Challenges/Problems in Appium

1. Can I automate this app (Hybrid) using Appium?
Answer: No, not for release builds due to secure WebView restrictions. Use debug builds or Native context only.

---

2. Native app/Hybrid Native app: Will there be an ID or Accessibility ID available?
Answer: Not always. If built with Flutter/React Native/etc., ID and accessibility Id may be missing. Fall back to **XPATH**.

---

3. Will Appium support all the latest versions of Android and iOS?
Answer: Yes, provided all dependencies (Appium, drivers, SDKs) are kept up to date.

---

4. How do you ensure dependencies and software are updated?
Answer:
Android

- Update Android Studio

- Update SDK tools (cmd-line tools, build tools, platform-tools)

- Update Appium: npm install -g appium

- Update Appium drivers: appium driver update --installed

iOS

- Update macOS

- Update Xcode

- Update Xcode command-line tools

- Update Appium: npm install -g appium

- Update Appium drivers: appium driver update --installed

- Update Homebrew: brew update && brew upgrade

---

5. Can I change the port number of Server, Client, Bootstrap.jar installed in the device?
Answer: Not unless explicitly configured. These ports are hardcoded in most setups.

---

6. Have you faced a "Socket Hangup" error? How do you overcome it?
Answer:
Yes. This can occur when a port stops responding. Solutions include:

1. Always call driver.quit() after test completion

2. Respect newCommandTimeout wait (e.g., 60s)

3. Uninstall residual Appium-related apps (AUT, Appium Settings, UiAutomator)

4. Kill and restart Appium server and ADB

---

7. App not getting installed on device?
Answer: Possible causes:

- Admin/firewall restrictions (especially on office laptops)

- USB Debugging settings in custom OS

- Certain OEMs (Redmi, Realme) may prompt manual installation for the first time

---

8. If I change the VDM in the script, the script fails. How to fix it?
Answer:

- Restart Appium server

- Remove existing VDM server/client and Appium Settings manually from device settings

---

9. Cannot view the screen in Appium Inspector?
Answer:

- Session might be dead: Restart it

- If screenshot is not loading, try capturing one manually

- If screenshot is blocked due to security flags, ask devs for a debug build without the secure flag, or use driver.getPageSource() for manual inspection

---

10. IDs or Accessibility IDs not detected in recent app versions?
Answer:

- Check with developers for code changes or obfuscation

- Use XPATH if IDs are removed or not accessible

---

11. "Not a launchable activity" or "App permission denied" error when using only app in desired capabilities
Answer: Include appPackage and appActivity explicitly.

---

12. driver.activateApp(appPackage) fails with similar error?
Answer: Use driver.executeScript("mobile: startActivity", Map.of("appPackage", "pkg", "appActivity", "activity")); with correct appPackage and appActivity.

---

13. PointerInput swipe not working?
Answer:

- Start/End coordinates must be on scrollable elements

- Ensure coordinates don't overlap other elements

- Debug coordinates via Appium Inspector

---

14. Can CAPTCHA/Graph be automated?
Answer: No. Like Selenium, they appear as a single element and can't be parsed.

---

15. Can Toast messages be automated?
Answer:

- Yes, if visible for >3 seconds

- No, if <3 seconds (risk of NoSuchElementException)

---

16. High failure rate using TouchAction / MultiTouchAction?
Answer: Use PointerInput, which has a lower failure rate.

---

17. PointerInput issues on iOS?
Answer: Older Appium versions had issues triggering clicks on createPointerDown. Upgrade or adjust starting coordinates to avoid overlap.

---

18. Scroll using PointerInput not working?
Answer: Make sure the target element has scrollable=true. Element must be **scrollable** and coordinates should be **validated**.

---

19. platformName in capabilities but session not created?
Answer: Don't explicitly provide it if using AndroidDriver/IOSDriver; it's already set internally.

---

20. After context switch (NATIVE/WEBVIEW), I get NoSuchElementException
Answer: Reset implicit wait after context switch.

---

21. driver.hideKeyboard() doesn't hide numeric keyboard?
Answer:

- Tap outside the keyboard area

- On Android, press the back button or driver.navigate().back() to hide

---

22. NoSuchElementException for off-screen elements?
Answer: Appium interacts only with visible UI. Scroll to bring elements into view first.

---

23. ADB over Wi-Fi stops working after turning off data?
Answer: Wireless ADB depends on mobile data. Use USB connection for such scenarios.

---

24. WebView automation: Keys.Enter not working?
Answer:
Use platform-specific key events:
- Android: ((PressesKey) driver).pressKey(new KeyEvent(AndroidKey.ENTER))

- iOS: Tap keyboard buttons directly.

---

25. GetContextHandles() shows multiple WebViews?

Answer: Android lists all background WebViews. iOS lists only current visible app WebView.

---

26. WebView switching fails when using hardcoded name in iOS?

Answer: Use partial match (e.g., contains "WEBVIEW") instead of hardcoded names.

---

27. Cannot inspect WebView elements in Sauce Labs/BrowserStack?

Answer: WebView locators must be identified locally using Chrome/Safari Inspectors, then reused in automation.

---

28. Using noReset=true: what is the impact?

Answer:

- App cache is retained
- App is not relaunched automatically
- driver.quit() doesn't kill the app

Solutions:

- Use forceAppLaunch or shouldTerminateApp capabilities to override this behavior

---

29. Using multi-finger PointerInput fails?

Answer: Ensure each pointer has a unique name in the constructor.

---

30. App crashes mid-execution - how to debug?

Check logcat (Android) or device logs (iOS) for crash stack trace.

---

31. Dynamic content in app delays element visibility - how to handle?

Use WebDriverWait with expected conditions like visibilityOfElementLocated.

---

32. Parallel execution on multiple devices not working?

Ensure unique systemPort (Android) or wdaLocalPort (iOS) for each device in capabilities.

---

33. Unexpected popups block automation flow

Use driver.getPageSource() to detect unexpected popups and handle with conditional logic.