

Automation Framework Set 1

(Interview Questions 1)

1. Explain your Automation Framework?

Note: Whenever you answer this question, then make sure that you follow the reverse Approach while describing the whole framework component

Start from Jenkins Project -> Scheduler -> parameter capturing-> and then whole Framework.

(One of the Key point to remember is the choice of your words. And the depth you go to explain the answer)

a. Overview

- i. We have created a **single Hybrid framework** that helps us to test **(UI), (API) and Mobile Components** of our entire system
- ii. The Benefit:
 - a. of this approach was that we really don't need to maintain 3 different repositories.
- iii. Our git branching Model helps us making sure that everyone is working on the updated piece of code and the test code only becomes part of the **CT pipeline when it's approved by the team lead.**
 1. The benefit of this approach is that **we tell our team to write good and well**
 2. **Maintainable tests** that add values in our automated tests.

b. Jenkins Project:

- i. The Jenkins Plan that we have created is a **Parameterized Jenkins Plan**
- ii. All the Run time arguments that are required for the framework i.e
 1. Test Type: (Sanity/Smoke/E2e)
 2. Component Type: (API/UI/Android)
 3. Env Type : Dev/QA/UAT
- iii. These parameters are selected to from the Jenkins Project and Passed to our Frameworks- **Runner Class (which has the main function inside it).**



Few Extra Questions:

- How to Create Parameterized Job in Jenkins and Read them in Code?
- How to Read Runtime Arguments in Java?
- Explain your Branching Model?

c. Runner Class:

- i. All the selected Parameters from Jenkins are passed to the Runner Class which has the main function as the Run time arguments
- ii. These Run time arguments are validated and based on them our Dynamic TestNG.xml file is created and executed!
- iii. **Example:**
 1. If component type is api then our package whose name is com.api.tests will be executed
 2. In **com.api.tests** we may have 500 tests and out of those 500 tests we want to run only sanity
 3. TestType parameter helps in including those tests in our dynamic.xml file
 4. Lastly the env type helps in which property file to pick up from the config folder

d. Test packages:

- i. We have 3 tests packages
 1. **com.api.tests**
 2. **com.ui.tests**
 3. **com.api.tests**
- ii. While writing our tests we have ensured that all tests are
 1. ***(Characteristic of a good test) very important***
 - a. Small
 - b. Independent
 - c. Belongs to certain group
 - d. Easy to read
 - e. Have Proper Assertions
- iii. Design Pattern Used: Fluent Design Pattern
- iv. OOPS concepts Used: Inheritance/ Encapsulation/ this keyword/super keyword



Few Extra Questions:

- How did you create the Dynamic TestNG xml for execution? (Very Important)
- How do you run only specific Tests from your framework?
- How do you ensure that your team is writing High Quality Tests?

e. DAO Component

- i. DAO is our Data Access Object Component.
- ii. It consist of Special DAO classes which are mapped to our Database tables.
- iii. There is a DB helper class which follows the Singleton design pattern which helps in connecting to the Database
- iv. The functions in DAO classes:
 - 1. Fires the SQL query to the database and retrieves the values and create a LIST of POJO or a simple POJO object and returns it back
- v. We use DAO component functions for our DB Validation and E2E test
- vi. Design Pattern Used: Singleton Design Pattern
- vii. OOPS concepts Used: Inheritance/ Encapsulation/ this keyword/super keyword.



Few Extra Questions:

- What is Singleton Class and where you have used it?
- How do you connect to your database in Java?
- How did you do DB validation in your framework?
- Can a constructor of a class be private?
- What exceptions are shown when you try use JDBC?
- Which Database your project uses?
- What is DAO?

f. Page Classes Component:

- i. For UI Automation we have used Page Classes
- ii. For all **unique** component of UI there is a specific page class present
- iii. Each Page Class consist of
 1. Private static final By (Locators)
 2. A Parameterized Constructor
 3. Functionality of that page
 - a. Each function of the Page class returns some data or page object of the next UI Page.



Few Extra Questions:

- **Benefit of Page Classes**
- **Explain Page Object Design Pattern**
- **What the are the substitute of Page Object Design Pattern?**
- **What are page objects?**

g. Listeners Component:

a. There 2 major listeners in our Project

i. ITestListeners

1. This is used for generating reports
2. Logs
3. Taking screenshot of failed Test

ii. IRetryListeners

1. This is used to re run failed test dynamically



Few Extra Questions:

- Which Listeners Used in your Project?
- What are the different interfaces used in your Project?
- How to do you generate Report and Log in your Project?
- How do you run failed Test?

h. POJO Component:

- i. For Sending Request Body to our API Test we have used to the POJO
- ii. Each Request Body has a specific POJO Class



Few Extra Questions:

- How do you pass PAYLOAD or request body to the API Request
- What is the purpose of POJO?]
- How do you validate complex JSON Response?
- How do you convert POJO to JSON?
- How do your convert JSON to POJO?

i. Test Data Component

- i. Our Framework can use Test Data from
1. Excel using apache poi
 2. CSV using open csv
 3. Database using JDBC
 4. Fake Data Creation using JavaFaker



Few Extra Questions:

- How to read excel file in Java?
- How to read CSV file in Java?
- How to read Properties File in Java?
- How do you connect to Database in Java?
- How did you create Fake Data in your Framework?
- How did you achieve data driven testing in your framework?
- How do you pass test data from file to test

j. Report Folder

- i. Test Reports are kept in Report folder generated by Extent Report



Few Extra Questions:

- How did you create reports in your project

k. Screenshot Folder

- i. All failed tests are stored in the screenshot folder.



Few Extra Questions:

- How did you take screenshot in for failed test

I. Config Folder

- i. All Environment Configurations are stored in the Config Folder
 1. DEV.properties
 2. UAT.properties
 3. QA.properties



Few Extra Questions:

- **Purpose of Properties file**
- **How to read Properties file in Java**
- **What are the common Exceptions you have faced while reading Properties file**

m. Utility Component:

- i. Utility Component deals with reusable and common functions used across the framework
 1. Fake Data Creator
 2. Read CSV file
 3. Read Excel File
 4. Read Properties File
 5. OTP Generator
 6. Capture ScreenShot
- ii. All these functions are static.



Few Extra Questions:

- **Purpose of Utility Component**
- **Why the functions of Test Utils class are static?**
- **What is static keyword?**

n. Browser Utility Component:

- i. While creating the framework we are NOT supposed to use any Selenium function directly
- ii. All selenium functions will be wrapped under a special wrapper functions.
- iii. These wrapper functions create abstraction in your framework
- iv. Browser Utility is a class which contains these wrapper functions which consist selenium methods.
- v. Benefit of this approach:
 1. Complete Abstraction for Selenium WebDriver
 2. People who write Test Scripts are not required to learn selenium
 3. Learn Curve is small

o. Android Utility Component

- i. Exact SAME reason for Browser utility



Few Extra Questions:

- Where you have used Abstraction in your Project?
- How did you achieve abstraction

p. Step Definition

- i. All the cucumber Step Definition Java files are kept in Step Definition Package

q. Feature Files

- i. All our Cucumber Feature Files are Kept in `src/test/resource/features` folder



Few Extra Questions:

- What are Hooks in Cucumber
- What is Feature
- What is Scenario
- What is Scenario Outline
- What is DataTable
- How do you execute Cucumber Test using Jenkins?