(); (method of Iterable)
Object[] toArray();
T[] toArray(T[] a);
boolean add(E e);
boolean remove(Object o);
boolean containsAll(Collection<?> c);
boolean addAll(Collection<? extends E> c);
boolean removeAll(Collection<?> c);
boolean retainAll(Collection<?> c);
void clear();
boolean equals(Object o);
int hashCode();
```

Big O

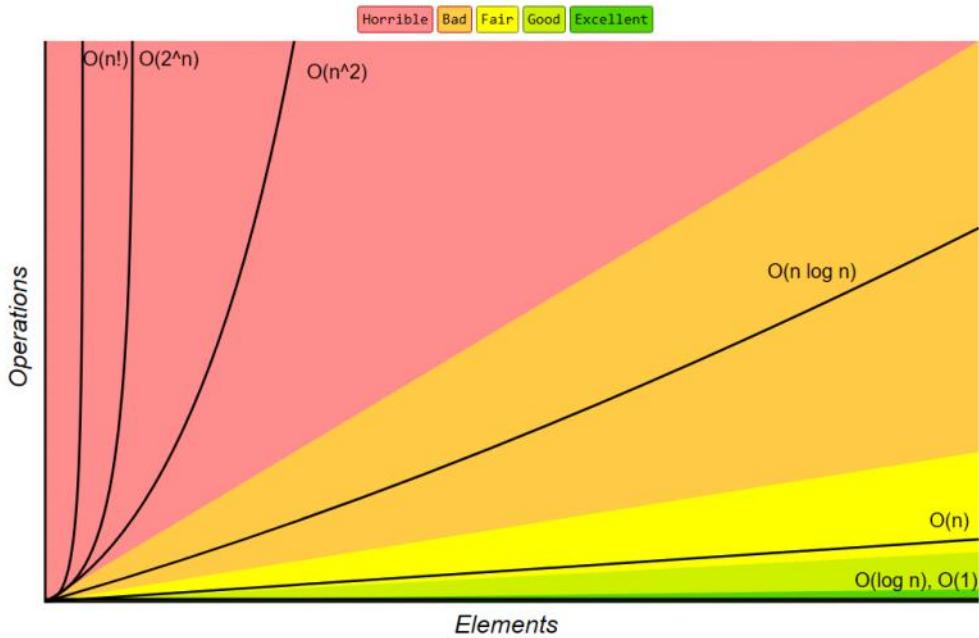
28 February 2024 08:34

Omega (Ω) – Best Case

Theta (Θ) - Average Case

Omicron-Big O (Θ) - Worst Case

Notation	Name	Description	Example
$O(1)$	Constant Time	Doesn't depend on the size of the data set.	Accessing an array element by its index.
$O(\log n)$	Logarithmic Time: Divide and Conquer	Splits the data in each step (divide and conquer).	Binary search.
$O(n)$	Linear Time: Propotional	Directly proportional to the data set size.	Looping through an array.
$O(n \log n)$	Linearithmic Time	Splits and sorts or searches data.	Merge sort, quick sort.
$O(n^2)$	Polynomial Time: Loop within Loop	Nested loops for each power of n.	Bubble sort ($O(n^2)$).



Data Structure	Time Complexity								Space Complexity	
	Average				Worst					
	Access	Search	Insertion	Deletion	Access	Search	Insertion	Deletion		
Array	O(1)	O(n)	O(n)	O(n)	O(1)	O(n)	O(n)	O(n)	O(n)	
Stack	O(n)	O(n)	O(1)	O(1)	O(n)	O(n)	O(1)	O(1)	O(n)	
Queue	O(n)	O(n)	O(1)	O(1)	O(n)	O(n)	O(1)	O(1)	O(n)	
Singly-Linked List	O(n)	O(n)	O(1)	O(1)	O(n)	O(n)	O(1)	O(1)	O(n)	
Doubly-Linked List	O(n)	O(n)	O(1)	O(1)	O(n)	O(n)	O(1)	O(1)	O(n)	
Skip List	O(log(n))	O(log(n))	O(log(n))	O(log(n))	O(n)	O(n)	O(n)	O(n)	O(n log(n))	
Hash Table	N/A	O(1)	O(1)	O(1)	N/A	O(n)	O(n)	O(n)	O(n)	
Binary Search Tree	O(log(n))	O(log(n))	O(log(n))	O(log(n))	O(n)	O(n)	O(n)	O(n)	O(n)	
Cartesian Tree	N/A	O(log(n))	O(log(n))	O(log(n))	N/A	O(n)	O(n)	O(n)	O(n)	
B-Tree	O(log(n))	O(log(n))	O(log(n))	O(log(n))	O(log(n))	O(log(n))	O(log(n))	O(log(n))	O(n)	
Red-Black Tree	O(log(n))	O(log(n))	O(log(n))	O(log(n))	O(log(n))	O(log(n))	O(log(n))	O(log(n))	O(n)	
Splay Tree	N/A	O(log(n))	O(log(n))	O(log(n))	N/A	O(log(n))	O(log(n))	O(log(n))	O(n)	
AVL Tree	O(log(n))	O(log(n))	O(log(n))	O(log(n))	O(log(n))	O(log(n))	O(log(n))	O(log(n))	O(n)	
KD Tree	O(log(n))	O(log(n))	O(log(n))	O(log(n))	O(n)	O(n)	O(n)	O(n)	O(n)	

Array Sorting Algorithms

Algorithm	Time Complexity			Space Complexity
	Best	Average	Worst	
Quicksort	$\Omega(n \log(n))$	$\Theta(n \log(n))$	$O(n^2)$	$O(\log(n))$
Mergesort	$\Omega(n \log(n))$	$\Theta(n \log(n))$	$O(n \log(n))$	$O(n)$
Timsort	$\Omega(n)$	$\Theta(n \log(n))$	$O(n \log(n))$	$O(n)$
Heapsort	$\Omega(n \log(n))$	$\Theta(n \log(n))$	$O(n \log(n))$	$O(1)$
Bubble Sort	$\Omega(n)$	$\Theta(n^2)$	$O(n^2)$	$O(1)$
Insertion Sort	$\Omega(n)$	$\Theta(n^2)$	$O(n^2)$	$O(1)$
Selection Sort	$\Omega(n^2)$	$\Theta(n^2)$	$O(n^2)$	$O(1)$
Tree Sort	$\Omega(n \log(n))$	$\Theta(n \log(n))$	$O(n^2)$	$O(n)$
Shell Sort	$\Omega(n \log(n))$	$\Theta(n(\log(n))^2)$	$O(n(\log(n))^2)$	$O(1)$
Bucket Sort	$\Omega(n+k)$	$\Theta(n+k)$	$O(n^2)$	$O(n)$
Radix Sort	$\Omega(nk)$	$\Theta(nk)$	$O(nk)$	$O(n+k)$
Counting Sort	$\Omega(n+k)$	$\Theta(n+k)$	$O(n+k)$	$O(k)$
Cubesort	$\Omega(n)$	$\Theta(n \log(n))$	$O(n \log(n))$	$O(n)$

Pointers

27 March 2024 10:59

```
int num1 = 11;
int num2 = num1;
System.out.println(num1); // 11
System.out.println(num2); // 11

num1 = 22;
System.out.println(num1); // 22
System.out.println(num2); // 11

Map<String, Integer> hm1 = new HashMap<>();
Map<String, Integer> hm2 = new HashMap<>();
hm1.put("Value", 1);
hm2 = hm1;

System.out.println(hm1); // {Value=1}
System.out.println(hm2); // {Value=1}

hm1.put("Value", 2); // or hm2.put("Value", 2);

System.out.println(hm1); // {Value=2}
System.out.println(hm2); // {Value=2}
```



- In the non-pointer example, num1 and num2 are independent variables. Changing num1 does not affect num2.
- In the pointer example, hm1 and hm2 reference the same HashMap object. Changes to the map via hm1 affect hm2 and vice versa.

List Interface

02 June 2021 09:34

Stores ordered collection of objects with possible duplicates.

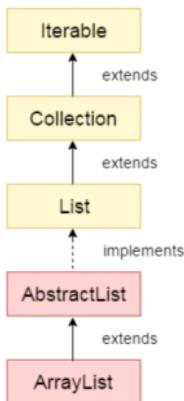
Additional Methods Of List:

```
E get(int index);  
E set(int index, E element);  
void add(int index, E element);  
E remove(int index);  
int indexOf(Object o);  
int lastIndexOf(Object o);  
ListIterator<E> listIterator();  
ListIterator<E> listIterator(int index);  
List<E> subList(int fromIndex, int toIndex);
```

ArrayList	LinkedList
<ul style="list-style-type: none">• Uses dynamic array.• Maintains insertion order, non-synchronized.• Allows random access.• ArrayList is not efficient for manipulation because too much shifting is required.• ArrayList is better to store and fetch data.• ArrayList takes less memory overhead as it stores only object	<ul style="list-style-type: none">• Uses doubly linked list.• Maintains insertion order, non-synchronized.• Fast manipulation due to no shifting.• LinkedList does not provide random access.• LinkedList takes more memory overhead, as it stores the object as well as the address of that object

ArrayList

17 June 2018 11:04



- Can contain duplicate elements, Maintains insertion order, Nonsynchronized.
- Allows random access (index basis).
- Manipulation is slow due to required shifting on removal.
- Initial Virtual Capacity: 10, when this is reached new array is created using formula $n+(n/2)+1=16$.
- `ArrayList<String> list2 = new ArrayList<String>(20); //Initial Capacity=20`
- `ArrayList<String> list = new ArrayList<String>(); list.ensureCapacity(20);`
- `trimToSize()` method is used to trim the capacity of arrayList to the current size of ArrayList
- Cannot store primitive types directly; requires wrapper classes.
- Supports adding, getting, iterating, and removing elements.
- Efficient for adding/removing from end, fast access for searching.
- Positional Access: Operations based on numerical positions: `add(int index, Object o)`, `addAll(int index, Collection c)`, `remove(int index)`, `get(int index)`, `set(int index, Object new)`.
- Search: Methods for finding elements: `indexOf(Object o)`, `lastIndexOf(Object o)`.

<u>Function</u>	<u>Usage Example</u>
<code>add(E e)</code>	<code>ArrayList<String> list = new ArrayList<String>(); list.add("Apple");</code>
<code>add(int index, E element)</code>	<code>list.add(1, "Banana");</code>
<code>get(int index)</code>	<code>String item = list.get(0);</code>
<code>set(int index, E element)</code>	<code>list.set(0, "Orange");</code>
<code>remove(int index)</code>	<code>list.remove(1);</code>
<code>remove(Object o)</code>	<code>list.remove("Apple"); list.remove(Integer.valueOf(30));</code>
<code>clear()</code>	<code>list.clear();</code>
<code>size()</code>	<code>int size = list.size();</code>
<code>isEmpty()</code>	<code>boolean empty = list.isEmpty();</code>
<code>contains(Object o)</code>	<code>boolean hasApple = list.contains("Apple");</code>
<code>indexOf(Object o)</code>	<code>int index = list.indexOf("Apple");</code>
<code>lastIndexOf(Object o)</code>	<code>int lastIndex = list.lastIndexOf("Apple");</code>
<code>toArray()</code>	<code>Object[] array = list.toArray();</code>
<code>subList(int fromIndex, int toIndex)</code>	<code>List<String> subList = list.subList(1, 3);</code>
<code>addAll(Collection<? extends E> c)</code>	<code>list.addAll(anotherList);</code>
<code>addAll(int index, Collection<? extends E> c)</code>	<code>list.addAll(1, anotherList);</code>
<code>retainAll(Collection<?> c)</code>	<code>list.retainAll(anotherList);</code>
<code>removeAll(Collection<?> c)</code>	<code>list.removeAll(anotherList);</code>
<code>iterator()</code>	<code>Iterator<String> it = list.iterator(); while (it.hasNext()) { String item = it.next(); }</code>

Synchronized ArrayList	<ul style="list-style-type: none"> • Better suited for write-heavy scenarios. • Locks the entire list during read and write operations. • Can be less efficient for frequent read operations. • Requires careful synchronization to avoid deadlocks and other concurrency issues. <table border="1"> <tbody> <tr> <td>Creation</td><td><code>List<String> syncList=Collections.synchronizedList(new ArrayList<String>());</code></td></tr> <tr> <td>Adding Elements</td><td><code>syncList.add("Saurav");</code></td></tr> <tr> <td>Get 1 Element</td><td><code>String element = synchronizedList.get(index);</code></td></tr> <tr> <td>Iterating</td><td><code>for (String element : synchronizedList) { System.out.println(element); }</code></td></tr> <tr> <td>Updating</td><td><code>synchronized (synchronizedList) { synchronizedList.set(index, newValue); }</code></td></tr> <tr> <td>Removing data</td><td><code>synchronized (synchronizedList) { synchronizedList.remove("Cherry"); }</code></td></tr> </tbody> </table>	Creation	<code>List<String> syncList=Collections.synchronizedList(new ArrayList<String>());</code>	Adding Elements	<code>syncList.add("Saurav");</code>	Get 1 Element	<code>String element = synchronizedList.get(index);</code>	Iterating	<code>for (String element : synchronizedList) { System.out.println(element); }</code>	Updating	<code>synchronized (synchronizedList) { synchronizedList.set(index, newValue); }</code>	Removing data	<code>synchronized (synchronizedList) { synchronizedList.remove("Cherry"); }</code>
Creation	<code>List<String> syncList=Collections.synchronizedList(new ArrayList<String>());</code>												
Adding Elements	<code>syncList.add("Saurav");</code>												
Get 1 Element	<code>String element = synchronizedList.get(index);</code>												
Iterating	<code>for (String element : synchronizedList) { System.out.println(element); }</code>												
Updating	<code>synchronized (synchronizedList) { synchronizedList.set(index, newValue); }</code>												
Removing data	<code>synchronized (synchronizedList) { synchronizedList.remove("Cherry"); }</code>												
CopyOnWriteArrayList	For scenarios where read operations are more frequent than write operations, consider using CopyOnWriteArrayList. It's a thread-safe alternative that creates a copy of the underlying array on each write operation, ensuring thread safety without the overhead of locking the entire list.												

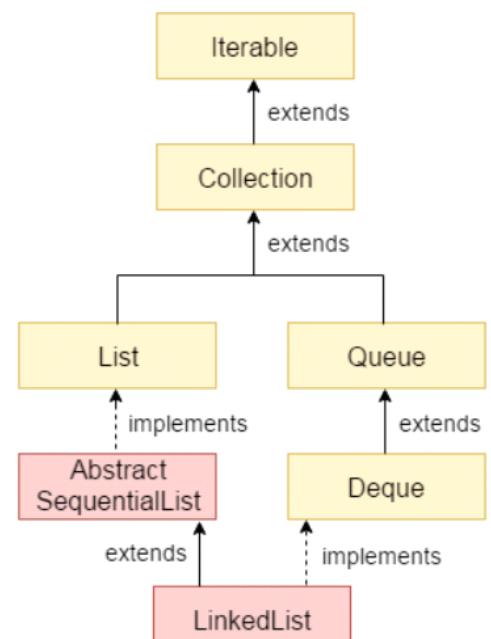
	<ul style="list-style-type: none"> Creates a copy of the underlying array on write operations. More efficient for read-heavy workloads. Can be less efficient for frequent write operations due to the overhead of copying. <pre>CopyOnWriteArrayList<String> empList=new CopyOnWriteArrayList<>(); empList.add("Tom"); empList.add("Steve"); empList.forEach(System.out::println);//No Sync Method needed, list is threadsafe list.remove("Banana");</pre>
ArrayList to Array	<pre>String[] array = new String[] {"ANDROID", "JSP", "JAVA", "STRUTS", "HADOOP", "JSF"}; ArrayList<String> list = new ArrayList<String>(Arrays.asList(array));</pre> <pre>String[] array = new String[] {"ANDROID", "JSP", "JAVA", "STRUTS", "HADOOP", "JSF"}; ArrayList<String> list = new ArrayList<String>(); Collections.addAll(list, array);</pre> <pre>List<Object> list = Arrays.stream(array).collect(Collectors.toList());</pre>
Array to ArrayList	<pre>String[] array = new String[list.size()]; list.toArray(array);</pre>

LinkedList

24 July 2018 21:25

- Linked Lists are the data structures where elements are not stored contiguously.
- Each element (node) has a data part and an address part, linked using pointers.
- Nodes are dynamic and allow easy insertions and deletions, preferred over arrays.
- Nodes cannot be accessed directly; traversal starts from the head to reach a desired node.
- Head and tail point to the first and last elements respectively. The last node points to Null.
- Head stores address of first Node.
- Can contain duplicate elements.
- Maintains insertion order.
- Non-synchronized.
- Fast manipulation as no shifting is needed.
- Can be used as a list, stack, or queue.

	Linked List	ArrayList
Append	O(1)	O(1)
Remove Last	O(n)	O(1)
Prepend	O(1)	O(n)
Remove First	O(1)	O(n)
Insert	O(n)	O(n)
Remove	O(n)	O(n)
Lookup by Index	O(n)	O(1)
Lookup by Value	O(n)	O(n)



Next has reference of next element

```
head= {  
    "value"= 11,  
    "next"= {  
        "value"= 3,  
        "next"= {  
            "value"= 23,  
            "next"= {  
                "value"= 7,  
                "next"= {  
                    "value"= 4,  
                    "next"= null  
                }  
            }  
        }  
    }  
}
```

LinkedList	<pre>public class LinkedList { private Node head, tail; private int length; public LinkedList(int value) { Node newNode = new Node(value); //Creates a New Node head = newNode; tail = newNode; length = 1; } }</pre>
------------	--

	<pre> public static class Node { int value; Node next; Node(int value) { this.value = value; } } </pre>
printList	<pre> public void printList() { Node temp = head; while (temp != null) { System.out.println("List Member Value -> " + temp.value); temp = temp.next; } } </pre>
Append	<pre> public void append(int value) { Node newNode = new Node(value); if (length == 0) { head = newNode; } else { tail.next = newNode; } tail = newNode; length++; System.out.println(value + " added to the List"); } </pre>
removeLastElement	<pre> public Node removeLastElement() { if (length == 0) return null; Node temp = head; Node pre = head; while (temp.next != null) { pre = temp; temp = temp.next; } tail = pre; tail.next = null; length--; if (length == 0) { head = null; tail = null; } return temp; } </pre>
Prepend	<pre> public void prepend(int value) { Node newNode = new Node(value); if (length == 0) { head = newNode; tail = newNode; } else { newNode.next = head; head = newNode; } length++; } </pre>
removeFirst	<pre> public Node removeFirst() { if (length == 0) { return null; } Node removedNode = head; head = head.next; removedNode.next = null; length--; } if (length == 0) { } </pre>

	<pre> tail = null; } return removedNode; } </pre>
Get	<pre> public Node get(int index) { if (index < 0 index >= length) return null; Node temp = head; for (int i = 0; i < index; i++) { temp = temp.next; } return temp; } </pre>
Set	<pre> public boolean set(int value, int index) { Node temp = get(index); if (temp != null) { temp.value = value; return true; } return false; } </pre>
Insert	<pre> public boolean insert(int value, int index) { if (index < 0 index > length) // Updated index check return false; if (index == 0) { prepend(value); return true; } if (index == length) { // Append at the end append(value); return true; } Node newNode = new Node(value); Node temp = get(index - 1); newNode.next = temp.next; temp.next = newNode; length++; return true; } </pre>
Remove	<pre> public Node remove(int index) { if (index < 0 index >= length) return null; if (index == 0) { return removeFirst(); } if (index == length - 1) { return removeLastElement(); } Node prev = get(index - 1); Node temp = prev.next; prev.next = temp.next; temp.next = null; length--; return temp; } </pre>
Reverse	<pre> public void reverse() { Node temp = head; head = tail; tail = temp; Node after; Node before = null; } </pre>

```
for (int i = 0; i < length; i++) {  
    after = temp.next;  
    temp.next = before;  
    before = temp;  
    temp = after;  
}  
}
```

Queue

02 June 2021 09:47

Queue	<p>A queue is a linear data structure that follows the First In First Out (FIFO) principle, where the element added first will be the first one to be removed. Imagine a queue in a supermarket: the first person to join the queue is the first one to be served. Cannot store null but can store duplicates.</p> <table border="1"> <thead> <tr> <th>Key Operations</th><th>Example</th></tr> </thead> <tbody> <tr> <td>Enqueue (Add)</td><td>Purpose: Adds an element to the end of the queue. Example: <code>queue.add(element)</code></td></tr> <tr> <td>Dequeue (Remove)</td><td>Purpose: Removes the element from the front of the queue. Example: <code>queue.remove()</code></td></tr> <tr> <td>Peek/Front:</td><td>Purpose: Returns the element at the front of the queue without removing it. Example: <code>queue.peek()</code></td></tr> <tr> <td>IsEmpty</td><td>Purpose: Checks if the queue is empty. Example: <code>queue.isEmpty()</code></td></tr> <tr> <td>Size</td><td>Purpose: Returns the number of elements in the queue. Example: <code>queue.size()</code></td></tr> </tbody> </table>	Key Operations	Example	Enqueue (Add)	Purpose: Adds an element to the end of the queue. Example: <code>queue.add(element)</code>	Dequeue (Remove)	Purpose: Removes the element from the front of the queue. Example: <code>queue.remove()</code>	Peek/Front:	Purpose: Returns the element at the front of the queue without removing it. Example: <code>queue.peek()</code>	IsEmpty	Purpose: Checks if the queue is empty. Example: <code>queue.isEmpty()</code>	Size	Purpose: Returns the number of elements in the queue. Example: <code>queue.size()</code>																
Key Operations	Example																												
Enqueue (Add)	Purpose: Adds an element to the end of the queue. Example: <code>queue.add(element)</code>																												
Dequeue (Remove)	Purpose: Removes the element from the front of the queue. Example: <code>queue.remove()</code>																												
Peek/Front:	Purpose: Returns the element at the front of the queue without removing it. Example: <code>queue.peek()</code>																												
IsEmpty	Purpose: Checks if the queue is empty. Example: <code>queue.isEmpty()</code>																												
Size	Purpose: Returns the number of elements in the queue. Example: <code>queue.size()</code>																												
Deque	<p>Deque interface extends the Queue interface. In Deque, we can remove and add the elements from both the side. Deque stands for a double-ended queue which enables us to perform the operations at both the ends. Allows both LIFO and FIFO. Can store null and duplicates.</p> <table border="1"> <thead> <tr> <th>Operation</th><th></th><th>Throws an exception if operation fails.</th><th>Returns null or false if operation fails.</th></tr> </thead> <tbody> <tr> <td>Insertion</td><td>Front End</td><td><code>addFirst()</code></td><td><code>offerFirst()</code></td></tr> <tr> <td></td><td>Rear End</td><td><code>addLast()</code></td><td><code>offerLast()</code></td></tr> <tr> <td>Retrieval</td><td>Front End</td><td><code>getFirst()</code></td><td><code>peekFirst()</code></td></tr> <tr> <td></td><td>Rear End</td><td><code>getLast()</code></td><td><code>peekLast()</code></td></tr> <tr> <td>Retrieval And Removal</td><td>Front End</td><td><code>removeFirst()</code></td><td><code>pollFirst()</code></td></tr> <tr> <td></td><td>Rear End</td><td><code>removeLast()</code></td><td><code>pollLast()</code></td></tr> </tbody> </table>	Operation		Throws an exception if operation fails.	Returns null or false if operation fails.	Insertion	Front End	<code>addFirst()</code>	<code>offerFirst()</code>		Rear End	<code>addLast()</code>	<code>offerLast()</code>	Retrieval	Front End	<code>getFirst()</code>	<code>peekFirst()</code>		Rear End	<code>getLast()</code>	<code>peekLast()</code>	Retrieval And Removal	Front End	<code>removeFirst()</code>	<code>pollFirst()</code>		Rear End	<code>removeLast()</code>	<code>pollLast()</code>
Operation		Throws an exception if operation fails.	Returns null or false if operation fails.																										
Insertion	Front End	<code>addFirst()</code>	<code>offerFirst()</code>																										
	Rear End	<code>addLast()</code>	<code>offerLast()</code>																										
Retrieval	Front End	<code>getFirst()</code>	<code>peekFirst()</code>																										
	Rear End	<code>getLast()</code>	<code>peekLast()</code>																										
Retrieval And Removal	Front End	<code>removeFirst()</code>	<code>pollFirst()</code>																										
	Rear End	<code>removeLast()</code>	<code>pollLast()</code>																										
PriorityQueue	<p>Elements are ordered according to their natural ordering or by a Comparator provided at queue construction time. PriorityQueue does not permit null elements. it is not synchronized Based on Complete Binary Tree Elements with higher priority are dequeued before elements with lower priority. In default PQ, smallest element will be removed 1st</p> <ul style="list-style-type: none"> • <code>PriorityQueue<Integer> pq = new PriorityQueue<>(Comparator.reverseOrder());</code> • <code>boolean add(E e):</code> Inserts the specified element into this priority queue. • <code>boolean offer(E e):</code> Inserts the specified element into this priority queue. Does not throw an exception if the insertion fails due to capacity restrictions, making it a safer option when you're not sure about the capacity. • <code>E poll():</code> Retrieves and removes the head of this queue, or returns null if this queue is empty. • <code>E peek():</code> Retrieves, but does not remove, the head of this queue, or returns null if this queue is empty. • <code>int size():</code> Returns the number of elements in this priority queue. • <code>void clear():</code> Removes all of the elements from this priority queue. • <code>Boolean remove(E e):</code> Removes a specific element from the queue. 																												

Feature	Stack	Queue	Deque (Double-ended Queue)
Data Structure Type	Linear	Linear	Linear
Principle	LIFO (Last In, First Out)	FIFO (First In, First Out)	Allows insertion and deletion from both ends
Basic Operations	<code>push()</code> (add to top), <code>pop()</code> (remove from top)	<code>enqueue()</code> (add to rear), <code>dequeue()</code> (remove from front)	Insertions: <code>enqueueFront()</code> , <code>enqueueRear()</code> ; Removals: <code>dequeueFront()</code> , <code>dequeueRear()</code>
Insertion	At the top	At the rear	At both front and rear
Deletion	From the top	From the front	From both front and rear
Use Cases	Undo mechanisms, browser history, recursive function calls	Order processing, scheduling, printers, BFS	Caching, undo-redo, palindromic checks
Direct Access	Not allowed	Not allowed	Not allowed
Storage	Typically array-based or linked list	Typically array-based or linked list	Typically array-based or linked list

Stack

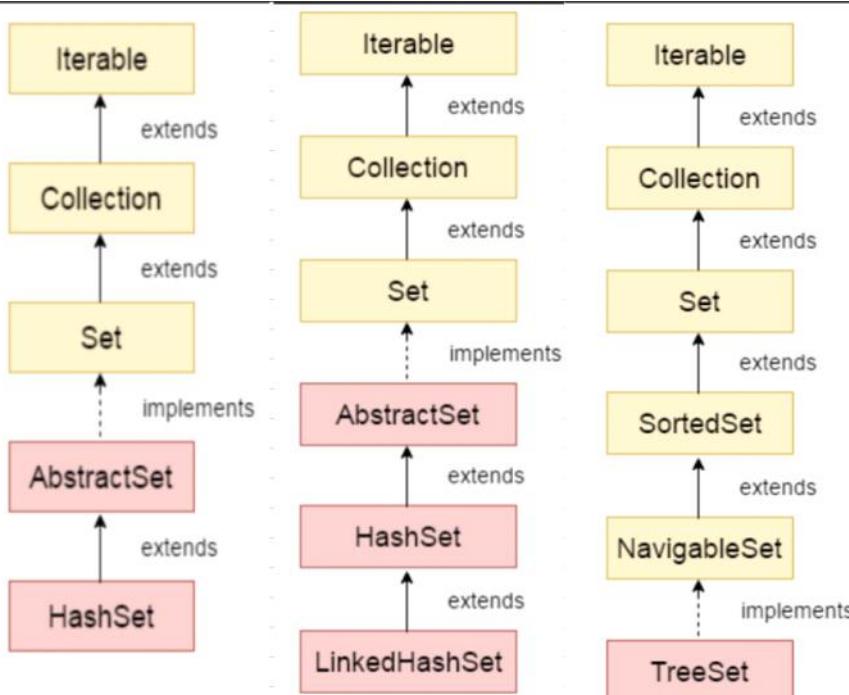
28 November 2024 09:00

A stack is a linear data structure that follows the Last In First Out (LIFO) principle. This means the last element added to the stack will be the first one to be removed. Imagine a stack of plates: you add a plate on top, and it's also the first one you take off.

Key Operations	Purpose
Push	Purpose: Adds an element to the top of the stack. Example: <code>stack.push(element)</code>
Pop	Purpose: Removes the element from the top of the stack. Example: <code>stack.pop()</code>
Peek	Purpose: Returns the element at the top of the stack without removing it. Example: <code>stack.peek()</code>
isEmpty	Purpose: Checks if the stack is empty. Example: <code>stack.isEmpty()</code>
size	Purpose: Returns the number of elements in the stack. Example: <code>stack.size()</code>

Set Interface

02 June 2021 10:39

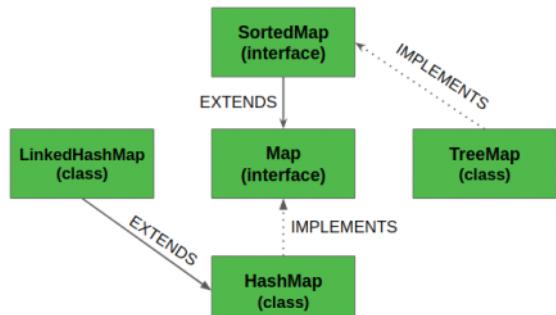
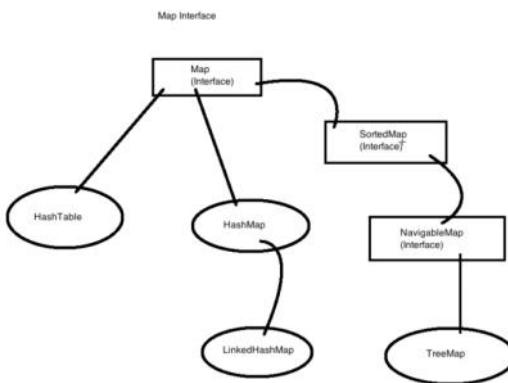


Set	Present in java.util package. It extends the Collection interface. It represents the unordered set of elements which doesn't allow us to store the duplicate items. We can store at most one null value in Set. Stores Unique Element and it will grow dynamically when elements are added to it.
HashSet	<p>Implements Set Interface. It represents the collection that uses a hash table for storage. Hashing is used to store the elements in the HashSet and contains unique items.</p> <ul style="list-style-type: none">• HashSet stores the elements by using a mechanism called hashing.• HashSet contains unique elements only.• HashSet allows 1 null value.• HashSet class is non synchronized.• HashSet doesn't maintain the insertion order. Elements are inserted on the basis of their hashCode.• HashSet is the best approach for search operations.• The initial default capacity of HashSet is 16, and the load factor is 0.75. Size doubles after that and rehashing occurs. <p>Methods in HashSet:</p> <ul style="list-style-type: none">• boolean add(E e): Used to add the specified element if it is not present, if it is present then return false.• void clear(): Used to remove all the elements from set.• boolean contains(Object o): Used to return true if an element is present in set.• boolean remove(Object o): Used to remove the element if it is present in set.• Iterator iterator(): Used to return an iterator over the element in the set.• boolean isEmpty(): Used to check whether the set is empty or not. Returns true for empty and false for non-empty condition for set.• int size(): Used to return the size of the set.• Object clone(): Used to create a shallow copy of the set. <p>HashSet uses HashMap internally to store its objects. Whenever you create a HashSet object, one HashMap object associated with it is also created. This HashMap object is used to store the elements you enter in the HashSet. The elements you add into HashSet are stored as keys of this HashMap object. The value associated with those keys will be a constant and the values will be a constant called PRESENT.</p> <p>Whenever we create a HashSet, it internally creates a HashMap and if we insert an element into this HashSet using add() method, it actually calls put() method on internally created HashMap object with element you have specified as its key and constant Object called "PRESENT" as its value. So we can say that a Set achieves uniqueness internally through HashMap.</p> <p>As we know in a HashMap each key is unique and when we call put(Key, Value) method, it returns the previous value associated with key, or null if there was no mapping for key. So in add() method we check the return value of map.put(key, value) method with null value.</p>

LinkedHashSet	<p>Represents the LinkedList implementation of Set Interface. It extends the HashSet class and implements Set interface. Like HashSet, It also contains unique elements. It maintains the insertion order and permits null elements. The insertion order of elements into LinkedHashMap are maintained by adding two new fields to this class. They are before and after. These two fields hold the references to previous and next elements. These two fields make LinkedHashMap to function as a doubly linked list.</p> <p>Java LinkedHashSet class is a Hashtable and doubly Linked list implementation of the set interface. It inherits HashSet class and implements Set interface.</p> <ul style="list-style-type: none"> • Java LinkedHashSet class contains unique elements only like HashSet. • Allows null element. • Java LinkedHashSet class provides all optional set operation and permits null elements. • Java LinkedHashSet class is non synchronized. • Java LinkedHashSet class maintains insertion order.
SortedSet	Alternate of Set interface that provides a total ordering on its elements. The elements of the SortedSet are arranged in the increasing (ascending) order. The SortedSet provides the additional methods that inhibit the natural ordering of the elements.
TreeSet	<p>Implements the Set interface and sorted set interface that uses a tree for storage. Like HashSet, TreeSet also contains unique elements. However, the access and retrieval time of TreeSet is quite fast logn. The elements in TreeSet stored in ascending order.</p> <ul style="list-style-type: none"> • Java TreeSet class contains unique elements only like HashSet. • Java TreeSet class access and retrieval times are quiet fast. • Java TreeSet class doesn't allow null element. • Java TreeSet class is non synchronized. • Java TreeSet class maintains ascending order.

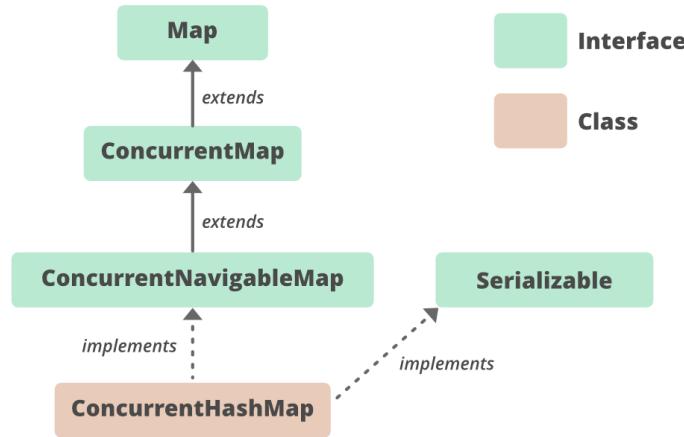
Map Interface

12 February 2019 22:38

Introduction	 <p>MAP Hierarchy in Java</p> <pre> graph TD SM[SortedMap
(Interface)] -- EXTENDS --> M[Map
(Interface)] M -- IMPLEMENTS --> TM[TreeMap
(class)] M -- EXTENDS --> LHM[LinkedHashMap
(class)] M -- IMPLEMENTS --> HM[HashMap
(class)] </pre>  <pre> graph TD MI[Map
(Interface)] --> HT([HashTable]) MI --> HM([HashMap]) HM --> LHM([LinkedHashMap]) MI --> SM[SortedMap
(Interface)] SM --> NM[NavigableMap
(Interface)] NM --> TM([TreeMap]) </pre> <p>A `Map` in Java stores key-value pairs, with each pair known as an entry. Keys in a Map must be unique. Maps are ideal for searching, updating, or deleting elements based on a key.</p> <p>Key Characteristics of Map Interface:</p> <ul style="list-style-type: none"> • Uniqueness: Each key is unique and maps to at most one value. • Null Handling: HashMap and LinkedHashMap allow null keys and values; TreeMap does not. • Order: The order of entries depends on the implementation (TreeMap and LinkedHashMap have predictable orders, HashMap does not). • Interfaces and Classes: Implemented via Map and SortedMap interfaces, and HashMap, TreeMap, and LinkedHashMap classes. • Traversal: Maps cannot be directly traversed; use keySet() or entrySet() to convert them to a Set. 														
Printing Values	<pre> for (Map.Entry<String, Integer> entry : map.entrySet()) { System.out.println("Key: " + entry.getKey() + ", Value: " + entry.getValue()); } map.forEach((key, value) -> System.out.println("Key: " + key + ", Value: " + value)); </pre>														
Common Methods	<table border="1"> <thead> <tr> <th data-bbox="365 1608 579 1653">Method</th><th data-bbox="579 1608 1569 1653">Action Performed</th></tr> </thead> <tbody> <tr> <td data-bbox="365 1653 579 1720">clear()</td><td data-bbox="579 1653 1569 1720">This method is used in Java Map Interface to clear and remove all of the elements or mappings from a specified Map collection.</td></tr> <tr> <td data-bbox="365 1720 579 1832">containsKey(Object)</td><td data-bbox="579 1720 1569 1832">This method is used in Map Interface in Java to check whether a particular key is being mapped into the Map or not. It takes the key element as a parameter and returns True if that element is mapped in the map.</td></tr> <tr> <td data-bbox="365 1832 579 1922">containsValue(Object)</td><td data-bbox="579 1832 1569 1922">This method is used in Map Interface to check whether a particular value is being mapped by a single or more than one key in the Map. It takes the value as a parameter and returns True if that value is mapped by any of the keys in the map.</td></tr> <tr> <td data-bbox="365 1922 579 2012">entrySet()</td><td data-bbox="579 1922 1569 2012">This method is used in Map Interface in Java to create a set out of the same elements contained in the map. It basically returns a set view of the map or we can create a new set and store the map elements into them.</td></tr> <tr> <td data-bbox="365 2012 579 2080">equals(Object)</td><td data-bbox="579 2012 1569 2080">This method is used in Java Map Interface to check for equality between two maps. It verifies whether the elements of one map passed as a parameter is equal to the elements of this map or not.</td></tr> <tr> <td data-bbox="365 2080 579 2129">get(Object)</td><td data-bbox="579 2080 1569 2129">This method is used to retrieve or fetch the value mapped by a particular key mentioned in the parameter. It returns NULL when the map contains no such mapping for the key.</td></tr> </tbody> </table>	Method	Action Performed	clear()	This method is used in Java Map Interface to clear and remove all of the elements or mappings from a specified Map collection.	containsKey(Object)	This method is used in Map Interface in Java to check whether a particular key is being mapped into the Map or not. It takes the key element as a parameter and returns True if that element is mapped in the map.	containsValue(Object)	This method is used in Map Interface to check whether a particular value is being mapped by a single or more than one key in the Map. It takes the value as a parameter and returns True if that value is mapped by any of the keys in the map.	entrySet()	This method is used in Map Interface in Java to create a set out of the same elements contained in the map. It basically returns a set view of the map or we can create a new set and store the map elements into them.	equals(Object)	This method is used in Java Map Interface to check for equality between two maps. It verifies whether the elements of one map passed as a parameter is equal to the elements of this map or not.	get(Object)	This method is used to retrieve or fetch the value mapped by a particular key mentioned in the parameter. It returns NULL when the map contains no such mapping for the key.
Method	Action Performed														
clear()	This method is used in Java Map Interface to clear and remove all of the elements or mappings from a specified Map collection.														
containsKey(Object)	This method is used in Map Interface in Java to check whether a particular key is being mapped into the Map or not. It takes the key element as a parameter and returns True if that element is mapped in the map.														
containsValue(Object)	This method is used in Map Interface to check whether a particular value is being mapped by a single or more than one key in the Map. It takes the value as a parameter and returns True if that value is mapped by any of the keys in the map.														
entrySet()	This method is used in Map Interface in Java to create a set out of the same elements contained in the map. It basically returns a set view of the map or we can create a new set and store the map elements into them.														
equals(Object)	This method is used in Java Map Interface to check for equality between two maps. It verifies whether the elements of one map passed as a parameter is equal to the elements of this map or not.														
get(Object)	This method is used to retrieve or fetch the value mapped by a particular key mentioned in the parameter. It returns NULL when the map contains no such mapping for the key.														

	<p><code>hashCode()</code></p> <p>This method is used in Map Interface to generate a hashCode for the given map containing keys and values.</p>
	<p><code>isEmpty()</code></p> <p>This method is used to check if a map is having any entry for key and value pairs. If no mapping exists, then this returns true.</p>
	<p><code>keySet()</code></p> <p>This method is used in Map Interface to return a Set view of the keys contained in this map. The set is backed by the map, so changes to the map are reflected in the set, and vice-versa.</p>
	<p><code>put(Object, Object)</code></p> <p>This method is used in Java Map Interface to associate the specified value with the specified key in this map.</p>
	<p><code>putAll(Map)</code></p> <p>This method is used in Map Interface in Java to copy all of the mappings from the specified map to this map.</p>
	<p><code>remove(Object)</code></p> <p>This method is used in Map Interface to remove the mapping for a key from this map if it is present in the map.</p>
	<p><code>size()</code></p> <p>This method is used to return the number of key/value pairs available in the map.</p>
	<p><code>values()</code></p> <p>This method is used in Java Map Interface to create a collection out of the values of the map. It basically returns a Collection view of the values in the HashMap.</p>
	<p><code>getOrDefault(Object key, V defaultValue)</code></p> <p>Returns the value to which the specified key is mapped, or defaultValue if this map contains no mapping for the key.</p>
<code>HashMap</code>	<p>HashMap is a part of the Java collection framework. It uses a technique called Hashing. It implements the map interface. It stores the data in the pair of Key and Value. HashMap contains an array of the nodes, and the node is represented as a class. It uses an array and LinkedList data structure internally for storing Key and Value.</p> <ul style="list-style-type: none"> • Key-Value Storage: Contains values based on the key. • Unique Keys: Only unique keys are allowed. • Null Handling: May have one null key and multiple null values. • Synchronization: Non-synchronized. • Order: Does not maintain any order. • Capacity and Load Factor <ul style="list-style-type: none"> • Default initial capacity: 16. • Default load factor: 0.75. • Threshold: Increases capacity from 16 to 32 after the 12th entry. The capacity of the HashMap is doubled each time it reaches the threshold. i.e the capacity is increased to 25=32, 26=64, 27=128..... when the threshold is reached. • Hashing : A technique of converting a large String to a small String that represents the same String. A shorter value helps in indexing and faster searches <p>HashMap uses keys in the same way as an Array uses an index</p>
<code>Synchronize HashMap</code>	<code>Map<String, String> sMap = Collections.synchronizedMap(hMap);</code>
<code>Hashtable</code>	<p>Java Hashtable class implements a hashtable, which maps keys to values. It inherits Dictionary class and implements the Map interface.</p> <ul style="list-style-type: none"> • A Hashtable is an array of a list. Each list is known as a bucket. The position of the bucket is identified by calling the hashCode() method. A Hashtable contains values based on the key. • Java Hashtable class contains unique elements. • Java Hashtable class doesn't allow null key or value. • Java Hashtable class is synchronized. • The initial default capacity of Hashtable class is 11 whereas loadFactor is 0.75.
<code>LinkedHashMap</code>	<p>It is just like HashMap with the additional feature of maintaining an order of elements inserted into it. HashMap provided the advantage of quick insertion, search, and deletion but it never maintained the track and order of insertion which the LinkedHashMap provides where the elements can be accessed in their insertion order.</p>
<code>TreeMap</code>	<p>The TreeMap in Java is used to implement the Map interface and NavigableMap along with the Abstract Class. The map is sorted according to the natural ordering of its keys, or by a Comparator provided at map creation time, depending on which constructor is used. This proves to be an efficient way of sorting and storing the key-value pairs. The storing order maintained by the treemap must be consistent with equals just like any other sorted map, irrespective of the explicit comparators</p>
<code>ConcurrentHashMap</code>	<p>It is a thread-safe implementation of the Map interface. It allows multiple threads to read and write data simultaneously, without the need for locking the entire map. Unlike a regular HashMap, which is not thread-safe, ConcurrentHashMap ensures that the operations are thread-safe, making it ideal for scenarios where multiple threads need to access and modify the map concurrently.</p> <ul style="list-style-type: none"> • Provides thread-safe operations without locking the entire map. • Allows multiple threads to operate concurrently by dividing the map into segments. • Supports atomic operations like <code>putIfAbsent()</code>, <code>replace()</code> and <code>remove()</code> • The default concurrency level of ConcurrentHashMap is 16 • Inserting null objects is not possible in ConcurrentHashMap as a key or value.





It implements `Serializable`, `ConcurrentMap<K, V>`, `Map<K, V>` interfaces and extends `AbstractMap<K, V>` class.

Advantages

- Thread-safe: `ConcurrentHashMap` is designed to be used by multiple threads simultaneously, making it an ideal choice for applications that need to handle concurrent access to data.
- Fine-grained locking: Unlike other synchronization mechanisms that lock the entire data structure, `ConcurrentHashMap` uses fine-grained locking to lock only the portion of the map being modified. This makes it highly scalable and efficient for concurrent operations.
- Atomic operations: `ConcurrentHashMap` provides several methods for performing atomic operations, such as `putIfAbsent()`, `replace()`, and `remove()`, which can be useful for implementing complex concurrent algorithms.
- High performance: Due to its fine-grained locking mechanism, `ConcurrentHashMap` is able to achieve high performance, even under heavy concurrent access.

Disadvantages

- Higher memory overhead: The fine-grained locking mechanism used by `ConcurrentHashMap` requires additional memory overhead compared to other synchronization mechanisms.
- Complexity: The fine-grained locking mechanism used by `ConcurrentHashMap` can make the code more complex, especially for developers who are not familiar with concurrent programming.

Working of HashMap

03 June 2021 11:39

Internally HashMap contains an array of Node and a node is represented as a class that contains 4 fields. It can be seen that the node contains a reference to its object. So it's a linked list.

1. int hash
2. K key
3. V value
4. Node next

Node<K,V>
int hash K key V value Node<K,V> next

The whole HashMap data structure is based on the principle of Hashing. Hashing is nothing but the function or algorithm or method which when applied on any object/variable returns an unique integer value representing that object/variable. This unique integer value is called hash code. Hash function or simply hash said to be the best if it returns the same hash code each time it is called on the same object. Two objects can have same hash code.

Before understanding the internal working of HashMap, you must be aware of hashCode() and equals() method.

- **equals():** It checks the equality of two objects. It compares the Key, whether they are equal or not. It is a method of the Object class. It can be overridden. If you override the equals() method, then it is mandatory to override the hashCode() method. HashMap uses equals() to compare the key to whether they are equal or not. If the equals() method return true, they are equal otherwise not equal.
- **hashCode():** This is the method of the object class. hashCode() method is used to get the hash code of an object. hashCode() method of the object class returns the memory reference of an object in integer form. In HashMap, hashCode() is used to calculate the bucket and therefore calculate the index.
- **Buckets:** Array of the node is called buckets. Each node has a data structure like a LinkedList. More than one node can share the same bucket. It may be different in capacity.

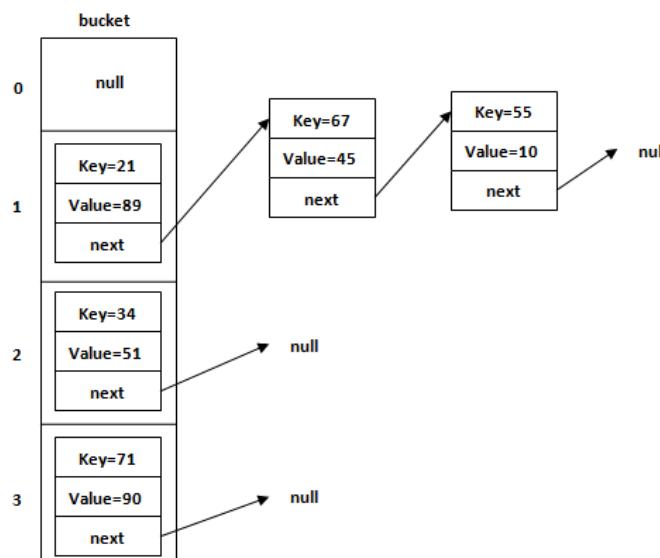


Figure: Allocation of nodes in Bucket

Insert Key, Value pair in HashMap

We use put() method to insert the Key and Value pair in the HashMap. The default size of HashMap is 16 (0 to 15).

```
HashMap<String, Integer> map = new HashMap<>();  
map.put("Aman", 19);  
map.put("Sunny", 29);  
map.put("Ritesh", 39);
```

When we call the put() method, then it calculates the hash code of the Key "Aman." Suppose the hash code of "Aman" is 2657860 To store the Key in memory, we have to calculate the index.

Calculating Index

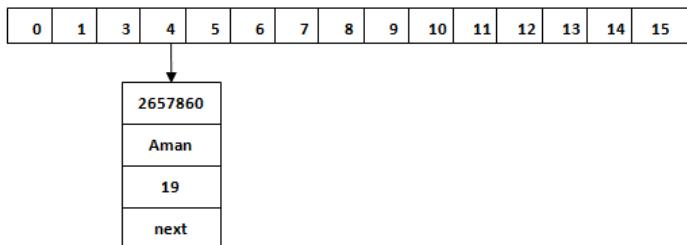
Index minimizes the size of the array. The Formula for calculating the index is:

Index = $\text{hashcode}(\text{Key}) \% (\text{n}-1)$ // n is size of hashmap

Where n is the size of the array. Hence the index value for "Aman" is:

Index = $2657860 \% (16-1) = 4$

The value 4 is the computed index value where the Key and value will store in HashMap.

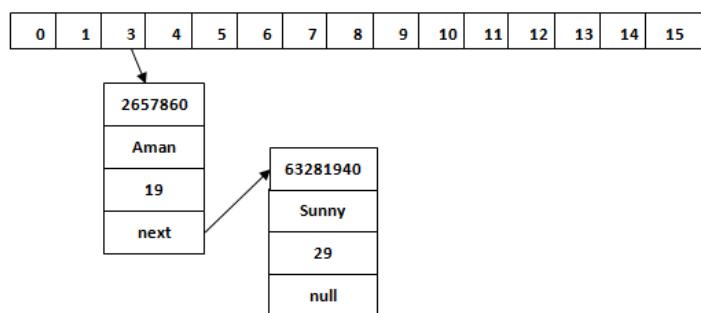


Hash Collision

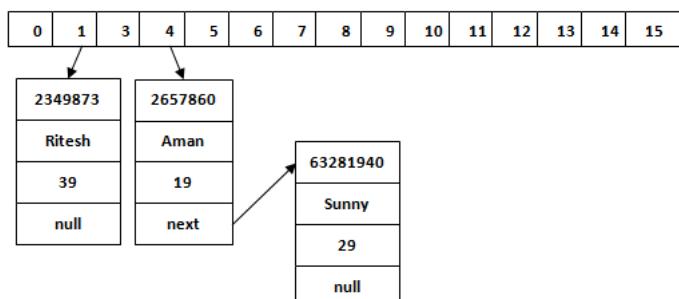
This is the case when the calculated index value is the same for two or more Keys. Let's calculate the hash code for another Key "Sunny." Suppose the hash code for "Sunny" is 63281940. To store the Key in the memory, we have to calculate index by using the index formula.

Index= $63281940 \% (16-1) = 4$

The value 4 is the computed index value where the Key will be stored in HashMap. In this case, equals() method check that both Keys are equal or not. If Keys are same, replace the value with the current value. Otherwise, connect this node object to the existing node object through the LinkedList. Hence both Keys will be stored at index 4.



Similarly, we will store the Key "Ritesh." Suppose hash code for the Key is 2349873. The index value will be 1. Hence this Key will be stored at index 1.



get() method in HashMap

get() method is used to get the value by its Key. It will not fetch the value if you don't know the Key. When get(K Key) method is called, it calculates the hash code of the Key.

Suppose we have to fetch the Key "Aman." The following method will be called.

```
map.get(new Key("Aman"));
```

It generates the hash code as 2657860. Now calculate the index value of 2657860 by using index formula. The index value will be 4, as we have calculated above. get() method search for the index value 4. It compares the first element Key with the given Key. If both keys are equal, then it returns the value else check for the next element in the node if it exists. In our scenario, it is found as the first element of the node and return the value 19.

Let's fetch another Key "Sunny."

The hash code of the Key "Sunny" is 63281940. The calculated index value of 63281940 is 4, as we have calculated for put() method. Go to index 4 of the array and compare the first element's Key with the given Key. It also compares Keys. In our scenario, the given Key is the second element, and the next of the node is null. It compares the second element Key with the specified Key and returns the value 29. It returns null if the next of the node is null.

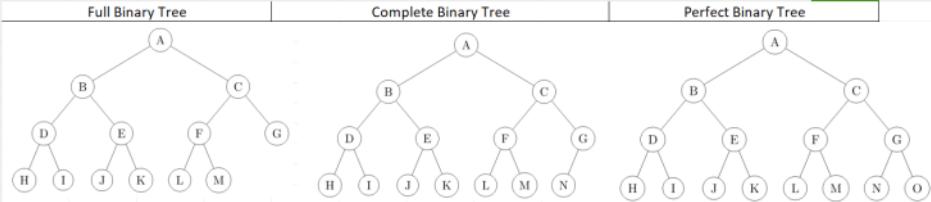
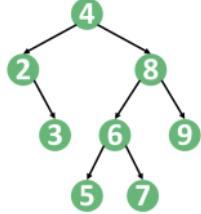
Interview

02 March 2019 08:36

Collection	Collection is a root level interface of the Java Collection Framework. It is a set of classes and interfaces designed for storing and manipulating object data. It includes classes like ArrayList, Vector, Stack, HashSet, and interfaces like List, Queue, and Set.								
Collections	Collections is an utility class in java.util package. It consists of only static methods which are used to operate on objects of type Collection. <ul style="list-style-type: none">- Collections.max():This method returns maximum element in the specified collection.- Collections.min():This method returns minimum element in the given collection.- Collections.sort():This method sorts the specified collection.- Collections.copy():This method copies all elements from one collection to another collection.- Collections.reverse():This method reverses the order of elements in the specified collection.								
Array to List	new ArrayList<>(Arrays.asList(array));								
List to Array	String[] array = fruitList.toArray(new String[fruitList.size()]);								
Iterator vs ListIterator	<table border="1"><thead><tr><th>Iterator</th><th>ListIterator</th></tr></thead><tbody><tr><td>Forward direction only</td><td>Forward and backward direction</td></tr><tr><td>Used in List, Set, Queue</td><td>Used in List only</td></tr><tr><td>Can perform remove operation</td><td>Can perform add, remove, and set operations</td></tr></tbody></table>	Iterator	ListIterator	Forward direction only	Forward and backward direction	Used in List, Set, Queue	Used in List only	Can perform remove operation	Can perform add, remove, and set operations
Iterator	ListIterator								
Forward direction only	Forward and backward direction								
Used in List, Set, Queue	Used in List only								
Can perform remove operation	Can perform add, remove, and set operations								
Fail-fast	The Iterator in Java that immediately throws ConcurrentModificationException if any structural modification occurs is called a Fail-fast iterator. It does not require extra space in memory.								
Synchronizing	Collections class provides methods to make List, Set, or Map elements synchronized: <ul style="list-style-type: none">- public static List synchronizedList(List l){}- public static Set synchronizedSet(Set s){}- public static SortedSet synchronizedSortedSet(SortedSet s){}- public static Map synchronizedMap(Map m){}- public static SortedMap synchronizedSortedMap(SortedMap m){}								
Length of Array vs. size of ArrayList:	Array length is obtained using the property of length; ArrayList uses size() method.								
Reverse ArrayList	Use Collections.reverse() method.								
Sort ArrayList in descending order	Use Collections.sort() method with Collections.reverseOrder() comparator								

BST

24 November 2024 09:18

Tree	<p>It is a data structure that is used to represent the data in hierarchical form. It can be defined as a collection of objects or entities called as nodes that are linked together to simulate a hierarchy. Tree is a non-linear data structure as the data in a tree is not stored linearly or sequentially. Node without childred is called leaf.</p> <p>Full Tree: Every node points to either zero or two nodes. Complete Tree: Nodes are filled from left to right with no gaps. Perfect Tree: All levels are fully filled with nodes.</p> 																				
BST	<p>It follows some order to arrange the elements. In a Binary search tree, the value of left node must be smaller than the parent node, and the value of right node must be greater than the parent node. This rule is applied recursively to the left and right subtrees of the root.</p> 																				
Space Time Complexity	<table border="1"><thead><tr><th>Operations</th><th>Best case time complexity</th><th>Average case time complexity</th><th>Worst case time complexity</th><th>Space complexity</th></tr></thead><tbody><tr><td>Insertion</td><td>$O(\log n)$</td><td>$O(\log n)$</td><td>$O(n)$</td><td>$O(n)$</td></tr><tr><td>Deletion</td><td>$O(\log n)$</td><td>$O(\log n)$</td><td>$O(n)$</td><td>$O(n)$</td></tr><tr><td>Search</td><td>$O(\log n)$</td><td>$O(\log n)$</td><td>$O(n)$</td><td>$O(n)$</td></tr></tbody></table>	Operations	Best case time complexity	Average case time complexity	Worst case time complexity	Space complexity	Insertion	$O(\log n)$	$O(\log n)$	$O(n)$	$O(n)$	Deletion	$O(\log n)$	$O(\log n)$	$O(n)$	$O(n)$	Search	$O(\log n)$	$O(\log n)$	$O(n)$	$O(n)$
Operations	Best case time complexity	Average case time complexity	Worst case time complexity	Space complexity																	
Insertion	$O(\log n)$	$O(\log n)$	$O(n)$	$O(n)$																	
Deletion	$O(\log n)$	$O(\log n)$	$O(n)$	$O(n)$																	
Search	$O(\log n)$	$O(\log n)$	$O(n)$	$O(n)$																	

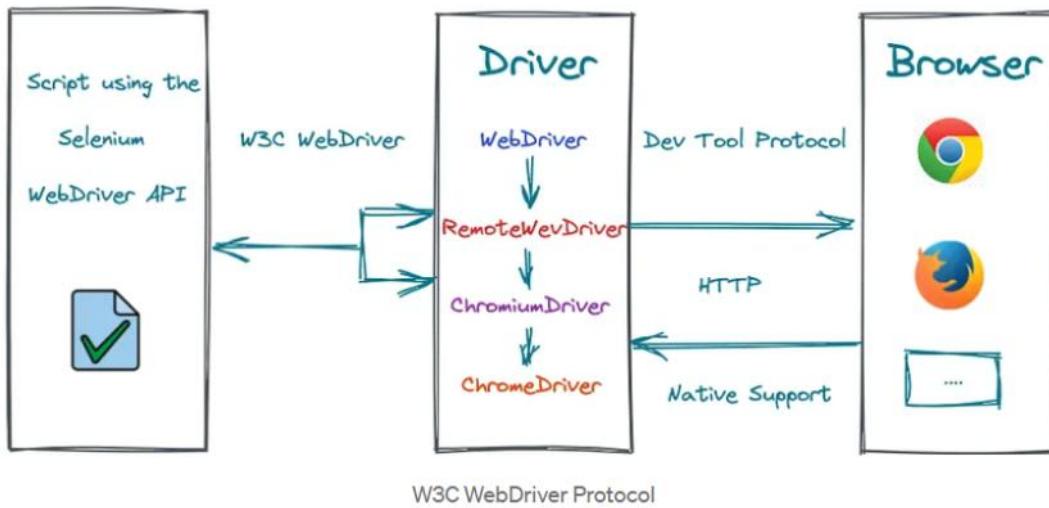
Introduction

Monday, May 21, 2018 12:32 PM

Selenium WebDriver is an open-source, cross-platform library designed to help automate browser testing. It defines set of methods but does not implements it.

How it Works

- You write your test script in your preferred language using methods like `findElement` provided by the language binding.
- The language binding translates these methods into commands that align with the WebDriver interface.
- The WebDriver implementation sends these commands to the appropriate browser driver.
- The browser driver interacts with the targeted browser to perform the actions (finding elements, clicking buttons, etc.).



W3C Webdriver

In Selenium 3.0, tests communicated with the browser at the end node through the JSON wire protocol at the local end, necessitating API encoding and decoding.

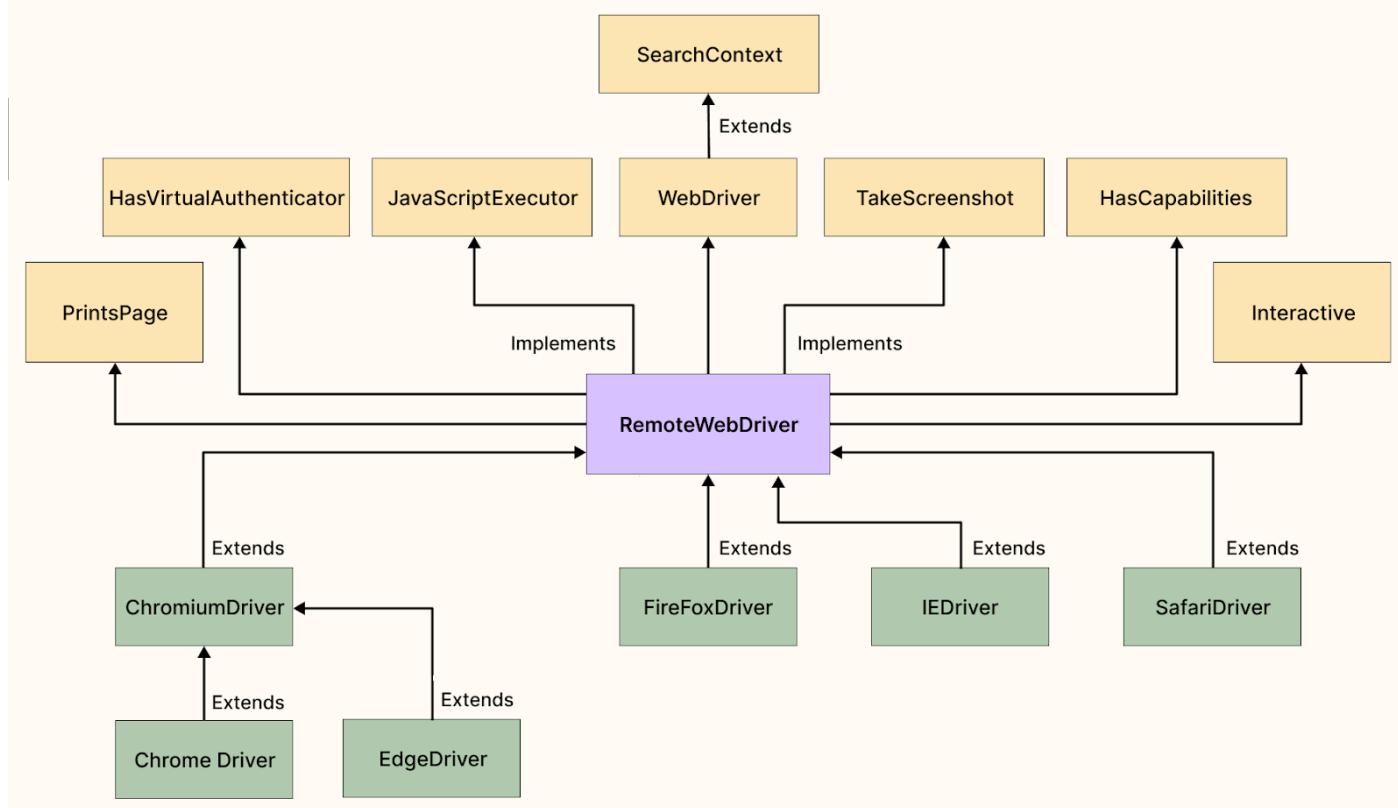
In Selenium 4, tests directly communicate through the W3C Protocol without encoding or decoding API requests. There is a direct information exchange between the client and server, without the need for the JSON Wire Protocol. As Selenium WebDriver and web browsers use the same protocol, automated Selenium testing will execute tests more consistently between different browsers. Flakiness in web automation can be considerably reduced if a standard protocol is used. JSON wire protocol was replaced with W3C protocol because browsers were following W3C standards, their executables were also compliant to W3C standards

Browser Drivers

Selenium uses drivers, specific to each browser in order to establish a secure connection with the browser without revealing the internal logic of browser's functionality.

WebDriver Hierarchy

Selenium WebDriver Hierarchy



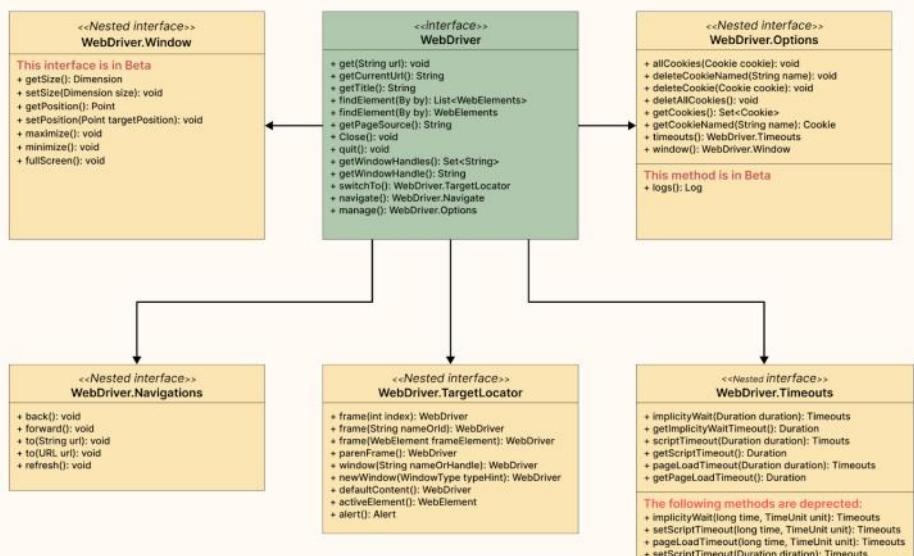
RemoteWebDriver Class It is a fully implemented WebDriver Interface class extended by every BrowserDriver class within the Selenium framework. It enables you to execute your test scripts on remote machines or browsers. Most Frequently used RemoteWebDriver Class Constructors: `RemoteWebDriver(Uri, ICapabilities)`

RemoteWebdriver class implements the following interfaces:

WebDriver Interface

WebDriver Interface is the core of the Selenium WebDriver as it has all the required methods and respective nested interfaces defined within it, which helps in simulating user actions inside the browser.

Abstract methods and Nested Interfaces in WebDriver Interface

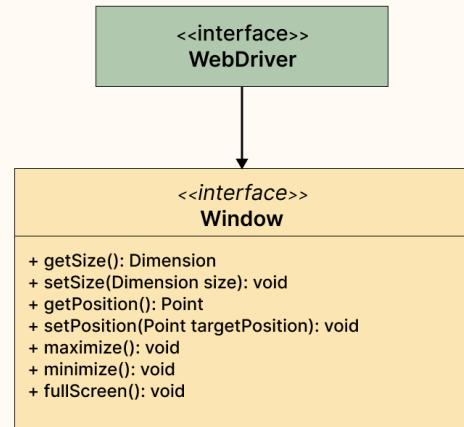


Nested Interfaces within WebDriver Interface

Window Interface

This interface has all the methods that help manage the current window.
Usage: `driver.manage().window()`.

Nested interface within WebDriver interface WebDriver.Window



Options Interface

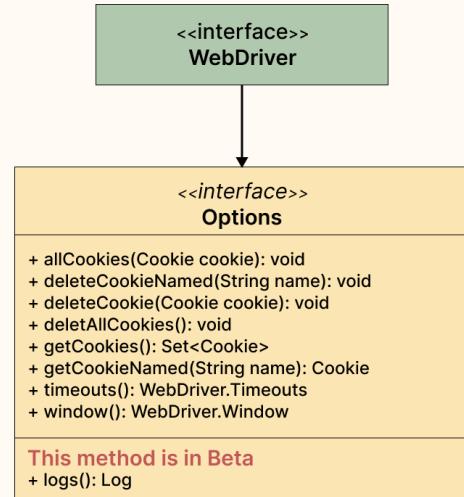
This interface has all the methods to help manage the stuff in a browser menu. With the help of this interface, we perform the following actions:

Add, get, and delete a cookie
Set timeouts in the browser
Manage window
Fetch different types of logs.

Usage: `driver.manage().window()`.

driver.manage(): Accesses the Options interface of WebDriver instance. The Options interface configures various settings for the browser session. Think of it as the place where you tweak the browser's behavior.

Nested interface within WebDriver interface WebDriver.Options



Navigation Interface

This interface has all the methods to access the browser's history and navigate to a URL. With the help of this interface, we perform the following actions:

Navigate Back, Forward in the browser
Navigate to a URL in the browser
Refresh the WebPage

Usage: `driver.navigate()`.

	<p style="text-align: center;">Nested interface within WebDriver interface WebDriver.Navigations</p> <pre> graph TD WebDriver["<<interface>> WebDriver"] Navigations["<<interface>> Navigations + back(): void + forward(): void + to(String url): void + to(URL url): void + refresh(): void"] WebDriver --> Navigations </pre>
TargetLocator Interface	<p>This interface has all the methods to send future commands to different frames and windows. With the help of this interface, we perform the following actions:</p> <p>Working with different Frames. Working with different windows or Tabs in the browser. Working with different Alerts in the browser.</p> <p>Usage: driver.switchTo().</p> <p style="text-align: center;">Nested interface within WebDriver interface WebDriver.TargetLocator</p> <pre> graph TD WebDriver["<<interface>> WebDriver"] TargetLocator["<<interface>> TargetLocator + frame(int index): WebDriver + frame(String nameOrId): WebDriver + frame(WebElement frameElement): WebDriver + parentFrame(): WebDriver + window(String nameOrHandle): WebDriver + newWindow(WindowType typeHint): WebDriver + defaultContent(): WebDriver + activeElement(): WebElement + alert(): Alert"] WebDriver --> TargetLocator </pre>
Timeouts Interface	<p>This interface has all the methods to manage the timeout behavior for WebDriver instances. With the help of this interface, we perform the following wait actions in Selenium:</p> <p>Implicit Wait Script timeout Page load timeout</p>

<h3 style="text-align: center;">Nested interface within WebDriver interface WebDriver.Timeouts</h3>	
	<div style="border: 1px solid black; padding: 10px;"> <div style="background-color: #c8e6c9; border-bottom: 1px solid black; padding-bottom: 5px; margin-bottom: 5px;"> <<interface>> WebDriver </div> <div style="border: 1px solid black; padding: 5px; background-color: #ffffcc;"> <<interface>> Timeouts </div> <div style="border: 1px solid black; padding: 5px; background-color: #ffffcc;"> + implicitWait(Duration duration): Timeouts + getImplicitWaitTimeout(): Duration + scriptTimeout(Duration duration): Timouts + getScriptTimeout(): Duration + pageLoadTimeout(Duration duration): Timeouts + getPageLoadTimeout(): Duration </div> <div style="border: 1px solid black; padding: 5px; background-color: #ffffcc;"> The following methods are deprecated: + implicitWait(long time, TimeUnit unit): Timeouts + setScriptTimeout(long time, TimeUnit unit): Timeouts + setPageLoadTimeout(long time, TimeUnit unit): Timeouts + setScriptTimeout(Duration diration): Timeouts </div> </div>
JavaScriptExecutor Interface	JavaScriptExecutor Interface provides the mechanism to WebDriver so that it can execute JavaScript code snippets
TakesScreenshot Interface	This interface helps the WebDriver take screenshots of the web page or WebElement as required and store them in different ways. The screenshot captured is returned to the WebDriver endpoint in Base64 format
HasVirtualAuthenticator Interface	This interface helps in allowing the WebDriver to access the virtual authenticator API.
PrintsPage Interface	
HasCapabilities Interface	
WebElement	<p>WebElement is an interface that extends the SearchContext and TakesScreenshot interfaces. RemoteWebElement class is the fully implemented class of the WebElement interface.</p> <div style="border: 1px solid black; padding: 10px; margin-top: 10px;"> <p style="text-align: center;">WebElement Hierarchy</p> <pre> graph TD SearchContext[SearchContext] -- Extends --> WebElement[WebElement] TakeScreenshot[TakeScreenshot] -- Extends --> WebElement RemoteWebElement[RemoteWebElement] -- Implements --> WebElement EventFiringWebElement[EventFiringWebElement] -- Implements --> WebElement </pre> </div>

Abstract methods in WebElement Interface

<<interface>> Options	
+ click(): void + submit(): void + sendKeys(CharSequence.. keysToSend): void + Clear(): void + getTagName(): String + getDomProperty(String name): String + getDomAttribute(String name): String + getAttribute(String name): String + getAriaRole(): String + getAccessibleName(): String + isSelected(): boolean + isEnabled: boolean + isDisplayed(): boolean + getText(): String + findElement(By by): List<WebElement> + findElements(By by): WebElement + getShadowRoot(): SearchContext + getLocation(): Point + getSize(): Dimension + getRect(): Rectangle + getCssValue(String propertyName): String	

Search Context	SearchContext is the topmost interface in Selenium WebDriver that provides the base for element searching. It defines the method used to find web elements on a page but does not itself provide implementation. <ul style="list-style-type: none">• It contains only two methods: findElement() and findElements().• WebDriver, WebElement, and ShadowRoot implement SearchContext.• This enables element searching at both the page and element level.
----------------	--

Locators

21 May 2018 13:03

Locators Supported by Selenium	1. ID 2. ClassName 3. Name 4. LinkText 5. Xpath 6. CssSelector 7. TagName 8. Partial LinkText
XPath Examples	- //input[@value='Log In']: replace input with * for wildcard - //td[@class='headertext'][@align='left'] - Child XPath: //*[@id='ctl00_mainContent_ddl_destinationStation1_CTNR']//a[@value='AMD'] - //input[@type='submit' and @name='submitButton'] - //input[@type='submit' or @type='button']
Traversing using XPath	- Sibling to Sibling: //*[@attribute='value']/following-sibling::li[2] - Child to Parent: //*[@attribute='value']/parent::ul
Xpath: Normalize-space()	Removes Leading and Trailing Whitespace, Collapses Consecutive Whitespace <code>//button[normalize-space(text())='Submit Form']</code>
Xpath Union Operator	<pre>List<WebElement> products = driver.findElements(By.xpath("//*[@id='product']/tbody/tr")); for (WebElement product : products) { List<WebElement> cells = product.findElements(By.xpath("./th ./td")); Other way: //*[@id='product']/tbody/tr/*[self::td or self::tr]</pre>
Xpath Axes	<p>XPath axes are used to specify the node selection relationship between nodes in an XML document. They define the direction in which the nodes are selected relative to the current node.</p> <pre> graph TD Child --> C((XPath Axes)) Parent --> C Descendant --> C Following --> C Ancestor --> C Preceding --> C PrecedingSibling --> C FollowingSibling --> C </pre>
	- WebElement lastInputField = driver.findElement(By.xpath("//input[last()-1]"));
Locating using Text	-//*[text()='Selenium'] - //a[contains(text(),'Samual')]
Xpath Indexing	- //*[@class='g'][2]
CSS Selector Examples	- input[value='Log In']: skip input for wildcard - tr[class='active'] td:nth-child(2): Accessing specific table cell using CSS Selector
Traversing using CSS	- Parent to child: parenttagname childtagname (just a space is required in CSS, whereas / is used in XPath)
Using Regular Expressions	- //input[contains(@value,'Log In')]: Contains, starts-with, ends-with -//*[contains(text(),'Assyst')] - CSS Starts with: driver.findElement(By.cssSelector("[class^='head']")) - CSS Ends with: driver.findElement(By.cssSelector("[class\$='tent']")) - CSS Contains: driver.findElements(By.cssSelector("[class*='enu']"))
Working on Console of Browser	- XPath: \$("") - CSS: \$("")
Locating Items using Compound Class	- driver.findElements(By.cssSelector(".button.btn.btn-primary.btn-xs")) (button is the tagname) - driver.findElement(By.cssSelector("div[class='cb-col cb-col-100 cb-scrd-itms']"))
Select all mobiles on flipkart of less than 10000	//div[@class="Nx9bqj_4b5DiR" and number(translate(text(),"₹","")) < 10000]
getAttribute Deprecated	<p>DOM Property: Retrieves the value of a DOM property. Properties are dynamic and can be calculated or derived from other properties or attributes. e.g. element.getDomProperty("value");</p> <p>DOM Attribute: Retrieves the value of a DOM attribute. Attributes are static and directly assigned to an element.</p>

	<pre>String textContent = element.getDomProperty("readonly");</pre>												
Advanced Xpath	<pre><?xml version="1.0" encoding="UTF-8"?> <bookstore> <book> <title lang="en">Design Patterns for High-quality Automated Tests</title> <price>9.99</price> </book> <book> <title lang="en">Testing in the Galaxy</title> <price>5.95</price> </book> </bookstore> <author>Anton Angelov</author></pre> <table border="1"> <tr> <td>//@lang</td> <td>Selects all nodes with lang attribute</td> </tr> <tr> <td>//book/title //book/price</td> <td>Selects all title and price nodes under book node</td> </tr> <tr> <td>/bookstore/book(last())</td> <td>Last book under bookstore</td> </tr> <tr> <td>/bookstore/book(last()-1)</td> <td></td> </tr> <tr> <td>/bookstore/book(position()<3)</td> <td>Selects 1st and 2nd Book</td> </tr> <tr> <td>/bookstore/book(price>35)</td> <td></td> </tr> </table>	//@lang	Selects all nodes with lang attribute	//book/title //book/price	Selects all title and price nodes under book node	/bookstore/book(last())	Last book under bookstore	/bookstore/book(last()-1)		/bookstore/book(position()<3)	Selects 1st and 2nd Book	/bookstore/book(price>35)	
//@lang	Selects all nodes with lang attribute												
//book/title //book/price	Selects all title and price nodes under book node												
/bookstore/book(last())	Last book under bookstore												
/bookstore/book(last()-1)													
/bookstore/book(position()<3)	Selects 1st and 2nd Book												
/bookstore/book(price>35)													

Waits

Wednesday, July 31, 2024 12:55 PM

Implicit Wait	<ul style="list-style-type: none">Used to set a default waiting time for the entire session. The WebDriver will wait for this duration before throwing a NoSuchElementException. Also applicable for pageLoadTimeout. Works only with findElement and findElements.<code>driver.manage().timeouts().implicitlyWait(Duration.ofSeconds(10));</code><code>driver.manage().timeouts().pageLoadTimeout(Duration.ofSeconds(30));</code>: This timeout determines how long WebDriver will wait for a page to completely load before throwing a TimeoutException<code>driver.manage().timeouts().scriptTimeout(Duration.ofSeconds(30));</code>: If a JavaScript script running in the browser takes longer than 30 seconds to complete, stop the script execution and throw an exception
Explicit Wait	<p>Allows waiting for a specific condition to be met before proceeding further. Can be customized for each element or condition. Utilizes WebDriverWait and ExpectedConditions.</p> <pre>WebDriverWait wait = new WebDriverWait(driver, Duration.ofSeconds(10)); wait.until(d->delayedText.isDisplayed());</pre> <pre>WebElement element = wait.until(ExpectedConditions.visibilityOfElementLocated(By.id("elementId"))); WebElement element = wait.until(ExpectedConditions.elementToBeClickable(By.id("clickableElementId"))); WebElement element = wait.until(ExpectedConditions.presenceOfElementLocated(By.id("elementId"))); boolean isVisible = wait.until(ExpectedConditions.invisibilityOfElementLocated(By.id("elementId"))); boolean isTextPresent = wait.until(ExpectedConditions.textToBePresentInElementLocated(By.id("elementId"), "expectedText")); boolean isAttributeValueCorrect = wait.until(ExpectedConditions.attributeToBe(By.id("elementId"), "attributeName", "expectedValue"));</pre> <p>Warning: Do not mix implicit and explicit waits. Doing so can cause unpredictable wait times. For example, setting an implicit wait of 10 seconds and an explicit wait of 15 seconds could cause a timeout to occur after 20 seconds.</p>
Fluent Wait(explicitWith Options)	<p>FluentWait in Selenium is an important class when we are dealing with AJAX elements. When a user is exploring the website on a slower network, these elements may take more time to load than what they would have taken in lab testing. To play safe, FluentWait helps us mimic such scenarios and produce the best quality web application.</p> <pre>Wait<WebDriver> wait = new FluentWait<>(driver) .withTimeout(Duration.ofSeconds(10)) .pollingEvery(Duration.ofSeconds(2)) .ignoring(NoSuchElementException.class);</pre> <pre>WebElement element = wait.until(driver > driver.findElement(By.id("elementId")));</pre> <p>Selenium 4 Enhancement: Inherits from FluentWait and provides more functionality.</p> <pre>WebDriverWait wait = new WebDriverWait(driver, Duration.ofSeconds(10)); wait.pollingEvery(Duration.ofMillis(500)); WebElement element = wait.until(driver > driver.findElement(By.id("elementId")));</pre>

Interactions

21 May 2018 14:04

Select dropdown	<pre>Select s = new Select(driver.findElement(By.name("my_html_select_box"))); s.selectByIndex(1); - Methods to Select Dropdown: 1. s.selectByValue(value) 2. s.selectByVisibleText(text) 3. s.selectByIndex(index) - Methods to Deselect Dropdown: 1. s.deselectAll() 2. s.deselectByValue(value) 3. s.deselectByVisibleText(text) 4. s.deselectByIndex(index) - Additional Dropdown Methods: - s.isMultiple(): Returns true if the dropdown allows multiple selection. - s.isSelected(): Returns true if the element is selected. - s.getOptions(): Gives a list of WebElement options. -getFirstSelectedOption() -getAllSelectedOptions() Without Using Select Class // Assuming the dropdown is a <div> element with a specific class WebElement dropdown = driver.findElement(By.cssSelector("div.dropdown-menu")); dropdown.click(); // Assuming the option is a <a> element with a specific text WebElement option = driver.findElement(By.xpath("//a[text()='Option 2']")); option.click();</pre>														
Alert	<pre>System.out.println(driver.switchTo().alert().getText()); driver.switchTo().alert().sendKeys("fesfe"); driver.switchTo().alert().accept(); //Accept = ok done yes driver.switchTo().alert().dismiss(); WebDriverWait wait = new WebDriverWait(driver, Duration.ofSeconds(10)); // Adjust timeout as needed try { Alert alert = wait.until(ExpectedConditions.alertIsPresent()); System.out.println("Alert text: " + alert.getText()); alert.accept(); // Or alert.dismiss() } catch (org.openqa.selenium.TimeoutException e) { System.out.println("Alert not found within the timeout.");}</pre>														
Actions	<pre>Actions a = new Actions(driver); <table border="1"> <tr> <td>Mouse Hover and Click</td> <td>a.moveToElement(driver.findElement(By.id("nav-link-accountList"))).click().build().perform();</td> </tr> <tr> <td>Mouse Hover, Click with Shift Key, and Typing</td> <td>a.moveToElement(driver.findElement(By.id("nav-link-accountList"))).click().keyDown(Keys.SHIFT).sendKeys("hello").build().perform();</td> </tr> <tr> <td>Keyboard Shortcuts</td> <td>a.keyDown(Keys.CONTROL).sendKeys("a").keyUp(Keys.CONTROL).perform();</td> </tr> <tr> <td>DragandDrop</td> <td>Action.clickAndHold("driver....").moveToElement("driver..").release().build().perform();</td> </tr> <tr> <td>Pause</td> <td> new Actions(driver) .moveToElement(clickable) .pause(Duration.ofSeconds(1)) .clickAndHold() .pause(Duration.ofSeconds(1)) .sendKeys("abc") .perform(); </td> </tr> <tr> <td>Release All Inputs</td> <td>((RemoteWebDriver) driver).resetInputState();</td> </tr> <tr> <td>sendKeysToActiveElement</td> <td> @Test public void sendKeysToActiveElement() { driver.get("https://www.selenium.dev/selenium/web/single_text_input.html"); new Actions(driver) .sendKeys("abc") </td> </tr> </table> </pre>	Mouse Hover and Click	a.moveToElement(driver.findElement(By.id("nav-link-accountList"))).click().build().perform();	Mouse Hover, Click with Shift Key, and Typing	a.moveToElement(driver.findElement(By.id("nav-link-accountList"))).click().keyDown(Keys.SHIFT).sendKeys("hello").build().perform();	Keyboard Shortcuts	a.keyDown(Keys.CONTROL).sendKeys("a").keyUp(Keys.CONTROL).perform();	DragandDrop	Action.clickAndHold("driver....").moveToElement("driver..").release().build().perform();	Pause	new Actions(driver) .moveToElement(clickable) .pause(Duration.ofSeconds(1)) .clickAndHold() .pause(Duration.ofSeconds(1)) .sendKeys("abc") .perform();	Release All Inputs	((RemoteWebDriver) driver).resetInputState();	sendKeysToActiveElement	@Test public void sendKeysToActiveElement() { driver.get("https://www.selenium.dev/selenium/web/single_text_input.html"); new Actions(driver) .sendKeys("abc")
Mouse Hover and Click	a.moveToElement(driver.findElement(By.id("nav-link-accountList"))).click().build().perform();														
Mouse Hover, Click with Shift Key, and Typing	a.moveToElement(driver.findElement(By.id("nav-link-accountList"))).click().keyDown(Keys.SHIFT).sendKeys("hello").build().perform();														
Keyboard Shortcuts	a.keyDown(Keys.CONTROL).sendKeys("a").keyUp(Keys.CONTROL).perform();														
DragandDrop	Action.clickAndHold("driver....").moveToElement("driver..").release().build().perform();														
Pause	new Actions(driver) .moveToElement(clickable) .pause(Duration.ofSeconds(1)) .clickAndHold() .pause(Duration.ofSeconds(1)) .sendKeys("abc") .perform();														
Release All Inputs	((RemoteWebDriver) driver).resetInputState();														
sendKeysToActiveElement	@Test public void sendKeysToActiveElement() { driver.get("https://www.selenium.dev/selenium/web/single_text_input.html"); new Actions(driver) .sendKeys("abc")														

		<pre> .perform(); WebElement textField = driver.findElement(By.id("textInput")); Assertions.assertEquals("abc", textField.getAttribute("value")); } </pre>
	sendKeysToDesignatedElement	<pre> @Test public void sendKeysToDesignatedElement() { driver.get("https://www.selenium.dev/selenium/web/single_text_input.html"); driver.findElement(By.tagName("body")).click(); WebElement textField = driver.findElement(By.id("textInput")); new Actions(driver) .sendKeys(textField, "Selenium!") .perform(); Assertions.assertEquals("Selenium!", textField.getAttribute("value")); } </pre>
	scrollToElement	<pre> WebElement iframe = driver.findElement(By.tagName("iframe")); new Actions(driver) .scrollToElement(iframe) .perform(); Scroll down by 500 pixels : actions.scrollByAmount(0, 500).perform(); </pre>
	Actions vs Action	<p>Actions is the class you use to construct a series of user interactions. Action is an interface that represents a single, completed sequence of actions built by the Actions class.</p> <pre> Actions actions = new Actions(driver); Action combinedAction = actions .moveToElement(element1) .click() .moveToElement(element2) .contextClick() .build(); combinedAction.perform(); </pre>
Window Switch		<pre> Set<String> windowHandles = driver.getWindowHandles(); List<String> windowHandlesList = new ArrayList<>(windowHandles); driver.switchTo().window(windowHandlesList.get(1)); </pre>
Frames		<pre> WebElement iframe = driver.findElement(By.tagName("iframe")); driver.switchTo().frame(iframe); driver.switchTo().frame("iframe-name"); switchTo().parentFrame(): In case of nested frames, switches the focus to the parent frame SwitchTo().defaultContent(): Switch to main page </pre>
Win Auth		driver.get("https://admin:admin@the-internet.herokuapp.com/");
Js Executor		<p>JavascriptExecutor is an interface that is used to execute JavaScript with Selenium. RemoteWebDriver implements JavaScriptExecutor.</p> <ul style="list-style-type: none"> • Use getElementById() when you know the exact ID of the element. • Use querySelector() for simple CSS selectors. • Use querySelectorAll() when you need to select multiple elements. • Use getElementsByTagName() when you know the tag name of the elements. • Use getElementsByClassName() when you know the class name of the elements.
executeScript()		This method executes the test script in the context of the currently selected window or frame
executeAsyncScript()		This method executes the asynchronous piece of JavaScript on the current window or frame. An asynchronous script will be executed while the rest of the page continues parsing, which enhances responsiveness and application performance.
Click		<pre> JavascriptExecutor js = (JavascriptExecutor) driver; js.executeScript("document.getElementById('element id ').click();"); WebElement firstYes = driver.findElement(By.xpath("//*[@id='yes']")); js.executeScript("arguments[0].click();", firstYes); js.executeScript("document.querySelector('#one').click();"); </pre>
Scroll Down		<pre> jsExecutor.executeScript("arguments[0].scrollIntoView(true);", element); // Scroll down by 1000 pixels js.executeScript("window.scrollBy(0, 1000)"); </pre>

		<pre>// Scroll up by 1000 pixels js.executeScript("window.scrollBy(0, -1000)"); // Scroll to the bottom of the page js.executeScript("window.scrollTo(0, document.body.scrollHeight)"); // Scroll to the top of the page js.executeScript("window.scrollTo(0, 0)");</pre>
	Send Text	WebElement fullName = driver.findElement(By.id("fullName")); js.executeScript("document.getElementById('fullName').value='saurav';");
	Get Text	String text = (String) js.executeScript("return arguments[0].value;", fullName); String text = (String) jsExecutor.executeScript("return arguments[0].innerText;", element);
	Type Text	js.executeScript("document.getElementById('Email').value='SoftwareTestingMaterial.com'");
File Upload		<pre>driver.get("https://the-internet.herokuapp.com/upload"); File uploadFile = new File("src/test/resources/ss.png"); WebElement fileInput = driver.findElement(By.cssSelector("input[type=file]")); fileInput.sendKeys(uploadFile.getAbsolutePath()); driver.findElement(By.id("file-submit")).click(); WebElement fileName = driver.findElement(By.id("uploaded-files")); Assert.assertEquals(fileName.getText(), "ss.png");</pre>
File Download		<pre>ChromeOptions options=new ChromeOptions(); String downloadPath = Paths.get(System.getProperty("user.dir"), "downloads").toString(); //creates downloads folder in the project Map<String, Object> prefs=newHashMap<>(); prefs.put("download.default_directory", downloadPath); prefs.put("download.prompt_for_download", false); //Disabledownloadprompt prefs.put("download.directory_upgrade", true); //Allowdirectorychanges prefs.put("safebrowsing.enabled", true); //Enablesafebrowsing options.setExperimentalOption("prefs", prefs); options.setCapability(CapabilityType.ACCEPT_INSECURE_CERTS, true); WebDriver driver=newChromeDriver(options);</pre>
Window Size		<pre>driver.manage().window().setSize(new Dimension(1024, 768)); Dimension size = driver.manage().window().getSize(); int width = size.getWidth(); int height = size.getHeight(); Rectangle rectangle = elementById("property").getRect(); System.out.println(rectangle.getDimension()); System.out.println(rectangle.getWidth() + " " + rectangle.getHeight()); Position of Window/Element Point position = driver.manage().window().getPosition(); int x = position.getX(); int y = position.getY(); System.out.println("Window Position: X = " + x + ", Y = " + y); Point point = elementById("position").getLocation(); System.out.println("X :" + point.getX()); System.out.println("Y :" + point.getY());</pre>
Cookies	Add Cookie	driver.manage().addCookie(new Cookie("key", "value"));
	Get Named Cookie	Cookie cookie = driver.manage().getCookieNamed("foo");
	Get All Cookies	Set<Cookie> cookies = driver.manage().getCookies(); for (Cookie cookie : cookies) { if (cookie.getName().equals("test1")) { Assertions.assertEquals(cookie.getValue(), "cookie1"); }}
	Delete Cookie	driver.manage().deleteCookieNamed("test1");
	Delete All Cookies	driver.manage().deleteAllCookies();

Exceptions

03 August 2021 11:34

Parent Class of All Exceptions: WebDriverException->RunTimeException->Exception->Throwable.
 Catch specific exceptions, log exception details, use asserts for testing purposes, use try-catch around risky code sections.

Exception Name	Description	Resolution
NoSuchElementException	Thrown when an operation attempts to access an element that is not present on the page.	<ul style="list-style-type: none"> Verify Locator Wait for Element Check Frame
ElementClickInterceptedException	The ElementClickInterceptedException occurs when Selenium tries to click on an element, but another element intercepts the click. This often happens when there are overlapping elements, such as pop-ups or overlays, that cover the target element	<ul style="list-style-type: none"> Scroll to the Element: <code>WebElement element = driver.findElement(By.id("elementId")); ((JavascriptExecutor) driver).executeScript("arguments[0].scrollIntoView(true);", element); element.click();</code> Use JavaScript Executor: <code>JavascriptExecutor js = (JavascriptExecutor) driver; js.executeScript("arguments[0].click()", element);</code> Check Resolution of the browser window Wait for Element to be Clickable: <code>WebElement element = wait.until(ExpectedConditions.elementToBeClickable(By.id("button")));</code>
ElementNotVisibleException	Thrown when an operation is attempted on an element that is present in the DOM but not visible on the page.	<ul style="list-style-type: none"> Scroll to the Element Use Explicit Waits Use JavaScript Executor
ElementNotSelectableException	Thrown when trying to select an element that is not selectable, typically with options in a dropdown.	<ul style="list-style-type: none"> Use JavaScript Executor Use Explicit Waits Select Correct tag i.e option instead of div
StaleElementReferenceException	<p>When we try to find an element using the <code>findElement()</code> method in WebDriver, Selenium keeps the referenceId of that element in memory if the element is found. Later when you try to interact with the element, instead of finding it again, it gets the saved referenceId and tries to locate the element with that referenceId.</p> <p>Although, there can be situations where you perform some action on the page, due to which the reference of a few elements gets updated or refreshed. In such cases, you would get <code>StaleElementReferenceException</code>.</p> <ul style="list-style-type: none"> The referenced element is not attached to the DOM anymore. The referenced element has been deleted permanently. 	<p>Solution 1- Refresh the web page</p> <pre>try { driver.findElement(By.cssSelector("webelement locator")).click(); } catch(StaleElementReferenceException e) { driver.navigate().refresh(); driver.findElement(By.cssSelector("webelement locator")).click(); }</pre> <p>Solution 2- Using PageFactory</p> <p>In POM design pattern or while using the principles of Page Factory, we locate an element using the <code>@FindBy</code> annotation. It helps in updating the reference of the web element each time before any action is performed on it. This becomes very useful in eliminating the occurrence of <code>StaleElementReferenceException</code>.</p>
TimeoutException	Indicates that an operation did not complete within the specified timeout period specified.	
NoSuchWindowException	Thrown when switching to a window that is not available or has been closed.	
NoSuchFrameException	Raised when the WebDriver is switching to a frame that is not available or does not exist.	
NoAlertPresentException	Occurs when an attempt is made to interact with an alert, but no alert is present on the page.	
InvalidSelectorException	Indicates that the provided selector is invalid or does not match a web element.	
NoSuchSessionException	Thrown when an attempt is made to interact with a session that is no longer available.	

Grid

03 April 2024 19:40

Architecture	<pre> graph LR Client[Client] --> Router[Router] Router --> SessionQueue[Session Queue] Router --> SessionMap[Session Map] SessionQueue --> Distributor[Distributor] SessionMap --> Distributor Distributor --> Nodes[Nodes] Nodes --> EventBus[Event Bus] EventBus --> SessionQueue EventBus --> SessionMap </pre>
Hub	<ul style="list-style-type: none"> The hub is the central point of control in Selenium Grid. It's typically located on your local machine where you write and trigger your Selenium tests. The hub receives test requests and then decides which node (or machine) will execute the test based on the requirements specified in the test (e.g., browser type, operating system). The hub can manage multiple nodes, directing test execution to the appropriate machine. A Hub is composed by the following components: <ul style="list-style-type: none"> Router: Receives and Forwards request to correct component Distributor: Routes test session requests to available nodes based on their capabilities. Session Map: Stores the relationship between the session id and the Node where the session is running New Session Queue: Holds all the new session requests in a FIFO order Event Bus: Communication path between the Nodes, Distributor, New Session Queue, and Session Map Hub receives user request of port 4444 Hub receives messages from node on 4442 Hub sends messages to browser on 4443
Router	<ul style="list-style-type: none"> The Router is the entry point of the Grid, receiving all external requests, and forwards them to the correct component. If the Router receives a new session request, it will be forwarded to the New Session Queue. If the request belongs to an existing session, the Router will query the Session Map to get the Node ID where the session is running, and then the request will be forwarded directly to the Node. The Router balances the load in the Grid by sending the requests to the component that is able to handle them better, without overloading any component that is not needed in the process.
Distributor	<p>The Distributor has two main responsibilities:</p> <ul style="list-style-type: none"> Register and keep track of all Nodes and their capabilities: A Node registers to the Distributor by sending a Node registration event through the Event Bus. The Distributor reads it, and then tries to reach the Node via HTTP to confirm its existence. If the request is successful, the Distributor registers the Node and keeps track of all Nodes capabilities through the GridModel. Query the New Session Queue and process any pending new session requests: When a new session request is sent to the Router, it gets forwarded to the New Session Queue, where it will wait in the queue. The Distributor will poll the New Session Queue for pending new session requests, and then finds a suitable Node where the session can be created. After the session has been created, the Distributor stores in the Session Map the relation between the session id and Node where the session is being executed
Session Map	<p>The Session Map is a data store that keeps the relationship between the session id and the Node where the session is running. It supports the Router in the process of forwarding a request to the Node. The Router will ask the Session Map for the Node associated to a session id</p>
New Session Queue	<p>The New Session Queue holds all the new session requests in a FIFO order. It has configurable parameters for setting the request timeout and request retry interval (how often the timeout will be checked).</p> <p>The Router adds the new session request to the New Session Queue and waits for the response. The New Session Queue regularly checks if any request in the queue has timed out, if so the request is rejected and removed immediately.</p> <p>The Distributor regularly checks if a slot is available. If so, the Distributor polls the New Session Queue for the first matching request. The Distributor then attempts to create a new session.</p> <p>Once the requested capabilities match the capabilities of any of the free Node slots, the Distributor attempts to get the available slot. If all the slots are busy, the Distributor will send the request back to the queue. If request times out while retrying or adding to the front of the queue, it will be rejected.</p> <p>After a session is created successfully, the Distributor sends the session information to the New Session Queue, which then gets sent back to the Router, and finally to the client.</p>
Nodes	<p>A Grid can contain multiple Nodes. Each Node manages the slots for the available browsers of the machine where it is running.</p> <p>The Node registers itself to the Distributor through the Event Bus, and its configuration is sent as part of the registration message.</p> <p>By default, the Node auto-registers all browser drivers available on the path of the machine where it runs. It also creates one slot per available CPU for Chromium based browsers and Firefox. For Safari, only one slot is created. Through a specific configuration, it can run</p>

	sessions in Docker containers or relay commands.		
	A Node only executes the received commands, it does not evaluate, make judgments, or control anything other than the flow of commands and responses. The machines where the Node is running does not need to have the same operating system as the other components. For example, A Windows Node might have the capability of offering IE Mode on Edge as a browser option, whereas this would not be possible on Linux or Mac, and a Grid can have multiple Nodes configured with Windows, Mac, or Linux		
Event Bus	The Event Bus serves as a communication path between the Nodes, Distributor, New Session Queue, and Session Map. The Grid does most of its internal communication through messages, avoiding expensive HTTP calls. When starting the Grid in its fully distributed mode, the Event Bus is the first component that should be started.		
Docker compose	<pre> docker-compose-hub.yml x-common-node: &common-node shm_size: '2gb' depends_on: - selenium-hub environment: - SE_EVENT_BUS_HOST=selenium-hub - SE_EVENT_BUS_PUBLISH_PORT=4442 - SE_EVENT_BUS_SUBSCRIBE_PORT=4443 #- SE_VNC_NO_PASSWORD=1 services: selenium-hub: image: selenium/hub:latest container_name: selenium-hub ports: - "4444:4444" - "4442:4442" - "4443:4443" networks: - selenium-network chrome: image: selenium/node-chrome:latest <<: *common-node environment: - SE_NODE_OVERRIDE_MAX_SESSIONS=true - SE_NODE_MAX_SESSIONS=2 networks: - selenium-network edge: image: selenium/node-edge:latest <<: *common-node networks: - selenium-network firefox: image: selenium/node-firefox:latest <<: *common-node networks: - selenium-network networks: selenium-network: driver: bridge </pre>		
Starting Grid	Docker	<pre> docker-compose -f docker-compose.yml up -d docker-compose -f docker-compose-node.yml down Access hub UI at localhost:4444 </pre>	
	Standalone	<pre>java -jar selenium-server-<version>.jar standalone</pre>	
	Hub and Node	Both on same machine	<pre> Hub: java -jar selenium-server-<version>.jar hub Node1: java -jar selenium-server-<version>.jar node Node2: java -jar selenium-server-<version>.jar node --port 5555 </pre>
		On Different Machines	<p>Registering Nodes when Hub listens on default port:</p> <pre>java -jar selenium-server-<version>.jar node --hub <a href="http://<hub-ip>:4444">http://<hub-ip>:4444</pre> <p>Registering Nodes when Hub listens on Non-default port:</p> <pre>java -jar selenium-server-<version>.jar hub --publish-events tcp://<hub-ip>:8886 --subscribe-events tcp://<hub-ip>:8887 --port 8888</pre>

		java -jar selenium-server-<version>.jar node --publish-events tcp://<hub-ip>:8886 --subscribe-events tcp://<hub-ip>:8887
Control number of browser instances from command line		docker-compose up --scale chrome=3 --scale firefox=2 -d
Code		<pre>ChromeOptions chromeOptions = new ChromeOptions(); chromeOptions.setCapability("browserVersion", "100"); chromeOptions.setCapability("platformName", "Windows"); // Showing a test name instead of the session id in the Grid UI chromeOptions.setCapability("se:name", "My simple test"); // Other type of metadata can be seen in the Grid UI by clicking on the // session info or via GraphQL chromeOptions.setCapability("se:sampleMetadata", "Sample metadata value"); WebDriver driver = new RemoteWebDriver(new URL("http://gridUrl:4444"), chromeOptions); driver.get("http://www.google.com"); driver.quit();</pre>

Thread Local

16 November 2021 16:56

ThreadLocal in Java is another way to achieve thread-safety. Java ThreadLocal class provides thread-local variables. It enables you to create variables that can only be read and write by the same thread. If two threads are executing the same code and that code has a reference to a ThreadLocal variable then the two threads can't see the local variable of each other.

```
// ThreadLocal to ensure each thread has its own WebDriver instance
private static final ThreadLocal<WebDriver> tlDriver = ThreadLocal.withInitial(() -> null);

public static void initDriver(String browser) {
    if (tlDriver.get() == null) {
        WebDriver driver = switch (browser.toLowerCase()) {
            case "chrome" -> new ChromeDriver();
            case "firefox" -> new FirefoxDriver();
            case "edge" -> new EdgeDriver();
            default -> throw new IllegalArgumentException("Unsupported browser: " + browser);
        };
        tlDriver.set(driver);
    }
}

public static WebDriver getDriver() {
    WebDriver driver = tlDriver.get();
    if (driver == null) {
        throw new IllegalStateException("Driver is not initialized. Call initDriver() first.");
    }
    return driver;
}

public static void quitBrowser() {
    WebDriver driver = tlDriver.get();
    if (driver != null) {
        driver.quit();
        tlDriver.remove();
    }
}
```

Design Patterns

30 September 2024 08:25

Introduction	Design Patterns are reusable Templates to common software design problems. They also promote code reusability, maintainability, and flexibility.		
	Creational Patterns: (How objects are created)	<ul style="list-style-type: none">Deals with object instantiation.Abstracts the instantiation process.	
	Singleton	<ul style="list-style-type: none">Purpose: Ensures a class has only one instance and provides a global point of accessUse Case: Logging, configuration management, thread pools.Implementation:<ul style="list-style-type: none">Private constructor.Static instance variable.Static method to get the instance. <pre>public class Singleton { private static Singleton instance; private Singleton() {} public static Singleton getInstance() { if (instance == null) { instance = new Singleton(); } return instance; } }</pre>	
	Factory Method	<ul style="list-style-type: none">Purpose: Defines an interface for creating an object, but lets subclasses decide which class to instantiate.Use Case: When you don't know the exact type of object needed at compile time.Implementation:<ul style="list-style-type: none">Abstract creator class with a factory method.Concrete creator subclasses that override the factory method.Example(Java): Creating different types of shapes.	
	Abstract Factory	<ul style="list-style-type: none">Purpose: Provides an interface for creating families of related or dependent objects without specifying their concrete classes.Use Case: Creating UI toolkits with different look and feel (e.g., Windows vs. macOS).Implementation:<ul style="list-style-type: none">Abstract factory interface.Concrete factory classes.Abstract product interfaces.Concrete product classes.Example(Java): Creating GUI elements for different operating systems.	
	Builder	<ul style="list-style-type: none">Purpose: Separates the construction of a complex object from its representation so that the same construction process can create different representations.Use Case: Creating complex objects with many optional parameters.Implementation:<ul style="list-style-type: none">Builder interface.Concrete builder classes.Director class (optional).Example(Java): Building a computer with various components.	
	Prototype	<ul style="list-style-type: none">Purpose: Specifies the kinds of objects to create using a prototypical instance, and create new objects by copying this prototype.Use Case: When creating objects is expensive and you want to copy existing objects.Implementation:<ul style="list-style-type: none">Prototype interface with a clone() method.Concrete prototype classes.Example(Java): Copying complex data objects.	
	Structural Patterns: (How objects are composed)	<ul style="list-style-type: none">Deals with object composition and relationships.Simplifies complex structures.	
	Adapter	<ul style="list-style-type: none">Purpose: Allows interfaces of existing classes to work together.Use Case: When you have an existing class with an incompatible interface.Implementation:<ul style="list-style-type: none">Adapter class that implements the target interface and wraps the adaptee.Example(Java): Adapting a legacy API to a new interface.	
	Bridge	<ul style="list-style-type: none">Purpose: Decouples an abstraction from its implementation so that the two can vary independently.Use Case: When you have multiple variations of both an abstraction and an implementation.Implementation:	

	<ul style="list-style-type: none"> • Abstraction interface and implementation interface. • Concrete abstraction and concrete implementation classes. • Example(Java): Creating different UI themes and drawing APIs.
Composite	<ul style="list-style-type: none"> • Purpose: Composes objects into tree structures to represent part-whole hierarchies. • Use Case: When you need to treat individual objects and compositions of objects uniformly. • Implementation: <ul style="list-style-type: none"> • Component interface. • Leaf and composite classes. • Example(Java): File system directories and files.
Decorator	<ul style="list-style-type: none"> • Purpose: Attaches additional responsibilities to an object dynamically. • Use Case: When you want to add behavior to an object without modifying its class. • Implementation: <ul style="list-style-type: none"> • Component interface. • Concrete component class. • Decorator abstract class. • Concrete decorator classes. • Example(Java): Adding features to a text editor.
Facade	<ul style="list-style-type: none"> • Purpose: Provides a unified interface to a set of interfaces in a subsystem. • Use Case: When you want to simplify the use of a complex subsystem. • Implementation: <ul style="list-style-type: none"> • Facade class that encapsulates the subsystem. • Example(Java): Simplifying a complex library.
Flyweight	<ul style="list-style-type: none"> • Purpose: Uses sharing to support large numbers of fine-grained objects efficiently. • Use Case: When you have many objects with shared state. • Implementation: <ul style="list-style-type: none"> • Flyweight interface. • Concrete flyweight classes. • Flyweight factory. • Example(Java): Rendering characters in a text editor.
Proxy	<ul style="list-style-type: none"> • Purpose: Provides a surrogate or placeholder for another object to control access to it. • Use Case: When you want to add a layer of control or protection around an object. • Implementation: <ul style="list-style-type: none"> • Subject interface. • Real subject class. • Proxy class. • Example(Java): Lazy loading, remote proxies.
Behavioral Patterns: (How objects interact)	<ul style="list-style-type: none"> • Deals with object communication and algorithms. • Defines how objects collaborate.
	<p>Chain of Responsibility</p> <ul style="list-style-type: none"> • Purpose: Avoids coupling the sender of a request to its receiver by giving more than one object a chance to handle the request. • Use Case: When multiple objects can handle a request. • Implementation: <ul style="list-style-type: none"> • Handler interface. • Concrete handler classes. • Example(Java): Handling event propagation.
	<p>Command</p> <ul style="list-style-type: none"> • Purpose: Encapsulates a request as an object, thereby letting you parameterize clients with different requests, queue or log requests, and support undoable operations. • Use Case: Implementing undo/redo, transaction processing. • Implementation: <ul style="list-style-type: none"> • Command interface. • Concrete command classes. • Invoker and receiver classes. • Example(Java): Implementing menu actions.
	<p>Interpreter</p> <ul style="list-style-type: none"> • Purpose: Given a language, define a representation for its grammar along with an interpreter that uses the representation to interpret sentences in the language. • Use Case: Parsing and evaluating expressions. • Implementation: <ul style="list-style-type: none"> • Abstract expression interface. • Terminal and non-terminal expression classes. • Example(Java): Implementing a simple calculator
	<p>Iterator</p> <ul style="list-style-type: none"> • Purpose: Provides a way to access the elements of an aggregate object sequentially without exposing its underlying representation. • Use Case: Traversing collections. • Implementation: <ul style="list-style-type: none"> • Iterator interface. • Concrete iterator class. • Aggregate interface. • Concrete aggregate class. • Example(Java): Traversing a list.
	<p>Mediator</p> <ul style="list-style-type: none"> • Purpose: Defines an object that encapsulates how a set of objects interact.

	<ul style="list-style-type: none"> • Use Case: Reducing coupling between objects. • Implementation: <ul style="list-style-type: none"> • Mediator interface. • Concrete mediator class. • Colleague classes. • Example(Java): Chat room, air traffic control. • Explanation: The Mediator pattern centralizes communication between objects. Instead of objects communicating directly with each other, they communicate through the mediator. This reduces coupling and simplifies the system.
Memento	<ul style="list-style-type: none"> • Purpose: Without violating encapsulation, capture and externalize an object's internal state so that the object can be restored to this state later. • Use Case: Implementing undo/redo, saving game states. • Implementation: <ul style="list-style-type: none"> • Originator class (the object whose state is saved). • Memento class (the object that stores the state). • Caretaker class (the object that manages the mementos). • Example(Java): Saving and restoring the state of a text editor. • Explanation: The Memento pattern allows you to save and restore the state of an object without exposing its internal details.
Observer	<ul style="list-style-type: none"> • Purpose: Defines a one-to-many dependency between objects so that when one object changes state, all its dependents are notified and updated automatically. • Use Case: Implementing event handling, model-view-controller (MVC). • Implementation: <ul style="list-style-type: none"> • Subject interface. • Concrete subject class. • Observer interface. • Concrete observer classes. • Example(Java): Implementing a stock ticker. • Explanation: The Observer pattern allows objects to subscribe to events and receive notifications when those events occur.
State	<ul style="list-style-type: none"> • Purpose: Allows an object to alter its behavior when its internal state changes. The object will appear to change its class. • Use Case: Implementing state machines, handling different modes of operation. • Implementation: <ul style="list-style-type: none"> • Context class (the object whose state changes). • State interface. • Concrete state classes. • Example(Java): Implementing a TCP connection. • Explanation: The State pattern encapsulates state-specific behavior in separate classes, making it easier to add or modify states.
Strategy	<ul style="list-style-type: none"> • Purpose: Defines a family of algorithms, encapsulates each one, and makes them interchangeable. Strategy lets the algorithm vary independently from clients that use it. • Use Case: Implementing different sorting algorithms, payment methods. • Implementation: <ul style="list-style-type: none"> • Strategy interface. • Concrete strategy classes. • Context class. • Example(Java): Implementing different compression algorithms. • Explanation: The Strategy pattern allows you to choose an algorithm at runtime, making your code more flexible.
Template Method	<ul style="list-style-type: none"> • Purpose: Defines the skeleton of an algorithm in an operation, deferring some steps to subclasses. Template Method lets subclasses redefine certain steps of an algorithm without changing the algorithm's structure. • Use Case: Implementing frameworks, defining common steps in a process. • Implementation: <ul style="list-style-type: none"> • Abstract class with a template method. • Concrete subclasses that override specific steps. • Example(Java): Implementing different report generation processes. • Explanation: The Template Method pattern provides a reusable structure for algorithms while allowing subclasses to customize specific steps.
Visitor	<ul style="list-style-type: none"> • Purpose: Represents an operation to be performed on the elements of an object structure. Visitor lets you define a new operation without changing the classes of the elements on which it operates. • Use Case: Implementing compilers, performing operations on complex data structures. • Implementation: <ul style="list-style-type: none"> • Visitor interface. • Concrete visitor classes. • Element interface. • Concrete element classes. • Example(Java): Implementing a compiler's syntax tree traversal. • Explanation: The Visitor pattern allows you to add new operations to an object structure without modifying the structure itself.

Singleton	<ul style="list-style-type: none"> • Create Private Static Instance of the Class • Create Private Constructor • Public Static getInstance for instance/object <pre>public class Browser { 4 usages new * private static Browser browser; 3 usages private Browser(){} public static Browser getBrowser(){ 1 usage new * if(browser==null){ browser=new Browser(); } return browser; } public void display(){ no usages new * System.out.println("Browser Info"); } }</pre> <ul style="list-style-type: none"> • Singleton pattern does not work well in parallel execution: Use ThreadLocal with Parallel Execution • Ensures that certain utilities (loggers) have only one instance across the entire framework.
Selenium Singleton	<pre>public class WebDriverSingleton { private static final ThreadLocal<WebDriver> driver = new ThreadLocal<>(); private WebDriverSingleton() {} public static WebDriver getDriver(String browser) { if (driver.get() == null) { switch (browser.toLowerCase()) { case "chrome": ChromeOptions chromeOptions = new ChromeOptions(); chromeOptions.addArguments("--start-maximized"); driver.set(new ChromeDriver(chromeOptions)); break; case "firefox": FirefoxOptions firefoxOptions = new FirefoxOptions(); firefoxOptions.addArguments("--start-maximized"); driver.set(new FirefoxDriver(firefoxOptions)); break; case "edge": EdgeOptions edgeOptions = new EdgeOptions(); edgeOptions.addArguments("--start-maximized"); driver.set(new EdgeDriver(edgeOptions)); break; default: throw new IllegalArgumentException("Unsupported browser: " + browser); } } return driver.get(); } public static void quitDriver() { if (driver.get() != null) { driver.get().quit(); driver.remove(); } } } @BeforeMethod @Parameters("browser")</pre>

```

public void setup(String browser) {
    driver = WebDriverSingleton.getDriver(browser);
}

@Test
public void testExample1() {
    driver.get("https://www.example.com");
    Assert.assertEquals(driver.getTitle(), "Example Domain");
}

```

Factory Pattern(Used by webDriver)

A Factory Pattern (Follows LSP) says that just define an interface or abstract class for creating an object but let the subcl asses decide which class to instantiate. In other words, subclasses are responsible to create the instance of the class.

The Factory Method pattern is a creational design pattern that provides an interface for creating objects in a superclass but allows subclasses to alter the type of objects that will be created. It encapsulates the object creation logic, decoupling it from t he client code that uses the objects.

The Factory Method Pattern is also known as Virtual Constructor.

Advantages

- The Factory Pattern allows WebDriver to abstract away the specific implementation details of different browser drivers.
- This makes it easier to switch between browsers without modifying client code.
- New browser drivers can be added to the framework without affecting existing tests.
- Different browsers can be configured independently.

`PageFactory.initElements(driver, this)` is used to initialize the page objects' elements. Selenium uses reflection to instanti ate and initialize the elements of the page, which is another form of the Factory pattern.

Components of Factory Method Design Pattern

- Product: Represents the interface of the objects created by the factory method.
- ConcreteProduct: Implements the Product interface and represents the concrete objects created by the factory method.
- Creator: Declares the factory method, which returns an instance of the Product interface.
- ConcreteCreator: Implements the factory method to create instances of ConcreteProduct.

```

public interface BrowserDriver {
    WebDriver createDriver();
}

public class ChromeDriverManager implements BrowserDriver {
    @Override
    public WebDriver createDriver() {
        return new ChromeDriver();
    }
}

public class EdgeDriverManager implements BrowserDriver {
    @Override
    public WebDriver createDriver() {
        return new EdgeDriver();
    }
}

public class DriverFactory {
    public static BrowserDriver getDriver(String browserType) {
        return switch (browserType.toLowerCase()) {
            case "chrome" -> new ChromeDriverManager();
            case "firefox" -> new FirefoxDriverManager();
            case "edge" -> new EdgeDriverManager();
            default -> throw new IllegalArgumentException("Unknown browser type: " + browserType);
        };
    }
}

public static void main(String[] args) {
    String browserType = "edge";
    WebDriver driver = DriverFactory.getDriver(browserType).createDriver();
    driver.manage().window().maximize();
    driver.get("https://www.google.com/");
    System.out.println(driver.getTitle());
    driver.quit();
}

```

The Factory Method Design Pattern is a powerful tool for creating objects in a flexible, decoupled, and extensible manner. By encapsulating object creation logic in subclasses, the Factory Method pattern promotes code reusability, maintainability, and scalability. While it may introduce some complexity, especially in large codebases, its benefits outweigh the drawbacks, making it a valuable addition to any developer's toolkit.

	<p>When PageFactory.initElements is called:</p> <ul style="list-style-type: none"> • It reads all the fields annotated with @FindBy (or similar annotations). • For each field, it dynamically generates a proxy object (using a mechanism like Java's Reflection or Proxy classes). • The proxy object doesn't immediately locate the WebElement; instead, it defers the lookup until the element is actually interacted with. This is known as lazy initialization. • The proxy handles the interaction with the WebDriver, locating the element on the page at runtime. 				
Fluent Interface	<p>It is an object-oriented API design that allows us to chain method calls together in a readable and intuitive manner.</p> <p>To implement it, we need to declare methods that return objects from the same class. As a result, we'll be able to chain together multiple method calls.</p> <pre>class Query { private String select; private String from; public Query select(String select) { this.select = select; return this; } public Query from(String from) { this.from = from; return this; } public void execute() { System.out.println("Executing query: SELECT " + select + " FROM " + from); } } Query query = new Query().select("*", "name").from("users").execute();</pre> <p>Java8's Stream API is using the fluent interface pattern and allows users to manipulate streams of data in a very declarative way.</p> <pre>Stream<String> processedNumbers = numbers.distinct() .filter(nr -> nr % 2 == 0) .skip(1) .limit(4) .map(nr -> "#" + nr) .peek(nr -> System.out.println(nr));</pre>				
Builder Pattern	<p>The Builder Design Pattern is a creational pattern that allows for the step-by-step creation of complex objects. It is mostly used when object can't be created in single step like in the de-serialization of a complex object.</p> <p>While fluent and builder patterns aim to improve code readability, there are distinct differences. The Fluent Interface is focused on method chaining, while the Builder Pattern is concerned with constructing complex objects</p> <p>Advantages</p> <ul style="list-style-type: none"> • Useful when the constructor has many fields • Useful when the constructor has few optional and few mandatory fields • Can be used in UI automation for fill registration forms <table border="1"> <thead> <tr> <th style="text-align: center;">With Fluent Interface</th> <th style="text-align: center;">Without Fluent Interface</th> </tr> </thead> <tbody> <tr> <td> <pre>@Data public class Product { private String name; private Double price; private String description; private String category; private Double discount; private boolean freeshipping; private int stock; // Private constructor to enforce usage of Builder private Product(ProductBuilder builder) { this.name = builder.name; this.price = builder.price; this.description = builder.description; this.category = builder.category; this.discount = builder.discount; this.freeshipping = builder.freeshipping; this.stock = builder.stock; } // Static inner ProductBuilder class }</pre> </td> <td> <pre>@Data public class Product { private final String name; // Required private final Double price; // Required private String description; // Optional private String category; // Optional private Double discount; // Optional private boolean freeshipping; // Optional private int stock; // Optional // Private constructor to enforce usage of Builder private Product(Builder builder) { this.name = builder.name; this.price = builder.price; this.description = builder.description; this.category = builder.category; this.discount = builder.discount; this.freeshipping = builder.freeshipping; this.stock = builder.stock; } // Static inner Builder class }</pre> </td> </tr> </tbody> </table>	With Fluent Interface	Without Fluent Interface	<pre>@Data public class Product { private String name; private Double price; private String description; private String category; private Double discount; private boolean freeshipping; private int stock; // Private constructor to enforce usage of Builder private Product(ProductBuilder builder) { this.name = builder.name; this.price = builder.price; this.description = builder.description; this.category = builder.category; this.discount = builder.discount; this.freeshipping = builder.freeshipping; this.stock = builder.stock; } // Static inner ProductBuilder class }</pre>	<pre>@Data public class Product { private final String name; // Required private final Double price; // Required private String description; // Optional private String category; // Optional private Double discount; // Optional private boolean freeshipping; // Optional private int stock; // Optional // Private constructor to enforce usage of Builder private Product(Builder builder) { this.name = builder.name; this.price = builder.price; this.description = builder.description; this.category = builder.category; this.discount = builder.discount; this.freeshipping = builder.freeshipping; this.stock = builder.stock; } // Static inner Builder class }</pre>
With Fluent Interface	Without Fluent Interface				
<pre>@Data public class Product { private String name; private Double price; private String description; private String category; private Double discount; private boolean freeshipping; private int stock; // Private constructor to enforce usage of Builder private Product(ProductBuilder builder) { this.name = builder.name; this.price = builder.price; this.description = builder.description; this.category = builder.category; this.discount = builder.discount; this.freeshipping = builder.freeshipping; this.stock = builder.stock; } // Static inner ProductBuilder class }</pre>	<pre>@Data public class Product { private final String name; // Required private final Double price; // Required private String description; // Optional private String category; // Optional private Double discount; // Optional private boolean freeshipping; // Optional private int stock; // Optional // Private constructor to enforce usage of Builder private Product(Builder builder) { this.name = builder.name; this.price = builder.price; this.description = builder.description; this.category = builder.category; this.discount = builder.discount; this.freeshipping = builder.freeshipping; this.stock = builder.stock; } // Static inner Builder class }</pre>				

```

public static class ProductBuilder {
    private String name;
    private Double price;
    //optional fields
    private String description;
    private String category;
    private Double discount;
    private boolean freeshipping;
    private int stock;

    //public constructor for mandatory fields
    public ProductBuilder(String name, Double price) {
        this.name = name;
        this.price = price;
    }

    //setter for optional fields
    public ProductBuilder setDescription(String description) {
        this.description = description;
        return this;
    }

    public ProductBuilder setCategory(String category) {
        this.category = category;
        return this;
    }

    public ProductBuilder setDiscount(Double discount) {
        this.discount = discount;
        return this;
    }

    public ProductBuilder setFreeshipping(boolean freeshipping) {
        this.freeshipping = freeshipping;
        return this;
    }

    public ProductBuilder setStock(int stock) {
        this.stock = stock;
        return this;
    }

    public Product build(){
        return new Product(this);
    }
}

public class Runner {
    public static void main(String[] args) {
        Product myTv = new Product.ProductBuilder("MI TV",
12500.0).build();
        System.out.println(myTv);

        Product appleTv = new Product.ProductBuilder("Apple TV",
12500.0)
            .setDescription("Mehenga TV")
            .setCategory("Electronics")
            .build();
        appleTv.setDiscount(5.2);
        System.out.println(appleTv);
    }
}

```

```

public static class Builder {
    private final String name;
    private final Double price;
    //optional fields
    private String description;
    private String category;
    private Double discount;
    private boolean freeshipping;
    private int stock;

    //public constructor for mandatory fields
    public Builder(String name, Double price) {
        this.name = name;
        this.price = price;
    }

    //setter for optional fields
    public void setDescription(String description) {
        this.description = description;
    }

    public void setCategory(String category) {
        this.category = category;
    }

    public void setDiscount(Double discount) {
        this.discount = discount;
    }

    public void setFreeshipping(boolean freeshipping) {
        this.freeshipping = freeshipping;
    }

    public void setStock(int stock) {
        this.stock = stock;
    }

    public Product build() {
        return new Product(this);
    }
}

public class Runner {
    public static void main(String[] args) {
        Product.Builder builder = new Product.Builder("MI TV",
12500.0);
        builder.setDescription("Affordable TV");
        builder.setCategory("Electronics");
        builder.setDiscount(10.0);
        builder.setFreeshipping(true);
        builder.setStock(50);

        Product myTv = builder.build();
        System.out.println(myTv);
    }
}

```

Strategy Pattern

Strategy Design Pattern is a behavioral design pattern that allows the behavior of a class instance to be changed dynamically. This pattern is useful when there are multiple ways or strategies to perform a task, and you want to switch between these strategies at runtime. Imagine a scenario where we are developing test scripts to validate API responses. For this task, we can employ multiple strategies, such as validating for a valid response or checking for an invalid response.

Designing the Strategy Design pattern has 4 steps

Step 1: Define the Strategy Interface	@FunctionalInterface public interface ValidationStrategy { void validate(Response response); }
---------------------------------------	---

Step 2: Implement Concrete Strategy Classes	import java.util.Set;
---	-----------------------

	<pre> public class ValidResponseStrategy implements ValidationStrategy { @Override public void validate(Response response) { int expectedStatusCode = 200; if (response.getStatusCode() != expectedStatusCode) { throw new AssertionError("Expected status code " + expectedStatusCode + ", but got " + response.getStatusCode()); } } } public class InvalidResponseStrategy implements ValidationStrategy { private static final Set<Integer> INVALID_STATUS_CODES = Set.of(400, 404, 500); @Override public void validate(Response response) { if (INVALID_STATUS_CODES.contains(response.getStatusCode())) { throw new AssertionError("Did not expect status code " + response.getStatusCode() + ", which is considered invalid"); } } } </pre>
Step 3: Create the Context Class	<pre> public class ResponseValidatorContext { private ValidationStrategy strategy; public ResponseValidatorContext(ValidationStrategy strategy) { this.strategy = strategy; } public void setStrategy(ValidationStrategy strategy) { this.strategy = strategy; } public void validateResponse(Response response) { strategy.validate(response); } } </pre>
Step 4: Utilize the Strategy Pattern	<pre> public class Main { public static void main(String[] args) { Response response = new Response(200); // Using ValidResponseStrategy with a lambda expression ValidationStrategy validStrategy = r -> { int expectedStatusCode = 200; if (r.getStatusCode() != expectedStatusCode) { throw new AssertionError("Expected status code " + expectedStatusCode + ", but got " + r.getStatusCode()); } }; // Using InvalidResponseStrategy with a lambda expression Set<Integer> invalidStatusCodes = Set.of(400, 404, 500); ValidationStrategy invalidStrategy = r -> { if (invalidStatusCodes.contains(r.getStatusCode())) { throw new AssertionError("Did not expect status code " + r.getStatusCode() + ", which is considered invalid"); } }; ResponseValidatorContext context = new ResponseValidatorContext(validStrategy); context.validateResponse(response); context.setStrategy(invalidStrategy); context.validateResponse(response); } } </pre>

The Strategy design pattern is a powerful tool in the arsenal of software engineers aiming to write clean, flexible, and maintainable code. Particularly useful in scenarios involving varying algorithms, it separates the concerns of algorithm selection from the context in which they are used. By embracing this pattern, developers can ensure that their code is both scalable and resilient to change, embodying the principles of good software design.

Decorator Design Pattern(OCP)	<p>Structural design pattern used to dynamically add new functionality to an object without altering its structure. It achieves this by wrapping the original object with a set of decorator objects that enhance or modify its behavior.</p> <ul style="list-style-type: none"> • Base Component: This is the interface or abstract class representing the core entity (e.g., Coffee). • Concrete Component: This is the base implementation of the component (e.g., PlainCoffee). • Decorator: This is an abstract class or interface implementing the base component, which adds the ability to decorate. • Concrete Decorators: These are the actual decorators that add functionality (e.g., Milk, Sugar). <pre> // Base Component interface Coffee { String getDescription(); double getCost(); } // Concrete Component class PlainCoffee implements Coffee { @Override public String getDescription() { return "Plain Coffee"; } @Override public double getCost() { return 5.0; // Base price of plain coffee } } // Decorator abstract class CoffeeDecorator implements Coffee { protected Coffee coffee; public CoffeeDecorator(Coffee coffee) { this.coffee = coffee; } @Override public String getDescription() { return coffee.getDescription(); } @Override public double getCost() { return coffee.getCost(); } } // Concrete Decorators class Milk extends CoffeeDecorator { public Milk(Coffee coffee) { super(coffee); } @Override public String getDescription() { return super.getDescription() + ", Milk"; } @Override public double getCost() { return super.getCost() + 1.5; // Additional cost for milk } } class Sugar extends CoffeeDecorator { public Sugar(Coffee coffee) { super(coffee); } @Override public String getDescription() { return super.getDescription() + ", Sugar"; } @Override public double getCost() { return super.getCost() + 0.5; // Additional cost for sugar } } </pre>
-------------------------------	--

```

        }

class WhippedCream extends CoffeeDecorator {
    public WhippedCream(Coffee coffee) {
        super(coffee);
    }

    @Override
    public String getDescription() {
        return super.getDescription() + ", Whipped Cream";
    }

    @Override
    public double getCost() {
        return super.getCost() + 2.0; // Additional cost for whipped cream
    }
}

// Main Class
public class CoffeeShop {
    public static void main(String[] args) {
        // Start with a plain coffee
        Coffee myCoffee = new PlainCoffee();
        System.out.println(myCoffee.getDescription() + " -> $" + myCoffee.getCost());

        // Add Milk
        myCoffee = new Milk(myCoffee);
        System.out.println(myCoffee.getDescription() + " -> $" + myCoffee.getCost());

        // Add Sugar
        myCoffee = new Sugar(myCoffee);
        System.out.println(myCoffee.getDescription() + " -> $" + myCoffee.getCost());

        // Add Whipped Cream
        myCoffee = new WhippedCream(myCoffee);
        System.out.println(myCoffee.getDescription() + " -> $" + myCoffee.getCost());
    }
}

```

Selenium Decorator	<pre> // Base Component interface Element { void click(); void sendKeys(String text); String getText(); } // Concrete Component class BaseElement implements Element { private WebElement webElement; public BaseElement(WebElement webElement) { this.webElement = webElement; } @Override public void click() { webElement.click(); } @Override public void sendKeys(String text) { webElement.sendKeys(text); } @Override public String getText() { return webElement.getText(); } } // Decorator abstract class ElementDecorator implements Element { protected Element element; public ElementDecorator(Element element) { </pre>
--------------------	---

```

        this.element = element;
    }

    @Override
    public void click() {
        element.click();
    }

    @Override
    public void sendKeys(String text) {
        element.sendKeys(text);
    }

    @Override
    public String getText() {
        return element.getText();
    }
}

// Concrete Decorators
class LoggingElement extends ElementDecorator {
    public LoggingElement(Element element) {
        super(element);
    }

    @Override
    public void click() {
        System.out.println("Clicking element");
        super.click();
    }

    @Override
    public void sendKeys(String text) {
        System.out.println("Typing: " + text);
        super.sendKeys(text);
    }

    @Override
    public String getText() {
        String text = super.getText();
        System.out.println("Reading text: " + text);
        return text;
    }
}

class ExceptionHandlingElement extends ElementDecorator {
    public ExceptionHandlingElement(Element element) {
        super(element);
    }

    @Override
    public void click() {
        try {
            super.click();
        } catch (Exception e) {
            System.err.println("Error clicking element: " + e.getMessage());
        }
    }

    @Override
    public void sendKeys(String text) {
        try {
            super.sendKeys(text);
        } catch (Exception e) {
            System.err.println("Error typing: " + e.getMessage());
        }
    }
}

```

Usage in Framework:

```

WebElement webElement = driver.findElement(By.id("example"));

// Wrap with decorators

```

```

Element element = new LoggingElement(new ExceptionHandlingElement(new BaseElement(webElement)));

// Perform actions
element.click();
element.sendKeys("Test");
System.out.println(element.getText());

```

Use Case 2: Screenshot on Failure

```

class ScreenshotElement extends ElementDecorator {
    private WebDriver driver;

    public ScreenshotElement(Element element, WebDriver driver) {
        super(element);
        this.driver = driver;
    }

    @Override
    public void click() {
        try {
            super.click();
        } catch (Exception e) {
            takeScreenshot();
            throw e;
        }
    }

    @Override
    public void sendKeys(String text) {
        try {
            super.sendKeys(text);
        } catch (Exception e) {
            takeScreenshot();
            throw e;
        }
    }

    private void takeScreenshot() {
        File screenshot = ((TakesScreenshot) driver).getScreenshotAs(OutputType.FILE);
        try {
            FileUtils.copyFile(screenshot, new File("screenshot_" + System.currentTimeMillis() + ".png"));
            System.out.println("Screenshot captured!");
        } catch (IOException e) {
            System.err.println("Failed to save screenshot: " + e.getMessage());
        }
    }
}

Usage:
Element element = new ScreenshotElement(new BaseElement(webElement), driver);
element.click();

```

Apache POI

15 December 2024 00:00

Read Data	<pre>public static Map<String, String> getRowData(String filePath, int rowNumber) throws IOException { Map<String, String> rowData = new HashMap<>(); try (FileInputStream fileInputStream = new FileInputStream(filePath); Workbook workbook = new XSSFWorkbook(fileInputStream)) { Sheet sheet = workbook.getSheetAt(0); Row headerRow = sheet.getRow(0); if (headerRow == null) { throw new IllegalArgumentException("Header row is missing in the sheet."); } Row dataRow = sheet.getRow(rowNumber); if (dataRow == null) { throw new IllegalArgumentException("Row number " + rowNumber + " is missing in the sheet."); } for (Cell headerCell : headerRow) { int columnIndex = headerCell.getColumnIndex(); String header = headerCell.getStringCellValue(); Cell dataCell = dataRow.getCell(columnIndex); String value = getCellValueAsString(dataCell); rowData.put(header, value); } } return rowData; } private static String getCellValueAsString(Cell cell) { return switch (cell.getCellType()) { case STRING -> cell.getStringCellValue(); case NUMERIC -> String.valueOf(cell.getNumericCellValue()); case BOOLEAN -> String.valueOf(cell.getBooleanCellValue()); default -> ""; }; }</pre>
Update Data	<pre>public static void updateCellData(String filePath, int rowNumber, String columnHeader, String newValue) throws IOException { try (FileInputStream fileInputStream = new FileInputStream(filePath); Workbook workbook = new XSSFWorkbook(fileInputStream)) { Sheet sheet = workbook.getSheetAt(0); Row headerRow = sheet.getRow(0); if (headerRow == null) { throw new IllegalArgumentException("Header row is missing in the sheet."); } int columnIndex = -1; for (Cell headerCell : headerRow) { if (headerCell.getStringCellValue().equalsIgnoreCase(columnHeader)) { columnIndex = headerCell.getColumnIndex(); break; } } if (columnIndex == -1) { throw new IllegalArgumentException("Column header '" + columnHeader + "' not found in the sheet."); } Row dataRow = sheet.getRow(rowNumber); if (dataRow == null) { dataRow = sheet.createRow(rowNumber); } Cell cellToUpdate = dataRow.getCell(columnIndex); if (cellToUpdate == null) { cellToUpdate = dataRow.createCell(columnIndex); } cellToUpdate.setCellValue(newValue); try (FileOutputStream fileOutputStream = new FileOutputStream(filePath)) { workbook.write(fileOutputStream); } } }</pre>

Cucumber

02 March 2024 07:45

Syntax	<p>@SmokeTest @RegressionTest Scenario: Successful Login</p> <p>Given: Preconditions are mentioned in the Given keyword. When: The purpose of the When Steps is to describe the user action. Then: The purpose of Then Steps is to observe the expected output. The observations should be related to the business value/benefit of your Feature description. And: Used to add additional conditions or actions to Given, When, or Then steps. But: Used to add conditions or actions that are exceptions to the previous steps.</p>								
Testng Runner	AbstractTestNGCucumberTests needs to be inherited by the runner								
Tagging	<p>A feature or scenario can have as many tags as you like. Separate them with spaces</p> <p>Tags can be placed above the following Gherkin elements:</p> <ul style="list-style-type: none"> • Feature • Scenario • Scenario Outline • Examples <p>Scenario Outline: Steps will run conditionally if tagged</p> <pre>Given user is logged in When user clicks <link> Then user will be logged out @mobile Examples: link logout link on mobile @desktop Examples: link logout link on desktop </pre> <p>It is not possible to place tags above Background or steps (Given, When, Then, And and But).</p> <p>Tags are inherited by child elements. Tags that are placed above a Feature will be inherited by Scenario, Scenario Outline, or Examples. Tags that are placed above a Scenario Outline will be inherited by Examples.</p>								
And Tag in Test Runner	cucumber --tags "@smoke and @regression"								
Or Tag in Test Runner	cucumber --tags "@smoke or @regression"								
NotTag in Test Runner	<p>cucumber --tags "not @wip"</p> <p>(@smoke or @ui) and (not @slow)</p>								
Hooks	<table border="1"> <tr> <td>Scenario hooks</td><td> <p>Scenario hooks run for every scenario.</p> <ul style="list-style-type: none"> • Before hooks run before the first step of each scenario • After hooks run after the last step of each scenario, even when the step result is failed, undefined, pending, or skipped </td></tr> <tr> <td>Step hooks</td><td> <p>Step hooks are invoked before and after a step. The hooks have 'invoke around' semantics, meaning that if a BeforeStep hook is executed the AfterStep hooks will also be executed regardless of the result of the step. If a step did not pass, the following step and its hooks will be skipped</p> <ul style="list-style-type: none"> • BeforeStep • AfterStep </td></tr> <tr> <td>Conditional hooks</td><td> <p>Hooks can be conditionally selected for execution based on the tags of the scenario. To run a particular hook only for certain scenarios, you can associate a Before or After hook with a tag expression.</p> <pre>@After("@browser and not @headless") public void doSomethingAfter(Scenario scenario){}</pre> </td></tr> <tr> <td>Global hooks</td><td> <p>Global hooks will run once before any scenario is run or after all scenario have been run.</p> <p>BeforeAll run before any scenario is run. AfterAll run after all scenarios have been executed.</p> </td></tr> </table>	Scenario hooks	<p>Scenario hooks run for every scenario.</p> <ul style="list-style-type: none"> • Before hooks run before the first step of each scenario • After hooks run after the last step of each scenario, even when the step result is failed, undefined, pending, or skipped 	Step hooks	<p>Step hooks are invoked before and after a step. The hooks have 'invoke around' semantics, meaning that if a BeforeStep hook is executed the AfterStep hooks will also be executed regardless of the result of the step. If a step did not pass, the following step and its hooks will be skipped</p> <ul style="list-style-type: none"> • BeforeStep • AfterStep 	Conditional hooks	<p>Hooks can be conditionally selected for execution based on the tags of the scenario. To run a particular hook only for certain scenarios, you can associate a Before or After hook with a tag expression.</p> <pre>@After("@browser and not @headless") public void doSomethingAfter(Scenario scenario){}</pre>	Global hooks	<p>Global hooks will run once before any scenario is run or after all scenario have been run.</p> <p>BeforeAll run before any scenario is run. AfterAll run after all scenarios have been executed.</p>
Scenario hooks	<p>Scenario hooks run for every scenario.</p> <ul style="list-style-type: none"> • Before hooks run before the first step of each scenario • After hooks run after the last step of each scenario, even when the step result is failed, undefined, pending, or skipped 								
Step hooks	<p>Step hooks are invoked before and after a step. The hooks have 'invoke around' semantics, meaning that if a BeforeStep hook is executed the AfterStep hooks will also be executed regardless of the result of the step. If a step did not pass, the following step and its hooks will be skipped</p> <ul style="list-style-type: none"> • BeforeStep • AfterStep 								
Conditional hooks	<p>Hooks can be conditionally selected for execution based on the tags of the scenario. To run a particular hook only for certain scenarios, you can associate a Before or After hook with a tag expression.</p> <pre>@After("@browser and not @headless") public void doSomethingAfter(Scenario scenario){}</pre>								
Global hooks	<p>Global hooks will run once before any scenario is run or after all scenario have been run.</p> <p>BeforeAll run before any scenario is run. AfterAll run after all scenarios have been executed.</p>								
Background	It runs before each and every scenario were for a feature in which it is defined								
Hooks Vs Background	Hook runs before Background Before Hooks->Background->Scenario->After Hook								

	<p>Background cannot be tagged, hooks can be: @Before("@Smoke")</p> <p>Orders Hooks: @Before(order=0),@Before(order=1)</p>		
{string} Regex in feature File	<p>{string} matches quoted strings: "text"</p> <p>Given the user is on the "login" page</p> <pre>@Given("the user is on the {string} page") public void theUserIsOnThePage(String page) { System.out.println("User is on the " + page + " page"); }</pre>		
(.*) Regex in feature File	<p>(.*) matches any character sequence, including spaces, without needing quotes</p> <p>Given the user is on the "login" page</p> <pre>@Given("^the user is on the (.*) page\$") public void theUserIsOnThePage(String page) { System.out.println("User is on the " + page + " page"); }</pre>		
{int}	Given the user has "42" points		
{float}	Given the price is "3.14"		
{word}	<p>Matches a single word (a sequence of characters without spaces).</p> <p>Given the user is logged in as username</p>		
Scenario Outline	<p>When The "AddComment" request is sent with the "Post" HTTP method on IssueID as "<ISSUEID>"</p> <p>Examples:</p> <pre> RowNumber ISSUEID DISPLAY_NAME UPDATE_AUTHOR_EMAIL_ADDRESS COMMENT 1 ISSUEID DISPLAY_NAME UPDATE_AUTHOR_EMAIL_ADDRESS COMMENT 2 ISSUEID DISPLAY_NAME UPDATE_AUTHOR_EMAIL_ADDRESS COMMENT </pre> <p>@When("The {string} request is sent with the {string} HTTP method on IssueID as {string}")</p> <pre>public void request_is_sent_with_http_method_on_issueid(String resource, String httpMethod, String issueID) { issueID = excelSheetReader.readCell(issueID); if (httpMethod.equalsIgnoreCase("POST")) { addCommentResponse = SpecBuilders.SendPostAndReturnResponse(updatedJsonPayload, issueID, resource); } }</pre>		
Dependency Injection	<p>DI allows you to "inject" the things a class needs (its dependencies) from the outside, instead of letting the class create or manage them itself.</p> <p>For Example: Car class might depend on a Engine class to run. Without DI, the Car class would directly create or manage the Engine instance within its code, which makes the two classes tightly coupled. This approach can create problems, particularly when you need to test, extend, or modify the classes in the future.</p>		
PicoContainer	<p>Dependency Injection (DI) in Cucumber using PicoContainer allows you to share state and objects between step definitions, hooks, and other classes.</p> <ul style="list-style-type: none"> No Manual Management: Dependency management is handled by PicoContainer, so you don't need to manually manage object lifecycles or dependencies between steps. Consistency Across Steps: Objects maintain their state across different steps within the same scenario, making tests easier to write and maintain. Simple Integration: Integrating PicoContainer with Cucumber requires minimal setup and aligns with Cucumber's aim to simplify writing BDD style tests. <p>Using PicoContainer to share state between steps in a scenario is easy and non-intrusive. All you need is a constructor that requires an object that PicoContainer can create and inject. PicoContainer is invisible. Add a dependency to cucumber-picocontainer and make sure that the constructors for the step classes require an instance of the same class. Simply hand it some classes and it will instantiate each one, correctly wired together via their constructors. That's it. Cucumber scans your classes with step definitions in them, passes them to PicoContainer, then asks it to create new instances for every scenario.</p> <ul style="list-style-type: none"> Add PicoContainer to the Project Create a Test Context class which will hold all the objects state Divide the Steps class into multiple steps classes with logical separation Write Constructor to share Test Context <table border="1"> <tr> <td>Add Dependency</td> <td> <pre><dependencies> <dependency> <groupId>io.cucumber</groupId> <artifactId>cucumber-picocontainer</artifactId> <version>7.0.0</version> <scope>test</scope> </dependency></pre> </td> </tr> </table>	Add Dependency	<pre><dependencies> <dependency> <groupId>io.cucumber</groupId> <artifactId>cucumber-picocontainer</artifactId> <version>7.0.0</version> <scope>test</scope> </dependency></pre>
Add Dependency	<pre><dependencies> <dependency> <groupId>io.cucumber</groupId> <artifactId>cucumber-picocontainer</artifactId> <version>7.0.0</version> <scope>test</scope> </dependency></pre>		

		</dependencies>
	Create Shared Class	<p>Define a class that will hold the shared state or objects.</p> <pre>public class SharedState { private String importantData; public void setImportantData(String data) { this.importantData = data; } public String getImportantData() { return importantData; } }</pre>
	Feature File	<p>Feature: Shared State Example</p> <p>Scenario: Sharing state between steps Given I set the important data to "expectedData" in step definition 1 Then I should have access to the important data in step definition 2</p>
	Step Definition with Dependency Injection	<pre>import io.cucumber.java.en.Given; public class StepDefinitions1 { private final SharedState sharedState; public StepDefinitions1(SharedState sharedState) { this.sharedState = sharedState; } @Given("I set the important data to {string} in step definition 1") public void setImportantData(String data) { sharedState.setImportantData(data); } } ----- import io.cucumber.java.en.Then; import static org.junit.Assert.assertEquals; public class StepDefinitions2 { private final SharedState sharedState; public StepDefinitions2(SharedState sharedState) { this.sharedState = sharedState; } @Then("I should have access to the important data in step definition 2") public void verifyImportantData() { String data = sharedState.getImportantData(); System.out.println("The important data is: " + data); assertEquals("Expected data not found", "expectedData", data); } }</pre>
	Output	The important data is: expectedData
DI-Selenium Example	Create a Test Context class	<pre>public class TestContext { private WebDriverManager webDriverManager; private PageObjectManager pageObjectManager; public TestContext(){ webDriverManager = new WebDriverManager(); pageObjectManager = new PageObjectManager(webDriverManager.getDriver()); } public WebDriverManager getWebDriverManager() { return webDriverManager; } public PageObjectManager getPageObjectManager() { return pageObjectManager; } }</pre>

		}								
Divide the Steps file	<p>OLD Step File</p> <pre>public class HomePageSteps { WebDriver driver; HomePage homePage; PageObjectManager pageObjectManager; WebDriverManager webDriverManager; @Given("^user is on Home Page\$") public void user_is_on_Home_Page(){ webDriverManager = new WebDriverManager(); driver = webDriverManager.getDriver(); pageObjectManager = new PageObjectManager(driver); homePage = pageObjectManager.getHomePage(); homePage.navigateTo_HomePage(); } @When("^he search for \"([^\"]*)\"\$") public void he_search_for(String product) { homePage.perform_Search(product); } }</pre>	<p>Constructor for Step Definition classes to share TestContext</p> <p>We need WebDriverManager & PageObjectManager in every step file, otherwise, you need to create objects for both classes using new operator again and again, which is Kill Bill.</p> <p>Now with just adding Constructor to HomePageSteps file and pass TestContext as a Parameter to constructor would take all the pain. Within the TestContext object we have everything available which is required for the test. So now let's see how the new HomePageSteps class would look.</p> <pre>public class HomePageSteps { TestContext testContext; HomePage homePage; public HomePageSteps(TestContext context) { testContext = context; homePage = testContext.getPageObjectManager().getHomeP age(); } @Given("^user is on Home Page\$") public void user_is_on_Home_Page(){ homePage.navigateTo_HomePage(); } @When("^he search for \"([^\"]*)\"\$") public void he_search_for(String product) { homePage.perform_Search(product); } }</pre>								
DataTable	<p>Scenario: Correct non-zero number of books found by author by map</p> <p>Given I have the following books in the store by map</p> <table border="1"> <thead> <tr> <th>title</th> <th>author</th> </tr> </thead> <tbody> <tr> <td>The Devil in the White City</td> <td>Erik Larson</td> </tr> <tr> <td>The Lion, the Witch and the Wardrobe</td> <td>C.S. Lewis</td> </tr> <tr> <td>In the Garden of Beasts</td> <td>Erik Larson</td> </tr> </tbody> </table> <p>When I search for books by author Erik Larson</p> <p>Then I find 2 books</p> <p>Cucumber repeats this process for each subsequent row</p> <pre>[{"title": "The Devil in the White City", "author": "Erik Larson"}, {"title": "The Lion, the Witch and the Wardrobe", "author": "C.S. Lewis"}, {"title": "In the Garden of Beasts", "author": "Erik Larson"}]</pre> <p>@Given("^I have the following books in the store by map\$") public void haveBooksInTheStoreByMap(DataTable table) {</p> <pre>List<Map<String, String>> rows = table.asMaps(String.class, String.class); for (Map<String, String> columns : rows) { store.addBook(new Book(columns.get("title"), columns.get("author"))); }</pre>	title	author	The Devil in the White City	Erik Larson	The Lion, the Witch and the Wardrobe	C.S. Lewis	In the Garden of Beasts	Erik Larson	
title	author									
The Devil in the White City	Erik Larson									
The Lion, the Witch and the Wardrobe	C.S. Lewis									
In the Garden of Beasts	Erik Larson									
Running Tests in Parallel	<pre>public class TestNGTestRunner extends AbstractTestNGCucumberTests { @Override @DataProvider(parallel = true) public Object[][] scenarios() { return super.scenarios(); } }</pre> <p>Add Surefire plugin. The default thread count of the dataprovider in parallel mode is 10. To change this the dataprovderthreadcount property needs to be added to the configuration section of the Surefire plugin in the POM.</p> <pre><build></pre>									

```

<plugins>
    <plugin>
        <groupId>org.apache.maven.plugins</groupId>
        <artifactId>maven-compiler-plugin</artifactId>
        <version>3.11.0</version>
        <configuration>
            <source>21</source>
            <target>21</target>
            <encoding>UTF-8</encoding>
        </configuration>
    </plugin>
    <plugin>
        <groupId>org.apache.maven.plugins</groupId>
        <artifactId>maven-surefire-plugin</artifactId>
        <version>3.2.2</version>
        <configuration>
            <suiteXmlFiles>
                <suiteXmlFile>testng.xml</suiteXmlFile>
            </suiteXmlFiles>
            <properties>
                <property>
                    <name>dataproviderthreadcount</name>
                    <value>2</value>
                </property>
            </properties>
        </configuration>
    </plugin>
</plugins>
</build>

```

This will run tests within same feature file in parallel.

To run tests across feature file in parallel use testng.xml

```

<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE suite SYSTEM "https://testng.org/testng-1.0.dtd">
<suite name="Suite" parallel="classes" thread-count="2">
    <test name="Test">
        <classes>
            <class name="TestRunner.TestRunner"/>
            <class name="TestRunner.TestRunner2"/>
        </classes>
    </test> <!-- Test -->
</suite> <!-- Suite -->

```

Run failed test cases in cucumber	<pre> @RunWith(Cucumber.class) @CucumberOptions(features = "src/test/resources/features", glue = "stepDefinitions", plugin = {"pretty", "html:target/cucumber-reports", "rerun:target/rerun.txt"}) public class RunCucumberTest { } New Runner @RunWith(Cucumber.class) @CucumberOptions(features = "@target/rerun.txt", glue = "stepDefinitions") public class RerunFailedTests { } </pre>
-----------------------------------	---

Cucumber Options	Cucumber Options are used to configure properties for Cucumber tests
------------------	--

Options Type	Purpose	Default Value
dryRun	true: Checks if all the Steps have the Step Definition	false
features	set: The paths of the feature files	{}
glue	set: The paths of the step definition files	{}
tags	instruct: What tags in the features files should be executed	{}
monochrome	true: Display the console Output in much readable way	false
format	set: What all report formatters to use	false
strict	true: Will fail execution if there are undefined or pending steps	false

TestNG

17 February 2024 13:22

Test Annotation	<ol style="list-style-type: none">1. @Test: Marks a method as a test method2. @BeforeSuite: Marks a method to run before any test suite3. @AfterSuite: Marks a method to run after all tests in a suite have run4. @BeforeTest: Marks a method to run before any test method belonging to the classes inside the <test> tag in XML5. @AfterTest: Marks a method to run after all test methods belonging to the classes inside the <test> tag in XML6. @BeforeClass: Marks a method to run before the first test method in the current class is invoked7. @AfterClass: Marks a method to run after all the test methods in the current class have been run8. @BeforeMethod: Marks a method to run before each test method9. @AfterMethod: Marks a method to run after each test method <pre><suite> <test> <classes> <class> <methods></pre>
@DataProvider	<p>Marks a method as a data provider for test methods. It provides data to a test method.</p> <pre>@DataProvider(name = "simpleData") public Object[][] createData() { return new Object[][] { { "John", 25 }, { "Jane", 30 }, { "Alex", 22 } }; } @Test(dataProvider = "simpleData") public void testWithDataProvider(String name, int age) { System.out.println("Name: " + name + ", Age: " + age); }</pre> <p>DP in another file: <code>@Test(dataProvider = "flights",dataProviderClass= DataProviders.class)</code></p>
@Parameters	<p>Marks a method parameter as a parameter for the test method. It allows passing parameters to test methods from the TestNG XML file.</p> <pre><suite name="Suite"> <test name="Test"> <parameter name="username" value="user123" /> <parameter name="password" value="secret123" /> <classes> <class name="com.example.MyTestClass" /> </classes> </test></suite> @Parameters({ "username", "password" }) @Test public void testLogin(String username, String password) { System.out.println("Username: " + username); System.out.println("Password: " + password); // Your test logic here using the provided parameters }</pre>
Disable a test	<pre>@Test(enabled=false)</pre>
Timeout	<pre>@Test(timeOut=10000)</pre> <p>Or</p> <pre><?xml version="1.0" encoding="UTF-8"?> <!DOCTYPE suite SYSTEM "https://testng.org/testng-1.0.dtd"> <suite name="Suite"> <test thread-count="5" name="Test" time-out="5000"> <classes> <class name="testRunner.TestNGRunner"/></pre>

	<pre></classes> </test> <!-- Test --> </suite> <!-- Suite --></pre>
Exclude Tests	<pre><test name="Test"> <classes> <class name="com.example.MyTestClass"> <methods> <exclude name="testMethodToExclude"/> </methods> </class> </classes> </test></pre> <p>We can also include</p> <p>Regex: <methods> <exclude name="test.*"/> </methods></p>
Exclude Tests Based on Groups	<pre>@Test(groups = {"smoke"}) public void testMethod1() { // Test code for smoke group } <test name="Test"> <groups> <run> <exclude name="smoke"/> <exclude name="regression"/> </run> </groups> <classes> <class name="com.example.MyTestClass"/> </classes> </test></pre>
Run Whole Package	<pre><!DOCTYPE suite SYSTEM "http://testng.org/testng-1.0.dtd"> <suite name="Suite"> <test name="Test"> <packages> <package name="com.example.tests"/> </packages> </test> </suite></pre>
Test Tagging	<pre>@Test(groups = {"smoke", "regression"}) public void testMethod1() { System.out.println("Test Method 1"); } <!DOCTYPE suite SYSTEM "https://testng.org/testng-1.0.dtd"> <suite name="Test Suite"> <test name="Test Methods"> <groups> <run> <include name="smoke"/> </run> </groups> <classes> <class name="org.saurav.TestSetup"/> <class name="org.saurav.TestMethods"/> <class name="org.saurav.TestExecution"/> </classes> </test> </suite></pre> <p><u>to run beforeclass/method/test when using groups give @BeforeClass(alwaysRun = true)</u></p> <pre>@BeforeClass(alwaysRun = true) public void beforeClass() { System.out.println("Before Class - Executed before the first test method in the current class."); } @AfterClass(alwaysRun = true) public void afterClass() { System.out.println("After Class - Executed after all test methods in the current class."); }</pre>

dependsOnMethods	<pre>@Test(groups = {"smoke"}, dependsOnMethods = {"testMethod2"}) else <!DOCTYPE suite SYSTEM "https://testng.org/testng-1.0.dtd"> <suite name="TestNG XML Dependency Suite"> <test name="ToolsQA"> <groups> <dependencies> <group depends-on="openbrowser" name="login"></group> </dependencies> </groups> <classes> <class name="GroupDependency" /> </classes> </test> </suite></pre>				
dependsOnGroups	@Test(groups = { "groupB"}, dependsOnGroups = {"groupA", "group2"})				
Order of Execution in Inheritance	TestNG runs @Before methods from the highest superclass to the subclass, and @After methods from the subclass back to the highest superclass				
preserve-order	<p>TestNG will run your tests in the order they are found in the XML file. If you want the classes and methods listed in this file to be run in an unpredictable order, set the preserve-order attribute to false</p> <pre><!DOCTYPE suite SYSTEM "http://testng.org/testng-1.0.dtd"> <suite name="preserve-order Suite"> <test name="preserve-order Test" preserve-order="true"> <classes> <class name="Testing_Pack.Test_One"/> <class name="Testing_Pack.Test_Two"/> <class name="Testing_Pack.Test_Three"/> </classes> </test> </suite></pre> <p>Tests will run as One,Two,Three, else it would have run as One,three,two</p>				
@Ignore	<pre>@Ignore public class TestcaseSample { @Test public void testMethod1() { } @Test public void testMethod2() { } }</pre>				
Rerunning failed tests	<table border="1"> <tr> <td>Using testng-failed.xml</td><td> <ul style="list-style-type: none"> Run your TestNG suite: After the initial run, TestNG generates a testng-failed.xml file in the test-output folder. This file contains the list of failed test cases. Execute testng-failed.xml: Right-click on this file and run it as a TestNG suite. This will re-execute only the failed test cases. </td></tr> <tr> <td>Using IRetryAnalyzer</td><td> <ul style="list-style-type: none"> Create a RetryAnalyzer class: Implement the IRetryAnalyzer interface and define a retry() method. This method determines whether a test case should be re-tried based on a certain condition (e.g., maximum retry count). Apply the RetryAnalyzer to Test Methods: Use the retryAnalyzer attribute in the @Test annotation to specify the RetryAnalyzer class. Run your TestNG suite: When a test case fails, the RetryAnalyzer will attempt to re-run it up to the specified maximum number of retries. <pre>----Setup RetryAnalyzer import org.testng.IRetryAnalyzer; import org.testng.ITestResult; public class RetryAnalyzer implements IRetryAnalyzer { private int count = 0; private static final int maxRetryCount = 3; public boolean retry(ITestResult result) { if (count < maxRetryCount) {</pre> </td></tr> </table>	Using testng-failed.xml	<ul style="list-style-type: none"> Run your TestNG suite: After the initial run, TestNG generates a testng-failed.xml file in the test-output folder. This file contains the list of failed test cases. Execute testng-failed.xml: Right-click on this file and run it as a TestNG suite. This will re-execute only the failed test cases. 	Using IRetryAnalyzer	<ul style="list-style-type: none"> Create a RetryAnalyzer class: Implement the IRetryAnalyzer interface and define a retry() method. This method determines whether a test case should be re-tried based on a certain condition (e.g., maximum retry count). Apply the RetryAnalyzer to Test Methods: Use the retryAnalyzer attribute in the @Test annotation to specify the RetryAnalyzer class. Run your TestNG suite: When a test case fails, the RetryAnalyzer will attempt to re-run it up to the specified maximum number of retries. <pre>----Setup RetryAnalyzer import org.testng.IRetryAnalyzer; import org.testng.ITestResult; public class RetryAnalyzer implements IRetryAnalyzer { private int count = 0; private static final int maxRetryCount = 3; public boolean retry(ITestResult result) { if (count < maxRetryCount) {</pre>
Using testng-failed.xml	<ul style="list-style-type: none"> Run your TestNG suite: After the initial run, TestNG generates a testng-failed.xml file in the test-output folder. This file contains the list of failed test cases. Execute testng-failed.xml: Right-click on this file and run it as a TestNG suite. This will re-execute only the failed test cases. 				
Using IRetryAnalyzer	<ul style="list-style-type: none"> Create a RetryAnalyzer class: Implement the IRetryAnalyzer interface and define a retry() method. This method determines whether a test case should be re-tried based on a certain condition (e.g., maximum retry count). Apply the RetryAnalyzer to Test Methods: Use the retryAnalyzer attribute in the @Test annotation to specify the RetryAnalyzer class. Run your TestNG suite: When a test case fails, the RetryAnalyzer will attempt to re-run it up to the specified maximum number of retries. <pre>----Setup RetryAnalyzer import org.testng.IRetryAnalyzer; import org.testng.ITestResult; public class RetryAnalyzer implements IRetryAnalyzer { private int count = 0; private static final int maxRetryCount = 3; public boolean retry(ITestResult result) { if (count < maxRetryCount) {</pre>				

	<pre> count++; return true; } return false; } } -----Applying RetryAnalyzer to a Test Method @Test(retryAnalyzer = RetryAnalyzer.class) public void testMethod() { // Test logic here } </pre>
@Factory Annotation	<pre> import org.testng.annotations.Factory; public class TestFactory { @Factory public Object[] factoryMethod() { return new Object[] { new TestClass(1), new TestClass(2) }; } } // Test class import org.testng.annotations.Test; public class TestClass { private int parameter; public TestClass(int parameter) { this.parameter = parameter; } @Test public void testMethod() { System.out.println("Test method is running with parameter: " + parameter); // Add your test logic here using the 'parameter' } } </pre>
@Test(invocationCount = 3)	<p>InvocationCount is used when we want to run the same test multiple times.</p> <p>threadPoolSize controls how many threads TestNG will use to execute the invocations of a test method</p> <p>Ex: @Test(threadPoolSize = 3, invocationCount = 6)</p> <pre> public void testMethod() { // Test logic } </pre>
TestNG Priorities	<p>@Test (priority = 1)</p> <p>If we have methods with priority 0,1,2,3 and some without any priority, First: 0 Priority Second: No priority Third: 1</p> <p>In case of conflict, alphabetical order of testName</p>
Soft Assert	<pre> SoftAssert softassert = new SoftAssert(); softassert.assertEquals(originalTitle, expectedTitle); softassert.assertAll(); </pre>
Parallel	<ul style="list-style-type: none"> Methods: This will run the parallel tests on all @Test methods in TestNG. Tests: All the test cases present inside the <test> tag will run with this value. Classes: All the test cases present inside the classes that exist in the XML will run in parallel. Instances: This value will run all the test cases parallelly inside the same instance. <pre> <?xml version="1.0" encoding="UTF-8"?> <!DOCTYPE suite SYSTEM "https://testng.org/testng-1.0.dtd"> <suite name="Suite" parallel="methods" thread-count="2"> <test name="Test"> <parameter name="userName" value="saurav"/> <classes> <class name="TestRunner.TestNGTest"/> </pre>

	<pre> </classes> </test> <!-- Test --> <test name="Test2"> <parameter name="userName" value="singh"/> <classes> <class name="TestRunner.TestNGTest2"/> </classes> </test> </suite><!-- Suite --> default thread count for parallel execution is 5 </pre>
Exception handling	<pre> @Test(expectedExceptions = NumberFormatException.class) public void testNumberFormatException() { Integer.parseInt("abc"); // This will throw a NumberFormatException } </pre> <p>This test will pass if exception occurs and fail when it doesn't</p>
ITestContext	<p>ITestContext is used to store and share data across the tests in selenium by using TestNG framework.</p> <p>This interface provides information about the current test execution context. It allows you to retrieve data such as the suite name, test name, test parameters, and test status.</p> <ul style="list-style-type: none"> • <code>getSuite()</code>: Get information about the test suite. • <code>getName()</code>: Get the name of the test. • <code>getStartDate()</code>: Get the start time of the test execution. • <code>getEndDate()</code>: Get the end time of the test execution. • <code>setAttribute(String name, Object value)</code>: Set an attribute in the context.//test level variable • <code>getSuite().setAttribute(String name, Object value)</code>: Set an attribute in the suite.//suite level variable • <code>getAttribute(String name)</code>: Retrieve an attribute from the context <pre> @BeforeTest public void setData(ITestContext context){ String Customer_id = "C11012034"; context.setAttribute("CustID", Customer_id); System.out.println("Value is stored in ITestContext"); System.out.println("++++++++++++++++++++++"); } @Test public void Test1a(ITestContext context){ String Customer_id1 = (String) context.getAttribute("CustID"); System.out.println("In Test1, Value stored in context is: "+Customer_id1); System.out.println("++++++++++++++++++++++"); } </pre>
ITestListener	<p>ITestListener is an interface implemented in the class, and that class overrides the ITestListener-defined methods. It listens to the desired events and executes the methods accordingly.</p> <ul style="list-style-type: none"> • <code>onStart()</code>: invoked after test class is instantiated and before execution of any testNG method. • <code>onTestSuccess()</code>: invoked on the success of a test • <code>onTestFailure()</code>: invoked on the failure of a test • <code>onTestSkipped()</code>: invoked when a test is skipped • <code>onTestFailedButWithinSuccessPercentage()</code>: invoked whenever a method fails but within the defined success percentage • <code>onFinish()</code>: invoked after all tests of a class are executedThe above-mentioned methods use the parameters ITestContext and ITestResult. The ITestContext is a class that contains information about the test run. The ITestResult is an interface that defines the result of the test. <pre> import org.testng.ITestContext; import org.testng.ITestListener; import org.testng.ITestResult; public class MyListener implements ITestListener { @Override public void onFinish(ITestContext contextFinish) { System.out.println("onFinish method finished"); } </pre>

```

@Override
public void onStart(ITestContext contextStart) {
System.out.println("onStart method started");
}

@Override
public void onTestFailedButWithinSuccessPercentage(ITestResult result) {
System.out.println("Method failed with certain success percentage"+ result.getName());

}

@Override
public void onTestFailure(ITestResult result) {
System.out.println("Method failed"+ result.getName());

}

@Override
public void onTestSkipped(ITestResult result) {
System.out.println("Method skipped"+ result.getName());

}

@Override
public void onTestStart(ITestResult result) {
System.out.println("Method started"+ result.getName());

}

@Override
public void onTestSuccess(ITestResult result) {
System.out.println("Method passed"+ result.getName());

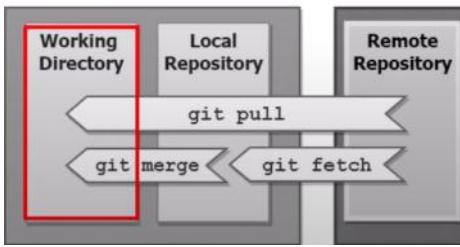
}
}

```

AutoRead Prop Files	Default location src/test/resources ResourceBundle.getBundle("routes");
Pass Data from xml to cucumber runner	<pre> public class TestRunner extends AbstractTestNGCucumberTests { @Override @DataProvider(parallel = false) public Object[][] scenarios() { return super.scenarios(); } @Parameters({ "browser" }) @BeforeMethod public static void setUpScenario(String browser) { WebDriverManager.initDriver(browser); } @AfterMethod public void tearDown(){ WebDriverManager.quitBrowser(); } } </pre>

Git

29 November 2024 08:21

Command	Description
git init	Initializes a new Git repository in the current directory.
git clone <url>	Clones an existing Git repository from a remote URL.
git add <file>	Adds a file to the staging area.
git commit -m "<message>"	Commits the staged changes with a message.
git status	Shows the current state of the working directory.
git diff	Shows the difference between the current state and the previous commit.
git log	Shows the commit history.
git checkout <branch>	Switches to a different branch.
git branch <branch-name>	Creates a new branch.
git checkout -b <name>	Creates and switches
git checkout commitHash	<p>used to switch your working directory to the state of a specific commit. This allows you to view or inspect a project's state at a particular point in history. However, this action places you in a detached HEAD state, meaning you're not on any branch.</p> <p>You are not on any branch. Any changes made here will not be associated with a branch unless you explicitly create one.</p> <p>If you make changes and want to keep them, create a new branch: git checkout -b new-branch-name</p> <p>Modern Alternative: git switch --detach <commitHash></p>
git merge <branch>	Merges a branch into the current branch. git checkout master git pull origin master git checkout subBranch git merge master
git rebase -i HEAD~2	Combines multiple commits into a single commit to maintain a linear commit history add squash text before other commits to be merged
git push <remote> <branch> -f to force when you revert commit in local	Pushes changes to a remote repository.
git pull <remote> <branch>	Pulls changes from a remote repository.
git remote add <name> <url>	Adds a remote repository.
git remote show <name>	Shows information about a remote repository.
git fetch <remote>	Fetches changes from a remote repository to local repository. Need to merge in order to see changes in working directory.
	 <p>The diagram illustrates the interaction between three components: Working Directory, Local Repository, and Remote Repository. It shows the flow of data and commands:</p> <ul style="list-style-type: none">Working Directory (highlighted with a red box) interacts with the Local Repository and the Remote Repository.git pull: An arrow points from the Remote Repository to the Local Repository.git merge: An arrow points from the Working Directory to the Local Repository.git fetch: An arrow points from the Remote Repository to the Local Repository.
git stash	Temporarily saves changes to the working directory.
git stash pop	Restores the most recently stashed changes.
git config --global user.name "<name>"	Sets the global user name.
git config --global user.email "<email>"	Sets the global user email.
git config --global core.editor "<editor>"	Sets the global default editor.
Delete untracked files	git clean -f
git reset	Staging to Working Directory

git reset --soft HEAD~1	revert commit and changes move to staging, hard will delete permanently. after ~ we can also give commit hash above which we want to delete things we can also give commit hash
git reset --hard HEAD~1	Revert commit and delete data
git revert <commit-hash>	git revert is used to undo a specific commit by creating a new commit that reverses the changes made by the original commit. Unlike git reset, which can rewrite history, git revert is a safe operation because it doesn't alter the commit history, making it ideal for collaborative workflows.
git revert <commit-hash1><commit-hash2>	For Conflicts git add <file> git revert --continue git push
git bisect start git bisect bad [<commit>] git bisect good [<commit>]	used to perform a binary search through a repository's commit history to identify the commit that introduced a bug or issue. It is particularly useful in large repositories with numerous commits, as it narrows down the problematic commit efficiently. The process involves marking a good commit (where the code works) and a bad commit (where the bug exists). Git then automatically checks out commits in the middle of the range for you to test, helping narrow down the problematic commit. Git will now check out a commit halfway between the good and bad commits. Test this commit to determine if it's good or bad: If the commit is bad, mark it as: git bisect bad else git bisect good
git commit --amend -m "New Message"	Fix a Commit Message
git add <file> git commit --amend	Add Changes to the Last Commit
git reset --soft HEAD~1 git restore --staged <file> git commit --amend	Remove Changes from the Last Commit

Aspect	Merge	Rebase	Squash
History	Preserves branch history with merge commits	Creates a linear history. Reapplies commits from one branch on top of another, creating a linear history by rewriting commit history.	Combines multiple commits into one
Commit Granularity	Keeps all original commits intact Creates One more commit	Keeps individual commits intact No Additional Commits	Reduces all commits to a single one
Complexity	Simple, no rewriting	More complex, rewrites history	Requires manual consolidation
Risk of Conflicts	During the merge process only	During the rebase process	During squashing, if conflicts exist
Use in Shared Branches	Safe	Risky	Risky unless handled carefully
Clean History	Can be messy	Clean and linear	Very clean and concise

Maven

16 February 2024 08:36

Maven is a build automation tool used primarily for Java projects. It simplifies project management by handling:

- Dependency management (Automatic JAR downloads)
- Build process (Compiling, testing, packaging, and deploying)
- Project structure standardization
- Plugin-based extensibility

Maven Coordinates: Group id, artifact id, packaging and Version

Maven Lifecycle

clean	Removes previous build artifacts (target/ folder)
validate	Checks if the project is correct
compile	Compiles the source code (.java → .class)
test	Runs unit tests (src/test/java)
package	Creates a JAR/WAR (target/my-app.jar)
verify	Runs integration tests (src/test/java using Failsafe)
install	Installs the package into the local Maven repository
deploy	Deploys the package to a remote repository
site	Generates project documentation

mvn clean install: This will **clean, compile, test, package, verify, and install** your project.

Settings.xml is used to modify behavior of Maven

Bin Folder: Contains scripts for executing Maven.

Boot: Contains a jar file responsible for creating a class loader for Maven execution

Conf: This has settings.xml which when customised is placed under m2 directory

Lib: Contains Maven core classes for building. Includes Maven dependencies and Java components necessary for Maven's operation.

mvn archetype:generate -DarchetypeGroupId=org.apache.maven.archetypes -DarchetypeArtifactId=maven-archetype-quickstart -DgroupId=com.example -DartifactId=my-project -Dversion=1.0-SNAPSHOT -DinteractiveMode=false

Maven goals represent individual tasks that can be executed during the build process, while plugins are extensions that provide additional functionality and encapsulate collections of goals. Together, goals and plugins enable Maven to efficiently manage and automate the build lifecycle of a project

Scopes

1. Compile is default scope: Available in all classpaths and propagated to dependent projects
2. test: Only available for test compilation and execution phases
3. provided: Available during compilation and testing, but not at runtime. Not for packaging
4. runtime: Not available during compilation, but available at runtime. Not for Compilation

maven-surefire-plugin (For Unit Testing)	Runs JUnit and TestNG tests in src/test/java(*test.java) Used in the test phase Fails the build on test failures
maven-failsafe-plugin (For Integration Testing)	Runs integration tests (*IT.java) Executes in integration-test & verify phases Ensures tests don't break the build immediately

Extent Reports

25 March 2025 08:57

```
public class ExtentReportManager {  
  
    public static ExtentReports extentReports;  
  
    public static ExtentReports createInstance(String fileName, String reportName, String  
documentTitle) {  
        ExtentSparkReporter extentSparkReporter = new ExtentSparkReporter(fileName);  
        extentSparkReporter.config().setReportName(reportName);  
        extentSparkReporter.config().setDocumentTitle(documentTitle);  
        extentSparkReporter.config().setTheme(Theme.STANDARD);  
        extentSparkReporter.config().setEncoding("utf-8");  
        extentReports = new ExtentReports();  
        extentReports.attachReporter(extentSparkReporter);  
        return extentReports;  
    }  
  
    public static String getReportNameWithTimeStamp() {  
        DateTimeFormatter dateTimeFormatter =  
DateTimeFormatter.ofPattern("yyyy_MM_dd_HH_mm_ss");  
        LocalDateTime localDateTime = LocalDateTime.now();  
        String formattedTime = dateTimeFormatter.format(localDateTime);  
        return "TestReport" + formattedTime + ".html";  
    }  
  
    public static void logPassDetails(String log) {  
        ExtentReportConfigListener.extentTest.get().pass(MarkupHelper.createLabel(log,  
ExtentColor.GREEN));  
    }  
    public static void logFailureDetails(String log) {  
        ExtentReportConfigListener.extentTest.get().fail(MarkupHelper.createLabel(log,  
ExtentColor.RED));  
    }  
}
```

```
public class ExtentReportConfigListener implements ITestListener {  
  
    private static ExtentReports extentReports;  
    public static ThreadLocal<ExtentTest> extentTest = new ThreadLocal<>();  
  
    public void onStart(ITestContext context) {  
        String fileName = ExtentReportManager.getReportNameWithTimeStamp();  
        String fullReportPath = System.getProperty("user.dir") + "\\reports\\" + fileName;  
        extentReports = ExtentReportManager.createInstance(fullReportPath, "Test API  
Automation Report",  
                "Test ExecutionReport");  
    }  
  
    public void onFinish(ITestContext context) {  
        if (extentReports != null)  
            extentReports.flush();  
    }  
}
```

```
}

public void onTestStart(ITestResult result) {
    ExtentTest test = extentReports.createTest(
        "Test Name " + result.getTestClass().getName() + " - " +
        result.getMethod().getMethodName());
    extentTest.set(test);
}

public void onTestFailure(ITestResult result) {
    ExtentReportManager.logFailureDetails(result.getThrowable().getMessage());
    String stackTrace = Arrays.toString(result.getThrowable().getStackTrace());
    stackTrace = stackTrace.replaceAll(", ", "<br>");
    String formmatedTrace = "<details>\n" + "  <summary>Click Here To See Exception
Logs</summary>\n" + "  "
        + stackTrace + "\n" + "</details>\n";
    ExtentReportManager.logExceptionDetails(formmatedTrace);
}
}
```

Log4j2.xml

25 March 2025 08:58

```
<?xml version="1.0" encoding="UTF-8"?>
<Configuration status="INFO">
    <Properties>
        <Property name="basePath">./logs</Property>
    </Properties>
    <Appenders>
        <Console name="Console" target="SYSTEM_OUT">
            <PatternLayout pattern="%d{HH:mm:ss.SSS} [%t] %-5level %logger{36} - %msg%n"/>
        </Console>
        <RollingFile name="File" fileName="${basePath}/automation.log"
            filePattern="${basePath}/automation-%d{yyyy-MM-dd}.log">
            <PatternLayout pattern="%d{HH:mm:ss.SSS} [%t] %-5level %logger{36} - %msg%n"/>
            <SizeBasedTriggeringPolicy size="50000"/>
        </RollingFile>
    </Appenders>
    <Loggers>
        <Root level="Info"><!-- Here you can use multiple log levels All < Trace < Debug < Info < Warn < Error < Fatal < Off -->
            <AppenderRef ref="File"/> <!-- <AppenderRef ref="Console"/> -->
        </Root>
    </Loggers>
</Configuration>
```

<Configuration status="WARN">	<ul style="list-style-type: none">The <Configuration> tag is the root element that encapsulates all Log4j2 configurations.The status attribute controls Log4j2's internal debugging logs, which are useful for diagnosing configuration issues.Options for status:<ul style="list-style-type: none">OFF: No internal logs from Log4j2.ERROR: Logs only errors in Log4j2 configuration.WARN (default): Logs warnings and errors about configuration issues.INFO: Provides more detailed information, such as component initialization.DEBUG: Outputs detailed logs, including internal processing of Log4j2.TRACE: Most verbose, showing fine-grained details of Log4j2 operations	
<Properties> <Property name="basePath">./logs</Property> </Properties>	<p>Defines reusable key-value pairs for configuration. These properties can be referenced elsewhere in the file using \${propertyName}.</p> <p>Common Usage: Set file paths, environment variables, or dynamic configurations. Example: \${env:USER_HOME} to get the user's home directory.</p>	
<Appenders> ... </Appenders>	Console Appender	<pre><Console name="Console" target="SYSTEM_OUT"> <PatternLayout pattern="%d{HH:mm:ss.SSS} [%t] %-5level %logger{36} - %msg%n"/> </Console></pre> <p>Purpose: Logs messages to the console (stdout or stderr). Options for target: SYSTEM_OUT: Writes logs to standard output (stdout). SYSTEM_ERR: Writes logs to standard error (stderr).</p>
	File Appender	<pre><RollingFile name="File" fileName="\${basePath}/automation.log" filePattern="\${basePath}/automation-%d{yyyy-MM-dd}.log"> ... </RollingFile></pre> <p>Purpose: Writes logs to files and supports file rollover.</p> <p>Attributes: fileName: The current log file. filePattern: The pattern for rollover files (e.g., \${basePath}/automation-%d{yyyy-MM-dd}.log for daily logs).</p> <p>Triggering Policies:<ul style="list-style-type: none">SizeBasedTriggeringPolicy: Rolls over when the file exceeds a specific size:<SizeBasedTriggeringPolicy size="50MB"/>TimeBasedTriggeringPolicy: Rolls over at specific time intervals.:<TimeBasedTriggeringPolicy interval="1" modulate="true"/>CompositeTriggeringPolicy: <CompositeTriggeringPolicy> <Policies></p>

		<pre> <SizeBasedTriggeringPolicy size="50MB"/> <TimeBasedTriggeringPolicy interval="1"/> </Policies> </CompositeTriggeringPolicy> </pre>
PatternLayout		<p><PatternLayout pattern="%d{HH:mm:ss.SSS} [%t] %-5level %logger{36} - %msg%n"/></p> <p>Common Pattern Elements:</p> <p>%d{pattern}: Date and time (e.g., %d{yyyy-MM-dd HH:mm:ss}).</p> <p>%t: Thread name.</p> <p>%level: Log level (INFO, DEBUG, etc.).</p> <p>%logger{length}: Logger name truncated to the specified length.</p> <p>%msg: The actual log message.</p> <p>%n: Newline.</p>
Loggers Section	Root Custom Loggers	<p>Root</p> <pre> <Root level="Info"> <AppenderRef ref="File"/> </Root> </pre> <p>The root logger applies globally to all loggers unless overridden.</p> <p>Attributes:</p> <p>level: Minimum level of logs to capture (e.g., DEBUG, INFO, WARN).</p> <p>AppenderRef: Specifies which appenders the logger uses.</p> <p>Custom Loggers</p> <pre> <Logger name="com.example" level="DEBUG" additivity="false"> <AppenderRef ref="Console"/> </Logger> </pre> <p>Attributes:</p> <p>name: Logger name (usually a package or class name).</p> <p>level: Minimum log level for this logger.</p> <p>additivity: Whether to inherit appenders from parent loggers.</p>

WebDriverListener

25 March 2025 08:59

Introduction	It intercepts and logs various events that occur during WebDriver interactions, providing insights into the execution flow and potential issues.
DriverListener	<pre>public class DriverListener implements WebDriverListener { private final WebDriver driver; public DriverListener(WebDriver driver) { this.driver = driver; } @Override public void beforeGetTitle(WebDriver driver) { System.out.println("Getting the title"); } @Override public void beforeClick(WebElement element) { takeScreenshot("before_click"); } @Override public void afterClick(WebElement element) { takeScreenshot("after_click"); } @Override public void onError(Object target, Method method, Object[] args, InvocationTargetException e) { takeScreenshot("On Error"); } // public void takeScreenshot(String message) { // String base64Screenshot = ((TakesScreenshot) driver).getScreenshotAs(OutputType.BASE64); // ExtentCucumberAdapter.addTestStepLog("Screenshot"); // ExtentCucumberAdapter.getCurrentStep() // .info(message, MediaEntityBuilder.createScreenCaptureFromBase64String(base64Screenshot).build()); // } public void takeScreenshot(String message) { try { File screenshot = ((TakesScreenshot) driver).getScreenshotAs(OutputType.FILE); String timestamp = new SimpleDateFormat("dd_MMM_yy_HH_mm_ss").format(new Date()); String screenshotName = "screenshot_" + timestamp + ".png"; String destPath = System.getProperty("user.dir") + "/ExtentReports/screenshots/" + screenshotName; FileUtils.copyFile(screenshot, new File(destPath)); ExtentCucumberAdapter.addTestStepLog("Screenshot"); ExtentCucumberAdapter.getCurrentStep().info(message, MediaEntityBuilder.createScreenCaptureFromPath(destPath).build()); } catch (IOException e) { e.printStackTrace(); } } }</pre>
Extent.properties	extent.reporter.spark.start=true extent.reporter.spark.out=/AutomationTestResult.html #FolderName basefolder.name=ExtentReports/SparkReport_ basefolder.datetimepattern=d_MMM_YY HH_mm_ss
EventFiringDecorator	EventFiringDecorator acts as a wrapper (a design pattern called the Decorator pattern) around your actual WebDriver instance (e.g., ChromeDriver, FirefoxDriver) When you call a method on decoratedDriver (e.g., decoratedDriver.get("...") or button.click()), the EventFiringDecorator intercepts that call. <ul style="list-style-type: none">• Before Event: Before executing the actual method on the originalDriver, the EventFiringDecorator checks if the corresponding "before" method is implemented in any of the registered WebDriverListeners. If beforeClick is implemented in MyLoggingListener, it calls listener.beforeClick(element).<ul style="list-style-type: none">◦ Actual Method Execution: The EventFiringDecorator then calls the actual click() method on the originalDriver.• After Event: After the actual method completes successfully, the EventFiringDecorator checks for and calls the corresponding "after" method (e.g., listener.afterClick(element)).• Error Event: If the actual method throws an exception, the EventFiringDecorator catches it and calls the listener.onError() method.
Decorate Your Driver	// 1. Create your actual WebDriver instance WebDriver originalDriver = new ChromeDriver(); // 2. Create your listener WebDriverListener listener = new MyLoggingListener(); // 3. Decorate your original driver with the listener

```
WebDriver decoratedDriver = new EventFiringDecorator(listener).decorate(originalDriver);

// Now, use the decoratedDriver for your interactions
decoratedDriver.get("https://www.example.com");

WebElement button = decoratedDriver.findElement(By.id("someButton"));
button.click(); // This will trigger the beforeClick and afterClick events in MyLoggingListener

decoratedDriver.quit();
```

Miscellaneous

21 May 2018 14:28

Screenshot of web element	<pre>WebElement logo = driver.findElement(By.xpath("//div[@id='divLogo']//img")); File file = logo.getScreenshotAs(OutputType.FILE); File destFile = new File("logo.png"); FileUtils.copyFile(file, destFile);</pre>		
Open a new tab	<p>In Selenium 4, users can open a URL in a new tab along with the existing tab.</p> <p>For example, if the user wants to open 2 URLs in two different tabs simultaneously, it can be done with Selenium 4.</p> <pre>driver.get("https://www.google.com/"); driver.switchTo().newWindow(WindowType.TAB); driver.navigate().to("https://www.crmpro.com/"); //switch to().window() is still required to operate on original window</pre>		
Open a new window	<p>Using Selenium 4, users can open a new window on the same browser.</p> <p>For example, if the user wants to access two applications in the same browser, it can now be achieved.</p> <pre>driver.get("https://www.google.com/"); driver.switchTo().newWindow(WindowType.WINDOW); driver.navigate().to("https://www.crmpro.com/"); //switch to().window() is still required to operate on original window</pre>		
Object Location	<p>Selenium 4 enables users to obtain coordinates, dimensions, height, width, etc., as the location of web elements or objects.</p> <pre>WebElement logo1 = driver.findElement(By.xpath("//div[@id='divLogo']//img")); System.out.println("Height: " + logo1.getRect().getDimension().getHeight()); System.out.println("Width: " + logo1.getRect().getDimension().getWidth()); System.out.println("X Location: " + logo1.getRect().getX()); System.out.println("Y Location: " + logo1.getRect().getY());</pre>		
Relative Locators	<p>These are also known as Friendly Locators, and this functionality is being added to find out the element which is present nearby to other web element or, we can say that it can find the web elements based on GUI location.</p> <p>There are five locators added in Selenium 4:</p> <ul style="list-style-type: none"> below(): Web element located below for the specified element. toLeftOf(): Target web element which is present to the left of specified element. toRightOf(): Target web element which is presented to the right of a specified element. above(): Web element located above for the specified element. near(): Target web element which is away(approx. 50 pixels) from the specified element. <p>sendKeys to Email Field:</p> <pre>driver.findElement(RelativeLocator.with(By.tagName("input")).above(password)).sendKeys("ab@rmal.com");</pre> <p>Chaining of Relative Locators</p> <pre>By submitLocator = RelativeLocator.with(By.tagName("button")).below(By.id("email")).toRightOf(By.id("cancel"));</pre>		
Chrome Dev tools	<p>Selenium 4 introduces changes in the Chrome DevTools API. The Chrome driver class now extends the Chromium Driver class, providing access to predefined methods for dev tool operations.</p> <p>Note: Chromium Driver extends the Remote Web driver class.</p> <p>Operations include:</p> <ul style="list-style-type: none"> - Enable Network Offline - Enable Network Online - Get Console Logs - Load Insecure Website <table border="1"> <tr> <td>Browser Emulation</td> <td> <pre>@Test public void testMobileEmulation() throws InterruptedException { WebDriverManager.chromedriver().setup(); ChromeDriver driver=new ChromeDriver(); driver.manage().window().maximize(); DevTools devtools=driver.getDevTools(); devtools.createSession(); devtools.send(Emulation .setDeviceMetricsOverride(600, 1000, 50, true, Optional.empty(), Optional.empty(), Optional.empty(), Optional.empty(), Optional.empty(), Optional.empty(), Optional.empty(), Optional.empty())); driver.get("https://rahulshettyacademy.com/angularAppdemo/");</pre> </td> </tr> </table>	Browser Emulation	<pre>@Test public void testMobileEmulation() throws InterruptedException { WebDriverManager.chromedriver().setup(); ChromeDriver driver=new ChromeDriver(); driver.manage().window().maximize(); DevTools devtools=driver.getDevTools(); devtools.createSession(); devtools.send(Emulation .setDeviceMetricsOverride(600, 1000, 50, true, Optional.empty(), Optional.empty(), Optional.empty(), Optional.empty(), Optional.empty(), Optional.empty(), Optional.empty(), Optional.empty())); driver.get("https://rahulshettyacademy.com/angularAppdemo/");</pre>
Browser Emulation	<pre>@Test public void testMobileEmulation() throws InterruptedException { WebDriverManager.chromedriver().setup(); ChromeDriver driver=new ChromeDriver(); driver.manage().window().maximize(); DevTools devtools=driver.getDevTools(); devtools.createSession(); devtools.send(Emulation .setDeviceMetricsOverride(600, 1000, 50, true, Optional.empty(), Optional.empty(), Optional.empty(), Optional.empty(), Optional.empty(), Optional.empty(), Optional.empty(), Optional.empty())); driver.get("https://rahulshettyacademy.com/angularAppdemo/");</pre>		

	<pre>driver.findElement(By.cssSelector(".navbar-toggler")).click(); Thread.sleep(3000); driver.findElement(By.linkText("Library")).click();</pre>																						
Geo Location	<pre>@Test public void geoTest() { WebDriverManager.chromedriver().setup(); ChromeDriver driver=new ChromeDriver(); driver.manage().window().maximize(); DevTools devtools=driver.getDevTools(); devtools.createSession(); Map<String, Object> geoLocale=new HashMap<>(); geoLocale.put("latitude", 35.652832); geoLocale.put("longitude", 139.839478); geoLocale.put("accuracy", 100); driver.executeCdpCommand("Emulation.setGeolocationOverride",geoLocale); driver.get("https://www.google.com"); driver.navigate().refresh(); }</pre>																						
Action Class Methods	<p><code>click(WebElement)</code>: Replaces <code>moveToElement(onElement).click()</code>. Used for clicking on a specific web element.</p> <p><code>clickAndHold(WebElement)</code>: Replaces <code>moveToElement(onElement).clickAndHold()</code>. Clicks on an element without releasing the click.</p> <p><code>contextClick(WebElement)</code>: Replaces <code>moveToElement(onElement).contextClick()</code>. Performs the right-click operation.</p> <p><code>doubleClick(WebElement)</code>: Replaces <code>moveToElement(element).doubleClick()</code>. Performs a double click on an element.</p>																						
Options	<p>In Selenium 3, capabilities were defined in a session by using Desired Capabilities classes. As of Selenium 4, you must use the browser options classes. For remote driver sessions, a browser options instance is required as it determines which browser will be used.</p> <table border="1"> <tr> <td><code>browserName</code></td><td>Browser name is set by default when using an Options class instance. <pre>ChromeOptions chromeOptions = new ChromeOptions(); String name = chromeOptions.getBrowserName();</pre></td></tr> <tr> <td><code>browserVersion</code></td><td>This capability is optional, this is used to set the available browser version at remote end. In recent versions of Selenium, if the version is not found on the system, it will be automatically downloaded by Selenium Manager <pre>ChromeOptions chromeOptions = new ChromeOptions(); String version = "latest"; chromeOptions.setBrowserVersion(version);</pre></td></tr> <tr> <td><code>pageLoadStrategy</code></td><td> <p>The <code>document.readyState</code> property of a document describes the loading state of the current document.</p> <table border="1"> <thead> <tr> <th>Strategy</th><th>Ready State</th><th>Purpose</th></tr> </thead> <tbody> <tr> <td>normal</td><td>complete</td><td>Used by default, waits for all resources to download</td></tr> <tr> <td>eager</td><td>interactive</td><td>DOM access is ready, but other resources like images may still be loading</td></tr> <tr> <td>none</td><td>Any</td><td>Does not block WebDriver at all</td></tr> </tbody> </table> <p>When navigating to a new page via URL, by default, WebDriver will hold off on completing a navigation method (e.g., <code>driver.navigate().get()</code>) until the document ready state is complete. This does not necessarily mean that the page has finished loading, especially for sites like Single Page Applications that use JavaScript to dynamically load content after the Ready State returns complete. Note also that this behavior does not apply to navigation that is a result of clicking an element or submitting a form.</p> <p>If a page takes a long time to load as a result of downloading assets (e.g., images, css, js) that aren't important to the automation, you can change from the default parameter of normal to eager or none to speed up the session. This value applies to the entire session, so make sure that your waiting strategy is sufficient to minimize flakiness.</p> </td></tr> <tr> <td><code>platformName</code></td><td> <p>This identifies the operating system at the remote-end, fetching the <code>platformName</code> returns the OS name.</p> <p>In cloud-based providers, setting <code>platformName</code> sets the OS at the remote-end.</p> <pre>ChromeOptions chromeOptions = new ChromeOptions(); String platform = "OS X 10.6"; chromeOptions.setPlatformName(platform);</pre> </td></tr> <tr> <td><code>acceptInsecureCerts</code></td><td> <p>This capability checks whether an expired (or) invalid TLS Certificate is used while navigating during a session.</p> <p>If the capability is set to false, an insecure certificate error will be returned as navigation encounters any domain certificate problems. If set to true, invalid certificate will be trusted by the browser.</p> </td></tr> </table>	<code>browserName</code>	Browser name is set by default when using an Options class instance. <pre>ChromeOptions chromeOptions = new ChromeOptions(); String name = chromeOptions.getBrowserName();</pre>	<code>browserVersion</code>	This capability is optional, this is used to set the available browser version at remote end. In recent versions of Selenium, if the version is not found on the system, it will be automatically downloaded by Selenium Manager <pre>ChromeOptions chromeOptions = new ChromeOptions(); String version = "latest"; chromeOptions.setBrowserVersion(version);</pre>	<code>pageLoadStrategy</code>	<p>The <code>document.readyState</code> property of a document describes the loading state of the current document.</p> <table border="1"> <thead> <tr> <th>Strategy</th><th>Ready State</th><th>Purpose</th></tr> </thead> <tbody> <tr> <td>normal</td><td>complete</td><td>Used by default, waits for all resources to download</td></tr> <tr> <td>eager</td><td>interactive</td><td>DOM access is ready, but other resources like images may still be loading</td></tr> <tr> <td>none</td><td>Any</td><td>Does not block WebDriver at all</td></tr> </tbody> </table> <p>When navigating to a new page via URL, by default, WebDriver will hold off on completing a navigation method (e.g., <code>driver.navigate().get()</code>) until the document ready state is complete. This does not necessarily mean that the page has finished loading, especially for sites like Single Page Applications that use JavaScript to dynamically load content after the Ready State returns complete. Note also that this behavior does not apply to navigation that is a result of clicking an element or submitting a form.</p> <p>If a page takes a long time to load as a result of downloading assets (e.g., images, css, js) that aren't important to the automation, you can change from the default parameter of normal to eager or none to speed up the session. This value applies to the entire session, so make sure that your waiting strategy is sufficient to minimize flakiness.</p>	Strategy	Ready State	Purpose	normal	complete	Used by default, waits for all resources to download	eager	interactive	DOM access is ready, but other resources like images may still be loading	none	Any	Does not block WebDriver at all	<code>platformName</code>	<p>This identifies the operating system at the remote-end, fetching the <code>platformName</code> returns the OS name.</p> <p>In cloud-based providers, setting <code>platformName</code> sets the OS at the remote-end.</p> <pre>ChromeOptions chromeOptions = new ChromeOptions(); String platform = "OS X 10.6"; chromeOptions.setPlatformName(platform);</pre>	<code>acceptInsecureCerts</code>	<p>This capability checks whether an expired (or) invalid TLS Certificate is used while navigating during a session.</p> <p>If the capability is set to false, an insecure certificate error will be returned as navigation encounters any domain certificate problems. If set to true, invalid certificate will be trusted by the browser.</p>
<code>browserName</code>	Browser name is set by default when using an Options class instance. <pre>ChromeOptions chromeOptions = new ChromeOptions(); String name = chromeOptions.getBrowserName();</pre>																						
<code>browserVersion</code>	This capability is optional, this is used to set the available browser version at remote end. In recent versions of Selenium, if the version is not found on the system, it will be automatically downloaded by Selenium Manager <pre>ChromeOptions chromeOptions = new ChromeOptions(); String version = "latest"; chromeOptions.setBrowserVersion(version);</pre>																						
<code>pageLoadStrategy</code>	<p>The <code>document.readyState</code> property of a document describes the loading state of the current document.</p> <table border="1"> <thead> <tr> <th>Strategy</th><th>Ready State</th><th>Purpose</th></tr> </thead> <tbody> <tr> <td>normal</td><td>complete</td><td>Used by default, waits for all resources to download</td></tr> <tr> <td>eager</td><td>interactive</td><td>DOM access is ready, but other resources like images may still be loading</td></tr> <tr> <td>none</td><td>Any</td><td>Does not block WebDriver at all</td></tr> </tbody> </table> <p>When navigating to a new page via URL, by default, WebDriver will hold off on completing a navigation method (e.g., <code>driver.navigate().get()</code>) until the document ready state is complete. This does not necessarily mean that the page has finished loading, especially for sites like Single Page Applications that use JavaScript to dynamically load content after the Ready State returns complete. Note also that this behavior does not apply to navigation that is a result of clicking an element or submitting a form.</p> <p>If a page takes a long time to load as a result of downloading assets (e.g., images, css, js) that aren't important to the automation, you can change from the default parameter of normal to eager or none to speed up the session. This value applies to the entire session, so make sure that your waiting strategy is sufficient to minimize flakiness.</p>	Strategy	Ready State	Purpose	normal	complete	Used by default, waits for all resources to download	eager	interactive	DOM access is ready, but other resources like images may still be loading	none	Any	Does not block WebDriver at all										
Strategy	Ready State	Purpose																					
normal	complete	Used by default, waits for all resources to download																					
eager	interactive	DOM access is ready, but other resources like images may still be loading																					
none	Any	Does not block WebDriver at all																					
<code>platformName</code>	<p>This identifies the operating system at the remote-end, fetching the <code>platformName</code> returns the OS name.</p> <p>In cloud-based providers, setting <code>platformName</code> sets the OS at the remote-end.</p> <pre>ChromeOptions chromeOptions = new ChromeOptions(); String platform = "OS X 10.6"; chromeOptions.setPlatformName(platform);</pre>																						
<code>acceptInsecureCerts</code>	<p>This capability checks whether an expired (or) invalid TLS Certificate is used while navigating during a session.</p> <p>If the capability is set to false, an insecure certificate error will be returned as navigation encounters any domain certificate problems. If set to true, invalid certificate will be trusted by the browser.</p>																						

	<p>browser.</p> <p>All self-signed certificates will be trusted by this capability by default. Once set, acceptInsecureCerts capability will have an effect for the entire session.</p> <pre>ChromeOptions chromeOptions = new ChromeOptions(); chromeOptions.setAcceptInsecureCerts(true);</pre>						
Timeouts	<p>A WebDriver session is imposed with a certain session timeout interval, during which the user can control the behaviour of executing scripts or retrieving information from the browser.</p> <p>Each session timeout is configured with combination of different timeouts as described below:</p> <table border="1"> <tr> <td>Script Timeout</td><td>Specifies when to interrupt an executing script in a current browsing context. The default timeout 30,000 is imposed when a new session is created by WebDriver.</td></tr> <tr> <td>Page Load Timeout</td><td>Specifies the time interval in which web page needs to be loaded in a current browsing context. The default timeout 300,000 is imposed when a new session is created by WebDriver. If page load limits a given/default time frame, the script will be stopped by TimeoutException.</td></tr> <tr> <td>Implicit Wait Timeout</td><td>This specifies the time to wait for the implicit element location strategy when locating elements. The default timeout 0 is imposed when a new session is created by WebDriver.</td></tr> </table>	Script Timeout	Specifies when to interrupt an executing script in a current browsing context. The default timeout 30,000 is imposed when a new session is created by WebDriver.	Page Load Timeout	Specifies the time interval in which web page needs to be loaded in a current browsing context. The default timeout 300,000 is imposed when a new session is created by WebDriver. If page load limits a given/default time frame, the script will be stopped by TimeoutException.	Implicit Wait Timeout	This specifies the time to wait for the implicit element location strategy when locating elements. The default timeout 0 is imposed when a new session is created by WebDriver.
Script Timeout	Specifies when to interrupt an executing script in a current browsing context. The default timeout 30,000 is imposed when a new session is created by WebDriver.						
Page Load Timeout	Specifies the time interval in which web page needs to be loaded in a current browsing context. The default timeout 300,000 is imposed when a new session is created by WebDriver. If page load limits a given/default time frame, the script will be stopped by TimeoutException.						
Implicit Wait Timeout	This specifies the time to wait for the implicit element location strategy when locating elements. The default timeout 0 is imposed when a new session is created by WebDriver.						
unhandledPromptBehavior	<p>Specifies the state of current session's user prompt handler. Defaults to dismiss and notify state</p> <ul style="list-style-type: none"> • dismiss • accept • dismiss and notify • accept and notify • ignore <pre>ChromeOptions chromeOptions = new ChromeOptions(); chromeOptions.setUnhandledPromptBehaviour(UnexpectedAlertBehaviour.DISMISS_AND_NOTIFY);</pre>						
proxy	<pre>Proxy proxy = new Proxy(); proxy.setHttpProxy("<HOST:PORT>"); ChromeOptions options = new ChromeOptions(); options.setCapability("proxy", proxy);</pre>						
Common Args	<pre>options.addArguments("--start-maximized");</pre> <p>Commonly used args include --start-maximized, --headless=new and --user-data-dir=... <code>options.addArguments("--blink-settings=imagesEnabled=false");</code></p>						
ChromeOptions.addArgument	<ul style="list-style-type: none"> • start-maximized: Opens Chrome in maximize mode: <code>options.addArguments("--start-maximized");</code> • incognito: Opens Chrome in incognito mode: <code>options.addArguments("--incognito");</code> • headless: Opens Chrome in headless mode: <code>options.addArguments("--headless");</code> • disable-extensions: Disables existing extensions on Chrome browser • disable-popup-blocking: Disables pop-ups displayed on Chrome browser • make-default-browser: Makes Chrome default browser • version: Prints chrome browser version • disable-infobars: Prevents Chrome from displaying the notification 'Chrome is being controlled by automated software' • Accept TLS: <code>options.setAcceptInsecureCerts(true);</code> • Change User Agent: "user-agent=Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/91.0.4472.124 Safari/537.36" 						
Add Chrome Extension	<pre>ChromeOptions options = getDefaultChromeOptions(); Path path = Paths.get("src/test/resources/extensions/webextensions-selenium-example.crx"); File extensionFilePath = new File(path.toUri()); options.addExtensions(extensionFilePath);</pre>						
Set Window Size	<pre>Dimension dimension = new Dimension(1024, 768); driver.manage().window().setSize(dimension);</pre>						

Right-click using WebDriver	Actions act = new Actions(driver); act.contextClick().perform();
Drag and drop using WebDriver	Actions act = new Actions(driver); WebElement source = driver.findElement(By.xpath(" -----")); // source element WebElement target = driver.findElement(By.xpath(" -----")); // target element act.dragAndDrop(source, target).perform();
Verify checkbox/radio state	boolean isChecked = element.isSelected();
Simulate browser back and forward	driver.navigate().back(); driver.navigate().forward(); driver.navigate().refresh();
Set Dimension Of Browser	Dimension d = new Dimension(420,600); driver.manage().window().setSize(d);
Double Click	Actions action = new Actions(driver); action.doubleClick(ele); action.perform();
Sending Keys to Element	Keyboard Keys with WebDriver: driver.findElement(By...).sendKeys(Keys.TAB);
isDisplayed()	to check visibility (only for hidden items in page source). To check if an element is present or not, use size() not isDisplayed(). isSelected(): Checks if the element is selected (e.g., checkboxes). isDisplayed(): Verifies if the element is present in the DOM and displayed on the web page. isEnabled(): Checks if the element is currently enabled for user interaction. Mostly textbox might be disabled. This function checks for disabled attribute to be true.
In headless mode better to manually scroll to element else test might fail in some cases.	JavascriptExecutor js = (JavascriptExecutor) driver; js.executeScript("window.scrollBy(0,500)");
Scroll Inside Table	js.executeScript("document.querySelector('.tableFixHead').scrollTop=5000");(scrollLeft is also a method)
Block PopUps	ChromeOptions options = new ChromeOptions(); options.setExperimentalOption("excludeSwitches", Arrays.asList("disable-popup-blocking"));
Taking Screenshot	File src = ((TakesScreenshot) driver).getScreenshotAs(OutputType.FILE); // Copy the screenshot to a destination file FileUtils.copyFile(src, new File("C:\\\\Users\\\\singh\\\\Desktop\\\\screenshot.png"));
Carry Xpath	public void selectValuefromDropdown(String dropdownLocator, String optionsLocator, String option) { clickElement (dropdownLocator, "Dropdown Clicked"); List<WebElement> dates = driver.findElements(By.xpath(optionsLocator)); dates.stream().filter(i > i.findElement(By.xpath("./span")).getText().equalsIgnoreCase(option)).limit(1).forEach(i->i.click()); }
Control Clicks	for (int i = 1; i < count; i++) { String tabclick = Keys.chord(Keys.CONTROL, Keys.ENTER); scope.findElements(By.tagName("a")).get(i).sendKeys(tabclick); }
Limit Scope	WebElement footer= driver.findElement.... System.out.println(Footer.findElements(By.tagName("a")).size()); Using String.format: String xpath = String.format("//button[@id='%s']", buttonId);
Broken Link Validator	public static void verifyLink(String url) { try { URL link = new URL(url); HttpURLConnection httpURLConnection = (HttpURLConnection) link.openConnection(); httpURLConnection.setConnectTimeout(3000); // Set connection timeout to 3 seconds httpURLConnection.connect(); if (httpURLConnection.getResponseCode() == 200) { System.out.println(url + " - " + httpURLConnection.getResponseMessage()); } else { System.out.println(url + " - " + httpURLConnection.getResponseMessage() + " - " + "is a broken link"); } } catch (Exception e) { System.out.println(url + " - " + "is a broken link"); } }
Page Factory	Structure: Similar to POM, but with a focus on initializing web elements at the time of page class creation. PageFactory is used to Initialize Elements of a Page class without having to use 'FindElement' or 'FindElements'. Initialization: Elements are initialized using the PageFactory.initElements() method, which is simpler and reduces the boilerplate code.

```

@FindBy(id = "username")
private WebElement usernameField;

public LoginPage(WebDriver driver) {
    PageFactory.initElements(driver, this);
}

public void enterUsername(String username) {
    usernameField.sendKeys(username);
}

```

Cannot be used when we are limiting the scope of driver or using iteration in xpath.

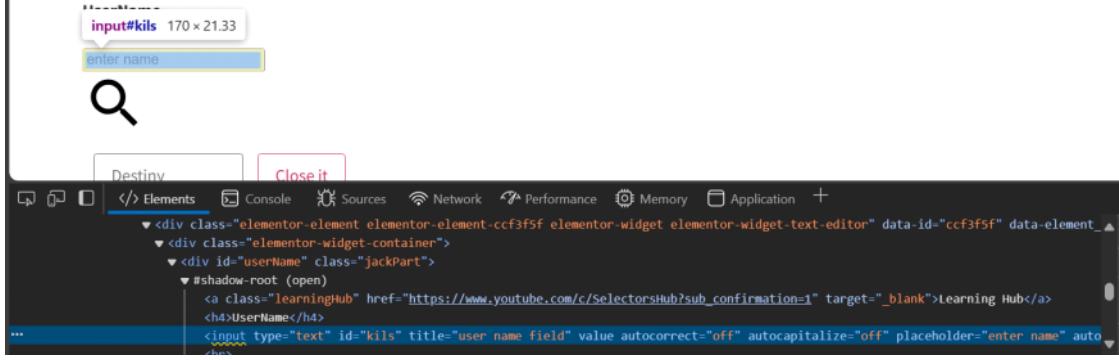
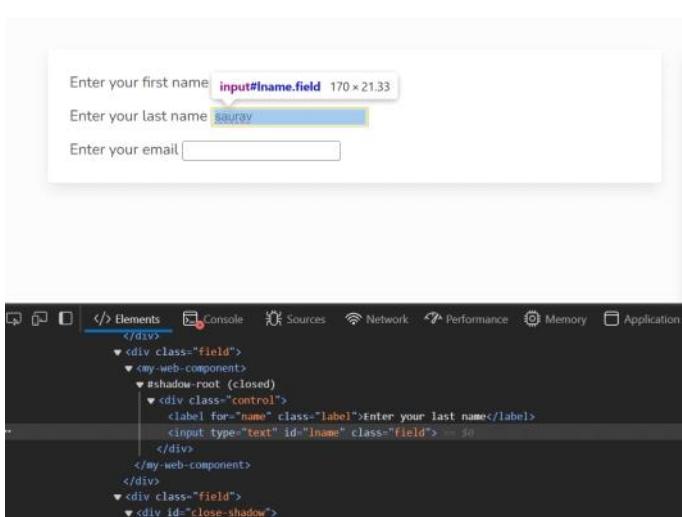
POM: Requires more code and manual element initialization, but allows greater flexibility.

Page Factory: Simplifies element initialization with PageFactory.initElements(), making it quicker to implement.

@FindBy	<p>This annotation locates the web element by using the AND condition on the search criteria</p> <pre> @FindBy(class="custom-control-check-box", id="game-chk-box") </pre> <p>WebElement chkBox;</p>
@FindAll	<p>Contrary to @FindBy, it uses an OR conditional relationship between the multiple @FindBy</p> <pre> @FindAll(@FindBy(id="btn", //doesn't match name="sbmtBtn", //Matches class="btn-primary") //doesn't match) </pre>
@CacheLookUp	<p>using @CacheLookUp, we can store the web elements in cache memory right after reading for the first time. It fastens our execution and the code, need not look up for the element on the web page and directly references it from memory.</p> <pre> @CacheLookUp @FindBy(class="custom-control-check-box", id="game-chk-box") </pre> <p>WebElement chkBox;</p>
lazy load concept using the AjaxElementLocatorFactory	<p>It can be used when your application uses Ajax elements. Additionally, you can use it while trying to find an element to perform an operation and pass a timeout value, until which the driver would wait before throwing an exception. In other words, it is a variant of implicit wait using the class AjaxElementLocatorFactory.</p> <pre> PageFactory.initElements(new AjaxElementLocatorFactory(driver, 20), this); </pre>

Design Patterns	a. Page Object Model (POM)						
	<table border="1" data-bbox="445 1666 1556 1801"> <tr> <td data-bbox="445 1666 588 1711">Purpose</td><td data-bbox="588 1666 1556 1711">promotes a clean separation between the test code and the UI structure of the application.</td></tr> <tr> <td data-bbox="445 1711 588 1801">How it works</td><td data-bbox="588 1711 1556 1801">Each page of the application is represented as a class, and all actions (methods) and elements (web elements) on the page are encapsulated within this class.</td></tr> </table>	Purpose	promotes a clean separation between the test code and the UI structure of the application.	How it works	Each page of the application is represented as a class, and all actions (methods) and elements (web elements) on the page are encapsulated within this class.		
Purpose	promotes a clean separation between the test code and the UI structure of the application.						
How it works	Each page of the application is represented as a class, and all actions (methods) and elements (web elements) on the page are encapsulated within this class.						
	b. Factory Pattern(PageObjectManager)						
	<table border="1" data-bbox="445 1830 1556 2127"> <tr> <td data-bbox="445 1830 588 1875">Purpose</td><td data-bbox="588 1830 1556 1875">To create objects without exposing the instantiation logic, allowing dynamic object creation at runtime</td></tr> <tr> <td data-bbox="445 1875 588 1943">Use Case in Selenium</td><td data-bbox="588 1875 1556 1943">A factory can be used to create WebDriver objects based on different browsers (Chrome, Firefox, etc.).</td></tr> <tr> <td data-bbox="445 1943 588 2127">Code</td><td data-bbox="588 1943 1556 2127"> <pre> public class WebDriverFactory { public static WebDriver getDriver(String browserType) { if (browserType.equalsIgnoreCase("chrome")) { return new ChromeDriver(); } else if (browserType.equalsIgnoreCase("firefox")) { </pre> </td></tr> </table>	Purpose	To create objects without exposing the instantiation logic, allowing dynamic object creation at runtime	Use Case in Selenium	A factory can be used to create WebDriver objects based on different browsers (Chrome, Firefox, etc.).	Code	<pre> public class WebDriverFactory { public static WebDriver getDriver(String browserType) { if (browserType.equalsIgnoreCase("chrome")) { return new ChromeDriver(); } else if (browserType.equalsIgnoreCase("firefox")) { </pre>
Purpose	To create objects without exposing the instantiation logic, allowing dynamic object creation at runtime						
Use Case in Selenium	A factory can be used to create WebDriver objects based on different browsers (Chrome, Firefox, etc.).						
Code	<pre> public class WebDriverFactory { public static WebDriver getDriver(String browserType) { if (browserType.equalsIgnoreCase("chrome")) { return new ChromeDriver(); } else if (browserType.equalsIgnoreCase("firefox")) { </pre>						

	<pre> return new FirefoxDriver(); } return null; } </pre>
Database	<pre> import java.sql.Connection; import java.sql.DriverManager; import java.sql.ResultSet; import java.sql.Statement; public class DatabaseTest { public static void main(String[] args) { // Database connection details String url = "jdbc:mysql://localhost:3306/yourDatabase"; String user = "yourUsername"; String password = "yourPassword"; // Connect to the database try (Connection conn = DriverManager.getConnection(url, user, password)) { // Create a statement Statement stmt = conn.createStatement(); // Execute a query ResultSet rs = stmt.executeQuery("SELECT * FROM yourTable"); // Process the results while (rs.next()) { System.out.println(rs.getString("column1") + " " + rs.getString("column2")); } } catch (Exception e) { e.printStackTrace(); } } } Using Assertj DB public class DatabaseTest { public static void main(String[] args) { // Database connection details String url = "jdbc:mysql://localhost:3306/yourDatabase"; String user = "yourUsername"; String password = "yourPassword"; // Try-with-resources ensures automatic resource cleanup try (Connection conn = DriverManager.getConnection(url, user, password)) { // Create an AssertJ-DB Table object Table table = new Table(conn, "yourTable"); // Assert that the table contains data and verify column values Assertions.assertThat(table) .row(0) // Check first row .value("column1").isNotNull() // Ensure column1 is not null .value("column2").isEqualTo("ExpectedValue"); // Check expected value System.out.println("Database assertion passed successfully!"); } catch (Exception e) { e.printStackTrace(); } } } </pre>
Shadow Dom(JS Executor)	<p>Shadow DOM is a web platform feature that encapsulates a part of the DOM and hides its internal structure from the main document. This can make it challenging to locate elements using traditional Selenium locators like XPath or CSS selectors.</p> <pre> // Access the shadow root JavascriptExecutor jsExecutor = (JavascriptExecutor) driver; WebElement shadowHost = driver.findElement(By.cssSelector("#shadow-host")); WebElement shadowRootElement = (WebElement) jsExecutor.executeScript("return arguments[0].shadowRoot", shadowHost); // Locate an element within the shadow root WebElement inputElement = (WebElement) jsExecutor.executeScript("return arguments[0].querySelector('input#shadow-input')", shadowRootElement) </pre>

	<pre>); inputElement.sendKeys("Hello, Shadow DOM!");</pre>
Shadow Dom(Selenium 4)	<p>Components:</p> <ul style="list-style-type: none"> Shadow Host: This is the regular DOM element to which a shadow DOM is attached. Shadow Tree: The DOM tree inside the shadow DOM. It is the root for the encapsulated DOM and CSS. Shadow Boundary: The boundary between the shadow DOM and the light DOM (the regular DOM outside the shadow DOM). Shadow Root: The root node of the shadow tree. It can be thought of as the entry point to the shadow DOM. <pre>WebElement shadowHost = driver.findElement(By.id("userName")); SearchContext shadowRoot = shadowHost.getShadowRoot(); WebElement usernameField = shadowRoot.findElement(By.cssSelector("#kils")); usernameField.sendKeys("your_username"); //document.querySelector("#userName").shadowRoot.querySelector("#app2").shadowRoot.querySelector("#pizza") WebElement shadowHost2 = shadowRoot.findElement(By.cssSelector("#app2")); SearchContext shadowRoot2 = shadowHost2.getShadowRoot(); WebElement pizzaText = shadowRoot2.findElement(By.cssSelector("#pizza")); pizzaText.sendKeys("Dominos");</pre> <p>Shadow-root open can only be identified by selenium</p> 
Closed Shadow Dom	 <pre>js.executeScript("document.querySelector(\"my-web-component\").myRoot.querySelector('#lname').value='saurav'");</pre>
Quit Vs Close	<p>Close: Closes Active Window, session id will be maintained</p> <p>Quit: Closes all window, session id=null</p>
Methods in WebDriver interface	<p>Some common methods include get, quit, close, getWindowHandle, getTitle, etc.</p>
Scroll down to a specific element	<pre>JavascriptExecutor js = (JavascriptExecutor) driver; js.executeScript("arguments[0].scrollIntoView(true);", element); Actions actions = new Actions(driver); actions.moveToElement(element).build().perform();</pre>
How can you input text into a field without using sendKeys()	<pre>JavascriptExecutor js = (JavascriptExecutor) driver; js.executeScript("document.getElementById('elementId').value='Your Text';");</pre>
Get vs navigate	<p>get(): Directly loads a new URL into the browser.</p> <p>navigate(): Provides more control over navigation, including:</p>

	<p>forward(): Navigates forward in the browser history. back(): Navigates backward in the browser history. refresh(): Reloads the current page.</p> <p>driver.get() : It's used to go to the particular website , But it doesn't maintain the browser History and cookies so , we can't use forward and backward button , if we click on that , page will not get schedule</p> <p>driver.navigate() : it's used to go to the particular website , but it maintains the browser history and cookies, so we can use forward and backward button to navigate between the pages during the coding of Testcase</p>
Tooltips in Selenium	String tooltipText = element.getAttribute("title");
Default Timeout	Implicit Wait: 0 seconds (fails immediately if element not found) Page Load Timeout: 300 seconds (5 minutes) Async Script Timeout: 30 seconds
SHIFT+ENTER	String press = Keys.chord(Keys.SHIFT,Keys.ENTER); webElement.sendKeys(press);
getCssValue	String color = element.getCssValue("color"); Assert.assertEquals(color, "rgba(0, 0, 0, 1)");
Streams	<pre>List<String> originalList = elementsList.stream().map(s -> s.getText()).collect(Collectors.toList()); List<String> sortedList = originalList.stream().sorted().collect(Collectors.toList()); Assert.assertTrue(originalList.equals(sortedList)); do { List<WebElement> rows = driver.findElements(By.xpath("//tr/td[1]")); price = rows.stream().filter(s -> s.getText().contains("Rice")).map(s -> getPriceVeggie(s)) .collect(Collectors.toList()); if (price.size() < 1) { driver.findElement(By.cssSelector("[aria-label='Next']")).click(); } } while (price.size() < 1);</pre>
Browser Instantiation	<p>We cannot instantiate the WebDriver interface directly as it is not a class.</p> <p>Browser-Specific Driver Instances: Example: FirefoxDriver driver = new FirefoxDriver();</p> <p>This creates an object specific to the Firefox browser and allows invocation of only methods supported by FirefoxDriver. Scripts using this setup will run only on Firefox. For other browsers, individual driver objects must be created, e.g., ChromeDriver driver = new ChromeDriver(); or InternetExplorerDriver driver = new InternetExplorerDriver();.</p> <p>Need for Cross-Browser Testing: Using browser-specific objects limits flexibility. Cross-browser compatibility requires dynamic object referencing.</p> <p>Dynamic Polymorphism: Using WebDriver driver = new FirefoxDriver(); allows flexibility. Here, driver is a reference of WebDriver, and the object is FirefoxDriver. This setup enables calling methods defined in the WebDriver interface, regardless of the specific browser driver being used. Other browser drivers like ChromeDriver, InternetExplorerDriver, etc., also implement the WebDriver interface via RemoteWebDriver.</p> <p>Important Note: Browser-specific methods cannot be invoked across different drivers. For example, a method specific to ChromeDriver cannot be called on a FirefoxDriver instance. Doing so will lead to script failure.</p>
Do	<ul style="list-style-type: none"> Selenium should not be used to prepare a test case. All repetitive actions and preparations for a test case, should be done through other methods. For example, most web UIs have authentication (e.g. a login form). Eliminating logging in via web browser before every test will improve both the speed and stability of the test. Mock external services Avoid sharing state: Do not share test data and Create a new WebDriver instance per test
Don't	<ul style="list-style-type: none"> Captchas File downloads HTTP response codes Gmail, email and Facebook logins Test dependency Performance testing Link spidering Two Factor Authentication
Test independency	Write each test as its own unit. Write the tests in a way that will not be reliant on other tests to complete
using a fluent API	<pre>public class GoogleSearchPage extends BasePage { public GoogleSearchPage(WebDriver driver) { super(driver); // Generally do not assert within pages or components. // Effectively throws an exception if the lambda condition is not met.</pre>

	<pre> new WebDriverWait(driver, Duration.ofSeconds(3)).until(d -> d.findElement(By.id("logo"))); } public GoogleSearchPage setSearchString(String sstr) { driver.findElement(By.id("gbqfq")).sendKeys(sstr); return this; } public void clickSearchButton() { driver.findElement(By.id("gbqfb")).click(); } } </pre>
POM	<p>The public methods represent the services that the page offers Try not to expose the internals of the page Generally don't make assertions Methods return other PageObjects Need not represent an entire page Different results for the same action are modelled as different methods</p> <pre> public class LoginPage { private final WebDriver driver; public LoginPage(WebDriver driver) { this.driver = driver; // Check that we're on the right page. if (!"Login".equals(driver.getTitle())) { // Alternatively, we could navigate to the login page, perhaps logging out first throw new IllegalStateException("This is not the login page"); } } // The login page contains several HTML elements that will be represented as WebElements. // The locators for these elements should only be defined once. By usernameLocator = By.id("username"); By passwordLocator = By.id("passwd"); By loginButtonLocator = By.id("login"); // The login page allows the user to type their username into the username field public LoginPage typeUsername(String username) { // This is the only place that "knows" how to enter a username driver.findElement(usernameLocator).sendKeys(username); // Return the current page object as this action doesn't navigate to a page represented by another PageObject return this; } // The login page allows the user to type their password into the password field public LoginPage typePassword(String password) { // This is the only place that "knows" how to enter a password driver.findElement(passwordLocator).sendKeys(password); // Return the current page object as this action doesn't navigate to a page represented by another PageObject return this; } // The login page allows the user to submit the login form public HomePage submitLogin() { // This is the only place that submits the login form and expects the destination to be the home page. // A separate method should be created for the instance of clicking login whilst expecting a login failure. driver.findElement(loginButtonLocator).submit(); // Return a new page object representing the destination. Should the login page ever // go somewhere else (for example, a legal disclaimer) then changing the method signature // for this method will mean that all tests that rely on this behaviour won't compile. return new HomePage(driver); } // The login page allows the user to submit the login form knowing that an invalid username and / or password were // entered public LoginPage submitLoginExpectingFailure() { // This is the only place that submits the login form and expects the destination to be the login page due to login // failure. driver.findElement(loginButtonLocator).submit(); } } </pre>

```

        // Return a new page object representing the destination. Should the user ever be navigated to the home page after
        // submiting a login with credentials
        // expected to fail login, the script will fail when it attempts to instantiate the LoginPage PageObject.
        return new LoginPage(driver);
    }

    // Conceptually, the login page offers the user the service of being able to "log into"
    // the application using a user name and password.
    public HomePage loginAs(String username, String password) {
        // The PageObject methods that enter username, password & submit login have already defined and should not be
        repeated here.
        typeUsername(username);
        typePassword(password);
        return submitLogin();
    }
}

```

Read PDF	<pre> import org.apache.pdfbox.pdmodel.PDDocument; import org.apache.pdfbox.text.PDFTextStripper; import org.testng.annotations.Test; import java.io.File; import java.io.IOException; import java.util.List; public class PDFTest { private static List<File> getListOfFiles(String directoryName) { File directory = new File(directoryName); File[] fList = directory.listFiles(); assert fList != null; return List.of(fList); } @Test public void pdfTest() { String downloadPath = System.getProperty("user.home") + "/Downloads/"; List<File> files = getListOfFiles(downloadPath); File downloadedFile = null; for (File file : files) { if (file.getName().endsWith(".pdf")) { downloadedFile = file; break; } } if (downloadedFile != null) { try { PDDocument document = PDDocument.load(downloadedFile); PDFTextStripper stripper = new PDFTextStripper(); String pdfText = stripper.getText(document); if (pdfText.contains("EPAM")) { System.out.println("PDF downloaded and validation passed."); } else { System.out.println("PDF downloaded but validation failed. Expected text not found."); } document.close(); } catch (IOException e) { System.err.println("Error reading PDF: " + e.getMessage()); } if (downloadedFile.delete()) { System.out.println("Downloaded file deleted."); } else { System.err.println("Failed to delete downloaded file."); } } else { System.out.println("Downloaded PDF not found."); } } } </pre>
----------	--

Autowiring PageObjectManager	<p>You can automate the creation of boilerplate code for your Page Object Manager using reflection and annotations, similar to how Spring Boot handles dependency injection. Here's how you can do this:</p> <p>### Steps:</p> <ol style="list-style-type: none"> 1. Define a Marker Annotation for Pages: Create a custom annotation to mark your page classes. 2. Scan for Annotated Classes: Use reflection to find all classes annotated with this marker. 3. Instantiate Pages Dynamically: Dynamically instantiate and cache these classes as needed.
------------------------------	---

Here's an updated approach to your `PageObjectManager`:

```
---
```

1. Create a Marker Annotation

```
```java
package annotations;

import java.lang.annotation.ElementType;
import java.lang.annotation.Retention;
import java.lang.annotation.RetentionPolicy;
import java.lang.annotation.Target;

@Target(ElementType.TYPE)
@Retention(RetentionPolicy.RUNTIME)
public @interface Page {
}
````
```

```
---
```

2. Annotate Your Page Classes

```
```java
package pages;

import annotations.Page;
import org.openqa.selenium.WebDriver;

@FindBy
public class HomePage {
 private WebDriver driver;

 public HomePage(WebDriver driver) {
 this.driver = driver;
 }
}
````
```

Repeat this for other page classes, e.g., `LoginSignupPage`, `SignUpPage`, etc.

```
---
```

3. Update `PageObjectManager`

```
```java
package pages;

import annotations.Page;
import org.openqa.selenium.WebDriver;

import java.util.HashMap;
import java.util.Map;

public class PageObjectManager {
 private final WebDriver driver;
 private final Map<Class<?>, Object> pageCache = new HashMap<>();

 public PageObjectManager(WebDriver driver) {
 this.driver = driver;
 }

 @SuppressWarnings("unchecked")
 public <T> T getPage(Class<T> pageClass) {
 return (T) pageCache.computeIfAbsent(pageClass, clazz -> {
 if (!clazz.isAnnotationPresent(Page.class)) {
 throw new IllegalArgumentException("Class " + clazz.getName() + " is not annotated with @Page");
 }
 try {
 return clazz.getDeclaredConstructor(WebDriver.class).newInstance(driver);
 } catch (Exception e) {
 throw new RuntimeException("Failed to create instance of " + clazz.getName(), e);
 }
 });
 }
}
````
```

```

    });
}
```

4. Use the `PageObjectManager`

```java
PageObjectManager manager = new PageObjectManager(driver);

// Access pages dynamically
HomePage homepage = manager.getPage(HomePage.class);
LoginSignupPage loginSignupPage = manager.getPage(LoginSignupPage.class);
```

Benefits:

1. **Dynamic Instantiation**: No need to manually define getter methods for each page.

2. **Scalable**: Adding a new page only requires annotating it with `@Page`.

3. **Cleaner Code**: Reduces boilerplate and centralizes instantiation logic.

4. **Easy to Extend**: You can add additional features like lazy initialization or logging.

```

This design mimics Spring Boot's autowiring feature, providing a clean, automated approach to manage your page objects.

|                                           |                                                                                                  |                                                                                                                                                                                                                                                                                                                                                                                 |
|-------------------------------------------|--------------------------------------------------------------------------------------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| disable images in selenium                | Options.addArguments("profile.managed_default_content_settings.images", 2); // 2 disables images |                                                                                                                                                                                                                                                                                                                                                                                 |
| Environment Variables via Properties File | Create global.properties                                                                         | # Active environment<br>env=qa<br><br># QA environment<br>qa.url=https://rahulshettyacademy.com/seleniumPractise<br>qa.browser=edge<br><br># UAT environment<br>uat.url=https://rahulshettyacademy.com/seleniumPractise<br>uat.browser=chrome                                                                                                                                   |
|                                           | In Test Base call variables per environment                                                      | String env = System.getProperty("env", "qa"); // Default to 'qa' if not provided<br>Properties prop = new Properties();<br>String propertiesFile = "src/test/resources/global.properties";<br>FileInputStream fis = new FileInputStream(propertiesFile);<br>prop.load(fis);<br>String url = prop.getProperty(env+".url");<br>String browser = prop.getProperty(env+".browser"); |

|                                            |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                              |
|--------------------------------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| NETX360                                    | <pre> public boolean waitForTheElementToAppear(String locator, int time, WebDriver driver) throws InterruptedException {     boolean isElementFound = false;     for (int i = 0; i &lt; time; i++) {         if (isElementPresent(locator, driver)) {             isElementFound = true;             break;         }         Thread.sleep(1000);     }     return isElementFound; }  public boolean isElementPresent(String locator, WebDriver driver) {     boolean isElementFound = false;     try {         waitForTheElementToAppear(locator, 10, driver);         if (driver.findElements(By.xpath(locator)).size() &gt; 0) {             isElementFound = true;         }     } catch (Exception e) {         e.printStackTrace();     }     return isElementFound; }  public boolean isElementDisplayed(String locator, WebDriver driver) {     boolean isElementFound = false;     try {         waitForTheElementToAppear(locator, 10, driver);         if (driver.findElements(By.xpath(locator)).size() &gt; 0) {             isElementFound = driver.findElement(By.xpath(locator)).isDisplayed();         }     } catch (Exception e) {         e.printStackTrace();     }     return isElementFound; } </pre> |
| isElementPresent                           | <pre> public boolean isElementPresent(String locator, WebDriver driver) {     return !driver.findElements(By.xpath(locator)).isEmpty(); }  ----- public boolean isElementPresent(By locator) {     try {         WebDriverWait wait = new WebDriverWait(driver, Duration.ofMillis(1500)); // Wait for 1.5s         wait.until(ExpectedConditions.presenceOfElementLocated(locator));         return true;     } catch (TimeoutException e) {         return false; // Element not found within 1.5s     } } </pre>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                           |
| Identifying Xpath of disappearing elements | <ul style="list-style-type: none"> <li>Inspect the element which will display the disappearing elements</li> <li>Right click on the element tag and select 'Break on' and select 'attribute modifications'. Once you selected them, you will be able to see it under DOM Breakpoints section.</li> <li>Now click on the element which will display the disappearing elements. Debugger mode will open.</li> <li>Now press F8, until you are able to inspect the disappearing element.</li> </ul> <p>-----</p> <ul style="list-style-type: none"> <li>Open Dev tools, click anywhere inside elements tab in dev tools</li> <li>Ctrl+Shift+P</li> <li>In run command type <b>emulate a focussed page</b></li> <li>Now elements wont disappear</li> </ul>                                                                                                                                                                                                                                                                                                                                                                                                                                                                       |

|                                                            |                                                                                                                                                                                                                                                                          |
|------------------------------------------------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Xpath when we want to iterate by updating numbers in xpath | <pre>String ele = "/div[@row-index='%s']"; public String getElementText(String ele, int num) {     WebElement element = driver.findElement(By.xpath(String.format(ele, String.valueOf(num))));     String actualtext = element.getText();     return actualtext; }</pre> |
|------------------------------------------------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|

|                            |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                           |
|----------------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Validate is field disabled | <pre>public void isFieldDisabled(String fieldName) {     boolean isElementDisabled;     if(fieldName.equals("Call / Put"))         isElementDisabled = commonActions             .getAttributeValue(By.xpath("//label[contains(text(),'"+fieldName+"'))][2]/parent::div/mat-form-field"),             "class")             .trim().contains("mat-form-field-disabled");     else         isElementDisabled = commonActions             .getAttributeValue(By.xpath("//label[contains(text(),'"+fieldName+"'))/parent::div/mat-form-field"),             "class")             .trim().contains("mat-form-field-disabled");      if (isElementDisabled) {         Screenshots.addStepInReport(true, fieldName +" field is displayed and disabled");     } else {         System.out.println(fieldName +" field is enabled");         AssertionLibrary.assertTrue(false, fieldName +" field is enabled", Screenshot.REQUIRED);     } }</pre> |
|----------------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|

|                     |                                                                                                                                                                                                                                                                                                                                        |
|---------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Get Attribute Value | <pre>public String getAttributeValue(String locator, String attributeName) {     String text = null;     try {         WebElement localWebElement = getElement(By.xpath(locator));         text = localWebElement.getAttribute(attributeName);     } catch (Exception e) {         e.printStackTrace();     }     return text; }</pre> |
|---------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|

# Basics

20 May 2021 10:52

REST (Representational State Transfer) is an architectural style for designing networked applications. RESTful APIs adhere to the principles of REST architecture, providing a standardized way for systems to communicate over the internet.

**Resources:** In REST, everything is considered a resource, which can be accessed using a unique identifier, typically a URL (Uniform Resource Locator).  
**HTTP Methods:** RESTful APIs use standard HTTP methods such as GET, POST, PUT, and DELETE to perform CRUD (Create, Read, Update, Delete) operations on resources.

**Representations:** Resources are represented in a format such as JSON (JavaScript Object Notation) or XML (eXtensible Markup Language), allowing for easy exchange of data between client and server.

| Constraint                        | Description                                                                    | Implication                                                                                             |
|-----------------------------------|--------------------------------------------------------------------------------|---------------------------------------------------------------------------------------------------------|
| <b>Statelessness</b>              | Each request must contain all the information needed by the server.            | Simplifies server design and scalability; no session state is stored between requests.                  |
| <b>Client-Server Architecture</b> | Separation between client and server.                                          | Allows independent evolution of client and server; defines clear responsibilities.                      |
| <b>Uniform Interface</b>          | Standardized way of interacting with resources through fixed methods and URIs. | Simplifies architecture, improves usability, and ensures consistency.                                   |
| <b>Resource-Based</b>             | API is designed around resources identified by URIs.                           | Resources are accessed and manipulated using standard HTTP methods, improving clarity and organization. |
| <b>Layered System</b>             | Architecture can be composed of multiple layers, each with specific functions. | Enhances scalability and flexibility; changes can be made independently at different layers.            |
| <b>Cacheability</b>               | Responses should indicate if they are cacheable.                               | Improves performance by reducing redundant requests; proper caching is crucial for efficiency.          |

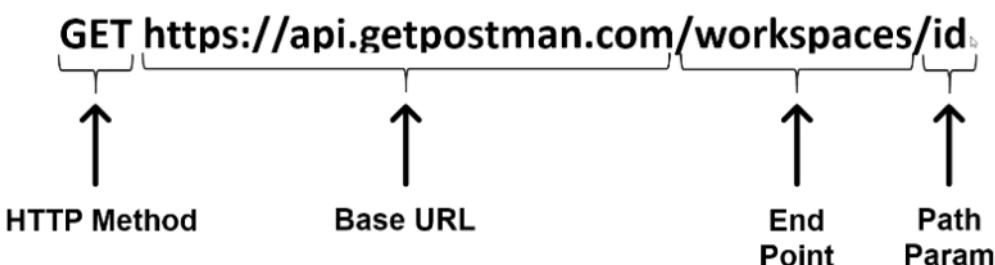
RESTful APIs provides a powerful and flexible way to build networked applications, enabling interoperability, scalability, and performance.

HTTP is the underlying protocol used by RESTful APIs, and REST builds on top of HTTP to provide a set of principles for designing scalable and interoperable web services. Rest can also use FTP or SMTP.

**Safe:** Doesn't change resources on server

**Idempotent:** performing the same operation multiple times produces the same result as performing it once

| Method  | Description                                                                         | Request Body | Response Body | Safe | Idempotent | Cacheable     |
|---------|-------------------------------------------------------------------------------------|--------------|---------------|------|------------|---------------|
| GET     | Transfer a current representation of the target resource                            | No           | Yes           | Yes  | Yes        | Yes           |
| HEAD    | Same as GET, but only transfer the status line and header section                   | No           | No            | Yes  | Yes        | Yes           |
| POST    | Perform resource-specific processing on the request payload                         | Yes          | Yes           | No   | No         | In some cases |
| PUT     | Replace all current representations of the target resource with the request payload | Yes          | No            | No   | Yes        | No            |
| DELETE  | Remove all current representations of the target resource                           | Optional     | Optional      | No   | Yes        | No            |
| CONNECT | Establish a tunnel to the server identified by the target resource                  | No           | Yes           | No   | No         | No            |
| OPTIONS | Describe the communication options for the target resource                          | No           | Yes           | Yes  | Yes        | No            |
| TRACE   | Perform a message loop-back test along the path to the target resource              | No           | No            | Yes  | Yes        | No            |
| PATCH   | Perform partial modification of the target resource                                 | Yes          | Yes           | No   | No         | No            |



Endpoint: The address where the API is hosted on the server.

|                  |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                               |
|------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Path Parameters  | <p>Used to specify values within the URL's path itself. They are often used to identify specific resources.</p> <p>Syntax: /resource/{id}</p> <pre>given()     .pathParam("id", 123) .when()     .get("/users/{id}") .then()     .statusCode(200);  Map&lt;String, Object&gt; pathParams = new HashMap&lt;&gt;(); pathParams.put("category", "books"); pathParams.put("productId", 456);  given()     .pathParams(pathParams) .when()     .get("/products/{category}/{productId}") .then()     .statusCode(200);</pre>                                                                                                                                                                                                                                                                                                        |
| Query Parameters | <p>Used to sort/filter resources, identified with "?". Query parameters are separated by an ampersand (&amp;) if there are multiple parameters.</p> <ul style="list-style-type: none"> <li>- <a href="https://amazon.com/orders?sort_by=2/20/2020">https://amazon.com/orders?sort_by=2/20/2020</a></li> </ul> <pre>given()     .queryParam("page", 2)     .queryParam("pageSize", 10) .when()     .get("/users") .then()     .statusCode(200);  Map&lt;String, Object&gt; queryParams = new HashMap&lt;&gt;(); queryParams.put("sort", "name"); queryParams.put("order", "asc");  given()     .queryParams(queryParams) .when()     .get("/products") .then()     .statusCode(200);  given()     .queryParam("filter", Arrays.asList("active", "verified")) .when()     .get("/accounts") .then()     .statusCode(200);</pre> |
| Form Parameters  | <p>Used to send data in the request body as application/x-www-form-urlencoded format. Commonly used for submitting forms.</p> <pre>given()     .formParam("username", "testuser")     .formParam("password", "password123") .when()     .post("/login") .then()     .statusCode(200);  Map&lt;String, Object&gt; formParams = new HashMap&lt;&gt;(); formParams.put("name", "New Product"); formParams.put("price", 99.99);  given()     .formParams(formParams) .when()     .post("/products") .then()     .statusCode(201);</pre>                                                                                                                                                                                                                                                                                           |

| Category             | Status Code                                                             |                                                                                       |
|----------------------|-------------------------------------------------------------------------|---------------------------------------------------------------------------------------|
| <b>Informational</b> | 100-199:Indicates the request has been received and is being processed. |                                                                                       |
| <b>Successful</b>    | 200-299                                                                 | Indicates that the request was successfully received, understood, and accepted.       |
|                      | 200                                                                     | OK: The request was successful.                                                       |
|                      | 201                                                                     | Created: The resource was successfully created.                                       |
|                      | 204                                                                     | No Content: The request was processed, but there is no content to send back.          |
| <b>Redirection</b>   | 300-399                                                                 | Indicates that further action is needed to complete the request.                      |
|                      | 301                                                                     | Moved Permanently: The resource has been moved to a new location.                     |
| <b>Client Error</b>  | 400-499                                                                 | Indicates there was an error with the client's request.                               |
|                      | 400                                                                     | Bad Request: The request was malformed or missing necessary information.              |
|                      | 401                                                                     | Unauthorized: Authentication failed (e.g., invalid or expired token).                 |
|                      | 403                                                                     | Forbidden: The client does not have permission to access the resource.                |
|                      | 404                                                                     | Not Found: The requested resource could not be found.                                 |
| <b>Server Error</b>  | 500-599                                                                 | Indicates the server failed to fulfill a valid request.                               |
|                      | 500                                                                     | Internal Server Error: A generic error indicating something went wrong on the server. |
|                      | 502                                                                     | Bad Gateway: Received an invalid response from an upstream server.                    |
|                      | 503                                                                     | Service Unavailable: The server is temporarily unable to handle the request.          |
|                      | 504                                                                     | Gateway Timeout: The gateway server did not receive a timely response.                |

# Hamcrest

16 March 2024 10:08

Hamcrest is a framework for writing matcher objects in Java. It provides a fluent interface for creating assertions in a more readable and expressive way.

- `equalTo`: Checks if the actual value is equal to the expected value.
- `containsString`: Checks if the actual value contains the expected substring.
- `hasItems`: Verifies that the actual list or array contains the specified items.
- `not`: Used to negate other matchers.
- `greaterThan`, `lessThan`, `greaterThanOrEqualTo`, `lessThanOrEqualTo`: Used to compare numbers.

```
body("userId", equalTo(1)).
body("id", greaterThan(0)).
body("title", containsString("sunt")).
body("body", notNullValue());
```

Anonymous JSON root validation: [1, 2, 3]:body("\$", hasItems(1, 2, 3)); // An empty string "" would work as well.

# Passing Headers

16 March 2024 12:32

| Method                                          | Example                                                                                                                                                                                                                                                                                                                                                                                                        |
|-------------------------------------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>Single Header (using header() method)</b>    | <code>header("header", "value1")</code>                                                                                                                                                                                                                                                                                                                                                                        |
| <b>Multiple Headers (using header() method)</b> | <code>Header header = new Header("header", "value1");<br/>Header matchHeader = new Header("x-mock-match-request-headers", "header");<br/>.header(header).header(matchHeader)</code>                                                                                                                                                                                                                            |
| <b>Multiple Headers (using Map)</b>             | <code>HashMap&lt;String, String&gt; headers = new HashMap&lt;&gt;();<br/><br/>headers.put("header", "value2");<br/>.headers(headers)</code>                                                                                                                                                                                                                                                                    |
| <b>Multiple Headers as String Parameters</b>    | <code>Headers extractedHeaders = given().baseUri("https://8f6d7436-aba9-4c1f-bc81-fdc881a11fb1.mock.pstmn.io").headers(headers).when().get("/get").then().assertThat().statusCode(200).extract().headers();<br/>for (Header header : extractedHeaders) {<br/>    System.out.print("header name = " + header.getName() + ", ");<br/>    System.out.println("header value = " + header.getValue());<br/>}</code> |
| <b>Assert Response Headers</b>                  | <code>Map&lt;String, String&gt; expectedHeaders = new HashMap&lt;&gt;();<br/>expectedHeaders.put("Content-Type", "application/json");<br/>expectedHeaders.put("Server", "nginx");<br/>response.then().headers(expectedHeaders);<br/><br/>String contentType = response.getHeader("Content-Type");<br/>assertEquals(contentType, "application/json");</code>                                                    |
| <b>Extract Response Headers</b>                 | <code>.extract().headers()</code>                                                                                                                                                                                                                                                                                                                                                                              |

# Logging

15 August 2024 12:50

- log().all()
- log().headers()
- log().body()
- log().cookies()
- log().parameters()
- log().status()
- log().statusCode()
- log().everything()
- log().ifError()
- log().ifValidationFails()

## Logging to a File

```
@BeforeClass
public void beforeClass() throws FileNotFoundException {
 PrintStream fileOutPutStream = new PrintStream("restAssured.log");// This will overide the logs, to preserve use
new PrintStream(new FileOutputStream("restAssured.log", true))

 RequestSpecBuilder requestSpecBuilder = new RequestSpecBuilder().addFilter(new RequestLoggingFilter(fileOutPutStream)).addFilter(new
ResponseLoggingFilter(fileOutPutStream));

 requestSpecification = requestSpecBuilder.build();

 ResponseSpecBuilder responseSpecBuilder = new ResponseSpecBuilder();
 responseSpecification = responseSpecBuilder.build();
}

@Test
public void loggingFilter() {
 given(requestSpecification).baseUri("https://postman-
echo.com").log().all().when().get("/get").then().spec(responseSpecification).log().all().assertThat().statusCode(200);
}
```

**.log().all():** This logs all request details to the console (in addition to the file, due to the filters).

# Request/Response Specification/Builder

28 June 2021 10:04

|                        |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                   |
|------------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Request Specification  | <ul style="list-style-type: none"><li>Purpose: An interface that can be used to group and retrieve repetitive actions, such as setting up the base URL, headers, and HTTP verbs.</li><li>Benefits: Reduces the number of lines of code and increases maintainability by reusing common configurations.</li><li>Usage: Define common request configurations that can be applied to multiple requests.</li></ul>                                                                                                                                                                                    |
| Response Specification | <ul style="list-style-type: none"><li>Purpose: An interface that allows you to specify how the expected response must look like for a test to pass.</li><li>Benefits: Used to validate a common response or a response needed for multiple tests, ensuring consistency in response validation.</li></ul>                                                                                                                                                                                                                                                                                          |
| RequestSpecBuilder     | <ul style="list-style-type: none"><li>Purpose: Builder class for constructing `RequestSpecification` instances.</li><li>Used for: Creating and configuring complex or reusable `RequestSpecification` instances.</li><li>Method: Use `RequestSpecBuilder` to set up parameters and call `build()` to get a `RequestSpecification`.</li><li>Example:<pre>RequestSpecification requestSpec = new RequestSpecBuilder()<br/>    .setBaseUri("https://rahulshettyacademy.com")<br/>    .addQueryParam("key", "qaclik123")<br/>    .setContent-Type(ContentType.JSON)<br/>    .build();</pre></li></ul> |
| ResponseSpecBuilder    | <ul style="list-style-type: none"><li>Purpose: Builder class for constructing `ResponseSpecification` instances.</li><li>Used for: Creating and configuring complex or reusable `ResponseSpecification` instances.</li><li>Method: Use `ResponseSpecBuilder` to set up expected responses and call `build()` to get a `ResponseSpecification`.</li><li>Example:<pre>ResponseSpecification responseSpec = new ResponseSpecBuilder()<br/>    .expectStatus-Code(200)<br/>    .expectHeader("server", "Apache/2.4.52 (Ubuntu)")<br/>    .build();</pre></li></ul>                                    |

```
RestAssured.baseURI = "https://rahulshettyacademy.com";

// Define request specification
RequestSpecification requestSpec = new RequestSpecBuilder()
 .setBaseUri(RestAssured.baseURI)
 .addQueryParam("key", "qaclik123")
 .setContent-Type(ContentType.JSON)
 .build();

// Define response specification
ResponseSpecification responseSpec = new ResponseSpecBuilder()
 .expectStatus-Code(200)
 .expectHeader("server", "Apache/2.4.52 (Ubuntu)")
 .build();

// Perform POST request
String postResponse = given().spec(requestSpec)
 .body(new File("src/test/resources/AddPlace.json"))
 .when().post("maps/api/place/add/json")
 .then().spec(responseSpec)
 .body("scope", equalTo("APP"))
 .extract().response().asString();
```

# Serialization

28 June 2021 09:57

Serialization is used when you want to send a request body with data. We convert objects into a format that the API can understand, typically JSON or XML.

```
String filePath = "src/test/resources/complex.json";

String jsonString = new String(Files.readAllBytes(Paths.get(filePath)));
```

Or

```
String jsonString = Files.readString(Paths.get(filePath));
```

| Library | Dependency                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                              | Usage                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                            |      |                                                                                                                                                                                               |      |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                         |
|---------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Jackson | <dependency><br><groupId>com.fasterxml.jackson.core</groupId><br><artifactId>jackson-databind</artifactId><br><version>2.13.1</version></dependency>                                                                                                                                                                                                                                                                                                                                                                                                                    | <p><b>Explicit Jackson Usage</b></p> <pre>@Test<br/>public void orgJacksonMockSerialization() throws JsonProcessingException {<br/>    // Parse JSON String<br/>    ObjectMapper objectMapper = new ObjectMapper();<br/>    Object jsonObject = objectMapper.readValue(jsonString, Object.class);<br/><br/>    // Serialize JSON Object_Pretty<br/>    String serializedJson =<br/>objectMapper.writerWithDefaultPrettyPrinter().writeValueAsString(jsonObject);<br/>    System.out.println(serializedJson);<br/>}</pre> <p><b>@Test</b></p> <pre>public void orgJacksonPOJOSerialization() throws JsonProcessingException {<br/>    ObjectMapper objectMapper = new ObjectMapper();<br/>    String serializedJson = objectMapper.writeValueAsString(setAddPlace());<br/>    System.out.println(serializedJson);<br/>}</pre> <p>Using RestAssured to use Jackson Behind the scenes</p> <table border="1"><tr><td>JSON</td><td>String jsonBody = "{\"name\":\"Alice\",\"age\":28}";<br/>given() .contentType(MediaType.JSON) .body(jsonBody) // Serializing the JSON string .when() .post("/users") .then() .statusCode(201);</td></tr><tr><td>Pojo</td><td>public class SerializationPojo {<br/><br/>    public static class User {<br/>        private String name;<br/>        private int age;<br/><br/>        public User() {}<br/><br/>        public User(String name, int age) {<br/>            this.name = name;<br/>            this.age = age;<br/>        }<br/><br/>        public String getName() {<br/>            return name;<br/>        }<br/><br/>        public int getAge() {<br/>            return age;<br/>        }<br/>    }<br/><br/>    public static void main(String[] args) {<br/>        User user = new User("Bob", 35);<br/><br/>        given()<br/>            .contentType(MediaType.JSON)<br/>            .body(user) // Serializing the POJO<br/>        .when()<br/>            .post("/users")<br/>        .then()<br/>    }<br/>}</td></tr></table> | JSON | String jsonBody = "{\"name\":\"Alice\",\"age\":28}";<br>given() .contentType(MediaType.JSON) .body(jsonBody) // Serializing the JSON string .when() .post("/users") .then() .statusCode(201); | Pojo | public class SerializationPojo {<br><br>public static class User {<br>private String name;<br>private int age;<br><br>public User() {}<br><br>public User(String name, int age) {<br>this.name = name;<br>this.age = age;<br>}<br><br>public String getName() {<br>return name;<br>}<br><br>public int getAge() {<br>return age;<br>}<br>}<br><br>public static void main(String[] args) {<br>User user = new User("Bob", 35);<br><br>given()<br>.contentType(MediaType.JSON)<br>.body(user) // Serializing the POJO<br>.when()<br>.post("/users")<br>.then()<br>}<br>} |
| JSON    | String jsonBody = "{\"name\":\"Alice\",\"age\":28}";<br>given() .contentType(MediaType.JSON) .body(jsonBody) // Serializing the JSON string .when() .post("/users") .then() .statusCode(201);                                                                                                                                                                                                                                                                                                                                                                           |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                  |      |                                                                                                                                                                                               |      |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                         |
| Pojo    | public class SerializationPojo {<br><br>public static class User {<br>private String name;<br>private int age;<br><br>public User() {}<br><br>public User(String name, int age) {<br>this.name = name;<br>this.age = age;<br>}<br><br>public String getName() {<br>return name;<br>}<br><br>public int getAge() {<br>return age;<br>}<br>}<br><br>public static void main(String[] args) {<br>User user = new User("Bob", 35);<br><br>given()<br>.contentType(MediaType.JSON)<br>.body(user) // Serializing the POJO<br>.when()<br>.post("/users")<br>.then()<br>}<br>} |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                  |      |                                                                                                                                                                                               |      |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                         |

|          |                                                                                                                                       |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                   |
|----------|---------------------------------------------------------------------------------------------------------------------------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
|          |                                                                                                                                       | <pre>         }     } } </pre>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                    |
| org.json | <dependency>     <groupId>org.json</groupId>     <artifactId>json</artifactId>     <version>20210307</version> </dependency>          | <pre> @Test public void orgJsonMockSerialization() {     JSONObject jsonObject = new JSONObject(jsonString);     String serializedJson = jsonObject.toString(4); // Indented with 4 spaces     System.out.println(serializedJson); }  @Test public void orgJsonPOJOSerialization(){     JSONObject jsonObject = new JSONObject(setAddPlace());     System.out.println(jsonObject.toString(4)); } </pre>                                                                                                                                                                                                                           |
| Gson     | <dependency>     <groupId>com.google.code.gson</groupId>     <artifactId>gson</artifactId>     <version>2.8.7</version> </dependency> | <pre> @Test public void orgGsonPOJOSerialization(){     Gson gson = new Gson();     String serializedJson = gson.toJson(setAddPlace());     System.out.println(serializedJson);     // Pretty-print JSON     Gson gsonPretty = new GsonBuilder().setPrettyPrinting().create();     String prettyJson = gsonPretty.toJson(setAddPlace());     System.out.println(prettyJson); }  @Test public void orgGsonMockSerialization() {     Gson gson = new Gson();     Object jsonObject = gson.fromJson(jsonString, Object.class);     String serializedJson = gson.toJson(jsonObject);     System.out.println(serializedJson); } </pre> |

# Deserialization

28 June 2021 09:58

Serialization in Rest Assured context is a process of converting a Java object into Request body (Payload). Rest Assured also Supports deserialization by converting Response body back to Java object.

Advantages:

- Easy to parse and extract response (Json/XML) values if they are wrapped as Java object.
- User-friendly Methods can be created which makes code more readable.

Design Approach:

- Java object is constructed with the support of POJO classes.
- POJO classes are created based on the request/Response payload.

Additional Libraries: For JSON, you need to have either Jackson, Jackson2, Gson, or Johnzon in the classpath. For XML, you need JAXB.

```
static class Book {
 public String name;
 public String isbn;
 public String aisle;
 public String author;
}
```

|          |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                         |
|----------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Jackson  | <pre>@Test<br/>public void JacksonDeserializationExample() throws JsonProcessingException {<br/>    String jsonResponse = "{ \"name\":\"Learn Appium Automation with Java\", \"isbn\":\"bcd1\", \"aisle\": \"2926\", \"author\": \"John foer\" }";<br/><br/>    // Deserialize JSON response using Jackson<br/>    ObjectMapper objectMapper = new ObjectMapper();<br/>    Book book = objectMapper.readValue(jsonResponse, Book.class);<br/><br/>    System.out.println("Book Name: " + book.name);<br/>    System.out.println("Author: " + book.author);<br/>}<br/><br/><b>Direct Approach</b><br/>@Test<br/>public void restAssuredDeserialization() {<br/>    Book book = given()<br/>        .when()<br/>        .get("/books/123") // Replace with your actual endpoint<br/>        .then()<br/>        .contentType(MediaType.APPLICATION_JSON) // Ensure the response is JSON<br/>        .extract()<br/>        .as(Book.class);<br/><br/>    System.out.println("Book Name: " + book.name);<br/>    System.out.println("Author: " + book.author);<br/>}</pre> |
| GSON     | <pre>@Test<br/>public void GsonDeserializationExample (){<br/>    String jsonResponse = "{ \"name\":\"Learn Appium Automation with Java\", \"isbn\":\"bcd1\", \"aisle\": \"2926\", \"author\": \"John foer\" }";<br/><br/>    // Deserialize JSON response using Gson<br/>    Gson gson = new Gson();<br/>    Book book = gson.fromJson(jsonResponse, Book.class);<br/><br/>    System.out.println("Book Name: " + book.name);<br/>    System.out.println("Author: " + book.author);<br/>}</pre>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                        |
| Org.json | <pre>@Test<br/>public void OrgJsonDeserializationExample (){<br/>    String jsonResponse = "{ \"name\":\"Learn Appium Automation with Java\", \"isbn\":\"bcd1\", \"aisle\": \"2926\", \"author\": \"John foer\" }";</pre>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                               |

|          |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                           |
|----------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
|          | <pre> // Deserialize JSON response using org.json JSONObject jsonObject = new JSONObject(jsonResponse);  String name = jsonObject.getString("name"); String author = jsonObject.getString("author");  System.out.println("Book Name: " + name); System.out.println("Author: " + author); } </pre>                                                                                                                                                                                                                                         |
| JSONPath | <pre> @Test public void JsonPathDeserializationExample(){     String jsonResponse = "{ \"name\": \"Learn Appium Automation with Java\", \"isbn\": \"bcd1\", \"aisle\": \"2926\", \"author\": \"John foer\" }";      // Parse JSON using JsonPath     JsonPath jsonPath = new JsonPath(jsonResponse);      // Extract specific values     String name = jsonPath.getString("name");     String author = jsonPath.getString("author");      System.out.println("Book Name: " + name);     System.out.println("Author: " + author); } </pre> |

## Update JSON File

28 March 2023 07:56

|                       |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                              |
|-----------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| BNY                   | <pre>"addlOrdData": [     "AON",     "NOTHELD" ],\n\n// create a JSONObject object     JSONObject jsonObject = new JSONObject(new String(Files.readAllBytes(Paths.get("src/test/resources/AddBook.json"))));\n\n// add a property to the JSONObject object jsonObject.put("propertyName", "PropertyValue");\n\n// update the "addlOrddata" property with a JSONArray containing two strings JSONArray jsonArray = new JSONArray(); jsonArray.put("aon"); jsonArray.put("ng"); jsonObject.put("addlOrddata", jsonArray);\n\n\n"solicited": false,\n\n// create a JSONObject object JSONObject jsonObject = new JSONObject();\n\n// set the initial value of the "solicited" property to false jsonObject.put("solicited", false);\n\n// update the value of the "solicited" property to true jsonObject.put("solicited", true);\n\n\n"commission": {     "commType": "RTE",     "commissionAmt": 2,     "splitIP1Pct": null,     "splitIP2Pct": null,     "splitIP1": null,     "splitIP2": null },\n\n// create a JSONObject object JSONObject jsonObject = new JSONObject();\n\n// update the value of the "commission" property with a nested object JSONObject commissionObject = new JSONObject(); commissionObject.put("commttype", "rte"); commissionObject.put("commvsque", 2); jsonObject.put("commission", commissionObject);</pre> |
| UpdateJsonWithJackson | <pre>@Test public void UpdateJsonWithJackson() throws JsonProcessingException {     String json = "{\"name\": \"Learn Appium Automation with Java\", \"isbn\": \"bcd1\", \"aisle\": \"2926\", \"author\": \"John foer\" };\n\n    // Parse JSON into a Map     ObjectMapper objectMapper = new ObjectMapper();     Map jsonMap = objectMapper.readValue(json, Map.class);\n\n    // Update the name field     jsonMap.put("name", "Learn Selenium with Python");\n\n    // Serialize back to JSON     String updatedJson = objectMapper.writeValueAsString(jsonMap);     System.out.println(updatedJson); }</pre>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                            |
| UpdateJsonWithGson    | <pre>@Test public void UpdateJsonWithGson () {     String json = "{\"name\": \"Learn Appium Automation with Java\", \"isbn\": \"bcd1\", \"aisle\": \"2926\", \"author\": \"John foer\" };</pre>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                              |

|                                |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                            |
|--------------------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
|                                | <pre> // Parse JSON into a JsonObject JsonObject jsonObject = JsonParser.parseString(json).getAsJsonObject();  // Update the name field jsonObject.addProperty("name", "Learn Selenium with Python");  // Serialize back to JSON String updatedJson = jsonObject.toString(); System.out.println(updatedJson); } </pre>                                                                                                                                                                                                                                                                                                                                                                                                                     |
| UpdateJsonWithOrgJson          | <pre> @Test public void UpdateJsonWithOrgJson () {     String json = "{ \"name\": \"Learn Appium Automation with Java\", \"isbn\": \"bcd1\", \"aisle\": \"2926\", \"author\":     \"John foer\" }";      // Parse JSON into a JsonObject     JsonObject jsonObject = new JsonObject(json);      // Update the name field     jsonObject.put("name", "Learn Selenium with Python");      // Serialize back to JSON     String updatedJson = jsonObject.toString();     System.out.println(updatedJson); } </pre>                                                                                                                                                                                                                            |
| UpdateJsonWithJaywayJsonPath   | <pre> import com.jayway.jsonpath.JsonPath; import net.minidev.json.JSONObject; @Test public void UpdateJsonWithJaywayJsonPath () throws JsonProcessingException {     String json = "{ \"name\": \"Learn Appium Automation with Java\", \"isbn\": \"bcd1\", \"aisle\": \"2926\", \"author\":     \"John foer\" }";      // Parse JSON     JSONObject jsonObject = JsonPath.parse(json).json();      // Update the name field     jsonObject.put("name", "Learn Selenium with Python");      // Serialize back to JSON     String updatedJson = jsonObject.toJSONString();     System.out.println(updatedJson); } </pre>                                                                                                                    |
| UpdateJsonWithJSONObject       | <pre> import javax.json.Json; import javax.json.JsonObject; import javax.json.JsonReader; import java.io.StringReader; @Test public void UpdateJsonWithJaywayJsonPath (){     String json = "{ \"name\": \"Learn Appium Automation with Java\", \"isbn\": \"bcd1\", \"aisle\": \"2926\", \"author\":     \"John foer\" }";      // Parse JSON     JsonReader reader = Json.createReader(new StringReader(json));     JsonObject jsonObject = reader.readObject();      // Update the name field     JsonObject updatedJsonObject = Json.createObjectBuilder(jsonObject)         .add("name", "Learn Selenium with Python")         .build();      // Serialize back to JSON     System.out.println(updatedJsonObject.toString()); } </pre> |
| UpdateJsonWithSimpleJSONObject | <pre> import org.json.simple.JSONObject; import org.json.simple.parser.JSONParser; @Test public void UpdateJsonWithJSONObject(){     String json = "{ \"name\": \"Learn Appium Automation with Java\", \"isbn\": \"bcd1\", \"aisle\": \"2926\", \"author\":     \"John foer\" }";      // Parse JSON     JSONParser parser = new JSONParser();     JSONObject jsonObject = (JSONObject) parser.parse(json); </pre>                                                                                                                                                                                                                                                                                                                         |

```
// Update the name field
jsonObject.put("name", "Learn Selenium with Python");

// Serialize back to JSON
System.out.println(jsonObject.toJSONString());
}
```

# Authentication

20 July 2021 09:46

| Authentication Method            | Description                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                             | Details                                                                                                                                                                                                                                                                                                                                                                                                                                                              |
|----------------------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Basic Authentication(Not Secure) | <p>Basic access authentication is a method for an HTTP user agent (e.g., a web browser) to provide a username and password when making a request.</p> <pre data-bbox="401 377 1133 760">Response response = RestAssured     .given()     .auth()     .basic(username, password)     .when()     .get(baseUrl);  Response response = RestAssured.given()     .auth().preemptive().basic("username", "password") // Preemptive Basic authentication     .when()     .get("https://example.com/protected-resource");</pre> | <p>Basic authentication sends the username and password in the request headers. The credentials are sent in an easily decodable format (Base64 encoded), which is less secure unless the communication is encrypted (e.g., using HTTPS).</p> <p>Basic Authentication is normally done after receiving a 401 Unauthorized response, whereas Preemptive Authentication sends the credentials right away without waiting for the 401 status</p>                         |
| Digest Authentication            | <p>Digest authentication is a method used by web servers to negotiate credentials securely. It involves hashing the username and password along with other parameters, making it more secure than plain text transmission.</p> <pre data-bbox="401 877 1133 1045">Response response = RestAssured     .given()     .auth()     .digest(username, password)     .when()     .get(baseUrl);</pre>                                                                                                                         | <p>Process:</p> <ul style="list-style-type: none"> <li>- HA1 = MD5(username:realm)</li> <li>- HA2 = MD5(method)</li> <li>- Response = MD5(HA1:nonce)</li> </ul> <p>Digest authentication is a more secure alternative to basic authentication. Instead of sending the username and password directly, the client sends a hashed value of the credentials and other information (like a random challenge) that is calculated on both the client and server sides.</p> |
| Bearer Authentication            | <p>Bearer authentication is a token-based authentication mechanism where a bearer token is passed in the HTTP header to authorize requests. This token is usually obtained through OAuth or another authentication mechanism.</p> <pre data-bbox="433 1221 933 1329">Response response = RestAssured.given()     .header("Authorization", "Bearer " + bearerToken)     .when()     .get("https://example.com/protected-resource");</pre>                                                                                | <p>Process:</p> <ul style="list-style-type: none"> <li>- Obtain Bearer Token from an authentication server.</li> <li>- Include Authorization: Bearer &lt;token&gt; in the HTTP request header.</li> <li>- Token must be valid and not expired.</li> </ul>                                                                                                                                                                                                            |
| API Key Authentication           | <pre data-bbox="449 1340 1075 1473">Response response = RestAssured     .given()     .header("X-API-Key", apiKey) // Add API key to the request header     .when()     .get(baseUrl);</pre>                                                                                                                                                                                                                                                                                                                             |                                                                                                                                                                                                                                                                                                                                                                                                                                                                      |
| Form Authentication              | <pre data-bbox="401 1484 1115 1862">SessionFilter filter = new SessionFilter();  Response loginResponse = RestAssured.given()     .formParam("username", "your-username") // Username parameter     .formParam("password", "your-password") // Password parameter     .when()     .post(loginUrl);  Response protectedResourceResponse = RestAssured.given()     .cookies(loginResponse.getCookies()) // Pass the session cookies from the login response     .when()     .get(protectedResourceUrl);</pre>             | <p>Form-based authentication is a method where the user submits a form (often including a username and password) to authenticate against a web application. In this type of authentication, credentials are typically sent as form parameters (such as username and password), and the server issues a session or token for subsequent requests.</p>                                                                                                                 |
| OAuth 2.0(Authorization Only)    | <p>OAuth 2.0 is a more advanced and flexible version of OAuth that provides a framework for token-based authorization. It supports various grant types, including authorization code, client credentials, and refresh tokens. Example using google photos to share your photos with 3rd party apps.</p> <pre data-bbox="401 2024 822 2120">// 1. First, obtain the access token String clientId = "your-client-id"; String clientSecret = "your-client-secret";</pre>                                                   | <p>Grant Types:</p> <ul style="list-style-type: none"> <li>- Authorization Code: Used for web and mobile applications.</li> <li>- Client Credentials: Used for server-to-server communication.</li> <li>- Password Credentials: Used for user-to-server communication.</li> <li>- Refresh Token: Used to obtain new access tokens.</li> </ul>                                                                                                                        |

|                                       |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                     |                                                                          |
|---------------------------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|--------------------------------------------------------------------------|
|                                       | <pre> String tokenUrl = "<a href="https://oauth-server.com/token">https://oauth-server.com/token</a>"; // Token endpoint URL  Response tokenResponse = RestAssured.given()     .auth().preemptive().basic(clientId, clientSecret) // Basic authentication for client credentials     .contentType(ContentType.URLENC) // Content type for OAuth token request     .formParam("grant_type", "client_credentials") // Grant type for OAuth     .when()     .post(tokenUrl);  // Extract access token from the response String accessToken = tokenResponse.jsonPath().getString("access_token");  // 2. Now, use the access token to authenticate requests to the API Response apiResponse = RestAssured.given()     .auth().oauth2(accessToken) // OAuth2 authentication using access token     .when()     .get("https://api.example.com/protected-resource"); </pre>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                | Usage: Include Authorization: Bearer <token> in the HTTP request header. |
| Authorization Code Flow(Social Login) | <p>OpenID Connect (OIDC) is an identity layer built on top of the OAuth 2.0 protocol. It allows for authentication and provides a way to verify the identity of users based on the authentication performed by an authorization server. It also allows for the retrieval of basic profile information about the user.</p> <pre> String tokenEndpoint = "<a href="https://authorization-server.com/oauth2/token">https://authorization-server.com/oauth2/token</a>"; String clientId = "yourClientId"; String clientSecret = "yourClientSecret"; String authorizationCode = "authorizationCode"; // Obtained from authorization server  Response response = RestAssured     .given()     .contentType("application/x-www-form-urlencoded")     .formParam("grant_type", "authorization_code")     .formParam("code", authorizationCode)     .formParam("redirect_uri", "<a href="https://your-redirect-uri.com">https://your-redirect-uri.com</a>")     .formParam("client_id", clientId)     .formParam("client_secret", clientSecret)     .when()     .post(tokenEndpoint);  String accessToken = response.jsonPath().getString("access_token"); System.out.println("Access Token: " + accessToken); </pre> <p>Once you have the access token, you can use it to make authenticated requests to your API endpoints</p> <pre> Response apiResponse = RestAssured.given()     .auth().oauth2(accessToken)     .when()     .get("https://api.example.com/protected-resource"); </pre> |                                                                          |

# Oauth 2.0

18 February 2023 13:00

OAuth is an authentication pattern used in REST APIs to manage access securely.

## Why Authentication?

In a banking application example, after logging in, a single API call (e.g., "account overview") is made. This API internally calls multiple other APIs to gather details (e.g., balances, credit card info, profile details) and displays them. Proper authentication is needed not just for the initial user login but also for backend communication between different APIs.

## Need for OAuth:

- Directly sharing customer credentials (username and password) with all internal APIs is insecure and impractical.
- OAuth introduces an Authorization Server to issue access tokens, which are then used to authenticate API calls securely.

## OAuth Workflow:

### 1. Authorization Server:

- Issues access tokens to internal services or applications.
- Validates requests using grant types to ensure proper access control.

### 2. Grant Types(Way to access Token):

- Client Credentials Grant: Uses a client ID and client secret to get an access token. Common in internal application interactions.
- Password Grant: Uses username and password to obtain an access token. Used in some scenarios but less secure.
- Authorization Code Grant: Another method for obtaining access tokens, often used in user-facing applications.

```
private String accessToken;
@BeforeClass
public void setUp() {
 RestAssured.baseURI = "https://rahulshettyacademy.com";
 Map<String, String> formParams = new HashMap<>();
 formParams.put("client_id", "692183103107-p0m7ent2hk7suguv4vq22hjcfhcr43pj.apps.googleusercontent.com");
 formParams.put("client_secret", "erZOWM9g3UtwNRj340YYaK_W");
 formParams.put("grant_type", "client_credentials");
 formParams.put("scope", "trust");

 String response = given()
 .formParams(formParams)
 .when()
 .post("oauthapi/oauth2/resourceOwner/token")
 .then()
 .statusCode(200)
 .extract()
 .response()
 .asString();

 JsonPath js = new JsonPath(response);
 accessToken = js.getString("access_token");
}

@Test
public void getCourseDetails() {
 // Fetch course details using the access token
 String courseDetails = given()
 .queryParam("access_token", accessToken)
 .when()
 .get("oauthapi/getCourseDetails")
 .then()
 .statusCode(200) // Add status code assertion here for better test validation
 .extract()
 .response()
 .asString();
 System.out.println(courseDetails);
}
```

--Content Type is: application/x-www-form-urlencoded

# Format JSON

02 March 2023 18:11

```
import com.fasterxml.jackson.databind.ObjectMapper;
import com.fasterxml.jackson.databind.ObjectWriter;
import com.fasterxml.jackson.databind.JsonNode;

// your JSON string
String jsonString = "{ \"name\": \"John\", \"age\": 30, \"city\": \"New York\" }";

// create an ObjectMapper object
ObjectMapper objectMapper = new ObjectMapper();

// read the JSON string into a JsonNode object
JsonNode jsonNode = objectMapper.readTree(jsonString);

// create an ObjectWriter with the pretty printer enabled
ObjectWriter objectWriter = objectMapper.writerWithDefaultPrettyPrinter();

// use the ObjectWriter to write the JsonNode as a pretty-printed JSON string
String prettyJsonString = objectWriter.writeValueAsString(jsonNode);

// print the pretty-printed JSON string
System.out.println(prettyJsonString);
```

# Miscellaneous

26 October 2024 18:26

|                                                                  |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                     |
|------------------------------------------------------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Read Files                                                       | .body(new String(Files.readAllBytes(Paths.get("src/test/resources/AddPlace.json"))))<br><br>String jsonString = Files.readString(Paths.get(filePath));                                                                                                                                                                                                                                                                                                                                                              |
| Object Mapper                                                    | ObjectMapper mapper = new ObjectMapper();<br>File file = new File("src/test/resources/complex.json");<br>ComplexJson.ComplexJsonData complexJsonData = mapper.readValue(file, ComplexJson.ComplexJsonData.class);                                                                                                                                                                                                                                                                                                   |
| Assert and Extract                                               | then().assertThat().statusCode(200).body("scope", equalTo("APP"))<br>.header("server", "Apache").extract().response().asString()                                                                                                                                                                                                                                                                                                                                                                                    |
| Static Import                                                    | By using static imports, we can directly access these methods without prefixing them with the class name, which can makes the test code cleaner and more concise. For example, instead of writing RestAssured.given(), you can simply write given().                                                                                                                                                                                                                                                                |
| Method Chaining                                                  | In RestAssured, the method chaining on given(), when(), and then() is made possible through the use of the builder pattern. Each of these methods returns an object of the same class which implements Request Specification interface, this allows to chain them together. This improves readability of the code.<br><br>Builder Pattern: The builder pattern is a design pattern in which an object is constructed step by step. Each step typically returns a modified builder object, allowing method chaining. |
| Sending Attachment                                               | .multiPart("file", new File(ATTACHMENT_PATH))<br><br>In Postman this is sent in Body via Form Data                                                                                                                                                                                                                                                                                                                                                                                                                  |
| JSON Path                                                        | String postResponse = response.asString();<br>JsonPath js = new JsonPath(postResponse);<br><br>String placeId = js.get("place_id");                                                                                                                                                                                                                                                                                                                                                                                 |
| List<Map<String, Object>> courses = jsonPath.getList("courses"); | "courses": [<br>{<br>"title": "Selenium Python",<br>"price": 50,<br>"copies": 6,<br>"sales": {<br>"udemy": 20,<br>"website": 5<br>},<br>"reviews": [<br>{<br>"channel": "udemy",<br>"reviewcount": 20<br>},<br>{<br>"channel": "website",<br>"reviewcount": 5<br>}<br>]<br>}                                                                                                                                                                                                                                        |
| Map<String, Object> dashboard = jsonPath.getMap("dashboard");    | "dashboard": {<br>"purchaseAmount": 910,<br>"website": "rahulshettyacademy.com"<br>}                                                                                                                                                                                                                                                                                                                                                                                                                                |
| Response as Class                                                | CourseDetails courseDetails = given().queryParam("access_token", accessToken).when().get("oauthapi/getCourseDetails").then().statusCode(200).extract().response().as(CourseDetails.class);                                                                                                                                                                                                                                                                                                                          |
| RelaxedHTTPSValidation                                           | RestAssured.useRelaxedHTTPSValidation();                                                                                                                                                                                                                                                                                                                                                                                                                                                                            |
| URL Encoding                                                     | String response = given().urlEncodingEnabled(false).get("https://jira.atlassian.com:443/rest/api/2.0.alpha1/search?jql=project%20=%20BAM%20AND%20issuetype%20=%20Bug").asString();                                                                                                                                                                                                                                                                                                                                  |
| how to get response time from Response object in restassured     | .time(lessThan(2000L)); // Milliseconds                                                                                                                                                                                                                                                                                                                                                                                                                                                                             |
| What is the difference between URL and URI?                      | URL: Uniform Resource Locator, identifies a resource on the internet.<br>URI: Uniform Resource Identifier, a more general term that includes URLs and other identifiers.                                                                                                                                                                                                                                                                                                                                            |
| JSONObject                                                       | {<br>"page": 2,<br>"per_page": 6,<br>"total": 12,<br>"total_pages": 2,<br>"data": [<br>]<br>}                                                                                                                                                                                                                                                                                                                                                                                                                       |

```

 {
 "id": 7,
 "email": "michael.lawson@reqres.in",
 "first_name": "Michael",
 "last_name": "Lawson",
 "avatar": "https://reqres.in/img/faces/7-image.jpg"
 },
 {
 "id": 8,
 "email": "lindsay.ferguson@reqres.in",
 "first_name": "Lindsay",
 "last_name": "Ferguson",
 "avatar": "https://reqres.in/img/faces/8-image.jpg"
 }
],
 "support": {
 "url": "https://contentcaddy.io?utm_source=reqres&utm_medium=json&utm_campaign=referral",
 "text": "Tired of writing endless social media content? Let Content Caddy generate it for you."
 }
}

JSONObject jsonObject=new JSONObject(response);
System.out.println(jsonObject.get("page"));
JSONArray data = jsonObject.getJSONArray("data");
System.out.println("Data Array: " + data);
for (int i = 0; i < data.length(); i++) {
 JSONObject record = data.getJSONObject(i);
 System.out.println("ID: " + record.getInt("id"));
}

JSONObject support=jsonObject.getJSONObject("support");
System.out.println(support.get("url"));

```

|          |                                          |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                     |
|----------|------------------------------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Security | Authentication                           | Validate token expiration, token revocation, replay attacks, and credential brute-forcing                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                           |
|          | Authorization                            | Check for privilege escalation, verify access control rules, and ensure that users cannot access data belonging to other users.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                     |
|          | Data Validation and Input Handling       | Analyze how APIs handle data validation and input handling. Assess how they respond to input formats, including invalid or unexpected data. Pay special attention to potential injection vulnerabilities                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                            |
|          | Error Handling and Exception Management: | Evaluate how APIs handle errors and exceptions. Test their response to different error conditions and ensure that sensitive information is not leaked in error messages                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                             |
|          | Test Cases                               | <ul style="list-style-type: none"> <li>• Status Code Validation for Valid Requests</li> <li>• Authentication Handling with Invalid Credentials</li> <li>• Graceful Handling of Missing or Invalid Parameters</li> <li>• Input Data Validation with Malformed Data(extra data, invalid data, invalid data type, invalid json,empty json)</li> <li>• Timeout Handling under Load</li> <li>• Pagination Functionality Verification:</li> <li>• Schema Validation</li> <li>• Error Handling Capabilities for Meaningful Messages: Evaluate the API's error-handling capabilities by intentionally causing errors, such as invalid inputs or unexpected situations, and confirm that it consistently returns meaningful error messages for troubleshooting.</li> <li>• Response Time Measurement for Various Requests</li> <li>• Handling Large Payloads (File Uploads)</li> <li>• Validate sorting, pagination, filter param</li> </ul> |

# JSON Schema Validation

23 November 2024 18:34

given the following schema

```
{
 "$schema": "http://json-schema.org/draft-04/schema#",
 "title": "Product set",
 "type": "array",
 "items": {
 "title": "Product",
 "type": "object",
 "properties": {
 "id": {
 "description": "The unique identifier for a product",
 "type": "number"
 },
 "name": {
 "type": "string"
 },
 "price": {
 "type": "number",
 "minimum": 0,
 "exclusiveMinimum": true
 },
 "tags": {
 "type": "array",
 "items": {
 "type": "string"
 },
 "minItems": 1,
 "uniqueItems": true
 },
 "dimensions": {
 "type": "object",
 "properties": {
 "length": {"type": "number"},
 "width": {"type": "number"},
 "height": {"type": "number"}
 },
 "required": ["length", "width", "height"]
 },
 "warehouseLocation": {
 "description": "Coordinates of the warehouse with the product",
 "$ref": "http://json-schema.org/geo"
 },
 "required": ["id", "name", "price"]
 }
 }
}
```

```
import io.restassured.RestAssured;
import io.restassured.module.jsv.JsonSchemaValidator;
```

```
RestAssured.given()
 .baseUri("https://api.example.com")
 .when()
 .get("/users")
 .then()
 .assertThat()
 .body(matchesJsonSchema(new File("path/to/your/schema.json")));
```

# JsonPath

25 December 2024 08:32

Read-only utility for extracting data from JSON responses.

```
import io.restassured.path.json.JsonPath;

public class JsonPathExample {
 public static void main(String[] args) {
 String jsonResponse = "{ \"store\": { \"book\": [{ \"category\": \"fiction\", \"author\": \"John\", \"price\": 10 }, { \"category\": \"non-fiction\", \"author\": \"Jane\", \"price\": 15 }], \"bicycle\": { \"color\": \"red\", \"price\": 19.95 } } }";

 // Parse JSON using JsonPath
 JsonPath jsonPath = new JsonPath(jsonPath);

 // Extracting a simple value
 String bicycleColor = jsonPath.getString("store.bicycle.color");
 System.out.println("Bicycle Color: " + bicycleColor);

 // Extracting an array element
 String firstBookAuthor = jsonPath.getString("store.book[0].author");
 System.out.println("First Book Author: " + firstBookAuthor);

 // Extracting all elements in an array
 List<String> bookCategories = jsonPath.getList("store.book.category");
 System.out.println("Book Categories: " + bookCategories);

 // Filtering data with a condition
 int expensiveBookPrice = jsonPath.getInt("store.book.find { it.price > 12 }.price");
 System.out.println("Expensive Book Price: " + expensiveBookPrice);
 }
}
```

## Common Methods in JsonPath

### Basic Data Retrieval:

`getString(path)`: Returns the value at the given path as a String.

`getInt(path)`: Returns the value at the given path as an int.

`getFloat(path)`: Returns the value at the given path as a float.

### Extracting Arrays and Objects:

`getList(path)`: Returns a list of values at the specified path.

`getMap(path)`: Returns a map of key-value pairs.

### Filtering and Searching:

`find()`: Searches for elements in a list that match a condition.

`findAll()`: Returns all elements in a list that match a condition.

### Default Value:

`getOrDefault(path, defaultValue)`: Returns the value at the specified path, or the default value if the path does not exist.

# Jackson

25 December 2024 09:36

Convert Java objects to JSON (serialization) and vice versa (deserialization).

Offers annotations like `@JsonProperty`, `@JsonIgnore`, and `@JsonInclude` for fine-grained control over serialization/deserialization.  
Highly optimized for performance in both serialization and deserialization.

## Useful Classes

### ObjectMapper

Central class for reading and writing JSON.  
Supports both data binding and tree model access.  
`JsonNode`

Represents a node in a JSON tree.  
Provides methods for traversing and manipulating JSON trees.

### ObjectMapper

| Method                                                                  | Description                                                             |
|-------------------------------------------------------------------------|-------------------------------------------------------------------------|
| <code>writeValueAsString(Object obj)</code>                             | Serialize a Java object into a JSON string.                             |
| <code>writeValue(File file, Object obj)</code>                          | Serialize a Java object into a JSON file.                               |
| <code>readValue(String json, Class&lt;T&gt; clazz)</code>               | Deserialize a JSON string into a Java object.                           |
| <code>readTree(String json)</code>                                      | Parse a JSON string into a <code>JsonNode</code> tree.                  |
| <code>convertValue(Object fromValue, Class&lt;T&gt; toValueType)</code> | Convert one Java object to another using JSON as an intermediary.       |
| <code>writerWithDefaultPrettyPrinter()</code>                           | Create a <code>ObjectWriter</code> that outputs pretty-printed JSON.    |
| <code>setSerializationInclusion(JsonInclude.Include inclusion)</code>   | Configure inclusion rules for serialization (e.g., ignore null values). |

### JsonNode Methods

| Method                              | Description                                                                   |
|-------------------------------------|-------------------------------------------------------------------------------|
| <code>get(String fieldName)</code>  | Retrieve a child node by field name.                                          |
| <code>asText()</code>               | Convert the value of the node to a String.                                    |
| <code>isArray() / isObject()</code> | Check if the node is an array or an object.                                   |
| <code>size()</code>                 | Get the size of an array or object node.                                      |
| <code>path(String fieldName)</code> | Navigate to a child node; returns a missing node if the field does not exist. |

| Annotations                   |                                                                        |
|-------------------------------|------------------------------------------------------------------------|
| <code>@JsonProperty</code>    | Map a JSON property to a specific Java field.                          |
| <code>@JsonIgnore</code>      | Ignore a field during serialization or deserialization.                |
| <code>@JsonInclude</code>     | Include/exclude fields based on certain rules (e.g., non-null values). |
| <code>@JsonFormat</code>      | Specify the format for Date and other custom types.                    |
| <code>@JsonDeserialize</code> | Specify a custom deserializer class for a field.                       |
| <code>@JsonSerialize</code>   | Specify a custom serializer class for a field.                         |

## 3. Tree Model Manipulation

java

Copy code

```
import com.fasterxml.jackson.databind.ObjectMapper;
import com.fasterxml.jackson.databind.node.ObjectNode;

public class JacksonTreeModelExample {
 public static void main(String[] args) throws Exception {
 String json = "{ \"name\": \"Learn Jackson\", \"isbn\": \"abc1\", \"aisle\": \"1234\", \"author\": \"John Doe\" }";

 ObjectMapper objectMapper = new ObjectMapper();

 // Parse JSON into a tree
 ObjectNode rootNode = (ObjectNode) objectMapper.readTree(json);

 // Update fields
 rootNode.put("name", "Learn Gson");
 rootNode.put("newField", "Added via Jackson");

 // Serialize back to JSON string
 String updatedJson = objectMapper.writeValueAsString(rootNode);
 System.out.println(updatedJson);
 }
}
```

# JS Basics

19 April 2024 08:37

|                    |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                       |
|--------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Prototype          | <p>JavaScript attaches a hidden object called `__proto__` to objects and functions, allowing us to access properties and methods such as `length` on arrays.</p> <p>### Understanding the Prototype Chain</p> <ol style="list-style-type: none"><li>1. **Prototype Property (`__proto__`)**:<ul style="list-style-type: none"><li>- Every object and function in JavaScript has a hidden property called `__proto__`.</li><li>- This property points to the prototype object from which the current object inherits properties and methods.</li></ul></li><li>2. **Function Prototype**:<ul style="list-style-type: none"><li>- Functions in JavaScript are also objects. They have a special property called `prototype`.</li><li>- This `prototype` property is an object which contains properties and methods that should be inherited by instances created by the function (when used as a constructor).</li></ul></li><li>3. **Prototype Chain**:<ul style="list-style-type: none"><li>- The `__proto__` of an object points to its prototype object. This creates a chain, called the prototype chain, which is used for inheritance.</li><li>- At the top of this chain is `Object.prototype`, whose `__proto__` is `null`. This signifies the end of the chain.</li></ul></li></ol> <p>### Example:</p> <pre>```javascript // Creating a function function Person(name) {     this.name = name; }  // Adding a method to the function's prototype Person.prototype.sayHello = function() {     return 'Hello, my name is \${this.name}'; };  // Creating an instance of Person const alice = new Person('Alice');  // Accessing properties and methods console.log(alice.name); // Output: Alice console.log(alice.sayHello()); // Output: Hello, my name is Alice  // Understanding the prototype chain, object has 3 while function has 2 chain links console.log(alice.__proto__ === Person.prototype); // Output: true console.log(Person.prototype.__proto__ === Object.prototype); // Output: true console.log(Object.prototype.__proto__); // Output: null ``` </pre> |
| Method Overloading | <p>In JavaScript, method overloading is not directly supported primarily because JavaScript functions do not have method signatures in the same way that languages like Java do.</p> <p>Method overloading relies on the ability to define multiple methods with the same name but with different parameter lists. When a method is called, the correct version is selected based on the number and types of arguments provided.</p> <p>However, in JavaScript, functions do not have a fixed number of parameters or predefined types for parameters. This means that JavaScript functions are not inherently aware of the number or types of arguments being passed to them at runtime, making it difficult to determine which overloaded version of the function should be called.</p> <p>JavaScript instead relies on dynamic typing and flexible argument handling, allowing functions to accept any number of arguments of any type. This dynamic nature of JavaScript functions makes it challenging to implement method overloading in the same way as in statically typed languages.</p> <p>While you can simulate method overloading by checking the number and types of arguments passed to a function and implementing different behavior accordingly, it lacks the clarity and type safety provided by method overloading in statically typed languages.</p>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                             |
| Method Overriding  | <p>In JavaScript, the concept of method overriding, as seen in languages like Java, is not directly supported due to differences in how inheritance and method resolution work.</p> <p>In languages like Java, method overriding allows a subclass to provide a specific implementation of a method that is already defined in its superclass. This enables polymorphic behavior, where the method called depends on the runtime type of the object.</p>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                              |

|          |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                               |
|----------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
|          | <p>However, JavaScript follows a prototype-based inheritance model rather than a class-based one. In JavaScript, objects inherit properties and behaviors from other objects directly through their prototype chain, rather than through the concept of classes and subclasses.</p> <p>When you define a method on an object in JavaScript, it becomes a property of that object. If you later define a method with the same name on a different object in the prototype chain, it simply overrides the previous definition. There's no inherent concept of "superclass" or "subclass" to facilitate method overriding.</p>                                                   |
| Array    | <pre>Arrays act as ArrayList in JS. var marks = Array(6); var marks = [10, 20, 30, 50]; console.log(marks[1]);//20 console.log("Array Length is " + marks.length);//4 marks.push(65);// Add element to end of array 'marks' marks.pop();// Remove element from end of array 'marks' marks.unshift(9);// Add element to beginning of array 'marks' and shift deletes 1st element and returns it console.log(marks); // [ 9, 10, 20, 30, 50 ] console.log(marks.indexOf(20));// Output: 2 console.log(marks.includes(20));// Output: true subMarks = marks.slice(1, 3);#[10,20] Delete subMarks[0] =&gt; this doesn't reduce array length rather creates a hole in array.</pre> |
| For Loop | <pre>// For loop to output numbers from 0 to 9 for (let j = 0; j &lt; 10; j++) {   console.log(j); }  //For of Loop: Use for...of to iterate over the values of an iterable object.  const array = ['Alice', 'Bob', 'Charlie'];  for (let name of array) {   console.log(name); }  //For in Loop: Use for...in to iterate over the properties of an object.  const person = {   name: 'Alice',   age: 25,   city: 'New York' };  for (let key in person) {   console.log(key + ': ' + person[key]); }  //ForEach array.forEach((element)=&gt; {   console.log(element); });</pre>                                                                                             |
| Reduce   | <pre>// Calculate sum of elements in 'marks' array var total = marks.reduce((sum, mark) =&gt; sum + mark, 0); console.log("Sum of Marks is " + total);  //Calculate Max Value let temp=[32,45,14,-2]; let ans=temp.reduce((max,num)=&gt;{ if(num&gt;max){   return num; }else{   return max } },temp[0]); console.log(ans);</pre>                                                                                                                                                                                                                                                                                                                                             |
| Filter   | <pre>// Filter even numbers from 'marks' array var evenMarks = marks.filter(a =&gt; a % 2 == 0); console.log("evenMarks is " + evenMarks);</pre>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                              |
| Map      | <pre>// Map operation on 'marks' array var modifiedMarks = marks.map(mark =&gt; mark + 2); console.log(modifiedMarks); // [ 11, 12, 22, 32, 52 ]</pre>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                        |

|                 |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                            |
|-----------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
|                 | <pre> let temp=[32,45,14,-2]; function tempToCelcius(fah){     return (fah-32)*(5/9); } temp=temp.map(tempToCelcius); console.log(temp); </pre>                                                                                                                                                                                                                                                                                                                                                                                                                            |
| Sort            | <pre> // Sort array 'fruits' const fruits = ["banana", "apple", "orange", "grape", "kiwi"]; fruits.sort(); console.log("Sorted fruits:", fruits);  // Sort array 'numbers' in ascending order const numbers = [3, 1, 8, 5, 2, 7]; numbers.sort((a, b) =&gt; a - b); //Simple sorts number by first digit only e.g 1,11,2,22,3,34,5 console.log("Sorted numbers in ascending order:", numbers);  // Sort array 'numbers' in descending order numbers.sort((a, b) =&gt; b - a); console.log("Sorted numbers in descending order:", numbers); </pre>                          |
| Var vs Let      | <pre> // Conditional block with 'var' declaration if (1 == 1) {     var temp = 2;     console.log("Temp is " + temp); } console.log("Temp is " + temp); //2  // Conditional block with 'let' declaration if (1 == 1) {     let t2 = 2;     console.log("Temp is " + t2); } //console.log("Temp is " + t2); //undefined </pre>                                                                                                                                                                                                                                              |
| Slice and Split | <pre> // Initialize 'day' variable let day="Tuesday "; console.log("Length of Day is "+day.length);  // Slice 'day' string. Creates a new array subDay=day.slice(1,4); console.log(subDay); //ues console.log(day[0]); //T  // Split 'day' string let splitDay=day.split("s"); console.log(splitDay); //['Tue', 'day '] console.log(splitDay[0].trim());  slice can handle negative indices by counting from the end of the string. substring treats negative indices as 0. In substring, if indexStart is greater than indexEnd, the method swaps the two indices. </pre> |
| Type Conversion | <pre> // Convert string to integer let date="23"; let IntDate=parseInt(date);  // Convert integer to string IntDate.toString(); let quote=day+" is a Funday"; console.log(quote.indexOf('day'));//4 console.log(quote.indexOf('day',5)); //17, 5 here is the starting point of the search </pre>                                                                                                                                                                                                                                                                           |
| JS Object       | <pre> // Define object 'person' let person = {     firstName: 'Saurav',     lastName: 'Singh',     age: 24,     fullName: function() {         return this.firstName + " " + this.lastName;     } };  console.log(person.fullName()); </pre>                                                                                                                                                                                                                                                                                                                               |

|                         |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                        |
|-------------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
|                         | <pre> console.log(person.firstName); console.log(person['lastName']);  // Add new property to 'person' object person.gender='Male'; console.log(person);//{ firstName: 'Saurav', lastName: 'Singh', gender: 'Male' }  // Delete property 'gender' from 'person' object delete person.gender;  // Check if 'gender' property exists in 'person' object let condition='gender' in person; console.log(condition);//false  // Iterate through 'person' object properties for(let key in person){     console.log(person[key]); } </pre>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                   |
| Splice                  | <p>used to add or remove elements from an array. Returns deleted items</p> <pre> array.splice(start, deleteCount, item1, item2, ..., itemN) start: The index at which to start changing the array. If greater than the length of the array, no elements will be removed. If negative, it will begin that many elements from the end of the array. deleteCount: An integer indicating the number of elements to remove from the array. If deleteCount is 0, no elements are removed. If deleteCount is omitted, all elements from the start index to the end of the array will be removed. item1, item2, ..., itemN: The elements to add to the array, starting from the start index. If no elements are specified, splice will only remove elements.  //Remove let fruits = ['apple', 'banana', 'cherry', 'date', 'elderberry']; let removed = fruits.splice(1, 2); // Removes 2 elements starting from index 1 console.log(fruits); // Output: ['apple', 'date', 'elderberry'] console.log(removed); // Output: ['banana', 'cherry']  //Add let fruits = ['apple', 'banana', 'cherry']; fruits.splice(1, 0, 'date', 'elderberry'); // Adds 'date' and 'elderberry' at index 1 console.log(fruits); // Output: ['apple', 'date', 'elderberry', 'banana', 'cherry']  //Replace let fruits = ['apple', 'banana', 'cherry']; fruits.splice(1, 1, 'date', 'elderberry'); // Removes 1 element at index 1 and adds 'date' and 'elderberry' console.log(fruits); // Output: ['apple', 'date', 'elderberry', 'cherry'] </pre> |
| concat                  | array1.concat(array2);                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                 |
| indexOf                 | <ul style="list-style-type: none"> <li>- The `indexOf` method returns the first index at which a given element can be found in the array.</li> <li>- If the element is not found, it returns -1.</li> </ul>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                            |
| includes                | <ul style="list-style-type: none"> <li>- The `includes` method determines whether an array includes a certain value among its entries.</li> <li>- It returns `true` if the array contains the value, otherwise it returns `false`.</li> </ul>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                          |
| Java Methods            | Concat,Split,Replace, ReplaceAll,trim, trimStart, trimEnd,startwith,endswith,IndexOf,lastIndexOf                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                       |
| Destructuring           | <p>Destructuring is a convenient way of extracting multiple values from data stored in objects and arrays.</p> <pre> const array = [1, 2, 3]; const [a, b, c] = array; console.log(a); // 1 console.log(b); // 2 console.log(c); // 3 </pre>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                           |
| Remove Array Duplicates | <pre> function removeDuplicates(array) {     return [...new Set(array)]; // Uses the spread operator (...) to convert the Set back into an array. }  ----- function removeDuplicates(array) {     return array.filter((item, index) =&gt; array.indexOf(item) === index); } </pre>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                     |
| Int to String           | Const numstr=String(num)                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                               |
| Float to Int            | Const intNum=parseInt(2.22)                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                            |
| Swap                    | Let a=5;<br>Let b=10;<br>[a,b]=[b,a];                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                  |
| Remove Falsy Values     | function removeFalsyValues(arr) {     return arr.filter(Boolean); }                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                    |

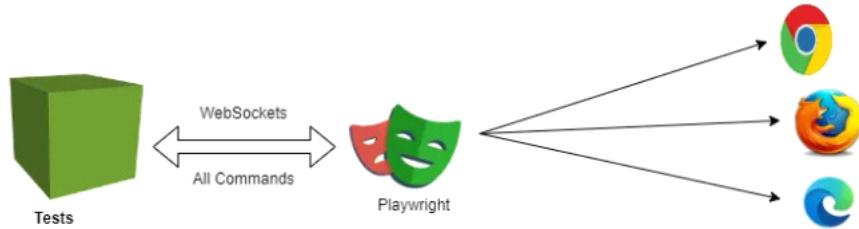
|                           |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                     |
|---------------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
|                           | <pre> }  // Example usage: const arrayWithFalsyValues = [0, 1, false, 2, "", 3, null, 4, undefined, 5, NaN, 6]; const cleanedArray = removeFalsyValues(arrayWithFalsyValues); console.log(cleanedArray); // Output: [1, 2, 3, 4, 5, 6] </pre>                                                                                                                                                                                                                                                                                                                                                                                       |
| Object has a property     | <pre> function hasNameProperty(person) {     return person.hasOwnProperty('name'); }  // Example usage: const person1 = { name: 'John', age: 30 }; const person2 = { age: 25 };  console.log(hasNameProperty(person1)); // Output: true console.log(hasNameProperty(person2)); // Output: false </pre>                                                                                                                                                                                                                                                                                                                              |
| Change Case               | <pre> const upperCaseStr = str.toUpperCase(); const lowerCaseStr = str.toLowerCase(); </pre>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                        |
| Array includes an element | <pre> arr.includes('java'); </pre>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                  |
| Empty Array               | <pre> Array.length === 0 </pre>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                     |
| String to float           | <pre> parseFloat("32");//32 parseFloat("32.21");//32.21 </pre>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                      |
| Join Array Element        | <pre> function joinArrayElements(arr, separator = ',') {     return arr.join(separator); }  // Example usage: const exampleArray = ['apple', 'banana', 'cherry']; const joinedStringDefault = joinArrayElements(exampleArray); // Using default separator (comma) const joinedStringCustom = joinArrayElements(exampleArray, '-'); // Using custom separator  console.log(joinedStringDefault); // Output: "apple,banana,cherry" console.log(joinedStringCustom); // Output: "apple - banana - cherry" </pre>                                                                                                                       |
| Read Object Property      | <pre> const person = {     name: 'John',     age: 30,     occupation: 'Engineer' };  const name = person['name']; const name = person.name; </pre>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                  |
| Shallow Copy              | <p>A shallow copy of an array means that a new array is created, but the elements inside the array are still references to the same objects in memory as those in the original array.</p> <pre> // Original array containing both primitive values and an object const originalArray = [1, 2, 3, { a: 1, b: 2 }];  // Creating a shallow copy using the slice() method const shallowCopy = originalArray.slice();  // Modifying the object in the original array originalArray[3].a = 99;  console.log(originalArray); // Output: [1, 2, 3, { a: 99, b: 2 }] console.log(shallowCopy); // Output: [1, 2, 3, { a: 99, b: 2 }] </pre> |
| Shallow Copy of Array     | <ol style="list-style-type: none"> <li>Using slice()       <pre> ----- const originalArray = [1, 2, 3, { a: 1, b: 2 }]; const shallowCopy = originalArray.slice(); console.log(shallowCopy); </pre> </li> <li>Using Spread Operator (...)        <pre> ----- const originalArray = [1, 2, 3, { a: 1, b: 2 }]; const shallowCopy = [...originalArray]; console.log(shallowCopy); </pre> </li> <li>Using Array.from()</li> </ol>                                                                                                                                                                                                      |

|                    |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                 |
|--------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
|                    | <pre>----- const originalArray = [1, 2, 3, { a: 1, b: 2 }]; const shallowCopy = Array.from(originalArray); console.log(shallowCopy);  4. Using concat() ----- const originalArray = [1, 2, 3, { a: 1, b: 2 }]; const shallowCopy = originalArray.concat(); console.log(shallowCopy);  5. Using map() ----- const originalArray = [1, 2, 3, { a: 1, b: 2 }]; const shallowCopy = originalArray.map(item =&gt; item); console.log(shallowCopy);</pre>                                                                                                                                                                                                                                                                                                                                                                                                                                                                             |
| Object to Array    | <p>1. Converting Object Values to an Array</p> <pre>----- const obj = { a: 1, b: 2, c: 3 }; const valuesArray = Object.values(obj); console.log(valuesArray); // Output: [1, 2, 3]</pre> <p>2. Converting Object Keys to an Array</p> <pre>----- const obj = { a: 1, b: 2, c: 3 }; const keysArray = Object.keys(obj); console.log(keysArray); // Output: ['a', 'b', 'c']</pre> <p>3. Converting Object Entries to an Array</p> <pre>----- const obj = { a: 1, b: 2, c: 3 }; const entriesArray = Object.entries(obj); console.log(entriesArray); // Output: [['a', 1], ['b', 2], ['c', 3]]</pre> <p>4. Converting Object to an Array of Key-Value Pairs</p> <pre>----- const obj = { a: 1, b: 2, c: 3 }; const keyValuePairsArray = Object.entries(obj).map(([key, value]) =&gt; ({ key, value })); console.log(keyValuePairsArray); // Output: [{ key: 'a', value: 1 }, { key: 'b', value: 2 }, { key: 'c', value: 3 }]</pre> |
| Regular Expression | <p>/More/ will find an element that has text that contains More:<br/>   /^More/ will find an element that has text that starts with More</p> <p>/More\$/ will find an element that has text that ends with More<br/>   /More[a-z]/ will find an element that contains More followed by a character in the range a-z</p>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                         |

# Architecture

Friday, June 14, 2024 1:34 PM

Playwright works on Web socket connection protocol, it means once you will trigger the test, the code will be converted into JSON format and will be sent to the server using Web socket protocol.



| Feature               | Selenium                                                                                      | Playwright                                                                                        |
|-----------------------|-----------------------------------------------------------------------------------------------|---------------------------------------------------------------------------------------------------|
| Communication Method  | Separate HTTP requests for each command, receiving JSON responses.                            | Single WebSocket connection for all requests.                                                     |
| Connection Management | Connection is terminated after each request, requiring re-establishment for the next request. | Connection stays in place until test execution is completed.                                      |
| Performance Impact    | Slower execution due to the need to re-establish connections, introducing flakiness.          | Faster execution as commands are sent quickly on a single connection, reducing points of failure. |
| Stability             | Potential for flakiness due to the time taken to establish connections for each request.      | More stable as the WebSocket connection is maintained throughout the test case execution.         |

# Locators and Page Actions

13 May 2024 11:46

| <u>Actions</u>                        | <u>Locator</u>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                            |
|---------------------------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Click                                 | <pre>await page.getByRole('button', { name: 'Sign In' }).click(); await page.getByRole('button', { name: 'Sign In' }).click({timeout:1500}); locator(selector).click({ force: true });//click even if the element is not interactable  // Double click await page.getText('Item').dblclick();  // Right click await page.getText('Item').click({ button: 'right' });  // Shift + click await page.getText('Item').click({ modifiers: ['Shift'] });  // Ctrl + click or Windows and Linux // Meta + click on macOS await page.getText('Item').click({ modifiers: ['ControlOrMeta'] });  // Click the top left corner await page.getText('Item').click({ position: { x: 0, y: 0 } });</pre> |
| Scroll                                | // Scroll the footer into view, forcing an "infinite list" to load more content<br>await page.getText('Footer text').scrollIntoViewIfNeeded();                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                            |
| Press Keys                            | // Hit Enter await page.getText('Submit').press('Enter');  // Dispatch Control+Right await page.getRole('textbox').press('Control+ArrowRight');  // Press \$ sign on keyboard await page.getRole('textbox').press('\$');                                                                                                                                                                                                                                                                                                                                                                                                                                                                  |
| Accessing List of WebElements         | First Element=await page.locator("div[class='card h-100'] h4 a").nth(0).click(); First Element=await page.locator("div[class='card h-100'] h4 a").first().click(); Last Element=await page.locator("div[class='card h-100'] h4 a").last().click();                                                                                                                                                                                                                                                                                                                                                                                                                                        |
| Select Dropdown                       | const roleSelect = page.getRole('combobox'); await roleSelect.selectOption('stud');                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                       |
| Check & Uncheck Radio Button/CheckBox | await termConditions.check(); await termConditions.uncheck();                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                             |
| Waiting For Element                   | await page.locator("div li").first().waitFor(); await page.waitForSelector("a[class='blinkingText']"); await page.waitForTimeout(1000); //HardWait waitForLoadState('networkidle') await locator.waitFor({ state: 'attached' }); Await page.waitForResponse('api URL');                                                                                                                                                                                                                                                                                                                                                                                                                   |
| Actions in New Page                   | const [newPage] = await Promise.all([context.waitForEvent('page'), await blinkingTextLink.click()]); If link opens more than 1 page then capture the page as another array element const [newPage,newPage2] await blinkingTextLink.click(); const nextPage = await context.waitForEvent('page'); console.log(await nextPage.locator(".red").textContent()); await page.bringToFront();                                                                                                                                                                                                                                                                                                    |
| Back and forward Navigation           | goBack() and goForward()                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                  |
| List of Elements                      | const allBoxes = await page.locator('.box'); const boxes = await allBoxes.all(); // Get all the matching elements  for (const box of boxes) {   await box.click(); }                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                      |
| Intelligence of Playwright Locators   | await page.locator('nb-card',{hasText:'Using the Grid'}).getByPlaceholder('Email').fill('test@test.com');  Here nb-card is the parent element, but playwright will locate only that nb card where any child has text Using the Grid                                                                                                                                                                                                                                                                                                                                                                                                                                                       |

|                                                |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                   |
|------------------------------------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Increase Test/expect Timeout                   | <pre>test('My Test with Custom Timeout', async ({ page }, testInfo) =&gt; {   testInfo.setTimeout(testInfo.timeout+2000);    test('basic test', async ({ page }) =&gt; {     await expect(page.getByRole('button')).toHaveText('Sign in', { timeout: 10000 });   });    ToHaveText is autowaiting });</pre>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                       |
| Click all options in Dropdown and validate css | <pre>const colors = {   "Light": "rgb(255, 255, 255)",   "Dark": "rgb(34, 43, 69)",   "Cosmic": "rgb(50, 50, 89)",   "Corporate": "rgb(255, 255, 255)" }  await dropDownMenu.click()  for(const color in colors){   await optionList.filter({hasText: color}).click()   await expect(header).toHaveCSS('background-color', colors[color])   if(color != ""){     await dropDownMenu.click()   } }</pre>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                           |
| JS Alert                                       | <pre>// Listen for the dialog event page.on('dialog', async dialog =&gt; {   console.log(dialog.message()); // Log the message of the alert   await dialog.accept(); // Accept the alert (click "OK") });</pre> <ul style="list-style-type: none"> <li>• <code>dialog.message()</code>: Retrieves the message displayed in the dialog (for alerts, confirms, and prompts).</li> <li>• <code>dialog.accept([text])</code>: Accepts the dialog, and if it's a prompt, you can pass a string as the text response (this simulates typing into the input).</li> <li>• <code>dialog.dismiss()</code>: Dismisses the dialog, simulating a click on "Cancel" for confirms and prompts, or "OK" for alerts in some cases.</li> <li>• <code>dialog.type()</code>: (For prompt dialogs) Lets you type a response into the prompt's input field (although <code>dialog.accept()</code> with text is more common).</li> </ul> |
| Hover                                          | <pre>await page.hover('#mousehover');</pre>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                       |
| iFrames                                        | <pre>const iframe= page.frameLocator('#courses-iframe'); await iframe.getByRole('link', { name: 'NEW Learning paths' }).click();</pre>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                            |
| Take Screenshot                                | <p>Page SS: <pre>await page.screenshot({ path: 'example.png' });</pre></p> <p>Element SS:<br/> <code>const element = await page.waitForSelector(selector);<br/> await element.screenshot({ path: screenshotPath });</code></p>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                    |
| String Array                                   | <pre>const fruits = ['apple', 'banana', 'cherry']; expect(fruits).toContain('banana'); // Passes because 'banana' is in the array</pre>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                           |
| File Upload                                    | <pre>// Select the file input element and upload a file const fileInput = await page.locator('input[type="file"]'); await fileInput.setInputFiles('path/to/your/file.txt'); // Specify the file path  // Optionally, you can submit the form or wait for the upload to complete await page.click('button[type="submit"]'); // Assuming there's a submit button</pre>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                              |
| File Download                                  | <pre>const [download] = await Promise.all([   page.waitForEvent('download'),   page.click('button#download-button') ]);  const downloadPath = await download.path(); console.log('Downloaded file at:', downloadPath);  // After test, clean up downloaded file if (downloadPath) {   fs.unlinkSync(downloadPath); // Remove the downloaded file }</pre>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                          |

| <u>Text Function</u>                          | <u>Locator</u>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                               |
|-----------------------------------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Sending Text                                  | await username.fill("selena@gomez.com");<br>await page.locator("input.input").nth(5).pressSequentially("ind");<br>await page.locator("input.input").nth(5).pressSequentially("ind", { delay: 100 });                                                                                                                                                                                                                                                                                                                                                                                                                                                                         |
| Clearing Text in Input                        | await username.clear();                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                      |
| Getting Text                                  | await page.locator("div[style*='block']").textContent();//Auto Waiting                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                       |
| Get Texts of All WebElements                  | console.log(await page.locator(".card-body b").allTextContents());//Not Autowaiting                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                          |
| Assert String                                 | expect(loginErrorMessage).toBe('Incorrect username/password.');//strict equality checks                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                      |
| Extracts the current value of the input field | const emailValue = await email.inputValue();                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                 |
| Retrieve Attribute                            | const attributeValue = await locator.getAttribute(attributeName);                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                            |
| Assert Input Value                            | await expect(input).toHaveValue('test value');                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                               |
| <u>Assert</u>                                 | <u>Locator</u>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                               |
| Page Title                                    | let pageTitle = await page.title();<br>await expect(page).toHaveTitle('Google');                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                             |
| Selection                                     | expect(await termConditions.isChecked()).toBeFalsy();<br>await expect(termConditions).toBeChecked();                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                         |
| Text with locator                             | await expect(page.locator(".user__name [type='text']").first()).toHaveText("selena@gomez.com");//FullMatch<br>await expect(page.locator("div[style*='block']")).toContainText('Incorrect');//partialMatch                                                                                                                                                                                                                                                                                                                                                                                                                                                                    |
| Attribute                                     | await expect(blinkingTextLink).toHaveAttribute('class', 'blinkingText');                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                     |
| Element Visibility                            | expect(await page.locator("h3:has-text('IPHONE 13 PRO')").isVisible()).toBeTruthy();<br>await expect(page.getByPlaceholder('Hide/Show Example')).toBeVisible();<br>await expect(page.getByPlaceholder('Hide/Show Example')).toBeHidden();                                                                                                                                                                                                                                                                                                                                                                                                                                    |
| Compare Screenshots                           | await expect(page).toHaveScreenshot(['..', '..', 'resources', 'landing.png']);<br>Update SS: npx playwright test --update-snapshots                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                          |
| Soft Assertion                                | const softExpect = expect.soft(page);<br>await softExpect.toHaveTitle('Playwright - Not the actual title');                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                  |
| Negative Matcher                              | expect(value).not.toEqual(0);<br>await expect(locator).not.toContainText('some text');                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                       |
| <u>Name</u>                                   | <u>Locator</u>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                               |
| Filter() and Locator Chaining                 | await page.locator("app-card").filter({hasText:'Nokia Edge'}).getByRole('button').click();                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                   |
| Text Locator                                  | await product.locator("text= Add To Cart").click();<br>await page.getText("Checkout").click();<br>Page.locator("h3: has-text('zara')");                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                      |
| Playwright Locator Parameters                 | await page.locator('.card-body', { has: page.locator('.child-element') });<br>await page.locator('.card-body', { hasText: 'Product Name' });<br>await page.locator('.card-body', { hasNotText: 'Out of Stock' });<br>await page.locator('.card-body', { hasExactText: 'Exact Product Name' });                                                                                                                                                                                                                                                                                                                                                                               |
| Playwright Locators                           | 1. page.getByRole(): await page.getByRole('button', { name: 'Add To Cart', exact: true });<br>2. page.getText():page.getText('Welcome to the site!');<br>3. page.getLabel():await page.getLabel('Checkbox Label').check()<br>4. page.getPlaceholder(): page.getPlaceholder('Enter your email');<br>5. page.getAltText():page.getAltText('Company Logo');<br>6. page.getTitle():const linkLocator = page.getTitle('Go to Example');                                                                                                                                                                                                                                           |
| getByRole tags                                | page.getByRole('button', { name: 'Submit' });<br>page.getByRole('link', { name: 'Home' });<br>page.getByRole('textbox', { name: 'Username' });<br>page.getByRole('checkbox', { name: 'Remember me' });<br>page.getByRole('radio', { name: 'Male' });<br>page.getByRole('combobox', { name: 'Country' });//dropdown<br>page.getByRole('heading', { level: 1, name: 'Welcome' });<br>page.getByRole('list', { name: 'Items' });<br>page.getByRole('listitem', { name: 'Item 1' });<br>page.getByRole('table', { name: 'Data Table' });<br>page.getByRole('img', { name: 'Logo' });<br>page.getByRole('row', { name: 'Row 1' });<br>page.getByRole('cell', { name: 'Cell 1' }); |
| Parent Traversal                              | page.locator("xpath=..").locator("xpath=..").locator().click();                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                              |

# API

10 June 2024 07:36

|                                 |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                               |
|---------------------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Import Request for HTTP request | <pre>import { test, expect, request } from '@playwright/test';</pre>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                          |
| Import Custom Classes           | <pre>import { APIUtils } from './utils/APIUtils';  const apiUtils = new APIUtils(apiContext);  export class APIUtils {     constructor(apiContext) {         this.apiContext = apiContext;     }      async getToken(loginPayload) {         const loginResponse = await this.apiContext.post("https://rahulshettyacademy.com/api/ecom/auth/login", { data: loginPayload });         const loginResponseJson = await loginResponse.json();         return loginResponseJson.token;     }      async createOrder(orderPayload, token) {         const orderResponse = await this.apiContext.post("https://rahulshettyacademy.com/api/ecom/order/create-order", {             data: orderPayload,             headers: {                 'Authorization': this.getToken(),                 'Content-Type': 'application/json'             }         });         const orderResponseJson = await orderResponse.json();         return orderResponseJson.orders[0];     } }</pre> |
| JS Object as Payload            | <pre>const loginPayload = { userEmail: "selena@gomez.com", userPassword: "Iamking@000" };</pre>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                               |
| API Context                     | <pre>const apiContext = await request.newContext();</pre>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                     |
| Sending POST Request            | <pre>const orderResponse = await this.apiContext.post("https://rahulshettyacademy.com/api/ecom/order/create-order", {     data: orderPayload,     headers: {         'Authorization':token,         'Content-Type': 'application/json'     } });</pre>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                        |
| Convert response to JSON        | <pre>const orderResponseJson = await orderResponse.json();</pre>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                              |
| Inject token into localStorage  | <pre>await page.addInitScript(token =&gt; {     window.localStorage.setItem('token', token); }, token);</pre>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                 |
| Session Storage                 | <pre>let webContext; test.beforeAll(async ({browser}) =&gt; {     const context=await browser.newContext();     const page=await context.newPage();     await page.goto("https://rahulshettyacademy.com/client");     await page.getByPlaceholder("email@example.com").fill('selena@gomez.com');     await page.getByPlaceholder('enter your password').fill('Iamking@000');     await page.getRole('button').click();     await page.waitForLoadState('networkidle');     await context.storageState({path:'state.json'});     webContext=await browser.newContext({storageState:'state.json'}); });  test('Client Test', async () =&gt; {      const page=await webContext.newPage();     await page.goto("https://rahulshettyacademy.com/client");</pre>                                                                                                                                                                                                                   |
| Intercept response              | <pre>test('Network Response Intercept Test', async ({ page }) =&gt; {     const requiredProductName = "IPHONE 13 PRO";      // Inject the token into localStorage     await page.addInitScript(value =&gt; {</pre>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                            |

|                   |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                               |
|-------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
|                   | <pre>         window.localStorage.setItem('token', value); }, token);  await page.goto("https://rahulshettyacademy.com/client"); await page.route("https://rahulshettyacademy.com/api/ecom/order/get-orders-for-customer/*",async route =&gt; {     const response = await page.request.fetch(route.request());     let body = JSON.stringify(fakePayLoadOrders);     route.fulfill({ response, body});//only override body in the response object });  // Go to orders page await page.getByRole('button', { name: 'Orders' }).click(); await page.waitForResponse("https://rahulshettyacademy.com/api/ecom/order/get-orders-for-customer/*") console.log(await page.locator(".mt-4").textContent()); console.log("Test Passed"); }); </pre>                                                                                                                 |
| Intercept Request | <pre> test('Security test request intercept', async ({ page }) =&gt; {      await page.addInitScript(value =&gt; {         window.localStorage.setItem('token', value);     }, token);     await page.goto("https://rahulshettyacademy.com/client");     await page.locator("button[routerlink*='myorders']").click();     await page.route("https://rahulshettyacademy.com/api/ecom/order/get-orders-details?id=*",         route =&gt; route.continue({ url: '<a href="https://rahulshettyacademy.com/api/ecom/order/get-orders-details?id=621661f884b053f6765465b6">https://rahulshettyacademy.com/api/ecom/order/get-orders-details?id=621661f884b053f6765465b6</a>' }));     await page.locator("button:has-text('View')").first().click();     await expect(page.locator("p").last()).toHaveText("You are not authorize to view this order"); }) </pre> |
| Block Request     | <pre> await page.route('**/*.{css}', route =&gt; route.abort()); await page.route('**/*.{jpg,png}', route =&gt; route.abort()); </pre>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                        |

# Excel

16 June 2024 10:49

|                                  |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                              |
|----------------------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Installation                     | <code>npm install exceljs</code>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                             |
| Importing the Library            | <code>const { Workbook } = require('exceljs');</code>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                        |
| Creating a Workbook Instance     | <code>const workbook = new Workbook();</code>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                |
| Reading the Excel File           | <code>await workbook.xlsx.readFile('../download.xlsx');</code>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                               |
| Accessing the Worksheet          | <code>const worksheet = workbook.getWorksheet('Sheet1');</code>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                              |
| Iterating Through Rows and Cells | <pre>worksheet.eachRow((row, rowNum) =&gt; {     row.eachCell((cell, colNumber) =&gt; {         console.log(cell.value);     }); });</pre> <p>The <code>eachRow</code> method iterates over all rows in the worksheet.</p> <ul style="list-style-type: none"><li>• <code>row</code> is the current row object.</li><li>• <code>rowNum</code> is the number of the current row.</li></ul> <p>Inside the row iteration, the <code>eachCell</code> method iterates over all cells in the current row.</p> <ul style="list-style-type: none"><li>• <code>cell</code> is the current cell object.</li><li>• <code>colNumber</code> is the number of the current column.</li></ul> |

# Miscellaneous Points

25 May 2024 11:28

|                        |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                 |
|------------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Installation           | <code>npm init playwright@latest --yes --quiet --browser=chromium --browser=firefox --browser=webkit</code>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                     |
| Test Execution         | <p>npx is a command-line tool that comes with Node.js. It allows you to execute packages and binaries from the node_modules directory without needing to install them globally. This is useful for running tools like Playwright directly without additional setup.</p> <p>Npx playwright test triggers all the js files in parallel and all tests in single file sequentially</p> <p>Headless: <code>npx playwright test</code><br/>     Headed: <code>npx playwright --headed</code><br/>     Debug: <code>npx playwright test ./tests/dropdowns.spec.js --debug</code><br/>     Codegen: <code>npx playwright codegen <a href="https://www.google.com/">https://www.google.com/</a></code><br/>     Run Test with UI Test Runner: <code>npx playwright test --ui</code><br/>     Run Test with Trace: <code>npx playwright test --trace on</code><br/> <code>test.fixme()</code> marks the test as failing. Playwright will not run this test, as opposed to the fail annotation. Use fixme when running the test is slow or crashes.<br/> <code>test.slow()</code> marks the test as slow and triples the test timeout.</p> <p>Skip a test: <code>test.skip</code><br/>     Skip Test on Condition<br/> <code>test('skip this test', async ({ page, browserName }) =&gt; {<br/>     test.skip(browserName === 'firefox', 'Still working on it');<br/> });</code></p> <p>Custom Config file: <code>npx playwright test --config ps.config1.js</code><br/>     Custom Project: <code>npx playwright test --config ps.config1.js --project=chrome</code><br/>     Await <code>page.pause()//Open Inspector tool</code></p> <p><code>test.describe.configure({mode:'parallel'});</code><br/> <code>test.describe.configure({mode:'serial'});//If one test fails, next tests will be skipped</code></p> <p><code>test.describe('Tag: @Client',{tag:'@demo'}, () =&gt; {<br/> npx playwright test --grep "@demo"<br/> npx playwright test --grep-invert @fast//skip @fast test<br/> npx playwright test --grep "@fast @slow"<br/> npx playwright test --grep "(?=.*@fast)(?=.*@slow)";//Run tests containing both tags</code></p> |
| Async & Await          | <p>The <code>async</code> keyword defines an asynchronous function, automatically returning a promise.</p> <p>The <code>await</code> keyword, usable only within <code>async</code> functions, pauses execution until the promise is resolved. Use <code>await</code> for function which have promise response type.</p> <p><code>await</code> activates when the <code>async</code> keyword is present in the test definition.</p> <p><code>async</code> and <code>await</code> simplify managing asynchronous operations, allowing the code to pause at specific points until promises are resolved.</p>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                      |
| Browser Fixture        | Browser argument is called fixture(global variable).These fixtures do not need explicit declaration and can be used directly.The browser fixture allows you to invoke a browser instance and needs to be decalred inside {} to be recognised as playwright fixture.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                             |
| Callback Function      | Passed as argument to another function                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                          |
| Before Hooks           | The `beforeAll` hook runs once before all tests in a test suite.<br>The `beforeEach` hook runs before each individual test in a test suite                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                      |
| Workers                | By Default 5 tests run in parallel                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                              |
| Tagging in Config file | <pre>const { defineConfig } = require('@playwright/test');  module.exports = defineConfig({   projects: [     {       name: 'smoke',       grep: /@smoke/,     },     {       name: 'regression',       grep: /@regression/,     },   ], });  npx playwright test --project=smoke npx playwright test --project=regression</pre>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                |

|                           |                                                                                                                                                                                                                                                                                                                                                                                                                                                         |
|---------------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Running from package.json | <pre>{   "name": "playwright",   "version": "1.0.0",   "description": "",   "main": "Excel.spec.js",   "scripts": {     "regression": "npx playwright test",     "web": "npx playwright test --grep @web"   },   "keywords": [],   "author": "",   "license": "ISC",   "devDependencies": {     "@playwright/test": "^1.44.0",     "@types/node": "^20.12.11"   },   "dependencies": {     "exceljs": "^4.4.0"   } }</pre> <p>npm run web</p>           |
| Share Page between Tests  | <pre>import { test, type Page } from '@playwright/test';  test.describe.configure({ mode: 'serial' });  let page: Page;  test.beforeAll(async ({ browser }) =&gt; {   page = await browser.newPage(); });  test.afterAll(async () =&gt; {   await page.close(); });  test('runs first', async () =&gt; {   await page.goto('https://playwright.dev/'); });  test('runs second', async () =&gt; {   await page.getText('Get Started').click(); });</pre> |
| Update PW                 | <pre># Update playwright npm install -D @playwright/test@latest  # Install new browsers npx playwright install</pre>                                                                                                                                                                                                                                                                                                                                    |
| FullScreen                | <pre>{   name: 'chromium',   use: {     ...devices['Desktop Chrome'],     deviceScaleFactor: undefined,     viewport: null,     launchOptions: {       args: ['--start-maximized']     },   }, },</pre>                                                                                                                                                                                                                                                 |

# Fixtures as Hook Alternative

03 July 2024 08:41

```
JS todo.spec.js > ⚡ test.describe('todo tests') callback
1 import { test } from '@playwright/test';
2 import { TodoPage } from '../pageObjects/todo-page';

3 test.describe('todo tests', () => {
4 let todoPage;
5
6 test.beforeEach(async ({ page }) => {
7 todoPage = new TodoPage(page);
8 await todoPage.goto();
9 await todoPage.addTodo('item1');
10 await todoPage.addTodo('item2');
11 });
12
13 test.afterEach(async () => {
14 await todoPage.removeAll();
15 });
16
17 test('should add an item', async () => {
18 await todoPage.addTodo('my item');
19 // ...
20 });
21
22 test('should remove an item', async () => {
23 await todoPage.remove('item1');
24 // ...
25 });
26});
```

```
tests > TS example.spec.ts > ...
1 import base from '@playwright/test';
2 import { TodoPage } from '../pageObjects/todo-page';
3
4 // Extend basic test by providing a "todoPage" fixture.
5 const test = base.extend({
6 todoPage: async ({ page }, use) => {
7 const todoPage = new TodoPage(page);
8 await todoPage.goto();
9 await todoPage.addTodo('item1');
10 await todoPage.addTodo('item2');
11 await use(todoPage);
12 await todoPage.removeAll();
13 },
14});
```

```
16 test('should add an item', async ({ todoPage }) => {
17 await todoPage.addTodo('my item');
18 // ...
19 });
20
21 test('should remove an item', async ({ todoPage }) => {
22 await todoPage.remove('item1');
23 // ...
24 });
```

```
import base from '@playwright/test';
import { TodoPage } from '../pageObjects/todo-page';

// Extend basic test by providing a "todoPage" fixture.
const test = base.extend({
 todoPage: async ({ page }, use) => {
 const todoPage = new TodoPage(page);
 await todoPage.goto();
 await todoPage.addTodo('item1');
 await todoPage.addTodo('item2');
 await use(todoPage);
 await todoPage.removeAll();
 },
});

test('should add an item', async ({ todoPage }) => {
 await todoPage.addTodo('my item');
 // ...
});

test('should remove an item', async ({ todoPage }) => {
 await todoPage.remove('item1');
 // ...
});
```

# Custom Fixtures

03 July 2024 09:07

```
import { test as base, expect as baseExpect } from '@playwright/test';
import { TodoPage } from '../pageObjects/todo-page';
import { SettingsPage } from '../pageObjects/settings-page';

const test = base.extend({
 todoPage: async ({ page }, use) => {
 const todoPage = new TodoPage(page);
 await todoPage.goto();
 await todoPage.addToDo('item1');
 await todoPage.addToDo('item2');

 await use(todoPage);

 await todoPage.removeAll();
 },
 settingsPage: async ({ page }, use) => {
 await use(new SettingsPage(page));
 },
});

export { test, baseExpect as expect };
```

Usage

```
import { test, expect } from '../utils/my-test-fixture';

✓ test.beforeEach(async ({ settingsPage }) => {
 await settingsPage.switchToDarkMode();
}

✓ test('basic test', async ({ todoPage, page }) => {
 await todoPage.addToDo('something nice');
 await expect(page.getByTestId('todo-title')).toContainText(['something nice']);
});
```

# Project Dependencies

03 July 2024 10:19

```
import { defineConfig } from '@playwright/test';
import path from 'path';
export const STORAGE_STATE = path.join(__dirname, 'playwright/.auth/user.json');

export default defineConfig({
 testDir: './tests',
 // Configure the reporter
 reporter: 'html',
 // Retry on CI only
 retries: process.env.CI ? 2 : 0,
 // Run tests in files in parallel
 fullyParallel: true,

 use: {
 //baseURL: 'https://rahulshettyacademy.com/client',
 // run traces on the first retry of a failed test
 trace: 'on-first-retry',
 headless:false
 },
 projects: [
 {
 name: 'setup',
 testMatch: '**/*setup.ts',
 },
 {
 name: 'e2e tests logged in',
 dependencies: ['setup'],
 use: {
 storageState: STORAGE_STATE,
 },
 },
 {
 name: 'e2e tests',
 testIgnore: ['**/*loggedin.spec.ts', '**/*setup.ts'],
 },
],
});
```

```
import { test as setup, expect } from '@playwright/test';
import { STORAGE_STATE } from '../playwright.config';

setup('do login', async ({ page }) => {
 await page.goto('https://rahulshettyacademy.com/client');
 await page.getByPlaceholder('email@example.com').fill('selena@gomez.com');
 await page.getByPlaceholder('enter your password').fill('Iamking@000');
 await page.getByRole('button').click();
 await page.waitForLoadState('networkidle');
 await page.context().storageState({ path: STORAGE_STATE });
});
```

```
import { test, expect } from '@playwright/test';

// test.beforeEach(async ({ page }) => {
// await page.goto('https://rahulshettyacademy.com/client');
// });

test('menu', async ({ page }) => [
 const productName = "IPHONE 13 PRO";
 await page.locator(".card-body", { hasText: productName }).getByRole('button', { name: 'Add To Cart' }).click();
 await page.getText("Cart").nth(0).click();
 await page.locator("div li").first().waitFor();
])
```

# CSV as Test Data

03 July 2024 20:16

```
import fs from 'fs';
import path from 'path';
import { test } from '@playwright/test';
import { parse } from 'csv-parse/sync';

const records = parse(fs.readFileSync(path.join(__dirname, 'input.csv')), {
 columns: true,
 skip_empty_lines: true
});

for (const record of records) {
 test(`foo: ${record.test_case}`, async ({ page }) => {
 console.log(record.test_case, record.some_value, record.some_other_value);
 });
}
```

# Basics

30 March 2025 18:19

| Introduction                | <b>Spring Framework</b> – Provides key features like Dependency Injection (DI) & auto-wiring, making apps maintainable.<br><b>Spring Boot</b> – Simplifies Spring setup, reducing boilerplate code drastically.<br><br>Before Spring: 1000+ lines of setup code<br>With Spring: ~700 lines<br>With Spring Boot: ~400 lines                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                            |                                                     |                      |                                                                                                                                                                                                                                                                                                                                                                                                                                        |                    |                                                                                                                                                                                                                                                                                                                                                                                                                                                 |                           |                                                                   |                    |                                              |                                             |                                                                         |                   |                                                               |                     |                                                       |
|-----------------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-----------------------------------------------------|----------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|--------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|---------------------------|-------------------------------------------------------------------|--------------------|----------------------------------------------|---------------------------------------------|-------------------------------------------------------------------------|-------------------|---------------------------------------------------------------|---------------------|-------------------------------------------------------|
| Jakarta Vs Spring           | <b>Jakarta EE</b> : Defines the standards (Servlet, JPA, etc.)<br><b>Spring</b> : Makes working with those standards easier and faster<br>Both <b>co-exist</b> — but <b>Spring is the preferred implementation</b> in modern Java web development. Spring is not a competitor to Java EE — it's built on top of it.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                   |                                                     |                      |                                                                                                                                                                                                                                                                                                                                                                                                                                        |                    |                                                                                                                                                                                                                                                                                                                                                                                                                                                 |                           |                                                                   |                    |                                              |                                             |                                                                         |                   |                                                               |                     |                                                       |
| Spring Projects             | <table border="1"> <tr><td><b>Spring Core</b></td><td>Foundation for all Spring projects; handles core principles</td></tr> <tr><td><b>Spring MVC</b></td><td>Builds traditional web applications using MVC architecture</td></tr> <tr><td><b>Spring Boot</b></td><td>Simplifies Spring application setup, especially for microservices</td></tr> <tr><td><b>Spring Data</b></td><td>Simplifies data access (JPA, NoSQL, etc.)</td></tr> <tr><td><b>Spring Security</b></td><td>Handles authentication, authorization, and overall application security</td></tr> <tr><td><b>Spring AOP</b></td><td>Supports Aspect-Oriented Programming (cross-cutting concerns)</td></tr> <tr><td><b>Spring Cloud</b></td><td>Helps build scalable, cloud-native microservices apps</td></tr> </table>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                              |                                                     | <b>Spring Core</b>   | Foundation for all Spring projects; handles core principles                                                                                                                                                                                                                                                                                                                                                                            | <b>Spring MVC</b>  | Builds traditional web applications using MVC architecture                                                                                                                                                                                                                                                                                                                                                                                      | <b>Spring Boot</b>        | Simplifies Spring application setup, especially for microservices | <b>Spring Data</b> | Simplifies data access (JPA, NoSQL, etc.)    | <b>Spring Security</b>                      | Handles authentication, authorization, and overall application security | <b>Spring AOP</b> | Supports Aspect-Oriented Programming (cross-cutting concerns) | <b>Spring Cloud</b> | Helps build scalable, cloud-native microservices apps |
| <b>Spring Core</b>          | Foundation for all Spring projects; handles core principles                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                           |                                                     |                      |                                                                                                                                                                                                                                                                                                                                                                                                                                        |                    |                                                                                                                                                                                                                                                                                                                                                                                                                                                 |                           |                                                                   |                    |                                              |                                             |                                                                         |                   |                                                               |                     |                                                       |
| <b>Spring MVC</b>           | Builds traditional web applications using MVC architecture                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                            |                                                     |                      |                                                                                                                                                                                                                                                                                                                                                                                                                                        |                    |                                                                                                                                                                                                                                                                                                                                                                                                                                                 |                           |                                                                   |                    |                                              |                                             |                                                                         |                   |                                                               |                     |                                                       |
| <b>Spring Boot</b>          | Simplifies Spring application setup, especially for microservices                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                     |                                                     |                      |                                                                                                                                                                                                                                                                                                                                                                                                                                        |                    |                                                                                                                                                                                                                                                                                                                                                                                                                                                 |                           |                                                                   |                    |                                              |                                             |                                                                         |                   |                                                               |                     |                                                       |
| <b>Spring Data</b>          | Simplifies data access (JPA, NoSQL, etc.)                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                             |                                                     |                      |                                                                                                                                                                                                                                                                                                                                                                                                                                        |                    |                                                                                                                                                                                                                                                                                                                                                                                                                                                 |                           |                                                                   |                    |                                              |                                             |                                                                         |                   |                                                               |                     |                                                       |
| <b>Spring Security</b>      | Handles authentication, authorization, and overall application security                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                               |                                                     |                      |                                                                                                                                                                                                                                                                                                                                                                                                                                        |                    |                                                                                                                                                                                                                                                                                                                                                                                                                                                 |                           |                                                                   |                    |                                              |                                             |                                                                         |                   |                                                               |                     |                                                       |
| <b>Spring AOP</b>           | Supports Aspect-Oriented Programming (cross-cutting concerns)                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                         |                                                     |                      |                                                                                                                                                                                                                                                                                                                                                                                                                                        |                    |                                                                                                                                                                                                                                                                                                                                                                                                                                                 |                           |                                                                   |                    |                                              |                                             |                                                                         |                   |                                                               |                     |                                                       |
| <b>Spring Cloud</b>         | Helps build scalable, cloud-native microservices apps                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                 |                                                     |                      |                                                                                                                                                                                                                                                                                                                                                                                                                                        |                    |                                                                                                                                                                                                                                                                                                                                                                                                                                                 |                           |                                                                   |                    |                                              |                                             |                                                                         |                   |                                                               |                     |                                                       |
| Spring Context              | A <b>Spring Container</b> (also called <b>Spring Context</b> or <b>IOC Container</b> ) is responsible for managing Spring beans and their lifecycle.<br><br><b>Steps to Create a Spring Context:</b> <ol style="list-style-type: none"> <li>Define a Configuration Class.</li> <li>Launch Spring Context using <code>AnnotationConfigApplicationContext</code>.</li> </ol>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                            |                                                     |                      |                                                                                                                                                                                                                                                                                                                                                                                                                                        |                    |                                                                                                                                                                                                                                                                                                                                                                                                                                                 |                           |                                                                   |                    |                                              |                                             |                                                                         |                   |                                                               |                     |                                                       |
| IoC                         | Inversion of Control is a design principle where the control of object creation and management is transferred from the application code to a container or framework (like Spring). It shifts control of object creation from the programmer to Spring which <b>Creates and manages objects</b> (IoC). <div style="border: 1px solid black; padding: 10px; margin-top: 10px;"> <table border="1"> <tr> <td>Traditional Approach</td> <td> <code>CricketCoach coach = new CricketCoach();</code><br/><br/>           This approach is rigid and tightly coupled.<br/>           With IoC, we <b>delegate the responsibility</b> of creating the object to an external container, making the application:           <ul style="list-style-type: none"> <li>More <b>configurable</b></li> <li>Easier to <b>maintain and extend</b></li> <li>Aligned with <b>best practices</b> like <b>SOLID principles</b></li> </ul> </td> </tr> <tr> <td>IoC Approach</td> <td>           The <b>Spring Container</b> does two major things:           <ol style="list-style-type: none"> <li><b>Creates and manages objects</b> (IoC)</li> <li><b>Injects dependencies</b> into those objects (Dependency Injection or DI)</li> </ol> <br/> <code>Coach myCoach = context.getBean("myCoach", Coach.class);</code><br/><br/>           Spring reads the configuration, creates the appropriate object (like <code>CricketCoach</code>), and gives you a reference.         </td> </tr> </table> </div> |                                                     | Traditional Approach | <code>CricketCoach coach = new CricketCoach();</code><br><br>This approach is rigid and tightly coupled.<br>With IoC, we <b>delegate the responsibility</b> of creating the object to an external container, making the application: <ul style="list-style-type: none"> <li>More <b>configurable</b></li> <li>Easier to <b>maintain and extend</b></li> <li>Aligned with <b>best practices</b> like <b>SOLID principles</b></li> </ul> | IoC Approach       | The <b>Spring Container</b> does two major things: <ol style="list-style-type: none"> <li><b>Creates and manages objects</b> (IoC)</li> <li><b>Injects dependencies</b> into those objects (Dependency Injection or DI)</li> </ol><br><code>Coach myCoach = context.getBean("myCoach", Coach.class);</code><br><br>Spring reads the configuration, creates the appropriate object (like <code>CricketCoach</code> ), and gives you a reference. |                           |                                                                   |                    |                                              |                                             |                                                                         |                   |                                                               |                     |                                                       |
| Traditional Approach        | <code>CricketCoach coach = new CricketCoach();</code><br><br>This approach is rigid and tightly coupled.<br>With IoC, we <b>delegate the responsibility</b> of creating the object to an external container, making the application: <ul style="list-style-type: none"> <li>More <b>configurable</b></li> <li>Easier to <b>maintain and extend</b></li> <li>Aligned with <b>best practices</b> like <b>SOLID principles</b></li> </ul>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                |                                                     |                      |                                                                                                                                                                                                                                                                                                                                                                                                                                        |                    |                                                                                                                                                                                                                                                                                                                                                                                                                                                 |                           |                                                                   |                    |                                              |                                             |                                                                         |                   |                                                               |                     |                                                       |
| IoC Approach                | The <b>Spring Container</b> does two major things: <ol style="list-style-type: none"> <li><b>Creates and manages objects</b> (IoC)</li> <li><b>Injects dependencies</b> into those objects (Dependency Injection or DI)</li> </ol><br><code>Coach myCoach = context.getBean("myCoach", Coach.class);</code><br><br>Spring reads the configuration, creates the appropriate object (like <code>CricketCoach</code> ), and gives you a reference.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                       |                                                     |                      |                                                                                                                                                                                                                                                                                                                                                                                                                                        |                    |                                                                                                                                                                                                                                                                                                                                                                                                                                                 |                           |                                                                   |                    |                                              |                                             |                                                                         |                   |                                                               |                     |                                                       |
| Auto-wiring                 | Auto-wiring = Automatic Dependency Injection<br>Spring automatically: <ul style="list-style-type: none"> <li>Scans for classes annotated with <code>@Component</code></li> <li>Matches them by <b>type</b> (interface or class)</li> <li>Injects the matching implementation</li> </ul>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                               |                                                     |                      |                                                                                                                                                                                                                                                                                                                                                                                                                                        |                    |                                                                                                                                                                                                                                                                                                                                                                                                                                                 |                           |                                                                   |                    |                                              |                                             |                                                                         |                   |                                                               |                     |                                                       |
| <code>@Configuration</code> | <ul style="list-style-type: none"> <li>Marks a class that declares one or more <b>Spring Beans</b>.</li> <li>Methods inside this class will define the <b>beans Spring manages</b>.</li> </ul>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                        |                                                     |                      |                                                                                                                                                                                                                                                                                                                                                                                                                                        |                    |                                                                                                                                                                                                                                                                                                                                                                                                                                                 |                           |                                                                   |                    |                                              |                                             |                                                                         |                   |                                                               |                     |                                                       |
| <code>@Bean</code>          | <ul style="list-style-type: none"> <li>Marks a method that <b>produces a bean</b> managed by the Spring container.</li> <li><code>@Bean</code> methods define objects that Spring should instantiate and manage.</li> <li>Created automatically when marked with <code>@Component</code>, <code>@Service</code>, or <code>@Repository</code></li> </ul>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                               |                                                     |                      |                                                                                                                                                                                                                                                                                                                                                                                                                                        |                    |                                                                                                                                                                                                                                                                                                                                                                                                                                                 |                           |                                                                   |                    |                                              |                                             |                                                                         |                   |                                                               |                     |                                                       |
| <code>@Component</code>     | <ul style="list-style-type: none"> <li>Marks a class as a Spring-managed <b>bean</b>.</li> <li>If a class is under a <b>component scan</b>, Spring will <b>automatically create</b> an instance of it.</li> <li>There are specialized annotations that provide <b>more context</b> to the framework about the class's role:           <ul style="list-style-type: none"> <li><code>@Controller</code> → Interface to business logic, accepts requests and returns response</li> <li><code>@Service</code> → Business logic processing</li> <li><code>@Repository</code> → Database interactions</li> </ul> </li> </ul>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                |                                                     |                      |                                                                                                                                                                                                                                                                                                                                                                                                                                        |                    |                                                                                                                                                                                                                                                                                                                                                                                                                                                 |                           |                                                                   |                    |                                              |                                             |                                                                         |                   |                                                               |                     |                                                       |
| Component Vs Bean           | <table border="1"> <thead> <tr> <th>Feature</th> <th><code>@Component</code></th> <th><code>@Bean</code></th> </tr> </thead> <tbody> <tr> <td>Usage</td> <td>Applied to a <b>class</b></td> <td>Applied to a <b>method</b> in a configuration class</td> </tr> <tr> <td>Where it is Used?</td> <td>Used in <b>custom application components</b></td> <td>Used in <b>Spring configuration classes</b></td> </tr> </tbody> </table>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                     |                                                     | Feature              | <code>@Component</code>                                                                                                                                                                                                                                                                                                                                                                                                                | <code>@Bean</code> | Usage                                                                                                                                                                                                                                                                                                                                                                                                                                           | Applied to a <b>class</b> | Applied to a <b>method</b> in a configuration class               | Where it is Used?  | Used in <b>custom application components</b> | Used in <b>Spring configuration classes</b> |                                                                         |                   |                                                               |                     |                                                       |
| Feature                     | <code>@Component</code>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                               | <code>@Bean</code>                                  |                      |                                                                                                                                                                                                                                                                                                                                                                                                                                        |                    |                                                                                                                                                                                                                                                                                                                                                                                                                                                 |                           |                                                                   |                    |                                              |                                             |                                                                         |                   |                                                               |                     |                                                       |
| Usage                       | Applied to a <b>class</b>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                             | Applied to a <b>method</b> in a configuration class |                      |                                                                                                                                                                                                                                                                                                                                                                                                                                        |                    |                                                                                                                                                                                                                                                                                                                                                                                                                                                 |                           |                                                                   |                    |                                              |                                             |                                                                         |                   |                                                               |                     |                                                       |
| Where it is Used?           | Used in <b>custom application components</b>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                          | Used in <b>Spring configuration classes</b>         |                      |                                                                                                                                                                                                                                                                                                                                                                                                                                        |                    |                                                                                                                                                                                                                                                                                                                                                                                                                                                 |                           |                                                                   |                    |                                              |                                             |                                                                         |                   |                                                               |                     |                                                       |

|                                                                                                                                                                                                                        | <b>Ease of Use</b><br><b>Autowiring Methods</b><br><b>Who Creates the Bean?</b><br><b>Recommended When?</b>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                 | Very simple, just annotate a class<br>Constructor Injection, Setter Injection, Field Injection<br>Spring via <b>component scanning</b><br>For custom components within the application | Requires manual bean creation logic<br>Autowired via <b>method calls or parameters</b><br>Developer manually writes the bean creation logic<br>When extra logic is needed before bean creation or for third-party library beans |                  |                  |                                                                                                                                                                                                                        |                                                                                                                                                                                                              |                                                                                                                                                 |                                                                                                                                                                                                          |
|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|------------------|------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-------------------------------------------------------------------------------------------------------------------------------------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| @ComponentScan                                                                                                                                                                                                         | <p><b>Component Scanning</b> is Spring's mechanism to <b>automatically detect and register beans</b> (Java objects) into the Spring container using annotations like @Component, @Service, @Repository, and @Controller.</p> <p>Tells Spring where to look for beans. If no package is specified, <b>Spring scans the current package and sub-packages</b>.</p> <p>By default, <b>Spring Boot only scans the package of your main class and its sub-packages</b>.</p> <pre>com.luv2code.springcoredemo // Main app class is here com.luv2code.util // Not a sub-package — won't be scanned</pre> <pre>@SpringBootApplication(scanBasePackages = {     "com.luv2code.springcoredemo",     "com.luv2code.util",     "org.acme.cart",     "edu.cmu.srs" })</pre>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                               |                                                                                                                                                                                        |                                                                                                                                                                                                                                 |                  |                  |                                                                                                                                                                                                                        |                                                                                                                                                                                                              |                                                                                                                                                 |                                                                                                                                                                                                          |
| @SpringBootApplication                                                                                                                                                                                                 | This annotation is a <b>meta-annotation</b> , composed of: <ul style="list-style-type: none"> <li>• <b>@EnableAutoConfiguration</b>: Enables Spring Boot's automatic configuration.</li> <li>• <b>@ComponentScan</b>: Enables component scanning for the current package and sub-packages.</li> <li>• <b>@Configuration</b>: Allows registration of beans or configuration classes.</li> </ul> <p>When you run your Spring Boot application using: <code>SpringApplication.run(MyApp.class, args)</code>; Spring Boot performs the following:</p> <ol style="list-style-type: none"> <li>1. Creates the application context.</li> <li>2. Starts scanning from the package of <code>MyApp.class</code> and its sub-packages.</li> <li>3. Registers all detected components.</li> <li>4. Starts an embedded server (e.g., Tomcat) automatically.</li> </ol>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                   |                                                                                                                                                                                        |                                                                                                                                                                                                                                 |                  |                  |                                                                                                                                                                                                                        |                                                                                                                                                                                                              |                                                                                                                                                 |                                                                                                                                                                                                          |
| @Primary                                                                                                                                                                                                               | A bean should be given preference when multiple candidates qualify                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                          |                                                                                                                                                                                        |                                                                                                                                                                                                                                 |                  |                  |                                                                                                                                                                                                                        |                                                                                                                                                                                                              |                                                                                                                                                 |                                                                                                                                                                                                          |
| @Qualifier                                                                                                                                                                                                             | A specific bean should be auto wired                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                        |                                                                                                                                                                                        |                                                                                                                                                                                                                                 |                  |                  |                                                                                                                                                                                                                        |                                                                                                                                                                                                              |                                                                                                                                                 |                                                                                                                                                                                                          |
| Bean Scope                                                                                                                                                                                                             | <table border="1"> <thead> <tr> <th style="text-align: center;"><b>Singleton</b></th><th style="text-align: center;"><b>Prototype</b></th></tr> </thead> <tbody> <tr> <td>A single instance of the bean is created per Spring IoC container.<br/><br/>Every request for the bean returns the same instance.<br/><br/>This ensures reusability and efficiency, as no new object is created each time.</td><td>A new instance is created every time the bean is requested.<br/><br/>Not managed beyond creation – Spring does not track or destroy prototype beans.<br/><br/>Used for stateful beans that should not be shared.</td></tr> <tr> <td><pre>@Component public class NormalClass {     public NormalClass() {         System.out.println("NormalClass instance created");     } }</pre></td><td><pre>@Component @Scope(ConfigurableBeanFactory.SCOPE_PROTOTYPE) public class PrototypeClass {     public PrototypeClass() {         System.out.println("PrototypeClass instance created");     } }</pre></td></tr> </tbody> </table>                                                                                                                                                                                                                                                                                                                                                                                                                                                               |                                                                                                                                                                                        |                                                                                                                                                                                                                                 | <b>Singleton</b> | <b>Prototype</b> | A single instance of the bean is created per Spring IoC container.<br><br>Every request for the bean returns the same instance.<br><br>This ensures reusability and efficiency, as no new object is created each time. | A new instance is created every time the bean is requested.<br><br>Not managed beyond creation – Spring does not track or destroy prototype beans.<br><br>Used for stateful beans that should not be shared. | <pre>@Component public class NormalClass {     public NormalClass() {         System.out.println("NormalClass instance created");     } }</pre> | <pre>@Component @Scope(ConfigurableBeanFactory.SCOPE_PROTOTYPE) public class PrototypeClass {     public PrototypeClass() {         System.out.println("PrototypeClass instance created");     } }</pre> |
| <b>Singleton</b>                                                                                                                                                                                                       | <b>Prototype</b>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                            |                                                                                                                                                                                        |                                                                                                                                                                                                                                 |                  |                  |                                                                                                                                                                                                                        |                                                                                                                                                                                                              |                                                                                                                                                 |                                                                                                                                                                                                          |
| A single instance of the bean is created per Spring IoC container.<br><br>Every request for the bean returns the same instance.<br><br>This ensures reusability and efficiency, as no new object is created each time. | A new instance is created every time the bean is requested.<br><br>Not managed beyond creation – Spring does not track or destroy prototype beans.<br><br>Used for stateful beans that should not be shared.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                |                                                                                                                                                                                        |                                                                                                                                                                                                                                 |                  |                  |                                                                                                                                                                                                                        |                                                                                                                                                                                                              |                                                                                                                                                 |                                                                                                                                                                                                          |
| <pre>@Component public class NormalClass {     public NormalClass() {         System.out.println("NormalClass instance created");     } }</pre>                                                                        | <pre>@Component @Scope(ConfigurableBeanFactory.SCOPE_PROTOTYPE) public class PrototypeClass {     public PrototypeClass() {         System.out.println("PrototypeClass instance created");     } }</pre>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                    |                                                                                                                                                                                        |                                                                                                                                                                                                                                 |                  |                  |                                                                                                                                                                                                                        |                                                                                                                                                                                                              |                                                                                                                                                 |                                                                                                                                                                                                          |
| Spring Bean Lifecycle                                                                                                                                                                                                  | <p><b>@PostConstruct</b></p> <p>Used to execute initialization logic after dependencies are injected</p> <p>When a bean is created, Spring injects dependencies first.</p> <p>After dependency injection is complete, any method annotated with <b>@PostConstruct</b> is executed.</p> <p>Common use case: Fetching data from a database, initializing resources, or performing setup tasks.</p> <p>The method <code>initialize()</code> will run <b>automatically</b> after the object is created and dependencies are set.</p> <pre>@Component public class SomeClass {     private final SomeDependency someDependency;      public SomeClass(SomeDependency someDependency) {         this.someDependency = someDependency;     }      @PostConstruct     public void initialize() {         System.out.println("Initialization logic executed");     } }</pre> <p><b>@PreDestroy</b></p> <p>Used to execute cleanup logic before the bean is removed from the Spring container</p> <ul style="list-style-type: none"> <li>• Before a bean is removed from the Spring container, a method annotated with <b>@PreDestroy</b> is called.</li> <li>• Common use case: <b>Closing database connections, releasing resources, or performing cleanup tasks</b>.</li> </ul> <pre>@PreDestroy public void cleanup() {     System.out.println("Cleanup before bean is removed."); }</pre> <p>This method executes just <b>before</b> the bean is destroyed, ensuring that any necessary cleanup is performed</p> |                                                                                                                                                                                        |                                                                                                                                                                                                                                 |                  |                  |                                                                                                                                                                                                                        |                                                                                                                                                                                                              |                                                                                                                                                 |                                                                                                                                                                                                          |

|                               | <pre>         System.out.println("All dependencies are ready!");     }      @PostConstruct     public void initialize() {         System.out.println("Initialization logic using someDependency");         someDependency.getReady();     } } </pre>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                   |                |                      |         |            |        |            |            |            |        |        |            |            |
|-------------------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|----------------|----------------------|---------|------------|--------|------------|------------|------------|--------|--------|------------|------------|
| Jakarta CDI                   | <ul style="list-style-type: none"> <li>• <b>CDI (Contexts and Dependency Injection)</b> is a <b>Jakarta EE specification</b> introduced in <b>Java EE 6 (2009)</b>.</li> <li>• <b>Spring supports CDI</b> as it provides similar functionality.</li> <li>• CDI is <b>only a specification</b> (interface) and does not have an implementation.</li> <li>• <b>Spring Framework implements CDI</b> and supports its annotations</li> </ul> <pre> &lt;dependency&gt;     &lt;groupId&gt;jakarta.inject&lt;/groupId&gt;     &lt;artifactId&gt;jakarta.inject-api&lt;/artifactId&gt;     &lt;version&gt;2.0.1&lt;/version&gt; &lt;/dependency&gt; </pre> <table border="1"> <thead> <tr> <th>CDI Annotation</th><th>Equivalent in Spring</th></tr> </thead> <tbody> <tr> <td>@Inject</td><td>@Autowired</td></tr> <tr> <td>@Named</td><td>@Component</td></tr> <tr> <td>@Qualifier</td><td>@Qualifier</td></tr> <tr> <td>@Scope</td><td>@Scope</td></tr> <tr> <td>@Singleton</td><td>@Singleton</td></tr> </tbody> </table> | CDI Annotation | Equivalent in Spring | @Inject | @Autowired | @Named | @Component | @Qualifier | @Qualifier | @Scope | @Scope | @Singleton | @Singleton |
| CDI Annotation                | Equivalent in Spring                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                   |                |                      |         |            |        |            |            |            |        |        |            |            |
| @Inject                       | @Autowired                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                             |                |                      |         |            |        |            |            |            |        |        |            |            |
| @Named                        | @Component                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                             |                |                      |         |            |        |            |            |            |        |        |            |            |
| @Qualifier                    | @Qualifier                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                             |                |                      |         |            |        |            |            |            |        |        |            |            |
| @Scope                        | @Scope                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                 |                |                      |         |            |        |            |            |            |        |        |            |            |
| @Singleton                    | @Singleton                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                             |                |                      |         |            |        |            |            |            |        |        |            |            |
| Spring Boot Starters          | <p>Predefined sets of dependencies for common use cases.</p> <ul style="list-style-type: none"> <li>• spring-web, spring-webmvc</li> <li>• spring-boot-starter-json → JSON conversion (Jackson)</li> <li>• spring-boot-starter-tomcat → Embedded Tomcat server</li> </ul>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                              |                |                      |         |            |        |            |            |            |        |        |            |            |
| Auto-Configuration            | <p><b>Auto-configuration automatically sets up Spring components based on the dependencies in your classpath</b> and any configurations you've already provided.</p> <p>Building a Spring app usually requires:</p> <ul style="list-style-type: none"> <li>• Configuring <b>Component Scan</b></li> <li>• Setting up <b>DispatcherServlet</b></li> <li>• Defining <b>DataSource</b> for DB access</li> <li>• JSON &lt;-&gt; Bean conversion setup</li> </ul> <p>Auto-configuration handles all of this <b>automatically</b>, reducing boilerplate and setup time.</p> <p>Auto Configuration works based on:</p> <ul style="list-style-type: none"> <li>• Dependencies (JARs) in the classpath</li> <li>• Existing user-defined configurations</li> </ul> <p>All logic is in <code>spring-boot-autoconfigure.jar</code></p>                                                                                                                                                                                             |                |                      |         |            |        |            |            |            |        |        |            |            |
| Spring Boot DevTools          | <p>A Spring Boot module that <b>boosts developer productivity</b> by enabling <b>automatic restart</b> of the application upon code changes</p> <p>Without DevTools:</p> <p>Manual server restarts are needed for every code change.</p> <p>With DevTools:</p> <p>Java, properties, and view file changes trigger an automatic restart except pom.xml changes</p> <pre> &lt;dependency&gt;     &lt;groupId&gt;org.springframework.boot&lt;/groupId&gt;     &lt;artifactId&gt;spring-boot-devtools&lt;/artifactId&gt;     &lt;optional&gt;true&lt;/optional&gt;&lt;!-- Optional for production exclusion --&gt; &lt;/dependency&gt; </pre>                                                                                                                                                                                                                                                                                                                                                                              |                |                      |         |            |        |            |            |            |        |        |            |            |
| Managing Config with Profiles | <p>Apps run in <b>multiple environments</b>:</p> <ul style="list-style-type: none"> <li>• Development</li> <li>• QA</li> <li>• Stage</li> <li>• Production</li> </ul> <p>Each environment needs <b>different configurations</b>:</p> <ul style="list-style-type: none"> <li>• Different DB connections</li> <li>• Different logging levels</li> <li>• Different API endpoints</li> </ul>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                               |                |                      |         |            |        |            |            |            |        |        |            |            |

|                                            |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                          |                                            |                                                                                                                                                                                                                                                                                                             |                    |                                                                                                                                                                                                                                                  |
|--------------------------------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|--------------------------------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|--------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
|                                            | <p>Spring <b>Profiles</b> allow you to maintain <b>environment-specific configuration</b> files.</p> <pre>application.properties      → Default (fallback) application-dev.properties → Dev config application-prod.properties → Prod config application-qa.properties   → QA config</pre> <p>How to Configure Profiles</p> <table border="1"> <tr> <td>Create Environment-Specific Property Files</td><td>           In src/main/resources:<br/> <ul style="list-style-type: none"> <li>• application.properties: logging.level.org.springframework=debug</li> <li>• application-dev.properties: logging.level.org.springframework=trace</li> <li>• application-prod.properties: logging.level.org.springframework=info</li> </ul> </td></tr> <tr> <td>Activate a Profile</td><td>           In application.properties: spring.profiles.active=prod<br/>           This tells Spring Boot to <b>merge application.properties + application-prod.properties</b>.<br/>           • If a property is in <b>both files</b>, the one in the <b>active profile wins</b>.         </td></tr> </table>                                                                                                                                                                                                                                                                                                          | Create Environment-Specific Property Files | In src/main/resources:<br><ul style="list-style-type: none"> <li>• application.properties: logging.level.org.springframework=debug</li> <li>• application-dev.properties: logging.level.org.springframework=trace</li> <li>• application-prod.properties: logging.level.org.springframework=info</li> </ul> | Activate a Profile | In application.properties: spring.profiles.active=prod<br>This tells Spring Boot to <b>merge application.properties + application-prod.properties</b> .<br>• If a property is in <b>both files</b> , the one in the <b>active profile wins</b> . |
| Create Environment-Specific Property Files | In src/main/resources:<br><ul style="list-style-type: none"> <li>• application.properties: logging.level.org.springframework=debug</li> <li>• application-dev.properties: logging.level.org.springframework=trace</li> <li>• application-prod.properties: logging.level.org.springframework=info</li> </ul>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                              |                                            |                                                                                                                                                                                                                                                                                                             |                    |                                                                                                                                                                                                                                                  |
| Activate a Profile                         | In application.properties: spring.profiles.active=prod<br>This tells Spring Boot to <b>merge application.properties + application-prod.properties</b> .<br>• If a property is in <b>both files</b> , the one in the <b>active profile wins</b> .                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                         |                                            |                                                                                                                                                                                                                                                                                                             |                    |                                                                                                                                                                                                                                                  |
| @ConfigurationProperties                   | <p>If you have <b>complex or grouped configuration</b> like:</p> <pre>currency-service.url = ... currency-service.username = ... currency-service.key = ...</pre> <p>Instead of reading each property manually, <b>map them to a POJO</b> using @ConfigurationProperties</p> <pre>@Component @ConfigurationProperties(prefix = "currency-service") public class CurrencyServiceConfiguration {     private String url;     private String username;     private String key;      // Generate getters &amp; setters }</pre> <ul style="list-style-type: none"> <li>• <b>@Component</b>: Makes the class a Spring-managed bean</li> <li>• <b>@ConfigurationProperties</b>: Binds config properties to fields</li> <li>• <b>prefix = "currency-service"</b>: Matches all properties starting with this</li> </ul> <p>Define Properties in application.properties</p> <ul style="list-style-type: none"> <li>• currency-service.url=http://default.in28minutes.com</li> <li>• currency-service.username=default-username</li> <li>• currency-service.key=default-key</li> </ul> <p>Override in application-dev.properties</p> <ul style="list-style-type: none"> <li>• currency-service.url=http://dev.in28minutes.com</li> <li>• currency-service.username=dev-username</li> <li>• currency-service.key=dev-key</li> </ul> <p>Specify Environment in application.properties: spring.profiles.active=dev</p> |                                            |                                                                                                                                                                                                                                                                                                             |                    |                                                                                                                                                                                                                                                  |
| Embedded Servers                           | <p>Simplify deployment of Spring Boot applications by embedding the web server directly into the JAR</p> <p><b>Spring Boot Approach (Embedded Server with JAR)</b></p> <ul style="list-style-type: none"> <li>• Spring Boot embeds the server (e.g., Tomcat) inside the application JAR.</li> <li>• Only need Java installed → run app via: java -jar learn-spring-boot-0.0.1-SNAPSHOT.jar</li> </ul>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                    |                                            |                                                                                                                                                                                                                                                                                                             |                    |                                                                                                                                                                                                                                                  |
| Spring Boot Actuator                       | <p>used to:</p> <ul style="list-style-type: none"> <li>• <b>Monitor</b> the application (health, metrics, beans, environment, etc.)</li> <li>• <b>Manage</b> the application in production</li> <li>• Get insights into internal behaviors of the Spring Boot app</li> </ul> <pre>&lt;dependency&gt;   &lt;groupId&gt;org.springframework.boot&lt;/groupId&gt;   &lt;artifactId&gt;spring-boot-starter-actuator&lt;/artifactId&gt; &lt;/dependency&gt;</pre> <p>Default EndPoint: <a href="http://localhost:8080/actuator">http://localhost:8080/actuator</a></p> <p>To Expose More Endpoints, In application.properties:</p> <pre>management.endpoints.web.exposure.include=* management.endpoints.web.exposure.include=health,metrics</pre>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                            |                                            |                                                                                                                                                                                                                                                                                                             |                    |                                                                                                                                                                                                                                                  |

# Dependency Injection

31 March 2025 11:51

Dependency Injection is a design pattern(technique to implement IoC) where an object receives its dependencies (helper components) from an external source (like the Spring Container) instead of creating them itself meaning **DI injects the required dependencies (objects) into a class, rather than the class creating them itself.**

**Spring automatically manages object creation and wiring.**

**Steps in DI:**

- Spring scans for components (@ComponentScan).
- It identifies dependencies of each bean.
- It injects them into the required classes.

**Key Point:** Removes manual object creation (new keyword), making applications more modular.

|                             |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                       |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                          |
|-----------------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Constructor-Based Injection | Component                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                             | <pre>@Component public class CricketCoach implements Coach {     @Override     public String getDailyWorkout() {         return "Practice fast bowling for 15 minutes";     } }</pre>                                                                                                                                                                                                                                                                                                                                                    |
|                             | RestController                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                        | <pre>@RestController public class MyRestController {     private final Coach myCoach;      @Autowired     public MyRestController(Coach coach) {         myCoach = coach;     }      @GetMapping("/dailyworkout")     public String getDailyWorkout() {         return myCoach.getDailyWorkout();     } }</pre>                                                                                                                                                                                                                          |
|                             | Explanation                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                           | <p><b>Scenario Overview:</b></p> <ul style="list-style-type: none"><li>• You have a Coach interface, a CricketCoach class implementing it, and a DemoController that needs a Coach.</li></ul> <p><b>What Spring Does Automatically:</b></p> <ol style="list-style-type: none"><li>1. Creates a new instance of CricketCoach.</li><li>2. Performs constructor injection — injects the CricketCoach into DemoController.</li><li>3. This way, Coach becomes a <b>dependency/helper</b> that is automatically provided by Spring.</li></ol> |
| Setter-Based Injection      | <pre>@Component class YourBusinessClass {      Dependency1 dependency1;     Dependency2 dependency2;      @Autowired     public void setDependency1(Dependency1 dependency1) {         System.out.println("Setter Injection - setDependency1 ");         this.dependency1 = dependency1;     }      @Autowired     public void setDependency2(Dependency2 dependency2) {         System.out.println("Setter Injection - setDependency2 ");         this.dependency2 = dependency2;     }      public String toString() {         return "Using " + dependency1 + " and " + dependency2;     } }</pre> |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                          |

```

@Component
class Dependency1 {}

@Component
class Dependency2 {}

@Configuration
@ComponentScan
public class DeplnjectionLauncherApplication {
 public static void main(String[] args) {
 try (var context = new AnnotationConfigApplicationContext(DeplnjectionLauncherApplication.class)) {
 Arrays.stream(context.getBeanDefinitionNames()).forEach(System.out::println);
 System.out.println(context.getBean(YourBusinessClass.class));
 }
 }
}

```

|                       |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                  |
|-----------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Field-Based Injection | <pre> @Component class YourBusinessClass {      @Autowired     Dependency1 dependency1;     @Autowired     Dependency2 dependency2;      public String toString() {         return "Using " + dependency1 + " and " + dependency2;     } }  @Component class Dependency1 {}  @Component class Dependency2 {}  @Configuration @ComponentScan public class DeplnjectionLauncherApplication {     public static void main(String[] args) {         try (var context = new AnnotationConfigApplicationContext(DeplnjectionLauncherApplication.class)) {             Arrays.stream(context.getBeanDefinitionNames()).forEach(System.out::println);             System.out.println(context.getBean(YourBusinessClass.class));         }     } } </pre> |
|-----------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|

| Feature                 | Constructor-Based                      | Setter-Based                          | Field-Based                                                                                                           |
|-------------------------|----------------------------------------|---------------------------------------|-----------------------------------------------------------------------------------------------------------------------|
| <b>Mutability</b>       | Immutable                              | Mutable                               | Mutable                                                                                                               |
| <b>Mandatory Fields</b> | Enforced                               | Not Enforced                          | Not Enforced                                                                                                          |
| <b>Testability</b>      | Easy to Mock                           | Easy to Mock                          | Hard to Mock                                                                                                          |
| <b>Reflection Usage</b> | No                                     | No                                    | Yes                                                                                                                   |
| <b>Preferred?</b>       | Yes                                    | Sometimes                             | Avoid                                                                                                                 |
| <b>Usage</b>            | Recommended for mandatory dependencies | Recommended for optional dependencies | Dependencies are injected directly into fields using <code>@Autowired</code> .<br>Uses reflection to set dependencies |

# Eager and Lazy Initialization

31 March 2025 13:14

|                                   |                                                                                                                                                                                                                                                                                                                                     |                                                                                                                                       |                                                                      |
|-----------------------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|---------------------------------------------------------------------------------------------------------------------------------------|----------------------------------------------------------------------|
| Eager Initialization<br>(Default) | <ul style="list-style-type: none"><li>All beans are created at application startup when the Spring context loads</li><li>Dependencies are injected immediately, ensuring all components are ready for use.</li><li>If any error occurs during bean creation, the application fails to start, preventing runtime failures.</li></ul> |                                                                                                                                       |                                                                      |
| Lazy Initialization               | <ul style="list-style-type: none"><li>The bean is not created at startup but only when it is first accessed.</li><li>Helps reduce initial memory usage and speeds up startup time.</li><li>Errors in bean creation appear at runtime, not at startup, which can be risky.</li></ul>                                                 |                                                                                                                                       |                                                                      |
|                                   | On Componenet                                                                                                                                                                                                                                                                                                                       | On Config Class                                                                                                                       |                                                                      |
|                                   | <pre>@Component<br/>@Lazy<br/>public class B {<br/>    public B() {<br/>        System.out.println("B initialized");<br/>    }<br/>}</pre>                                                                                                                                                                                          | <pre>@Configuration<br/>@Lazy<br/>public class AppConfig {<br/>    // All beans defined in this class will be lazy-loaded<br/>}</pre> |                                                                      |
| Comparison                        | <b>Feature</b>                                                                                                                                                                                                                                                                                                                      | <b>Lazy Initialization (@Lazy)</b>                                                                                                    | <b>Eager Initialization (Default)</b>                                |
|                                   | <b>When is the bean initialized?</b>                                                                                                                                                                                                                                                                                                | When the bean is first accessed.                                                                                                      | At application startup.                                              |
|                                   | <b>Default behavior?</b>                                                                                                                                                                                                                                                                                                            | <b>Not default</b> (Must use @Lazy).                                                                                                  | <b>Default behavior</b> (No @Lazy required).                         |
|                                   | <b>Error Handling</b>                                                                                                                                                                                                                                                                                                               | Errors appear <b>at runtime</b> when the bean is first used.                                                                          | Errors appear <b>at startup</b> , preventing the app from launching. |
|                                   | <b>Memory Consumption</b>                                                                                                                                                                                                                                                                                                           | Lower initial memory usage (bean is not created until needed).                                                                        | Higher initial memory usage (all beans are loaded at startup).       |
|                                   | <b>Performance Impact</b>                                                                                                                                                                                                                                                                                                           | Faster startup but may cause delays when the bean is accessed.                                                                        | Slower startup but better runtime performance.                       |
|                                   | <b>When to use?</b>                                                                                                                                                                                                                                                                                                                 | For <b>rarely used beans</b> to avoid unnecessary startup costs.                                                                      | For <b>most beans</b> to catch errors early and improve stability.   |

# Hibernate

03 May 2025 15:17

|           |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                             |
|-----------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| JDBC Test | <pre>private static final String JDBC_URL = "jdbc:postgresql://localhost:5432/postgres?user=postgres&amp;password=postgres"; private static final String USERNAME = "postgres"; private static final String PASSWORD = "postgres"; private final Faker faker = new Faker();  private Connection getConnection() throws SQLException {     return DriverManager.getConnection(JDBC_URL, USERNAME, PASSWORD); }  private Integer getMaxRollNo() {     String sql = "SELECT MAX(roll_no) AS max_roll_no FROM student";     try (Connection conn = getConnection();         PreparedStatement stmt = conn.prepareStatement(sql);         ResultSet rs = stmt.executeQuery()) {         if (rs.next()) {             return rs.getInt("max_roll_no");         }     } catch (SQLException e) {         System.out.println(e.getMessage());     }     return null; }  @BeforeClass(enabled = true) public void setup() {     String sql = "CREATE TABLE IF NOT EXISTS student (" +         "roll_no INT PRIMARY KEY, " +         "name VARCHAR(100), " +         "age INT)";     try (Connection conn = getConnection();         Statement stmt = conn.createStatement()) {         stmt.execute(sql);     } catch (SQLException e) {         System.out.println(e.getMessage());     } }  @AfterClass public void teardown() {     printAllStudents(); }  @Test public void printAllStudents() {     String sql = "SELECT * FROM student";     try (Connection conn = getConnection();         PreparedStatement stmt = conn.prepareStatement(sql);         ResultSet rs = stmt.executeQuery()) {         System.out.println("\n--- All Students ---");         while (rs.next()) {             System.out.println("Roll No: " + rs.getInt("roll_no") +                 ", Name: " + rs.getString("name") +                 ", Age: " + rs.getInt("age"));         }     } catch (SQLException e) {         System.out.println(e.getMessage());     } }  @Test public void insertTest() {     Integer maxRollNo = getMaxRollNo();     int newRollNo = (maxRollNo != null ? maxRollNo + 1 : 1);     String sql = "INSERT INTO student (roll_no, name, age) VALUES (?, ?, ?)";     try (Connection conn = getConnection();         PreparedStatement stmt = conn.prepareStatement(sql)) {         stmt.setInt(1, newRollNo);         stmt.setString(2, faker.harryPotter().character());         stmt.setInt(3, faker.number().numberBetween(3, 30));         stmt.executeUpdate();     } catch (SQLException e) {         System.out.println(e.getMessage());     } }  @Test public void getSingleData() {     Integer maxRollNo = getMaxRollNo();     if (maxRollNo != null) {         String sql = "SELECT * FROM student WHERE roll_no = ?";         try (Connection conn = getConnection();             PreparedStatement stmt = conn.prepareStatement(sql)) {             stmt.setInt(1, maxRollNo);         }     } }</pre> |
|-----------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|

```

 ResultSet rs = stmt.executeQuery();
 if (rs.next()) {
 System.out.println("Retrieved: Roll No: " + rs.getInt("roll_no") +
 ", Name: " + rs.getString("name") +
 ", Age: " + rs.getInt("age"));
 } else {
 System.out.println("No student found with roll_no " + maxRollNo);
 }
 } catch (SQLException e) {
 System.out.println(e.getMessage());
 }
} else {
 System.out.println("No students found in the database.");
}
}

@Test
public void updateData() {
 Integer maxRollNo = getMaxRollNo();
 if (maxRollNo != null) {
 String sql = "UPDATE student SET name = ?, age = ? WHERE roll_no = ?";
 try (Connection conn = getConnection()) {
 PreparedStatement stmt = conn.prepareStatement(sql));
 stmt.setString(1, faker.harryPotter().character()); // Set name
 stmt.setInt(2, faker.number().numberBetween(3, 30)); // Set age
 stmt.setInt(3, maxRollNo); // Set roll_no
 stmt.executeUpdate();
 System.out.println("Updated student with roll_no: " + maxRollNo);
 } catch (SQLException e) {
 System.out.println(e.getMessage());
 }
 } else {
 System.out.println("No students found in the database.");
 }
}

@Test
public void deleteTest() {
 Integer maxRollNo = getMaxRollNo();
 if (maxRollNo != null) {
 String sql = "DELETE FROM student WHERE roll_no = ?";
 try (Connection conn = getConnection()) {
 PreparedStatement stmt = conn.prepareStatement(sql));
 stmt.setInt(1, maxRollNo);
 stmt.executeUpdate();
 System.out.println("Deleted student with roll_no: " + maxRollNo);
 } catch (SQLException e) {
 System.out.println(e.getMessage());
 }
 } else {
 System.out.println("No students found in the database.");
 }
}

```

|                |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                |
|----------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| With Hibernate | <pre> private static SessionFactory sessionFactory; private final Faker faker = new Faker();  @BeforeClass public void setup() {     LogManager.getLogManager().reset();     Logger.getLogger("org.hibernate").setLevel(Level.WARNING);     sessionFactory = new Configuration().addAnnotatedClass(Student.class).configure().buildSessionFactory(); }  @AfterClass public void teardown() {     printAllStudents();     if (sessionFactory != null) {         sessionFactory.close();     } }  private Integer getMaxRollNo(Session session) {     return session.createQuery("select max(roll_no) from Student", Integer.class).uniqueResult(); }  private void printAllStudents() {     try (Session session = sessionFactory.openSession()) {         List&lt;Student&gt; allStudents = session.createQuery("from Student", Student.class).list();         System.out.println("\n--- All Students ---");         allStudents.forEach(System.out::println);     } }  @Test public void insertTest() {     try (Session session = sessionFactory.openSession()) {         Integer rno = getMaxRollNo(session);         Transaction transaction = session.beginTransaction();         Student stu = new Student(             rno != null ? ++rno : faker.number().numberBetween(0, 2), </pre> |
|----------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|

```

 faker.harryPotter().character(),
 faker.number().numberBetween(3, 18)
);
 session.persist(stu);
 transaction.commit();
}
}

@Test
public void getSingleData() {
 try (Session session = sessionFactory.openSession()) {
 Integer rno = getMaxRollNo(session);
 if (rno != null) {
 Student student = session.get(Student.class, rno);
 System.out.println(student);
 } else {
 System.out.println("No students found in the database.");
 }
 }
}

@Test
public void updateData() {
 try (Session session = sessionFactory.openSession()) {
 Integer rno = getMaxRollNo(session);
 if (rno != null) {
 Transaction transaction = session.beginTransaction();
 Student stu = new Student(rno, faker.funnyName().name(), faker.number().numberBetween(3, 16));
 session.merge(stu);
 transaction.commit();

 Student updatedStudent = session.get(Student.class, rno);
 System.out.println(updatedStudent);
 } else {
 System.out.println("No students found in the database.");
 }
 }
}

@Test
public void deleteTest() {
 try (Session session = sessionFactory.openSession()) {
 Integer rno = getMaxRollNo(session);
 if (rno != null) {
 Transaction transaction = session.beginTransaction();
 Student student = session.get(Student.class, rno);
 if (student != null) {
 session.remove(student);
 System.out.println("Deleted student with roll_no: " + rno);
 } else {
 System.out.println("Student with roll_no " + rno + " not found.");
 }
 transaction.commit();
 } else {
 System.out.println("No students found in the database.");
 }
 }
}
}

```

```

HQL
private static SessionFactory sessionFactory;

@BeforeClass
public void setup() {
 LogManager.getLogManager().reset();
 Logger.getLogger("org.hibernate").setLevel(Level.WARNING);
 sessionFactory = new Configuration().addAnnotatedClass(Laptop.class).configure().buildSessionFactory();
 insertTest();
}

@AfterClass
public void teardown() {
 getAllLaptops();
 if (sessionFactory != null) {
 sessionFactory.close();
 }
}

private void getAllLaptops() {
 try (Session session = sessionFactory.openSession()) {
 List<Laptop> allLaptops = session.createQuery("from Laptop", Laptop.class).list();
 System.out.println("\n--- All Laptops ---");
 allLaptops.forEach(System.out::println);
 }
}

public void insertTest() {
 try (Session session = sessionFactory.openSession()) {
 Transaction transaction = session.beginTransaction();
 Laptop laptop1 = new Laptop(1, "Dell", "XPS", 32);
 Laptop laptop2 = new Laptop(2, "Dell", "Vostro", 16);
 Laptop laptop3 = new Laptop(3, "Lenovo", "IdeaPad", 16);
 }
}

```

```

 session.persist(laptop1);
 session.persist(laptop2);
 session.persist(laptop3);
 transaction.commit();
 }

 @Test
 public void getById() {
 //Select * from laptop where ram=32 ->SQL
 //from Laptop where ram=32 -> HQL
 try (Session session = sessionFactory.openSession()) {
 System.out.println(session.find(Laptop.class, 3)); // By Primary Key
 Query<Laptop> query = session.createQuery("from Laptop where ram_size=32 and brand in(?1)", Laptop.class);
 query.setParameter(1, "Dell");
 List<Laptop> laptops = query.getResultList();
 System.out.println(laptops);
 }
 }
}

```

```

Hibernate.cfg.xml <?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE hibernate-configuration PUBLIC
 "-//Hibernate/Hibernate Configuration DTD 3.0//EN"
 "http://www.hibernate.org/dtd/hibernate-configuration-3.0.dtd">
<hibernate-configuration>
 <session-factory>
 <property name="hibernate.connection.driver_class">org.postgresql.Driver</property>
 <property name="hibernate.connection.url">jdbc:postgresql://localhost:5432/postgres?
user=postgres&password=postgres</property>
 <property name="hibernate.connection.username">postgres</property>
 <property name="hibernate.connection.password">postgres</property>
 <property name="hibernate.connection.pool_size">10</property>
 <property name="hibernate.dialect">org.hibernate.dialect.PostgreSQLDialect</property>
 <property name="hibernate.current_session_context_class">thread</property>
 <property name="hibernate.cache.provider_class">org.hibernate.cache.internal.NoCacheProvider</property>
 <property name="hibernate.show_sql">false</property>
 <property name="hibernate.format_sql">false</property>
 <property name="hibernate.hbm2ddl.auto">update</property>
 </session-factory>
</hibernate-configuration>

```

# JPA(Jakarta Persistence API)

05 April 2025 17:05

|                                        |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                 |                  |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                             |                                 |                                                                                                                                                                                                                                                                                                                                                                                           |
|----------------------------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|---------------------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Dependencies to Add                    | <ul style="list-style-type: none"> <li>• <b>Spring Web</b> <ul style="list-style-type: none"> <li>• spring-boot-starter-web</li> <li>• Enables REST/web features</li> </ul> </li> <li>• <b>Spring Data JDBC</b> <ul style="list-style-type: none"> <li>• spring-boot-starter-data-jdbc</li> <li>• For direct JDBC interaction using Spring Data</li> </ul> </li> <li>• <b>Spring Data JPA</b> <ul style="list-style-type: none"> <li>• spring-boot-starter-data-jpa</li> <li>• Simplifies ORM and database interactions via Hibernate/JPA</li> </ul> </li> <li>• <b>H2 Database</b> <ul style="list-style-type: none"> <li>• com.h2database:h2</li> <li>• In-memory, fast, ideal for development/testing</li> </ul> </li> </ul>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                 |                  |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                             |                                 |                                                                                                                                                                                                                                                                                                                                                                                           |
| H2 Console                             | <p>Open application.properties (under src/main/resources):</p> <pre>spring.h2.console.enabled=true spring.datasource.url=jdbc:h2:mem:testdb</pre> <p>URL: <a href="http://localhost:8080/h2-console">http://localhost:8080/h2-console</a><br/> DB URL: jdbc:h2:mem:testdb</p> <p>Create <b>schema.sql</b> in src/main/resources</p> <pre>CREATE TABLE course (     id BIGINT NOT NULL,     name VARCHAR(255) NOT NULL,     author VARCHAR(255) NOT NULL,     PRIMARY KEY (id) );</pre> <ul style="list-style-type: none"> <li>• Spring Boot <b>auto-detects</b> schema.sql on startup if you're using Spring Data JPA</li> <li>• It <b>automatically executes</b> the script to create the table</li> </ul>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                     |                  |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                             |                                 |                                                                                                                                                                                                                                                                                                                                                                                           |
| Executing SQL Queries with Spring JDBC | <table border="1"> <tr> <td>Create SQL Query</td> <td> <pre>@Repository public class CourseJdbcRepository {      @Autowired     private JdbcTemplate springJdbcTemplate;      private static final String INSERT_QUERY = """         insert into course (id, name, author)         values (1, 'Learn AWS', 'in28minutes') """;      public void insert() {         springJdbcTemplate.update(INSERT_QUERY);     } }</pre> <ul style="list-style-type: none"> <li>• <b>@Repository</b>: tells Spring it's a DB-access class</li> <li>• <b>JdbcTemplate.update()</b>: used to run <b>insert/update/delete</b> queries</li> <li>• <b>"""</b> is a <b>Text Block</b> (Java 15+): for multi-line strings, keeps SQL readable</li> </ul> </td> </tr> <tr> <td>Run the Insert Query at Startup</td> <td> <pre>@Component public class CourseJdbcCommandLineRunner implements CommandLineRunner {      @Autowired     private CourseJdbcRepository repository;      @Override     public void run(String... args) throws Exception {         repository.insert();     } }</pre> <ul style="list-style-type: none"> <li>• <b>@Component</b>: So Spring can discover and manage this class</li> </ul> </td></tr> </table> | Create SQL Query | <pre>@Repository public class CourseJdbcRepository {      @Autowired     private JdbcTemplate springJdbcTemplate;      private static final String INSERT_QUERY = """         insert into course (id, name, author)         values (1, 'Learn AWS', 'in28minutes') """;      public void insert() {         springJdbcTemplate.update(INSERT_QUERY);     } }</pre> <ul style="list-style-type: none"> <li>• <b>@Repository</b>: tells Spring it's a DB-access class</li> <li>• <b>JdbcTemplate.update()</b>: used to run <b>insert/update/delete</b> queries</li> <li>• <b>"""</b> is a <b>Text Block</b> (Java 15+): for multi-line strings, keeps SQL readable</li> </ul> | Run the Insert Query at Startup | <pre>@Component public class CourseJdbcCommandLineRunner implements CommandLineRunner {      @Autowired     private CourseJdbcRepository repository;      @Override     public void run(String... args) throws Exception {         repository.insert();     } }</pre> <ul style="list-style-type: none"> <li>• <b>@Component</b>: So Spring can discover and manage this class</li> </ul> |
| Create SQL Query                       | <pre>@Repository public class CourseJdbcRepository {      @Autowired     private JdbcTemplate springJdbcTemplate;      private static final String INSERT_QUERY = """         insert into course (id, name, author)         values (1, 'Learn AWS', 'in28minutes') """;      public void insert() {         springJdbcTemplate.update(INSERT_QUERY);     } }</pre> <ul style="list-style-type: none"> <li>• <b>@Repository</b>: tells Spring it's a DB-access class</li> <li>• <b>JdbcTemplate.update()</b>: used to run <b>insert/update/delete</b> queries</li> <li>• <b>"""</b> is a <b>Text Block</b> (Java 15+): for multi-line strings, keeps SQL readable</li> </ul>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                     |                  |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                             |                                 |                                                                                                                                                                                                                                                                                                                                                                                           |
| Run the Insert Query at Startup        | <pre>@Component public class CourseJdbcCommandLineRunner implements CommandLineRunner {      @Autowired     private CourseJdbcRepository repository;      @Override     public void run(String... args) throws Exception {         repository.insert();     } }</pre> <ul style="list-style-type: none"> <li>• <b>@Component</b>: So Spring can discover and manage this class</li> </ul>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                       |                  |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                             |                                 |                                                                                                                                                                                                                                                                                                                                                                                           |

|                                                              |                                                      |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                         |
|--------------------------------------------------------------|------------------------------------------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
|                                                              |                                                      | <ul style="list-style-type: none"> <li>• CommandLineRunner: Interface used to <b>execute code at app startup</b></li> <li>• run() method will be called when the Spring Boot app starts</li> </ul>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                      |
| Spring JDBC – Dynamic Insert & Delete with CommandLineRunner | Create the Course POJO                               | <pre>public class Course {     private long id;     private String name;     private String author;      // No-arg constructor     public Course() {     }      // All-arg constructor     public Course(long id, String name, String author) {         this.id = id;         this.name = name;         this.author = author;     }      // Getters     public long getId() { return id; }     public String getName() { return name; }     public String getAuthor() { return author; }      @Override     public String toString() {         return "Course [id=" + id + ", name=" + name + ", author=" + author + "]";     } }</pre>                                                                                                                                                                                                                                                                                                                                                                                 |
|                                                              | Modify CourseJdbcRepository to Accept Dynamic Insert | <pre>@Repository public class CourseJdbcRepository {      @Autowired     private JdbcTemplate springJdbcTemplate;      private static String INSERT_QUERY = """         insert into course (id, name, author) values (?, ?, ?) """;      private static String DELETE_QUERY = """         delete from course where id = ? """;      private static final String SELECT_QUERY =         "SELECT * FROM course WHERE id = ?";      public void insert(Course course) {         springJdbcTemplate.update(INSERT_QUERY,             course.getId(),             course.getName(),             course.getAuthor()         );     }      public void deleteById(long id) {         springJdbcTemplate.update(DELETE_QUERY, id);     }      public Course findById(long id) {         return jdbcTemplate.queryForObject(             SELECT_QUERY,             new BeanPropertyRowMapper&lt;&gt;(Course.class),             id);     }     //BeanPropertyRowMapper matches column names with     //Java field names. }</pre> |
|                                                              | Use CommandLineRunner to Run at Startup              | <pre>@Component public class CourseJdbcCommandLineRunner implements CommandLineRunner {</pre>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                           |

|                                    |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                 |
|------------------------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
|                                    | <pre>     @Autowired     private CourseJdbcRepository repository;      @Override     public void run(String... args) throws Exception {         repository.insert(new Course(1, "Learn AWS Now!",         "in28minutes"));         repository.insert(new Course(2, "Learn Azure",         "in28minutes"));         repository.insert(new Course(3, "Learn DevOps",         "in28minutes"));          // Delete course with id = 1         repository.deleteById(1);     } } </pre>                                                                                                                                                                                                                              |
| Why JPA over Spring JDBC           | <ul style="list-style-type: none"> <li>Spring JDBC requires SQL queries written manually.</li> <li>As tables and relationships grow, managing SQL becomes complex.</li> <li>JPA abstracts SQL and manages persistence using Java objects (entities).</li> </ul>                                                                                                                                                                                                                                                                                                                                                                                                                                                 |
| Entity                             | <ul style="list-style-type: none"> <li>Java class that maps directly to a database table.</li> <li>Annotated using <code>@Entity</code>.</li> </ul> <pre> @Entity public class Course {     @Id     private Long id;     private String name;     private String author;      // Getters and Setters }  • @Table(name="Course_Details") - if table name differs • @Column(name="course_name") - if field name ≠ column name </pre> <p><b>Behind the Scenes</b></p> <ul style="list-style-type: none"> <li>JPA maps the Java class (Course) to a database table.</li> <li>Fields map to columns.</li> <li>Handles SQL generation for insert, delete, update, and find</li> </ul>                                 |
| JPA Repository using EntityManager | <pre> @Repository @Transactional public class CourseJpaRepository {      @PersistenceContext // preferred over @Autowired, injects EntityManager     private EntityManager entityManager;      public void insert(Course course) {         entityManager.merge(course); // Inserts/Updates entity     }      public Course findById(long id) {         return entityManager.find(Course.class, id); // Find by Primary Key     }      public void deleteById(long id) {         Course course = entityManager.find(Course.class, id);         entityManager.remove(course); // Deletes entity     } }  • merge() → Insert or Update entity • find() → Retrieve by Primary Key • remove() → Delete entity </pre> |
| Running via CommandLineRunner      | <pre> @Component public class CourseCommandLineRunner implements CommandLineRunner {      @Autowired     private CourseJpaRepository repository;      @Override     public void run(String... args) throws Exception { </pre>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                   |

```

repository.insert(new Course(1, "Learn AWS JPA", "Saurav"));
repository.insert(new Course(2, "Learn Azure JPA", "Saurav"));
repository.insert(new Course(3, "Learn DevOps JPA", "Saurav"));
repository.deleteById(1);
System.out.println(repository.findById(2));
System.out.println(repository.findById(3));
}
}

```

To see generated SQL queries, add this to application.properties: spring.jpa.show-sql=true

## Spring Data JPA

- With **JDBC**: A lot of SQL + Java code.
- With **Spring JDBC**: Less Java, still lots of SQL.
- With **JPA**: SQL abstracted, but still need to use EntityManager.
- Spring Data JPA** simplifies JPA further — no EntityManager required.

|       |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                 |
|-------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Setup | <p><b>1. Entity Class</b></p> <ul style="list-style-type: none"> <li>Already have Course.java annotated with @Entity and @Id.</li> </ul> <p><b>2. Repository Interface</b></p> <ul style="list-style-type: none"> <li>Create an interface:</li> <pre>public interface CourseRepository extends JpaRepository&lt;Course, Long&gt; {}</pre> <li>JpaRepository&lt;T, ID&gt; is part of Spring Data JPA.</li> <li>No need for manual implementation or EntityManager.</li> </ul> <pre>public interface CourseSpringDataJpaRepository extends JpaRepository&lt;Course, Long&gt; {     List&lt;Course&gt; findByAuthor(String author);     List&lt;Course&gt; findByName(String name); }</pre> <p><b>3. Using the Repository</b></p> <ul style="list-style-type: none"> <li>Autowire the repository in CommandLineRunner or any service class.</li> <li>Call methods like: <ul style="list-style-type: none"> <li>save() – insert or update.</li> <li>findById(id), deleteById(id), findAll(), count(), existsById(id) — all built-in.</li> </ul> </li> </ul> <pre>@Component public class CourseCommandLineRunner implements CommandLineRunner{      // @Autowired     // private CourseJdbcRepository repository;      // @Autowired     // private CourseJpaRepository repository;      @Autowired     private CourseSpringDataJpaRepository repository;      @Override     public void run(String... args) throws Exception {         repository.save(new Course(1, "Learn AWS Jpa!", "in28minutes"));         repository.save(new Course(2, "Learn Azure Jpa!", "in28minutes"));         repository.save(new Course(3, "Learn DevOps Jpa!", "in28minutes"));          repository.deleteById(1);          System.out.println(repository.findById(2));         System.out.println(repository.findById(3));          System.out.println(repository.findAll()); //Internally fires: SELECT * FROM course.         System.out.println(repository.count()); //Internally fires: SELECT COUNT(*) FROM course.          System.out.println(repository.findByAuthor("in28minutes"));         System.out.println(repository.findByAuthor(""));          System.out.println(repository.findByName("Learn AWS Jpa!"));         System.out.println(repository.findByName("Learn DevOps Jpa!"));     } }</pre> <p><b>Benefits</b></p> <ul style="list-style-type: none"> <li><b>Zero boilerplate</b>: No SQL, no EntityManager, minimal code.</li> <li><b>Auto-implementation</b>: Just define an interface and you're good to go.</li> <li><b>Rich API</b>: JpaRepository provides dozens of ready-to-use methods.</li> </ul> |
|-------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|

• **Productivity boost:** Saves development time and reduces errors.

# MVC

12 May 2025 07:49

|                                           |                                                                                                                                                                                                                                                                                                                                                                              |
|-------------------------------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Servlets                                  | <ul style="list-style-type: none"><li>• <b>Servlets</b> are Java classes that handle incoming HTTP requests.</li><li>• They act as <b>controllers</b> in the MVC architecture.</li><li>• Typically used to:<ul style="list-style-type: none"><li>• Receive request parameters</li><li>• Process logic (like fetching DB data)</li><li>• Return responses</li></ul></li></ul> |
| JSP (JavaServer Pages)                    | <ul style="list-style-type: none"><li>• Used for <b>view generation</b> — creating HTML pages with embedded Java.</li><li>• You write <b>HTML first</b>, and sprinkle <b>Java code</b> (using scriptlets) inside.</li><li>• JSP pages are <b>compiled into servlets</b> behind the scenes by Tomcat.</li></ul>                                                               |
| Model                                     | POJO                                                                                                                                                                                                                                                                                                                                                                         |
| Flow                                      | Controller->Model->View                                                                                                                                                                                                                                                                                                                                                      |
| Why Use JSP Over Writing HTML in Servlets | <ul style="list-style-type: none"><li>• Mixing HTML and Java in servlets becomes <b>messy and hard to maintain</b>.</li><li>• JSP allows a <b>cleaner separation</b>:<ul style="list-style-type: none"><li>• UI (HTML) and logic (Java) are separate</li><li>• Easier to manage in teams (designers + developers)</li></ul></li></ul>                                        |
| Tomcat                                    | <ul style="list-style-type: none"><li>• Tomcat is both a <b>Web Server</b> and a <b>Servlet Container</b></li><li>• JSP pages are internally <b>converted to Servlets</b></li><li>• You write simple JSP → Tomcat converts it → executes as servlet</li></ul>                                                                                                                |
|                                           |                                                                                                                                                                                                                                                                                                                                                                              |

# Lessons Learned in S/W Testing

02 July 2025 17:44

|                                                                   |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                      |
|-------------------------------------------------------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| You are the headlights of the project                             | Testing is done to find information. Critical decisions about the project or the product are made on the basis of that information.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                  |
| You discover things that will “bug” someone whose opinion matters | Your group’s mission includes (or should include) informing clients about anything that threatens the value of the product, according to your clients’ definition(s) of value. If you can show that the product will not be valued even if it works as intended, it’s your duty to report your concerns. If your clients choose to dismiss the report, that’s their prerogative.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                     |
| Find important bugs fast                                          | <ul style="list-style-type: none"><li>• Test things that are changed before things that are the same. Fixes and updates mean fresh risk.</li><li>• Test core functions before contributing functions. Test the critical and the popular things that the product does. Test the functions that make the product what it is.</li><li>• Test capability before reliability. Test whether each function can work at all before going deep into the examination of how any one function performs under many different conditions.</li><li>• Test common situations, common threats, high impact problems and most wanted areas before esoteric situations.</li></ul> <p>You will also find important problems sooner if you know more about the product, the software and hardware it must interact with, and the people who will use it. Study these well.</p>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                           |
| Run with the programmers                                          | Supporting the programmers is probably a key part of your mission. When you test the things that the programmers are building right now, or have most recently built, your feedback will help them work more efficiently. When they deliver it, you test it. When they make a change, you test the change. Aim for the shortest, quickest feedback loop you can.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                     |
| Question everything, but not necessarily out loud                 | <p>It’s possible to test without questioning, but it’s not possible to test well. Questions are fundamental to your role on the project. If you do not question, your testing will be aimless and mechanical. However, explicit questions can be provocative. They often put people on the defensive.</p> <p>Questions can be like strong medicine—best asked in low doses or taken with a meal (i.e., other kinds of communication). Fortunately, the value of questions is not limited to those that are asked out loud. Any question that occurs to you may help provoke your own thoughts in directions that lead to vital insights. If you ever find yourself testing and realize that you have no questions about the product, take a break.</p>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                               |
| You focus on failure, so your clients can focus on success        | Testers focus on failure because it improves their chances of finding it. Look for key problems in the product with all your creativity and skill. If you don’t find them, they can’t be fixed, and then the users may find them for you.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                            |
| You will not find all the bugs                                    | It’s your job to find and report significant bugs. But you won’t find all of them. To find all of them, you’d have to look everywhere there could be a bug, you’d have to look there under every different situation that could arise, and you’d need a foolproof way of recognizing every different kind of bug when it occurred. If you think you can do that, you have either a very simple product or a very limited imagination. You have to make choices about how to spend your time, knowing and accepting that you can’t do everything.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                     |
| Beware of testing “completely”                                    | Saying “it will take me five days to test that” can be interpreted to mean that you think you will have completely tested that part of the product in five calendar days. And that might be taken to mean you will find every bug in five days.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                      |
| You don’t assure quality by testing                               | Quality comes from the people who build the product                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                  |
| Never be the gatekeeper!                                          | <p>The problem is that when testers control the release, they also must bear the full responsibility for the quality of the product. The rest of the team will relax a little bit, or maybe a lot. If any bug sneaks by the testers and out the gate, the rest of the team can (and will) shrug and blame the testers. After all, why did the testers ship such a buggy product? On the other hand, if the testers delay the release, they bear intense scrutiny and pressure for being such quality fanatics.</p> <p>Ultimately, it’s the people who control the project who are best equipped to bear the responsibility to release the product. However, most highly effective projects we’ve seen use some kind of consensus approach. If you’re ever given authority to control the release, we recommend immediately insisting on sharing that authority with the other roles on the team.</p>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                 |
| Beware of the not-my-job theory of Testing                        | <p>Testing is so complex and so interconnected with other project activities that you may be tempted to get a better handle on it by adopting a narrow view of the testing mission. Some testers feel that their mission is simply to find variances between the product and the specifications. Anything beyond that—such as usability problems, requirements problems, data quality, and supportability concerns—are “not my job.” We urge you to take a more expansive view. All other things being equal, your mission should be to inform the team, to the best of your ability, about any problems that could adversely impact the value of the product. For this reason, excellent test teams include a diverse group of people who collectively understand the whole equation of the product: how it will be designed, manufactured, marketed, sold, used, serviced, and upgraded.</p> <p>Another temptation to say “not my job” comes when you’re placed in a difficult testing situation. Your colleagues on the programming side may write crummy specs. They may deliver their code so late that you don’t have time for a reasonable test process. They may claim that an important problem you’ve found is actually a figment of your imagination. It’s tempting to refuse to test under those circumstances. You could say it’s not your job to interpret ambiguous specifications or to test something in such a rush. In severe cases, that may well be the right thing to do, but consider first whether your expectations are realistic and consider whether there’s an alternative way to get what you need. If you adopt the philosophy that it’s your job to make a reasonable effort to adapt and improvise, the programmers are more likely to consider you a boon instead of a burden. That, in turn, encourages them to make it their job to help you.</p> |
| Beware of becoming a                                              | Sometimes you get tired of finding problems, and you wonder whether it might be better to prevent problems. Maybe                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                    |

|                                                                                |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                              |
|--------------------------------------------------------------------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| process improvement group                                                      | there would be fewer problems if the programmers did their work more carefully. This makes good sense, as far as it goes. But then again, it also makes sense to helpfully tell the ones you love how to live their lives better. If you've tried that, you know that good advice is not always sensibly received. Sensibility is not the issue; feelings are. No matter what else it's about, process improvement is always about feelings.<br>Even if you have strong management support for testing to drive a quality improvement initiative, there are many ways that the rest of the team can simultaneously foil your efforts and make you look incompetent. Yes, you can successfully participate in process improvement efforts and be successful if it's a whole team effort, but we urge you to resist any temptation to "elevate" the test team into a process criticism society. That way lies madness.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                         |
| Don't expect anyone to understand testing, or what you need to do it well.     | It's up to you to let your clients know what you need in order to do your job effectively.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                   |
| Testing is applied epistemology                                                | Epistemology is the study of how you know what you know. It's the study of evidence and reasoning. It establishes the foundations of scientific practice. We use the term more broadly than it's classically defined so that we can take advantage of much of the recent work in critical thinking. Applied to software testing, epistemology asks questions like the following: <ul style="list-style-type: none"><li>• How do you know the software is good enough?</li><li>• How would you know if it wasn't good enough?</li><li>• How do you know you've tested enough?</li></ul>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                       |
| Testing is in your head                                                        | The difference between excellent testing and mediocre testing is how you think: your test design choices, your ability to interpret what you observe, and your ability to tell a compelling story about it.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                  |
| Testing requires inference, not just comparison of output to expected results. |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                              |
| Good testers think technically, creatively, critically, and practically.       | <ul style="list-style-type: none"><li>• <b>Technical thinking:</b> The ability to model technology and understand causes and effects. This includes things like knowledge of relevant technical facts and the ability to use tools and predict the behavior of systems.</li><li>• <b>Creative thinking:</b> The ability to generate ideas and see possibilities. You will test only in ways that you can imagine testing. You will look only for problems that you imagine can exist.</li><li>• <b>Critical thinking:</b> The ability to evaluate ideas and make inferences. This includes the ability to detect and eliminate errors from your thinking, to relate product observations to quality criteria, and to build a compelling case for a particular belief or suggested course of action.</li><li>• <b>Practical thinking:</b> The ability to put ideas into practice. This ability includes such skills as applying test tools and making test techniques and effort fit within the scope of the project.</li></ul> <p>Overall, thinking like a tester leads you to believe that things may not be as they seem. However things are, they could be different. We find that when the test process fails in the most damaging ways, the root cause is most likely to be tunnel vision. In other words, it's not that we ran 10,000 tests and should have run 10,001. It's that we failed to imagine an entire category of test; testing we wouldn't have performed even if we had twice the time and resources.</p> |
|                                                                                |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                              |

# General Systems Thinking

02 July 2025 19:51

|           |                                                                                                                                                                                                                                                                                                                                                                                                                                                  |
|-----------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Chapter 1 | Jerry Weinberg is introducing us to a <b>big problem in science and thinking</b> : How do we understand complicated things—like the brain, a society, or the economy—when we can't see or control all the little parts? He starts by pointing out that <b>our usual ways of thinking</b> (especially from physics) work well only for <b>simple or predictable systems</b> . But most of life and real-world problems are <b>not like that</b> . |
|-----------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|

## From Machines to Systems

Back in Newton's time, people thought of the universe like a **giant machine**:

- You break it into parts,
- Understand each part,
- Put it back together,
- And—boom—you understand the whole thing!

That worked for things like planets or simple machines. Newton was a genius at doing this. But...

The real world doesn't always act like a perfect machine.

## What's the Problem with Today's Science?

Jerry says modern science often focuses too much on **numbers and equations**, and forgets about **understanding the causes and logic** behind things.

- People are obsessed with **quantifying** (measuring stuff).
- But sometimes, understanding **why** something happens (causality) is more important than just seeing a pattern (correlation).
- Equations and stats are tools—but if we don't think clearly first, they can **confuse more than they help**.

Smart science starts with clear thinking, not fancy math.

## Why the Old Tricks Don't Always Work

In old-style physics (like Newton's), people could simplify problems by:

- Looking at just a few objects (like planets),
- Ignoring smaller effects,
- Assuming perfect parts that don't affect each other.

But in **complex systems** (like people, ecosystems, or even weather), that doesn't work:

- There are **too many parts** (like molecules in air),
- Everything **affects everything else**,
- Small things can have **huge impacts**.

So Jerry says: we need **new ways of thinking**.

## Enter Systems Thinking

Instead of treating everything like a machine, systems thinking says:

- **Look at the whole picture**.
- **Focus on how parts interact**.
- Understand **patterns, relationships, and behavior** of the system, not just individual parts.

Example:

- You can't understand a school just by looking at desks and chairs.
- You have to understand **students, teachers, rules, feelings, learning, etc.**
- The **system** is more than the sum of its parts.

## The Human Brain and Limits of Thinking

Jerry also talks about the **Square Law of Computation**:

- The more parts in a system, the **much harder** it is to compute.
- Even supercomputers struggle with huge systems.
- And our **brains are limited too**—so we need simplifications and models to survive mentally.  
That's what Newton did: he made smart simplifications.

But today's problems need new simplifications—because the world is more connected and complicated.

## Why Chapter 1 Matters

This first chapter sets the stage for the rest of the book. It says:

- **Old ways of thinking** (machine-based, number-driven, over-simplified) **aren't enough**.
- We need **systems thinking**—a way to understand complexity by studying **interactions, context, and structure**.
- It's not about avoiding math—but about using it **wisely**, only when it helps us think better.

## Key Takeaways:

1. **Don't mistake numbers for understanding.**
2. **Real-world problems are messy—look at the big picture.**
3. **Simplify smartly**, but don't ignore important parts just to make things easier.
4. **Think in systems**—see how things are connected and influence each other.
5. **Your brain has limits**, but if you model things well, you can solve complex problems better.

Chapter 2

# Chapter 2: The Nature of Systems

## What Is a System?

Jerry begins by asking a big question:

“What exactly is a *system*?”

It sounds simple, but the word *system* is used in so many ways—in science, schools, businesses, and daily life—that it can get blurry. Weinberg wants to clear that up.

## A System Is Not Just a Collection

A **system** isn’t just a group of parts thrown together. For something to be called a system:

- The parts must be **connected** in a meaningful way.
- The system must show **behavior** or a **purpose** that comes from the whole, not just the parts.

### Analogy:

A pile of car parts is *not* a system.

But a car, with all parts connected and working together to drive, **is** a system.

## Key Properties of Systems:

1. **Interaction** – The parts of a system affect each other.  
Think: If one part changes, it affects the whole system.
2. **Emergence** – A system shows behaviors that **none of the parts alone can do**.  
A single ant can’t build a colony; many ants together can.
3. **Boundaries** – A system has a limit: what’s inside (part of the system) and what’s outside (the environment).  
Your body is a system. The air around you is your environment.
4. **Purpose or Function** – Systems usually do something, even if it’s just reacting to their environment.

## Types of Systems:

Weinberg introduces **two big types** of systems:

### 1. Linear (Machine-Like) Systems

- Predictable
- Simple cause-effect
- You can break them into parts and solve each part separately
- Example: A clock, a factory machine

### 2. Non-Linear (Complex) Systems

- Hard to predict
- The output is **not** directly proportional to the input
- Tiny changes can cause big effects
- Example: A school, an ecosystem, the stock market

Most real-world systems are **non-linear**, but we often treat them like linear ones—which leads to trouble.

## Why This Matters

Weinberg warns: if you treat **complex systems** as if they’re just **simple machines**, you’ll misunderstand or break them.

Example: If a teacher tries to “fix” a school by just changing rules like flipping switches, they might make things worse—because schools are **social systems**, not machines.

So the general systems thinker learns to:

- Identify what kind of system they’re dealing with,
- Understand how the parts interact,
- Think about **boundaries, feedback, and behavior**, not just parts.

## Feedback Loops (Preview)

This chapter sets up ideas for later ones, like **feedback loops**—how systems react to changes and adjust themselves. That’s key for understanding how systems grow, adapt, or break down.

## Final Message of Chapter 2:

“A system is more than just the sum of its parts—it’s also how the parts fit and work together.”

Understanding systems means:

- Seeing the **relationships**, not just the pieces.
- Respecting **complexity**, not over-simplifying.
- Using **models** and **boundaries** wisely to make sense of the mess.

# BBST Foundations

29 June 2025 09:13

Software Testing is an empirical, technical investigation conducted to provide stakeholders with information about the quality of the product or service under test. Testing software involves investigating a product under tight constraints.

- Knowledge and skills are important to testing practitioners
- Context-driven
  - Diverse contexts call for diverse practices.
  - We don't teach "best practices." Instead, we teach practices that are useful in the appropriate circumstances.

|         |                                                                                                                                           |
|---------|-------------------------------------------------------------------------------------------------------------------------------------------|
| Testing | An investigation to discover information about the quality of a product. Tests are designed and run to gain information about the product |
| Quality | Value to some person. It's subjective and context-dependent.                                                                              |
| Bug     | Anything that threatens the value of the product to someone e.g. coding error, design error etc                                           |
|         |                                                                                                                                           |

# Performance Testing

31 March 2025 08:45

|                              |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                            |
|------------------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| What is Performance Testing  | Evaluating a software application's speed, response time, stability, reliability, scalability, and resource usage under specific workloads. Its purpose is to identify and eliminate performance bottlenecks before launch.<br><br>Key Factors:<br><b>Speed:</b> Application's responsiveness.<br><b>Scalability:</b> Maximum user load capacity.<br><b>Stability:</b> Performance under varying workloads.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                |
| Types of Performance Testing | <b>Load Testing</b> Evaluates performance(Throughput and RT) under normal and peak usage. Determines maximum load capacity.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                |
|                              | <b>Stress Testing</b> Finds the system's breaking point by gradually increasing load. Assesses recovery capabilities and stability. Response time is the key parameter                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                     |
|                              | <b>Volume Testing</b> Tests performance with large data volumes. Ensures handling of growing databases.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                    |
|                              | <b>Capacity Testing</b> Checks ability to handle future user loads. Identifies required resources.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                         |
|                              | <b>Reliability Testing</b> Verifies recovery from failures. Measures time to return to normal state.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                       |
|                              | <b>Scalability Testing</b> Ensures ability to scale up with increased user load.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                           |
|                              | <b>Endurance Testing/Soak</b> Expected load for long period of time                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                        |
|                              | <b>Spike Testing</b> Reaction to sudden large spikes of load                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                               |
| Key Concepts                 | <p><b>1. Virtual Users</b></p> <ul style="list-style-type: none"> <li>○ Simulate real users during testing.</li> <li>○ Navigate the system, send requests, and collect data.</li> <li>○ Generated by performance testing tools to replicate actual user behavior.</li> </ul> <p><b>2. Bottlenecks</b></p> <ul style="list-style-type: none"> <li>○ A point where system performance is restricted.</li> <li>○ Example: A road narrows, causing congestion.</li> <li>○ Common bottlenecks: CPU, memory, network bandwidth.</li> </ul> <p><b>3. Scalability</b></p> <ul style="list-style-type: none"> <li>○ The ability to <b>scale up</b> (add resources) or <b>scale down</b> (remove resources) based on demand.</li> <li>○ Helps handle varying loads efficiently.</li> </ul> <p><b>4. Latency</b></p> <ul style="list-style-type: none"> <li>○ Time taken for data to travel between client and server.</li> <li>○ Example: <ul style="list-style-type: none"> <li>■ Request latency = 200ms</li> <li>■ Response latency = 300ms</li> <li>■ <b>Total (Round-trip) latency = 500ms</b></li> </ul> </li> </ul> <p><b>5. Throughput</b></p> <ul style="list-style-type: none"> <li>○ Measures system efficiency.</li> <li>○ Number of work requests processed in a given time.</li> <li>○ Key metric in performance testing.</li> </ul> <p><b>6. Response Time</b></p> <ul style="list-style-type: none"> <li>○ Time taken to complete a request, including: <ul style="list-style-type: none"> <li>■ Request transmission time</li> <li>■ Server processing time</li> <li>■ Response transmission time</li> </ul> </li> </ul> <p><b>7. Saturation</b></p> <ul style="list-style-type: none"> <li>○ Occurs when a resource reaches its maximum capacity.</li> <li>○ Beyond this point, performance degrades.</li> <li>○ Example: 100% CPU or memory usage.</li> </ul> <p><b>8. CPU &amp; Memory Utilization</b></p> <ul style="list-style-type: none"> <li>○ <b>CPU Utilization:</b> Percentage of CPU used during processing.</li> <li>○ <b>Memory Utilization:</b> Percentage of memory consumed during execution.</li> </ul> <p><b>9. Concurrent vs. Simultaneous Users</b></p> <ul style="list-style-type: none"> <li>○ <b>Concurrent Users:</b> Multiple users performing different tasks at the same time.</li> <li>○ <b>Simultaneous Users:</b> Multiple users performing the <b>same task</b> at the same time.</li> </ul> <p><b>10. Think Time</b></p> <ul style="list-style-type: none"> <li>• The time a user takes before performing the next action.</li> <li>• Important to simulate real-world scenarios in performance tests.</li> </ul> <p><b>11. Peak Time</b></p> <ul style="list-style-type: none"> <li>• The busiest time for the system when user requests are at their highest.</li> <li>• Example: <b>Theme parks on weekends and holidays.</b></li> </ul> <p><b>12. Peak Load</b></p> <ul style="list-style-type: none"> <li>• The highest number of users the system is expected to handle at peak time.</li> <li>• Example: <b>Max visitors at a theme park during the evening.</b></li> </ul> |
| Important                    | <ul style="list-style-type: none"> <li>• <b>Processor Usage:</b> Time spent on active threads.</li> </ul>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                  |

|                             |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                             |
|-----------------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Performance Metrics         | <ul style="list-style-type: none"> <li>• <b>Memory Usage:</b> RAM used by app.</li> <li>• <b>Disk Time:</b> Time spent on read/write.</li> <li>• <b>Bandwidth:</b> Network usage (bits/sec).</li> <li>• <b>Private Bytes:</b> App's exclusive memory allocation.</li> <li>• <b>Committed Memory:</b> Total virtual memory in use.</li> <li>• <b>Response Time:</b> Time between request &amp; first byte of response.</li> <li>• <b>Throughput:</b> Requests processed/sec.</li> <li>• <b>Connection Pooling:</b> Requests handled via pooled connections.</li> <li>• <b>Thread Count.</b></li> </ul>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                       |
| Performance Testing Process | <p><b>1. Requirement Gathering &amp; Analysis</b></p> <ul style="list-style-type: none"> <li>◦ Identify performance goals (response time, throughput, resource utilization).</li> <li>◦ Understand application architecture, user load, and usage patterns.</li> <li>◦ Crucial for troubleshooting performance issues.</li> </ul> <p><b>2. Planning &amp; Design</b></p> <ul style="list-style-type: none"> <li>◦ Create a <b>Performance Test Plan</b> outlining: <ul style="list-style-type: none"> <li>▪ Scope &amp; objectives</li> <li>▪ Testing approach</li> <li>▪ Environment setup details</li> </ul> </li> <li>◦ Define test scenarios and workload modeling based on user behavior.</li> </ul> <p><b>3. Environment Setup</b></p> <ul style="list-style-type: none"> <li>◦ Ensure the test environment closely mirrors the production environment.</li> <li>◦ Set up required hardware, software, and network configurations for realistic results.</li> </ul> <p><b>4. Tool Selection</b></p> <ul style="list-style-type: none"> <li>◦ Choose tools based on: <ul style="list-style-type: none"> <li>▪ Application under test</li> <li>▪ Open-source vs. paid options</li> <li>▪ Team skillset</li> </ul> </li> </ul> <p><b>5. Test Script Development</b></p> <ul style="list-style-type: none"> <li>◦ Simulate user actions with test scripts.</li> <li>◦ Optimize scripts with <b>parameterization &amp; correlation</b> for accuracy.</li> </ul> <p><b>6. Test Execution</b></p> <ul style="list-style-type: none"> <li>◦ Gradually increase the load while monitoring system behavior.</li> <li>◦ Track key metrics like <b>response time, throughput, CPU/memory usage, disk I/O</b>.</li> </ul> <p><b>7. Monitoring &amp; Analysis</b></p> <ul style="list-style-type: none"> <li>◦ Analyze test results to identify bottlenecks, slow response times, and resource constraints.</li> </ul> <p><b>8. Tuning &amp; Optimization</b></p> <ul style="list-style-type: none"> <li>◦ Work with developers to resolve issues: <ul style="list-style-type: none"> <li>▪ <b>Code-related?</b> Optimize the code.</li> <li>▪ <b>Database issue?</b> Optimize queries.</li> <li>▪ <b>Infrastructure issue?</b> Coordinate with admin teams.</li> </ul> </li> </ul> <p><b>9. Retesting</b></p> <ul style="list-style-type: none"> <li>◦ Re-run tests to verify that fixes work.</li> <li>◦ Ensure no new performance issues have been introduced.</li> </ul> <p><b>10. Reporting</b></p> <ul style="list-style-type: none"> <li>• Document test results, including metrics, observations, and identified issues.</li> <li>• Provide a detailed report to stakeholders, highlighting areas for improvement.</li> </ul> <p><b>11. Continuous Monitoring</b></p> <ul style="list-style-type: none"> <li>• Monitor production performance regularly.</li> <li>• Adjust tests and scripts based on changing user behavior and system demands.</li> </ul> |

|          |              |                                                                                                                                                                                                                                                                                                                                                                                                                                                                  |
|----------|--------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Elements | Definition   | <ul style="list-style-type: none"> <li>• <b>Elements</b> = Building blocks or components in JMeter.</li> <li>• Each element serves a <b>specific purpose</b>.</li> <li>• Examples: <ul style="list-style-type: none"> <li>▪ <b>Thread Group</b></li> <li>▪ <b>Samplers</b></li> <li>▪ <b>Listeners</b></li> <li>▪ <b>Configuration Elements</b></li> <li>▪ (Others like Assertions, Timers, etc.)</li> </ul> </li> </ul>                                         |
|          | Thread Group | <ul style="list-style-type: none"> <li>• Represents a <b>group of users (threads)</b>.</li> <li>• Each thread simulates <b>one user request</b>.</li> <li>• You can configure: <ul style="list-style-type: none"> <li>◦ Number of threads (users)</li> <li>◦ Ramp-up period</li> <li>◦ Loop count (iterations)</li> </ul> </li> </ul> <p>Example:<br/>If you configure 100 threads, JMeter simulates 100 users hitting the server at once or as per ramp-up.</p> |
|          | Samplers     | <ul style="list-style-type: none"> <li>• Define <b>what kind of requests</b> are sent by users.</li> <li>• Examples of Sampler types: <ul style="list-style-type: none"> <li>◦ HTTP Request</li> <li>◦ FTP Request</li> <li>◦ JDBC Request</li> <li>◦ SOAP/XML-RPC Request</li> <li>◦ SMTP Sampler</li> </ul> </li> </ul>                                                                                                                                        |

|                        |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                        |
|------------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
|                        | <ul style="list-style-type: none"> <li>◦ BSF Sampler</li> </ul> <p>Each sampler represents an <b>action a user performs</b> (like visiting a web page or making an API call).</p>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                      |
| Listeners              | <ul style="list-style-type: none"> <li>• Used to <b>display test results</b>.</li> <li>• Different formats available: <ul style="list-style-type: none"> <li>◦ <b>Graph Results</b> – shows response times graphically</li> <li>◦ <b>View Results Tree</b> – HTML-based request &amp; response</li> <li>◦ <b>Summary Report</b> – tabular format</li> <li>◦ <b>Log Files</b> – plain text format</li> </ul> </li> </ul> <p>Helps you <b>analyze performance metrics</b> visually or in logs.</p>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                       |
| Configuration Elements | <ul style="list-style-type: none"> <li>• Set <b>default values and variables</b> used by samplers.</li> <li>• Examples: <ul style="list-style-type: none"> <li>◦ <b>CSV Data Set Config</b> – for data-driven tests</li> <li>◦ <b>HTTP Request Defaults</b> – base URL and parameters</li> <li>◦ <b>HTTP Cookie Manager</b> – manage cookies</li> <li>◦ <b>Login Config Element</b> – authentication setup</li> </ul> </li> </ul> <p>Reduces duplication and makes the test <b>more dynamic and reusable</b>.</p>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                      |
| Assertions             | <ul style="list-style-type: none"> <li>• <b>Response Assertion:</b> Used in test scripts to validate a pattern in the response body, header, code, message etc. There are different pattern matching rules to validate the response.</li> <li>• <b>Size Assertion:</b> Used to validate the size of the response with a specified value in bytes.</li> <li>• <b>Duration Assertion:</b> Used to validate that the sampler request gets processed within a specified amount of time.</li> <li>• <b>HTML Assertion :</b> The HTML assertion is used to check the HTML syntax of the response.</li> <li>• <b>XML Assertion:</b> The XML assertion is used to validate that the response follows a valid XML syntax.</li> <li>• <b>XML Schema Assertion:</b> The XML Schema Assertion is used to validate the response against a specified XML schema.</li> <li>• <b>XPath Assertion:</b> The XPath assertion is used to validate the response using XPath expressions.</li> <li>• <b>JSON Assertion:</b> The JSON assertion is used to validate the response using JSON expressions.</li> </ul>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                           |
| Listeners              | <p><b>What are Listeners?</b></p> <ul style="list-style-type: none"> <li>• Provide access to info JMeter gathers during test runs.</li> <li>• Show results in different formats: <ul style="list-style-type: none"> <li>◦ Tree</li> <li>◦ Table</li> <li>◦ Graph</li> <li>◦ Reports</li> </ul> </li> <li>• Can also save results to log files.</li> </ul> <p><b>Types of Listeners (Examples):</b></p> <ul style="list-style-type: none"> <li>• <b>View Results in Table:</b> <ul style="list-style-type: none"> <li>◦ Columns: Sample Start Time, Thread Name, Label, Sample Time, Status, Bytes, Sent Bytes, Latency, Connection Time.</li> <li>◦ <b>Latency:</b> Time between request sent and <i>start</i> of response received (initial acknowledgement).</li> <li>◦ <b>Response Time (Sample Time/Load Time/Elapsed Time):</b> Time between request sent and <i>full</i> response received.</li> <li>◦ <b>Formula:</b> Response Time = Latency Time + Processing Time (includes throughput).</li> <li>◦ <b>Throughput:</b> Number of requests processed by the server per second.</li> <li>◦ <b>Scope:</b> Listener at Thread Group level = results <i>only</i> for that group. Listener at Test Plan level = <i>combined</i> results.</li> </ul> </li> <li>• <b>View Results Tree:</b> <ul style="list-style-type: none"> <li>◦ Shows details for <i>each</i> request.</li> <li>◦ Tabs: <ul style="list-style-type: none"> <li>▪ <b>Sample Result:</b> Thread Name, Start Time, Load Time, Connect Time, Latency, etc.</li> <li>▪ <b>Request:</b> What was sent to the server (including headers and body).</li> <li>▪ <b>Response Data:</b> Server's response (e.g., HTML).</li> </ul> </li> </ul> </li> <li>• <b>Graph Results:</b> <ul style="list-style-type: none"> <li>◦ Visual representation of performance over time.</li> <li>◦ Shows: Samples, Latest Sample, Average, Median, Throughput, Deviation.</li> <li>◦ Good for real-time monitoring.</li> </ul> </li> <li>• <b>Aggregate Report:</b> <ul style="list-style-type: none"> <li>◦ Tabular summary of results.</li> <li>◦ Key metrics: <ul style="list-style-type: none"> <li>▪ <b>Samples:</b> Total number of requests.</li> <li>▪ <b>Average:</b> Average response time.</li> <li>▪ <b>Median:</b> 50% of requests completed within this time.</li> <li>▪ <b>90%/95%/99% Line:</b> Percentage of requests completed within these times.</li> <li>▪ <b>Min/Max:</b> Minimum and maximum response times.</li> <li>▪ <b>Error %:</b> Percentage of failed requests.</li> <li>▪ <b>Throughput:</b> Requests per second.</li> <li>▪ <b>Received/Sent KB/sec:</b> Data transfer rates.</li> </ul> </li> </ul> </li> <li>• <b>Aggregate Graph:</b> <ul style="list-style-type: none"> <li>◦ Graphical representation of Aggregate Report data.</li> <li>◦ Can select which metrics to display (Average, Median, Min, Max, etc.) in Settings.</li> </ul> </li> <li>• <b>Summary Report:</b> <ul style="list-style-type: none"> <li>◦ Similar to Aggregate Report, provides a summary.</li> <li>◦ Includes Standard Deviation.</li> <li>◦ Shows totals if multiple requests are present.</li> </ul> </li> <li>• <b>Simple Data Writer:</b> <ul style="list-style-type: none"> <li>◦ Writes results directly to a file (CSV, TXT, log).</li> <li>◦ Configurable: Choose which data to include.</li> </ul> </li> </ul> |

|                             |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                    |                     |                                                                                                                                                                                                                                                                                                                                                                                             |                        |                                                                                                                                                                                                             |                       |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                    |                             |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                          |                     |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                  |
|-----------------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|---------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|------------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-----------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-----------------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|---------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
|                             | <ul style="list-style-type: none"> <li>○ <b>Key Advantage:</b> Doesn't populate UI during the test, saves memory and improves performance, especially for large loads.</li> <li>○ Can overwrite or append to existing files.</li> <li>○ <b>Recommendation for Real-time Heavy Load:</b> Use this instead of UI-based listeners.</li> </ul> <p><b>Important Considerations:</b></p> <ul style="list-style-type: none"> <li>• UI-based listeners (especially graphs and tree view) can consume significant resources (CPU, memory) during heavy load tests. This can impact the accuracy of the results.</li> <li>• For large-scale testing, <b>disable</b> resource-intensive UI listeners after initial setup and verification.</li> <li>• <b>Simple Data Writer</b> is the most efficient listener for production-like load tests.</li> </ul>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                     |                     |                                                                                                                                                                                                                                                                                                                                                                                             |                        |                                                                                                                                                                                                             |                       |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                    |                             |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                          |                     |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                  |
| Timers                      | <table border="1"> <tr> <td><b>Introduction</b></td><td> <ul style="list-style-type: none"> <li>• By default, JMeter sends requests immediately without any delay between samplers.</li> <li>• This can overload the server and doesn't simulate realistic user behavior.</li> <li>• Timers are JMeter elements used to introduce delays between each sampler or request.</li> <li>• They help simulate real-world user traffic patterns.</li> </ul> </td></tr> <tr> <td><b>Types of Timers</b></td><td> <ul style="list-style-type: none"> <li>• Constant Timer</li> <li>• Uniform Random Timer</li> <li>• Gaussian Random Timer</li> <li>• Beanshell Timer</li> <li>• BSF Timer</li> <li>• JSR223 Timer</li> </ul> </td></tr> <tr> <td><b>Constant Timer</b></td><td> <ul style="list-style-type: none"> <li>• Applies a fixed delay (in milliseconds) between each request.</li> <li>• Can be added at the: <ul style="list-style-type: none"> <li>○ Thread Group level (applies to all requests within the group).</li> <li>○ Sampler (Request) level (applies only to that specific request).</li> <li>○ Test Plan level (applies globally).</li> </ul> </li> <li>• If multiple timers are applied to a single request, their delay times are added together.</li> <li>• <b>Example:</b> A Constant Timer with a value of 5000 milliseconds (5 seconds) will introduce a 5-second delay before each request it applies to.</li> </ul> </td></tr> <tr> <td><b>Uniform Random Timer</b></td><td> <ul style="list-style-type: none"> <li>• Introduces a random delay between requests within a specified range.</li> <li>• Has two key fields: <ul style="list-style-type: none"> <li>○ <b>Random Delay Maximum:</b> The upper bound for the random delay (in milliseconds).</li> <li>○ <b>Constant Delay Offset:</b> A fixed delay that is added to the random delay (in milliseconds).</li> </ul> </li> <li>• <b>Formula (Internal):</b> Approximately <math>\\$0.x * \{\text{Random Delay Max}\} + \{\text{Constant Delay Offset}\}</math> (where 'x' represents digits 0-9).</li> <li>• Allows simulating more realistic user behavior with varying delays.</li> </ul> </td></tr> <tr> <td><b>Other Timers</b></td><td> <ul style="list-style-type: none"> <li>• <b>Gaussian Random Timer:</b> Delays requests based on a Gaussian distribution (bell curve). Requires configuring a mean and standard deviation for the delay.</li> <li>• <b>Beanshell Timer:</b> Allows defining custom delay logic using Beanshell scripting. Offers more flexibility but requires scripting knowledge.</li> <li>• <b>BSF Timer:</b> Similar to Beanshell Timer but uses Bean Scripting Framework (BSF) and supports various scripting languages.</li> <li>• <b>JSR223 Timer:</b> The recommended scripting timer, supporting languages like Groovy, which often offers better performance than Beanshell.</li> </ul> </td></tr> </table> | <b>Introduction</b> | <ul style="list-style-type: none"> <li>• By default, JMeter sends requests immediately without any delay between samplers.</li> <li>• This can overload the server and doesn't simulate realistic user behavior.</li> <li>• Timers are JMeter elements used to introduce delays between each sampler or request.</li> <li>• They help simulate real-world user traffic patterns.</li> </ul> | <b>Types of Timers</b> | <ul style="list-style-type: none"> <li>• Constant Timer</li> <li>• Uniform Random Timer</li> <li>• Gaussian Random Timer</li> <li>• Beanshell Timer</li> <li>• BSF Timer</li> <li>• JSR223 Timer</li> </ul> | <b>Constant Timer</b> | <ul style="list-style-type: none"> <li>• Applies a fixed delay (in milliseconds) between each request.</li> <li>• Can be added at the: <ul style="list-style-type: none"> <li>○ Thread Group level (applies to all requests within the group).</li> <li>○ Sampler (Request) level (applies only to that specific request).</li> <li>○ Test Plan level (applies globally).</li> </ul> </li> <li>• If multiple timers are applied to a single request, their delay times are added together.</li> <li>• <b>Example:</b> A Constant Timer with a value of 5000 milliseconds (5 seconds) will introduce a 5-second delay before each request it applies to.</li> </ul> | <b>Uniform Random Timer</b> | <ul style="list-style-type: none"> <li>• Introduces a random delay between requests within a specified range.</li> <li>• Has two key fields: <ul style="list-style-type: none"> <li>○ <b>Random Delay Maximum:</b> The upper bound for the random delay (in milliseconds).</li> <li>○ <b>Constant Delay Offset:</b> A fixed delay that is added to the random delay (in milliseconds).</li> </ul> </li> <li>• <b>Formula (Internal):</b> Approximately <math>\\$0.x * \{\text{Random Delay Max}\} + \{\text{Constant Delay Offset}\}</math> (where 'x' represents digits 0-9).</li> <li>• Allows simulating more realistic user behavior with varying delays.</li> </ul> | <b>Other Timers</b> | <ul style="list-style-type: none"> <li>• <b>Gaussian Random Timer:</b> Delays requests based on a Gaussian distribution (bell curve). Requires configuring a mean and standard deviation for the delay.</li> <li>• <b>Beanshell Timer:</b> Allows defining custom delay logic using Beanshell scripting. Offers more flexibility but requires scripting knowledge.</li> <li>• <b>BSF Timer:</b> Similar to Beanshell Timer but uses Bean Scripting Framework (BSF) and supports various scripting languages.</li> <li>• <b>JSR223 Timer:</b> The recommended scripting timer, supporting languages like Groovy, which often offers better performance than Beanshell.</li> </ul> |
| <b>Introduction</b>         | <ul style="list-style-type: none"> <li>• By default, JMeter sends requests immediately without any delay between samplers.</li> <li>• This can overload the server and doesn't simulate realistic user behavior.</li> <li>• Timers are JMeter elements used to introduce delays between each sampler or request.</li> <li>• They help simulate real-world user traffic patterns.</li> </ul>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                        |                     |                                                                                                                                                                                                                                                                                                                                                                                             |                        |                                                                                                                                                                                                             |                       |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                    |                             |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                          |                     |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                  |
| <b>Types of Timers</b>      | <ul style="list-style-type: none"> <li>• Constant Timer</li> <li>• Uniform Random Timer</li> <li>• Gaussian Random Timer</li> <li>• Beanshell Timer</li> <li>• BSF Timer</li> <li>• JSR223 Timer</li> </ul>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                        |                     |                                                                                                                                                                                                                                                                                                                                                                                             |                        |                                                                                                                                                                                                             |                       |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                    |                             |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                          |                     |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                  |
| <b>Constant Timer</b>       | <ul style="list-style-type: none"> <li>• Applies a fixed delay (in milliseconds) between each request.</li> <li>• Can be added at the: <ul style="list-style-type: none"> <li>○ Thread Group level (applies to all requests within the group).</li> <li>○ Sampler (Request) level (applies only to that specific request).</li> <li>○ Test Plan level (applies globally).</li> </ul> </li> <li>• If multiple timers are applied to a single request, their delay times are added together.</li> <li>• <b>Example:</b> A Constant Timer with a value of 5000 milliseconds (5 seconds) will introduce a 5-second delay before each request it applies to.</li> </ul>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                 |                     |                                                                                                                                                                                                                                                                                                                                                                                             |                        |                                                                                                                                                                                                             |                       |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                    |                             |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                          |                     |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                  |
| <b>Uniform Random Timer</b> | <ul style="list-style-type: none"> <li>• Introduces a random delay between requests within a specified range.</li> <li>• Has two key fields: <ul style="list-style-type: none"> <li>○ <b>Random Delay Maximum:</b> The upper bound for the random delay (in milliseconds).</li> <li>○ <b>Constant Delay Offset:</b> A fixed delay that is added to the random delay (in milliseconds).</li> </ul> </li> <li>• <b>Formula (Internal):</b> Approximately <math>\\$0.x * \{\text{Random Delay Max}\} + \{\text{Constant Delay Offset}\}</math> (where 'x' represents digits 0-9).</li> <li>• Allows simulating more realistic user behavior with varying delays.</li> </ul>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                           |                     |                                                                                                                                                                                                                                                                                                                                                                                             |                        |                                                                                                                                                                                                             |                       |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                    |                             |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                          |                     |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                  |
| <b>Other Timers</b>         | <ul style="list-style-type: none"> <li>• <b>Gaussian Random Timer:</b> Delays requests based on a Gaussian distribution (bell curve). Requires configuring a mean and standard deviation for the delay.</li> <li>• <b>Beanshell Timer:</b> Allows defining custom delay logic using Beanshell scripting. Offers more flexibility but requires scripting knowledge.</li> <li>• <b>BSF Timer:</b> Similar to Beanshell Timer but uses Bean Scripting Framework (BSF) and supports various scripting languages.</li> <li>• <b>JSR223 Timer:</b> The recommended scripting timer, supporting languages like Groovy, which often offers better performance than Beanshell.</li> </ul>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                   |                     |                                                                                                                                                                                                                                                                                                                                                                                             |                        |                                                                                                                                                                                                             |                       |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                    |                             |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                          |                     |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                  |
| Logic Controllers           | <ul style="list-style-type: none"> <li>• Logic Controllers in JMeter allow you to control the order in which Samplers (requests) are processed within a thread.</li> <li>• They determine <i>when</i> and <i>how</i> requests are sent to the web server.</li> <li>• JMeter provides various types of Logic Controllers (e.g., Critical Section Controller, ForEach Controller, If Controller, Include Controller).</li> <li>• Enables a user request (Sampler) to run a specified number of times or indefinitely.</li> </ul>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                     |                     |                                                                                                                                                                                                                                                                                                                                                                                             |                        |                                                                                                                                                                                                             |                       |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                    |                             |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                          |                     |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                  |

# SQL

14 December 2023 10:00

|                               |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                  |
|-------------------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Creating a database           | <code>CREATE DATABASE mydatabase;</code>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                         |
| Using the created database    | <code>USE mydatabase;</code>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                     |
| Creating a table              | <code>CREATE TABLE employees (     employee_id INT PRIMARY KEY,     first_name VARCHAR(50) NOT NULL,     last_name VARCHAR(50) NOT NULL,     department_id INT,     salary DECIMAL(10, 2) DEFAULT 0.00,     CONSTRAINT fk_department FOREIGN KEY (department_id) REFERENCES departments(department_id) );</code>                                                                                                                                                                                                                                                                                                                                 |
| Between                       | <p><code>SELECT * FROM payment WHERE amount BETWEEN 8 AND 9;</code><br/> <code>SELECT * FROM payment WHERE amount NOT BETWEEN 8 AND 9;</code></p> <p>In SQL, the BETWEEN operator's range is inclusive. That means both the start and end values of the range are included! When using between on timestamp BETWEEN '2023-01-04' AND '2023-01-05' will result in records having timestamp less than 2023-01-04 23:59:00 i.e records of 0001 HRS of 2023-01-05 will not come up.</p> <p>Else Use<br/> <code>SELECT * FROM your_table</code><br/> <code>WHERE timestamp_column BETWEEN '2023-01-04 00:00:00' AND '2023-01-05 23:59:59';</code></p> |
| LIKE                          | <p>The LIKE operator is used in a WHERE clause to search for a specified pattern in a column.</p> <ul style="list-style-type: none"> <li>• % - Represents zero, one, or multiple characters.</li> <li>• - Represents a single character.</li> </ul> <p>ILIKE is case insensitive</p> <p><code>SELECT * FROM customer WHERE fname LIKE '%er%'</code> will also yield Jennifer</p>                                                                                                                                                                                                                                                                 |
| DISTINCT                      | <p>To find unique combinations of multiple columns, such as `department` and `job_title`:</p> <pre>SELECT DISTINCT department, job_title FROM employees;</pre> <p>- This will return unique pairs of `department` and `job_title`.</p>                                                                                                                                                                                                                                                                                                                                                                                                           |
| Inserting data into the table | <code>INSERT INTO employees (employee_id, first_name, last_name, department_id, salary)</code><br><code>VALUES (1, 'John', 'Doe', 101, 50000.00), (2, 'Jane', 'Smith', 102, 60000.00);</code>                                                                                                                                                                                                                                                                                                                                                                                                                                                    |
| Updating records              | <code>UPDATE employees SET salary = 55000.00 WHERE employee_id = 1;</code>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                       |
| Altering table structure      | <code>ALTER TABLE employees</code><br><code>ADD COLUMN hire_date DATE;</code>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                    |
| Limiting results              | <code>SELECT employee_id, first_name, last_name</code><br><code>FROM employees</code><br><code>LIMIT 5;</code>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                   |
| Deleting records              | <code>DELETE FROM employees WHERE employee_id = 2;</code>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                        |
| UNION                         | <p>The UNION operator in SQL is used to combine the results of two or more SELECT statements into a single result set. It removes duplicate rows by default. To include duplicates, you can use UNION ALL. The number of columns and their data types in the result set are determined by the first SELECT statement.</p> <pre>SELECT employee_id, first_name FROM employees UNION SELECT employee_id, last_name FROM employees;</pre>                                                                                                                                                                                                           |
| JOIN                          | <code>SELECT employees.employee_id, employees.first_name, departments.department_name</code><br><code>FROM employees</code><br><code>JOIN departments ON employees.department_id = departments.department_id;</code>                                                                                                                                                                                                                                                                                                                                                                                                                             |
| Find the third-highest salary | <code>SELECT*FROM`employee` ORDERBY`salary` DESC LIMIT 1 OFFSET2;</code>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                         |
| Order by                      | ORDER BY in SQL sorts the resulting rows in ascending (ASC)<br><p>you can substitute numbers for column names in the ORDER BY clause. The numbers correspond to the columns you specify in the SELECT clause.</p>                                                                                                                                                                                                                                                                                                                                                                                                                                |
| Group By                      | <p>When using a GROUP BY clause, any columns that are not part of an aggregate function (such as SUM, AVG, COUNT, etc.) must be included in the GROUP BY clause. This is because SQL needs to know how to group the rows when aggregating data.</p> <p><code>select customer_id, rental_id, sum(amount) from payment group by rental_id order by customer_id; //This will not work</code></p> <p><code>select customer_id, rental_id, sum(amount) from payment group by rental_id, customer_id order by customer_id; //this will work</code></p>                                                                                                 |

|                |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                       |     |                                         |       |                                           |     |                                         |     |                                          |     |                                          |     |                                          |      |                                                                                                               |       |                                                                                                                                                                                                                                                                                                                                                                                                               |       |                                                                                                                       |       |                                                                                                                             |      |                                                                                  |        |                                                  |        |                                             |       |                                            |         |                                                            |         |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                        |         |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                |    |                                                                                                      |
|----------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-----|-----------------------------------------|-------|-------------------------------------------|-----|-----------------------------------------|-----|------------------------------------------|-----|------------------------------------------|-----|------------------------------------------|------|---------------------------------------------------------------------------------------------------------------|-------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-------|-----------------------------------------------------------------------------------------------------------------------|-------|-----------------------------------------------------------------------------------------------------------------------------|------|----------------------------------------------------------------------------------|--------|--------------------------------------------------|--------|---------------------------------------------|-------|--------------------------------------------|---------|------------------------------------------------------------|---------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|---------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|----|------------------------------------------------------------------------------------------------------|
|                | <p>HAVING is used to filter the aggregation results.</p> <p>Aggregation only works in SELECT and HAVING</p>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                           |     |                                         |       |                                           |     |                                         |     |                                          |     |                                          |     |                                          |      |                                                                                                               |       |                                                                                                                                                                                                                                                                                                                                                                                                               |       |                                                                                                                       |       |                                                                                                                             |      |                                                                                  |        |                                                  |        |                                             |       |                                            |         |                                                            |         |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                        |         |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                |    |                                                                                                      |
| Case Statement | <p>It is a conditional expression that allows you to perform different actions based on specified conditions. It's similar to the if-else construct in other programming languages.</p> <pre>SELECT     id,     name,     salary,     CASE         WHEN salary &gt;= 10000 THEN 'A'         WHEN salary &gt;= 8000 AND salary &lt; 10000 THEN 'B'         WHEN salary &gt;= 6000 AND salary &lt; 8000 THEN 'C'         ELSE 'D'     END AS salary_grade FROM     employees;</pre>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                     |     |                                         |       |                                           |     |                                         |     |                                          |     |                                          |     |                                          |      |                                                                                                               |       |                                                                                                                                                                                                                                                                                                                                                                                                               |       |                                                                                                                       |       |                                                                                                                             |      |                                                                                  |        |                                                  |        |                                             |       |                                            |         |                                                            |         |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                        |         |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                |    |                                                                                                      |
| Functions      | <table border="1"> <tr> <td>AVG</td><td>SELECT AVG(column_name) FROM table_name</td></tr> <tr> <td>COUNT</td><td>SELECT COUNT(column_name) FROM table_name</td></tr> <tr> <td>MAX</td><td>SELECT MAX(column_name) FROM table_name</td></tr> <tr> <td>MIN</td><td>SELECT MIN(column_name) FROM table_name;</td></tr> <tr> <td>SUM</td><td>SELECT SUM(column_name) FROM table_name;</td></tr> <tr> <td>ABS</td><td>SELECT ABS(column_name) FROM table_name;</td></tr> <tr> <td>CEIL</td><td>Returns the smallest integer greater than or equal to the number<br/>SELECT CEIL(column_name) FROM table_name;</td></tr> <tr> <td>FLOOR</td><td>Returns the largest integer less than or equal to the number<br/>SELECT FLOOR(column_name) FROM table_name;<br/><br/>Floor returns the largest integer value that is less than or equal to a given numeric expression.FLOOR(AVG(r.revenue)). Calculates the average revenue for each city and then applies the FLOOR function to get the largest integer less than or equal to that average revenue.<br/>5.678-&gt;5</td></tr> <tr> <td>ROUND</td><td>Rounds a number to a specified number of decimal places<br/>SELECT ROUND(column_name, decimal_places) FROM table_name;</td></tr> <tr> <td>POWER</td><td>Returns the value of a number raised to the power of another number<br/>SELECT POWER(column_name, exponent) FROM table_name;</td></tr> <tr> <td>SQRT</td><td>Returns the square root of a number<br/>SELECT SQRT(column_name) FROM table_name;</td></tr> <tr> <td>CONCAT</td><td>SELECT CONCAT(string1, string2) FROM table_name;</td></tr> <tr> <td>LENGTH</td><td>SELECT LENGTH(column_name) FROM table_name;</td></tr> <tr> <td>UPPER</td><td>SELECT UPPER(column_name) FROM table_name;</td></tr> <tr> <td>REPLACE</td><td>SELECT REPLACE(column_name, 'old', 'new') FROM table_name;</td></tr> <tr> <td>EXTRACT</td><td>extracts a specific part of a date or time value in SQL<br/>-- Extracting Year:<br/>SELECT EXTRACT(YEAR FROM your_date_column) AS year FROM your_table;<br/>-- Extracting Month:<br/>SELECT EXTRACT(MONTH FROM your_date_column) AS month FROM your_table;<br/>-- Extracting Day:<br/>SELECT EXTRACT(DAY FROM your_date_column) AS day FROM your_table;<br/>-- Extracting Hour:<br/>SELECT EXTRACT(HOUR FROM your_timestamp_column) AS hour FROM your_table;<br/>-- Extracting Minute:<br/>SELECT EXTRACT(MINUTE FROM your_timestamp_column) AS minute FROM your_table;<br/>-- Extracting Second:<br/>SELECT EXTRACT(SECOND FROM your_timestamp_column) AS second FROM your_table;</td></tr> <tr> <td>TO_CHAR</td><td>The 'TO_CHAR' function in SQL is used to convert a date or timestamp to a string with a specified format. This is particularly useful for displaying dates in a specific format, performing string operations, or preparing data for reports.<br/><br/>TO_CHAR(date_value, 'format_model')<br/><br/>SELECT TO_CHAR(SYSDATE, 'DD-MON-YYYY') AS formatted_date FROM dual;<br/>SELECT TO_CHAR(SYSDATE, 'Month DD, YYYY') AS formatted_date FROM dual;<br/>SELECT TO_CHAR(SYSDATE, 'YYYY/MM/DD HH24:MI:SS') AS formatted_date FROM dual;</td></tr> <tr> <td>As</td><td>Alias made using as cannot be used in where or having as alias are processed at the end of the query</td></tr> </table> | AVG | SELECT AVG(column_name) FROM table_name | COUNT | SELECT COUNT(column_name) FROM table_name | MAX | SELECT MAX(column_name) FROM table_name | MIN | SELECT MIN(column_name) FROM table_name; | SUM | SELECT SUM(column_name) FROM table_name; | ABS | SELECT ABS(column_name) FROM table_name; | CEIL | Returns the smallest integer greater than or equal to the number<br>SELECT CEIL(column_name) FROM table_name; | FLOOR | Returns the largest integer less than or equal to the number<br>SELECT FLOOR(column_name) FROM table_name;<br><br>Floor returns the largest integer value that is less than or equal to a given numeric expression.FLOOR(AVG(r.revenue)). Calculates the average revenue for each city and then applies the FLOOR function to get the largest integer less than or equal to that average revenue.<br>5.678->5 | ROUND | Rounds a number to a specified number of decimal places<br>SELECT ROUND(column_name, decimal_places) FROM table_name; | POWER | Returns the value of a number raised to the power of another number<br>SELECT POWER(column_name, exponent) FROM table_name; | SQRT | Returns the square root of a number<br>SELECT SQRT(column_name) FROM table_name; | CONCAT | SELECT CONCAT(string1, string2) FROM table_name; | LENGTH | SELECT LENGTH(column_name) FROM table_name; | UPPER | SELECT UPPER(column_name) FROM table_name; | REPLACE | SELECT REPLACE(column_name, 'old', 'new') FROM table_name; | EXTRACT | extracts a specific part of a date or time value in SQL<br>-- Extracting Year:<br>SELECT EXTRACT(YEAR FROM your_date_column) AS year FROM your_table;<br>-- Extracting Month:<br>SELECT EXTRACT(MONTH FROM your_date_column) AS month FROM your_table;<br>-- Extracting Day:<br>SELECT EXTRACT(DAY FROM your_date_column) AS day FROM your_table;<br>-- Extracting Hour:<br>SELECT EXTRACT(HOUR FROM your_timestamp_column) AS hour FROM your_table;<br>-- Extracting Minute:<br>SELECT EXTRACT(MINUTE FROM your_timestamp_column) AS minute FROM your_table;<br>-- Extracting Second:<br>SELECT EXTRACT(SECOND FROM your_timestamp_column) AS second FROM your_table; | TO_CHAR | The 'TO_CHAR' function in SQL is used to convert a date or timestamp to a string with a specified format. This is particularly useful for displaying dates in a specific format, performing string operations, or preparing data for reports.<br><br>TO_CHAR(date_value, 'format_model')<br><br>SELECT TO_CHAR(SYSDATE, 'DD-MON-YYYY') AS formatted_date FROM dual;<br>SELECT TO_CHAR(SYSDATE, 'Month DD, YYYY') AS formatted_date FROM dual;<br>SELECT TO_CHAR(SYSDATE, 'YYYY/MM/DD HH24:MI:SS') AS formatted_date FROM dual; | As | Alias made using as cannot be used in where or having as alias are processed at the end of the query |
| AVG            | SELECT AVG(column_name) FROM table_name                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                               |     |                                         |       |                                           |     |                                         |     |                                          |     |                                          |     |                                          |      |                                                                                                               |       |                                                                                                                                                                                                                                                                                                                                                                                                               |       |                                                                                                                       |       |                                                                                                                             |      |                                                                                  |        |                                                  |        |                                             |       |                                            |         |                                                            |         |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                        |         |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                |    |                                                                                                      |
| COUNT          | SELECT COUNT(column_name) FROM table_name                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                             |     |                                         |       |                                           |     |                                         |     |                                          |     |                                          |     |                                          |      |                                                                                                               |       |                                                                                                                                                                                                                                                                                                                                                                                                               |       |                                                                                                                       |       |                                                                                                                             |      |                                                                                  |        |                                                  |        |                                             |       |                                            |         |                                                            |         |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                        |         |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                |    |                                                                                                      |
| MAX            | SELECT MAX(column_name) FROM table_name                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                               |     |                                         |       |                                           |     |                                         |     |                                          |     |                                          |     |                                          |      |                                                                                                               |       |                                                                                                                                                                                                                                                                                                                                                                                                               |       |                                                                                                                       |       |                                                                                                                             |      |                                                                                  |        |                                                  |        |                                             |       |                                            |         |                                                            |         |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                        |         |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                |    |                                                                                                      |
| MIN            | SELECT MIN(column_name) FROM table_name;                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                              |     |                                         |       |                                           |     |                                         |     |                                          |     |                                          |     |                                          |      |                                                                                                               |       |                                                                                                                                                                                                                                                                                                                                                                                                               |       |                                                                                                                       |       |                                                                                                                             |      |                                                                                  |        |                                                  |        |                                             |       |                                            |         |                                                            |         |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                        |         |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                |    |                                                                                                      |
| SUM            | SELECT SUM(column_name) FROM table_name;                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                              |     |                                         |       |                                           |     |                                         |     |                                          |     |                                          |     |                                          |      |                                                                                                               |       |                                                                                                                                                                                                                                                                                                                                                                                                               |       |                                                                                                                       |       |                                                                                                                             |      |                                                                                  |        |                                                  |        |                                             |       |                                            |         |                                                            |         |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                        |         |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                |    |                                                                                                      |
| ABS            | SELECT ABS(column_name) FROM table_name;                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                              |     |                                         |       |                                           |     |                                         |     |                                          |     |                                          |     |                                          |      |                                                                                                               |       |                                                                                                                                                                                                                                                                                                                                                                                                               |       |                                                                                                                       |       |                                                                                                                             |      |                                                                                  |        |                                                  |        |                                             |       |                                            |         |                                                            |         |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                        |         |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                |    |                                                                                                      |
| CEIL           | Returns the smallest integer greater than or equal to the number<br>SELECT CEIL(column_name) FROM table_name;                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                         |     |                                         |       |                                           |     |                                         |     |                                          |     |                                          |     |                                          |      |                                                                                                               |       |                                                                                                                                                                                                                                                                                                                                                                                                               |       |                                                                                                                       |       |                                                                                                                             |      |                                                                                  |        |                                                  |        |                                             |       |                                            |         |                                                            |         |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                        |         |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                |    |                                                                                                      |
| FLOOR          | Returns the largest integer less than or equal to the number<br>SELECT FLOOR(column_name) FROM table_name;<br><br>Floor returns the largest integer value that is less than or equal to a given numeric expression.FLOOR(AVG(r.revenue)). Calculates the average revenue for each city and then applies the FLOOR function to get the largest integer less than or equal to that average revenue.<br>5.678->5                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                         |     |                                         |       |                                           |     |                                         |     |                                          |     |                                          |     |                                          |      |                                                                                                               |       |                                                                                                                                                                                                                                                                                                                                                                                                               |       |                                                                                                                       |       |                                                                                                                             |      |                                                                                  |        |                                                  |        |                                             |       |                                            |         |                                                            |         |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                        |         |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                |    |                                                                                                      |
| ROUND          | Rounds a number to a specified number of decimal places<br>SELECT ROUND(column_name, decimal_places) FROM table_name;                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                 |     |                                         |       |                                           |     |                                         |     |                                          |     |                                          |     |                                          |      |                                                                                                               |       |                                                                                                                                                                                                                                                                                                                                                                                                               |       |                                                                                                                       |       |                                                                                                                             |      |                                                                                  |        |                                                  |        |                                             |       |                                            |         |                                                            |         |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                        |         |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                |    |                                                                                                      |
| POWER          | Returns the value of a number raised to the power of another number<br>SELECT POWER(column_name, exponent) FROM table_name;                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                           |     |                                         |       |                                           |     |                                         |     |                                          |     |                                          |     |                                          |      |                                                                                                               |       |                                                                                                                                                                                                                                                                                                                                                                                                               |       |                                                                                                                       |       |                                                                                                                             |      |                                                                                  |        |                                                  |        |                                             |       |                                            |         |                                                            |         |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                        |         |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                |    |                                                                                                      |
| SQRT           | Returns the square root of a number<br>SELECT SQRT(column_name) FROM table_name;                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                      |     |                                         |       |                                           |     |                                         |     |                                          |     |                                          |     |                                          |      |                                                                                                               |       |                                                                                                                                                                                                                                                                                                                                                                                                               |       |                                                                                                                       |       |                                                                                                                             |      |                                                                                  |        |                                                  |        |                                             |       |                                            |         |                                                            |         |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                        |         |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                |    |                                                                                                      |
| CONCAT         | SELECT CONCAT(string1, string2) FROM table_name;                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                      |     |                                         |       |                                           |     |                                         |     |                                          |     |                                          |     |                                          |      |                                                                                                               |       |                                                                                                                                                                                                                                                                                                                                                                                                               |       |                                                                                                                       |       |                                                                                                                             |      |                                                                                  |        |                                                  |        |                                             |       |                                            |         |                                                            |         |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                        |         |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                |    |                                                                                                      |
| LENGTH         | SELECT LENGTH(column_name) FROM table_name;                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                           |     |                                         |       |                                           |     |                                         |     |                                          |     |                                          |     |                                          |      |                                                                                                               |       |                                                                                                                                                                                                                                                                                                                                                                                                               |       |                                                                                                                       |       |                                                                                                                             |      |                                                                                  |        |                                                  |        |                                             |       |                                            |         |                                                            |         |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                        |         |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                |    |                                                                                                      |
| UPPER          | SELECT UPPER(column_name) FROM table_name;                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                            |     |                                         |       |                                           |     |                                         |     |                                          |     |                                          |     |                                          |      |                                                                                                               |       |                                                                                                                                                                                                                                                                                                                                                                                                               |       |                                                                                                                       |       |                                                                                                                             |      |                                                                                  |        |                                                  |        |                                             |       |                                            |         |                                                            |         |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                        |         |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                |    |                                                                                                      |
| REPLACE        | SELECT REPLACE(column_name, 'old', 'new') FROM table_name;                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                            |     |                                         |       |                                           |     |                                         |     |                                          |     |                                          |     |                                          |      |                                                                                                               |       |                                                                                                                                                                                                                                                                                                                                                                                                               |       |                                                                                                                       |       |                                                                                                                             |      |                                                                                  |        |                                                  |        |                                             |       |                                            |         |                                                            |         |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                        |         |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                |    |                                                                                                      |
| EXTRACT        | extracts a specific part of a date or time value in SQL<br>-- Extracting Year:<br>SELECT EXTRACT(YEAR FROM your_date_column) AS year FROM your_table;<br>-- Extracting Month:<br>SELECT EXTRACT(MONTH FROM your_date_column) AS month FROM your_table;<br>-- Extracting Day:<br>SELECT EXTRACT(DAY FROM your_date_column) AS day FROM your_table;<br>-- Extracting Hour:<br>SELECT EXTRACT(HOUR FROM your_timestamp_column) AS hour FROM your_table;<br>-- Extracting Minute:<br>SELECT EXTRACT(MINUTE FROM your_timestamp_column) AS minute FROM your_table;<br>-- Extracting Second:<br>SELECT EXTRACT(SECOND FROM your_timestamp_column) AS second FROM your_table;                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                |     |                                         |       |                                           |     |                                         |     |                                          |     |                                          |     |                                          |      |                                                                                                               |       |                                                                                                                                                                                                                                                                                                                                                                                                               |       |                                                                                                                       |       |                                                                                                                             |      |                                                                                  |        |                                                  |        |                                             |       |                                            |         |                                                            |         |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                        |         |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                |    |                                                                                                      |
| TO_CHAR        | The 'TO_CHAR' function in SQL is used to convert a date or timestamp to a string with a specified format. This is particularly useful for displaying dates in a specific format, performing string operations, or preparing data for reports.<br><br>TO_CHAR(date_value, 'format_model')<br><br>SELECT TO_CHAR(SYSDATE, 'DD-MON-YYYY') AS formatted_date FROM dual;<br>SELECT TO_CHAR(SYSDATE, 'Month DD, YYYY') AS formatted_date FROM dual;<br>SELECT TO_CHAR(SYSDATE, 'YYYY/MM/DD HH24:MI:SS') AS formatted_date FROM dual;                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                        |     |                                         |       |                                           |     |                                         |     |                                          |     |                                          |     |                                          |      |                                                                                                               |       |                                                                                                                                                                                                                                                                                                                                                                                                               |       |                                                                                                                       |       |                                                                                                                             |      |                                                                                  |        |                                                  |        |                                             |       |                                            |         |                                                            |         |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                        |         |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                |    |                                                                                                      |
| As             | Alias made using as cannot be used in where or having as alias are processed at the end of the query                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                  |     |                                         |       |                                           |     |                                         |     |                                          |     |                                          |     |                                          |      |                                                                                                               |       |                                                                                                                                                                                                                                                                                                                                                                                                               |       |                                                                                                                       |       |                                                                                                                             |      |                                                                                  |        |                                                  |        |                                             |       |                                            |         |                                                            |         |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                        |         |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                |    |                                                                                                      |

|        |                                                                                                                                                                                                                                                                                                                                 |
|--------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| FILTER | Used with aggregate functions like COUNT(), SUM(), AVG(), etc., to apply a condition only to the rows considered by that aggregate function.<br><br>SELECT<br>COUNT(*) FILTER (WHERE device_type = 'laptop') AS laptop_views,<br>COUNT(*) FILTER (WHERE device_type IN ('tablet', 'phone')) AS mobile_views<br>FROM viewership; |
|--------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|

Exists keyword

Check constraint check(length (phone)>15

# Mongo

28 June 2025 11:09

- MongoDB is a NoSQL database (not based on relational tables).
- It stores data in BSON (Binary JSON) documents, which look like JSON objects.
- It's schema-less—meaning each document in a collection can have a different structure.

| Concept           | Description                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                  |  |
|-------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|--|
| Database          | A container for collections                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                  |  |
| Collection        | Like a table in RDBMS, holds documents                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                       |  |
| Document          | A single record, stored as JSON-like data                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                    |  |
| Field             | A key-value pair inside a document                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                           |  |
| _id               | Unique identifier for each document (auto-generated if not provided)                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                         |  |
| Example           | <pre>{   "_id": "123abc",   "name": "Saurav",   "skills": ["Java", "Selenium"],   "experience": 10 }</pre>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                   |  |
| List databases    | show dbs                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                     |  |
| Switch/create db  | use mydb                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                     |  |
| Create collection | db.createCollection("users")                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                 |  |
| Insert document   | db.users.insertOne({ name: "Saurav", role: "QA" })                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                           |  |
| Insert Many       | <pre>db.employees.insertMany([   { name: "Alice", role: "Dev", experience: 5 },   { name: "Bob", role: "DevOps", experience: 7 } ])</pre>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                    |  |
| Query documents   | <pre>SELECT * FROM orders</pre> <pre>db.orders.find()</pre> <pre>SELECT * FROM orders WHERE customer_name = 'Saurav'</pre> <pre>db.orders.find({ "customer.name": "Saurav" })</pre> <pre>SELECT * FROM orders WHERE status = 'Delivered'</pre> <pre>db.orders.find({ status: "Delivered" })</pre> <pre>SELECT * FROM orders WHERE customer_name = 'Saurav' AND status = 'Delivered';</pre> <pre>db.orders.find({   "customer.name": "Saurav",   status: "Delivered" })</pre> <pre>SELECT * FROM orders WHERE order_date &gt;= '2024-12-01';</pre> <pre>db.orders.find({ orderDate: { \$gte: "2024-12-01T00:00:00Z" } })</pre> <pre>SELECT * FROM orders WHERE EXISTS (SELECT 1 FROM items WHERE price &gt; 5000);</pre> <pre>db.orders.find({ "items.price": { \$gt: 5000 } })</pre> <pre>SELECT * FROM orders WHERE customer_email LIKE '%@example.com';</pre> <pre>db.orders.find({ "customer.email": /@example\.com\$/ })</pre> <pre>SELECT order_id, customer_name FROM orders;</pre> <pre>db.orders.find({}, { orderId: 1, "customer.name": 1, _id: 0 })</pre> <pre>SELECT * FROM orders WHERE order_id IN (1001, 1002, 1003);</pre> <pre>db.orders.find({ orderId: { \$in: [1001, 1002, 1003] } })</pre> <pre>SELECT * FROM orders WHERE status IN ('Processing', 'Shipped');</pre> <pre>db.orders.find({ status: { \$in: ["Processing", "Shipped"] } })</pre> <pre>SELECT * FROM orders ORDER BY order_date DESC;</pre> <pre>db.orders.find().sort({ orderDate: -1 })</pre> <pre>SELECT * FROM orders LIMIT 10 OFFSET 20;</pre> <pre>db.orders.find().skip(20).limit(10)</pre> <pre>SELECT COUNT(*) FROM orders WHERE status = 'Delivered';</pre> <pre>db.orders.countDocuments({ status: "Delivered" })</pre> <pre>SELECT order_id, SUM(price * quantity) AS total FROM order_items GROUP BY order_id;</pre> <pre>db.orders.aggregate([ { \$unwind: "\$items" }, { \$group: { _id: "\$orderId", total: { \$sum: { \$multiply: ["\$items.price", "\$items.quantity"] } } } } ])</pre> |  |
| Distinct          | db.orders.distinct("customer.name")                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                          |  |
| Count of distinct | db.orders.distinct("customer.name").length                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                   |  |

# Mongo Code

28 June 2025 17:34

|                     |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                |
|---------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| MongoConnectionUtil | <pre>public class MongoConnectionUtil {     private static final String CONNECTION_STRING = "mongodb+srv://singhsaurav:%40Mech27041993 @mycluster.25jr9k.mongodb.net/?retryWrites=true&amp;w=majority&amp;appName=MyCluster";      public static MongoClient createMongoClient() {         ServerApi serverApi = ServerApi.builder()             .version(ServerApiVersion.V1)             .build();          MongoClientSettings settings = MongoClientSettings.builder()             .applyConnectionString(new ConnectionString(CONNECTION_STRING))             .serverApi(serverApi)             .build();          return MongoClients.create(settings);     } }</pre>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                    |
| Tests               | <pre>@Test(enabled = false) public void testConnection() {     try (MongoClient mongoClient = MongoConnectionUtil.createMongoClient()) {         try {             MongoDB database = mongoClient.getDatabase("admin");             database.runCommand(new Document("ping", 1));             System.out.println("Pinged your deployment. You successfully connected to MongoDB!");         } catch (MongoException e) {             e.printStackTrace();         }     } }  @Test public void printAmazon() {     List&lt;Document&gt; results = new ArrayList&lt;&gt;();     try (MongoClient mongoClient = MongoConnectionUtil.createMongoClient()) {         try {             MongoDB database = mongoClient.getDatabase("amazon");             MongoCollection&lt;Document&gt; collection = database.getCollection("orders");             try (MongoCursor&lt;Document&gt; cursor = collection.find().sort(new Document("orderDate", -1)).iterator()) {                 while (cursor.hasNext()) {                     Document doc = cursor.next();                     results.add(doc);                     System.out.println(doc.toJson());                 }             }             exportToExcel(results, "printAmazon.xlsx");             System.out.println("Exported to orders.xlsx");         } catch (MongoException   IOException e) {             e.printStackTrace();         }     } }  @Test public void printOrderIdAndCustomerName() {     List&lt;Document&gt; results = new ArrayList&lt;&gt;();     try (MongoClient mongoClient = MongoConnectionUtil.createMongoClient()) {         MongoDB database = mongoClient.getDatabase("amazon");         MongoCollection&lt;Document&gt; collection = database.getCollection("orders");         Document projection = new Document("orderId", 1).append("customer.name", 1).append("_id", 0);         try (MongoCursor&lt;Document&gt; cursor = collection.find(new Document()).projection(projection).iterator()) {             while (cursor.hasNext()) {                 Document doc = cursor.next();                 results.add(doc);                 System.out.println(doc.toJson());             }         }         exportToExcel(results, "orders.xlsx");         System.out.println("Exported to orders.xlsx");     } catch (Exception e) {         e.printStackTrace();     } } }</pre> |
| ExcelExport         | <pre>public class ExcelExport {      public static void exportToExcel(List&lt;Document&gt; data, String filePath) throws IOException {         if (data == null    data.isEmpty()) return;          // Collect all unique keys (including nested) from the first document         Set&lt;String&gt; headersSet = new LinkedHashSet&lt;&gt;();         for (Document doc : data) {</pre>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                        |

```

 collectKeys(doc, "", headersSet);
 }
 List<String> headers = new ArrayList<>(headersSet);

 try (Workbook workbook = new XSSFWorkbook()) {
 Sheet sheet = workbook.createSheet("Export");
 int rowIdx = 0;
 // Header row
 Row headerRow = sheet.createRow(rowIdx++);
 for (int i = 0; i < headers.size(); i++) {
 headerRow.createCell(i).setCellValue(headers.get(i));
 }
 // Data rows
 for (Document doc : data) {
 Row row = sheet.createRow(rowIdx++);
 for (int i = 0; i < headers.size(); i++) {
 String header = headers.get(i);
 Object value = getNestedValue(doc, header);
 row.createCell(i).setCellValue(value != null ? String.valueOf(value) : "");
 }
 }
 try (FileOutputStream out = new FileOutputStream(filePath)) {
 workbook.write(out);
 }
 }

 // Recursively collect all keys (dot notation for nested)
 private static void collectKeys(Document doc, String prefix, Set<String> keys) {
 for (Map.Entry<String, Object> entry : doc.entrySet()) {
 String key = prefix.isEmpty() ? entry.getKey() : prefix + "." + entry.getKey();
 if (entry.getValue() instanceof Document) {
 collectKeys((Document) entry.getValue(), key, keys);
 } else {
 keys.add(key);
 }
 }
 }

 // Helper to get nested value using dot notation
 private static Object getNestedValue(Document doc, String path) {
 String[] parts = path.split("\\.");
 Object value = doc;
 for (String part : parts) {
 if (value instanceof Document) {
 value = ((Document) value).get(part);
 } else {
 return null;
 }
 }
 return value;
 }
}

```

# Netx360 WF

26 October 2024 10:59

|                    |                                                                                                                                       |
|--------------------|---------------------------------------------------------------------------------------------------------------------------------------|
| UI                 | Submit Order -> Sends data mapping to Mainframe after transforming to Mainframe's format<br>Buy->B<br>Sell->S<br>Source of Input->soi |
| Rules Engine       | Below Min amount for account/security<br>90 Day Trading restriction<br>Order already exists                                           |
| MDS                | Quotes                                                                                                                                |
| Ria                | Registered Investment advisor                                                                                                         |
| BD                 | Broker Dealer                                                                                                                         |
| Atoms              | Order Execution                                                                                                                       |
| M/F                | Order Storage                                                                                                                         |
| Account Summary    | Per Account<br>All Asset Type                                                                                                         |
| Order Modification | Fetch order detail from M/F based on orderdetail key                                                                                  |
| Cancel Replace     | Change Qty                                                                                                                            |
| Correct            | Update Comment, commission                                                                                                            |
| BD                 | Extra Fields Comments, order qualifier, Commission, order date and time                                                               |
| Account Detail     | MF                                                                                                                                    |
| Positions balances | M/F                                                                                                                                   |
| Entitlement        | DASF                                                                                                                                  |

|                        |                                                                                                                                                                                                                                                                                                                                                                                                                                      |
|------------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Src/main/resources     | Has log4j.xml                                                                                                                                                                                                                                                                                                                                                                                                                        |
| SRC/Main/java          | Has utils                                                                                                                                                                                                                                                                                                                                                                                                                            |
| SRC/Test/java          | In nextgen.cucumber <ul style="list-style-type: none"> <li>• Pageobjectmanager</li> <li>• Page classes extending basepageobject: has wait, click, send input functions, get elements after adding waits</li> <li>• Step Def extends abstractsteps. Abs step has start driver() which starts approp driver</li> <li>• Test runners extends abstracttestngcucumbertest</li> <li>• Bddreport listner implements itestlistner</li> </ul> |
| SRC/Test/resources     | <ul style="list-style-type: none"> <li>• Data</li> <li>• Feature files</li> <li>• Properties</li> <li>• Extent.properties</li> </ul>                                                                                                                                                                                                                                                                                                 |
| Framework Architecture | <ul style="list-style-type: none"> <li>• Type: Hybrid (Data/Page Object Model) + BDD (if using Gherkin)</li> <li>• Core Principles: Scalability, Reusability, Low Maintenance (Self-healing scripts)</li> <li>• Design Patterns: Page Object Model (POM), Singleton</li> <li>• Modularity: Separated test scripts, libraries, configs, and test data</li> </ul>                                                                      |
| Tech Stack             | <ul style="list-style-type: none"> <li>• Language: Java + Selenium</li> <li>• Test Runner: TestNG</li> <li>• CI/CD: GitLab CI (Pipeline-as-Code)</li> <li>• Cloud: BrowserStack</li> <li>• Reporting: ExtentReports</li> <li>• Test Management: Jira/ Zephyr</li> <li>• Version Control: Git</li> </ul>                                                                                                                              |

# Career

15 March 2025 18:25

Hard work can only reach you up to a certain limit. You need to create your perception and your visibility among peers and bosses. Know what others are saying about you.

|                               |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                    |
|-------------------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>Early Wins</b>             | <p>Start Strong and Build Foundations.</p> <p>Identify low-effort, high-impact tasks to deliver early wins. This builds momentum and trust.</p>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                    |
| <b>Active Listening</b>       | Observe team dynamics and understand unspoken expectations.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                        |
| <b>Help the Business</b>      | <p>Understand how our work impacts business and its criticality.</p> <p>Aim to value add to business with your work. Aim for cost saving/ROI.</p>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                  |
| <b>Quality over Quantity</b>  | Focus on key business transactions rather than quantity of scripts.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                |
| <b>Approach</b>               | Figure out a consistent approach, take a step back and start small, Identify key processes, document and become SME.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                               |
| <b>Stakeholder Management</b> | Make a list of people with whom you need to create your visibility. Identify key influencers beyond immediate stakeholders. Building rapport with influential individuals can boost visibility. Identify each stakeholder's preferred communication method (detailed reports, quick updates, etc.) and tailor your interactions accordingly.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                       |
| <b>Achievements Sheet</b>     | <p>List all accomplishments, even small ones.</p> <p>Maintain balance in your brag.</p> <p>Try to sit with your stakeholders and brag at least 1 point. Don't go overboard and balance this appropriately.</p> <p><b>Quantify Achievements:</b> Focus not only on achievements but also their impact (e.g., cost savings, efficiency improvements, risk mitigation).</p> <p><b>Storytelling Skills:</b> Develop concise narratives that turn your accomplishments into memorable stories for your stakeholders.</p> <p><b>Document Contributions:</b></p> <p>Maintain a detailed record of your contributions, including projects you worked on, tasks you completed, and results you achieved.</p> <p>This documentation will be valuable when discussing your performance with your manager and advocating for a promotion.</p>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                  |
| <b>Boss Management</b>        | <p>Share his pressure.</p> <p>Understand his situation, help him in his deliverables.</p> <p>Do this repeatedly.</p> <p>Be there for him.</p>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                      |
| <b>Peer Management</b>        | <p>Correct articulation in front of the right people.</p> <p>Learn how to talk, or the art of storytelling.</p> <p>Be knowledgeable and take care of your team.</p> <p>Build trust among peers.</p> <p>Don't only be a loud mouth.</p>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                             |
| <b>Persistent Delivery</b>    | This brings trust. Identify repetitive tasks that can be automated to free up time for strategic contributions.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                    |
| <b>People Skills</b>          | <p><b>Handling Politics and Difficult People</b></p> <ul style="list-style-type: none"><li>Emotional Intelligence: Develop empathy and emotional control to manage tense situations.</li><li>Influence Tactics: Practice persuasion techniques like social proof, reciprocity, and consensus.</li></ul> <p><b>Visibility and Perception</b></p> <ul style="list-style-type: none"><li>Internal Branding: Position yourself as a go-to expert for specific technical or domain-related queries.</li><li>Proactive Communication: Share insights in meetings, discussions, or internal forums.</li></ul> <p><b>Dealing with fake people:</b></p> <p>Be vulnerable to correct people and share your struggles.</p> <p>Seek help, not complain at the right moment, like in one-on-ones.</p> <p><b>Saying The Right Things:</b></p> <p>Don't appear irrational, be cool, calmly composed.</p> <p>Have patience when others are triggered.</p> <p>Don't overshare.</p> <p><b>Understand the landscape:</b> Like how is the relationship between peers, bosses, super bosses.</p> <p><b>Build relationships all across.</b></p> <p><b>Effective Communication:</b> Crisp and concise.</p> <p><b>Strategic Communication:</b></p> <ul style="list-style-type: none"><li>Refine your communication skills further.</li><li>Focus on being clear, concise, and persuasive in your interactions.</li><li>Tailor your communication style to your audience and the situation.</li><li>Practice active listening and seek to understand different perspectives.</li></ul> <p><b>Diplomatic:</b> Find common ground and don't burn bridges.</p> <p><b>Stay informed about systems and situations.</b></p> <p><b>Networking:</b></p> <ul style="list-style-type: none"><li>Expand your network within the company and industry.</li><li>Attend company events, join professional organizations, and connect with people in different departments.</li></ul> <p><b>Mentorship:</b></p> <ul style="list-style-type: none"><li>Seek out a mentor within the company who can provide guidance and support.</li></ul> |

|                                       |                                                                                                                                                                                                                                                      |
|---------------------------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
|                                       | Consider also mentoring junior colleagues.                                                                                                                                                                                                           |
| <b>Seek Feedback Regularly</b>        | Proactively ask your manager and stakeholders for feedback on your performance. Use this feedback to identify areas for improvement and adjust your approach.                                                                                        |
| <b>Communicate Career Aspirations</b> | Be transparent with your manager about your career goals and aspirations. Discuss what you need to do to achieve that goal.                                                                                                                          |
| <b>Misc</b>                           | <p>Make Notes<br/>Listen more than you speak.<br/>Never Complain<br/>Never be disrespectful<br/>Understand boss expectations and set regular touchpoints.<br/>Touchpoints with other stakeholders<br/>Solve Problems<br/>Be Patient and Positive</p> |

| Competence                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                               | Commitment                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                             | Compatibility                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                  |
|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <ul style="list-style-type: none"> <li>Develop a roadmap for skill enhancement relevant to your new role. Identify emerging trends in your domain and proactively learn those.</li> <li><b>Deepen Domain Knowledge:</b> While you mentioned identifying key processes and becoming an SME, explicitly state a plan to deepen your understanding of the business domain. In your case, you have experience in capital markets and investment banking. Make a focused effort to learn more about the specifics of your company's products, services, and the industry trends impacting them. This will add more value to your contributions.</li> <li><b>Technical Skills Upgrade:</b> Technology evolves rapidly. Commit to continuous learning in your technical skills. Identify any skill gaps and create a plan to address them through courses, certifications, or self-study. Consider adding specific technologies relevant to your new role that you want to learn.</li> <li><b>Proactive Problem Solving:</b> Go beyond identifying problems; focus on proposing solutions. When you encounter a challenge, analyze it thoroughly, develop potential solutions, and present them to your team or manager. This demonstrates initiative and problem-solving capabilities.</li> <li>Take a minute before asking questions and share the steps followed till now to get to the solution.</li> </ul> | <ul style="list-style-type: none"> <li><b>Ask about expectations from you</b> <ul style="list-style-type: none"> <li>What are my top priorities?</li> <li>How would you like us to communicate?</li> <li>What your top performers do, what do you suggest I do?</li> </ul> </li> <li><b>Ownership and Accountability:</b> <ul style="list-style-type: none"> <li>Take full ownership of your tasks and deliverables.</li> <li>Be accountable for your results, both successes and failures.</li> <li>If you encounter setbacks, focus on finding solutions and learning from the experience.</li> </ul> </li> <li><b>Go the Extra Mile:</b> <ul style="list-style-type: none"> <li>Look for opportunities to go beyond your assigned responsibilities.</li> <li>This could involve assisting colleagues, volunteering for additional projects, or taking on tasks that others might avoid.</li> </ul> </li> <li><b>Time Management and Prioritization:</b> <ul style="list-style-type: none"> <li>Master your time management skills.</li> <li>Prioritize tasks effectively, meet deadlines consistently, and manage your workload efficiently.</li> </ul> </li> </ul> | <ul style="list-style-type: none"> <li><b>Initiate conversations</b> <ul style="list-style-type: none"> <li>Identify who is responsible for what.</li> <li>Understand the hierarchy.</li> <li>Identify the gaps in processes/tasks.</li> <li>Practice adaptive communication—mirror the communication style of your manager and stakeholders.</li> </ul> </li> <li><b>Emotional Intelligence:</b> <ul style="list-style-type: none"> <li>Develop your EQ to navigate workplace dynamics effectively, build strong relationships, and communicate constructively.</li> </ul> </li> <li><b>Collaboration and Teamwork:</b> <ul style="list-style-type: none"> <li>Focus on being a collaborative team player.</li> <li>Share your knowledge and expertise with others, support your colleagues, and contribute to a positive team environment.</li> <li>Be willing to help others and ask for help when needed.</li> </ul> </li> <li><b>Adaptability and Flexibility:</b> <ul style="list-style-type: none"> <li>Be open to change and adaptable to new situations.</li> <li>Demonstrate flexibility and a willingness to learn and grow.</li> </ul> </li> </ul> |

#### Key Questions to Ask

- What are my top priorities?
- What are the expectations from me in the next 3, 6, and 12 months?
- Journal of work done, achievement, Find right time to share this achievement and the Problem statement this achievement solved.
- Don't sound cheap, focus on organisation goals. Focus on what the stakeholder values, mold your achievement around that.
- Once a month meet with manager.
- Spin negatives to positive solution.
- Work alone doesn't help in promotion, what else-> Quantify efforts skip level, understand pain points ,set clear expectations for promotion
- Skip level: understand super boos view,what he wants to achieve . Expect mentorship .operate at next level

#### Promotion Plan

- 1st 6 Months: Learn and Understand
- Next 6 Months: Overdeliver and shine
- Next 6 Months: Share learnings, mentor juniors, present to stakeholders, communicate your expectations like how can I put myself in the best position for promotion in the next 2 cycles. Ask on where to improve.
- Identify a mentor within the organization to guide you through politics, growth opportunities, and strategic insights.

# Unspoken Rules

17 May 2025 09:49

|                                            |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                        |
|--------------------------------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| See the big picture                        | <p>When joining a new team</p> <ul style="list-style-type: none"> <li>• Research what the team does</li> <li>• What its objectives are</li> <li>• Whom it serves</li> <li>• How your role will help the team and organization achieve their goals.</li> </ul> <p>When taking on a new assignment</p> <ul style="list-style-type: none"> <li>• Understand the broader objective</li> <li>• What success looks like</li> <li>• How your work fits into the big picture.</li> </ul>                                                                                                                                                                                                                                                                                                                                       |
| Do and show your homework                  | When you have a question, avoid immediately pulling others aside. Look through your emails and files and search online first. If you can't find the answer, bundle and escalate.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                       |
| Know your internal and external narratives | <p>Know why you do what you do.</p> <p>When introducing yourself, talk about your past, present, and future: share what you've done, what you're working on, and, if relevant, what you're trying to achieve.</p> <p>When giving a status update, talk about what you've done, followed by what you still have to do.</p>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                              |
| Think multiple steps ahead                 | <p>Learn what your manager might ask of you—and have it ready.</p> <p>Know what issues your manager might face—and offer a solution.</p> <p>Before submitting your work or entering a meeting, brainstorm what you might be asked and have a response ready.</p> <p>When others tell you to do something, think multiple steps ahead: Do their directions make sense? Might their idea cause issues for others?</p>                                                                                                                                                                                                                                                                                                                                                                                                    |
| Work backward from the end goal            | <p>Understand what you are trying to achieve, then work backward, mapping out all the steps and deadlines between you and the end goal.</p> <p>Make sure you are clear on what you need to do, how you need to do it, and by when.</p> <p>Ask colleagues and superiors, "When would it make sense to check in?"</p> <p>Repeat back what you think you heard before walking away. Then, constantly assess whether what you are doing is getting you closer to the end goal.</p>                                                                                                                                                                                                                                                                                                                                         |
| Show performance and potential             | <p>Know that you are being evaluated based on both your performance (how effective you are in your current role) and your potential (how effective you might be in your next role).</p> <p>To show your potential</p> <ul style="list-style-type: none"> <li>• claim an unclaimed swimlane <ul style="list-style-type: none"> <li>◦ do what hasn't been done</li> <li>◦ fix what hasn't been fixed</li> <li>◦ bridge what hasn't been bridged</li> <li>◦ know what others don't know.</li> </ul> </li> </ul> <p>Don't let potential go unrecognized. Ask for what you want—and deserve. Observe the people around you at work. Notice how those who get ahead have mastered most, if not all, of these unspoken rules—and how those who struggle repeatedly stumble with at least one or more of these same rules.</p> |
| 3 C's                                      | <p>"I trust you with more important responsibilities."</p> <p>"I want to work with you."</p> <p>"I trust you with more important responsibilities."</p> <p>"COMPATIBILITY<br/>Do you get along with us?"</p> <p>"COMPETENCE<br/>Can you do your job well?"</p> <p>"COMMITMENT<br/>Are you excited to be here?"</p> <p>"I want to invest in you."</p>                                                                                                                                                                                                                                                                                                                                                                                                                                                                   |

|                     |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                         |
|---------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
|                     | <p><b>Competence</b> Competence means you can do your job fully, accurately, and promptly without needing to be micromanaged — and without making others look bad.</p> <p>This means not undershooting to the point of looking clueless and not overshooting to the point of looking overbearing.</p>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                   |
|                     | <p><b>Commitment</b> Commitment means you are fully present and eager to help your team achieve its goals — but not so eager that you put others on the defensive.</p> <p>This means not undershooting to the point of looking apathetic and not overshooting to the point of looking threatening.</p> <p>A summer camp counselor was accused of being lazy by the camp director despite working hard and taking on extra tasks. One of his fellow counselors pulled him aside and helped him see that the issue wasn't his lack of effort; it was his lack of enthusiasm. He looked mellow and often had his phone out. His fellow counselors, on the other hand, behaved like they were in a musical about peppy camp counselors. Over the following weeks, this counselor threw on a smile, walked faster, and added a spring to his step. To his surprise, the director started taking him seriously — all because he no longer looked apathetic.</p> <p>Another time, a college student had gotten an internship at an investment bank. Anytime he got his own work done early, he'd start doing his teammates' work without asking. He sometimes even corrected his manager in front of higher-ups. In the end, he was one of two interns who didn't receive a full-time job offer — all from being threatening.</p>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                              |
|                     | <p><b>Compatibility</b> Compatibility means you make others comfortable and eager to be around you — without coming across as inauthentic or trying too hard. This means not undershooting to the point of looking passive and not overshooting to the point of looking like a poser.</p> <p>One time, a cashier at a movie theater was told by her manager to be more of a "team player." She was confused. She always showed up to work on time and dealt with customers politely. But it wasn't enough because she hardly smiled and didn't make small talk with her manager as her coworkers did. She didn't pass her probation period — all because she looked passive and withdrawn.</p> <p>Another time, a recent graduate of an American MBA program joined a corporate strategy team at an energy company in Asia. One day, he attended a supplier presentation with some senior coworkers. At the end of the presentation, the vendor asked, "Any questions?" The room went silent. Not realizing that his coworkers were following a cultural norm of waiting for the most senior person to speak first, he blurted, "Well, if none of you have questions, I've got a question." His coworkers all rolled their eyes — and saw him as a poser.</p>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                           |
| <b>Be Proactive</b> | <ul style="list-style-type: none"> <li>• Have I met my supervisor(s)?</li> <li>• Have I clarified my reporting lines?</li> <li>• Have I clarified expectations for my role? <ul style="list-style-type: none"> <li>• If you are filling a new role (e.g., you are the first recruiter the team has hired), then you will want to understand why your position was created, what your mandate is (how you are expected to do more, better, faster, or cheaper), and how things have typically been done in the past.</li> <li>• If you are filling an existing role, then you will want to understand what your predecessor did and how they did it. That way you can do the job just as well — or better.</li> <li>• Here are five questions to consider asking: <ul style="list-style-type: none"> <li>◦ "Which tasks and deliverables are top priorities in my role? Which ones are secondary?"</li> <li>◦ "What should I be able to do by the end of the first three months? Six months?"</li> <li>◦ "What does success look like in my role? Are there any metrics I should keep in mind?"</li> <li>◦ "Is there anyone else you'd suggest I introduce myself to?"</li> <li>◦ "What should day-to-day and week-to-week collaboration look like between us? When should I be proactive and when should I be reactive?"</li> </ul> </li> </ul> </li> <li>• Have I found a regular interaction schedule with my manager? If your manager doesn't mention one-on-one meetings, consider asking, "Would it be helpful for us to have some sort of regular check-in?" followed by "What's most convenient: weekly, biweekly, monthly?"</li> <li>• Have I introduced myself to my coworkers?</li> <li>• Have I learned my team's priorities? As you meet your teammates, consider asking, "What are you currently working on?" or "What are your and the team's top priorities these days?" The better you understand what everyone is working on and stressed out about, the better you will be at finding ways to make yourself useful</li> <li>• If you have any questions, bundle, escalate and show your homework.</li> <li>• If unsure on the next steps, think as an owner, what will I do if I owned this situation and didn't have anyone to lean on.</li> <li>• If you need someone's help, try to minimize their workload like giving them necessary files and links.</li> <li>• If you are making any decision, see the big picture and play out the consequences like sharing the updated code with others if it's a shared module.</li> <li>• If you see an error, flag it/correct it. Applause in public, criticism in private.</li> <li>• Stay one step ahead of your boss</li> <li>• Before submitting check have I followed all the directions?</li> <li>• Spark Relationships</li> <li>• People can't read your mind, so they don't know how hard you've been working or how good a job you've been doing. But they will observe how you come across in meetings (and in other settings) — and assume it fully and accurately reflects how you are doing in your job overall.</li> </ul> |

## Seven questions to ask yourself to prepare for any meeting

### When to ask

- |                          |                                                                                                                                                                                                      |
|--------------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Before<br>the<br>meeting | What will this meeting be about and who will be in the room?<br>What's my role in this meeting?<br>What questions might I be asked?<br>What's my one thoughtful comment and one thoughtful question? |
| During                   | When should I speak up?<br>How can I best deliver my point?                                                                                                                                          |
| After                    | What (if anything) do I need to do to follow up?                                                                                                                                                     |

|                                          |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                 |
|------------------------------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Answering Meeting Questions              | <ul style="list-style-type: none"> <li>I'm working on _____, which is an effort to _____.</li> <li>So far I've done _____ and will focus on _____ next.</li> <li>I expect to finish _____ by _____.</li> <li>I could use some help with _____ because of _____.</li> <li>I know someone asked about _____ last time. I've looked into it and found out _____.</li> </ul>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                        |
| Asking Feedback from Manager             | <p>"Would you have a few minutes to chat in the coming days? I'd love to hear how I am doing and if there is any way I can improve in my work."</p> <p>"Thanks for taking the time to chat. I value your opinion a lot and would love to talk about anything I can do to improve and take my work to the next level."</p> <ul style="list-style-type: none"> <li>• "What should I start doing? Stop doing? Keep doing?"</li> <li>• "Am I on track with [whatever project I've been assigned]?"</li> <li>• "Am I on track to [receive a full-time job offer or get promoted]?"</li> <br/> <li>• If you receive critical feedback that's not specific enough. "Interesting. Can you share any specific moments when I _____?"</li> <li>• If you aren't sure how to apply certain feedback. "Good point. I'd love to _____. Any suggestions on how I can apply this feedback going forward?"</li> <li>• If you receive feedback that's difficult to apply. "That's a good point. How would you suggest I balance _____ with _____?"</li> <li>• If you receive feedback you disagree with and you want to explain yourself. "Thanks for that. I could definitely improve on _____. I'm just thinking back to that moment and wonder if what went through my head was _____. "</li> <li>• If you want help achieving a certain goal. "I'd love to _____ and would appreciate your advice. How would you suggest I go about navigating _____?"</li> <li>• When you don't know what to say next. "I appreciate you bringing that up."/" That's a good point."/" That's interesting."/" That's helpful."/" Thanks for that."</li> </ul> |
| Applying Feedback                        | <p>If you attempt to apply their feedback, but realize that it is unreasonable or infeasible, you could report back and say, "I've been trying to apply your advice to _____, but have been struggling with _____. Do you have any advice on how to better navigate these situations?" If the feedback ends up being helpful, however, you could report back with, "Thanks so much for your suggestion to _____. I've been trying it on _____ and _____ occasions and have seen _____ improvements."</p> <p>In the end, it's often not about how positive or negative the feedback is, but rather about how readily you embrace and apply it.</p>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                               |
| How to Position Yourself for a Promotion | <p>What can I do that hasn't been done?: Most organizations value the same four things:</p> <ul style="list-style-type: none"> <li>• more customers, clients, donors, and fans;</li> <li>• better products, services, and reviews;</li> <li>• faster ways of getting things done;</li> <li>• cheaper ways to keep everything running.</li> </ul> <p>If you can achieve one or more of these goals, you can boost others' perceptions of your potential— and your promotability.</p>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                             |

# Test Lead Interview Questions

20 January 2025 09:24

|                                                                                                                                                                                                                           |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                       |                                                                                                                                                                                                                                                                                                   |                                                                                                                                                                                                                                                                                                                                                                                                                                            |                                                                                                                                                                                                                                                                                                                                                                                                                                            |                                                                                                                                                                                             |                                                                                                                                                                                                                                                                                                                                                                                                                              |                                                                                                                                                                             |                                                                                                                                                                                                                                                                                                   |                                                                                                                                  |                                                                                                                                                                                                                                                                                                                                                                |                                                                                                                                                                                                                        |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                   |                        |                                                                                                                                                                                                                                                                                                                                                                               |
|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|----------------------------------------------------------------------------------------------------------------------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|------------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Explain how do you pick which test cases to automate                                                                                                                                                                      | <table border="1"> <tr> <td>Frequency of Execution</td><td>Prioritize test cases that are executed repeatedly, such as regression tests or smoke tests. Automation can significantly reduce the time and effort required for these tests.</td></tr> <tr> <td>Business Criticality</td><td>Concentrate on test cases that cover critical functionalities or areas with high business impact. These tests should be automated to ensure the stability and reliability of core features.</td></tr> <tr> <td>Test Complexity</td><td>Consider automating complex test scenarios that involve multiple steps or interactions. Automation can help ensure consistent execution and reduce the risk of human error.</td></tr> <tr> <td>Repetitive Tasks</td><td>Automate tasks that are prone to human error, such as data entry or repetitive clicks. This can improve accuracy and efficiency.</td></tr> <tr> <td>Automation Suitability/Stability</td><td>Ensure that the test cases can be technically automated using available tools and technologies. Consider factors like the application's architecture, accessibility of elements, and the complexity of the test logic.</td></tr> </table>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                       | Frequency of Execution                                                                                                                                                                                                                                                                            | Prioritize test cases that are executed repeatedly, such as regression tests or smoke tests. Automation can significantly reduce the time and effort required for these tests.                                                                                                                                                                                                                                                             | Business Criticality                                                                                                                                                                                                                                                                                                                                                                                                                       | Concentrate on test cases that cover critical functionalities or areas with high business impact. These tests should be automated to ensure the stability and reliability of core features. | Test Complexity                                                                                                                                                                                                                                                                                                                                                                                                              | Consider automating complex test scenarios that involve multiple steps or interactions. Automation can help ensure consistent execution and reduce the risk of human error. | Repetitive Tasks                                                                                                                                                                                                                                                                                  | Automate tasks that are prone to human error, such as data entry or repetitive clicks. This can improve accuracy and efficiency. | Automation Suitability/Stability                                                                                                                                                                                                                                                                                                                               | Ensure that the test cases can be technically automated using available tools and technologies. Consider factors like the application's architecture, accessibility of elements, and the complexity of the test logic. |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                   |                        |                                                                                                                                                                                                                                                                                                                                                                               |
| Frequency of Execution                                                                                                                                                                                                    | Prioritize test cases that are executed repeatedly, such as regression tests or smoke tests. Automation can significantly reduce the time and effort required for these tests.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                        |                                                                                                                                                                                                                                                                                                   |                                                                                                                                                                                                                                                                                                                                                                                                                                            |                                                                                                                                                                                                                                                                                                                                                                                                                                            |                                                                                                                                                                                             |                                                                                                                                                                                                                                                                                                                                                                                                                              |                                                                                                                                                                             |                                                                                                                                                                                                                                                                                                   |                                                                                                                                  |                                                                                                                                                                                                                                                                                                                                                                |                                                                                                                                                                                                                        |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                   |                        |                                                                                                                                                                                                                                                                                                                                                                               |
| Business Criticality                                                                                                                                                                                                      | Concentrate on test cases that cover critical functionalities or areas with high business impact. These tests should be automated to ensure the stability and reliability of core features.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                           |                                                                                                                                                                                                                                                                                                   |                                                                                                                                                                                                                                                                                                                                                                                                                                            |                                                                                                                                                                                                                                                                                                                                                                                                                                            |                                                                                                                                                                                             |                                                                                                                                                                                                                                                                                                                                                                                                                              |                                                                                                                                                                             |                                                                                                                                                                                                                                                                                                   |                                                                                                                                  |                                                                                                                                                                                                                                                                                                                                                                |                                                                                                                                                                                                                        |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                   |                        |                                                                                                                                                                                                                                                                                                                                                                               |
| Test Complexity                                                                                                                                                                                                           | Consider automating complex test scenarios that involve multiple steps or interactions. Automation can help ensure consistent execution and reduce the risk of human error.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                           |                                                                                                                                                                                                                                                                                                   |                                                                                                                                                                                                                                                                                                                                                                                                                                            |                                                                                                                                                                                                                                                                                                                                                                                                                                            |                                                                                                                                                                                             |                                                                                                                                                                                                                                                                                                                                                                                                                              |                                                                                                                                                                             |                                                                                                                                                                                                                                                                                                   |                                                                                                                                  |                                                                                                                                                                                                                                                                                                                                                                |                                                                                                                                                                                                                        |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                   |                        |                                                                                                                                                                                                                                                                                                                                                                               |
| Repetitive Tasks                                                                                                                                                                                                          | Automate tasks that are prone to human error, such as data entry or repetitive clicks. This can improve accuracy and efficiency.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                      |                                                                                                                                                                                                                                                                                                   |                                                                                                                                                                                                                                                                                                                                                                                                                                            |                                                                                                                                                                                                                                                                                                                                                                                                                                            |                                                                                                                                                                                             |                                                                                                                                                                                                                                                                                                                                                                                                                              |                                                                                                                                                                             |                                                                                                                                                                                                                                                                                                   |                                                                                                                                  |                                                                                                                                                                                                                                                                                                                                                                |                                                                                                                                                                                                                        |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                   |                        |                                                                                                                                                                                                                                                                                                                                                                               |
| Automation Suitability/Stability                                                                                                                                                                                          | Ensure that the test cases can be technically automated using available tools and technologies. Consider factors like the application's architecture, accessibility of elements, and the complexity of the test logic.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                |                                                                                                                                                                                                                                                                                                   |                                                                                                                                                                                                                                                                                                                                                                                                                                            |                                                                                                                                                                                                                                                                                                                                                                                                                                            |                                                                                                                                                                                             |                                                                                                                                                                                                                                                                                                                                                                                                                              |                                                                                                                                                                             |                                                                                                                                                                                                                                                                                                   |                                                                                                                                  |                                                                                                                                                                                                                                                                                                                                                                |                                                                                                                                                                                                                        |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                   |                        |                                                                                                                                                                                                                                                                                                                                                                               |
| What are the important validation you will put on a API.                                                                                                                                                                  | <table border="1"> <tr> <td data-bbox="468 595 658 1090" rowspan="3">Functional Validation</td><td data-bbox="658 595 793 798">Input Validation</td><td data-bbox="793 595 1569 798"> <ul style="list-style-type: none"> <li>Data Type: Verify that the API accepts the correct data types for all parameters (e.g., strings, numbers, booleans).</li> <li>Data Format: Ensure that the data is in the expected format (e.g., date, time, currency).</li> <li>Data Range: Check that the values of parameters fall within acceptable ranges.</li> <li>Required Fields: Verify that all mandatory fields are provided.</li> </ul> </td></tr> <tr> <td data-bbox="658 798 793 1000">Output Validation</td><td data-bbox="793 798 1569 1000"> <ul style="list-style-type: none"> <li>Data Structure: Validate that the response data structure matches the expected format (e.g., JSON, XML).</li> <li>Data Accuracy: Check that the values in the response are correct and consistent with the input.</li> <li>Status Codes: Verify that the API returns the appropriate HTTP status codes (e.g., 200 for success, 400 for bad request, 500 for server error).</li> </ul> </td></tr> <tr> <td data-bbox="658 1000 793 1090">Error Handling</td><td data-bbox="793 1000 1569 1090"> <ul style="list-style-type: none"> <li>Error Messages: Ensure that the API returns meaningful error messages when invalid input is provided or unexpected errors occur.</li> <li>Error Codes: Verify that the API returns appropriate error codes to help identify the specific issue.</li> </ul> </td></tr> <tr> <td data-bbox="468 1090 658 1248">Performance Validation</td><td data-bbox="658 1090 1569 1248"> <ul style="list-style-type: none"> <li>Response Time: Measure the time it takes for the API to respond to requests under different load conditions.</li> <li>Throughput: Determine the number of requests the API can handle per second.</li> <li>Resource Utilization: Monitor CPU, memory, and network usage to identify performance bottlenecks.</li> </ul> </td></tr> <tr> <td data-bbox="468 1248 658 1450">Security Validation</td><td data-bbox="658 1248 1569 1450"> <ul style="list-style-type: none"> <li>Authentication and Authorization: Verify that the API properly authenticates users and authorizes access to resources.</li> <li>Data Encryption: Ensure that sensitive data is encrypted both in transit and at rest.</li> <li>Input Sanitization: Validate and sanitize user input to prevent injection attacks (e.g., SQL injection, cross-site scripting).</li> <li>Security Headers: Check that the API sets appropriate security headers (e.g., X-Frame-Options, Content-Security-Policy).</li> </ul> </td></tr> <tr> <td data-bbox="468 1450 658 1585">Reliability Validation</td><td data-bbox="658 1450 1569 1585"> <ul style="list-style-type: none"> <li>Fault Tolerance: Test the API's ability to handle failures gracefully (e.g., network outages, database errors).</li> <li>Availability: Ensure that the API is available when it's expected to be.</li> <li>Scalability: Verify that the API can handle increasing traffic load without significant performance degradation.</li> </ul> </td></tr> </table> | Functional Validation                                                                                                                                                                                                                                                                             | Input Validation                                                                                                                                                                                                                                                                                                                                                                                                                           | <ul style="list-style-type: none"> <li>Data Type: Verify that the API accepts the correct data types for all parameters (e.g., strings, numbers, booleans).</li> <li>Data Format: Ensure that the data is in the expected format (e.g., date, time, currency).</li> <li>Data Range: Check that the values of parameters fall within acceptable ranges.</li> <li>Required Fields: Verify that all mandatory fields are provided.</li> </ul> | Output Validation                                                                                                                                                                           | <ul style="list-style-type: none"> <li>Data Structure: Validate that the response data structure matches the expected format (e.g., JSON, XML).</li> <li>Data Accuracy: Check that the values in the response are correct and consistent with the input.</li> <li>Status Codes: Verify that the API returns the appropriate HTTP status codes (e.g., 200 for success, 400 for bad request, 500 for server error).</li> </ul> | Error Handling                                                                                                                                                              | <ul style="list-style-type: none"> <li>Error Messages: Ensure that the API returns meaningful error messages when invalid input is provided or unexpected errors occur.</li> <li>Error Codes: Verify that the API returns appropriate error codes to help identify the specific issue.</li> </ul> | Performance Validation                                                                                                           | <ul style="list-style-type: none"> <li>Response Time: Measure the time it takes for the API to respond to requests under different load conditions.</li> <li>Throughput: Determine the number of requests the API can handle per second.</li> <li>Resource Utilization: Monitor CPU, memory, and network usage to identify performance bottlenecks.</li> </ul> | Security Validation                                                                                                                                                                                                    | <ul style="list-style-type: none"> <li>Authentication and Authorization: Verify that the API properly authenticates users and authorizes access to resources.</li> <li>Data Encryption: Ensure that sensitive data is encrypted both in transit and at rest.</li> <li>Input Sanitization: Validate and sanitize user input to prevent injection attacks (e.g., SQL injection, cross-site scripting).</li> <li>Security Headers: Check that the API sets appropriate security headers (e.g., X-Frame-Options, Content-Security-Policy).</li> </ul> | Reliability Validation | <ul style="list-style-type: none"> <li>Fault Tolerance: Test the API's ability to handle failures gracefully (e.g., network outages, database errors).</li> <li>Availability: Ensure that the API is available when it's expected to be.</li> <li>Scalability: Verify that the API can handle increasing traffic load without significant performance degradation.</li> </ul> |
| Functional Validation                                                                                                                                                                                                     | Input Validation                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                      |                                                                                                                                                                                                                                                                                                   | <ul style="list-style-type: none"> <li>Data Type: Verify that the API accepts the correct data types for all parameters (e.g., strings, numbers, booleans).</li> <li>Data Format: Ensure that the data is in the expected format (e.g., date, time, currency).</li> <li>Data Range: Check that the values of parameters fall within acceptable ranges.</li> <li>Required Fields: Verify that all mandatory fields are provided.</li> </ul> |                                                                                                                                                                                                                                                                                                                                                                                                                                            |                                                                                                                                                                                             |                                                                                                                                                                                                                                                                                                                                                                                                                              |                                                                                                                                                                             |                                                                                                                                                                                                                                                                                                   |                                                                                                                                  |                                                                                                                                                                                                                                                                                                                                                                |                                                                                                                                                                                                                        |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                   |                        |                                                                                                                                                                                                                                                                                                                                                                               |
|                                                                                                                                                                                                                           | Output Validation                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                     |                                                                                                                                                                                                                                                                                                   | <ul style="list-style-type: none"> <li>Data Structure: Validate that the response data structure matches the expected format (e.g., JSON, XML).</li> <li>Data Accuracy: Check that the values in the response are correct and consistent with the input.</li> <li>Status Codes: Verify that the API returns the appropriate HTTP status codes (e.g., 200 for success, 400 for bad request, 500 for server error).</li> </ul>               |                                                                                                                                                                                                                                                                                                                                                                                                                                            |                                                                                                                                                                                             |                                                                                                                                                                                                                                                                                                                                                                                                                              |                                                                                                                                                                             |                                                                                                                                                                                                                                                                                                   |                                                                                                                                  |                                                                                                                                                                                                                                                                                                                                                                |                                                                                                                                                                                                                        |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                   |                        |                                                                                                                                                                                                                                                                                                                                                                               |
|                                                                                                                                                                                                                           | Error Handling                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                        | <ul style="list-style-type: none"> <li>Error Messages: Ensure that the API returns meaningful error messages when invalid input is provided or unexpected errors occur.</li> <li>Error Codes: Verify that the API returns appropriate error codes to help identify the specific issue.</li> </ul> |                                                                                                                                                                                                                                                                                                                                                                                                                                            |                                                                                                                                                                                                                                                                                                                                                                                                                                            |                                                                                                                                                                                             |                                                                                                                                                                                                                                                                                                                                                                                                                              |                                                                                                                                                                             |                                                                                                                                                                                                                                                                                                   |                                                                                                                                  |                                                                                                                                                                                                                                                                                                                                                                |                                                                                                                                                                                                                        |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                   |                        |                                                                                                                                                                                                                                                                                                                                                                               |
| Performance Validation                                                                                                                                                                                                    | <ul style="list-style-type: none"> <li>Response Time: Measure the time it takes for the API to respond to requests under different load conditions.</li> <li>Throughput: Determine the number of requests the API can handle per second.</li> <li>Resource Utilization: Monitor CPU, memory, and network usage to identify performance bottlenecks.</li> </ul>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                        |                                                                                                                                                                                                                                                                                                   |                                                                                                                                                                                                                                                                                                                                                                                                                                            |                                                                                                                                                                                                                                                                                                                                                                                                                                            |                                                                                                                                                                                             |                                                                                                                                                                                                                                                                                                                                                                                                                              |                                                                                                                                                                             |                                                                                                                                                                                                                                                                                                   |                                                                                                                                  |                                                                                                                                                                                                                                                                                                                                                                |                                                                                                                                                                                                                        |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                   |                        |                                                                                                                                                                                                                                                                                                                                                                               |
| Security Validation                                                                                                                                                                                                       | <ul style="list-style-type: none"> <li>Authentication and Authorization: Verify that the API properly authenticates users and authorizes access to resources.</li> <li>Data Encryption: Ensure that sensitive data is encrypted both in transit and at rest.</li> <li>Input Sanitization: Validate and sanitize user input to prevent injection attacks (e.g., SQL injection, cross-site scripting).</li> <li>Security Headers: Check that the API sets appropriate security headers (e.g., X-Frame-Options, Content-Security-Policy).</li> </ul>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                     |                                                                                                                                                                                                                                                                                                   |                                                                                                                                                                                                                                                                                                                                                                                                                                            |                                                                                                                                                                                                                                                                                                                                                                                                                                            |                                                                                                                                                                                             |                                                                                                                                                                                                                                                                                                                                                                                                                              |                                                                                                                                                                             |                                                                                                                                                                                                                                                                                                   |                                                                                                                                  |                                                                                                                                                                                                                                                                                                                                                                |                                                                                                                                                                                                                        |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                   |                        |                                                                                                                                                                                                                                                                                                                                                                               |
| Reliability Validation                                                                                                                                                                                                    | <ul style="list-style-type: none"> <li>Fault Tolerance: Test the API's ability to handle failures gracefully (e.g., network outages, database errors).</li> <li>Availability: Ensure that the API is available when it's expected to be.</li> <li>Scalability: Verify that the API can handle increasing traffic load without significant performance degradation.</li> </ul>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                         |                                                                                                                                                                                                                                                                                                   |                                                                                                                                                                                                                                                                                                                                                                                                                                            |                                                                                                                                                                                                                                                                                                                                                                                                                                            |                                                                                                                                                                                             |                                                                                                                                                                                                                                                                                                                                                                                                                              |                                                                                                                                                                             |                                                                                                                                                                                                                                                                                                   |                                                                                                                                  |                                                                                                                                                                                                                                                                                                                                                                |                                                                                                                                                                                                                        |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                   |                        |                                                                                                                                                                                                                                                                                                                                                                               |
| In testing a login feature, you notice that entering correct credentials works in Chrome but fails in Safari. What could be the cause, and how would you troubleshoot it?                                                 | <ul style="list-style-type: none"> <li>Clear Browser Cache and Cookies</li> <li>Check Browser Console for Errors:</li> <li>Test in Incognito Mode:</li> <li>Disable Browser Extensions:</li> <li>Check Network Requests: Use the network tab in Safari's developer tools to examine the HTTP requests and responses during the login process. Look for any unusual errors or unexpected behavior.</li> <li>Test on Different Devices and Networks</li> </ul>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                          |                                                                                                                                                                                                                                                                                                   |                                                                                                                                                                                                                                                                                                                                                                                                                                            |                                                                                                                                                                                                                                                                                                                                                                                                                                            |                                                                                                                                                                                             |                                                                                                                                                                                                                                                                                                                                                                                                                              |                                                                                                                                                                             |                                                                                                                                                                                                                                                                                                   |                                                                                                                                  |                                                                                                                                                                                                                                                                                                                                                                |                                                                                                                                                                                                                        |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                   |                        |                                                                                                                                                                                                                                                                                                                                                                               |
| You're testing a user registration form that has a date picker for selecting birth date. Users report the date sometimes shows incorrectly in the confirmation email. What could be the cause, and how would you test it? | <table border="1"> <tr> <td>Date Format Mismatch</td><td>The date picker might use a different date format than the one used in the confirmation email template. This could lead to incorrect parsing and display of the date.</td></tr> <tr> <td>Time Zone Differences</td><td>The system might not be handling time zones correctly. If the user's time zone differs from the server's time zone, the date and time might be misinterpreted.</td></tr> <tr> <td>Date Picker Bugs</td><td>The date picker component itself might have bugs or inconsistencies in how it handles date selection and formatting.</td></tr> <tr> <td>Email Template Issues</td><td>The template might not be correctly rendering the date variable or might have errors in the date formatting logic.</td></tr> <tr> <td>Server-Side Processing Errors</td><td>The server-side logic might be incorrectly processing or storing the date information.</td></tr> </table>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                | Date Format Mismatch                                                                                                                                                                                                                                                                              | The date picker might use a different date format than the one used in the confirmation email template. This could lead to incorrect parsing and display of the date.                                                                                                                                                                                                                                                                      | Time Zone Differences                                                                                                                                                                                                                                                                                                                                                                                                                      | The system might not be handling time zones correctly. If the user's time zone differs from the server's time zone, the date and time might be misinterpreted.                              | Date Picker Bugs                                                                                                                                                                                                                                                                                                                                                                                                             | The date picker component itself might have bugs or inconsistencies in how it handles date selection and formatting.                                                        | Email Template Issues                                                                                                                                                                                                                                                                             | The template might not be correctly rendering the date variable or might have errors in the date formatting logic.               | Server-Side Processing Errors                                                                                                                                                                                                                                                                                                                                  | The server-side logic might be incorrectly processing or storing the date information.                                                                                                                                 |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                   |                        |                                                                                                                                                                                                                                                                                                                                                                               |
| Date Format Mismatch                                                                                                                                                                                                      | The date picker might use a different date format than the one used in the confirmation email template. This could lead to incorrect parsing and display of the date.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                 |                                                                                                                                                                                                                                                                                                   |                                                                                                                                                                                                                                                                                                                                                                                                                                            |                                                                                                                                                                                                                                                                                                                                                                                                                                            |                                                                                                                                                                                             |                                                                                                                                                                                                                                                                                                                                                                                                                              |                                                                                                                                                                             |                                                                                                                                                                                                                                                                                                   |                                                                                                                                  |                                                                                                                                                                                                                                                                                                                                                                |                                                                                                                                                                                                                        |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                   |                        |                                                                                                                                                                                                                                                                                                                                                                               |
| Time Zone Differences                                                                                                                                                                                                     | The system might not be handling time zones correctly. If the user's time zone differs from the server's time zone, the date and time might be misinterpreted.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                        |                                                                                                                                                                                                                                                                                                   |                                                                                                                                                                                                                                                                                                                                                                                                                                            |                                                                                                                                                                                                                                                                                                                                                                                                                                            |                                                                                                                                                                                             |                                                                                                                                                                                                                                                                                                                                                                                                                              |                                                                                                                                                                             |                                                                                                                                                                                                                                                                                                   |                                                                                                                                  |                                                                                                                                                                                                                                                                                                                                                                |                                                                                                                                                                                                                        |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                   |                        |                                                                                                                                                                                                                                                                                                                                                                               |
| Date Picker Bugs                                                                                                                                                                                                          | The date picker component itself might have bugs or inconsistencies in how it handles date selection and formatting.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                  |                                                                                                                                                                                                                                                                                                   |                                                                                                                                                                                                                                                                                                                                                                                                                                            |                                                                                                                                                                                                                                                                                                                                                                                                                                            |                                                                                                                                                                                             |                                                                                                                                                                                                                                                                                                                                                                                                                              |                                                                                                                                                                             |                                                                                                                                                                                                                                                                                                   |                                                                                                                                  |                                                                                                                                                                                                                                                                                                                                                                |                                                                                                                                                                                                                        |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                   |                        |                                                                                                                                                                                                                                                                                                                                                                               |
| Email Template Issues                                                                                                                                                                                                     | The template might not be correctly rendering the date variable or might have errors in the date formatting logic.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                    |                                                                                                                                                                                                                                                                                                   |                                                                                                                                                                                                                                                                                                                                                                                                                                            |                                                                                                                                                                                                                                                                                                                                                                                                                                            |                                                                                                                                                                                             |                                                                                                                                                                                                                                                                                                                                                                                                                              |                                                                                                                                                                             |                                                                                                                                                                                                                                                                                                   |                                                                                                                                  |                                                                                                                                                                                                                                                                                                                                                                |                                                                                                                                                                                                                        |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                   |                        |                                                                                                                                                                                                                                                                                                                                                                               |
| Server-Side Processing Errors                                                                                                                                                                                             | The server-side logic might be incorrectly processing or storing the date information.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                |                                                                                                                                                                                                                                                                                                   |                                                                                                                                                                                                                                                                                                                                                                                                                                            |                                                                                                                                                                                                                                                                                                                                                                                                                                            |                                                                                                                                                                                             |                                                                                                                                                                                                                                                                                                                                                                                                                              |                                                                                                                                                                             |                                                                                                                                                                                                                                                                                                   |                                                                                                                                  |                                                                                                                                                                                                                                                                                                                                                                |                                                                                                                                                                                                                        |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                   |                        |                                                                                                                                                                                                                                                                                                                                                                               |

|                                             |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                 |                             |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                        |                        |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                         |                              |                                                                                                                                                                                                                                                                                                                                                                                   |                                 |                                                                                                                                                                                  |                      |                                                                                                                                                                             |                             |                                                                                                                                                                                                      |                  |                                                                                                                                                                                                     |                             |                                                                                                                                                                                                        |                |                                                                                                                            |                              |                           |
|---------------------------------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-----------------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|------------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|------------------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|---------------------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|----------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-----------------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-----------------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|----------------|----------------------------------------------------------------------------------------------------------------------------|------------------------------|---------------------------|
|                                             | <p><b>Testing Approach:</b></p> <p><b>Test with Different Date Formats:</b><br/>Use various date formats (e.g., DD/MM/YYYY, MM/DD/YYYY, YYYY-MM-DD) to see if the issue persists.<br/>Pay attention to how the date is displayed in the confirmation email.</p> <p><b>Test with Different Time Zones:</b><br/>Use a virtual machine or browser extension to simulate different time zones.<br/>Verify that the date is displayed correctly in the confirmation email.</p> <p><b>Test with Edge Cases:</b><br/>Use dates at the beginning and end of months and years to check for boundary conditions.<br/>Test with leap years and other special dates.</p> <p><b>Inspect Network Requests:</b><br/>Use browser developer tools to examine the network requests and responses.<br/>Look for any errors or inconsistencies in the data being sent to the server and received from the server.</p>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                               |                             |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                        |                        |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                         |                              |                                                                                                                                                                                                                                                                                                                                                                                   |                                 |                                                                                                                                                                                  |                      |                                                                                                                                                                             |                             |                                                                                                                                                                                                      |                  |                                                                                                                                                                                                     |                             |                                                                                                                                                                                                        |                |                                                                                                                            |                              |                           |
| How will you test file upload functionality | <table border="1"> <tr> <td>Valid File Upload</td><td>Upload a file of an allowed type (e.g., .jpg, .png, .pdf) and within the specified size limit.<br/>Verify that the file is uploaded successfully and displayed correctly.</td></tr> <tr> <td>Invalid File Type</td><td>Attempt to upload a file with a disallowed file type (e.g., .exe, .bat).<br/>Verify that an appropriate error message is displayed, preventing the upload.</td></tr> <tr> <td>File Size Limit</td><td>Upload a file that exceeds the maximum allowed size.<br/>Verify that an error message is displayed, indicating the size limit violation.</td></tr> <tr> <td>Special Characters in File Name</td><td>Upload a file with special characters in its name (e.g., file@#\$%^&amp;*.txt).<br/>Ensure that the file is uploaded correctly, and the special characters are handled appropriately.</td></tr> <tr> <td>Multiple File Upload</td><td>If the application supports multiple file uploads, try uploading multiple files simultaneously.<br/>Verify that all files are uploaded successfully and displayed correctly.</td></tr> <tr> <td>Network Connectivity Issues</td><td>Simulate network interruptions or slow internet speeds.<br/>Verify that the application handles these scenarios gracefully, either by retrying the upload or displaying an appropriate error message.</td></tr> <tr> <td>Security Testing</td><td>Attempt to upload malicious files (e.g., files containing viruses or malware).<br/>Ensure that the application has robust security measures in place to prevent such uploads and protect the system.</td></tr> <tr> <td>File Download Functionality</td><td>Once files are uploaded, test the download functionality.<br/>Verify that files can be downloaded correctly and without corruption.<br/>Check if the downloaded file matches the original uploaded file.</td></tr> <tr> <td>Mobile Testing</td><td>If the application supports mobile devices, test the file upload functionality on various mobile platforms (iOS, Android).</td></tr> <tr> <td>Browser and platform testing</td><td>Test on safari, linux mac</td></tr> </table>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                               | Valid File Upload           | Upload a file of an allowed type (e.g., .jpg, .png, .pdf) and within the specified size limit.<br>Verify that the file is uploaded successfully and displayed correctly.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                               | Invalid File Type      | Attempt to upload a file with a disallowed file type (e.g., .exe, .bat).<br>Verify that an appropriate error message is displayed, preventing the upload.                                                                                                                                                                                                                                                                                                                                                               | File Size Limit              | Upload a file that exceeds the maximum allowed size.<br>Verify that an error message is displayed, indicating the size limit violation.                                                                                                                                                                                                                                           | Special Characters in File Name | Upload a file with special characters in its name (e.g., file@#\$%^&*.txt).<br>Ensure that the file is uploaded correctly, and the special characters are handled appropriately. | Multiple File Upload | If the application supports multiple file uploads, try uploading multiple files simultaneously.<br>Verify that all files are uploaded successfully and displayed correctly. | Network Connectivity Issues | Simulate network interruptions or slow internet speeds.<br>Verify that the application handles these scenarios gracefully, either by retrying the upload or displaying an appropriate error message. | Security Testing | Attempt to upload malicious files (e.g., files containing viruses or malware).<br>Ensure that the application has robust security measures in place to prevent such uploads and protect the system. | File Download Functionality | Once files are uploaded, test the download functionality.<br>Verify that files can be downloaded correctly and without corruption.<br>Check if the downloaded file matches the original uploaded file. | Mobile Testing | If the application supports mobile devices, test the file upload functionality on various mobile platforms (iOS, Android). | Browser and platform testing | Test on safari, linux mac |
| Valid File Upload                           | Upload a file of an allowed type (e.g., .jpg, .png, .pdf) and within the specified size limit.<br>Verify that the file is uploaded successfully and displayed correctly.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                        |                             |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                        |                        |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                         |                              |                                                                                                                                                                                                                                                                                                                                                                                   |                                 |                                                                                                                                                                                  |                      |                                                                                                                                                                             |                             |                                                                                                                                                                                                      |                  |                                                                                                                                                                                                     |                             |                                                                                                                                                                                                        |                |                                                                                                                            |                              |                           |
| Invalid File Type                           | Attempt to upload a file with a disallowed file type (e.g., .exe, .bat).<br>Verify that an appropriate error message is displayed, preventing the upload.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                       |                             |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                        |                        |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                         |                              |                                                                                                                                                                                                                                                                                                                                                                                   |                                 |                                                                                                                                                                                  |                      |                                                                                                                                                                             |                             |                                                                                                                                                                                                      |                  |                                                                                                                                                                                                     |                             |                                                                                                                                                                                                        |                |                                                                                                                            |                              |                           |
| File Size Limit                             | Upload a file that exceeds the maximum allowed size.<br>Verify that an error message is displayed, indicating the size limit violation.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                         |                             |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                        |                        |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                         |                              |                                                                                                                                                                                                                                                                                                                                                                                   |                                 |                                                                                                                                                                                  |                      |                                                                                                                                                                             |                             |                                                                                                                                                                                                      |                  |                                                                                                                                                                                                     |                             |                                                                                                                                                                                                        |                |                                                                                                                            |                              |                           |
| Special Characters in File Name             | Upload a file with special characters in its name (e.g., file@#\$%^&*.txt).<br>Ensure that the file is uploaded correctly, and the special characters are handled appropriately.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                |                             |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                        |                        |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                         |                              |                                                                                                                                                                                                                                                                                                                                                                                   |                                 |                                                                                                                                                                                  |                      |                                                                                                                                                                             |                             |                                                                                                                                                                                                      |                  |                                                                                                                                                                                                     |                             |                                                                                                                                                                                                        |                |                                                                                                                            |                              |                           |
| Multiple File Upload                        | If the application supports multiple file uploads, try uploading multiple files simultaneously.<br>Verify that all files are uploaded successfully and displayed correctly.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                     |                             |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                        |                        |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                         |                              |                                                                                                                                                                                                                                                                                                                                                                                   |                                 |                                                                                                                                                                                  |                      |                                                                                                                                                                             |                             |                                                                                                                                                                                                      |                  |                                                                                                                                                                                                     |                             |                                                                                                                                                                                                        |                |                                                                                                                            |                              |                           |
| Network Connectivity Issues                 | Simulate network interruptions or slow internet speeds.<br>Verify that the application handles these scenarios gracefully, either by retrying the upload or displaying an appropriate error message.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                            |                             |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                        |                        |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                         |                              |                                                                                                                                                                                                                                                                                                                                                                                   |                                 |                                                                                                                                                                                  |                      |                                                                                                                                                                             |                             |                                                                                                                                                                                                      |                  |                                                                                                                                                                                                     |                             |                                                                                                                                                                                                        |                |                                                                                                                            |                              |                           |
| Security Testing                            | Attempt to upload malicious files (e.g., files containing viruses or malware).<br>Ensure that the application has robust security measures in place to prevent such uploads and protect the system.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                             |                             |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                        |                        |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                         |                              |                                                                                                                                                                                                                                                                                                                                                                                   |                                 |                                                                                                                                                                                  |                      |                                                                                                                                                                             |                             |                                                                                                                                                                                                      |                  |                                                                                                                                                                                                     |                             |                                                                                                                                                                                                        |                |                                                                                                                            |                              |                           |
| File Download Functionality                 | Once files are uploaded, test the download functionality.<br>Verify that files can be downloaded correctly and without corruption.<br>Check if the downloaded file matches the original uploaded file.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                          |                             |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                        |                        |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                         |                              |                                                                                                                                                                                                                                                                                                                                                                                   |                                 |                                                                                                                                                                                  |                      |                                                                                                                                                                             |                             |                                                                                                                                                                                                      |                  |                                                                                                                                                                                                     |                             |                                                                                                                                                                                                        |                |                                                                                                                            |                              |                           |
| Mobile Testing                              | If the application supports mobile devices, test the file upload functionality on various mobile platforms (iOS, Android).                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                      |                             |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                        |                        |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                         |                              |                                                                                                                                                                                                                                                                                                                                                                                   |                                 |                                                                                                                                                                                  |                      |                                                                                                                                                                             |                             |                                                                                                                                                                                                      |                  |                                                                                                                                                                                                     |                             |                                                                                                                                                                                                        |                |                                                                                                                            |                              |                           |
| Browser and platform testing                | Test on safari, linux mac                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                       |                             |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                        |                        |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                         |                              |                                                                                                                                                                                                                                                                                                                                                                                   |                                 |                                                                                                                                                                                  |                      |                                                                                                                                                                             |                             |                                                                                                                                                                                                      |                  |                                                                                                                                                                                                     |                             |                                                                                                                                                                                                        |                |                                                                                                                            |                              |                           |
| Test Scenarios for lift                     | <ul style="list-style-type: none"> <li>• Are we testing a single lift or multiple lifts?</li> <li>• What type of lift are we testing, is it a passenger lift, goods lift, service lift, express lift for VIPs, Ski lift or any other type of lift?</li> <li>• Where would the lift be installed, is it going to be installed in a residential apartment or private residence?</li> <li>• How many floors will the lift intend to service?</li> <li>• What country is the lift intended to be installed in, whether the safety instructions is given in the language of the target users?</li> <li>• What is the capacity of the lift?</li> <li>• What are the health and safety or regulatory requirements that the lift needs to fulfill?</li> </ul> <table border="1"> <tr> <td><b>Functional Scenarios</b></td><td> <ul style="list-style-type: none"> <li>• Verify the lift moves up and down correctly between floors.</li> <li>• Verify the lift stops at the requested floors.</li> <li>• Verify the lift responds to floor button inputs inside the lift.</li> <li>• Verify the lift responds to floor call buttons outside on each floor.</li> <li>• Verify the lift doors open and close correctly at each floor.</li> <li>• Verify the lift doesn't move while doors are open.</li> <li>• Verify the lift's capacity limit and its response when overloaded.</li> <li>• Verify the emergency stop button works correctly.</li> <li>• Verify that the lift resumes operation after an emergency stop is cleared.</li> <li>• Verify the lift's behavior when multiple floors are requested in a sequence.</li> </ul> </td></tr> <tr> <td><b>UI/UX Scenarios</b></td><td> <ul style="list-style-type: none"> <li>• Verify the floor number display inside and outside the lift updates correctly.</li> <li>• Verify that all button lights inside and outside the lift illuminate upon being pressed.</li> <li>• Verify that the "Door Open" and "Door Close" buttons work as expected.</li> <li>• Verify the display or audio announcement for the current floor is clear and accurate.</li> <li>• Verify the visibility and accessibility of emergency instructions inside the lift.</li> </ul> </td></tr> <tr> <td><b>Performance Scenarios</b></td><td> <ul style="list-style-type: none"> <li>• Verify the lift's response time from floor request to arrival.</li> <li>• Verify the lift's travel time between floors under normal load.</li> <li>• Verify the lift's behavior under peak load (e.g., during rush hours).</li> <li>• Verify the lift's performance during power fluctuations or transitions to backup power.</li> </ul> </td></tr> <tr> <td><b>Safety Scenarios</b></td><td> <ul style="list-style-type: none"> <li>• Verify the lift does not operate if the weight exceeds the safety threshold.</li> </ul> </td></tr> </table> | <b>Functional Scenarios</b> | <ul style="list-style-type: none"> <li>• Verify the lift moves up and down correctly between floors.</li> <li>• Verify the lift stops at the requested floors.</li> <li>• Verify the lift responds to floor button inputs inside the lift.</li> <li>• Verify the lift responds to floor call buttons outside on each floor.</li> <li>• Verify the lift doors open and close correctly at each floor.</li> <li>• Verify the lift doesn't move while doors are open.</li> <li>• Verify the lift's capacity limit and its response when overloaded.</li> <li>• Verify the emergency stop button works correctly.</li> <li>• Verify that the lift resumes operation after an emergency stop is cleared.</li> <li>• Verify the lift's behavior when multiple floors are requested in a sequence.</li> </ul> | <b>UI/UX Scenarios</b> | <ul style="list-style-type: none"> <li>• Verify the floor number display inside and outside the lift updates correctly.</li> <li>• Verify that all button lights inside and outside the lift illuminate upon being pressed.</li> <li>• Verify that the "Door Open" and "Door Close" buttons work as expected.</li> <li>• Verify the display or audio announcement for the current floor is clear and accurate.</li> <li>• Verify the visibility and accessibility of emergency instructions inside the lift.</li> </ul> | <b>Performance Scenarios</b> | <ul style="list-style-type: none"> <li>• Verify the lift's response time from floor request to arrival.</li> <li>• Verify the lift's travel time between floors under normal load.</li> <li>• Verify the lift's behavior under peak load (e.g., during rush hours).</li> <li>• Verify the lift's performance during power fluctuations or transitions to backup power.</li> </ul> | <b>Safety Scenarios</b>         | <ul style="list-style-type: none"> <li>• Verify the lift does not operate if the weight exceeds the safety threshold.</li> </ul>                                                 |                      |                                                                                                                                                                             |                             |                                                                                                                                                                                                      |                  |                                                                                                                                                                                                     |                             |                                                                                                                                                                                                        |                |                                                                                                                            |                              |                           |
| <b>Functional Scenarios</b>                 | <ul style="list-style-type: none"> <li>• Verify the lift moves up and down correctly between floors.</li> <li>• Verify the lift stops at the requested floors.</li> <li>• Verify the lift responds to floor button inputs inside the lift.</li> <li>• Verify the lift responds to floor call buttons outside on each floor.</li> <li>• Verify the lift doors open and close correctly at each floor.</li> <li>• Verify the lift doesn't move while doors are open.</li> <li>• Verify the lift's capacity limit and its response when overloaded.</li> <li>• Verify the emergency stop button works correctly.</li> <li>• Verify that the lift resumes operation after an emergency stop is cleared.</li> <li>• Verify the lift's behavior when multiple floors are requested in a sequence.</li> </ul>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                          |                             |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                        |                        |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                         |                              |                                                                                                                                                                                                                                                                                                                                                                                   |                                 |                                                                                                                                                                                  |                      |                                                                                                                                                                             |                             |                                                                                                                                                                                                      |                  |                                                                                                                                                                                                     |                             |                                                                                                                                                                                                        |                |                                                                                                                            |                              |                           |
| <b>UI/UX Scenarios</b>                      | <ul style="list-style-type: none"> <li>• Verify the floor number display inside and outside the lift updates correctly.</li> <li>• Verify that all button lights inside and outside the lift illuminate upon being pressed.</li> <li>• Verify that the "Door Open" and "Door Close" buttons work as expected.</li> <li>• Verify the display or audio announcement for the current floor is clear and accurate.</li> <li>• Verify the visibility and accessibility of emergency instructions inside the lift.</li> </ul>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                         |                             |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                        |                        |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                         |                              |                                                                                                                                                                                                                                                                                                                                                                                   |                                 |                                                                                                                                                                                  |                      |                                                                                                                                                                             |                             |                                                                                                                                                                                                      |                  |                                                                                                                                                                                                     |                             |                                                                                                                                                                                                        |                |                                                                                                                            |                              |                           |
| <b>Performance Scenarios</b>                | <ul style="list-style-type: none"> <li>• Verify the lift's response time from floor request to arrival.</li> <li>• Verify the lift's travel time between floors under normal load.</li> <li>• Verify the lift's behavior under peak load (e.g., during rush hours).</li> <li>• Verify the lift's performance during power fluctuations or transitions to backup power.</li> </ul>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                               |                             |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                        |                        |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                         |                              |                                                                                                                                                                                                                                                                                                                                                                                   |                                 |                                                                                                                                                                                  |                      |                                                                                                                                                                             |                             |                                                                                                                                                                                                      |                  |                                                                                                                                                                                                     |                             |                                                                                                                                                                                                        |                |                                                                                                                            |                              |                           |
| <b>Safety Scenarios</b>                     | <ul style="list-style-type: none"> <li>• Verify the lift does not operate if the weight exceeds the safety threshold.</li> </ul>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                |                             |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                        |                        |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                         |                              |                                                                                                                                                                                                                                                                                                                                                                                   |                                 |                                                                                                                                                                                  |                      |                                                                                                                                                                             |                             |                                                                                                                                                                                                      |                  |                                                                                                                                                                                                     |                             |                                                                                                                                                                                                        |                |                                                                                                                            |                              |                           |

|                                                                                                                               |                                                                                                                                                                                                                                                                                                                                                                                                                                                 |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                |                 |                                |            |                                                                          |               |                                                          |                  |                                                                |               |                                      |           |                                                       |                 |                                                                  |        |                                                                                                                                                                                                                                                                                                                                                                                                                                                 |
|-------------------------------------------------------------------------------------------------------------------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-----------------|--------------------------------|------------|--------------------------------------------------------------------------|---------------|----------------------------------------------------------|------------------|----------------------------------------------------------------|---------------|--------------------------------------|-----------|-------------------------------------------------------|-----------------|------------------------------------------------------------------|--------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
|                                                                                                                               |                                                                                                                                                                                                                                                                                                                                                                                                                                                 | <p>Verify the lift stops if there is an obstruction while the door is closing.<br/>     Verify the alarm system works correctly when triggered.<br/>     Verify that the lift operates correctly during a power outage on backup power.<br/>     Verify that the lift halts between floors in case of a malfunction, and emergency protocols are activated.</p>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                |                 |                                |            |                                                                          |               |                                                          |                  |                                                                |               |                                      |           |                                                       |                 |                                                                  |        |                                                                                                                                                                                                                                                                                                                                                                                                                                                 |
|                                                                                                                               | <b>Negative Scenarios</b>                                                                                                                                                                                                                                                                                                                                                                                                                       | <p>Verify the lift's response if a button is pressed repeatedly.<br/>     Verify the lift's response to invalid inputs (e.g., pressing buttons for non-existent floors).<br/>     Verify the lift doesn't stop unnecessarily when no floor requests are made.<br/>     Verify the system doesn't allow conflicting inputs (e.g., simultaneous "Door Open" and "Door Close").<br/>     Verify the lift's behavior when an external or internal button fails.</p>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                |                 |                                |            |                                                                          |               |                                                          |                  |                                                                |               |                                      |           |                                                       |                 |                                                                  |        |                                                                                                                                                                                                                                                                                                                                                                                                                                                 |
| How would you test the appointment scheduling feature                                                                         |                                                                                                                                                                                                                                                                                                                                                                                                                                                 | <p>Happy Path: Test scheduling appointments for available slots and verify confirmation notifications (email/SMS).<br/>     Edge Cases: Attempt to book appointments at the same time for the same doctor to check for conflict resolution.<br/>     Time Zone Testing: Verify the application handles different time zones for teleconsultations.<br/>     Integration Testing: Check integration with calendar APIs (Google Calendar, Outlook) for appointment syncing.<br/>     Error Handling: Verify appropriate error messages when slots are full or invalid date/time is entered.</p>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                  |                 |                                |            |                                                                          |               |                                                          |                  |                                                                |               |                                      |           |                                                       |                 |                                                                  |        |                                                                                                                                                                                                                                                                                                                                                                                                                                                 |
| Non Functional Cases for Adding beneficiary                                                                                   |                                                                                                                                                                                                                                                                                                                                                                                                                                                 | <p>Security Test Cases:<br/>     Account number masking: Verify that the account number field is masked (e.g., with asterisks or bullets) to protect sensitive information.<br/>     OTP generation and validation: Enter valid data and check if an OTP is generated.<br/>     Verify that the system prompts for the OTP and does not add the beneficiary without it.<br/>     Enter an invalid OTP and confirm that the beneficiary is not added and an error message is displayed.<br/>     Check if the OTP is not entered within the time limit, the OTP is invalidated, and the beneficiary is not added.</p> <p>Performance Test Cases:<br/>     Page load time: Measure the time it takes to load each page involved in the beneficiary addition process.<br/>     Concurrent access: Open the same account in two different browsers and try to add the same beneficiary simultaneously.<br/>     Verify if the system handles this scenario correctly and prevents duplicate entries.</p> <p>Additional Considerations:<br/>     Error Handling: Test how the system handles various error scenarios, such as invalid input data, network errors, or system failures.<br/>     User Interface: Ensure that the user interface is clear, intuitive, and provides appropriate feedback during the beneficiary addition process.<br/>     Security Best Practices: Verify that the system adheres to security best practices, such as data encryption, secure authentication, and input validation.</p>                                                                                                |                 |                                |            |                                                                          |               |                                                          |                  |                                                                |               |                                      |           |                                                       |                 |                                                                  |        |                                                                                                                                                                                                                                                                                                                                                                                                                                                 |
| Points for test cases                                                                                                         |                                                                                                                                                                                                                                                                                                                                                                                                                                                 | Cover Functional, Performance, Security, UI, Cross browser, cross device scenarios                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                             |                 |                                |            |                                                                          |               |                                                          |                  |                                                                |               |                                      |           |                                                       |                 |                                                                  |        |                                                                                                                                                                                                                                                                                                                                                                                                                                                 |
| Most Memorable Bug                                                                                                            |                                                                                                                                                                                                                                                                                                                                                                                                                                                 | Cancel Replace from holding screen was not loading data for RIA Accounts only.<br>In SQL Date field was varchar to between was not working for inter months                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                    |                 |                                |            |                                                                          |               |                                                          |                  |                                                                |               |                                      |           |                                                       |                 |                                                                  |        |                                                                                                                                                                                                                                                                                                                                                                                                                                                 |
| Which exception is shown in selenium when there is a delay in the loading time of the elements with which we are interacting? |                                                                                                                                                                                                                                                                                                                                                                                                                                                 | Without any wait: NOSUCHELEMENTEXCEPTION else TIMEOUTEXCEPTION                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                 |                 |                                |            |                                                                          |               |                                                          |                  |                                                                |               |                                      |           |                                                       |                 |                                                                  |        |                                                                                                                                                                                                                                                                                                                                                                                                                                                 |
| How do you manage flaky tests in an automation framework to ensure reliable execution, especially in CI/CD pipelines          |                                                                                                                                                                                                                                                                                                                                                                                                                                                 | When tests are flaky in CI/CD pipelines, despite running smoothly in local environments, I first investigate root causes through enhanced logging and screenshots to capture runtime details. Often, flakiness in headless mode arises from resolution or element load issues, which I address by explicitly setting the browser resolution and using adaptive waits. I also consider retry mechanisms, adding waits and test isolation strategies to further stabilize execution                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                              |                 |                                |            |                                                                          |               |                                                          |                  |                                                                |               |                                      |           |                                                       |                 |                                                                  |        |                                                                                                                                                                                                                                                                                                                                                                                                                                                 |
| Test Plan                                                                                                                     |                                                                                                                                                                                                                                                                                                                                                                                                                                                 | <p>Detailed document that outlines the scope, approach, resources, and schedule of intended testing activities. It provides a roadmap for how testing will be conducted, detailing the specific tasks, who will perform them, when they will be done, and what resources are required. The test plan is tactical, focusing on the execution of the strategy by describing the "how" of testing.</p> <table border="1"> <tr> <td>Test Objectives</td><td>The specific goals of testing.</td></tr> <tr> <td>Test Scope</td><td>Detailed description of what features or functionalities will be tested.</td></tr> <tr> <td>Test Approach</td><td>Detailed approach to testing based on the test strategy.</td></tr> <tr> <td>Test Environment</td><td>Specific configurations and settings for the test environment.</td></tr> <tr> <td>Test Schedule</td><td>Timeline for the testing activities.</td></tr> <tr> <td>Resources</td><td>Required tools, people, and environments for testing.</td></tr> <tr> <td>Risk Management</td><td>Identification of potential risks and strategies for mitigation.</td></tr> <tr> <td>Sample</td><td> <h2>1. Introduction</h2> <h3>1.1 Objective</h3> <p>This test plan aims to validate the functionality, performance, security, and reliability of the trading platform that supports:</p> <ul style="list-style-type: none"> <li>• Equities Trading</li> <li>• Options Trading</li> <li>• Complex Options Trading (e.g., spreads, butterflies, condors)</li> <li>• Fixed Income Trading (Bonds, Treasury Bills, Corporate Bonds, etc.)</li> </ul> </td></tr> </table> | Test Objectives | The specific goals of testing. | Test Scope | Detailed description of what features or functionalities will be tested. | Test Approach | Detailed approach to testing based on the test strategy. | Test Environment | Specific configurations and settings for the test environment. | Test Schedule | Timeline for the testing activities. | Resources | Required tools, people, and environments for testing. | Risk Management | Identification of potential risks and strategies for mitigation. | Sample | <h2>1. Introduction</h2> <h3>1.1 Objective</h3> <p>This test plan aims to validate the functionality, performance, security, and reliability of the trading platform that supports:</p> <ul style="list-style-type: none"> <li>• Equities Trading</li> <li>• Options Trading</li> <li>• Complex Options Trading (e.g., spreads, butterflies, condors)</li> <li>• Fixed Income Trading (Bonds, Treasury Bills, Corporate Bonds, etc.)</li> </ul> |
| Test Objectives                                                                                                               | The specific goals of testing.                                                                                                                                                                                                                                                                                                                                                                                                                  |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                |                 |                                |            |                                                                          |               |                                                          |                  |                                                                |               |                                      |           |                                                       |                 |                                                                  |        |                                                                                                                                                                                                                                                                                                                                                                                                                                                 |
| Test Scope                                                                                                                    | Detailed description of what features or functionalities will be tested.                                                                                                                                                                                                                                                                                                                                                                        |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                |                 |                                |            |                                                                          |               |                                                          |                  |                                                                |               |                                      |           |                                                       |                 |                                                                  |        |                                                                                                                                                                                                                                                                                                                                                                                                                                                 |
| Test Approach                                                                                                                 | Detailed approach to testing based on the test strategy.                                                                                                                                                                                                                                                                                                                                                                                        |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                |                 |                                |            |                                                                          |               |                                                          |                  |                                                                |               |                                      |           |                                                       |                 |                                                                  |        |                                                                                                                                                                                                                                                                                                                                                                                                                                                 |
| Test Environment                                                                                                              | Specific configurations and settings for the test environment.                                                                                                                                                                                                                                                                                                                                                                                  |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                |                 |                                |            |                                                                          |               |                                                          |                  |                                                                |               |                                      |           |                                                       |                 |                                                                  |        |                                                                                                                                                                                                                                                                                                                                                                                                                                                 |
| Test Schedule                                                                                                                 | Timeline for the testing activities.                                                                                                                                                                                                                                                                                                                                                                                                            |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                |                 |                                |            |                                                                          |               |                                                          |                  |                                                                |               |                                      |           |                                                       |                 |                                                                  |        |                                                                                                                                                                                                                                                                                                                                                                                                                                                 |
| Resources                                                                                                                     | Required tools, people, and environments for testing.                                                                                                                                                                                                                                                                                                                                                                                           |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                |                 |                                |            |                                                                          |               |                                                          |                  |                                                                |               |                                      |           |                                                       |                 |                                                                  |        |                                                                                                                                                                                                                                                                                                                                                                                                                                                 |
| Risk Management                                                                                                               | Identification of potential risks and strategies for mitigation.                                                                                                                                                                                                                                                                                                                                                                                |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                |                 |                                |            |                                                                          |               |                                                          |                  |                                                                |               |                                      |           |                                                       |                 |                                                                  |        |                                                                                                                                                                                                                                                                                                                                                                                                                                                 |
| Sample                                                                                                                        | <h2>1. Introduction</h2> <h3>1.1 Objective</h3> <p>This test plan aims to validate the functionality, performance, security, and reliability of the trading platform that supports:</p> <ul style="list-style-type: none"> <li>• Equities Trading</li> <li>• Options Trading</li> <li>• Complex Options Trading (e.g., spreads, butterflies, condors)</li> <li>• Fixed Income Trading (Bonds, Treasury Bills, Corporate Bonds, etc.)</li> </ul> |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                |                 |                                |            |                                                                          |               |                                                          |                  |                                                                |               |                                      |           |                                                       |                 |                                                                  |        |                                                                                                                                                                                                                                                                                                                                                                                                                                                 |

Both **manual** and **automated** testing approaches will be leveraged to ensure high quality and compliance with financial regulations.

## 1.2 Scope

The test scope includes:

- **Trading functionalities** (Order Placement, Execution, Modifications, Cancellations)
- **Market Data Validation**
- **Order Matching and Settlement**
- **Risk Management** (Margin, Limits, Position Sizing)
- **Regulatory Compliance** (SEC, FINRA, MiFID II, etc.)
- **API Testing** (FIX, REST, WebSocket)
- **UI and Mobile App Testing**
- **Performance Testing** (Load, Stress, Failover)
- **Security and Penetration Testing**
- **Database and Backend Testing**
- **Integration with Payment Gateways, Market Data Feeds, and Clearing Houses**

## 2. Test Strategy

### 2.1 Test Approach

A mix of **Manual** and **Automation Testing** will be used:

- **Manual Testing** for exploratory, regulatory, usability, and edge cases.
- **Automation Testing** for regression, high-frequency tests, API validation, and performance.

### 2.2 Testing Types

| Test Type                            | Description                                                                            |
|--------------------------------------|----------------------------------------------------------------------------------------|
| <b>Functional Testing</b>            | Verify trading workflows (Order Lifecycle, Execution, Cancel, Modify, Market Depth)    |
| <b>Integration Testing</b>           | Validate interactions with market data providers, FIX gateways, clearing houses        |
| <b>UI Testing</b>                    | Validate user workflows across web and mobile (ReactJS, iOS, Android)                  |
| <b>API Testing</b>                   | Test REST, WebSockets, and FIX APIs for trading and market data                        |
| <b>Performance Testing</b>           | Measure order execution speed, latency, and system scalability                         |
| <b>Security Testing</b>              | Penetration testing, API security, encryption, and fraud detection                     |
| <b>Regulatory Compliance Testing</b> | Verify compliance with financial regulations                                           |
| <b>Database Testing</b>              | Validate trade persistence, transaction rollbacks, and reconciliation                  |
| <b>Failover &amp; Recovery</b>       | Test system recovery under failure conditions (network disconnects, database failures) |

### 2.3 Automation Strategy

- **Test Automation Framework:** Selenium (UI), Playwright (UI), RestAssured (API), JMeter (Performance)
- **Scripting Language:** Java + Cucumber (BDD for business-readable tests)
- **Test Execution:** CI/CD integration with GitLab
- **Browser Coverage:** Chrome, Edge, Safari, Firefox (for cross-browser compatibility)
- **Mobile Testing:** Appium (for native iOS and Android apps)
- **Performance Testing:** JMeter (Load, Stress, Soak, Spike Testing)
- **Security Testing:** OWASP ZAP, Burp Suite, Static Code Analysis (SonarQube)

## 3. Test Environment Setup

### 3.1 Infrastructure

| Component                | Details                                                 |
|--------------------------|---------------------------------------------------------|
| <b>Trading Platform</b>  | Web-based, Mobile App                                   |
| <b>Backend Services</b>  | Microservices, Cloud-based                              |
| <b>Market Data Feeds</b> | Real-time, FIX Protocol, WebSockets                     |
| <b>Trading API</b>       | REST API, FIX API                                       |
| <b>Database</b>          | MySQL/PostgreSQL for transactions, Redis for caching    |
| <b>Test Execution</b>    | Selenium Grid, Kubernetes-based containerized execution |

### 3.2 Test Data Management

- **Synthetic Test Data** for different order types (Market, Limit, Stop, OCO, IOC, GTC)
- **Historical Market Data** for backtesting and simulation
- **Mock Accounts** with different margin limits and portfolios

## 4. Test Execution & Reporting

### 4.1 Test Execution Plan

| Phase                      | Activities                                   |
|----------------------------|----------------------------------------------|
| <b>Sprint Testing</b>      | Agile-based execution, story-level testing   |
| <b>Regression Testing</b>  | Automated suite execution in CI/CD           |
| <b>Performance Testing</b> | Monthly execution under high-load conditions |

|                                                                                                                | <p><b>Security Audits</b>      Quarterly security assessments</p> <h2>4.2 Defect Management</h2> <ul style="list-style-type: none"> <li>• <b>Bug Tracking Tool:</b> JIRA with Zephyr/Xray</li> <li>• <b>Severity Classification:</b> <ul style="list-style-type: none"> <li>◦ <b>Critical:</b> Trade execution failures, incorrect pricing</li> <li>◦ <b>High:</b> UI crashes, API response delays</li> <li>◦ <b>Medium:</b> UI glitches, formatting errors</li> <li>◦ <b>Low:</b> Cosmetic issues</li> </ul> </li> </ul> <h2>4.3 Reporting &amp; Metrics</h2> <ul style="list-style-type: none"> <li>• Daily Test Execution Reports (Pass/Fail, Execution Time)</li> <li>• Defect Leakage Rate</li> <li>• Performance Benchmarks</li> <li>• Automation Coverage Reports</li> </ul> <h2>5. Risks &amp; Mitigation</h2> <table border="1"> <thead> <tr> <th>Risk</th><th>Mitigation</th></tr> </thead> <tbody> <tr> <td><b>Market Data Latency</b></td><td>Stress test order flow under peak conditions</td></tr> <tr> <td><b>Regulatory Non-Compliance</b></td><td>Conduct regular audits with compliance teams</td></tr> <tr> <td><b>Order Execution Failures</b></td><td>Implement failover testing and database recovery tests</td></tr> <tr> <td><b>Test Data Complexity</b></td><td>Use data masking and automated data creation</td></tr> </tbody> </table> <p>This plan ensures <b>comprehensive test coverage</b> for a high-stakes trading system. Would you like me to refine any section further?</p> | Risk                  | Mitigation                             | <b>Market Data Latency</b> | Stress test order flow under peak conditions | <b>Regulatory Non-Compliance</b> | Conduct regular audits with compliance teams                                       | <b>Order Execution Failures</b> | Implement failover testing and database recovery tests | <b>Test Data Complexity</b> | Use data masking and automated data creation        |                                 |                                                                                                                      |
|----------------------------------------------------------------------------------------------------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-----------------------|----------------------------------------|----------------------------|----------------------------------------------|----------------------------------|------------------------------------------------------------------------------------|---------------------------------|--------------------------------------------------------|-----------------------------|-----------------------------------------------------|---------------------------------|----------------------------------------------------------------------------------------------------------------------|
| Risk                                                                                                           | Mitigation                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                       |                       |                                        |                            |                                              |                                  |                                                                                    |                                 |                                                        |                             |                                                     |                                 |                                                                                                                      |
| <b>Market Data Latency</b>                                                                                     | Stress test order flow under peak conditions                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                     |                       |                                        |                            |                                              |                                  |                                                                                    |                                 |                                                        |                             |                                                     |                                 |                                                                                                                      |
| <b>Regulatory Non-Compliance</b>                                                                               | Conduct regular audits with compliance teams                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                     |                       |                                        |                            |                                              |                                  |                                                                                    |                                 |                                                        |                             |                                                     |                                 |                                                                                                                      |
| <b>Order Execution Failures</b>                                                                                | Implement failover testing and database recovery tests                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                           |                       |                                        |                            |                                              |                                  |                                                                                    |                                 |                                                        |                             |                                                     |                                 |                                                                                                                      |
| <b>Test Data Complexity</b>                                                                                    | Use data masking and automated data creation                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                     |                       |                                        |                            |                                              |                                  |                                                                                    |                                 |                                                        |                             |                                                     |                                 |                                                                                                                      |
| Test Strategy                                                                                                  | <p>High-level document that outlines the general approach and objectives for testing. It is the "big picture" view, setting the direction and purpose for testing activities. This document typically covers the overarching goals, methodologies, tools, environments, and other factors influencing how testing will be performed.</p> <p>Provides the "why" and "what" of testing. It is more static and rarely changes throughout the project unless there is a significant change in project goals or scope. <b>Test Strategy</b> is a <b>company-wide testing policy</b>, whereas a <b>Test Plan</b> is a <b>customized playbook for a specific project</b> based on that policy</p> <table border="1"> <tr> <td>Major Test Objectives</td><td>What are the primary goals of testing?</td></tr> <tr> <td>Scope of Testing</td><td>What is included and excluded in testing?</td></tr> <tr> <td>Types of Testing</td><td>What kinds of testing (functional, performance, security, etc.) will be conducted?</td></tr> <tr> <td>Test Environment</td><td>What environments will be used for testing?</td></tr> <tr> <td>Risks and Assumptions</td><td>What risks are associated with the testing process?</td></tr> <tr> <td>Critical Success Factors (CSFs)</td><td>Key attributes that the product must achieve to be considered successful (e.g., performance, security, reliability).</td></tr> </table>                                                                                                 | Major Test Objectives | What are the primary goals of testing? | Scope of Testing           | What is included and excluded in testing?    | Types of Testing                 | What kinds of testing (functional, performance, security, etc.) will be conducted? | Test Environment                | What environments will be used for testing?            | Risks and Assumptions       | What risks are associated with the testing process? | Critical Success Factors (CSFs) | Key attributes that the product must achieve to be considered successful (e.g., performance, security, reliability). |
| Major Test Objectives                                                                                          | What are the primary goals of testing?                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                           |                       |                                        |                            |                                              |                                  |                                                                                    |                                 |                                                        |                             |                                                     |                                 |                                                                                                                      |
| Scope of Testing                                                                                               | What is included and excluded in testing?                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                        |                       |                                        |                            |                                              |                                  |                                                                                    |                                 |                                                        |                             |                                                     |                                 |                                                                                                                      |
| Types of Testing                                                                                               | What kinds of testing (functional, performance, security, etc.) will be conducted?                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                               |                       |                                        |                            |                                              |                                  |                                                                                    |                                 |                                                        |                             |                                                     |                                 |                                                                                                                      |
| Test Environment                                                                                               | What environments will be used for testing?                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                      |                       |                                        |                            |                                              |                                  |                                                                                    |                                 |                                                        |                             |                                                     |                                 |                                                                                                                      |
| Risks and Assumptions                                                                                          | What risks are associated with the testing process?                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                              |                       |                                        |                            |                                              |                                  |                                                                                    |                                 |                                                        |                             |                                                     |                                 |                                                                                                                      |
| Critical Success Factors (CSFs)                                                                                | Key attributes that the product must achieve to be considered successful (e.g., performance, security, reliability).                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                             |                       |                                        |                            |                                              |                                  |                                                                                    |                                 |                                                        |                             |                                                     |                                 |                                                                                                                      |
| Pairwise Testing                                                                                               | <p>It is used to reduce the number of test cases while maintaining effective test coverage. It works by focusing on the interactions between different input parameters in pairs, rather than testing every possible combination of inputs (which can be impractical for systems with many parameters)</p> <p>Suppose you're testing a system with 3 parameters:</p> <p>Browser: Chrome, Firefox, Edge<br/>OS: Windows, Mac, Linux<br/>Connection: Wifi, Ethernet</p> <p>If you wanted to test every possible combination, you'd need <math>3 \times 3 \times 2 = 18</math> test cases. With pairwise testing, you'd select test cases that cover all pairs of inputs, reducing the total number of cases while still providing good coverage. Instead of 18, you might only need something like 6-9 test cases, depending on the relationships between inputs.</p>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                              |                       |                                        |                            |                                              |                                  |                                                                                    |                                 |                                                        |                             |                                                     |                                 |                                                                                                                      |
| Design Principles                                                                                              | Followed SOLID principles, ensured modularity, and created reusable components to manage complexity.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                             |                       |                                        |                            |                                              |                                  |                                                                                    |                                 |                                                        |                             |                                                     |                                 |                                                                                                                      |
| How are variables stored in memory?                                                                            | Variables are stored in memory based on their type. Primitive types are stored directly on the stack, whereas objects are stored in the heap, with the stack holding references to the heap memory.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                              |                       |                                        |                            |                                              |                                  |                                                                                    |                                 |                                                        |                             |                                                     |                                 |                                                                                                                      |
| Automation Challenges: What challenges do you face during test automation?                                     | Common challenges include handling dynamic content, managing test data, dealing with flaky tests, integrating with CI/CD pipelines, and ensuring cross-browser compatibility.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                    |                       |                                        |                            |                                              |                                  |                                                                                    |                                 |                                                        |                             |                                                     |                                 |                                                                                                                      |
| Why might Selenium's hover or click not work on an element                                                     | Possible reasons: dynamic content or overlays.<br>Solutions: use JavaScriptExecutor to perform the action or scroll the element into view using JavaScript or Selenium's Actions class.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                          |                       |                                        |                            |                                              |                                  |                                                                                    |                                 |                                                        |                             |                                                     |                                 |                                                                                                                      |
| What are essential credit card test cases for validations?                                                     | Check number length, format, and type-specific rules such as Luhn algorithm for validity, card expiration date, CVV, and name format.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                            |                       |                                        |                            |                                              |                                  |                                                                                    |                                 |                                                        |                             |                                                     |                                 |                                                                                                                      |
| How would you validate the dynamic structure of a Gmail API response where the number of unread emails varies? | When working with dynamic API responses, like those from the Gmail API where the number of unread emails can vary, it's crucial to implement robust validation strategies. Here are some key approaches to ensure the integrity of your data:<br><br>1. Schema Validation:                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                       |                       |                                        |                            |                                              |                                  |                                                                                    |                                 |                                                        |                             |                                                     |                                 |                                                                                                                      |

|                                                        |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                        |
|--------------------------------------------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
|                                                        | <p>JSON Schema: Define a JSON schema that outlines the expected structure of the response, including the messages array and its individual elements.</p> <p>Validation Libraries: Use a library like org.everit.json.schema to validate the response against the schema.</p> <pre>import org.everit.json.schema.Schema; import org.everit.json.schema.ValidationException; import org.json.JSONObject;  // ... (code to fetch the API response as a JSON string)  JSONObject responseJson = new JSONObject(responseString);  // Load the JSON schema from a file or string Schema schema = Schema.parse(new JSONObject(schemaString));  try {     schema.validate(responseJson);     System.out.println("Response is valid"); } catch (ValidationException e) {     System.err.println("Validation error: " + e.getMessage()); }</pre> |
|                                                        | <p>2. Data Type Validation:</p> <p>Type Checking: Verify that each field in the response has the expected data type (e.g., string, integer, boolean).</p> <p>Range Checks: Ensure that numerical values fall within specific ranges.</p> <p>Format Validation: Validate date and time formats, email addresses, and other structured data.</p>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                         |
|                                                        | <p>3. Content Validation:</p> <p>Required Fields: Check for the presence of mandatory fields in each message object.</p> <p>Field Values: Validate the content of fields like id, threadId, labelIds, snippet, and historyId.</p> <p>Data Consistency: Verify that related data (e.g., sender, recipient, subject) is consistent within each message.</p>                                                                                                                                                                                                                                                                                                                                                                                                                                                                              |
|                                                        | <p>4. Response Status and Error Handling:</p> <p>HTTP Status Codes: Check for expected status codes (e.g., 200 for success).</p> <p>Error Messages: Parse error messages to identify specific issues and take appropriate actions.</p> <p>Retry Logic: Implement retry mechanisms for transient errors or rate limits.</p>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                             |
| Why do we use interfaces in Java                       | <ul style="list-style-type: none"> <li>Abstraction: Provides a way to achieve abstraction. Interfaces allow us to define methods that should be implemented by classes.</li> <li>Multiple Inheritance: Java does not support multiple inheritance in classes, but a class can implement multiple interfaces.</li> <li>Loose Coupling: Interfaces help in reducing the coupling between the classes.</li> <li>Polymorphism: Interfaces allow us to use different implementation classes through a common reference type.</li> </ul>                                                                                                                                                                                                                                                                                                     |
| What are commonly used interfaces in Selenium?         | WebDriver, WebElement, TakesScreenshot, JavascriptExecutor, Alert, WindowHandles                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                       |
| How do you handle different windows in Selenium?       | Use getWindowHandles() to get all window handles.<br>Switch to a specific window using switchTo().window() with the desired handle.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                    |
| Test Plan for e-commerce website                       | <p>Purpose of test Plan: Cover functional, security, performance and usability</p> <p>Define the Test Approach, scope of testing, data security, acceptable performance</p> <p>Scope: Modules to be tested like search, filter etc</p> <p>Approach: CBT, CDT, performance, usability</p> <p>Env Set up</p> <p>Tools to be used</p> <p>Test artifacts to be provided at the end</p> <p>Entry and exit criteria</p> <p>Schedule</p> <p>Roles and responsibilities</p> <p>Risks</p>                                                                                                                                                                                                                                                                                                                                                       |
| Difference between mvn clean, mvn verify ,mvn test     | <p>mvn clean: Deletes the target directory to remove previous build artifacts.</p> <p>mvn verify: Validates the project, runs integration tests, and checks project quality criteria.</p> <p>mvn test: Runs the unit tests for the project.</p>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                        |
| How to Design Test Cases for an Online Payment Gateway | <p>Whenever asked to design test cases you can consider the below approach to cover all aspects, start from UI Test -&gt; functional and negative tests -&gt; boundary cases -&gt; security tests -&gt; performance validation:</p> <ol style="list-style-type: none"> <li>1. UI &amp; Usability Tests: <ul style="list-style-type: none"> <li>Test mobile responsiveness and verify layout across various devices.</li> <li>Ensure clarity of instructions, ease of navigation, and visibility of error/success messages.</li> <li>Test accessibility features, making sure the process is smooth for all users.</li> </ul> </li> <li>2. Functional Tests: <ul style="list-style-type: none"> <li>Validate payment workflows for different payment methods (credit card, debit card, UPI, PayPal, etc.).</li> </ul> </li> </ol>       |

|                                                                                                                                                                                                                                                             |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                |                                                                                                                                                                                                                          |                                                                                                                                                                                                                                                                                                                                                                                                                                                      |                                                                                                                                                                                                                          |                                                          |                                                                                                                                                                                                                                                                                                                                  |                                                                                              |                                                    |                                                                                                                                                                                                                                                                                                          |                                                                                                                                                                                            |
|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|----------------------------------------------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|----------------------------------------------------------------------------------------------|----------------------------------------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
|                                                                                                                                                                                                                                                             | <ul style="list-style-type: none"> <li>■ Ensure successful payments are reflected accurately in the user's transaction history.</li> <li>■ Confirm that each step—like entering card details, OTP, and confirmation—functions smoothly.</li> </ul> <p><b>3. Negative Test Cases:</b></p> <ul style="list-style-type: none"> <li>■ Test invalid inputs like expired cards, incorrect CVVs, and incorrect billing addresses.</li> <li>■ Check the system's behavior with insufficient funds and blocked cards.</li> </ul> <p>Validate error messages and see how the system handles canceled payments and re-tries.</p> <p><b>4. Boundary Cases:</b></p> <ul style="list-style-type: none"> <li>■ Test payments at the minimum and maximum transaction amounts allowed by the platform.</li> <li>■ Check how the system responds to network timeouts or session timeouts during payment.</li> <li>■ Verify handling of simultaneous transactions from the same account.</li> </ul> <p><b>5. Security Tests:</b></p> <ul style="list-style-type: none"> <li>■ Ensure secure transfer protocols (e.g., HTTPS) and validate encryption of sensitive information.</li> <li>■ Test against vulnerabilities like SQL injection, XSS, and ensure no sensitive data (like card numbers) is stored insecurely.</li> <li>■ Validate user session timeouts and check logout mechanisms for security.</li> </ul> <p><b>6. Performance &amp; Load Tests:</b></p> <ul style="list-style-type: none"> <li>■ Simulate high traffic to assess if the gateway can handle peak loads without slowing down.</li> <li>■ Measure transaction speed to ensure there's no noticeable delay in the user experience.</li> <li>■ Check response times during multi-currency conversions if applicable.</li> </ul>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                           |                                                                                                                                                                                                                          |                                                                                                                                                                                                                                                                                                                                                                                                                                                      |                                                                                                                                                                                                                          |                                                          |                                                                                                                                                                                                                                                                                                                                  |                                                                                              |                                                    |                                                                                                                                                                                                                                                                                                          |                                                                                                                                                                                            |
| Smoke vs Sanity                                                                                                                                                                                                                                             | <p>Smoke Testing verifies the critical functionalities of the system whereas Sanity Testing verifies the new functionality like bug fixes.</p> <p>Smoke testing is a subset of acceptance testing whereas Sanity testing is a subset of Regression Testing.</p> <p>Smoke-&gt;Sanity-&gt;Regression</p>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                         |                                                                                                                                                                                                                          |                                                                                                                                                                                                                                                                                                                                                                                                                                                      |                                                                                                                                                                                                                          |                                                          |                                                                                                                                                                                                                                                                                                                                  |                                                                                              |                                                    |                                                                                                                                                                                                                                                                                                          |                                                                                                                                                                                            |
| <pre>class A {     static { System.out.print("6"); }      { System.out.print("2"); }      public A() { System.out.print("3"); }      public class Test {         public static void main(String[] args)         { A a1 = new A(); A a2 = new A(); } }</pre> | <p>Output: 62323</p> <ul style="list-style-type: none"> <li>• The static block in class A is executed only once when the class is loaded, printing "6".</li> <li>• Each time an object of class A is created, the instance initializer block is executed, printing "2", followed by the constructor, printing "3".</li> <li>• Since two objects are created, "23" is printed twice.</li> </ul>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                 |                                                                                                                                                                                                                          |                                                                                                                                                                                                                                                                                                                                                                                                                                                      |                                                                                                                                                                                                                          |                                                          |                                                                                                                                                                                                                                                                                                                                  |                                                                                              |                                                    |                                                                                                                                                                                                                                                                                                          |                                                                                                                                                                                            |
| TDD                                                                                                                                                                                                                                                         | <p>TDD follows the <b>Red-Green-Refactor</b> cycle:</p> <p><b>Red</b> – Write a test case <b>before</b> implementing the functionality. Since the functionality doesn't exist yet, the test <b>fails</b>.</p> <p><b>Green</b> – Write the <b>minimum amount of code</b> required to <b>pass the test</b>.</p> <p><b>Refactor</b> – Optimize and clean up the code while ensuring that all tests <b>still pass</b>.</p> <p>This cycle repeats until all functionality is implemented. This software development approach can be used in API Automation framework</p> <table border="1"> <tr> <td><b>Step 1:</b> Write a Failing Test (Red Phase)</td><td> <pre>@Test public void testCreateUser() {     String requestBody = "{\"name\":\"John\", \"job\": \"QA Engineer\"}";      given()         .baseUri("https://reqres.in/api")         .contentType(MediaType.JSON)         .body(requestBody)     .when()         .post("/users")     .then()         .statusCode(201) // Expecting a 201 Created response         .body("name", equalTo("John"))         .body("job", equalTo("QA Engineer")); }</pre> </td><td><b>Expected Outcome:</b><br/>This test <b>fails</b> initially because: <ul style="list-style-type: none"> <li>• The API might not be implemented yet.</li> <li>• The response might not match expected values.</li> </ul> </td></tr> <tr> <td><b>Step 2:</b> Implement API or Fix Issues (Green Phase)</td><td> <ul style="list-style-type: none"> <li>• If testing an API under development, the backend team implements the API to return the expected response.</li> <li>• If testing an existing API, update assertions if necessary.</li> <li>• Ensure the API actually <b>returns</b> { "name": "John", "job": "QA Engineer" }.</li> </ul> </td><td><b>Expected Outcome:</b><br/>After API implementation or fixes, the test <b>should pass</b>.</td></tr> <tr> <td><b>Step 3:</b> Refactor Test Code (Refactor Phase)</td><td> <p>Now, improve the test structure by:</p> <ul style="list-style-type: none"> <li>• Extracting base configurations to a separate class.</li> <li>• Using reusable methods for API calls.</li> <li>• Adding validations for headers, response time, etc.</li> </ul> <pre>public class UserApiTest {</pre> </td><td> <ul style="list-style-type: none"> <li>• <b>Base URI</b> is moved to <code>@BeforeAll</code>.</li> <li>• <b>API call logic</b> is moved to <code>createUser()</code> for reuse.</li> </ul> </td></tr> </table> | <b>Step 1:</b> Write a Failing Test (Red Phase)                                                                                                                                                                          | <pre>@Test public void testCreateUser() {     String requestBody = "{\"name\":\"John\", \"job\": \"QA Engineer\"}";      given()         .baseUri("https://reqres.in/api")         .contentType(MediaType.JSON)         .body(requestBody)     .when()         .post("/users")     .then()         .statusCode(201) // Expecting a 201 Created response         .body("name", equalTo("John"))         .body("job", equalTo("QA Engineer")); }</pre> | <b>Expected Outcome:</b><br>This test <b>fails</b> initially because: <ul style="list-style-type: none"> <li>• The API might not be implemented yet.</li> <li>• The response might not match expected values.</li> </ul> | <b>Step 2:</b> Implement API or Fix Issues (Green Phase) | <ul style="list-style-type: none"> <li>• If testing an API under development, the backend team implements the API to return the expected response.</li> <li>• If testing an existing API, update assertions if necessary.</li> <li>• Ensure the API actually <b>returns</b> { "name": "John", "job": "QA Engineer" }.</li> </ul> | <b>Expected Outcome:</b><br>After API implementation or fixes, the test <b>should pass</b> . | <b>Step 3:</b> Refactor Test Code (Refactor Phase) | <p>Now, improve the test structure by:</p> <ul style="list-style-type: none"> <li>• Extracting base configurations to a separate class.</li> <li>• Using reusable methods for API calls.</li> <li>• Adding validations for headers, response time, etc.</li> </ul> <pre>public class UserApiTest {</pre> | <ul style="list-style-type: none"> <li>• <b>Base URI</b> is moved to <code>@BeforeAll</code>.</li> <li>• <b>API call logic</b> is moved to <code>createUser()</code> for reuse.</li> </ul> |
| <b>Step 1:</b> Write a Failing Test (Red Phase)                                                                                                                                                                                                             | <pre>@Test public void testCreateUser() {     String requestBody = "{\"name\":\"John\", \"job\": \"QA Engineer\"}";      given()         .baseUri("https://reqres.in/api")         .contentType(MediaType.JSON)         .body(requestBody)     .when()         .post("/users")     .then()         .statusCode(201) // Expecting a 201 Created response         .body("name", equalTo("John"))         .body("job", equalTo("QA Engineer")); }</pre>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                           | <b>Expected Outcome:</b><br>This test <b>fails</b> initially because: <ul style="list-style-type: none"> <li>• The API might not be implemented yet.</li> <li>• The response might not match expected values.</li> </ul> |                                                                                                                                                                                                                                                                                                                                                                                                                                                      |                                                                                                                                                                                                                          |                                                          |                                                                                                                                                                                                                                                                                                                                  |                                                                                              |                                                    |                                                                                                                                                                                                                                                                                                          |                                                                                                                                                                                            |
| <b>Step 2:</b> Implement API or Fix Issues (Green Phase)                                                                                                                                                                                                    | <ul style="list-style-type: none"> <li>• If testing an API under development, the backend team implements the API to return the expected response.</li> <li>• If testing an existing API, update assertions if necessary.</li> <li>• Ensure the API actually <b>returns</b> { "name": "John", "job": "QA Engineer" }.</li> </ul>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                               | <b>Expected Outcome:</b><br>After API implementation or fixes, the test <b>should pass</b> .                                                                                                                             |                                                                                                                                                                                                                                                                                                                                                                                                                                                      |                                                                                                                                                                                                                          |                                                          |                                                                                                                                                                                                                                                                                                                                  |                                                                                              |                                                    |                                                                                                                                                                                                                                                                                                          |                                                                                                                                                                                            |
| <b>Step 3:</b> Refactor Test Code (Refactor Phase)                                                                                                                                                                                                          | <p>Now, improve the test structure by:</p> <ul style="list-style-type: none"> <li>• Extracting base configurations to a separate class.</li> <li>• Using reusable methods for API calls.</li> <li>• Adding validations for headers, response time, etc.</li> </ul> <pre>public class UserApiTest {</pre>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                       | <ul style="list-style-type: none"> <li>• <b>Base URI</b> is moved to <code>@BeforeAll</code>.</li> <li>• <b>API call logic</b> is moved to <code>createUser()</code> for reuse.</li> </ul>                               |                                                                                                                                                                                                                                                                                                                                                                                                                                                      |                                                                                                                                                                                                                          |                                                          |                                                                                                                                                                                                                                                                                                                                  |                                                                                              |                                                    |                                                                                                                                                                                                                                                                                                          |                                                                                                                                                                                            |

• Logging is added to capture responses.

```

@BeforeAll
public static void setup() {
 RestAssured.baseURI = "https://regres.in/api";
}

private Response createUser(String name, String job) {
 String requestBody = "{\"name\":\"" + name + "\", \"job\": \"" + job +
 "\}";
 return given()
 .contentType(MediaType.APPLICATION_JSON)
 .body(requestBody)
 .when()
 .post("/users");
}

@Test
public void testCreateUser() {
 Response response = createUser("John", "QA Engineer");
 response.then()
 .statusCode(201)
 .body("name", equalTo("John"))
 .body("job", equalTo("QA Engineer"))
 .log().all();
}
}

```

Test a service which returns the schedule of flights available based on input parameters

Lets assume the web service returns available flight schedules based on input parameters like:

- Departure City
- Arrival City
- Departure Date
- Optional: Return Date, Class, Number of Passengers

#### A. Input Validation Tests:

- Invalid Input Formats:

- Test with incorrect date formats (e.g., "2023-13-45", "abc").
- Test with non-existent city names.
- Test with invalid class types (e.g., "Luxury").
- Test with non numeric passenger count.
- Same to and from City
- Test with Past Date

- Missing Required Parameters:

- Test with missing departure city.
- Test with missing arrival city.
- Test with missing departure date.

- Boundary Value Analysis:

- Test with minimum and maximum allowed passenger counts.
- Test with dates near the service's supported date range.

- Null/Empty Input:

- Test with null or empty values for each parameter.

- SQL Injection/Cross-Site Scripting (XSS) Prevention:

- Test with malicious strings in the input parameters to ensure the service is protected against security vulnerabilities.

#### B. Functional Tests:

- Valid Input, Expected Results:

- Test with valid departure city, arrival city, and date, and verify that the service returns the correct flight schedules.
- Test with return dates, and verify that round trip flights are returned correctly.
- Test with class parameters, and verify the correct class flights are returned.
- Test with passenger count, and verify that the correct number of seats are available.

- No Flights Available:

- Test with input parameters that should result in no available flights, and verify that the service returns an appropriate message (e.g., an empty list or a "no flights found" message).

- Edge Cases:

- Test with same departure and arrival city (e.g., for connecting flights).
- Test with flights that cross time zones.
- Test with flights that change dates due to time zone changes.
- Test with flight schedules that are very close to the departure time.

- Data Integrity:

- Verify that the returned flight data is accurate (e.g., flight numbers, times, prices).

#### C. Performance Tests:

- Load Testing:

- Simulate a large number of concurrent users to test the service's performance under heavy load.

- Stress Testing:

- Push the service beyond its expected load to determine its breaking point.

- Response Time Testing:

- Measure the service's response time for different input parameters and load levels.

- **Throughput Testing:**

- Measure the number of transactions the service can handle per second.

- D. **Security Tests:**

- **Authentication and Authorization:**

- If the service requires authentication, test with valid and invalid credentials.
  - Verify that authorized users can access the service and that unauthorized users are denied access.

- **Data Encryption:**

- If sensitive data is transmitted, verify that it is encrypted using HTTPS or other secure protocols.

- **Vulnerability Scanning:**

- Use security scanning tools to identify potential vulnerabilities in the service.

- E. **Error Handling Tests:**

- **Unexpected Errors:**

- Simulate unexpected errors (e.g., database connection failures, network errors) and verify that the service handles them gracefully.

- Verify that the service returns meaningful error messages.

- **Timeout Tests:**

- Test what happens when the service takes longer than the allowed timeout period.

# Witcher 3

Wednesday, July 23, 2025 6:19 AM

Absolutely, Saurav! Since you're just starting The Witcher 3 on PlayStation and new to the Witcher world, here's a comprehensive guide to help you:

---

?? The Witcher 3: Beginner's Guide + Step-by-Step Strategy for White Orchard (Starting Area)

This will cover:

1. ✕ Basics you must know before playing
2. ㉑ Step-by-step mission flow (main + side quests)
3. ⚡ How to get maximum loot, XP & benefits
4. 🔫 How to use rewards to boost Geralt's strength
5. ㉑ Roleplaying and moral decisions explained

---

✖ PART 1: Basics You Must Know Before Playing

| Category | Info |
|----------|------|
|----------|------|

Geralt A Witcher – genetically enhanced monster hunter for hire. Uses swords, signs (magic), potions, and alchemy.

Combat Uses 2 swords: Steel (humans), Silver (monsters). Uses Signs (magic) and potions to enhance abilities.

Signs Small spells. Most useful early: Quen (shield), Igni (fire), Yrden (trap for ghosts), Axii (mind control).

Potions Give buffs like health regen or attack boost. Must be crafted from herbs. Once crafted, they refill automatically during Meditation if you have alcohol (Alcohest).

Looting Use Witcher Senses (hold L2) to find hidden items in houses, bodies, crates. Loot everything.

Crafting & Repairs Visit blacksmiths for weapons, armorers for armor. They also repair gear.

Fast Travel Only between signposts. Discover them by exploring.

Inventory Tip Don't sell too much early; dismantle junk to get materials for crafting better gear.

---

¶ PART 2: Step-by-Step Optimal Progression in White Orchard

✓ 0. Before You Leave the Inn

Loot everything: barrels, food, water (useful for healing)

Talk to Gaunter O'Dimm (foreshadowing for later expansions)

Choose to fistfight the three drunks (gives minor XP and moral superiority)

---

✓ 1. Go to the Notice Board (White Orchard Village)

📍 Location: Near the village center

Pick up all available quests:

Main Quest: Beast of White Orchard (Griffin Hunt)

Side Quests:

Twisted Firestarter

Missing in Action

On Death's Bed

Precious Cargo

Witcher Contract:

Devil by the Well

¶ This fills your journal with all content available in this zone.

---

✓ 2. Twisted Firestarter (Side Quest)

📍 Talk to the dwarf blacksmith Willis

1. Accept his quest to find out who burned his forge.

2. Use Witcher Senses (L2) to track the arsonist.

3. Confront him and decide:

Spare him: He'll make amends. Morally good.

Turn him in: Villagers beat him. Harsh, lawful.

🔧 Regardless of choice, Willis reopens his forge. 💰 He now offers weapon crafting, repairs, and discounts. 🔮 Use this later to craft Viper swords and repair damaged gear.

---

✓ 3. On Death's Bed (Side Quest)

👉 Visit Herbalist Tomira (south of White Orchard)

1. She needs Celandine herb and a Swallow potion to help a wounded girl.

2. Gather Celandine (yellow flowers) using Witcher Senses.

3. Craft Swallow potion using ingredients found in her hut.

4. Give her the potion.

💡 Learn alchemy, gain XP, and gain access to cheap potion ingredients. 💕 Swallow potion heals Geralt over time – super useful early on.

---

✓ 4. Beast of White Orchard (Main Quest: Griffin Hunt)

Steps:

1. Visit Nilfgaardian Garrison → accept contract.

2. Talk to Hunter Mislav (track him using map).

3. Examine the Griffin's nest.

4. Visit Tomira again for Buckthorn herb.

5. Brew Thunderbolt potion (boosts attack power).

6. Prepare for fight:

Quen sign

Silver sword equipped

Thunderbolt potion ready

Grapeshot bomb if you have it

#### 🦅 Griffin Fight Tips:

Use Quen to shield.

Roll (not dodge) to avoid swoops.

Use crossbow (L1 → R1) to bring it to ground.

Hit while it's grounded.

#### 👑 Rewards:

150–200 XP

Griffin Trophy (+10% XP from humans & nonhumans — equip on horse)

Crowns (gold)

➡ Visit Willis after for sword crafting using this money.

---

#### ✓ 5. Devil by the Well (Witcher Contract)

👉 Start at Notice Board → Odolan's house

1. Investigate haunted well using Witcher Senses.

2. Find corpse, journal, and story.

3. Burn remains → fight Noonwraith

4. Use Yrden sign (trap) to make her vulnerable.

#### 👑 Rewards:

Noonwraith Trophy

XP

Crowns (you can negotiate reward using Axii sign if upgraded)

---

✓ 6. Missing in Action (Side Quest)

👉 Speak to Dune at the farm

1. Use his dog's trail to track Bastien.

2. Decide whether to let the wounded enemy soldier live with them.

Letting him live = moral choice + peaceful resolution

⌚ Reward: XP, reputation building

---

✓ 7. Precious Cargo (Side Quest)

👉 Near the broken bridge

1. Help a merchant retrieve his goods.

2. Investigate tracks with Witcher Senses.

3. Discover he's a smuggler. Choose:

Let him go = keep loot

Turn him in = morality boost

⌚ Reward: Loot or XP

---

✓ 8. Explore Hidden Locations (For Max XP & Gear)

DO NOT leave White Orchard until you've done the following:

| Activity                 | Location                 | Reward                     |
|--------------------------|--------------------------|----------------------------|
| Place of Power (5 total) | Dotted around the map    | +5 Skill Points            |
| Guarded Treasures        | Multiple – marked with ? | Crafting gear, money       |
| Monster Nests            | Use bombs (Grapeshot)    | XP + crafting materials    |
| Abandoned Sites          | Clear them of enemies    | XP + repopulates merchants |

📍 Use the map to track question marks (?) → clear each one

---

### 💰 PART 3: How to Use Coins, Rewards, and NPCs for Maximum Benefit

Use Willis to craft better swords and repair gear.

Dismantle junk (don't just sell) for materials like leather, cloth, steel.

Buy:

Grapeshot bomb formula

Gwent cards (innkeeper)

Alcohest (for potion refills)

Brew:

Swallow (healing)

Thunderbolt (attack boost)

↗ Buy better saddles and saddlebags when available — improves stamina and inventory capacity.

---

### ❓ Roleplaying Choices That Matter

| Quest               | Decision                      | Benefit                      |
|---------------------|-------------------------------|------------------------------|
| Twisted Firestarter | Spare arsonist or turn in     | Roleplaying consequence only |
| Devil by the Well   | Negotiate fee or accept       | Use Axii for bonus gold      |
| Precious Cargo      | Report or spare smuggler      | Morality vs loot             |
| Missing in Action   | Let enemy soldier stay or not | Builds Geralt's character    |

→ No long-term effect yet, but these build the tone of your story.

---

- ✓ Final Checklist Before Leaving White Orchard
- ✓ Beast of White Orchard (Main)
- ✓ Devil by the Well (Contract)
- ✓ All Side Quests
- ✓ All 5 Places of Power
- ✓ Crafted Swallow and Thunderbolt potions
- ✓ Viper Swords (found in ruins and tombs)
- ✓ Looted Monster Nests, Abandoned Sites, Treasures
- ✓ Gained Griffin Trophy
- ✓ Used Willis and Tomira effectively
- ✓ Reached Level 3–4

---

Would you like a printable version, checklist, or map with key spots marked?

Also, let me know when you're ready to start Velen/No Man's Land, or want a guide on:

Gwent

Skill trees & builds

Crafting and upgrade flow

Enjoy the Continent, Witcher !

# Default Stages

30 March 2024 08:14

## .pre Stage:

- The .pre stage is a special stage in GitLab CI/CD pipelines that runs before any other stages.
- It can be used for tasks such as setting up environment variables, initializing dependencies, or performing pre-build checks.
- Tasks in this stage are executed once before the main pipeline stages start.

## Build Stage:

- The build stage is where the source code is compiled, dependencies are resolved, and artifacts are generated.
- It typically includes tasks like compiling code, running build scripts, and packaging the application for deployment.
- The success of this stage indicates that the code can be built successfully and is ready for testing.

## Test Stage:

- The test stage involves running various types of tests to verify the functionality and quality of the code.
- It can include unit tests, integration tests, end-to-end tests, and other forms of automated testing.
- The goal of this stage is to identify any bugs or issues in the codebase before it is deployed to production.

## Deploy Stage:

- The deploy stage is where the built artifacts are deployed to the target environment, such as a testing, staging, or production server.
- It includes tasks like transferring files, configuring servers, and setting up the application environment.
- The success of this stage indicates that the application has been successfully deployed to the target environment.

## .post Stage:

- The .post stage is another special stage in GitLab CI/CD pipelines that runs after all other stages have completed.
- It can be used for tasks such as cleanup operations, sending notifications, or performing post-deployment checks.
- Tasks in this stage are executed once after all the main pipeline stages have finished.

These stages provide a structured approach to building, testing, deploying, and managing applications in GitLab CI/CD pipelines. They can be customized and extended to fit the specific requirements of each project.

Order of .pre and .post cannot change, while for build, test and deploy order can be changed.

# Script

30 March 2024 08:44

Script Section in GitLab CI/CD:

- It defines the docker commands or shell scripts to be executed within each job of the pipeline.
- Each job in the pipeline can have its own `script` section, allowing for customization and flexibility.
- The `script` section is where you specify the actions to be taken to accomplish the goals of the job, such as building, testing, deploying, etc.

stages:

- build
- test
- deploy

build\_job:

```
stage: build
script:
 - echo "Building the project..."
 - mvn clean install
```

test\_job:

```
stage: test
script:
 - echo "Running tests..."
 - mvn test
```

deploy\_job:

```
stage: deploy
script:
 - echo "Deploying the application..."
 - mvn deploy
```

# Before and After Script

30 March 2024 08:47

- These sections allow for configuring actions to be performed before and after executing the main script of a job.
- The `before\_script` section is executed before the main script of each job in the pipeline. It is commonly used for setting up the environment, installing dependencies, or initializing variables.
- Conversely, the `after\_script` section is executed after the main script of each job completes. It is typically used for cleanup tasks, gathering logs, or sending notifications.
- Both `before\_script` and `after\_script` sections can contain multiple commands or scripts, which are executed sequentially.

Example:

```
```yaml
stages:
- build
- test

build_job:
stage: build
before_script:
- echo "Setting up environment..."
- apt-get update && apt-get install -y some-package
script:
- echo "Building the project..."
- mvn clean install
after_script:
- echo "Cleaning up..."
- rm -rf some-directory

test_job:
stage: test
before_script:
- echo "Preparing for testing..."
script:
- echo "Running tests..."
- mvn test
after_script:
- echo "Test cleanup..."```

```

Variables

30 March 2024 08:51

Predefined Variables (Built-in Variables)	<p>These variables are automatically set by GitLab during the pipeline execution.</p> <p>Examples:</p> <ul style="list-style-type: none">`CI_COMMIT_BRANCH`: The branch or tag name for which the pipeline is running.`CI_COMMIT_SHA`: The commit SHA for the current revision.`CI_PROJECT_ID`: The project ID.`CI_PROJECT_PATH`: The project namespace and name.`CI_PIPELINE_ID`: The ID of the current pipeline.`CI_JOB_STAGE`: The stage of the current job.`CI_REGISTRY`: the container registry of the project. <p>Many more regarding git, runner information, and environment. These are incredibly useful for dynamically configuring your pipeline based on the context.</p>
Project Variables	<p>These are variables defined at the project level in GitLab's settings. They are accessible to all pipelines within the project. You can define them as:</p> <ul style="list-style-type: none">Variables: Standard key-value pairs.File Variables: Variables that store the content of a file.Masked Variables: Variables whose values are masked in job logs (for sensitive information like passwords or API keys). <p>Project variables are great for storing configuration values that are common across multiple pipelines in a project.</p>
Group Variables	<p>Similar to project variables, but defined at the group level.</p> <p>They are inherited by all projects within the group.</p> <p>This is useful for sharing configuration across multiple related projects.</p>
Instance Variables	<p>These are variables defined at the GitLab instance level (for self-managed GitLab instances).</p> <p>They are accessible to all projects within the instance.</p> <p>These are used to set configurations that are relevant to the whole gitlab instance.</p>
Environment Variables	<p>Variables defined within the ` `.gitlab-ci.yml` file, specifically within the `variables` keyword.</p> <p>They can be defined at the global level (affecting all jobs) or at the job level (affecting only a specific job).</p> <p>Example:</p> <pre>```yaml variables: DEPLOY_ENVIRONMENT: "staging" deploy_job: stage: deploy variables: DEPLOY_ENVIRONMENT: "production" script: - echo "Deploying to \$DEPLOY_ENVIRONMENT" ``` In the example above, the deploy job uses the production environment, and all other jobs would use the staging environment if they used the variable.</pre>
CI/CD Variable files	<p>These are files that contain variables, that can be included in the gitlab-ci.yml file.</p> <p>This is very useful for large amounts of variables, or when variables need to be generated by scripts.</p>

Building Docker Image

30 March 2024 09:04

```
stages:
  - build
docker_build_job:
  stage : build
  tags:
    - CI
    - Devops
  before_script:
    - echo "Build Started"
    - sudo docker login registry.gitlab.com -u autopilotdevops -p $Access_Token
  script:
    - sudo docker build -t registry.gitlab.com/autopilotdevops/autopilot-devops .
    - sudo docker push registry.gitlab.com/autopilotdevops/autopilot-devops
```

Only and Except

30 March 2024 09:06

only:

- The only keyword specifies the conditions under which a job should be executed.
- If the conditions specified with only are met, the job will run.
- Multiple conditions can be specified using an array.
- If no only or except keyword is defined, the job will run by default.

except:

- The except keyword specifies the conditions under which a job should not be executed.
- If the conditions specified with except are met, the job will not run.
- Multiple conditions can be specified using an array.
- If no only or except keyword is defined, the job will run by default.

stages:

- test

test_job:

stage: test

script:

- echo "Running tests..."

only:

- master # Run only on the master branch
- tags # Run only on tagged commits
- schedules # Run only on scheduled pipelines
- triggers # Run only on pipelines triggered by a webhook or API call
- web # Run only on web pipeline triggers
- api # Run only on API pipeline triggers
- merge_requests # Run only on merge requests
- branches@gitlab-org/gitlab # Run only on branches of a specific forked repository

except:

- triggers # Do not run on pipelines triggered by a webhook or API call
- schedules # Do not run on scheduled pipelines
- web # Do not run on web pipeline triggers
- api # Do not run on API pipeline triggers

Rules

30 March 2024 09:09

- The `rules` keyword allows defining conditions under which jobs should be included or excluded from a pipeline.

- Each rule consists of conditions and an associated action.

- Multiple rules can be defined, and they are evaluated in order.

- If no rules match, the job will not be executed.

- Rules can use predefined variables, expressions, and logical operators to define conditions.

- Actions associated with rules include `if`, `when`, `allow_failure`, `changes`, and `exists`.

- Cannot be used with only/except

It's a more flexible and expressive alternative to the "only" and "except" keywords. It also allows you to specify conditions based on different factors like branch names, tags, changes to files, variables, and more.

Example:

stages:

- test

test_job:

stage: test

script:

- echo "Running tests..."

rules:

- if: '\$CI_COMMIT_BRANCH == "master"

- if: '\$CI_COMMIT_TAG =~ /^v\d+\.\d+\.\d+\$/'

- when: always

In this example:

- The `test_job` will run if either the branch of the commit is `master` or the commit is tagged with a version tag matching the pattern `vX.X.X`.

- Additionally, the job will always run regardless of the outcome of previous jobs in the pipeline.

job1:

script:

- echo "This job will execute if any of the conditions are met"

rules:

- changes

- when: always

- exists: true

- when: on_success

job2:

script:

- echo "This job will execute if all conditions are met"

rules:

- if: '\$CI_COMMIT_BRANCH == "master" && \$CI_PIPELINE_SOURCE == "web"'

Timeout

30 March 2024 09:10

Timeout in GitLab CI/CD:

- The `timeout` keyword is used to define the maximum amount of time a job is allowed to run.
- It specifies the duration in which the job must complete its execution, after which it will be automatically terminated. It should be less than runner timeout.
- Timeout can be defined at the job level or globally for all jobs in the pipeline.
- Timeout values can be specified in seconds, minutes, hours, or days.
- If a job exceeds the specified timeout duration, it will be marked as failed, and subsequent actions (such as cleanup or notifications) may be triggered.

Example:

stages:

- build

build_job:

 stage: build

 script:

- echo "Building the project..."

 timeout: 1 hour 30 minutes

In this example:

- The `build_job` has a timeout of 1 hour 30 minutes specified.
- If the job exceeds one hour 30 minutes of execution time, it will be automatically terminated.

When

30 March 2024 09:12

When in GitLab CI/CD:

- The `when` keyword is used to define the conditions under which a job should be executed.
- It allows controlling the execution flow of jobs based on the status of previous jobs or pipeline triggers.
- Different values for `when` include `on_success`, `on_failure`, `always`, `manual`, and `delayed`.
- `on_success` executes the job only if the previous jobs in the pipeline were successful.
- `on_failure` executes the job only if the previous jobs in the pipeline failed.
- `always` always executes the job, regardless of the status of previous jobs.
- `manual` requires manual intervention to run the job.
- `delayed` schedules the job to run after a specified duration.

Example:

stages:

- test
- deploy

test_job:

stage: test

script:

- echo "Running tests..."

when: on_success

deploy_job:

stage: deploy

script:

- echo "Deploying the application..."

when: manual

In this example:

- The `test_job` will run only if the previous jobs in the pipeline were successful.
- The `deploy_job` requires manual intervention to run and will not be executed automatically.

Allow Failure

30 March 2024 09:13

Allow Failure in GitLab CI/CD:

- The `allow_failure` keyword is used to indicate that a job is allowed to fail without causing the entire pipeline to fail.
- It is useful for jobs that are non-critical or experimental, where failure does not necessarily indicate a problem with the codebase.
- If a job marked as `allow_failure` fails, the pipeline will continue to execute, and subsequent jobs will still run.
- Jobs marked as `allow_failure` are displayed differently in the GitLab CI/CD interface to indicate that they are non-blocking.
- Although the job is allowed to fail, it will still be executed, and its output will be available for review.

Example:

stages:

- test

test_job:

stage: test

script:

- echo "Running tests..."

allow_failure: true

In this example:

- The `test_job` is marked as `allow_failure`, indicating that it is allowed to fail without impacting the overall pipeline result.
- If `test_job` fails, the pipeline will continue to execute, and subsequent jobs will still run.

Concurrent Property

30 March 2024 09:14

Concurrent Property in GitLab Runner:

- The `concurrent` property in GitLab Runner configuration specifies the maximum number of job instances that can be run concurrently on the same Runner.
- It controls the level of parallelism, allowing multiple jobs to be executed simultaneously on a single Runner.
- By default, the `concurrent` property is set to the number of CPU cores available on the Runner machine.
- Setting a higher value for `concurrent` increases the number of jobs that can be processed concurrently, potentially improving pipeline execution speed.
- However, it's important to consider the hardware resources and performance limitations of the Runner machine when adjusting the `concurrent` property.
- Increasing the value of `concurrent` without sufficient resources may lead to resource contention, performance degradation, or job failures.

Example GitLab Runner Configuration (config.toml):

```
[[runners]]  
name = "My Runner"  
url = "https://gitlab.com/"  
token = "my-runner-token"  
executor = "shell"  
[runners.custom_build_dir]  
[runners.cache]  
  [runners.cache.s3]  
  [runners.cache.gcs]  
concurrent = 4
```

In this example:

- The `concurrent` property is set to 4, allowing the Runner to execute up to 4 job instances concurrently.
- Adjusting the `concurrent` property according to the hardware resources and workload requirements can optimize job execution and overall pipeline performance.

Needs

30 March 2024 09:16

Needs in GitLab CI/CD:

- The `needs` keyword is used to define dependencies between jobs in a GitLab CI/CD pipeline.
- It allows specifying which jobs must successfully complete before a particular job can start.
- With `needs`, you can create directed acyclic graphs (DAGs) of job dependencies, enabling complex pipeline workflows.
- Jobs specified in the `needs` keyword can be from the same or different stages in the pipeline.
- If a job specified in `needs` fails, the dependent job will not start, and the pipeline will fail.
- The `needs` keyword replaces the `dependencies` keyword, providing a more flexible and expressive way to define job dependencies.

Example:

stages:

- build
- test

build_job:

 stage: build

 script:

- echo "Building the project..."

test_job:

 stage: test

 script:

- echo "Running tests..."

 needs:

- build_job

In this example:

- The `test_job` depends on the successful completion of the `build_job`.
- If `build_job` fails, `test_job` will not start, and the pipeline will fail.

Artifact

30 March 2024 09:17

Artifact in GitLab CI/CD:

- The `artifacts` keyword is used to specify files and directories generated by a job that should be passed to subsequent jobs or made available for download.
- It allows capturing and sharing build artifacts, test reports, logs, and other output produced during job execution.
- Artifacts can be defined at the job level to include specific files or directories, or they can be defined globally for all jobs in the pipeline.
- Artifacts are stored on GitLab's server and can be accessed by other jobs or pipelines within the same project.
- Artifacts can be used for various purposes, such as sharing build outputs, publishing documentation, archiving test results, etc.
- Artifacts can be downloaded manually from the GitLab UI or via the GitLab API for further analysis or debugging.
- The `exclude` keyword allows excluding specific files or directories from the artifacts to be uploaded.
- The `expire_in` keyword sets the expiration time for artifacts, after which they are automatically deleted from GitLab's server.

Example:

stages:

- build

build_job:

stage: build

script:

- make build

artifacts:

paths:

- binaries/

- reports/

exclude:

- logs/

expire_in: 1 week

In this example:

- The `build_job` generates artifacts located in the `binaries/` and `reports/` directories, excluding the `logs/` directory.
- These artifacts will be available for download and can be passed to subsequent jobs in the pipeline.
- The artifacts will expire and be automatically deleted after 1 week.

Template

30 March 2024 09:20

```
stages:
  - build
  - test

.docker_template: &template
  tags:
    - CI
    - Devops
  rules:
    - if: SCI_COMMIT_BRANCH == "main"
  timeout: 1 hours 30 minutes

  docker_build_job:
    <<: *template
    stage: build
    before_script:
      - echo "Build Started"
      - sudo docker login registry.gitlab.com -u autopilotdevops -p $Access-Token
    script:
      - sudo docker build -t registry.gitlab.com/autopilotdevops/autopilot-devops .
      - sudo docker push registry.gitlab.com/autopilotdevops/autopilot-devops
    artifacts:
      paths:
        - SCI_PROJECT_DIR/

  docker_test_job:
    <<: *template
    stage: test
    before_script:
      - echo "Image Verification Started"
    script:
      - sudo docker images
      - sudo docker image inspect registry.gitlab.com/autopilotdevops/autopilot-devops
```

Retry

03 April 2024 19:10

The "retry" keyword is used to specify the number of times a job should be retried if it fails. This can be useful for jobs that occasionally fail due to transient issues such as network problems or resource constraints.

```
job_name:  
  script:  
    - echo "This is my job"  
  retry: 3
```

YML

04 April 2024 08:51

YAML stands for “YAML Ain’t Markup Language.” It is a data serialization language that is known for its simplicity, human readability, and ease of understanding.

YAML uses three node types:

- **Mappings** (Dictionaries): Unordered sets of key/value pairs, with unique keys.
 - animal: pets

pets:

- Cat
- Dog
- Goldfish

- **Sequences** (Arrays/Lists): Ordered series of nodes, which may contain duplicates or even themselves.

- Cat
- Dog
- Goldfish

- **Scalars** (Variables): Data presented as Unicode characters, including strings, numbers, and booleans.

integer: 25
string: "25"
float: 25.0
boolean: true

Indentation: Indentation is crucial in YAML to indicate nesting, similar to Python. Tabs are not allowed; only spaces can be used. The number of spaces doesn’t matter as long as it’s consistent within the same level.

Adding . In front of job will disable the

CI/CD

04 April 2024 09:46

Continuous Integration (CI):

- Continuous Integration is a development practice where developers integrate code into a shared repository frequently, preferably several times a day.
- The main goal of CI is to detect and fix integration errors quickly by automating the build and test process.
- In GitLab, CI is achieved using GitLab CI/CD, a built-in feature that allows for the automation of the build, test, and deployment processes.

Continuous Delivery (CD):

- Continuous Delivery is an extension of Continuous Integration where every change that passes automated tests is made ready for deployment to production manually, allowing for additional checks or approvals before releasing the changes.
- Delivery through its pipelines feature, enabling the automation of the build, test, and deployment processes while providing the flexibility to control when changes are deployed to production.

Continuous Deployment (CD):

- Continuous Deployment is an extension of Continuous Integration where every change that passes automated tests is automatically deployed to production.
- The goal of CD is to release software to production quickly and frequently, reducing the time between writing code and making it available to users.
- GitLab supports Continuous Deployment through its pipelines feature, allowing for the automated deployment of code to various environments.

Cache

15 February 2025 15:08

The cache keyword in your .gitlab-ci.yml file allows you to specify directories that should be cached between jobs. This can significantly speed up your pipelines by avoiding the need to re-download dependencies or rebuild artifacts in subsequent jobs.

```
job_name:  
  stage: build  
  cache:  
    paths:  
      - node_modules/  
      - .cache/  
      - build/  
  script:  
    - npm install  
    - npm run build
```

Docker

02 July 2025 18:02

Common Commands	Image Commands	<ul style="list-style-type: none"> docker images: List all images on the local machine docker pull [repository-host:port]/[owner]/image_name [:tag]: Pulls an image from a registry. If image already exists and there is an update, the delta would be pulled from server. Commands under [] are optional docker rmi <image_name> - Remove one or more images.
	Container Commands	<ul style="list-style-type: none"> docker ps : List running containers. docker ps -a : List all containers, including stopped ones. docker run <image_name> : Create and start a container from an image. docker start <container_name> : Start a stopped container. docker stop <container_name> : Stop a running container. docker rm <container_name> : Remove one or more containers. docker run -it <container_name> <command> : Execute a command inside a running container. <ul style="list-style-type: none"> Docker run -it ubuntu: opens container with interactive shell access docker system prune -f : removes all stopped containers docker system prune -a : removes all stopped containers and unused images. System prune doesn't touches running containers. docker logs containerName Docker rm \$(docker ps -aq): Removes non running containers Docker rm -f \$(docker ps -aq): Removes all containers -d :Run container in background -i :Attach standard input to the container -t :Attach standard output to the container --name myContainerName :Assign a name to the container --entrypoint=/bin/bash :Overrides the container's entry point
	Port Mapping	<p>Port mapping is crucial when exposing containerized services to external clients or users outside the Docker environment. Each service is attached to a docker port and by mapping host port to container port, the service can be accessed outside docker environment.</p> <pre>-p 8080:8080 :Map host port to container port. docker run -p 8080:8080 mongo docker run -p <host_port>:<container_port> <image_name></pre>
	Volume Mapping	<p>Used for managing data that needs to persist beyond the lifecycle of a container or for sharing data between the host system and the container.</p> <pre>-v /home/saurav/a:/user/share :Maps host directory to container directory</pre>
	Docker Network	<p>This is used to establish a connection between two containers and allow containers to communicate with each other, as well as with the host system and external networks. By default, containers running on separate Docker networks cannot communicate with each other unless explicitly allowed.</p> <pre>docker network create my_network //Create a new network with name my_network docker run -d --name my_nginx --network my_network nginx docker run -it --name my_ubuntu --network my_network ubuntu</pre>
	Docker Compose Commands	<ul style="list-style-type: none"> docker-compose up : Create and start containers defined in the docker-compose.yml file. docker-compose down : Stop and remove containers defined in the docker-compose.yml file.
Dockerfile	<p>FROM: Specifies the base image to use for the Docker image. It is the first instruction in a Dockerfile.</p> <ul style="list-style-type: none"> FROM alpine:latest <p>ADD: Adds files from the host machine to the Docker image.</p> <ul style="list-style-type: none"> ADD /my/Test.java /a/b/c/Test.java. This command copies Test.java from the path /my/ on the host machine to the path /a/b/c/ inside the image. <p>COPY: Copies files or directories from the host machine into the Docker image.</p> <ul style="list-style-type: none"> Example: COPY . /app . This copies all the contents from the current directory on the host machine to the /app directory inside the Docker image. <p>RUN: Runs a command in the shell to install packages or perform tasks.</p> <ul style="list-style-type: none"> RUN apt-get install java.This installs Java in the image. RUN apk --no-cache add openjdk17. This installs OpenJDK 17 without caching the installation files to keep the image size small. RUN pip install python. Installs Python using pip. RUN apk update && apk add python3 && pip3 install some-package. This updates the package lists and installs Python and pip using the apk package manager. <p>ENV: Sets an environment variable inside the Docker image.</p> <ul style="list-style-type: none"> ENV JAVA_HOME=/some/path. Sets the JAVA_HOME environment variable to /some/path. <p>WORKDIR: Sets the working directory for subsequent instructions like RUN, CMD, ENTRYPOINT, COPY, and ADD.</p> <ul style="list-style-type: none"> WORKDIR /a/b/c .Sets the default working directory to /a/b/c. <p>EXPOSE: Informs Docker that the container listens on the specified network port at runtime.</p> <ul style="list-style-type: none"> EXPOSE 8050 <p>CMD: Specifies the default command to run when the container starts.</p> <ul style="list-style-type: none"> CMD ["java", "-version"] Runs the java -version command when the container starts. <p>ENTRYPOINT: Configures the container to run as an executable.</p>	

	<table border="1"> <tr> <td>Building an Image</td><td>docker build . --file Dockerfile --tag singhsaurav/seleniumdocker:latest</td></tr> <tr> <td>Logging into Docker Hub</td><td>docker login</td></tr> <tr> <td>Image Tagging</td><td>docker tag my-python-app yourusername/my-python-appNew</td></tr> <tr> <td>Pushing an Image to Docker Hub</td><td>docker push yourusername/my-python-app</td></tr> </table>	Building an Image	docker build . --file Dockerfile --tag singhsaurav/seleniumdocker:latest	Logging into Docker Hub	docker login	Image Tagging	docker tag my-python-app yourusername/my-python-appNew	Pushing an Image to Docker Hub	docker push yourusername/my-python-app								
Building an Image	docker build . --file Dockerfile --tag singhsaurav/seleniumdocker:latest																
Logging into Docker Hub	docker login																
Image Tagging	docker tag my-python-app yourusername/my-python-appNew																
Pushing an Image to Docker Hub	docker push yourusername/my-python-app																
Compose	<p>Tool for defining and running multi-container Docker applications. It uses YAML files to configure the application's services and dependencies. The `docker-compose.yml` file defines the services, networks, and volumes for the application. It specifies how containers should be built, configured, and linked together.</p> <p>Services: Each container in a Docker Compose application is called a service.</p> <pre>version: "3" services: [service-name]: image: [image-name] container_name: [some-name] entrypoint: ["command to invoke"] depends_on: - [any-other-service] working_dir: /a/b/c environment: - KEY1=value1 - KEY2=value2 ports: - 80:80 - 1234:1234 - 8080:3344 volumes: - ./relative-path/host-path1:/a/b/c - /absolute-path/host-path234:/c/d/e</pre> <table border="1"> <tr> <td>docker-compose up</td> <td>Builds, (re)creates, starts, and attaches to containers for a service. Use `-d` flag to run containers in detached mode.</td> </tr> <tr> <td>docker-compose down</td> <td>Stops and removes containers, networks, and volumes created by `docker-compose up`</td> </tr> <tr> <td>docker-compose build</td> <td>Builds or rebuilds services specified in the `docker-compose.yml` file</td> </tr> <tr> <td>docker-compose pull</td> <td>Pulls images for services defined in the `docker-compose.yml` file</td> </tr> <tr> <td>Docker-compose up -d</td> <td>Detached mode</td> </tr> <tr> <td>Depends-on</td> <td>Will not wait for main container to start completely, it just denotes starting order on the containers, i.e depends on container will start after main has been triggered.</td> </tr> <tr> <td>Docker-compose ps</td> <td></td> </tr> <tr> <td>Docker-compose logs</td> <td></td> </tr> </table>	docker-compose up	Builds, (re)creates, starts, and attaches to containers for a service. Use `-d` flag to run containers in detached mode.	docker-compose down	Stops and removes containers, networks, and volumes created by `docker-compose up`	docker-compose build	Builds or rebuilds services specified in the `docker-compose.yml` file	docker-compose pull	Pulls images for services defined in the `docker-compose.yml` file	Docker-compose up -d	Detached mode	Depends-on	Will not wait for main container to start completely, it just denotes starting order on the containers, i.e depends on container will start after main has been triggered.	Docker-compose ps		Docker-compose logs	
docker-compose up	Builds, (re)creates, starts, and attaches to containers for a service. Use `-d` flag to run containers in detached mode.																
docker-compose down	Stops and removes containers, networks, and volumes created by `docker-compose up`																
docker-compose build	Builds or rebuilds services specified in the `docker-compose.yml` file																
docker-compose pull	Pulls images for services defined in the `docker-compose.yml` file																
Docker-compose up -d	Detached mode																
Depends-on	Will not wait for main container to start completely, it just denotes starting order on the containers, i.e depends on container will start after main has been triggered.																
Docker-compose ps																	
Docker-compose logs																	

Capital Markets

13 July 2020 17:51

Capital Markets	Market for financial Instruments.
Participants	Issuers, investors, and intermediaries.
Primary Markets	<ul style="list-style-type: none">• Securities markets• Derivatives markets• Currency markets• Insurance markets
Function	<ul style="list-style-type: none">• Channeling Wealth to Real Investment: Moving wealth to real investments like building factories, buying equipment, training workers, etc.• Pricing: Capital markets send signals on where returns are high or low, driving capital accordingly. High return = Attract more capital; low return = drive capital away• Liquidity: Facilitates converting investments into cash for consumption or reinvestment.• Managing Risk: Not just reducing risk but optimizing risk to suit one's risk tolerance or aversion. E.g., hedging risk through various financial instruments.
Securities Markets	<ul style="list-style-type: none">• Equity Securities: Ownership in a company, like stocks. Investors can earn dividends and benefit from stock price increases. They are paid after debt holders if the issuer is liquidated• Debt Securities: Loans to an issuer, such as bonds. Investors receive interest and principal repayment. They are paid before equity holders if the issuer is liquidated.• Hybrid Securities: Mix of debt and equity (e.g., convertible bonds).
Derivatives Markets	<p>Categories: Futures & forwards, options, swaps.</p> <p>Derivatives allow risk-sharing and are often used for hedging or speculation. Contracts based on the value of underlying assets. Used for risk management or speculation. Gains and losses are balanced among participants (zero-sum game).</p>
Currency Markets	Trading in national currencies Primarily facilitates cross-border trade, investment, and changing capital forms.
Insurance Markets	Securitization of insurance risks (e.g., catastrophe bonds). Similar to derivatives in that they optimize risk-sharing.

Financial Markets

14 July 2020 18:00

Category	Details	
Capital and Money Markets	Capital Markets	Long-term funding through debt or equity; includes activities like raising capital.
	Money Markets	Short-term funding instruments, typically repaid within a year.
Debt	Forms of Debt	Loans (negotiated) and bonds (transferable securities).
	Perpetual Debt	Debt instruments with no final repayment date, providing perpetual interest payments.
	Government Borrowing	Governments use debt (mainly bonds) for funding, often borrowing in foreign currencies if domestic markets are underdeveloped.
Types of Derivatives	Swaps	Contracts to exchange cash flows based on underlying assets.
	Forwards	Contracts to buy/sell assets at a future date at a price determined today.
	Options	Contracts giving the right (but not obligation) to buy/sell assets at a predetermined price.
Exchange vs OTC	Exchanges provide centralized trading with transparency OTC trading involves private negotiations without a centralized platform.	
Types of Markets	Primary Market	A market where new financial instruments are issued for the first time; public offerings are done through a prospectus.
	Secondary Market	A place where primary market instruments, once issued, are bought and sold. Provides an easy way to buy or sell financial assets.
	Stock Markets	Organized trading of stocks through exchanges, where orders are matched based on quoted price, time, quantity, and type.
	Bond Markets	Markets for issuing and trading bonds and other debt instruments. Includes government, municipal, and corporate bonds.
	Derivatives Markets	Markets for trading derivative instruments like futures and options.
	Money Markets	Markets for short-term debt instruments, such as certificates of deposit and commercial paper.
	Forex Markets	Global decentralized market for trading currencies, determining exchange rates.
	Commodity Markets	Physical or virtual marketplaces for buying, selling, and trading raw or primary products.

Financial Intermediaries

07 September 2024 11:24

Aspect	Role	Description	Usage/Function
Investment Banks(GoldMan Sachs)	Financial Advisor and Capital Raiser	Investment banks play a critical role in the issuance of new securities (such as IPOs or bond issuances) and act as underwriters, helping firms raise capital. They also provide advisory services related to mergers, acquisitions, and other corporate finance activities	<ul style="list-style-type: none"> Underwriting: Investment banks assess the risk and help companies sell their new securities to the public by buying these securities and reselling them to investors. Advisory: They advise companies on major financial decisions, such as going public, raising debt, or selling assets
Commercial Banks(JPMC)	Loan Provider and Trader	Offer loans, deposits, and may also engage in trading and underwriting.	Provide loans, handle deposits, engage in trading.
Broker(Robinhood)	Trade Executor	Executes trades on behalf of clients without capital risk.	Facilitate client trades, earn commissions.
Dealer(GoldMan Sachs)	Proprietary Trader	Trades securities for their own account, buying and selling to make a profit.	Provide liquidity, profit from bid-ask spread.
Broker-Dealer Firms(Charles Schwab)	Facilitator and Trader	Broker-dealers facilitate the buying and selling of securities for clients in secondary markets. They act as brokers (executing trades on behalf of clients) and dealers (trading on their own behalf).	Earn commissions, trade for own account.
Market Maker(Citadel Securities)	Liquidity Provider	Quotes bid (buy) and ask (sell) prices to ensure liquidity in the market.	Profit from bid-ask spread, maintain market efficiency.
Proprietary Trading(Hedge Fund)	Firm Trader	Firms trade using their own capital to profit from market movements.	Engage in short-term or long-term trading.

Equity

14 July 2020 11:11

Category	Details
Definition	Tradable securities representing ownership in a company, also known as shares or stock.
Ownership Representation	Represents a percentage of ownership in a company. Example: 10 shares represent 10% each; 100 shares represent 1% each.
Types of Equity Securities	<ul style="list-style-type: none"> Common Stock: Voting rights, dividends if declared. Preferred Stock: Fixed dividends, fewer voting rights, if a company declares bankruptcy, the preferred stock holders are paid before common shareholders <p>Priority level if company gets liquidated is:</p> <ul style="list-style-type: none"> Debt holders Preferred Shares Common Shares
Dividends	Cash Dividend: Monetary payout per share. Stock Dividend: Additional shares issued. Property Dividend: Tangible assets.
Important Dividend Dates	Declaration Date: Announcement of dividend. Record Date: Entitlement to dividend. Ex-Dividend Date: Cutoff for eligibility. Payment Date: When dividend is paid out.
Preferred Stock Types	Cumulative: Accrued dividends paid before common stock. Redeemable: Issuer can repurchase. Convertible: Can be converted to common stock. Participating: Additional dividends after common stockholders.
Depository Receipts (DRs)	Represent foreign shares and trade independently, facilitating domestic investment in foreign stocks. American Depository Receipts (ADR) The purpose of an ADR is to facilitate the domestic trading of a foreign stock. An ADR is a receipt for a specified number of foreign shares owned by an American bank. ADRs trade like shares, either on a U.S. Exchange or Over the Counter. The owner of an ADR has voting rights and also has the right to receive any declared dividends. An example would be Infosys ADRs that are traded in NASDAQ.
Exchange-Traded Funds	Instruments tracking assets or indices, traded on stock markets like individual stocks.
Private Equity	Financing for non-public companies, filling the gap between personal capital and public markets.
Stock Exchanges	Exchanges provide primary and secondary markets.
Algorithmic Trading	Uses algorithms for automated trading decisions and execution. High-frequency trading focuses on rapid order execution.
Key Order Types	Market Orders: Buy/sell immediately at the best price. Limit Orders: Buy/sell at specified price or better. Stop Orders: Buy/sell when price reaches a specified level. Stop-Limit Orders: Combines stop and limit orders.
Other Order Types	Day Orders: Expire at the end of the trading day if not filled. Good 'til Canceled (GTC): Remain until executed or canceled. Fill-or-Kill (FOK): Must be executed immediately in full or canceled. Immediate-or-Cancel (IOC): Must be executed immediately, with unfilled portions canceled.
Dividend Yield	Ratio of annual dividend per share (DPS) to share price: $\text{Dividend Yield} = \text{DPS} / \text{Share Price}$ Example: DPS = USD 0.50, Share Price = USD 10.00; Dividend Yield = 5%.
Earnings Per Share (EPS)	Calculated by dividing annual net income by the number of outstanding shares: $\text{EPS} = \text{Net Income} / \text{Number of Outstanding Shares}$ Example: Net Income = USD 10 million, Shares = 1 million; EPS = USD 10.00 per share.
Price/Earnings (P/E) Ratio	Relates share price to EPS: $\text{P/E Ratio} = \text{Share Price} / \text{EPS}$ Example: EPS = USD 0.50, Share Price = USD 10.00; P/E Ratio = 20 times.

Fixed Income

14 July 2020 11:12

What Is a Bond?	A debt instrument representing money borrowed by the issuer, with repayment of principal and periodic interest payments.			
Key Bond Features	Term	Definition		
	Face Value (Par Value)	The amount the bondholder will be repaid at maturity. Most bonds have a par value of \$1,000, though this can vary.		
	Coupon Rate	The interest rate the bond issuer agrees to pay the bondholder, typically a percentage of the face value (e.g., 5% coupon on a \$1,000 bond pays \$50 annually).		
	Coupon Payments	The periodic interest payments made to bondholders, typically semi-annually or annually, based on the coupon rate.		
	Maturity Date	The date when the bond's principal or face value is repaid to the bondholder. Maturities can range from a few months to over 30 years.		
	Issuer	The entity that issues the bond and borrows the funds. Issuers include governments, municipalities, corporations, and international organizations.		
	Yield	The return a bondholder gets on the bond. Types include Current Yield: Annual coupon payment divided by the bond's current market price Yield to Maturity (YTM): Total return expected if the bond is held to maturity, considering coupon payments and price difference.		
	Credit Rating	Bonds are rated by agencies (e.g., Moody's, S&P, Fitch) based on issuer creditworthiness. Higher-rated bonds (e.g., AAA) are safer, while lower-rated bonds offer higher yields but carry more risk.		
	Market Price	Bonds can be traded in the secondary market, and their price can fluctuate above or below face value based on factors like interest rates, inflation, and the issuer's creditworthiness.		
Types of Bonds	Type of Bond	Definition	Examples/Types	Features
	Government Bonds	Bonds issued by national governments to finance spending. Considered one of the safest investments, especially in stable countries.	U.S. Treasury Bonds, Gilts (UK), Sovereign Bonds	Low risk, used as a benchmark for risk-free rates.
	Municipal Bonds (Muni Bonds)	Bonds issued by local governments or agencies to fund public projects like schools, roads, utilities.	General Obligation Bonds, Revenue Bonds	Interest income often tax-exempt; safer than corporate bonds, riskier than government bonds.
	Corporate Bonds	Bonds issued by corporations to raise capital for business activities such as expansion or research.	Investment-Grade Bonds, High-Yield (Junk) Bonds	Higher yields than government bonds, credit quality varies with company's financial health.
	Zero-Coupon Bonds	Bonds issued at a deep discount with no regular interest payments; mature at face value. The difference between purchase price and face value is the return.	U.S. Treasury STRIPS	Ideal for lump-sum payout at maturity; more sensitive to interest rate changes.
	Convertible Bonds	Corporate bonds that can be converted into a set number of shares of the issuing company's stock.	-	Potential for capital appreciation, lower coupon rate than regular corporate bonds.
	Inflation-Linked Bonds	Bonds with principal or interest payments indexed to inflation, protecting purchasing power.	U.S. TIPS (Treasury Inflation-Protected Securities)	Attractive to investors concerned about inflation eroding fixed-income returns.
	Perpetual Bonds	Bonds with no maturity date, meaning they pay interest indefinitely.	-	Issuer pays interest forever; higher coupon rates due to perpetual nature.
	Foreign Bonds	Bonds issued in a country by a foreign entity in the domestic currency of that country.	Yankee Bonds (U.S.), Samurai Bonds (Japan)	Currency risk for investors; payments in the issuer's domestic currency.
	Callable Bonds	Bonds that allow the issuer to redeem the bond before the maturity date, usually at a premium.	-	Issuers may call bonds if interest rates drop; offer higher yields to compensate for call risk.
Price vs. Yield	Yield has an inverse relationship with bond price. If bond price rises, yield falls, and vice versa.			
Types of Fixed Income Instruments	Bonds: Loans by governments or corporates, with interest paid as the coupon rate. Par/Face/Principal Value: Amount returned at maturity; prices fluctuate with interest rates. Coupon Rate: Annual interest rate, typically paid every six months; zero-coupon bonds do not pay periodic interest. Maturity Period/Term to Maturity: Date on which principal value is paid; ranges from 1 month to over 40 years.			
Classification of Bonds	<ul style="list-style-type: none"> Central Government Bonds (Treasuries): Highly liquid, low-risk. Treasury Bills (T-bills): Short-term loans maturing within a year. 			

	<ul style="list-style-type: none"> • Treasury Notes: Bonds maturing within one to ten years. • Treasury Bonds: Bonds maturing after ten years. • Municipal Bonds: Issued by local governments. • Corporate Bonds: Issued by corporations. • Mortgage Bonds: Secured by property. • Collateral Trust Bonds: Secured by financial assets. • Zero-Coupon Bonds: Issued at a discount, redeemed at face value; difference between purchase price and face value is the return.
Default Risk	Refers to the risk of not receiving the promised money in full or on time.
Interest Rate Risk	Comprises Price/Market Risk and Reinvestment Risk .
Purchasing Power Risk (Inflation/Deflation)	Uncertainty about what the bondholder can buy with future payments. Inflation decreases purchasing power; Deflation increases it, benefiting bondholders. Longer maturities are more exposed to this risk.
Liquidity Risk	<ul style="list-style-type: none"> • The ease of converting an investment into cash. • Government securities are highly liquid, while less liquid investments offer higher returns but increased risk.
Sources of Return	Cash Flows (coupons and principal repayments) Reinvestment Income (return on cash reinvested) Price Changes (driven by yield changes).

Bond Calculations

14 July 2020 11:14

Primary characteristics of a bond :

- Price of the bond
- Rate of return
- Maturity period
- Price

Bond is nothing but a series of cash flow. When you buy a bond you have negative cash flow, and when you get the coupons, they can be equalled as positive cash flows. Thus we need to discount the cash flows with the interest rates prevalent. The rate used should be the opportunity cost of capital, i.e. the rate of a similar risk – same maturity bond.

We will first try to understand the various rates that are normally mentioned:

- **Spot Rate:** Spot rate is the interest rate on an investment starting today and ending after some specified day. Thus 3 year spot rate from today means the rate of return on an instrument with a three-year maturity and starting from today.
- **Forward Rate:** A forward rate is an interest rate contracted today on an investment that will be initiated after some time in future. In other words it is spot rate in future. Forward rates and spot rates are interlinked.

Years	Spot rates
1	5%
2	6%

- Here Row 1 shows that if one invests 100 Rs. today then after 1 year it will be 105 Rs.
- Also Row 2 shows that if one invests 100 Rs. today, after 2 years it will become $100 * (1.06)^2 = 112.36$ Rs.

1-year forward rate from next year: For that, you will have to see that one year forward rate (fr1) has to satisfy following equation:

Investing in 100 Rs today for one year and then re-investing the payments received after the first year for another year = Investing the 100 Rs today for two years.

i.e. sr1 = Spot Rate for 1 year

sr2 = Spot Rate for two years

1fr1 = Forward rate for 1 year starting one year from now

Therefore, $100 * (1 + sr1)(1 + 1fr1) = 100(1 + sr2)^2$

$$100(1.05)(1+fr1) = 100(1.06)^2$$

$$1fr1 = (112.36/105) - 1 = 7\%$$

Now why have we told you about the concept of forward rates and spot rates? For one thing, you will have to know which rate to use while discounting the cash flows of a bond. Whenever we calculate the NPV we use the spot rate. The reason is because we want to discount the cash flow to the present date. Hence any rate that we use should identify the rate from present date. So what will we do if we just have forward rates? In that case we will calculate the spot rates based on the forward rates. Actually the price of a bond fluctuates. This is so because the total amount of return from a bond is fixed. Thus as the spot rate of the economy changes the present value of the future amounts would also vary. To make it less confusing let's look at this example.

Suppose you have a bond with a face value of 100 \$ and will give a 10 % coupon for next three years. So the pay offs are:

	Returns	Investment	Yearly Returns
Present Yr	0	100	-100
First Yr	10	0	10
Seconds Yr	10	0	10
Third Yr	110	0	110

$$\text{Bond Price} = \sum_{i=1}^n \frac{C}{(1+r)^n} + \frac{F}{(1+r)^n}$$

where C = Periodic coupon payment,

- F = Face / Par value of bond,
- r = Yield to maturity (YTM) and
- n = No. of periods till maturity

So you can see that the total investment is 100 \$ and the returns are 130\$. The 10\$ in the third year does not have the same value as 10\$ in the first year. So the actual returns will have to be the discounted values of the cash flows to the present year.

Let us assume that the spot rate of a bond having similar risk is 10% in first 2 years and 5 % in third year. Thus after calculating NPV the table looks like

Year	Returns	Investment	NPV	Total yearly Return
Present Year	0	100\$	100\$	-100\$
First year	10\$	0	9.09\$	+9.09\$
Second Year	10\$	0	8.26\$	+8.26\$
Third Year	110\$	0	95.02\$	+95.02\$
	Total Inflow (Add 2,3,4 rows)			+112.37\$

$$9.09 = 10 / (1 + 0.1)$$

$$8.26 = 10 / (1 + 0.1)^2$$

$$95.02 = 10 / (1 + 0.05)^3 + (100 / (1 + 0.05)^3)$$

Yield

The yield of a bond is, roughly speaking, the return on the bond i.e. overall return expected from a bond, considering both the coupon payments and any capital gains or losses if held until maturity. The yield is expressed as the annual rate based on the par value.

There are several types of yield:

- **Nominal** yield is the same as the coupon rate. It is the rate of income that you receive based on the par values of the bond. Supposing you buy a bond with par value of 1000 Rs. and a coupon of 100 Rs Annually. Then the nominal yield is $100 / 1000 = 10\%$.
- **Current** yield is a measure of the return on the bond based on the current price. Since the current price of a bond fluctuates the current yield of a bond also fluctuates. If the par value of a bond is 1000 Rs with a coupon of 120 Rs (i.e. 12%), and the current price is 1200 Rs then the current yield is 10%.
- **Yield to maturity** is the overall return on the bond if it is held to maturity. It reflects all the interest payments that are available through maturity and the principal that will be repaid, and **assumes that all coupon payments will be reinvested at the current yield on the bond**. This is the most valuable measure of yield because it reflects the total income that you can receive. If you purchase the bond at a discount, yield to maturity will reflect the fact that at maturity you will have additional income because of the difference between the price paid and the principal returned. For example, if you purchase a \$1,000 par value bond for \$800, you have \$200 extra in income at maturity. The calculation is slightly different if the bond is purchased at a premium, but it takes into account the extra amount paid for the bond.

	Returns	Investment	Total yearly Return
Present Year	0	100\$	-100\$
First year	10\$	0	10\$
Second Year	10\$	0	10\$
Third Year	110\$	0	110\$

Nominal Yield:

The nominal yield is the coupon rate, which is

Coupon value/ original price of bond

$$= 10 / 100 = 10\%$$

Current Yield

Current yield is:

Coupon value/ Current market price

$$= 10 / 112.37 = 8.89\%$$

Yield to Maturity

YTM for the bond is the rate at which the NPV of the bond is zero. Thus the YTM is when "r" when:

$$112.37 = 10 / (1 + r) + 10 / (1 + r)^2 + 110 / (1 + r)^3$$

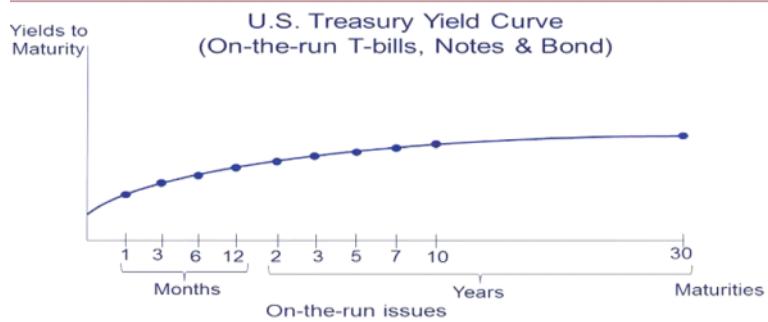
Hence YTM is 5.4%

- Normal Curve: Upward sloping; longer-term yields higher than short-term.
- Inverted Curve: Short-term rates higher than long-term rates; potential recession signal.
- Flat or Humped Curve: Similar short-term and long-term rates, or medium-term rates differing from short/long-term.

Yield Curve: graphical representation of interest rates (yields) for securities of the same type but with different maturities. It shows the relationship

between the interest rates and the time to maturity of those securities

Yield Curves



FX

07 September 2024 12:47

Aspect	Explanation
Definition of FX Trading	FX trading involves buying and selling currencies in the foreign exchange market. It facilitates international trade, investments, and finance by allowing currency conversion.
Key Participants	<ul style="list-style-type: none"> • Banks/Financial Institutions: Conduct large volume trades on behalf of clients or themselves. • Corporations: Exchange currencies for international trade. • Governments/Central Banks: Manage reserves and rates. • Investors/Hedge Funds: Speculate on currency movements. • Retail Traders: Trade small amounts via brokers.
Currency Pairs	<p>Currencies are traded in pairs (e.g., EUR/USD). The price indicates how much of the quote currency (USD) is needed to buy one unit of the base currency (EUR). Example: EUR/USD = 1.15 means 1 EUR = 1.15 USD.</p> <p>Buying EUR/USD: A trader buys the EUR/USD pair if they believe the Euro will strengthen against the U.S. Dollar (the exchange rate will rise).</p> <p>Selling EUR/USD: A trader sells the EUR/USD pair if they believe the Euro will weaken against the U.S. Dollar (the exchange rate will fall).</p>
Types of FX Markets	<ul style="list-style-type: none"> • Spot Market: Immediate currency exchange at current market prices. • Forward Market: Currency exchanged at a future date, agreed rate (used for hedging). • Futures Market: Standardized contracts traded on exchanges for future delivery. • Options Market: Gives the right (not obligation) to exchange currency at a future date/price.
Trading Mechanism	FX trading is OTC (over-the-counter), meaning trades happen directly between participants (banks, brokers, investors) via electronic platforms or internal systems.
Trading Hours	FX market operates 24 hours during the working week, spanning multiple time zones (London, New York, Tokyo, Sydney), ensuring continuous liquidity.
Bid/Ask Spread	<p>Bid Price: Price at which traders are willing to buy a currency.</p> <p>Ask Price: Price at which traders are willing to sell a currency.</p> <p>Spread: Difference between bid and ask price, representing transaction costs. A narrower spread typically indicates a more liquid market, while a wider spread can indicate lower liquidity or higher transaction costs.</p>
FX Risks	<p>Currency Risk: Fluctuations in exchange rates.</p> <p>Interest Rate Risk: Changing interest rates affect currency values.</p> <p>Credit Risk: Default risk in OTC transactions.</p> <p>Liquidity Risk: Difficulty trading without affecting price.</p>

Hybrids

14 July 2020 11:15

Aspect	Description
Definition	Financial instruments that combine features of both debt and equity.
Types of Hybrids	Convertible Bonds: Can be converted into shares. Convertible Preferred Shares: Fixed dividends and convertible to common stock. Perpetual Bonds: Indefinite interest payments, no maturity. Preferred Stock: Fixed dividends, priority over common stock. Subordinated Debt: Lower ranking debt, riskier. Contingent Convertible Bonds (CoCos): Convert to equity on trigger events.
Benefits	Income generation through regular payments. Potential for capital gains via conversion to equity. Diversification of risk/reward profile.
Risks	Interest rate sensitivity affecting prices. Credit risk, as they may be repaid last in distress. Market risk related to underlying stock performance.
Purpose	Capital Flexibility: Raise capital with less debt impact. Attractive to Investors: Fixed income with equity upside. Balance Sheet Benefits: Improve debt-to-equity ratio without immediate dilution.
Example	Convertible bonds provide regular interest and the option to convert to stock, offering the potential for appreciation if the stock performs well.

Derivatives

14 July 2020 17:26

A derivative is a product whose value is derived from the value of an underlying asset, index or reference rate. The underlying asset can be equity, foreign exchange, commodity or any other item.

Feature	Description
Underlying Asset	The asset, index, or benchmark that determines the derivative's value (e.g., stocks, bonds, commodities).
Leverage	Derivatives allow traders to control a larger position with a smaller amount of capital, magnifying both gains and losses.
Contractual Agreement	Derivatives are agreements between parties that specify terms, including the underlying asset, price, expiration date, and settlement method.
Market Complexity	Derivatives can be complex and involve various factors, including interest rates, volatility, and time until expiration.
Regulatory Framework	Derivatives trading is subject to regulation to ensure transparency, reduce systemic risk, and protect investors.

Type	Description
Forwards	Customized contracts between two parties to buy or sell an asset at a specified future date and price. Not traded on exchanges and have counterparty risk.
Futures	Standardized contracts to buy or sell an asset at a predetermined price on a specific date in the future. Traded on exchanges, which reduces counterparty risk.
Options	Contracts that give the holder the right, but not the obligation, to buy or sell an asset at a specified price before a certain date. Types include call and put options.
Swaps	Agreements between parties to exchange cash flows or other financial instruments. Common types include interest rate swaps and currency swaps.
Exotic Derivatives	More complex derivatives with features not found in standard derivatives, often tailored to specific needs (e.g., Asian options, barrier options).

Options

10 October 2024 09:55

An option is a contract that gives the buyer the right, but not the obligation, to buy or sell shares of the underlying security at a specific price (the strike price) on or before a specified date (the expiration date). The value of options depends on the value of an underlying investment, which can be a stock, index, currency, commodity, or other securities.

Key Features of Options:

- **Underlying Asset:** Options are based on various underlying assets.
- **Strike Price:** The price at which the holder can buy (call option) or sell (put option) the underlying asset.
- **Expiration Date:** The date by which the option must be exercised.
- **Premium:** The price paid for the option, which is the cost to purchase the option contract.

Type	Description
Call Option	Gives the holder the right to buy the underlying asset at the strike price before the expiration date. Typically used when the investor expects the asset price to rise.
Put Option	Provides the holder with the right to sell shares of the underlying stock at the strike price at the maturity date. Typically used when the investor expects the asset price to fall.
American Options	Can be exercised at any time before or on the expiration date.
European Options	Can only be exercised on the expiration date itself.
Exotic Options	More complex options with unique features (e.g., Asian options, barrier options) that may depend on certain conditions or averages over time.
Stock Options	A contract that guarantees the investor the right to buy or sell shares of the underlying stock at a fixed price prior to a certain date. The buyer is known as the holder, and the seller is called the writer. If the option contract is exercised, the writer must fulfill the terms of the contract.
Index Options	A contract that guarantees the investor the right to buy or sell the underlying index at a fixed price prior to a certain date. Settlement is always on a cash basis.

Strategy	Description	Market Outlook
Covered Call	Holding a long position in an asset and selling call options on the same asset to generate income.	Neutral to moderately bullish.
Protective Put	Buying a put option while holding a long position in the underlying asset to limit potential losses.	Bullish (but seeks downside protection).
Straddle	Buying both a call and a put option with the same strike price and expiration date, profiting from large price movements in either direction.	Expecting high volatility.
Strangle	Buying a call and a put option with different strike prices but the same expiration date, similar to a straddle but usually cheaper.	Expecting high volatility.
Bull Call Spread	Buying a call option at a lower strike price and selling another call option at a higher strike price to limit potential losses.	Bullish (moderate upside).
Bear Put Spread	Buying a put option at a higher strike price and selling another put option at a lower strike price.	Bearish (moderate downside).
Iron Condor	Selling a bear call spread and a bull put spread simultaneously to profit from low volatility in the underlying asset.	Neutral (expecting low volatility).
Calendar Spread	Buying and selling options with the same strike price but different expiration dates, profiting from time decay and volatility differences.	Expecting moderate volatility.

Forwards and Futures

14 July 2020 17:41

Futures: A futures contract is a standardized, transferable, exchange-traded contract that requires delivery of a commodity, bond, currency, or stock index, at a specified price, on a specified future date. Unlike options, futures contracts convey an obligation to buy. Futures are formal in nature and regulated.

Forward: The mechanism of forward contract is the same as the ones of a futures contract. The only difference is that the forward contract is not traded on an exchange. A forward contract is a direct agreement between two investors

Aspect	Forwards	Futures
Customization	Highly customizable	Standardized
Trading Venue	Over-the-counter (OTC)	Traded on exchanges
Settlement	Settled at expiration	Can be settled physically or in cash
Counterparty Risk	Higher due to lack of regulation	Lower due to exchange clearinghouses
Margin Requirements	None (unless otherwise agreed)	Requires margin deposits
Regulation	Less regulated	Highly regulated

Swaps

14 July 2020 17:49

Definition	Swaps(OTC) are agreements between at least two counter-parties to exchange cash flows in the future according to a pre-specified formula. They can therefore be regarded as portfolios of forward contracts. The most common one is an agreement on the exchange of a fixed rate for a floating rate contract.																																
Interest Rate Swaps	<ul style="list-style-type: none"> • Exchange of one set of cash flows for another, based on a pre-set index, notional amount, and set of exchange dates. • The most common type is fixed-for-floating rate swaps. • Example: CTS pays ICICI Bank a fixed interest of 7% on 100 crore rupees, while ICICI pays CTS an interest of inflation rate + 1%. • Helps manage interest rate risk by shifting exposure between fixed and floating rates. <table border="1" style="margin-left: auto; margin-right: auto;"> <thead> <tr> <th></th><th>Month 1</th><th>Month 2</th><th>Month 3</th><th>Month 4</th><th>Month 5</th><th>Month 6</th><th>Total</th></tr> </thead> <tbody> <tr> <td>Inflation rate</td><td>5</td><td>5.25</td><td>6</td><td>6.75</td><td>5.5</td><td>5.5</td><td></td></tr> <tr> <td>ICICI pays</td><td>6</td><td>6.25</td><td>7</td><td>7.75</td><td>6.5</td><td>6.5</td><td>40</td></tr> <tr> <td>CTS pays</td><td>7</td><td>7</td><td>7</td><td>7</td><td>7</td><td>7</td><td>42</td></tr> </tbody> </table> <p>So CTS pays 42 crores whereas ICICI pays 40 crores. Do you think CTS went in loss? Well actually it has, but if you see that the loss of 2crore has helped CTS transfer the risk of inflation rate volatility. Thus ICICI has in a way bought the inflation rate risk for Rs 3 crores.</p> <p>In case the inflation rate would have increased to 8 or 9 % then ICICI would have had to pay the additional money to CTS, and CTS would have been completely protected from the high inflation rates.</p> <p><i>Note:</i> In case you haven't noticed, the 100 crores of principal amount have NOT been exchanged. The only exchange is of the interests and the principal is just a notional amount.</p>		Month 1	Month 2	Month 3	Month 4	Month 5	Month 6	Total	Inflation rate	5	5.25	6	6.75	5.5	5.5		ICICI pays	6	6.25	7	7.75	6.5	6.5	40	CTS pays	7	7	7	7	7	7	42
	Month 1	Month 2	Month 3	Month 4	Month 5	Month 6	Total																										
Inflation rate	5	5.25	6	6.75	5.5	5.5																											
ICICI pays	6	6.25	7	7.75	6.5	6.5	40																										
CTS pays	7	7	7	7	7	7	42																										
Currency Swaps	<ul style="list-style-type: none"> • Involves exchanging principal and interest payments denominated in different currencies at an agreed exchange rate. • Unlike interest rate swaps, principal amounts are exchanged at the start and the end of the contract. • Used by multinational corporations to manage foreign currency exposure. 																																
Equity Swaps	<ul style="list-style-type: none"> • Exchange of cash flows where at least one leg is linked to an equity index or stock. • Example: One party pays a fixed or floating interest rate, while the other pays the returns on an equity index like S&P 500. • Used by investors to gain exposure to stock markets without directly buying the securities. 																																
Credit derivatives	<ul style="list-style-type: none"> • A contract where one party transfers credit risk to another by paying a fee. • Example: A bank holding risky corporate bonds can hedge default risk by purchasing a CDS from another institution. • Used in managing credit risk and mitigating losses from bond defaults. 																																
Asset Based Securities	A 1000 people take loans from ICICI bank with a combined value of 100 crore Rs. ICICI bank decides to get money based on these loans. So it groups together the loans worth 100 crores Rs. and starts selling them as securities. Now the investors buy the securities. The dividends that they get are nothing but the interest payments from these investors. Of course some of the investors may not pay the interests and some may default altogether. This will fluctuate the price of the ABS. Also the change in the interest rates will change the price of the underlying loans, thus changing the value of the assets																																
Commodity Swaps	<ul style="list-style-type: none"> • Exchange of cash flows based on the price of a commodity, such as oil, gold, or agricultural products. • Used by producers and consumers to hedge against price fluctuations. • Example: An airline company may enter an oil swap to lock in fuel costs and avoid exposure to rising prices. 																																
Volatility Swaps	<ul style="list-style-type: none"> • Allows parties to speculate or hedge against the future volatility of an underlying asset. • Unlike other swaps, the payout is based on realized volatility rather than price movements. 																																
Key Concepts in Swaps	<ul style="list-style-type: none"> • Counterparty Risk: Since swaps are OTC contracts, there is a risk that one party defaults, leading to credit exposure. • Mark-to-Market (MTM) Adjustments: The value of swaps is frequently recalculated based on market conditions. • Discounting & Collateralization: Institutions discount future swap cash flows using risk-free rates and may post collateral to mitigate credit risk. • ISDA Agreements: Most swap contracts follow the International Swaps and Derivatives Association (ISDA) Master Agreement to standardize terms and mitigate disputes. 																																

Facilitation Institutions

30 July 2020 09:01

Entity	Explanation	Analogy	Example
Custodian	Safely holds securities and handles income collection, settlements, and reporting.	Bank locker for stocks	Holds your Apple shares and collects dividends.
Exchange	Marketplace where securities are traded with standard rules and transparency.	Supermarket	NYSE or Nasdaq brings buyers and sellers together.
Clearing Firm	Ensures trade settlements between buyers and sellers, reducing risks.	Referee in a trade	Guarantees you get your shares and the seller gets paid.
Clearing Bank	Handles money transfers for clearing firms to settle trades.	Payment gateway	Moves your payment to the seller when you buy shares.
Depository Trust Company (DTC)	Ensures ownership changes hands smoothly through electronic settlement.		Transfers Google shares digitally from seller to buyer.

Buy Side and Sell Side

30 July 2020 09:12

Buy-Side

- Buy-side firms focus on making investments
- Refers to investment institutions like mutual funds, pension funds, and insurance firms that purchase securities for portfolio management.
- Buy-side analysts work in non-brokerage firms, conducting research exclusively for internal money managers.
- Their main focus is to assess whether an investment aligns with the firm's strategy and portfolio needs.
- Recommendations from buy-side analysts are **internal-only** and not available to the public.
- They often use research from sell-side analysts but conduct their own independent analysis before making investment decisions.

Sell-Side

- Sell-side firms provide research, execution, and capital-raising services.
- Includes brokerage firms, investment banks, and research departments that **sell securities** and provide investment recommendations to the public.
- Sell-side analysts work for brokerage firms and issue public ratings such as "strong buy," "outperform," "neutral," or "sell."
- Their focus is on analyzing investments to guide their firm's clients in buying or selling stocks.
- Sell-side firms also **place orders on exchanges** on behalf of buy-side clients and ensure best execution prices.
- They assist companies in raising capital by creating and selling new securities.

How They Interact

- A buy-side firm (e.g., a mutual fund) designs an investment portfolio and collaborates with a sell-side brokerage firm to place orders on an exchange.
- The sell-side ensures the best execution of these trades.
- If an institutional portfolio manager moves to a brokerage firm and becomes a registered representative, they transition from the **buy-side** to the **sell-side**.

Feature	Buy-Side	Sell-Side
Purpose	Invest in securities for portfolios	Advise and execute trades for clients
Entities	Mutual funds, pension funds, insurance firms	Brokerage firms, investment banks
Analyst Role	Internal research for investment decisions	Public recommendations on stocks
Output	Portfolio allocation, long-term growth	Stock ratings, price targets
Clients	Institutional investors	Retail and institutional traders

Front, Middle and Back Office

13 February 2019 00:19

Front Office	Revenue Generation & Client Interaction <ul style="list-style-type: none">Represents the client-facing part of the bank and involves in direct client interaction.Key functions include:<ul style="list-style-type: none">Sales and Trading: Advising clients on buying and selling securities. Helps companies raise funds in capital markets through IPOs, bond issuance, and mergers & acquisitions (M&A). Also provides financial advisory servicesResearch: Providing stock recommendations based on detailed analysis.Goals: Generate revenue and manage client relationships.
Middle Office	Risk & Financial Management <ul style="list-style-type: none">Acts as a bridge between the front and back office.Key responsibilities include<ul style="list-style-type: none">Risk Management: Ensuring activities comply with regulations and do not exceed risk thresholds.Compliance & Strategy Teams: Ensure adherence to laws and organizational strategy.Financial Controls: Monitor and control financial processes.Purpose: Maintain checks and balances for front office operations.
Back Office	Operations & Technology Support (Non-client-facing roles) <ul style="list-style-type: none">IT and Technology Teams: Maintain infrastructure, tools, and systems.Operations: Ensure the smooth processing of transactions and business functions and compliance with regulatory requirements

Trade Life Cycle

09 February 2021 11:43

Stage	Description
Issuance	<ul style="list-style-type: none"> Overview: The trade life cycle starts with the creation of a financial instrument (e.g., stocks, bonds). For example, a company raises capital through an Initial Public Offering (IPO), where shares are sold to the public for the first time, or a government may issue bonds via auctions. Key Insight: Post issuance, these securities may be listed on exchanges, such as the NYSE, for secondary market trading. Some securities, like government bonds, may not be listed and may be traded Over the Counter (OTC).
Pre-Trade	<p>Pre-Trade Analysis: Investors or firms conduct research to analyze the security and evaluate its impact on their portfolio. This stage involves:</p> <ul style="list-style-type: none"> Risk Management: Evaluate liquidity and credit risks. Portfolio Management: Analyze how the trade affects the overall portfolio. Client Onboarding: Verify clients' authorization to trade and prepare necessary documentation. Compliance & Regulations: Ensure adherence to trading laws (MiFID, Dodd-Frank). Collateral Management: Use collateral to reduce risk. Legal Agreements: Establish proper controls before trading, especially for OTC derivatives.
Trade Execution	<p>Order Placement: Orders to buy/sell are placed through brokers or exchanges.</p> <p>Trade Motivation: Investors trade for liquidity, hedging, or portfolio diversification.</p> <p>Trade Communication: Institutions interact via exchanges or OTC markets.</p>
Clearing and Settlement	<p>This stage involves multiple processes that ensure the trade is accurate, both sides agree on the terms, and the assets and cash are transferred between parties.</p> <p>Clearing</p> <ul style="list-style-type: none"> Matching & Confirming: Ensure trade details align between parties. Trade Capture & Enrichment: <ul style="list-style-type: none"> Record details (asset type, price, quantity, date). Add security identifiers, counterparty data. Trade Validation: Verify data accuracy. Affirmation & Confirmation: Participants confirm trade execution. Trade Reporting: Report transactions to regulators. Settlement Instructions: Prepare final steps for asset transfer. <p>Settlement</p> <ul style="list-style-type: none"> Final exchange of securities for cash. Methods: <ul style="list-style-type: none"> Delivery vs. Payment (DVP): Ensures cash and securities transfer simultaneously, reducing risk. Free of Payment (FOP): Delivery and payment occur separately, increasing default risk. Key Dates: <ul style="list-style-type: none"> Trade Date: Agreement date. Value Date: Scheduled exchange date. Settlement Date: Actual exchange date; if delayed, settlement failure may occur.
Post-Trade	<ul style="list-style-type: none"> Record Keeping: All the trade details are recorded for accounting and regulatory purposes. Reporting: Reports are generated for clients and regulators. Custody: Where the securities are held and managed.

Clearing and Settlement Process

15 February 2021 11:22

Clearing is the process of updating the accounts of the trading parties and arranging for the transfer of money and securities.

Settlement is the actual exchange of money and securities between the parties of a trade on the settlement date after agreeing earlier on the trade **Process**

At the end of each trading day, concluded or locked-in trades are received. All trades concluded during a particular trading period are settled together. A multilateral netting procedure is adopted to determine the net settlement obligations (delivery/receipt positions) of Broker-Dealers. The netting procedure determines the delivery and receipt obligation of funds and securities by each member. Settlement is deemed to be complete upon declaration and release of pay-out of funds and securities.

Players involved in the Process

Brokers-Dealers: Also called members. These are the persons or the firms who are allowed to trade in an exchange.

Clients / Custodian: The settlement of the trades (money and securities) done by a member-broker on his own account or on behalf of his individual, corporate or institutional clients may be either through the broker-dealer himself or through a registered custodian appointed by him/client. In case the delivery/payment in respect of a transaction executed by a broker-dealer is to be given or taken by a registered custodian, then the latter has to confirm the trade done by a broker-dealer on the trading system.

- Clearing House
- Exchange

Preparing for Clearing and Settlement

There are 4 major steps involved in the run-up to clearing and settlement. These are:

- **Trade Matching:** This occurs when the counterparties to the trade match the details of completed trades to ensure that both sides to a trade agree in advance of the settlement. This reduces the chances of a fail due to mismatched instructions. This is usually done systematically by the exchanges and by the brokers themselves
- **Confirmation:** Confirmations are sent to investors to officially tell them what their brokers have done on their behalf. Retail investors get paper confirmations while institutional investors get electronic confirmations, though they can get paper confirmations too if they so desire. In the US, electronic confirmations are obtained through the Depository Trust Company's (DTC) automated system for securities held by the DTC. If a customer identifies an error in the confirmation, he is required to inform the broker promptly.
- **Affirmation:** Institutional investors are required to affirm the broker's confirmation; otherwise the custodian bank must let the trade fail. The affirmation acts as a positive signal to the custodian to release funds or securities during settlement.
- **Netting:** A process of bringing together all of the trades made by the participants. The various trades by a participant for a specific security offset each other, thereby creating a single net debit or credit position for the participant at the end of the day for cash as well as for each security.

Netting can be of 2 types:

Bilateral Netting: In bilateral netting, each broker nets all of his trades with each of the other counterparties.

Multilateral Netting: In multilateral netting, all of each broker's trades with each of the other brokers are simultaneously netted.

	A	B	C
A ->	0	2	3
B ->	4	0	7
C ->	5	3	0

Figure 121 : Netting

Case 1: No Netting:

In the absence of netting, the number of shares that have to be transferred between A, B, C are shown in the diagram below.

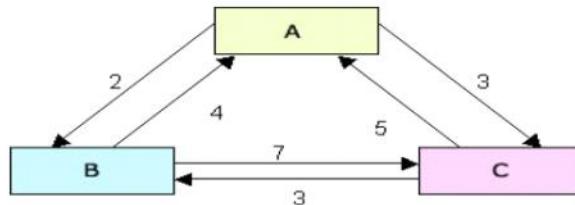


Figure 22: No Netting

Case 2: Bilateral Netting:

In this case transactions between a pair brokers offset each other. For instance, instead of A transferring 2 shares to B and B transferring 4 shares to A, there will be a single transaction in which B will transfer 2 shares to A. Thus the final transactions will be as shown below.

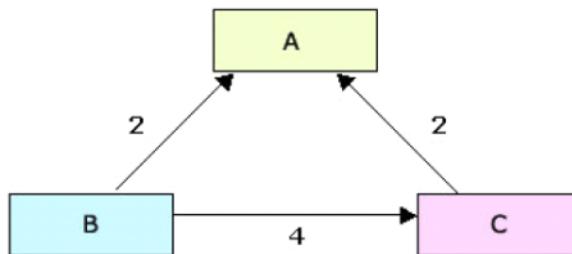


Figure 22: Bilateral Netting

Case 3: Multilateral Netting:

In this case, a broker takes part in one transfer only. For instance, instead of B transferring 2 shares to A and 4 to C, B will transfer 6 shares to a body known as Clearinghouse (we shall look at this in detail later).

Trading systems based on bilateral netting are more efficient than those that don't have netting (also known as trade-for-trade). Multilateral netting in turn is more efficient than bilateral netting. Now suppose that in the example above, B defaults i.e. fails to deliver the stock. Both A and C would suffer from the default, as they would not get delivery of the stock. To prevent this, there are bodies known as clearinghouse who steps into the middle of a trade, becomes the counter-party to both the buyer and the seller and guarantee completion of the transaction. It guarantees that it will complete trades on the original terms, even if the original contra-party fails. This process of transferring obligation from one party to the other is also known as "Novation".

Reconciliation

18 February 2021 11:32

The process of proving that a securities trading firm's books and records are accurate is commonly known as reconciliation. An efficient trading firm seeks proof that its books and records are accurate, by comparing each component of its securities positions and cash balances with the outside world on a daily basis, as well as ensuring that its books and records reconcile internally.

Types

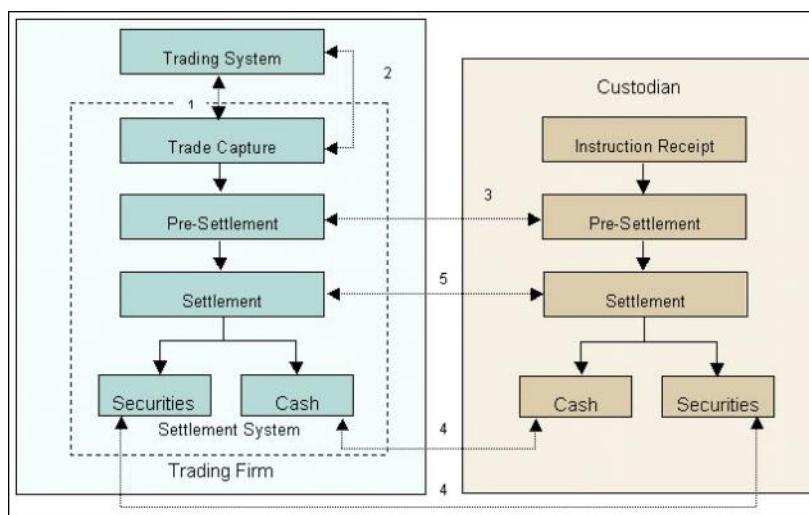
Trade-by-Trade Reconciliation: This is an automated reconciliation that proves that all trades captured by the traders are successfully captured within the settlement system. Not conducting this reconciliation can result in failure to process individual trades. The possibility of trades not coming into the settlement system is very much there. That's why this reconciliation is important. Its importance will increase as the settlement cycle shrinks.

Trading Position Reconciliation: This reconciliation is designed to prove that the trade dated security positions calculated by the trading system agree with the equivalent positions calculated by the settlement system. Not conducting this reconciliation can result in trades being executed from the incorrect trading position, where the trading system is found to be incorrect. Discrepancies can arise where trades have not been received by the settlement system, and where trade amendments and cancellations have been effected within one system but not the other. To be certain of being in a fully reconciled position, both the trade-by-trade and trading position reconciliations should be done.

Open Trades Reconciliation: This reconciliation aims to prove that open trades held within internal books and records are actually open at the relevant custodian, or generally have the same status at the custodian.

Custodian Position Reconciliation: The aim of this reconciliation is to prove that settled securities positions held within internal books and records agree with the equivalent positions as advised by the custodian. This is therefore another fundamental control and confirms to the firm that the securities that it believes are held by the custodian are actually held by the custodian.

Settlement System Integrity Reconciliation: This reconciliation intends to prove that the settlement system is in balance i.e. the total of securities owned is equal to the sum of the location position. If the firm employs double-entry bookkeeping, theoretically this will be true by default. Still this reconciliation must be performed frequently to prove that the system is in balance.



Methods of Reconciliation

The reconciliation described above can be categorized into either non-cumulative reconciliation or cumulative reconciliation.

Non-cumulative Reconciliation

The trade-by-trade, open trade, settlement and custody system integrity reconciliation do not involve an opening and closing balance. Rather they are 'snapshot' type reconciliation. Such reconciliation is called non-cumulative reconciliation.

Cumulative Reconciliation

The trading position and the custodian position reconciliation involve opening balances, movements and closing balances and are therefore called cumulative reconciliation.

Options Explanation

23 July 2021 16:59

1. Assigning :assigning an existing owned option to the exercise

Net outcome

Cost of Buy =(1* 100)

Proceeds from sale = (2*100)

Net Proceeds=100

Strike price of Owned Option :100

Strike price of sold option: 120

Market price: 150

Net proceeds from assignment for seller of the call= (100*120)- (100 *100)+ Net Proceeds from

initial sale of option

=2000+100

=2100

Net outcomeCost of Buy = (1*100)Proceeds from sale = (2*100)Net proceeds = 100Strike price of
Owned Option: 100Strike price of sold option: 120Market price: 150Net proceeds from assignment
for seller of the call= (100*120)-(100*100) + Net Proceeds from initial sale of option.<= 2000 = 2000
+100=2100 2100

Net proceeds from exercise for the exerciser (person who bought the option you sold)

= (Market Price *100)- (Cost of Option+(Strike Price* 100))

=15000- (200+12000)

=15000- 12200=2800

Basic Option Strategies

- **Buy 2 TSLA Jan 30 calls @4(Bullish)**

1. Owner has a right to buy 2x100 shares of TSLA Before expiration at price of 30
2. 2 Contracts=200 Shares
3. Options expire on Saturday following 3rd Friday of the Month
4. 30 is Strike Price
5. Calls: Right to Buy
6. 4 is Premium
7. Jan 30 is Series: Combination of expiration and Strike price

Out of Money option:Market Value(20)<Strike Price-> Option wont be exercised and owner will suffer a loss of 800(4X2X100).

In the Money option: Market Value(50)>Strike Price ->Owner will exercise the option and gain 3200(20-4=16,16x200=3200)

At the Money option: Market value=Strike Price

Max Gain: Unlimited

Max Loss: 4/share,

Break Even: when share goes to 34

- **Write 10 TSLA May 40 call @3(Bearish)**

Writing(Selling Calls/Naked Call): obligated to deliver/sell stock at strike Price, options are written to generate income.

Max Gain: When market value=0, 3 Per share

Max Loss:Unlimited

Break Even: market Value:43

- **Buy 1 TSLA Oct 30 put @4(Bearish)**

Buying Put Options Right to Sell:

In the Money: MV<SP

Out of Money: MV>SP

Max gain: When MV=0, MG=26/share

Max Loss: 4/share

Break Even: 26 share price

CRD

13 October 2022 08:30

Order Life Cycle	<p>Prelim: When the order is created</p> <p>Pre Alert: Alerts and warnings are created</p> <p>Pre-ok: Once all the alerts and warning are closed order moves to pre ok status</p> <p>New: When send to trading is done</p> <p>Open: when trader takes ownership</p> <p>Work: either the trader has accepted the order or its in flex</p> <p>Ready: order is filled</p> <p>Accounting: Order is recorded in books of accounts</p>
Manager Workbench	<p>It is a portfolio management tool where we can see whole portfolio at a glance and based on that we can take the action. Like Portfolio allocation. 60% equity, 30% fixed income, 10% crypto.</p> <p>A Portfolio manager cannot edit on each security manually instead he can edit on values . For eg . I want to sell 10000\$ shares, then ill say -10000\$ then 10000 will be deducted and automatically sell order will be created called as slicing</p>
Rebalance	<p>Means matching Portfolio returns with Market returns, and doing buy/sell whenever its required. In CRD we select account and select model. Model like high risk taking individuals ,moderate risk taking individual, conservative individual.</p> <p>Navigation To Rebalnce Window--> Manager workbench --> Accounts --> select model --> enter Account number --> press ok-</p> <p>Then we will see Cur %(2.2) and Model%(1.97) Click on 2nd last dropdown icon and click rebalance.</p> <p>There will be some discrepancies and once rebalance is done all discrepancies will be resolved and sell orders will be created at one go</p>
Compliance	<p>user restrictions placed for order so that they don't violate regulatory requirements or customer requirements or any other conditions put on it</p> <p>Eg . cannot trade any security which is selling tobacco For amazon I don't want 20% of Amzn exposure</p> <p>If these rules are violated we get alerts</p> <p>Warnings approved by portfolio manager</p> <p>Alerts can be approved by algo team and assign compliance team , if compliance team is happy they will approve and then order can go ahead .</p> <p>If its reject -PM has to correct the order he has to reduce the amount.</p> <p>Once approved order is moved to pre ok status Compliance runs on whole basket.</p> <p>-Post compliance run if order is split , orders should retain completed status -Post compliance run if order is merged , each allocation should go to prelim</p> <p>Cancel send to accounting done by trader Its done due to :</p> <ol style="list-style-type: none"> 1. We want to revert the order 2. We want to replace it 3. Manual correction is required for the order. <p>Action-> cancel send to account It will cancel and create another order in ready status</p>