HeadSpin Appium Capabilities

Capability	Description	Values
headspin:appiumVersion	Appium version to be used for the test. Defaults to the latest stable version installed on the host. See Setting the Appium Version below.	e.g. `1.10.0`, `1.9.1`
headspin:capture	Enable video and network sessions capture. Default `false`.	`true`, `false`
headspin:capture.video	Enable video capture. When this is set, it takes precedence over `headspin:capture`. Also, `headspin:capture` does not need to be set for this take effect.	`true`, `false`
headspin:capture.network	Enable network capture. When this is set, it takes precedence over `headspin:capture`. Also, `headspin:capture` does not need to be set for this take effect.	`true`, `false`

Capability	Description	Values
headspin:capture.ignoreHosts	A list of ignored `host:port` regex patterns. These hosts are ignored from the network capture session. The pattern can be just a host name also, in which case all ports are matched.	e.g. `['abc\.example\.com:443', '.*\.mydomain.com']`
headspin:capture.networkConfig	Configures the network used by the device during the capture session.	Allowed keys are `shaping`, `redirectRules`, `spoofRules`, `headerRules`. See Network Configbelow for details of each.
headspin:testName	Full name of a test in the performance monitoring section. This is only valid if `headspin:capture` is `true`. Analysis metrics of the captured data will be exported to the user flow named by the value. For details on adding passed/failed/excluded status on the session, see User Flow Status below.	e.g. `app_load_test`

Capability	Description	Values
headspin:testData	Add custom measurements to the session. This data will be inserted into the performance test, for this session ID	e.g. `[{"key":"App Load Time", "value":20, "title": "Custom Metrics", "units": "seconds"}]`
headspin:quitOnDisconnect	Quit the session when client disconnects	`true`, `false`
headspin:restartDeviceOnSessionStart	Restarts the device before starting the test. Default `false`.	`true`, `false`
headspin:useAppiumUnlock	Unlock the device using appium's built-in mechanism when set to `true`. In all cases, HeadSpin's software+hardware "pintap" system will activate if the device is still locked, which will handle MDM and other hard lock cases. Default `false`.	`true`, `false`

Capability	Description	Values
headspin:removeAppPackages	Provide a list of package globs to uninstall before starting the test	e.g. `['com.example.app', 'com.xyz.app', 'com.foo.*']`
headspin:resetUiAutomator2	Uninstall `io.appium.uiautomator2.server` and `io.appium.uiautomator2.server.test` packages before starting the test. Default `false`.	`true`, `false`
headspin:controlLock	Control the device from remote control UI while the test is running. This should be used with tools like Appium Desktop where you may want to manually interact with the device during a session. UI will display the device screen in view only mode when the value is `false`. Default `false`.	`true`, `false`
headspin:useMjpegScreenshotUrl	Use HeadSpin MJPEG server to capture device screen. Defaults to `false`	`true`, `false`

Capability	Description	Values
headspin:logPath	Custom log path for storing Appium server logs	e.g. `/tmp/appium.log`

Load balancer capabilities

Capability	Description	Values
headspin:selector	A selector string of the device to use. This takes precedence over `deviceName` and `udid`. See selectorsfor the selector syntax. The load balancer will consider all the devices in the pool and use the healthiest that is most available to run the session.	`os: android os_version: >7.0`
headspin:requestTimeout	Handle a request timeout from AppiumLB to each devices. It affects only when a client communicate with the target Appium server via AppiumLB. Defaults to `90` sec.	e.g. `120`

Setting the Appium Version

New appium versions are made available to customers per customer request, as soon as they are certified on the platform. HeadSpin works with the Appium team to test and provide new versions ^3. A few select versions are available on every host by default, typically those used by the HeadSpin recorder, which work with the widest set of Android and iOS devices. The current recorder version is **1.11.1**.

You should always set `headspin:appiumVersion` to the version you are using, to avoid changes related to system updates in the future.

Network Config

Key Name	Description	Values
`shaping`	Network conditioning targets. Rate targets are upper bounds on the native link speed. Round trip time and packet loss are additive to the behavior of the native link. All values must be zero or positive. `down` is the download rate in megabit per second (mbps). `up` is the upload rate in megabit per second (mbps). `rtt` is the round trip time in millisecond. `loss` is the packet loss in percentage.	e.g. `{"down": 5.1, "up": 1.2, "rtt": 10, "loss": 0.5}`

Key Name	Description	Values
`redirectRules`	A list of rules `host_regex=destination_host`. If the host regex matches a host, the request will be rewritten as if the client sent it to `destination_host` instead. Backtick back references are allowed in the destination host.	e.g. `["foo\.com=bar.com", "(.*)headspin\.io=\1headspin.com"]`
`spoofRules`	A list of rules `host_regex=destination_host`. If the host regex matches a host, the request IP will be changed to the IP resolved by destination host, as if the DNS record for the original host were changed to the destination host's.	e.g. `["foo\.com=13.33.148.190", ".*\.headspin.io=13.33.148.51"]`
`headerRules`	A list of rules `host_regex=header:value`. If the host regex matches, the header will be injected into the request. Multiple headers of the same name will be comma separated in the final request. Backtick back	e.g. `[".*=X-Custom:MyApp"]`

Key Name	Description	Values
	references are allowed in the header and value.	

User Flow Status

Out of the box appium does not provide facilities to mark sessions passed, failed, or in some error state. To fill this gap and provide visibility across all your tests:

1. set `headspin:testName` on the session [*]
2. POST to the performance API to set the status of the session, using the session ID from the driver

For example the python script below sets the test status using requests.

```
from appium import webdriver
import requests

driver = webdriver.Remote ...

# status can be one of passed, failed, excluded
perf data = {
    'session_id'  driver.session_
    'status'  'passed'

requests.post 'https://api-dev.headspin.io/v0/perftests/upload'
json=perf_data
```

[*] If you did not do step 1, it is possible to set the `testName` after the fact also using the performance API.