n **YouTube Automation with Runnable JAR –**

**Step-by-Step Guide**

1. **YouTube.java – Main Test Class**

package com.test.demoqa;

import org.openqa.selenium.By;

import org.openqa.selenium.Keys;

import org.openqa.selenium.TimeoutException;

import org.openqa.selenium.WebDriver;

import org.openqa.selenium.WebElement;

import org.openqa.selenium.chrome.ChromeDriver;

import org.openqa.selenium.chrome.ChromeOptions;

import org.openqa.selenium.edge.EdgeDriver;

import org.openqa.selenium.firefox.FirefoxDriver;

import org.openqa.selenium.support.ui.ExpectedConditions;

import org.openqa.selenium.support.ui.WebDriverWait;

import org.testng.Reporter;

import org.testng.annotations.\*;

import io.github.bonigarcia.wdm.WebDriverManager;

import java.time.Duration;

import java.util.List;

import java.util.Scanner;

import java.util.concurrent.ExecutorService;

import java.util.concurrent.Executors;

import java.util.concurrent.Future;

import java.util.concurrent.TimeUnit;

public class YouTube {

WebDriver driver;

WebDriverWait wait;

@BeforeClass

public void launchBrowser() {

String browser = "chrome"; // default fallback

System.out.println("🧪 Enter browser name (chrome / ichrome / firefox / edge): (waiting 20 sec, default = chrome)");

ExecutorService executor = Executors.newSingleThreadExecutor();

Future<String> futureInput = executor.submit(() -> new Scanner(System.in).nextLine());

try {

browser = futureInput.get(20, TimeUnit.SECONDS); // Wait for user input

if (browser == null || browser.trim().isEmpty()) {

browser = "chrome";

}

} catch (TimeoutException e) {

System.out.println("⏰ No input received in 20 seconds. Using default browser: chrome");

browser = "chrome";

} catch (Exception e) {

System.out.println("⚠️ Error reading browser input. Using default: chrome");

browser = "chrome";

} finally {

executor.shutdownNow();

}

if (browser.equalsIgnoreCase("chrome")) {

Reporter.log("<b>🚀 Launching Chrome browser (Normal Mode)...</b><br/>", true);

WebDriverManager.chromedriver().setup();

ChromeOptions options = new ChromeOptions();

options.addArguments("--remote-allow-origins=\*");

driver = new ChromeDriver(options);

}

else if (browser.equalsIgnoreCase("ichrome")) {

Reporter.log("<b>🚀 Launching Chrome browser (Incognito Mode)...</b><br/>", true);

WebDriverManager.chromedriver().setup();

ChromeOptions options = new ChromeOptions();

options.addArguments("--remote-allow-origins=\*");

options.addArguments("--incognito");

driver = new ChromeDriver(options);

}

else if (browser.equalsIgnoreCase("firefox")) {

Reporter.log("<b>🚀 Launching Firefox browser...</b><br/>", true);

WebDriverManager.firefoxdriver().setup();

driver = new FirefoxDriver();

}

else if (browser.equalsIgnoreCase("edge")) {

Reporter.log("<b>🚀 Attempting to launch Edge browser...</b><br/>", true);

try {

WebDriverManager.edgedriver().setup();

driver = new org.openqa.selenium.edge.EdgeDriver();

Reporter.log("<b>✅ Edge browser launched successfully.</b><br/>", true);

} catch (Exception e) {

Reporter.log("<b>❌ Failed to launch Edge browser:</b> " + e.getMessage() + "<br/>", true);

Reporter.log("<b>🔁 Falling back to Chrome browser.</b><br/>", true);

WebDriverManager.chromedriver().setup();

ChromeOptions fallbackOptions = new ChromeOptions();

fallbackOptions.addArguments("--remote-allow-origins=\*");

driver = new ChromeDriver(fallbackOptions);

}

}

else {

Reporter.log("<b>🚀 Launching Chrome browser (Normal Mode)...</b><br/>", true);

WebDriverManager.chromedriver().setup();

ChromeOptions options = new ChromeOptions();

options.addArguments("--remote-allow-origins=\*");

driver = new ChromeDriver(options);

}

driver.manage().window().maximize();

driver.manage().timeouts().implicitlyWait(Duration.ofSeconds(10));

wait = new WebDriverWait(driver, Duration.ofSeconds(20));

}

@Test

public void playMostViewedSongFromSearch() throws InterruptedException {

String favoriteSong = "Zingat Ajay Atul";

Reporter.log("<hr/>", true);

Reporter.log("<h3>🎬 <u>Test Case:</u> Play Most Viewed Song from YouTube Search</h3>", true);

Reporter.log("<p><b>Description:</b> This test launches YouTube in the browser, plays your favorite song first (" +

"<i>"+favoriteSong+"</i>), and then allows the user to search for additional songs manually. " +

"For each search, the test automatically identifies and plays the video with the highest view count " +

"from the top 5 results. It also skips ads and switches to full-screen mode for better playback experience.</p>", true);

Reporter.log("<b>🔁 After playing the default song, the test enters an input loop to accept multiple song requests.</b><br/>", true);

Reporter.log("<b>✅ Purpose:</b> To automate YouTube video search and playback intelligently using Selenium WebDriver.<br/>", true);

Reporter.log("<hr/>", true);

Scanner scanner = new Scanner(System.in);

// ⭐ Play your favorite song "Zingat Ajay Atul" first

Reporter.log("<hr/><b>🎵 Playing your favorite song first:</b> <i>" + favoriteSong + "</i><br/>", true);

searchAndPlayMostViewedVideo(favoriteSong);

// 🔁 Start loop to let user play more songs manually

for (int attempt = 1; attempt <= 100; attempt++) {

Reporter.log("<hr/><b>🎧 Attempt " + attempt + "</b><br/>", true);

System.out.print(

"\n🎧 Attempt " + attempt +

": Enter the name of the song you want to play, or type 'Skip' to end the program ➡ "

);

String songName = scanner.nextLine();

// 🚪 Exit the loop if user types "Skip"

if (songName.equalsIgnoreCase("Skip")) {

Reporter.log("🛑 <b>Program skipped by user.</b> Exiting the loop.<br/><hr/>", true);

return;

}

// 🎬 Play the song with the highest view count

searchAndPlayMostViewedVideo(songName);

}

scanner.close();

}

public void searchAndPlayMostViewedVideo(String songName) throws InterruptedException {

// Open YouTube and search

driver.get("https://www.youtube.com/");

Reporter.log("🔍 <b>Searching on YouTube:</b> <i>" + songName + "</i><br/>", true);

try {

WebElement acceptCookiesBtn = driver.findElement(By.xpath("//button[contains(text(),'Accept')]"));

acceptCookiesBtn.click();

} catch (Exception ignored) {}

WebElement searchInput = driver.findElement(By.name("search\_query"));

searchInput.clear();

searchInput.sendKeys(songName);

Thread.sleep(1000);

searchInput = driver.findElement(By.name("search\_query"));

searchInput.sendKeys(Keys.ENTER);

Thread.sleep(3000); // wait for results

List<WebElement> videoTitleElements = driver.findElements(By.xpath("//a[@id='video-title']"));

List<WebElement> videoMetadataElements = driver.findElements(By.xpath("//ytd-video-renderer//div[@id='metadata-line']"));

if (videoTitleElements.size() == 0 || videoMetadataElements.size() == 0) {

Reporter.log("❌ <b>No videos found for:</b> " + songName + "<br/>", true);

return;

}

double highestViewCount = 0;

int indexOfMostViewed = 0;

Reporter.log("📊 <u>Top 5 Videos (View Count):</u><br/>", true);

for (int j = 0; j < Math.min(5, videoMetadataElements.size()); j++) {

String viewsText = videoMetadataElements.get(j).getText().split("\n")[0]; // "6.4M views"

String rawViews = viewsText.replace("views", "").trim(); // "6.4M"

double numericViews = convertToNumber(rawViews);

Reporter.log("&nbsp;&nbsp;&nbsp;&nbsp;📺 Video " + (j + 1) + ": " + rawViews + " ➡ " + numericViews + "<br/>", true);

if (numericViews > highestViewCount) {

highestViewCount = numericViews;

indexOfMostViewed = j;

}

}

WebElement mostViewedVideo = videoTitleElements.get(indexOfMostViewed);

String playingTitle = mostViewedVideo.getAttribute("aria-label");

mostViewedVideo.click();

Reporter.log("▶️ <b>Now playing:</b> <i>" + playingTitle + "</i><br/>", true);

try {

WebElement skipAdBtn = wait.until(ExpectedConditions.visibilityOfElementLocated(By.xpath("//button[contains(@id,'skip-button')]")));

skipAdBtn.click();

Reporter.log("⏩ <i>Skipped ad</i><br/>", true);

} catch (Exception ignored) {}

try {

WebElement fullscreenButton = driver.findElement(By.xpath("//button[@class='ytp-fullscreen-button ytp-button']"));

fullscreenButton.click();

Reporter.log("🖥️ Entered <b>full-screen mode</b><br/>", true);

} catch (Exception ignored) {}

Thread.sleep(3000);

}

@AfterClass

public void closeBrowser() {

Reporter.log("🛑 <b>Closing browser...</b><br/>", true);

if (driver != null) {

driver.quit(); // Uncomment when you're ready to close

}

}

/\*\*

\* Converts a YouTube view count string like "6.4M" or "1.2K" into a number.

\*/

public static double convertToNumber(String viewCountText) {

viewCountText = viewCountText.toUpperCase();

try {

if (viewCountText.endsWith("M")) {

return Double.parseDouble(viewCountText.replace("M", "")) \* 1\_000\_000;

} else if (viewCountText.endsWith("K")) {

return Double.parseDouble(viewCountText.replace("K", "")) \* 1\_000;

} else if (viewCountText.endsWith("B")) {

return Double.parseDouble(viewCountText.replace("B", "")) \* 1\_000\_000\_000;

} else {

return Double.parseDouble(viewCountText.replaceAll(",", ""));

}

} catch (Exception e) {

Reporter.log("⚠️ <b>Could not parse view count:</b> " + viewCountText + "<br/>", true);

return 0;

}

}

}

1. **YouTubeRunner.java – Runnable Class :-**

**package com.test.singletoon;**

**import org.testng.TestNG;**

**import com.test.demoqa.YouTube;**

**import java.util.Collections;**

**public class YouTubeRunner {**

**public static void main(String[] args) {**

**System.out.println("📦 Starting TestNG execution from Runnable JAR...");**

**// Create TestNG object**

**TestNG testng = new TestNG();**

**// Set the test class(es) to run**

**testng.setTestClasses(new Class[] { YouTube.class });**

**// Optional: enable HTML reports**

**testng.setDefaultSuiteName("YouTube Automation Suite");**

**testng.setDefaultTestName("YouTube Search & Play Test");**

**// Run the test**

**testng.run();**

**System.out.println("✅ TestNG execution finished.");**

**}**

**}**

To convert your Selenium TestNG-based YouTube automation project into a **runnable JAR file**, you need to do **two things**:

**✅ 1. Create a Runner Class with a main() Method**

This will **manually invoke TestNG** from Java, so it works as a standalone Java program (ideal for JARs).

**✅ 2. Export as Runnable JAR using Eclipse/IntelliJ**

**✅ Step 1: Create a Runner Class**

java

CopyEdit

package com.test.demoqa;

import org.testng.TestNG;

import java.util.Collections;

public class YouTubeRunner {

public static void main(String[] args) {

System.out.println("📦 Starting TestNG execution from Runnable JAR...");

// Create TestNG object

TestNG testng = new TestNG();

// Set the test class(es) to run

testng.setTestClasses(new Class[] { YouTube.class });

// Optional: enable HTML reports

testng.setDefaultSuiteName("YouTube Automation Suite");

testng.setDefaultTestName("YouTube Search & Play Test");

// Run the test

testng.run();

System.out.println("✅ TestNG execution finished.");

}

}

**✅ Step 2: Build Runnable JAR in Eclipse**

1. **Right-click your project > Export**
2. Choose: Java > Runnable JAR file
3. Click **Next**
4. In Launch configuration: select YouTubeRunner - com.test.demoqa
5. Choose the destination path for .jar
6. Select:
   * **Package required libraries into generated JAR** (recommended)
7. Click **Finish**

**✅ Notes:**

* Ensure YouTubeRunner.java is in your src folder.
* TestNG must be in your build path (.classpath or pom.xml if Maven).
* When running .jar, **ChromeDriver** must be accessible (via WebDriverManager or PATH).

**✅ Run JAR from CMD:**

bash

CopyEdit

java -jar YouTubeAutomation.jar