

Associate a device with a Perfecto Connect tunnel

Let's have a quick look at the following Appium script.

```
public class AppiumTest {

    public static void main(String[] args) throws MalformedURLException, IOException {
        System.out.println("Run started");

        String browserName = "mobileOS";
        DesiredCapabilities capabilities = new DesiredCapabilities(browserName, "", Platform.ANY);
        String host = "demo.perfectomobile.com";
        // capabilities.setCapability("user", "testuser@perfectomobile.com");
        // capabilities.setCapability("password", "testpass");
        capabilities.setCapability("securityToken", "eyJhbGciOiJSUzI1IiwiaWF0IjoiYXc7BRhmkEo7qrA");
        capabilities.setCapability("deviceName", "DEVICEID");
        capabilities.setCapability("automationName", "Appium");
        capabilities.setCapability("tunnelId", "0a334599-210a-4d42-aa8e-445071a4070b");

        AndroidDriver driver = new AndroidDriver(new URL("http://" + host + "/nexperience
/perfectomobile/wd/hub"), capabilities);
        driver.manage().timeouts().implicitlyWait(15, TimeUnit.SECONDS);

        // Reporting client. For more details, see http://developers.perfectomobile.com/display/PD
/Reporting
        PerfectoExecutionContext perfectoExecutionContext = new PerfectoExecutionContext.
PerfectoExecutionContextBuilder()
        .withProject(new Project("My Project", "1.0"))
        .withJob(new Job("My Job", 45))
        .withContextTags("tag1")
        .withWebDriver(driver)
        .build();

        ReportiumClient reportiumClient = new ReportiumClientFactory().createPerfectoReportiumClient
(perfectoExecutionContext);

        try {
            reportiumClient.testStart("My test name", new TestContext("tag2", "tag3"));

            // write your code here
            Thread.sleep(5000);

            driver.get("http://local-web-server");

            Thread.sleep(20000);

            reportiumClient.testStop(TestResultFactory.createSuccess());
        } catch (Exception e) {
            reportiumClient.testStop(TestResultFactory.createFailure(e.getMessage(), e));
            e.printStackTrace();
        } finally {
            try {
                driver.quit();
            } catch (Exception e) {
                e.printStackTrace();
            }
        }
        System.out.println("Run ended");
    }
}
```

Following is a list of the capabilities we are setting for this execution:

- **securityToken** - your Perfecto security token
- **user** - your Perfecto username (use only if security token not supported for your Perfecto Lab)
- **password** - your password (use only if security token not supported for your Perfecto Lab)
- **deviceName** - the Device ID which you'd like to use (we're using a single device for demonstration purposes, but you may use several)
- **tunnelId** - the Tunnel ID received from **perfectoconnect**

This script makes use of the `tunnelId` capability, which configures the system to associate the device with this tunnel when opened. This means that all HTTP/S requests and responses generated by and for the device, during the execution of the script, are routed through the tunnel created earlier by `perfectoconnect`. This allows the device to access your **local network**. In the example above, `http://local-web-server` is a local web server that resides within your local network.

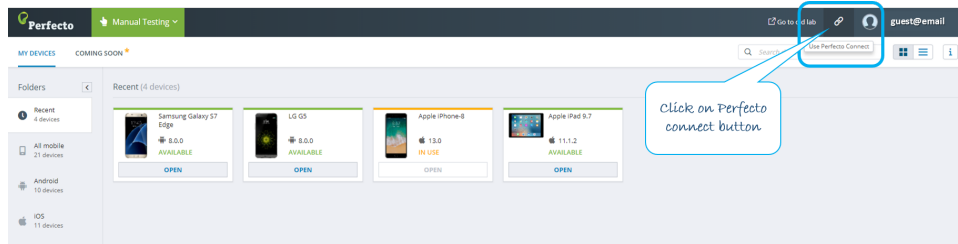
Important: Perfecto Connect for fast web devices is only supported with Selenium scripts. The script should provide the tunnel-id using the `tunnelId` capability, as shown above.

Connect via the Manual Testing view

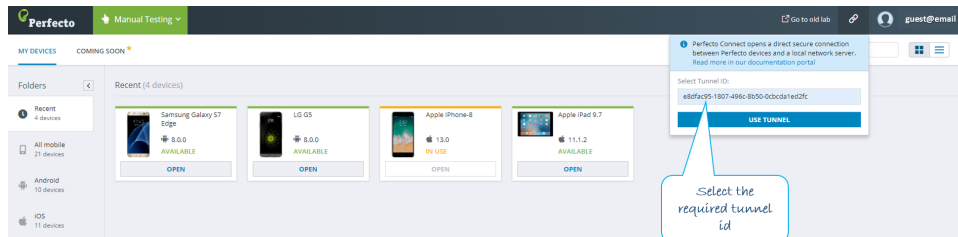
Prerequisite: Before you can use Perfecto Connect on mobile devices in Perfecto's **Manual Testing** view, you need to run the `perfectoconnect` client and save the `tunnelId`, as described above.

In the **Manual Testing** view, perform the following steps:

1. Click the **Perfecto Connect** icon .

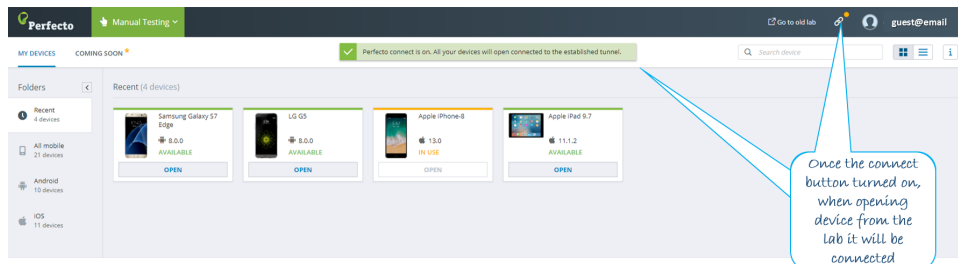


2. From the menu that opens, select the required `tunnelId`.



3. Open the required device or Web session.

When Perfecto Connect is turned on, the device you open in the lab automatically connects.



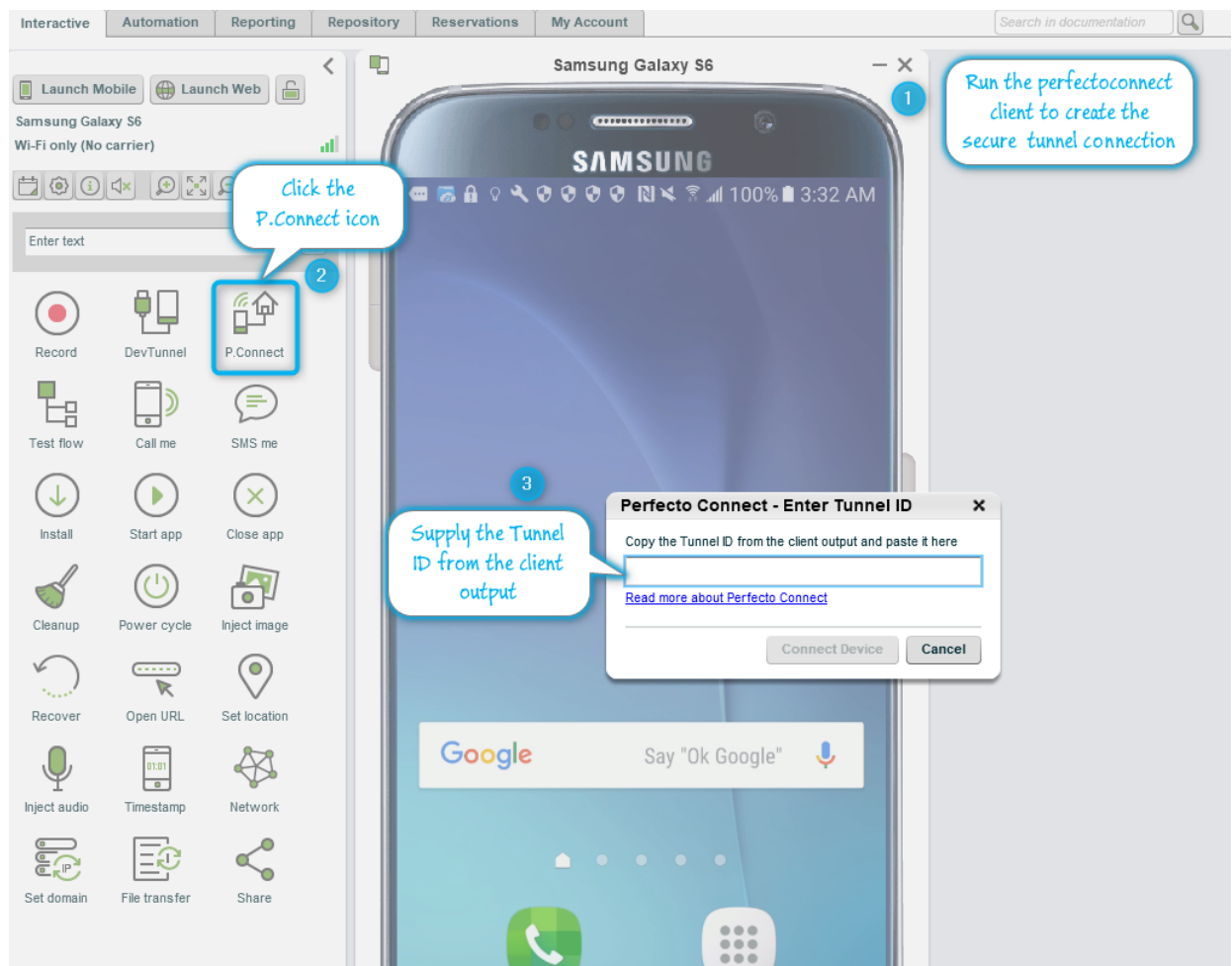
4. To end the session, click the **Perfecto Connect** button again to stop and end the connection.

Connect through the legacy UI

Prerequisite: Before using Perfecto Connect in Interactive mode on mobile devices in the Perfecto Lab, run the `perfectoconnect` client and save the `tunnelID`, as described above.

In Interactive mode, perform the following steps:

1. Launch the device to connect.
2. Click the **P.Connect** function icon, as shown in the following image.
3. Copy the `tunnelID` from the `perfectoconnect` client and paste it into the **Perfecto Connect - Enter Tunnel ID** dialog box.
4. Click **Connect Device**.



Connect through a Perfecto automation script

Prerequisite: Before using Perfecto Connect in Interactive mode, run the `perfectoconnect` client and save the `tunnelId`, as described above.

In your Automation script, perform the following steps:

1. Launch the device to connect.
2. Copy the `tunnelID` from the `perfectoconnect` client.
3. Use the `perfectoconnect.attach` function, supplying the `tunnelId` as the parameter value.
4. To disconnect the device, use the `perfectoconnect.detach` function, supplying the `tunnelId` as the parameter value.

For information on using Perfecto Connect with your CI job (Jenkins), click [here](#).