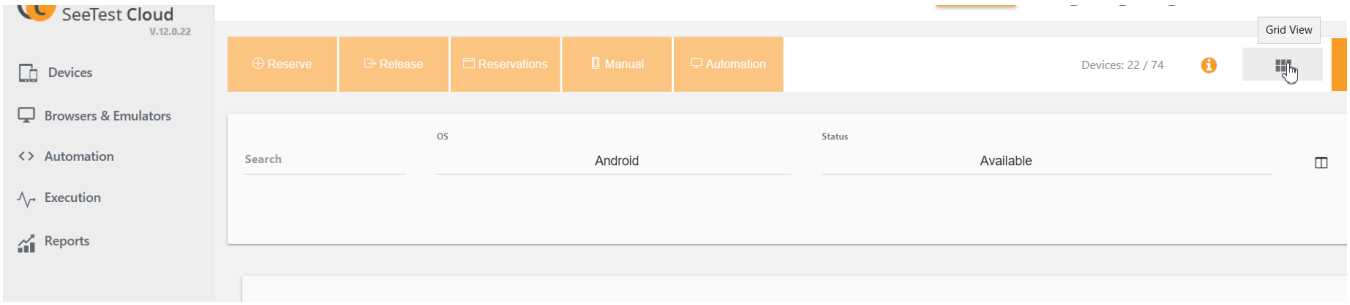
Running Automated tests using Reserved Device

This article demonstrates how to run automated tests on a device reserved in **seetest.io**

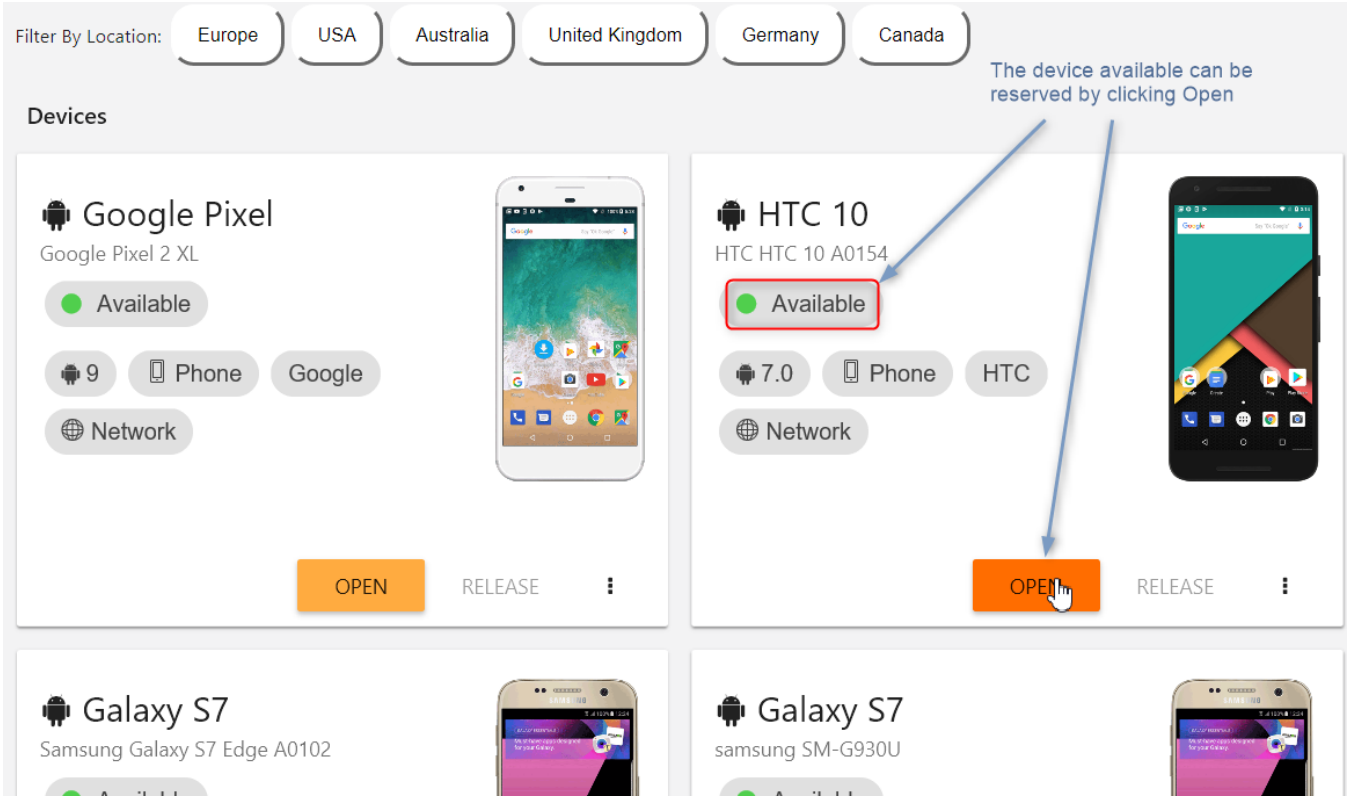
There are times when you would want to run test, on a device of your choice. It makes perfect sense as a test user, to reserve a device first and then run the test against the device in such circumstances.

Reserve Device and making it ready for Automation

Logon to seetest cloud with your credentials and make sure you are in the **Grid View**. You can switch to grid view as shown as below.



Reserve the device you want to test as shown below.

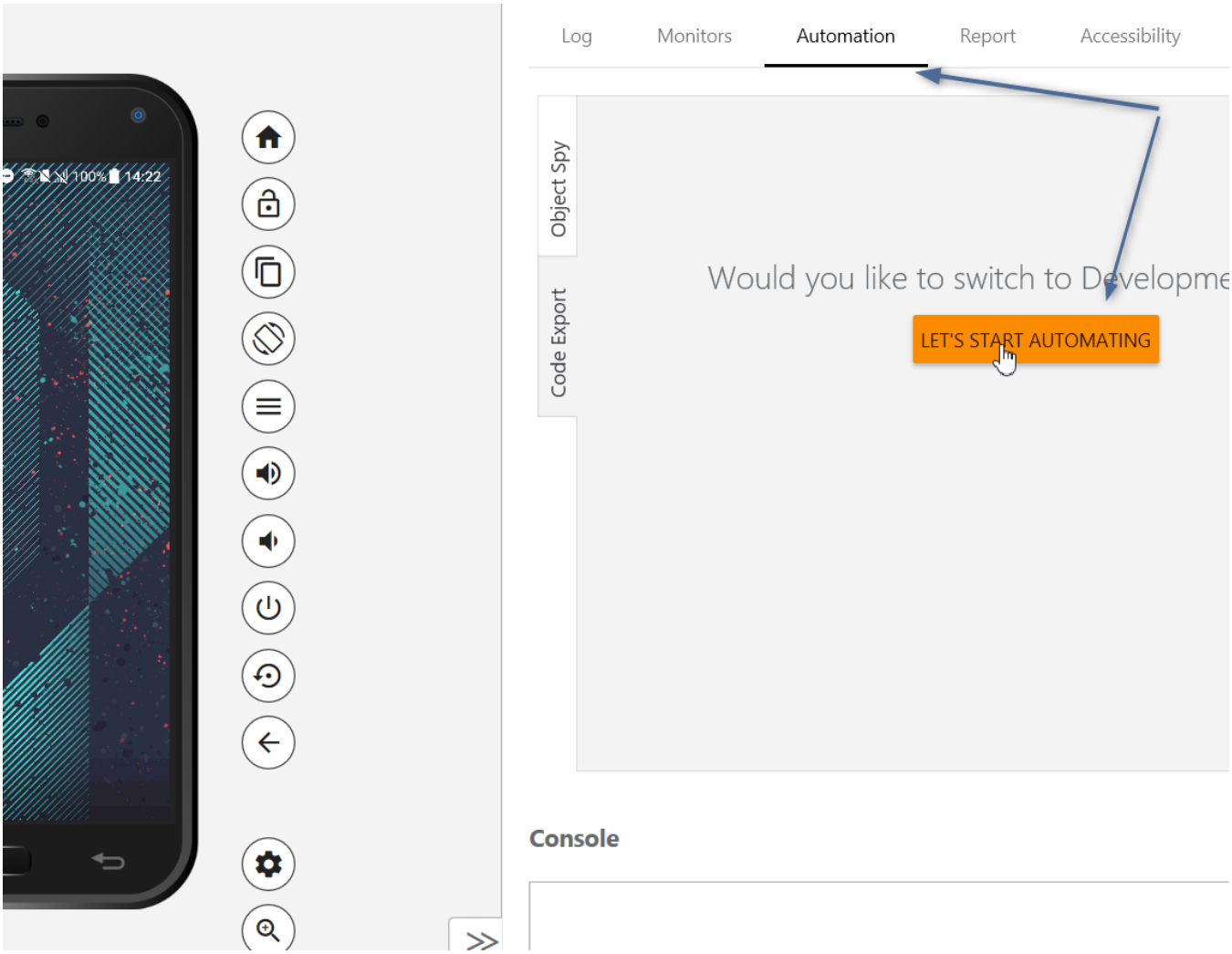


Once you press the **Open** button**,** a new page will get opened in a browser tab.

This page has two sections, left side of the page has the device and operations which can be performed on it.

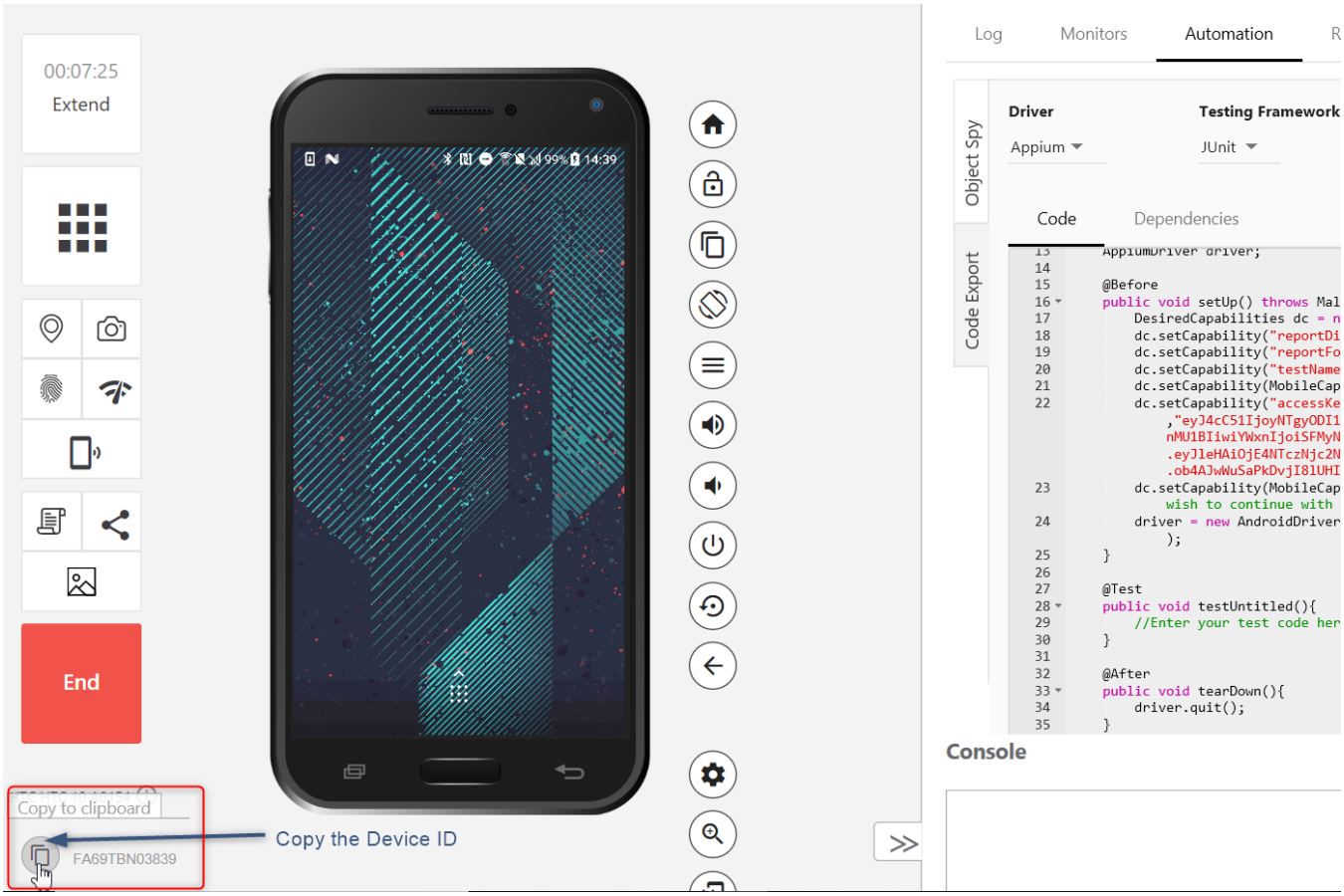
Section on right side displays useful utility operations like **viewing logs**, **automation** etc.

Now on the right side of screen, click **Automation tab** followed by "**LETS START AUTOMATING**" button.



Once you have completed step before, you will get a screen which looks like screen below.

Make sure you **copy the Device ID** from this page as shown in the screenshot below.



This concludes all the steps to **Reserve** and make device ready to run **Automation**.

Running the Automation Tests

Lets now move on the Automation code. We will now do a minor modification on the Sample Project in our git repository and run it again on the Reserved device.

- Fork the git repository.
- Open the project in InteliJ or Eclipse as Gradle project.
- Run the project. Sample Project is packaged with a Readme with all necessary instructions to run the project.

Once you have the project running, lets do a minor modification to use the reserved device.

Let us consider, the **platform to be Android** and the Device ID to be **FA69TBN03839** (copied previously from the last step of Reserve Device and making it ready for Automation)

Open **TestBase.java** and look for **initDefaultDesiredCapabilities** function and replace it with following function.

Testbase modification

```
/**
 * Initialize default properties.
 *
 */
protected void initDefaultDesiredCapabilities() {
    LOGGER.info("Setting up Desired Capabilities");
    String accessKey = System.getenv(ENV_VAR_ACCESS_KEY);

    if (accessKey == null || accessKey.length() < 10) {
        LOGGER.error("Access key must be set in Environment variable SEETEST_IO_ACCESS_KEY");
        LOGGER.info("To get access get to to https://cloud.seetest.io or learn at https://docs.seetest.io
/display/SEET/Execute+Tests+on+SeeTest+-+Obtaining+Access+Key", accessKey);
        throw new RuntimeException("Access key invalid : accessKey - " + accessKey);
    }

    dc.setCapability(SeeTestCapabilityType.ACCESS_KEY, accessKey);
    dc.setCapability(MobileCapabilityType.FULL_RESET, FULL_RESET);
    // *** Changed code
    if ("android".equals(os)) {
        dc.setCapability(MobileCapabilityType.UDID, "FA69TBN03839");
        LOGGER.info("Reserving device - FA69TBN03839");
    } else {
        String query = String.format("@os='%s'", os);
        dc.setCapability(SeeTestCapabilityType.DEVICE_QUERY, query);
        LOGGER.info("Device Query = {}", query);
    }
    // *** Changed code
    LOGGER.info("Desired Capabilities setup complete");
}
```

Modification was done to replace the device query with the capability **MobileCapabilityType.UDID** to set to specific device for Android platform.

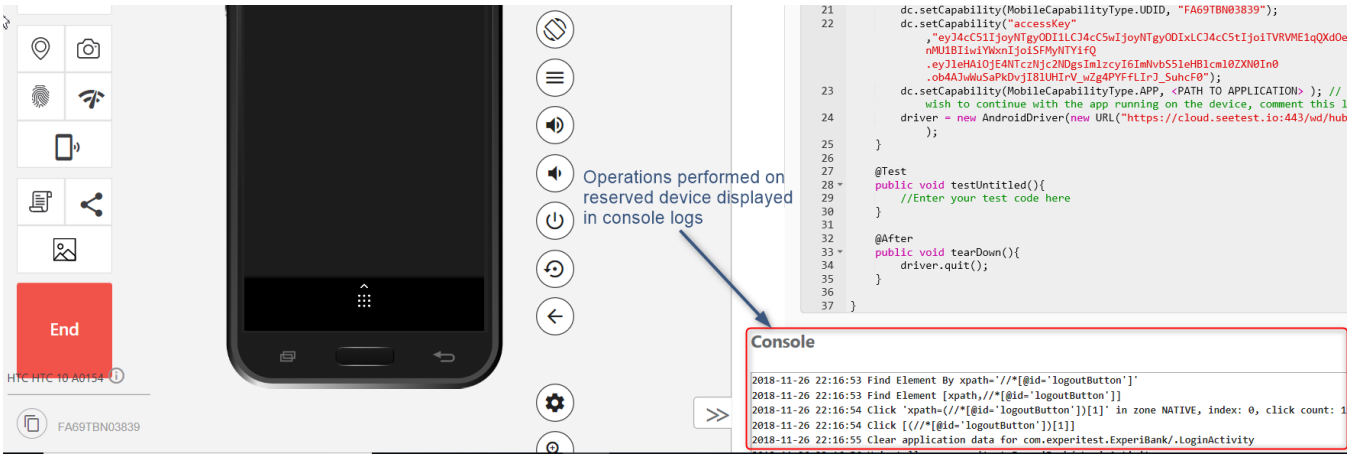
Modification in Test.java

```
String query = String.format("@os='%s'", os);
dc.setCapability(SeeTestCapabilityType.DEVICE_QUERY, query);
LOGGER.info("Device Query = {}", query);

with

if ("android".equals(os)) {
        dc.setCapability(MobileCapabilityType.UDID, "FA69TBN03839");
        LOGGER.info("Reserving device", "FA69TBN03839");
} else {
    String query = String.format("@os='%s'", os);
    dc.setCapability(SeeTestCapabilityType.DEVICE_QUERY, query);
    LOGGER.info("Device Query = {}", query);
}
```

Save and **Run** the test again.

You will now observe that, all the tests in **AndroidTestNGExampleTest.java** are run on the device you had earlier reserved.

If you observe the device which you had reserved in seetest.io you will be able to see the operations performed in the console log.



Note : Since we would like reserve and then run the test, setting **MobileCapabilityType.UDID** is important.

However, if you do not want to reserve the device before hand and would like to just choose the device while running the test, you could use device query as @os='<platform>' and @serialnumber='<deviceid>'.