

Command line

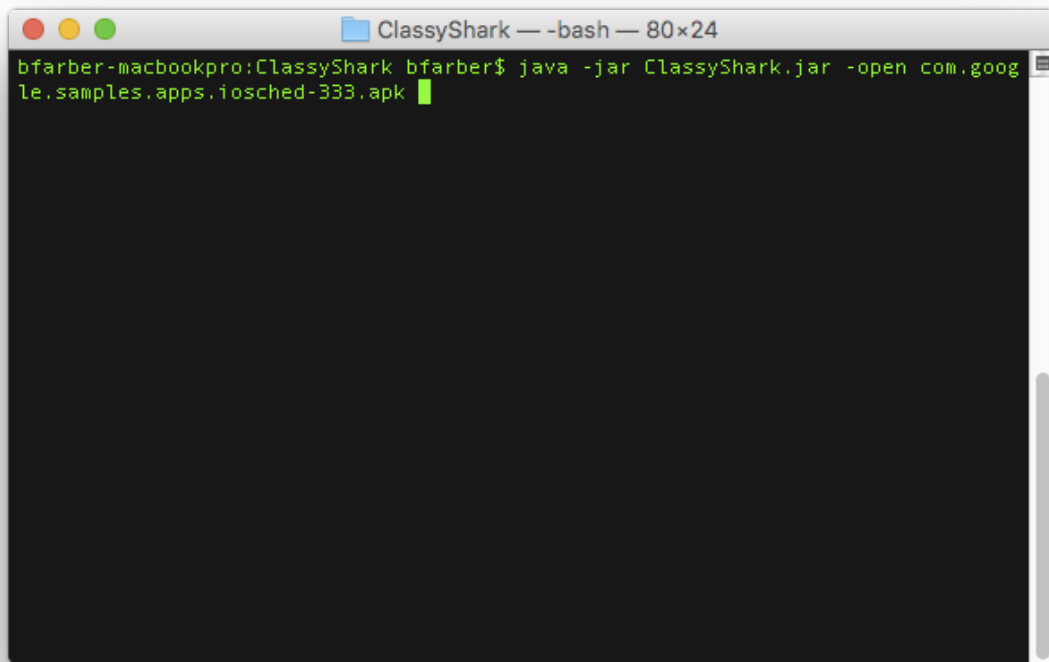
ver 1.0

Together with a stylish UI ClassyShark provides a rich API and command line functionality. In this part we are going to explore what ClassyShark can do for you from the command line. In the upcoming API section we see how ClassyShark provides APIs to every data in String form. One can use the export Java APIs as part of one's build and continuous integration pipeline. Here are the command line services.

- [1. Open your APK in ClassyShark](#)
- [2. Open your APK in ClassyShark and show a specific class in GUI](#)
- [3. Export APK](#)
- [4. Export a specific class from APK](#)
- [5. Inspect APK\(experimental\)](#)

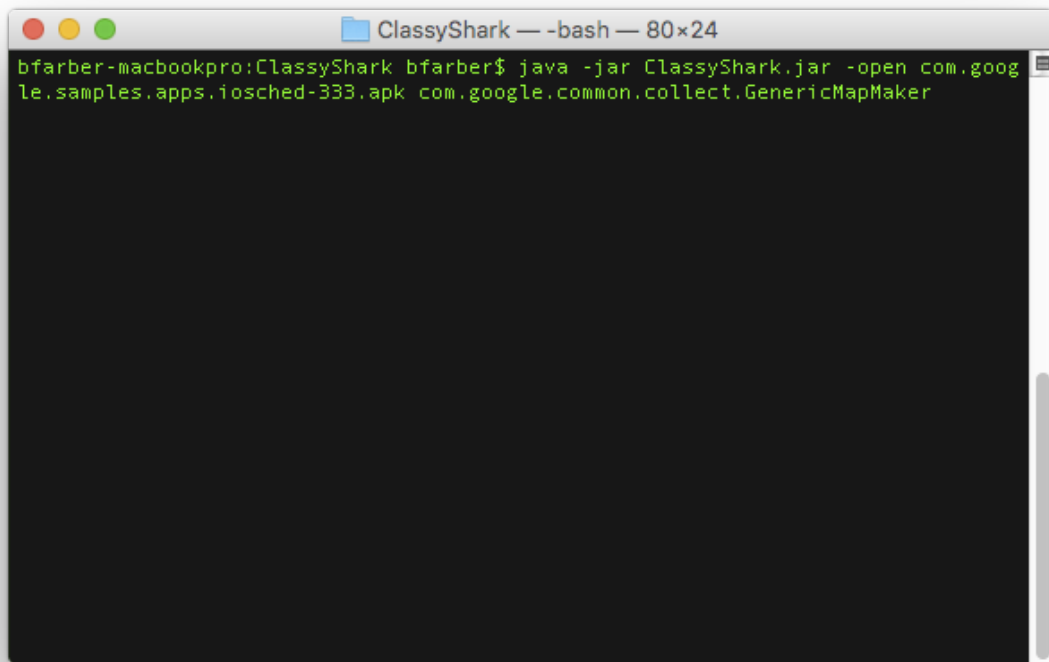
1. Open your APK in ClassyShark

```
java -jar ClassyShark.jar -open <YOUR_APK.apk>
```



2. Open your APK in ClassyShark and show a specific class in GUI

java -jar ClassyShark.jar -open <BINARY_FILE> <FULLY_QUALIFIED_CLASS_NAME>

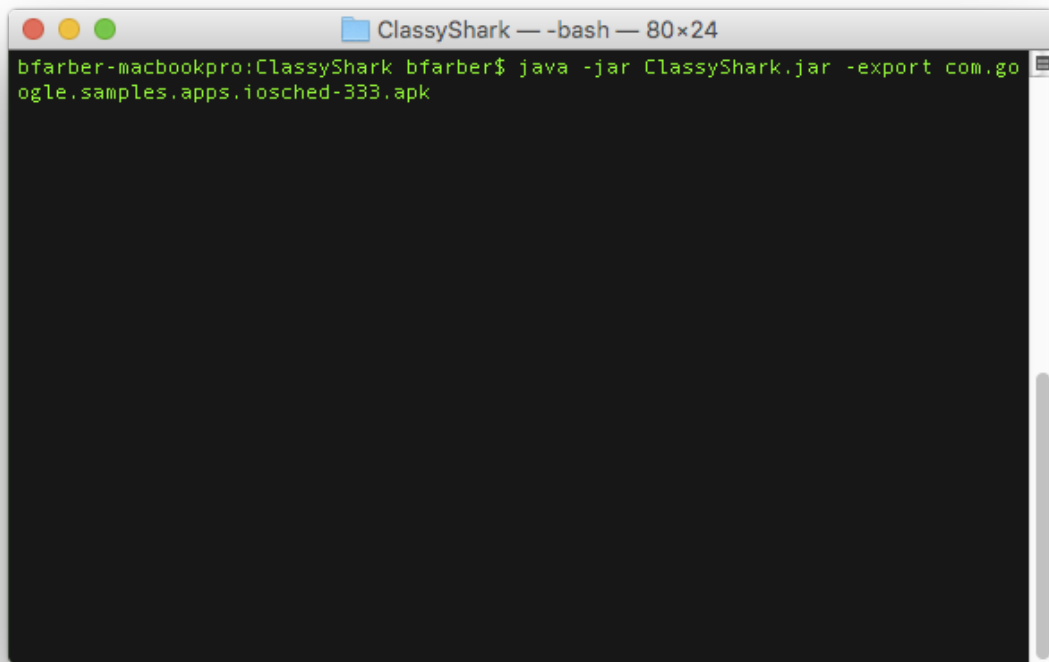
A screenshot of a macOS terminal window titled "ClassyShark — -bash — 80x24". The terminal shows a command being executed: `bfarber-macbookpro:ClassyShark bfarber$ java -jar ClassyShark.jar -open com.google.samples.apps.iosched-333.apk com.google.common.collect.GenericMapMaker`. The command is split across two lines. The terminal background is black, and the text is green. The window has standard macOS window controls (red, yellow, green buttons) at the top left.

```
bfarber-macbookpro:ClassyShark bfarber$ java -jar ClassyShark.jar -open com.google.samples.apps.iosched-333.apk com.google.common.collect.GenericMapMaker
```

Opens the ClassyShark GUI with the specific class showing in the UI plane.

3. Export APK

```
java -jar ClassyShark.jar -export <BINARY_FILE>
```

A screenshot of a macOS terminal window titled "ClassyShark — -bash — 80x24". The terminal shows the command `java -jar ClassyShark.jar -export com.google.samples.apps.iosched-333.apk` being executed. The background of the terminal is black, and the text is green. The window has standard macOS window controls (red, yellow, green buttons) in the top-left corner.

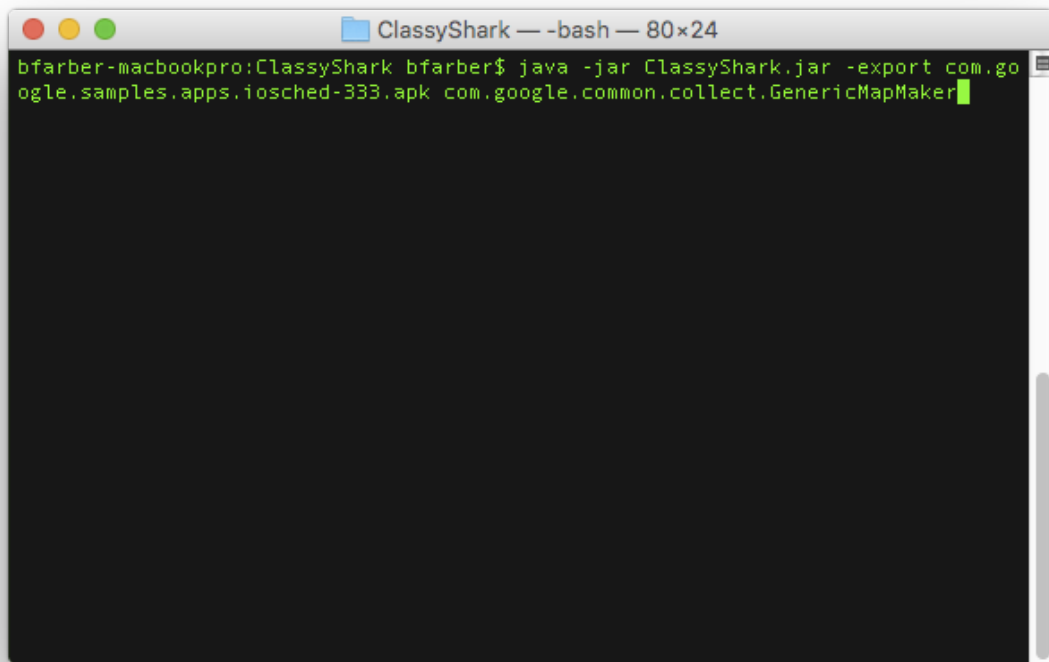
```
ClassyShark — -bash — 80x24
bfarber-macbookpro:ClassyShark bfarber$ java -jar ClassyShark.jar -export com.google.samples.apps.iosched-333.apk
```

ClassyShark will analyze the APK and dump out the following files, in a grep friendly fashion

- all_classes.txt - list of all classes
- all_methods.txt - all the method names and signatures from all the dexes
- all_strings.txt - all the string tables from all the dexes
- all_strings.txt - all the string tables from all the dexes
- AndroidManifest.xml_dump - goes without saying

4. Export a specific class from APK

```
java -jar ClassyShark.jar -export <BINARY_FILE> <FULLY_QUALIFIED_CLASS_NAME>
```

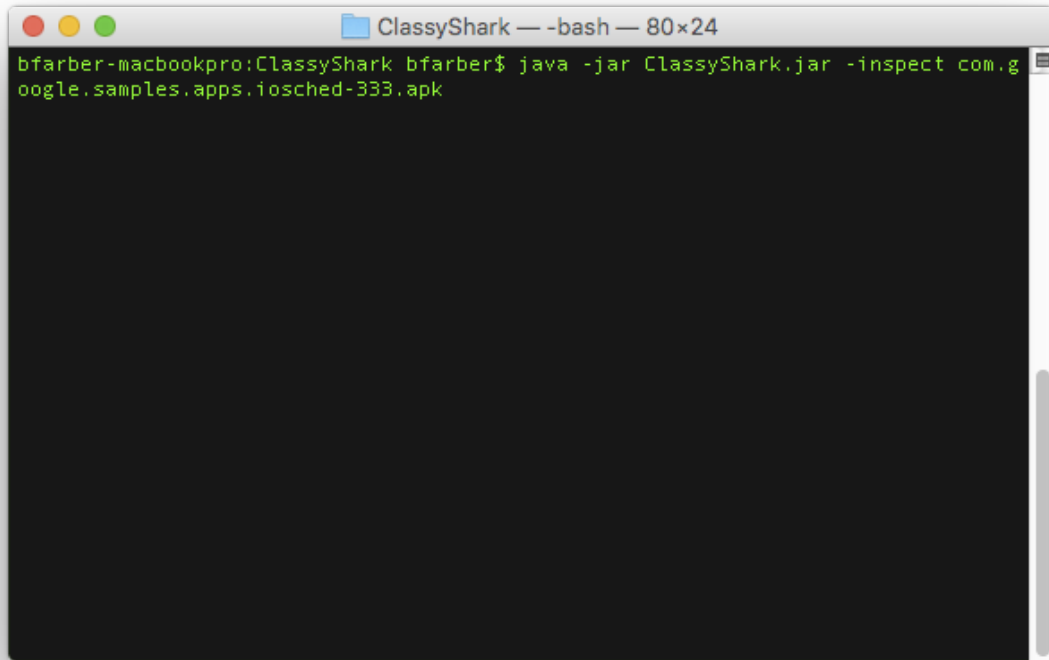
A screenshot of a macOS terminal window titled "ClassyShark — -bash — 80x24". The terminal shows a command being executed: `bfarber-macbookpro:ClassyShark bfarber$ java -jar ClassyShark.jar -export com.google.samples.apps.iosched-333.apk com.google.common.collect.GenericMapMaker`. The command is split across two lines. The terminal has a dark background and a light-colored cursor at the end of the second line.

```
bfarber-macbookpro:ClassyShark bfarber$ java -jar ClassyShark.jar -export com.google.samples.apps.iosched-333.apk com.google.common.collect.GenericMapMaker
```

ClassyShark generates and dumps a text file, which is a human readable representation of the class, passed as parameter.

5. Inspect APK(experimental)

java -jar ClassyShark.jar -inspect <YOUR_APK.apk>



Dumps out to command prompt all the classes that have test as their class names, native and abstract methods