# Install and start an application

This section explains how to upload, install, and launch an app when working with Appium/Selenium.
You can also instrument a hybrid Android app.

**On this page:**

## Upload the file to the Perfecto Lab repository media folder

Installing an application onto a Perfecto Lab device from within your automation script requires that the application file (for example, .apk or .ipa) be uploaded to the Perfecto Lab repository. You can upload the application externally to the automation script by using the Perfecto IDE or REST API, or from within an Appium/Selenium automation script using the ***uploadMedia()*** method of the **PerfectoLabUtils** class.

Perform ***uploadMedia()*** prior to creating the AppiumDriver instance.

**Uploading Media to Repository**

```
// Uploads local apk file to Media repository
PerfectoLabUtils.uploadMedia(cloudName, securityToken, localFilePath,
repositoryKey);
```

For this example in context, see https://github.com/PerfectoMobileSA/PerfectoJavaSample/blob/master/src/main/java/com/perfecto/sampleproject/PerfectoAppium.java

For an example of ***PerfectoLabUtils.java***, see https://github.com/PerfectoMobileSA/PerfectoJavaSample/blob/master/src/main/java/com/perfecto/sampleproject/PerfectoLabUtils.java.

## Install the file and start the app on the device

Because Appium is designed for mobile application testing, the Appium drivers can automatically install and activate the application as part of the driver initialization process**.** For alternative methods to instrument, install, and start the application, see below.
The AppiumDriver uses the "app" capability to identify the application file to install on the target device. Perfecto supports this functionality but requires that the location of the file be in the media section of the Repository:

- “PUBLIC:/” for files in the Public Media folder
- “PRIVATE:<folder>/” for files in the My Media folder

The AppiumDriver will then, normally, start the application on the device.

Other capabilities that affect the installation and launch of the application include:

- “fullReset” - indicates whether application should be uninstalled prior to installation. This guarantees that you are working with a clean version of the application.
- "noReset" - indicates whether the status of the application should be cleared prior to starting the app
- "autoLaunch" - indicates whether the application should automatically launched or not. Scripts that may need to set certain conditions prior to launching the app may use this to delay the application launch.

Also, set the capabilities for *appPackage* (Android) or *bundleID* (iOS). For Android, you can specify the app activity in addition to the package name.

**Installing and Launching the App**

```
//install and start application iOS with autoLaunch
capabilities.setCapability("autoLaunch",true);
capabilities.setCapability("fullReset",true);
capabilities.setCapability("app","PRIVATE:applications/Errands.ipa");
capabilities.setCapability("bundleId", "com.yoctoville.errands");

//start application Android
capabilities.setCapability("appPackage", "com.google.android.keep");
capabilities.setCapability("appActivity", ".activities.BrowseActivity");
```

If you chose to not automatically launch the app through the "autoLaunch" capability, use the AppiumDriver's ***launchApp()*** method to start the application that was specified by the *appPackage* or *bundleID* capability.

# Instrument the app

Perfecto supports the following types of instrumentation of applications installed on the devices:

- If you are checking a hybrid application running on an Android device, you should instrument the app as part of the driver creation process. Perfecto's Appium support includes a capability that will automatically instrument the application. Use the “autoInstrument” capability set to true.

  ```
  capabilities.setCapability("autoInstrument", true);
  ```

- If your application make use of the device sensors (for example the camera or fingerprint reader) as an input source, you can use Perfecto *data injection* commands to supply test input after performing sensor instrumentation. Use "sensorInstrument" capability set to true.

  ```
  capabilities.setCapability("sensorInstrument", true);
  ```

# Alternative procedures

If you are working with Selenium RWD or for some reason cannot use the Appium capabilities to install and start the app, use the following procedures.

## Install an application

Use the Perfecto proprietary command ***mobile:application:install*** to install an application that was previously uploaded to the Perfecto Repository.

**Install application**

```
//declare the Map for script parameters
Map<String, Object> params = new HashMap<>();

params.put("file", "PRIVATE:applications/Errands.ipa");
params.put("instrument", "noinstrument");
driver.executeScript("mobile:application:install", params);
```

See the Function Reference page for the Install application function for details of supplying parameters for application instrumentation using the parameters:

- **instrument** - for hybrid applicatons
- **sensorInstrument** - for sensor instrumentation.

## Start an application

Use the Perfecto proprietary command ***mobile:application:open*** to start an installed application.

**Install application**

```
//declare the Map for script parameters
Map<String, Object> params = new HashMap<>();

params.put("identifier", "com.google.android.apps.maps");
driver.executeScript("mobile:application:open", params);
```

## Instrument an Android application

In addition to instrumentation during installation, Perfecto supports an offline application to instrument Android applications. After instrumentation, the apk file can be installed on a Perfecto Lab Android device.