Zipkin and Sleuth implementation:

Source Code link: https://github.com/vishakhaakumar/SQA-ProjectCode_Zipkin_Sleuth.git

Steps to reproduce:

1. Clone the code from the GitHub above link. (All the code changes and dependencies needed to implement Zipkin and Sleuth is already included in the code)
2. Run each of the microservices individually using
    a. an IDE such as IntelliJ – right click on the main file and run.
    b. or through a command prompt – mvn compile and mvn run.
3. Call the microservices from browser or postman by sending valid URL and body.
   Eg: http://localhost:4444/orders/

4. Open the Zipkin dashboard on the browser on the link: http://localhost:9411/zipkin/
5. Open the IDE console to see the automatic logging functionality of sleuth that produces the tracing ID and span ID.

SonarLint and SonarQube implementation:

Source Code link:
https://github.com/vishakhaakumar/SQA-ProjectCode_Zipkin_Sleuth.git
https://github.com/spring-projects/spring-petclinic.git

Steps to reproduce:

1. Clone the code from the GitHub above link.
2. Add the SonarLint plugin to the IDE:
    a. File -> Settings -> Plugins -> search for SonarLint and install it.
3. Analyze the code using SonarLint:
    a. Right click on code file -> analyze with SonarLint.
    b. This shows all the programming errors and issues in the file that can be corrected by the developer.
4. Install Docker for desktop.
5. Install SonarQube by pulling the docker image: 'sonarqube:latest'
   ```
   docker run -d -p 9000:9000 sonarqube:latest
   ```
6. Test the sonarQube dashboard on http://localhost:9000/
7. Setup the login credentials and generate the token for authentication.
8. Write a sonar_project.properties file with config details for the project.
9. Scan the code on SonarQube using the below command:
   ```
   mvn clean verify sonar:sonar -Dsonar.projectKey=pet-project -
   Dsonar.host.url=http://localhost:9000 -Dsonar.login="***"
   ```

10. Verify the scan results, bugs, code coverage, code smells, security vulnerabilities etc on the dashboard.

TestNG - Integrate JUnit with TestNG :

Source Code Link: https://github.com/ssoad/BankingSystem/tree/master/src

Steps to reproduce:

1. Download the code from the GitHub link above.
2. Import it to the Eclipse IDE.
3. Install the TestNG from the Eclipse IDE and add it to the build path.
4. Add the test class folder from the file which is provided.
5. Create a XML file with the suite tag and junit=true as shown below in the project folder.

```
<?xml version="1.0" encoding="UTF-8"?>

<!DOCTYPE suite SYSTEM "https://testng.org/testng-1.0.dtd">

<suite name="Suite">

  <test thread-count="5" name="Test" junit="true">

    <classes>

      <class name="tests.ExceptionsTest"/>

      <class name="tests.StudentAccountTest"/>

      <class name="tests.SavingsAccountTest"/>

      <class name="tests.BankAccountTest"/>

      <class name="tests.FileOTest"/>

      <class name="tests.BankTest"/>

    </classes>

  </test> <!-- Test -->

</suite> <!-- Suite -->
```

6. Run the xml file which you created with the above command as TestNG.
7. The result is shown in the console.
8. Refresh the project. Upon refresh a test-output folder gets created with index.html, testing-results.xml, testing-failed.xml and etc.

9. Open the index.html file in the web browser. We can see the detailed explanation about the result.

DesigniteJava - Installation and Implementation:

Project : https://github.com/hkristanto/JHotDraw

Steps to Reproduce:

1. Install the DesigniteJava using the below link(Try to get the licenced version it is free for student)

   https://www.designite-tools.com/designitejava/

2. A DesigniteJava.jar file will be downloaded.
3. Download the project from the above link.
4. Create an empty output or result folder where the results are generated after the execution .
5. Keep the project, Designitejava.jar file and the empty output folder in one one folder.
6. Open the command prompt from the above folder and enter the following syntax.

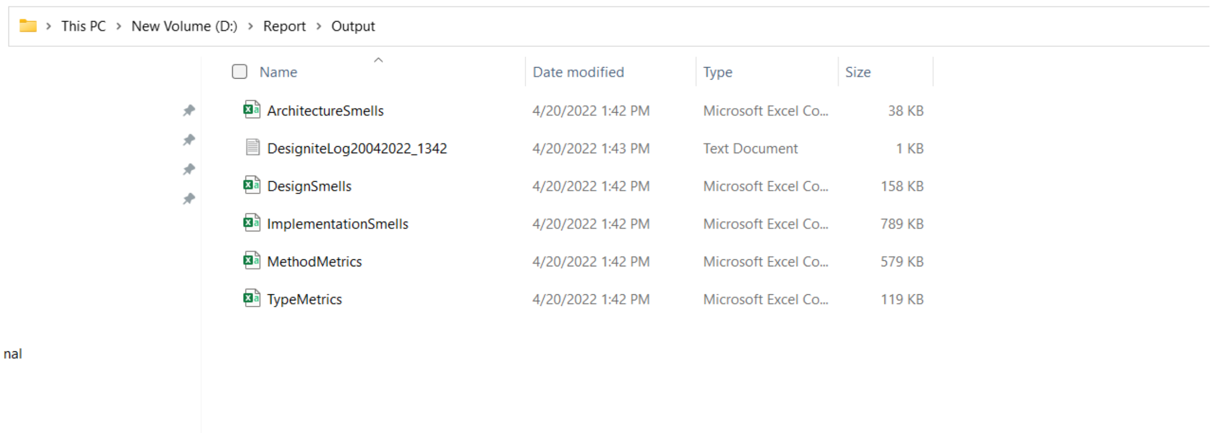
   ```
   java -jar DesigniteJava.jar -i <path of the input source folder> -o <path of the output folder>
   ```

   ```
   Ex : java -jar DesigniteJava.jar -i D:\Report\JHotDraw -o D:\Report\Output
   ```

   Path of the input source folder - The folder link of the project(In our case D:\Report\JHotDraw - Just open the project folder and copy the link)

   Path of the output folder is - The link of the output folder which you created.(D:\Report\Output - Open the output folder and copy the link - It should be empty)

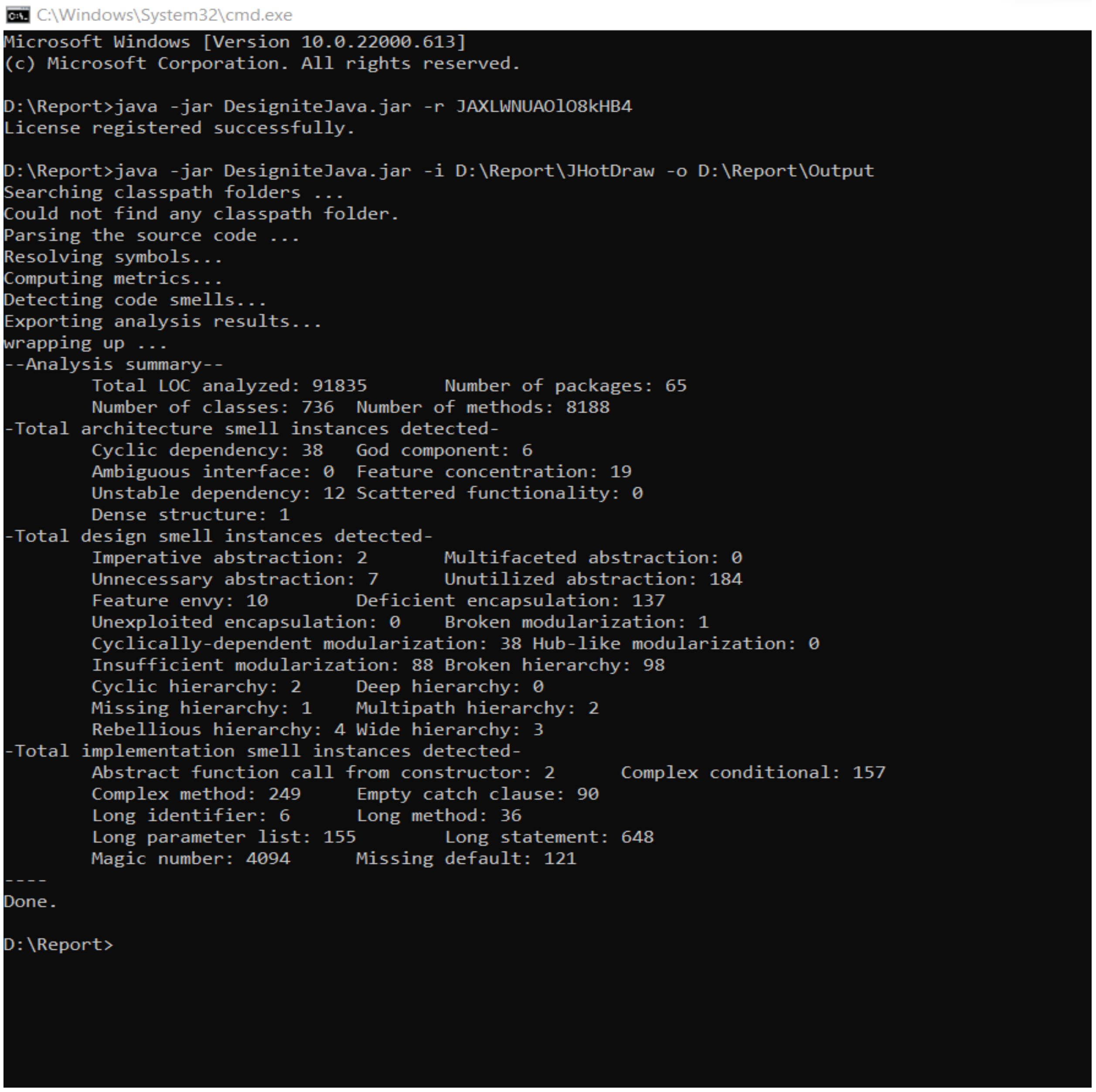
7. The output will be shown as below
8. The detailed result reports will be generated in the Output folder as shown below. We can analyze the different code smells from those files.



```
This PC > New Volume (D:) > Report > Output
```

Name	Date modified	Type	Size
ArchitectureSmells	4/20/2022 1:42 PM	Microsoft Excel Co...	38 KB
DesigniteLog20042022_1342	4/20/2022 1:43 PM	Text Document	1 KB
DesignSmells	4/20/2022 1:42 PM	Microsoft Excel Co...	158 KB
ImplementationSmells	4/20/2022 1:42 PM	Microsoft Excel Co...	789 KB
MethodMetrics	4/20/2022 1:42 PM	Microsoft Excel Co...	579 KB
TypeMetrics	4/20/2022 1:42 PM	Microsoft Excel Co...	119 KB

nal



C:\Windows\System32\cmd.exe

```
Microsoft Windows [Version 10.0.22000.613]
(c) Microsoft Corporation. All rights reserved.

D:\Report>java -jar DesigniteJava.jar -r JAXLWNUAO1O8kHB4
License registered successfully.

D:\Report>java -jar DesigniteJava.jar -i D:\Report\JHotDraw -o D:\Report\Output
Searching classpath folders ...
Could not find any classpath folder.
Parsing the source code ...
Resolving symbols...
Computing metrics...
Detecting code smells...
Exporting analysis results...
wrapping up ...
--Analysis summary--
        Total LOC analyzed: 91835       Number of packages: 65
        Number of classes: 736  Number of methods: 8188
-Total architecture smell instances detected-
        Cyclic dependency: 38   God component: 6
        Ambiguous interface: 0  Feature concentration: 19
        Unstable dependency: 12 Scattered functionality: 0
        Dense structure: 1
-Total design smell instances detected-
        Imperative abstraction: 2       Multifaceted abstraction: 0
        Unnecessary abstraction: 7      Unutilized abstraction: 184
        Feature envy: 10        Deficient encapsulation: 137
        Unexploited encapsulation: 0    Broken modularization: 1
        Cyclically-dependent modularization: 38 Hub-like modularization: 0
        Insufficient modularization: 88 Broken hierarchy: 98
        Cyclic hierarchy: 2     Deep hierarchy: 0
        Missing hierarchy: 1    Multipath hierarchy: 2
        Rebellious hierarchy: 4 Wide hierarchy: 3
-Total implementation smell instances detected-
        Abstract function call from constructor: 2      Complex conditional: 157
        Complex method: 249     Empty catch clause: 90
        Long identifier: 6      Long method: 36
        Long parameter list: 155        Long statement: 648
        Magic number: 4094      Missing default: 121
----
Done.

D:\Report>
```

DbFit - Implementation :

Steps to be followed:

1. Download the dataset using this link https://www.kaggle.com/datasets/rio2016/olympic-games
2. Download Mysql database
3. After installing the mysql database, go to the c drive, look for program files, open java and go to the bin folder then open the command prompt from the bin folder.
4. Try connecting to the mysql database with the below command

   mysql - u -root -p

   Enter the password for the root user

5. Create a database using the command - Create Database SQLDatabase;
6. Type Use SQLDATABASE;
7. Create the tables using below commands.

Database Schema:

1) Athletes :

CREATE TABLE athletes (id integer, name VARCHAR(50), nationality VARCHAR(50),sex VARCHAR(50),dob date,height double precision,weight double precision,sport VARCHAR(50),gold integer,silver integer,bronze integer,

PRIMARY KEY (id));

2) Events:

```
CREATE TABLE events (
 id integer,
 sport VARCHAR,
 discipline VARCHAR,
 name VARCHAR,
 venues VARCHAR
);
```

DML Statements:

```
COPY athletes(id, name, nationality, sex, dob, height, weight, sport, gold, silver, bronze)
FROM 'C:\Users\ykulk\Desktop\Database\athletes.csv'
DELIMITER ','
CSV HEADER;
```

```
select * from athletes;
```

```
COPY events(id, sport, discipline, name, venues)
FROM 'C:\Users\ykulk\Desktop\Database\events.csv'
DELIMITER ','
CSV HEADER;
```

```
select * from events;
```

```
COPY countries(country, code, population, gdp_per_capita)
FROM 'C:\Users\ykulk\Desktop\Database\countries.csv'
DELIMITER ','
CSV HEADER;
```

```
select * from countries;
```

8. To run DbFit you need Java Runtime Environment (JRE) 7 or higher version.
9. Download the DbFit from this link http://dbfit.github.io/dbfit/
10. Unzip the file and run the startFitnesse.bat file in the command line.
11. Wait for the commands to complete and keep the server running.
12. Open the http://localhost:8085/ in the web browser
13. Open http://localhost:8085/HelloWorldTest in your browser. You should see an editor where you can create and run your test page.
14. !|dbfit.MySqlTest|

    !|Connect|localhost|dbfit_user|password|dbfit| - This Connects to the mysql.

15. !path lib/*.jar - Add this line below the above commands - In order to load the DbFit extension into FitNesse, your test pages have to load the correct libraries by including the following command:
16. !|Query|select* from Athletes|

    |id |name|

|1|Test |

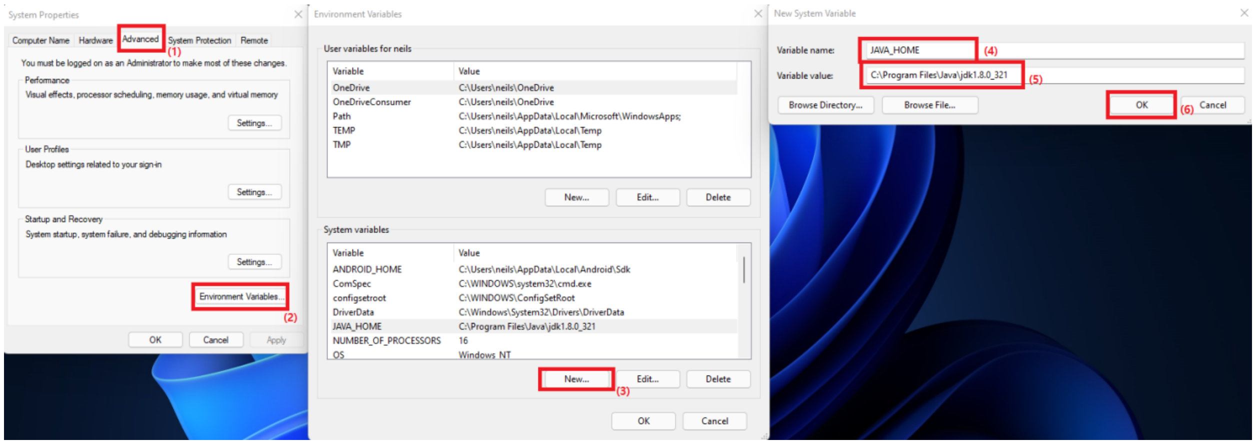
17. Save the page and click on the Test button.

UI Test Setup

In this part of the document, the setup process for UI testing will be explained.
This process was verified on a Windows 11 environment.

1. Install Java SDK 1.8.0_321 version

1. Download Java SDK 1.8.0_321 version for Windows from the link below:
   https://www.oracle.com/java/technologies/downloads/#java8-windows
2. Install Java SDK 1.8.0_321 version
3. Set JAVA_HOME and Path environment variable:
   Windows Settings > System > About > Advanced system settings > Environment variables (2)
   > 1. Add 'JAVA_HOME' system variable
     (New (3)
     → Variable name: "JAVA_HOME" (4)
     → Variable value: Java SDK installed location (5)
       (e.g. "C:\Program Files\Java\jdk1.8.0_321")
     → Press OK button (6))



You must be logged on as an Administrator to make most of these changes.

Visual effects, processor scheduling, memory usage, and virtual memory

Desktop settings related to your sign-in

System startup, system failure, and debugging information

User variables for neils

Variable	Value
OneDrive	C:\Users\neils\OneDrive
OneDriveConsumer	C:\Users\neils\OneDrive
Path	C:\Users\neils\AppData\Local\Microsoft\WindowsApps;
TEMP	C:\Users\neils\AppData\Local\Temp
TMP	C:\Users\neils\AppData\Local\Temp

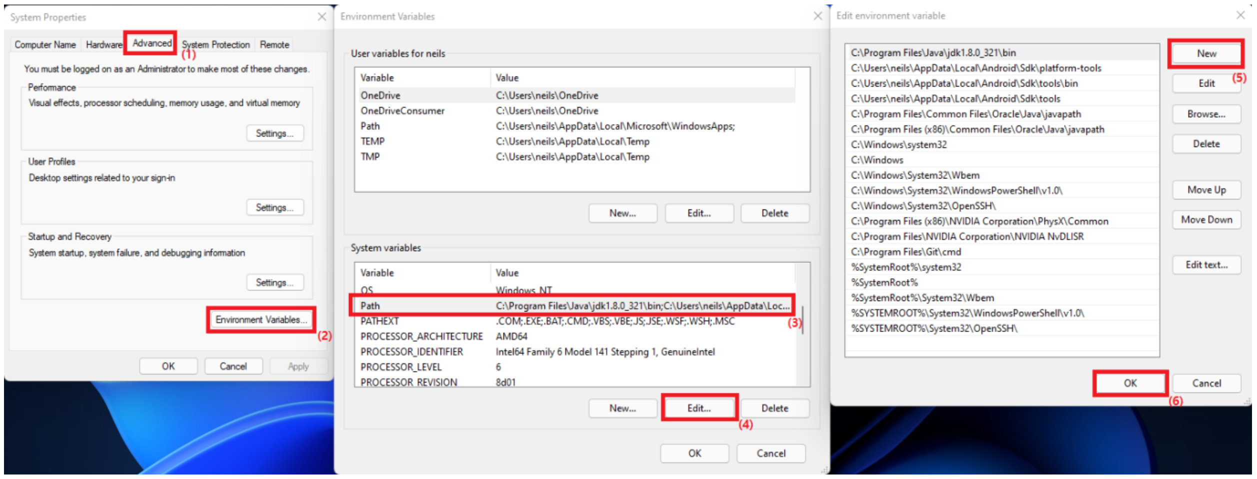
System variables

Variable	Value
ANDROID_HOME	C:\Users\neils\AppData\Local\Android\Sdk
ComSpec	C:\WINDOWS\system32\cmd.exe
configsetroot	C:\WINDOWS\ConfigSetRoot
DriverData	C:\Windows\System32\Drivers\DriverData
JAVA_HOME	C:\Program Files\Java\jdk1.8.0_321
NUMBER_OF_PROCESSORS	16
OS	Windows_NT

Variable name: JAVA_HOME

Variable value: C:\Program Files\Java\jdk1.8.0_321

   > 2. Add "%JAVA_HOME%\bin" folder to 'Path' system variable
     (Click 'Path' in the system variable list (3)
     → Press Edit button (4)
     → Press New button (5)
     → Type 'bin' folder location in the Java SDK installation folder
       (e.g. "C:\Program Files\Java\jdk1.8.0_321\bin")
     → Press OK button (6))



You must be logged on as an Administrator to make most of these changes.

Visual effects, processor scheduling, memory usage, and virtual memory

Desktop settings related to your sign-in

System startup, system failure, and debugging information

User variables for neils

Variable	Value
OneDrive	C:\Users\neils\OneDrive
OneDriveConsumer	C:\Users\neils\OneDrive
Path	C:\Users\neils\AppData\Local\Microsoft\WindowsApps;
TEMP	C:\Users\neils\AppData\Local\Temp
TMP	C:\Users\neils\AppData\Local\Temp

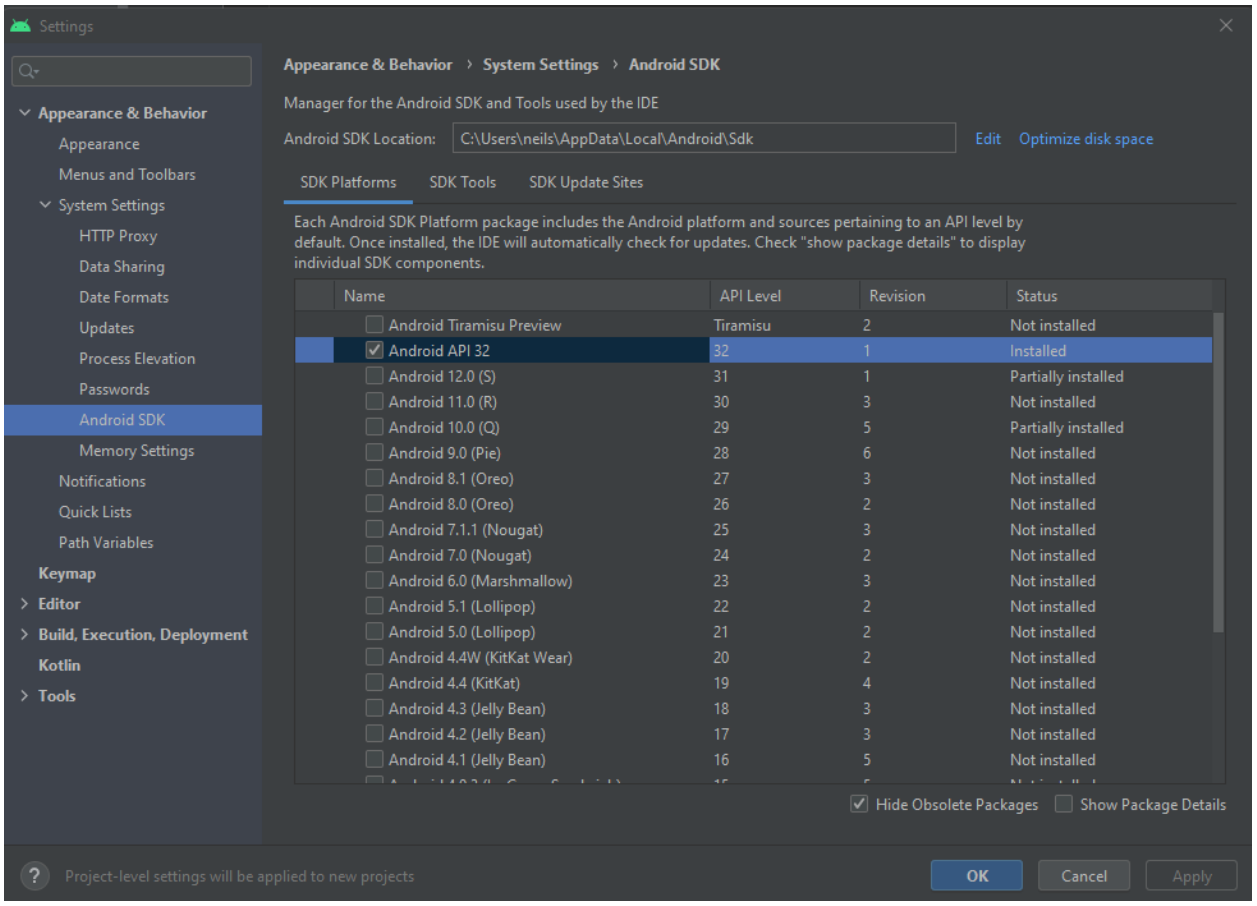
System variables

Variable	Value
OS	Windows_NT
Path	C:\Program Files\Java\jdk1.8.0_321\bin;C:\Users\neils\AppData\Loc...
PATHEXT	.COM;.EXE;.BAT;.CMD;.VBS;.VBE;.JS;.JSE;.WSF;.WSH;.MSC
PROCESSOR_ARCHITECTURE	AMD64
PROCESSOR_IDENTIFIER	Intel64 Family 6 Model 141 Stepping 1, GenuineIntel
PROCESSOR_LEVEL	6
PROCESSOR_REVISION	8d01

```
C:\Program Files\Java\jdk1.8.0_321\bin
C:\Users\neils\AppData\Local\Android\Sdk\platform-tools
C:\Users\neils\AppData\Local\Android\Sdk\tools\bin
C:\Users\neils\AppData\Local\Android\Sdk\tools
C:\Program Files\Common Files\Oracle\Java\javapath
C:\Program Files (x86)\Common Files\Oracle\Java\javapath
C:\Windows\system32
C:\Windows
C:\Windows\System32\Wbem
C:\Windows\System32\WindowsPowerShell\v1.0\
C:\Windows\System32\OpenSSH\
C:\Program Files (x86)\NVIDIA Corporation\PhysX\Common
C:\Program Files\NVIDIA Corporation\NVIDIA NvDLISR
C:\Program Files\Git\cmd
%SystemRoot%\system32
%SystemRoot%
%SystemRoot%\System32\Wbem
%SYSTEMROOT%\System32\WindowsPowerShell\v1.0\
%SYSTEMROOT%\System32\OpenSSH\
```

2. Install Android Studio, Android SDK, Android Virtual Device

1. Download Android Studio from the link below:
   https://developer.android.com/studio
2. Install Android Studio
3. Create any empty project to start Android Studio
4. Open Android SDK manager (Settings > Appearance & Behavior > System Settings > Android SDK)
5. Select "Android API 32" and press the OK button to install it.



Settings

Appearance & Behavior › System Settings › Android SDK

Manager for the Android SDK and Tools used by the IDE

Android SDK Location: C:\Users\neils\AppData\Local\Android\Sdk

SDK Platforms | SDK Tools | SDK Update Sites

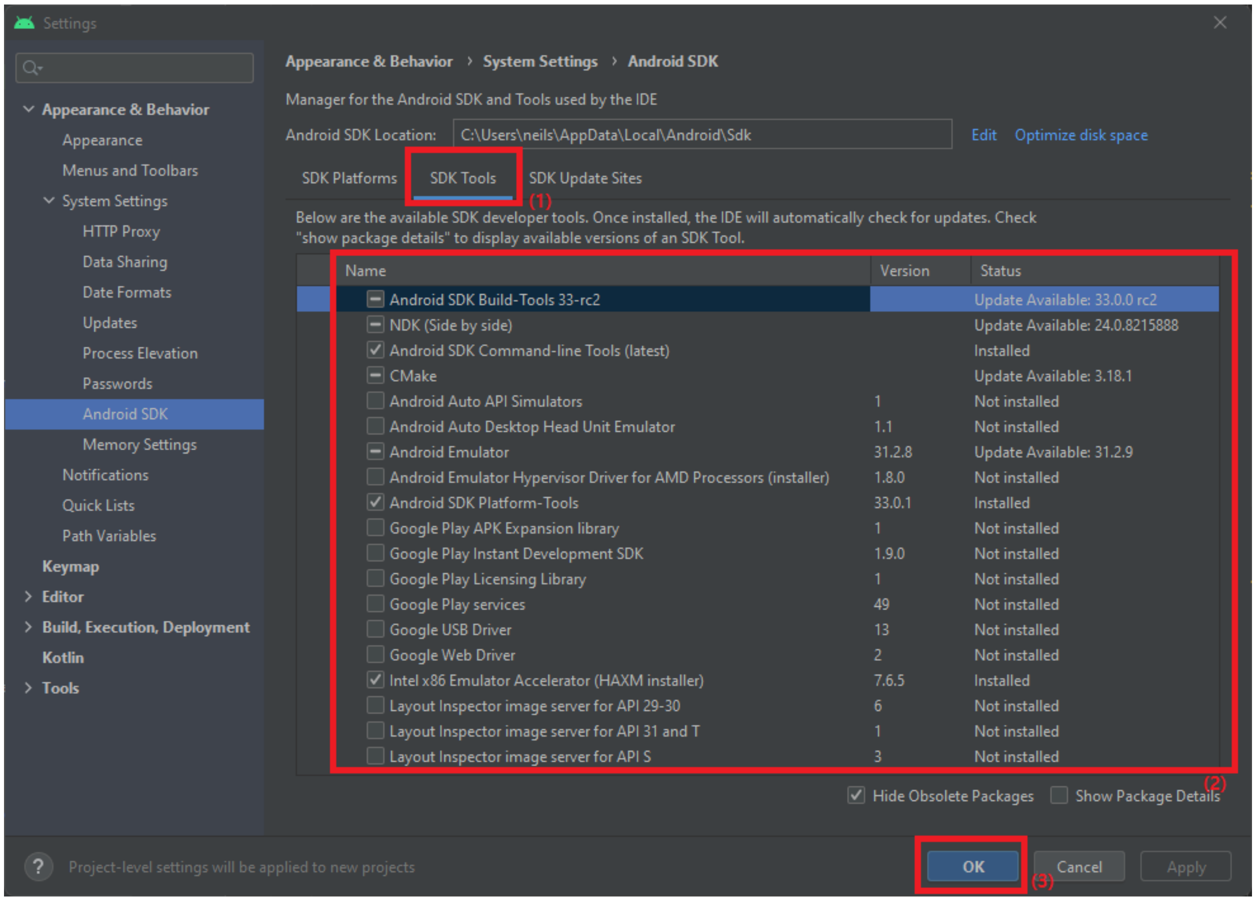
Each Android SDK Platform package includes the Android platform and sources pertaining to an API level by default. Once installed, the IDE will automatically check for updates. Check "show package details" to display individual SDK components.

Name	API Level	Revision	Status
☐ Android Tiramisu Preview	Tiramisu	2	Not installed
✓ Android API 32	32	1	Installed
☐ Android 12.0 (S)	31	1	Partially installed
☐ Android 11.0 (R)	30	3	Not installed
☐ Android 10.0 (Q)	29	5	Partially installed
☐ Android 9.0 (Pie)	28	6	Not installed
☐ Android 8.1 (Oreo)	27	3	Not installed
☐ Android 8.0 (Oreo)	26	2	Not installed
☐ Android 7.1.1 (Nougat)	25	3	Not installed
☐ Android 7.0 (Nougat)	24	2	Not installed
☐ Android 6.0 (Marshmallow)	23	3	Not installed
☐ Android 5.1 (Lollipop)	22	2	Not installed
☐ Android 5.0 (Lollipop)	21	2	Not installed
☐ Android 4.4W (KitKat Wear)	20	2	Not installed
☐ Android 4.4 (KitKat)	19	4	Not installed
☐ Android 4.3 (Jelly Bean)	18	3	Not installed
☐ Android 4.2 (Jelly Bean)	17	3	Not installed
☐ Android 4.1 (Jelly Bean)	16	5	Not installed

✓ Hide Obsolete Packages ☐ Show Package Details

Project-level settings will be applied to new projects

6. Open "SDK tools' tab and install the following tools:
   - Android SDK Build-Tools
   - Android SDK Platform-Tools
   - Android SDK Command-line Tools
   - Intel x86 Emulator Accelerator (HAXM installer)
   - Android Emulator
   - CMake



Manager for the Android SDK and Tools used by the IDE

Android SDK Location: C:\Users\neils\AppData\Local\Android\Sdk

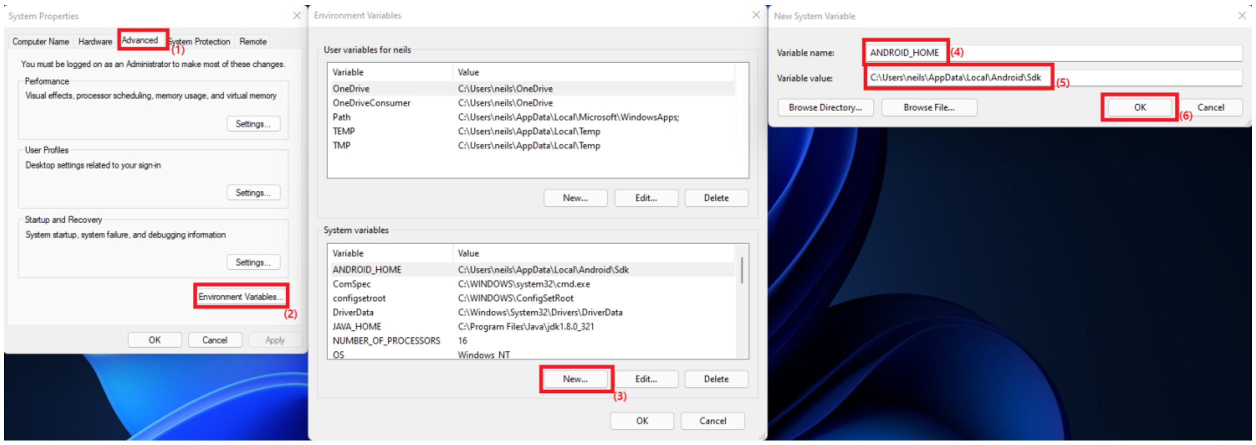
Below are the available SDK developer tools. Once installed, the IDE will automatically check for updates. Check "show package details" to display available versions of an SDK Tool.

	Name	Version	Status
−	Android SDK Build-Tools 33-rc2		Update Available: 33.0.0 rc2
−	NDK (Side by side)		Update Available: 24.0.8215888
✓	Android SDK Command-line Tools (latest)		Installed
−	CMake		Update Available: 3.18.1
☐	Android Auto API Simulators	1	Not installed
☐	Android Auto Desktop Head Unit Emulator	1.1	Not installed
−	Android Emulator	31.2.8	Update Available: 31.2.9
☐	Android Emulator Hypervisor Driver for AMD Processors (installer)	1.8.0	Not installed
✓	Android SDK Platform-Tools	33.0.1	Installed
☐	Google Play APK Expansion library	1	Not installed
☐	Google Play Instant Development SDK	1.9.0	Not installed
☐	Google Play Licensing Library	1	Not installed
☐	Google Play services	49	Not installed
☐	Google USB Driver	13	Not installed
☐	Google Web Driver	2	Not installed
✓	Intel x86 Emulator Accelerator (HAXM installer)	7.6.5	Installed
☐	Layout Inspector image server for API 29-30	6	Not installed
☐	Layout Inspector image server for API 31 and T	1	Not installed
☐	Layout Inspector image server for API S	3	Not installed

[✓] Hide Obsolete Packages ☐ Show Package Details

Project-level settings will be applied to new projects

7. Set ANDROID_HOME and Path environment variable:
   Windows Settings > System > About > Advanced system settings > Environment variables (2)
   > 1. Add ‘ANDRIOD_HOME’ system variable
      (New (3)
      → Variable name: “ANDROID_HOME” (4)
      → Variable value: Android SDK installed location (5)
         (e.g. “C:\Users\neils\AppData\Local\Android\Sdk”)
      → Press OK button (6))



You must be logged on as an Administrator to make most of these changes.

Performance
Visual effects, processor scheduling, memory usage, and virtual memory

User Profiles
Desktop settings related to your sign-in

Startup and Recovery
System startup, system failure, and debugging information

User variables for neils

Variable	Value
OneDrive	C:\Users\neils\OneDrive
OneDriveConsumer	C:\Users\neils\OneDrive
Path	C:\Users\neils\AppData\Local\Microsoft\WindowsApps;
TEMP	C:\Users\neils\AppData\Local\Temp
TMP	C:\Users\neils\AppData\Local\Temp

System variables

Variable	Value
ANDROID_HOME	C:\Users\neils\AppData\Local\Android\Sdk
ComSpec	C:\WINDOWS\system32\cmd.exe
configsetroot	C:\WINDOWS\ConfigSetRoot
DriverData	C:\Windows\System32\Drivers\DriverData
JAVA_HOME	C:\Program Files\Java\jdk1.8.0_321
NUMBER_OF_PROCESSORS	16
OS	Windows_NT

Variable name: ANDROID_HOME

Variable value: C:\Users\neils\AppData\Local\Android\Sdk

> 2. Add “platform-tools”, “tools\bin”, and “tools” folder in Android SDK folder to ‘Path’ system variable

(Click ‘Path’ in the system variable list (3)

→ Press Edit button (4)

→ Press New button (5)

→ Enter ‘platform-tools’ folder location in the Java SDK installation folder

(e.g. “C:\Users\neils\AppData\Local\Android\Sdk\platform-tools”)

→ Press New button (5)

→ Enter ‘tools\bin’ folder location in the Java SDK installation folder

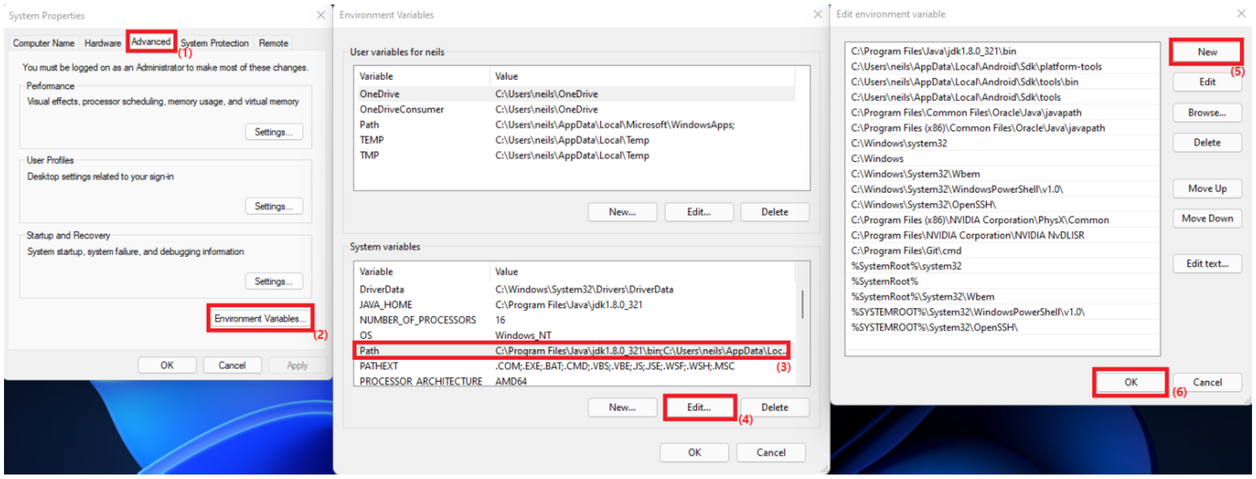
(e.g. “C:\Users\neils\AppData\Local\Android\Sdk\tools\bin”)

→ Press New button (5)

→ Enter ‘tools’ folder location in the Java SDK installation folder

(e.g. “C:\Users\neils\AppData\Local\Android\Sdk\tools”)

→ Press OK button (6))



You must be logged on as an Administrator to make most of these changes.

Performance
Visual effects, processor scheduling, memory usage, and virtual memory

User Profiles
Desktop settings related to your sign-in

Startup and Recovery
System startup, system failure, and debugging information

User variables for neils

Variable	Value
OneDrive	C:\Users\neils\OneDrive
OneDriveConsumer	C:\Users\neils\OneDrive
Path	C:\Users\neils\AppData\Local\Microsoft\WindowsApps;
TEMP	C:\Users\neils\AppData\Local\Temp
TMP	C:\Users\neils\AppData\Local\Temp

System variables

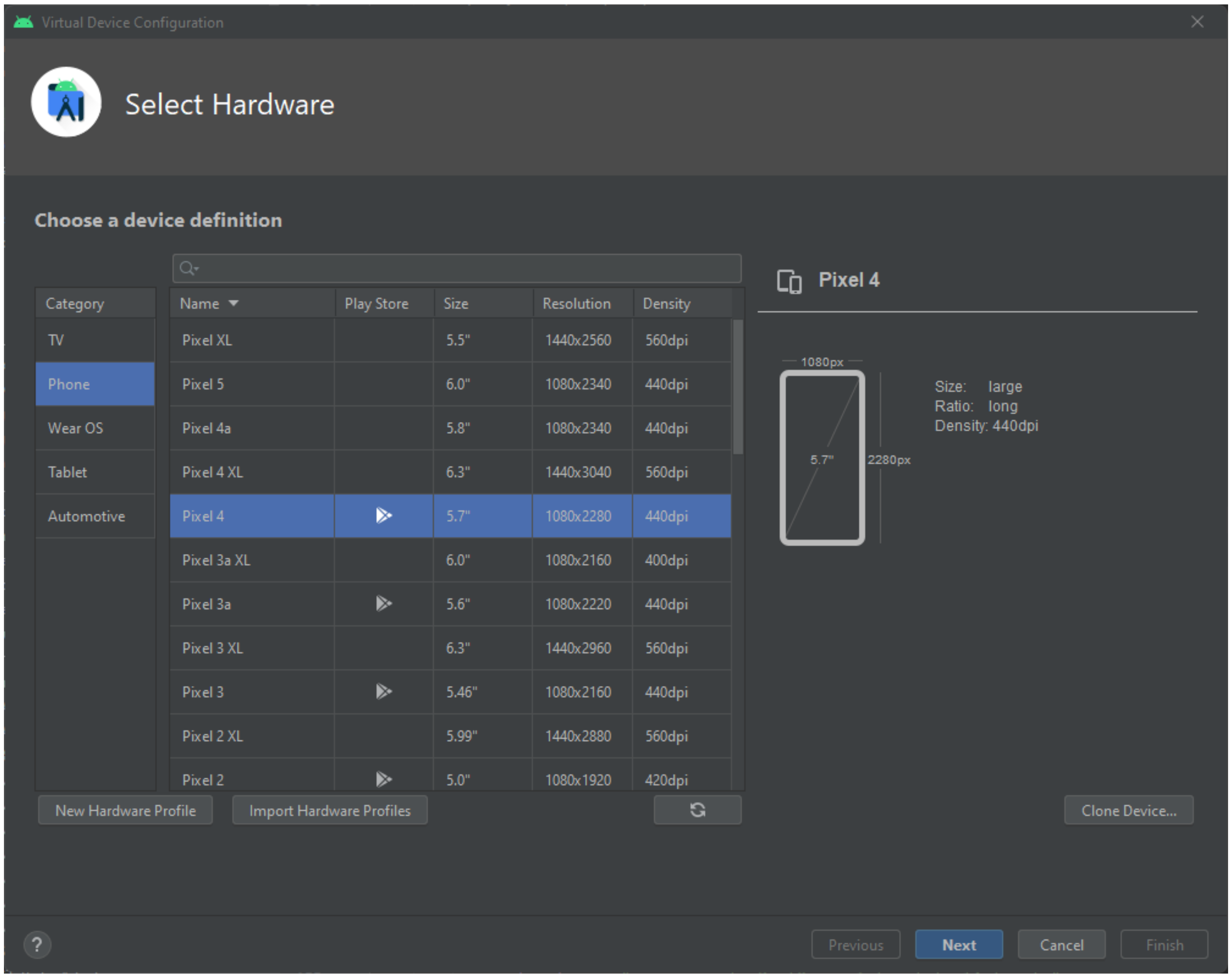
Variable	Value
DriverData	C:\Windows\System32\Drivers\DriverData
JAVA_HOME	C:\Program Files\Java\jdk1.8.0_321
NUMBER_OF_PROCESSORS	16
OS	Windows_NT
Path	C:\Program Files\Java\jdk1.8.0_321\bin;C:\Users\neils\AppData\Loc...
PATHEXT	.COM;.EXE;.BAT;.CMD;.VBS;.VBE;.JS;.JSE;.WSF;.WSH;.MSC
PROCESSOR_ARCHITECTURE	AMD64

```
C:\Program Files\Java\jdk1.8.0_321\bin
C:\Users\neils\AppData\Local\Android\Sdk\platform-tools
C:\Users\neils\AppData\Local\Android\Sdk\tools\bin
C:\Users\neils\AppData\Local\Android\Sdk\tools
C:\Program Files\Common Files\Oracle\Java\javapath
C:\Program Files (x86)\Common Files\Oracle\Java\javapath
C:\Windows\system32
C:\Windows
C:\Windows\System32\Wbem
C:\Windows\System32\WindowsPowerShell\v1.0\
C:\Windows\System32\OpenSSH\
C:\Program Files (x86)\NVIDIA Corporation\PhysX\Common
C:\Program Files\NVIDIA Corporation\NVIDIA NvDLISR
C:\Program Files\Git\cmd
%SystemRoot%\system32
%SystemRoot%
%SystemRoot%\System32\Wbem
%SYSTEMROOT%\System32\WindowsPowerShell\v1.0\
%SYSTEMROOT%\System32\OpenSSH\
```

3. Create an Android Virtual Device

1. Android Studio > Tools > Device Manager > Create device >

2. Select Hardware: “Phone” > “Pixel 4” > “Next”



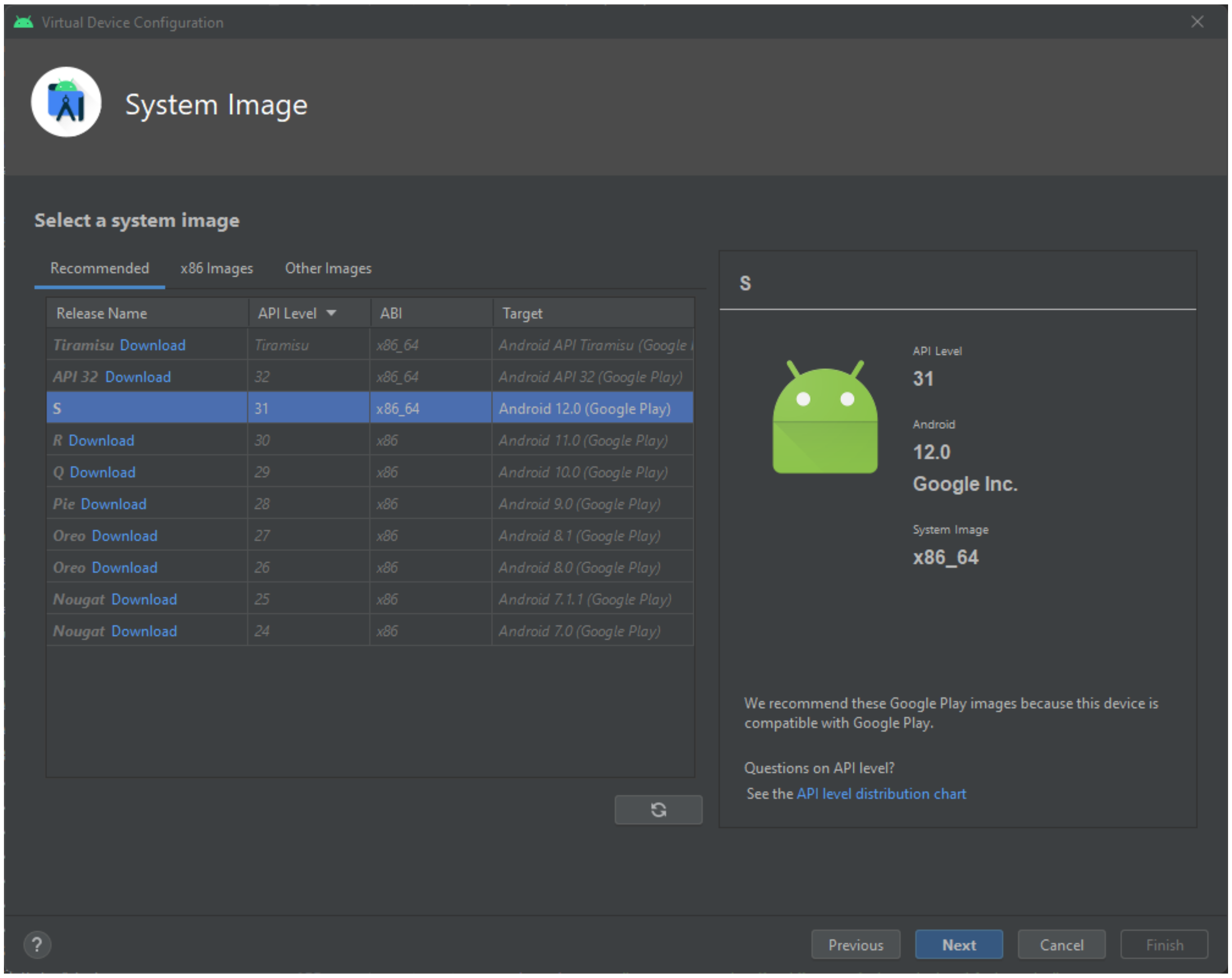
Category	Name	Play Store	Size	Resolution	Density
TV	Pixel XL		5.5"	1440x2560	560dpi
Phone	Pixel 5		6.0"	1080x2340	440dpi
Wear OS	Pixel 4a		5.8"	1080x2340	440dpi
Tablet	Pixel 4 XL		6.3"	1440x3040	560dpi
Automotive	Pixel 4	▶	5.7"	1080x2280	440dpi
	Pixel 3a XL		6.0"	1080x2160	400dpi
	Pixel 3a	▶	5.6"	1080x2220	440dpi
	Pixel 3 XL		6.3"	1440x2960	560dpi
	Pixel 3	▶	5.46"	1080x2160	440dpi
	Pixel 2 XL		5.99"	1440x2880	560dpi
	Pixel 2	▶	5.0"	1080x1920	420dpi

Pixel 4

1080px
5.7"
2280px

Size: large
Ratio: long
Density: 440dpi

3. Select a system image: “API Level 31” (Release Name: “S’) > “Next”



Select a system image

Release Name	API Level	ABI	Target
Tiramisu Download	Tiramisu	x86_64	Android API Tiramisu (Google
API 32 Download	32	x86_64	Android API 32 (Google Play)
S	31	x86_64	Android 12.0 (Google Play)
R Download	30	x86	Android 11.0 (Google Play)
Q Download	29	x86	Android 10.0 (Google Play)
Pie Download	28	x86	Android 9.0 (Google Play)
Oreo Download	27	x86	Android 8.1 (Google Play)
Oreo Download	26	x86	Android 8.0 (Google Play)
Nougat Download	25	x86	Android 7.1.1 (Google Play)
Nougat Download	24	x86	Android 7.0 (Google Play)

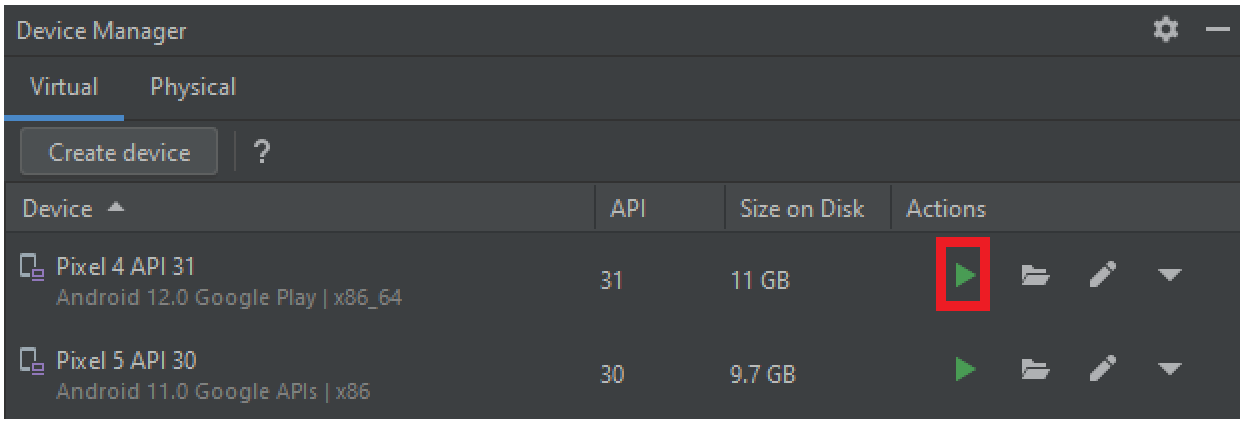
S

API Level
31
Android
12.0
Google Inc.
System Image
x86_64

We recommend these Google Play images because this device is compatible with Google Play.

Questions on API level?
See the API level distribution chart

4. Press “Finish” button
5. After creating the virtual device, press Start button next to the device created.



Device	API	Size on Disk	Actions
Pixel 4 API 31 Android 12.0 Google Play | x86_64	31	11 GB	
Pixel 5 API 30 Android 11.0 Google APIs | x86	30	9.7 GB	

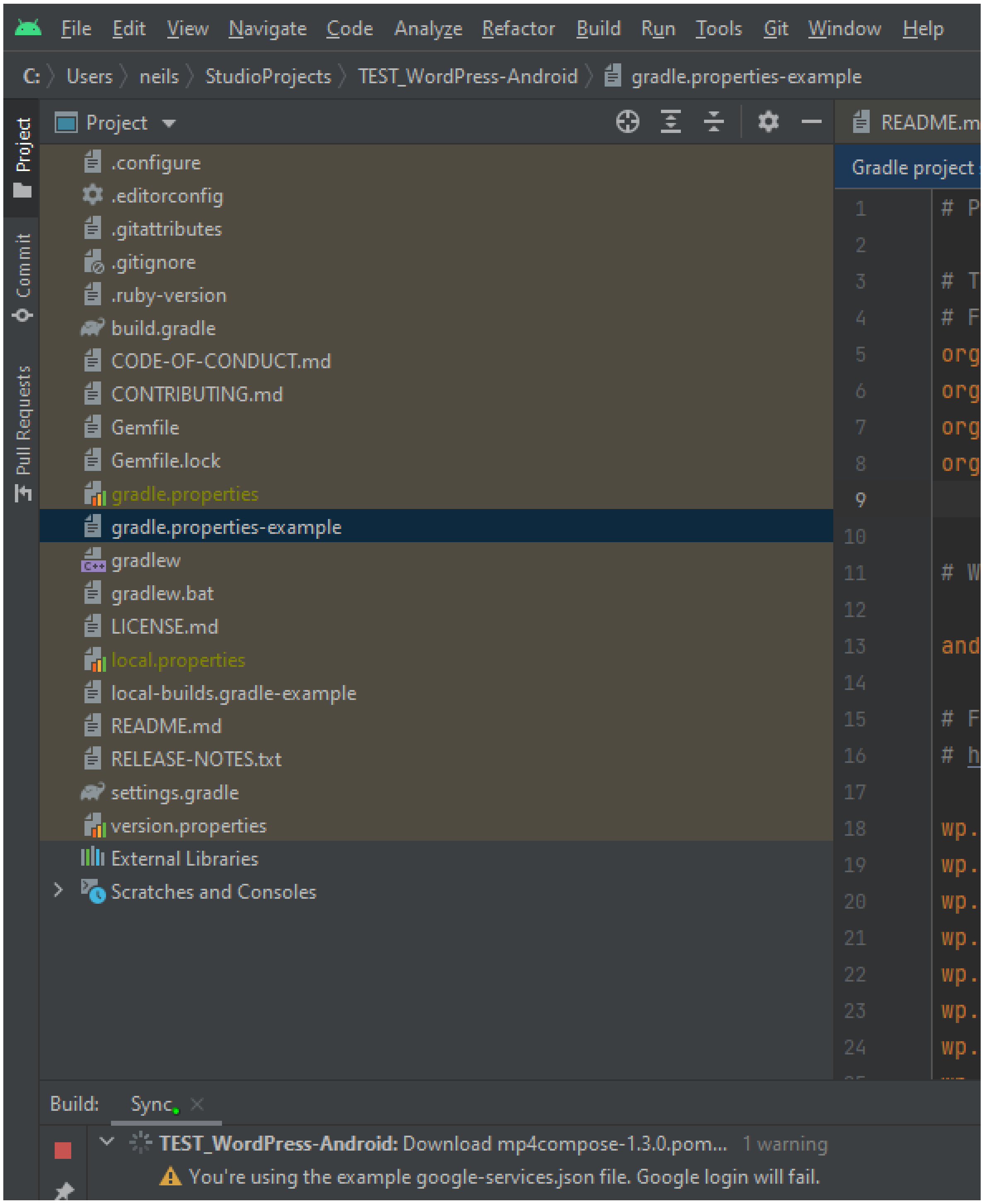
4. Download and import WordPress Android app on Android Studio

1. Android Studio > Git > Clone
2. Type WordPress-Android’s repository url:

   https://github.com/wordpress-mobile/WordPress-Android

Then, Android Studio will download the source code.

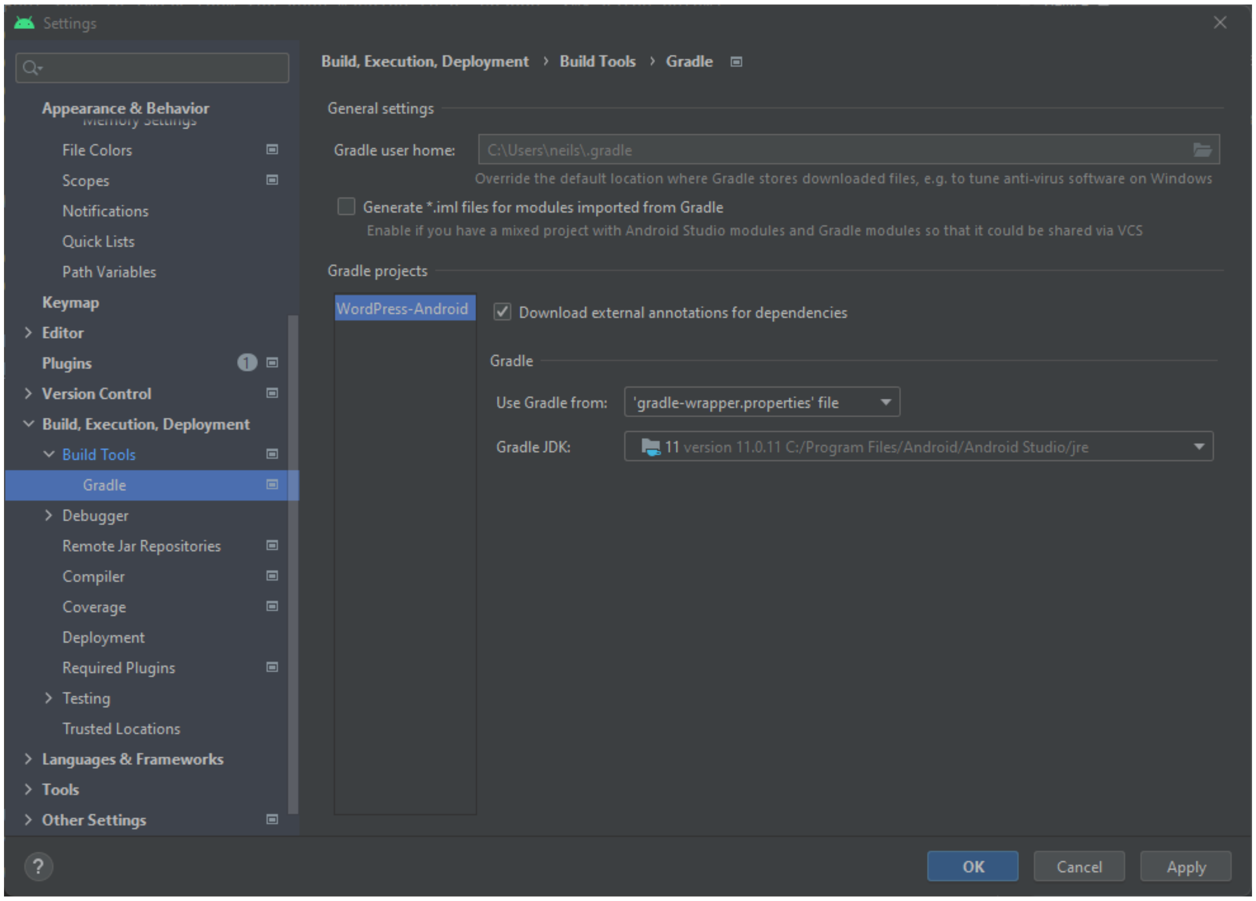
3. After downloading the source code and opening the project, Android Studio will automatically try to build the app. However it might fail because "gradle.properties" file does not exist. Copy "gradle.properties-example" and paste it as "gradle.properties"



```
TEST_WordPress-Android: Download mp4compose-1.3.0.pom...   1 warning
    You're using the example google-services.json file. Google login will fail.
```

4. Gradle JDK needs to be changed to Java SDK 11 version.

Android Studio > File > Settings > Build, Execution, Deployment > Build Tools > Gradle > Gradle JDK: “11 version”


Settings

Build, Execution, Deployment › Build Tools › Gradle

General settings

Gradle user home: C:\Users\neils\.gradle

Override the default location where Gradle stores downloaded files, e.g. to tune anti-virus software on Windows

☐ Generate *.iml files for modules imported from Gradle

Enable if you have a mixed project with Android Studio modules and Gradle modules so that it could be shared via VCS

Gradle projects

WordPress-Android

☑ Download external annotations for dependencies

Gradle

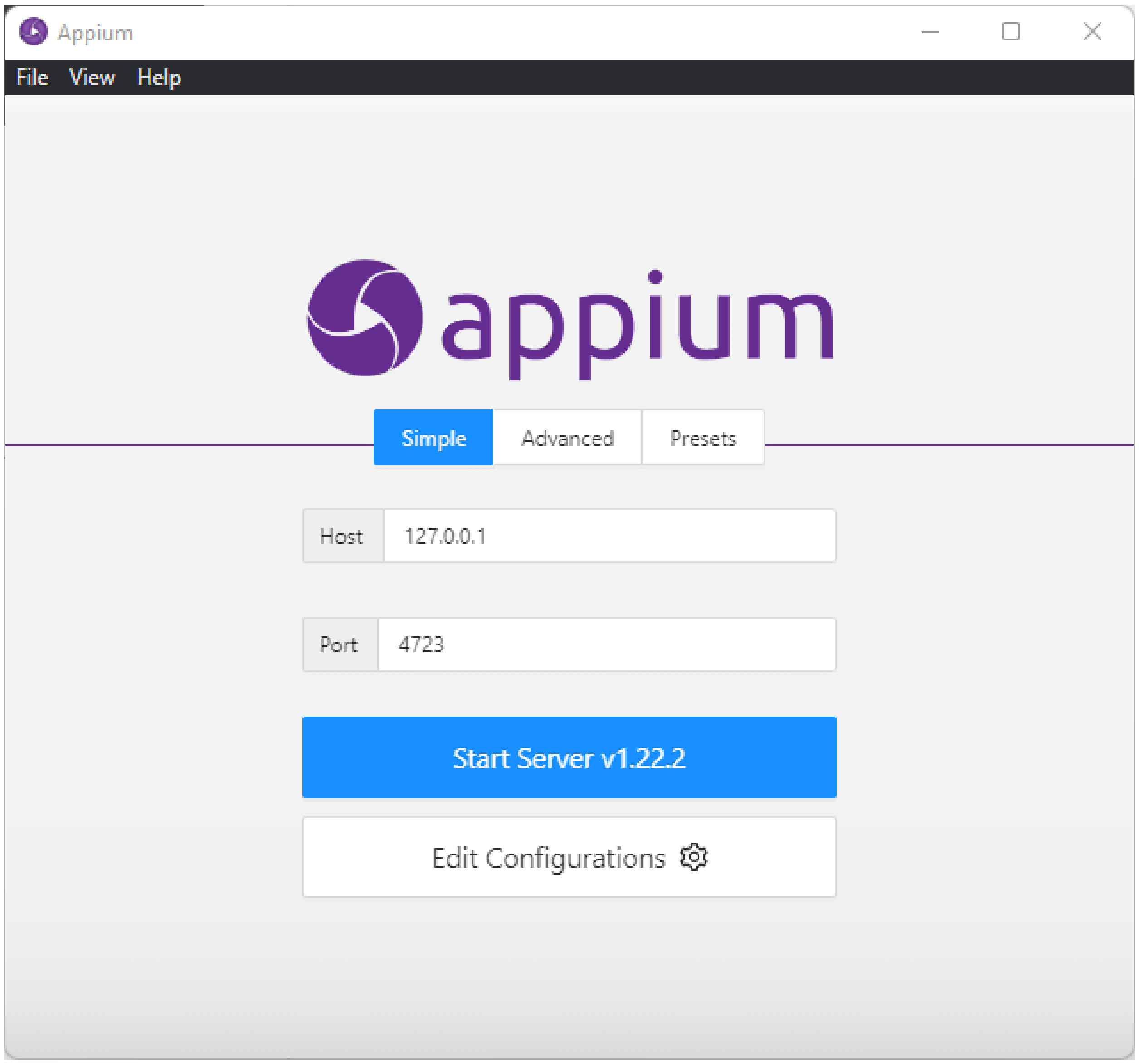
Use Gradle from: 'gradle-wrapper.properties' file

Gradle JDK: 11 version 11.0.11 C:/Program Files/Android/Android Studio/jre

5. Test run the app

5. Download and install Appium Server, Appium Inspector

1. Download and install Appium Server from: https://github.com/appium/appium-desktop/releases/tag/v1.22.3
2. Open Appium Server GUI
3. Put 127.0.0.1 for Host, and ‘4723’ for Port. Then press Start Server button

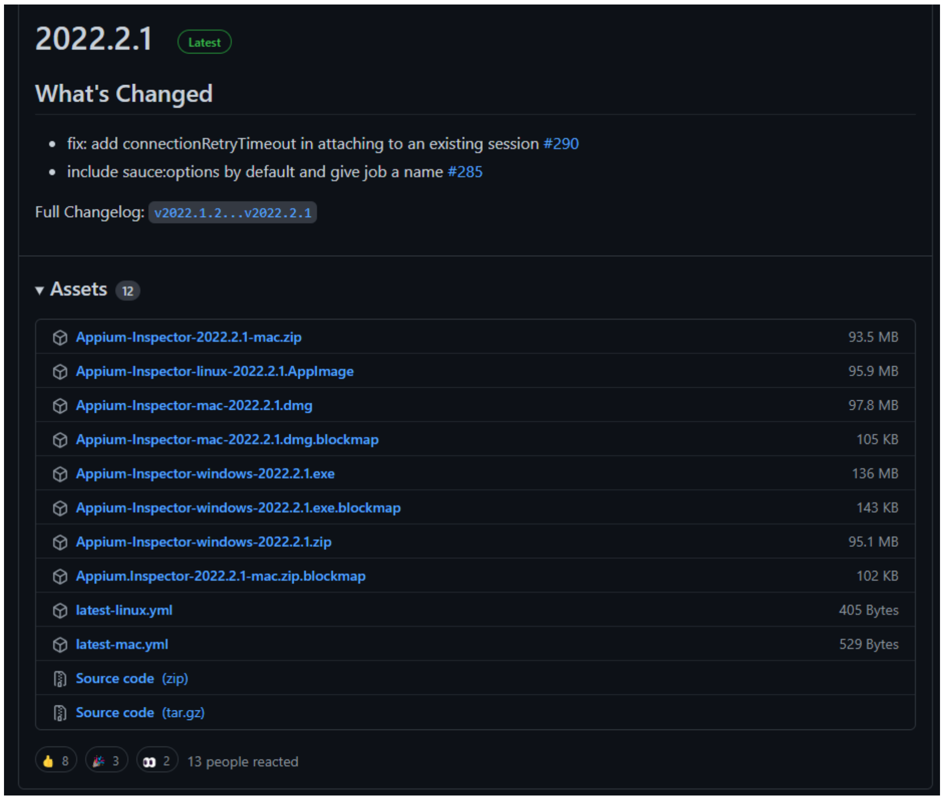


Host 127.0.0.1

Port 4723

4. Then, Appium Server will start

5. Download and install Appium Inspector from: https://github.com/appium/appium-inspector/releases



2022.2.1 Latest

What's Changed

- fix: add connectionRetryTimeout in attaching to an existing session #290
- include sauce:options by default and give job a name #285

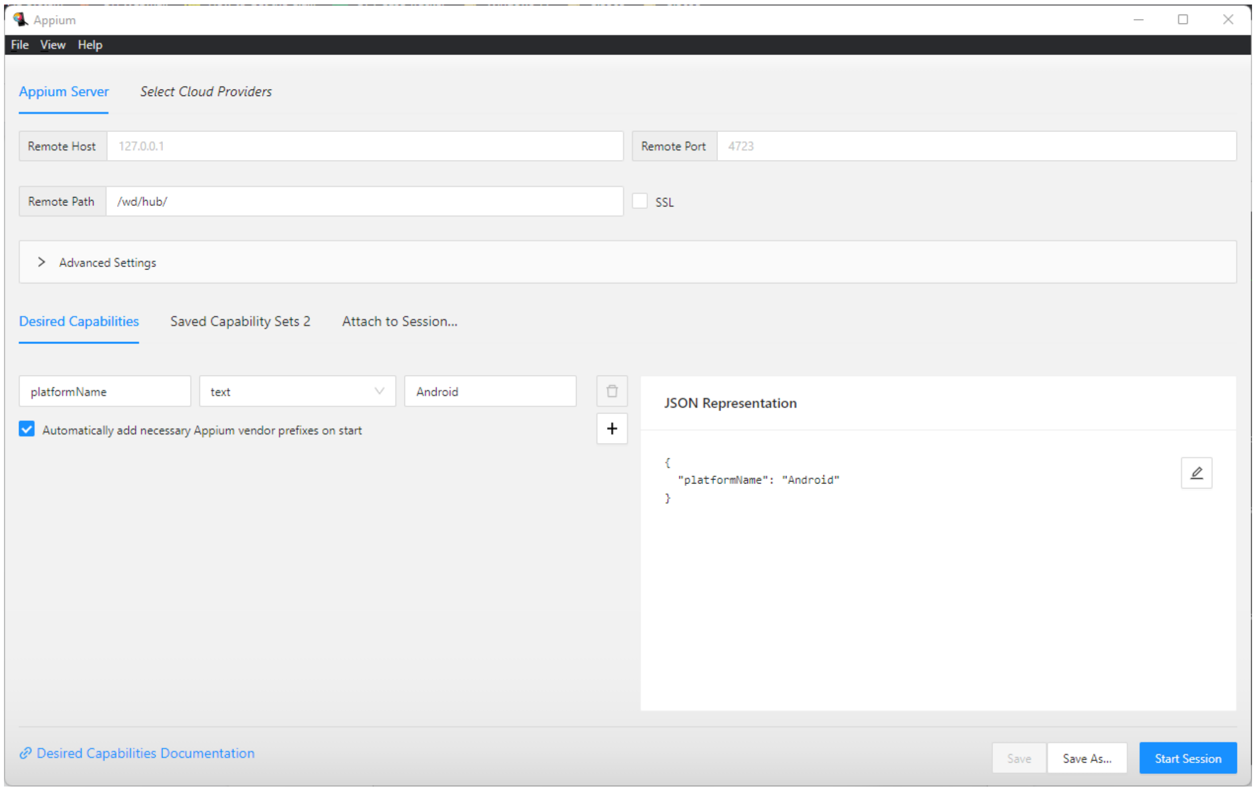
Full Changelog: v2022.1.2...v2022.2.1

Assets 12

Appium-Inspector-2022.2.1-mac.zip	93.5 MB
Appium-Inspector-linux-2022.2.1.AppImage	95.9 MB
Appium-Inspector-mac-2022.2.1.dmg	97.8 MB
Appium-Inspector-mac-2022.2.1.dmg.blockmap	105 KB
Appium-Inspector-windows-2022.2.1.exe	136 MB
Appium-Inspector-windows-2022.2.1.exe.blockmap	143 KB
Appium-Inspector-windows-2022.2.1.zip	95.1 MB
Appium.Inspector-2022.2.1-mac.zip.blockmap	102 KB
latest-linux.yml	405 Bytes
latest-mac.yml	529 Bytes
Source code (zip)	
Source code (tar.gz)	

8 3 2 13 people reacted

6. Remote Host: 127.0.0.1
   Remote Port: 4723
   Remote Path: /wd/hub/
   Desired Capabilities >
      Name: platformName
      Value: Android

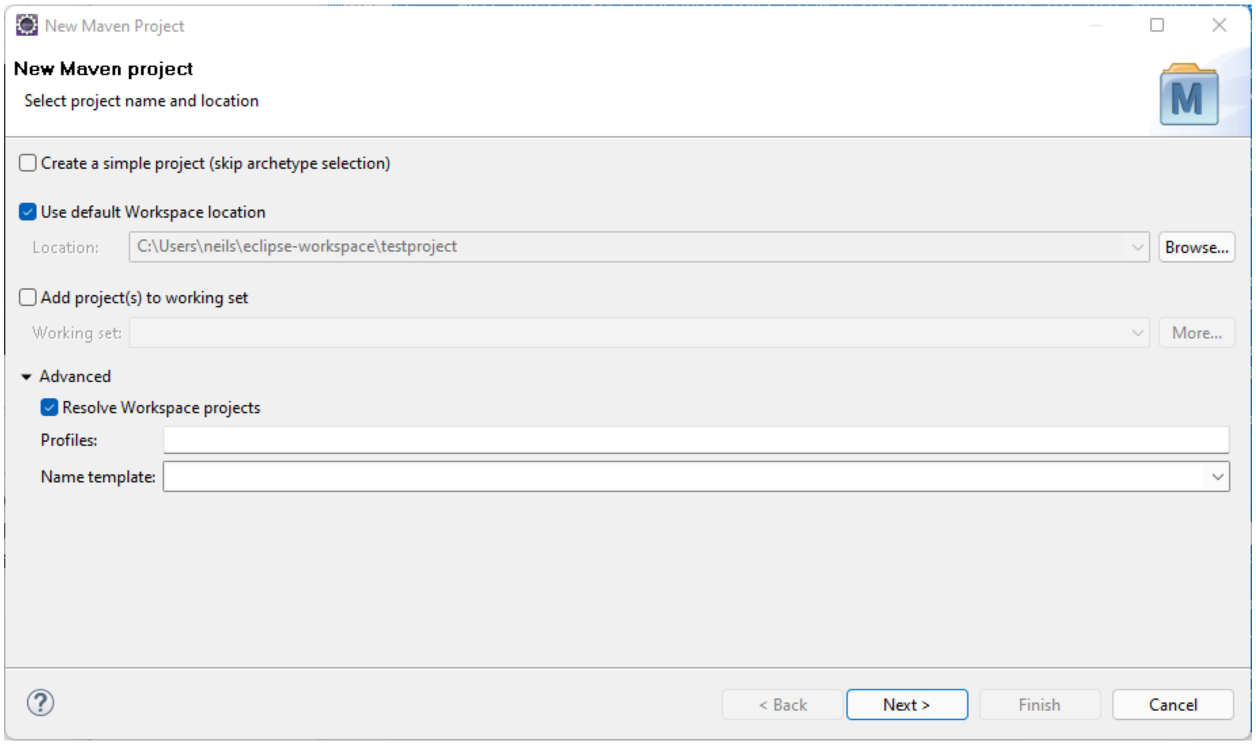


```
{
  "platformName": "Android"
}
```

7. Press Start Session button on the bottom right.

6. Download and setup Eclipse and TestNG

1. Download Eclipse IDE for Java EE Developers: https://www.eclipse.org/downloads/
2. Launch Eclipse, select “Help” > “Install New Software” from the menu bar. Paste “http://beust.com/eclipse” into the “Work with:” field and press Enter. Eclipse will take a while to fetch the relevant package to display, select the checkbox next to TestNG and click “Next” to complete the dialogue. Eclipse will take a while to install the TestNG package and then prompt you to restart Eclipse.
3. Eclipse > File > New > Maven Project
4. Next


New Maven Project

New Maven project

Select project name and location

☐ Create a simple project (skip archetype selection)

☑ Use default Workspace location

Location: C:\Users\neils\eclipse-workspace\testproject

☐ Add project(s) to working set

Working set:

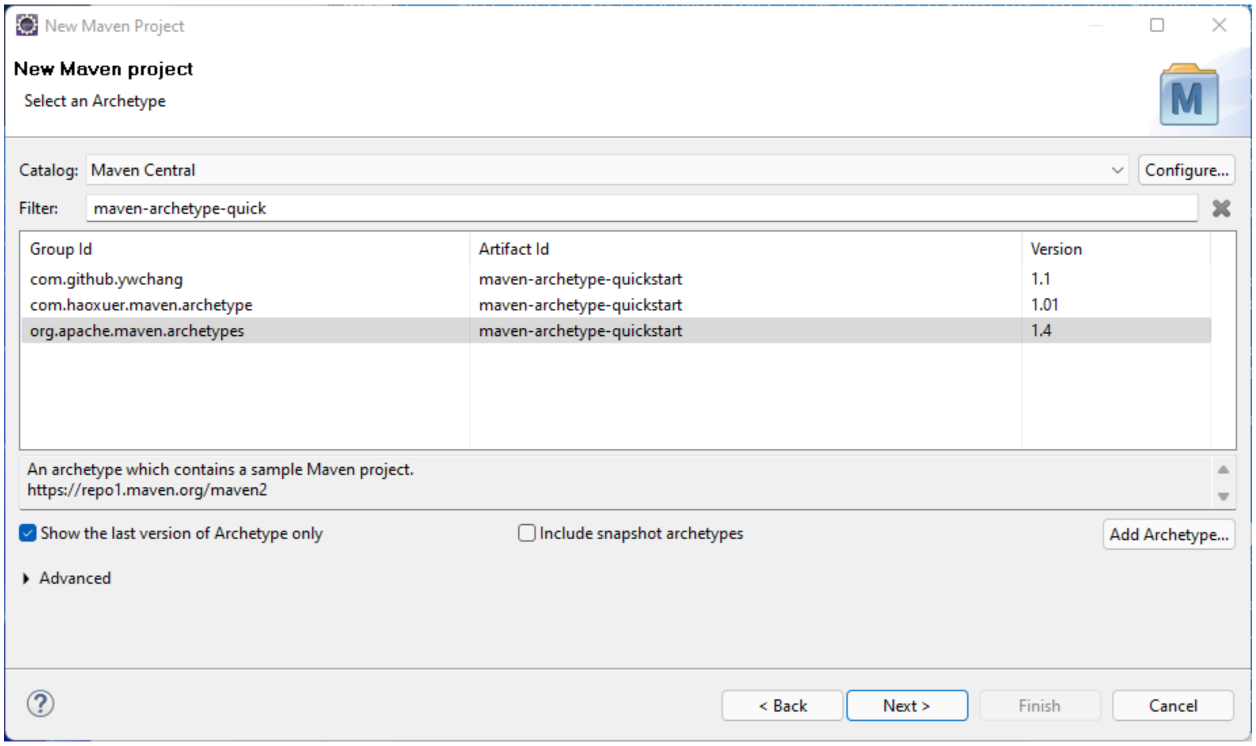
Advanced

☑ Resolve Workspace projects

Profiles:

Name template:

5. Select the following Archetype
    a. Group ID: org.apache.maven.archetypes
    b. Artifact ID: maven-archetype-quickstart


New Maven Project

New Maven project

Select an Archetype

Catalog: Maven Central

Filter: maven-archetype-quick

Group Id	Artifact Id	Version
com.github.ywchang	maven-archetype-quickstart	1.1
com.haoxuer.maven.archetype	maven-archetype-quickstart	1.01
org.apache.maven.archetypes	maven-archetype-quickstart	1.4

An archetype which contains a sample Maven project.
https://repo1.maven.org/maven2

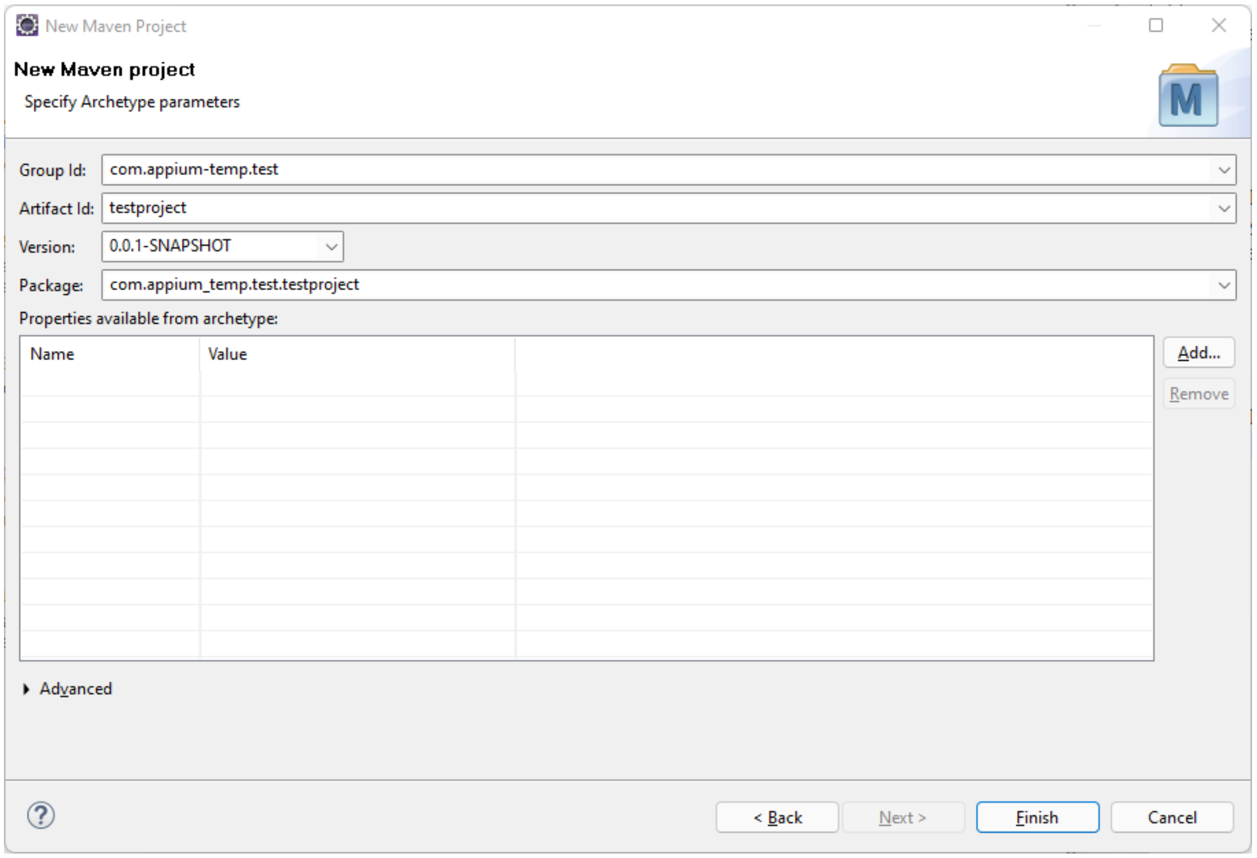
☑ Show the last version of Archetype only ☐ Include snapshot archetypes

Advanced

6. Type the following and press Finish button:
    a. Group ID: com.appium-temp.test

b. Artifact ID: testproject

c. Package: com.appium_temp.test.testproject



New Maven project

Specify Archetype parameters

Group Id: com.appium-temp.test

Artifact Id: testproject

Version: 0.0.1-SNAPSHOT

Package: com.appium_temp.test.testproject

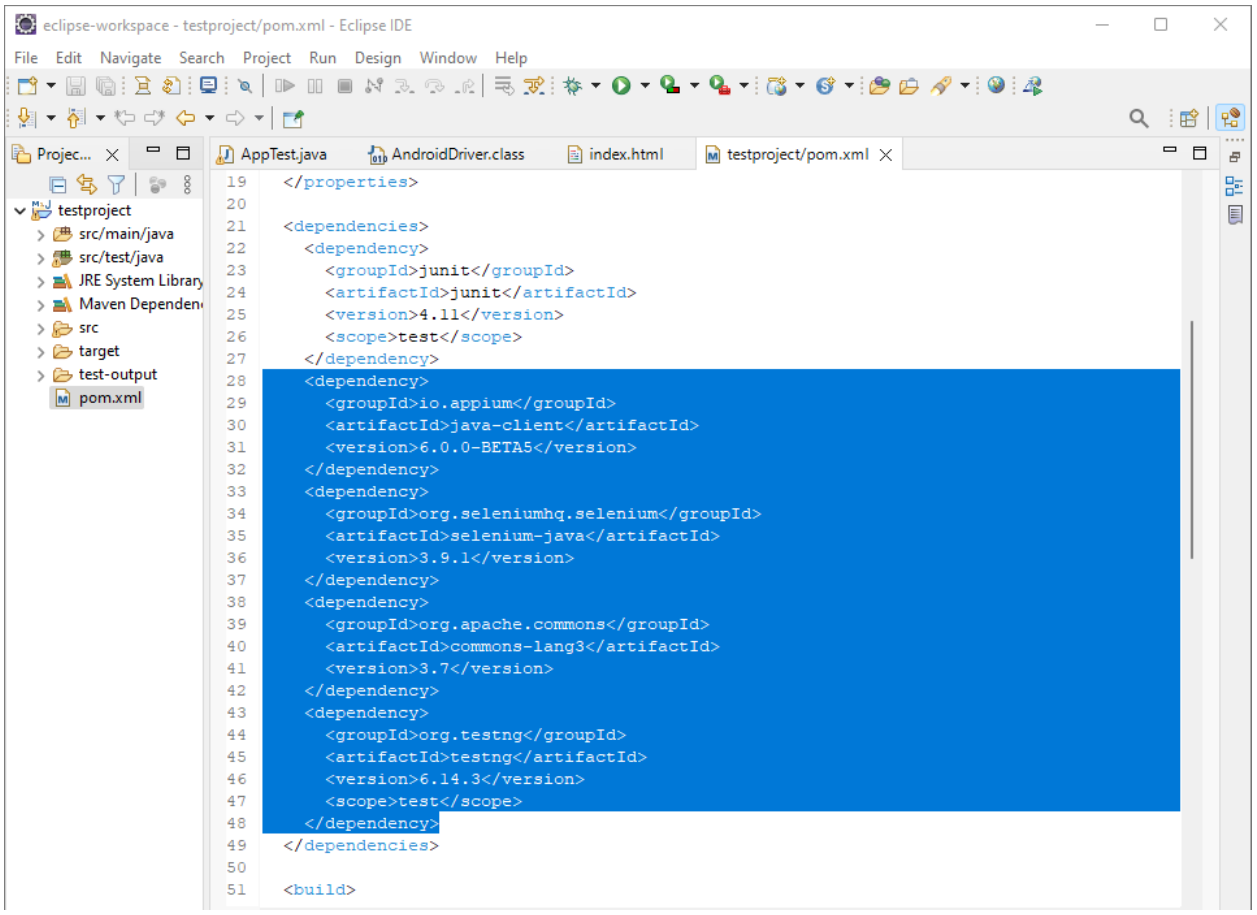
Properties available from archetype:

Name	Value

7. After the project is created, select "pom.xml" file and add the following dependencies:

```
<dependency>
  <groupId>io.appium</groupId>
  <artifactId>java-client</artifactId>
  <version>6.0.0-BETA5</version>
</dependency>
<dependency>
  <groupId>org.seleniumhq.selenium</groupId>
  <artifactId>selenium-java</artifactId>
  <version>3.9.1</version>
</dependency>
<dependency>
  <groupId>org.apache.commons</groupId>
  <artifactId>commons-lang3</artifactId>
  <version>3.7</version>
</dependency>
<dependency>
```

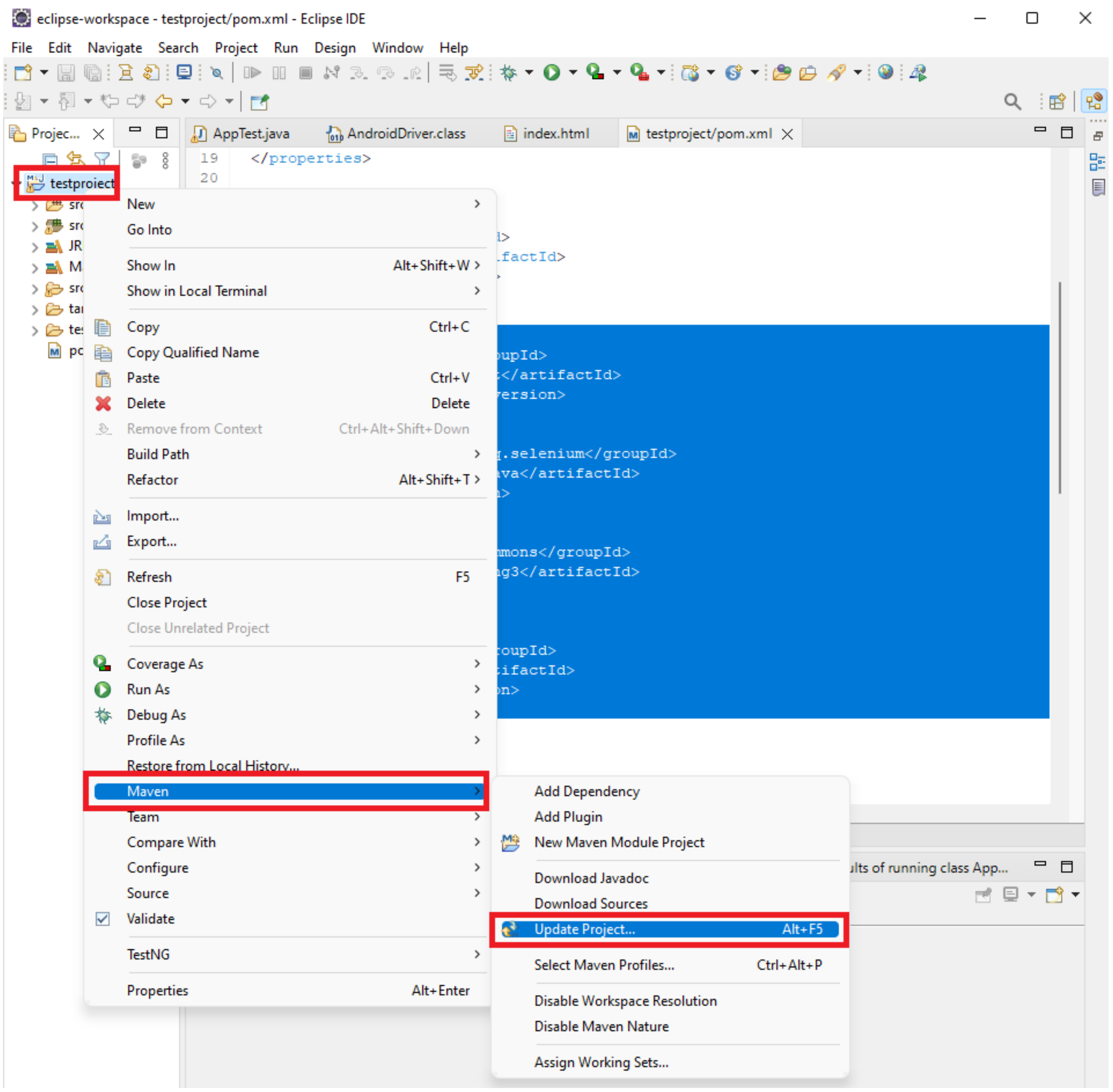
```
    <groupId>org.testng</groupId>
    <artifactId>testng</artifactId>
    <version>6.14.3</version>
    <scope>test</scope>
  </dependency>
```



```
  </properties>

  <dependencies>
    <dependency>
      <groupId>junit</groupId>
      <artifactId>junit</artifactId>
      <version>4.11</version>
      <scope>test</scope>
    </dependency>
    <dependency>
      <groupId>io.appium</groupId>
      <artifactId>java-client</artifactId>
      <version>6.0.0-BETA5</version>
    </dependency>
    <dependency>
      <groupId>org.seleniumhq.selenium</groupId>
      <artifactId>selenium-java</artifactId>
      <version>3.9.1</version>
    </dependency>
    <dependency>
      <groupId>org.apache.commons</groupId>
      <artifactId>commons-lang3</artifactId>
      <version>3.7</version>
    </dependency>
    <dependency>
      <groupId>org.testng</groupId>
      <artifactId>testng</artifactId>
      <version>6.14.3</version>
      <scope>test</scope>
    </dependency>
  </dependencies>

  <build>
```

Then, save the file.

8. On the Project panel, right-click the project name > Maven > Update Project



9. Open /src/test/java > com.appium_temp.test.testproject > AppTest.java and paste the following code:

```
package com.appium_temp.test.testproject;

import java.net.MalformedURLException;
import java.net.URL;
import java.util.concurrent.TimeUnit;

import org.openqa.selenium.By;
import org.openqa.selenium.remote.DesiredCapabilities;
import org.openqa.selenium.support.ui.ExpectedConditions;
import org.openqa.selenium.support.ui.WebDriverWait;
import org.testng.annotations.BeforeSuite;
import org.testng.annotations.AfterSuite;
import org.testng.annotations.BeforeMethod;
import org.testng.annotations.BeforeTest;
import org.testng.annotations.Test;
import io.appium.java_client.MobileElement;
import io.appium.java_client.TouchAction;
import io.appium.java_client.android.AndroidDriver;
import io.appium.java_client.remote.MobileCapabilityType;
import org.junit.Assert;
public class AppTestForDemo {
  public static URL url;
  public static DesiredCapabilities capabilities;
  public static AndroidDriver<MobileElement> driver;
  public static WebDriverWait wait;

  @BeforeSuite
  public void setupAppium() throws MalformedURLException {

    final String URL_STRING = "http://127.0.0.1:4723/wd/hub";
    url = new URL(URL_STRING);

    capabilities = new DesiredCapabilities();
    capabilities.setCapability(MobileCapabilityType.DEVICE_NAME, "Android Device");
    capabilities.setCapability("appPackage", "org.wordpress.android.prealpha");
    capabilities.setCapability("appActivity", "org.wordpress.android.ui.WPLaunchActivity");
    capabilities.setCapability(MobileCapabilityType.NO_RESET, true);
    capabilities.setCapability(MobileCapabilityType.AUTOMATION_NAME, "UiAutomator2");
    //4
    driver = new AndroidDriver<MobileElement>(url, capabilities);
    driver.manage().timeouts().implicitlyWait(2, TimeUnit.SECONDS);
    driver.resetApp();

    wait = new WebDriverWait(driver, 30);

  }

  @BeforeMethod
  public void setupBeforeEachMethod() throws InterruptedException {
                driver.resetApp();
  }

  @AfterSuite
  public void uninstallApp() throws InterruptedException {
               driver.resetApp();
  }

  @Test (enabled=true) public void loginTest001InvalidEmailFormat() throws InterruptedException {

                wait.until(ExpectedConditions.elementToBeClickable(By.id("org.wordpress.android.prealpha:id/continue_with_wpcom_button")));
                driver.findElement(By.id("org.wordpress.android.prealpha:id/continue_with_wpcom_button")).click();

                wait.until(ExpectedConditions.elementToBeClickable(By.id("org.wordpress.android.prealpha:id/input")));
                driver.findElement(By.id("org.wordpress.android.prealpha:id/input")).sendKeys("invalid_email_format");

                wait.until(ExpectedConditions.elementToBeClickable(By.id("org.wordpress.android.prealpha:id/login_continue_button")));
                driver.findElement(By.id("org.wordpress.android.prealpha:id/login_continue_button")).click();

                wait.until(ExpectedConditions.elementToBeClickable(By.id("org.wordpress.android.prealpha:id/text_input_error_icon")));
                wait.until(ExpectedConditions.elementToBeClickable(By.id("org.wordpress.android.prealpha:id/textinput_error")));
                Assert.assertEquals(driver.findElement(By.id("org.wordpress.android.prealpha:id/textinput_error")).getText(), "Enter a valid email address");

  }
}
```

10. Run the code.

Performance testing:

Jmeter Setup and implementation:

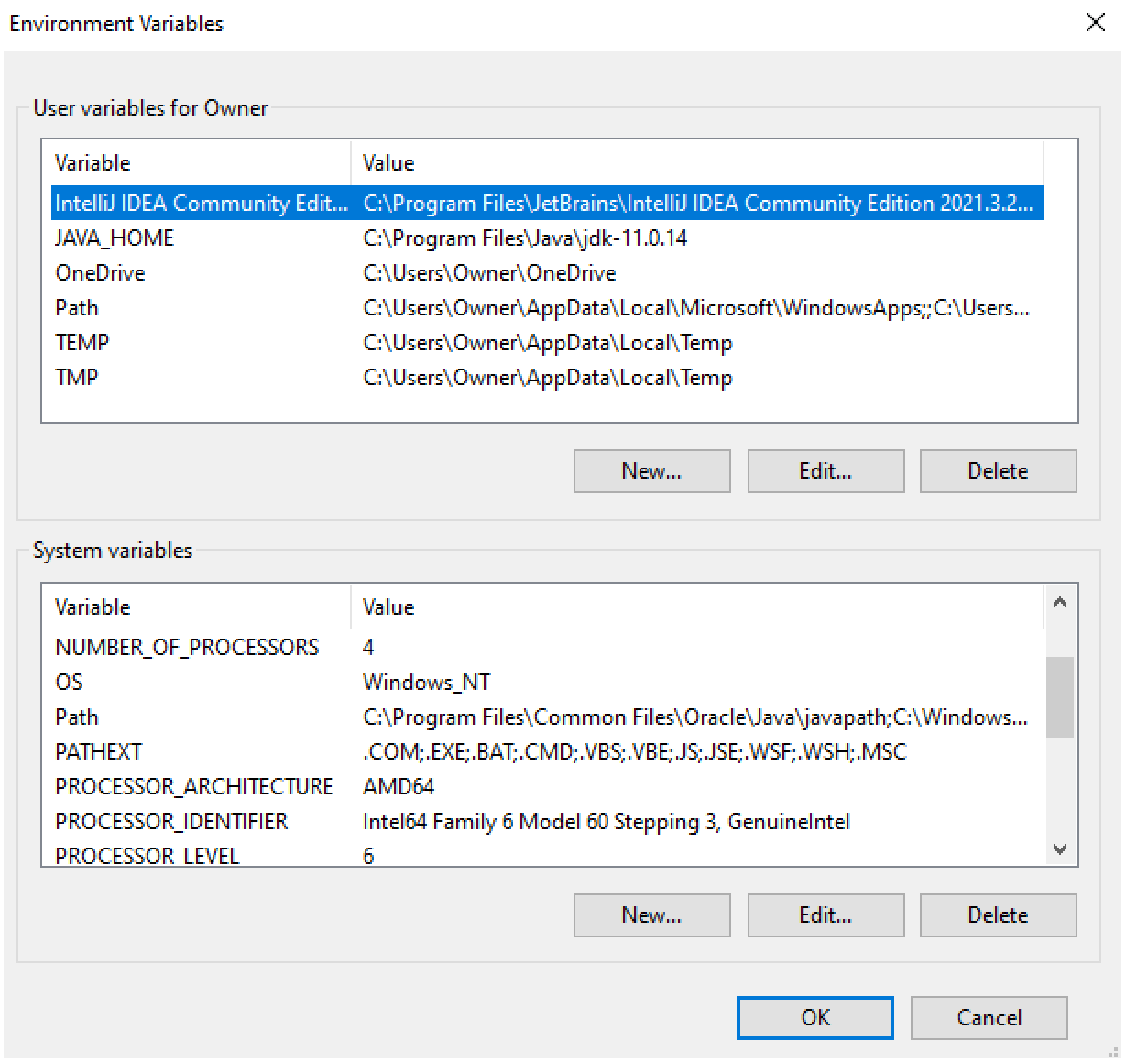
Install java jdk 11.0.4 from following website

https://docs.google.com/document/d/1AfckoQ947dhBYXEooFwZb6r1QzwHGW6i/edit

Install Jmeter from the following site

https://jmeter.apache.org/download_jmeter.cgi

After installation environment Variable name and path variable should be declared steps are following



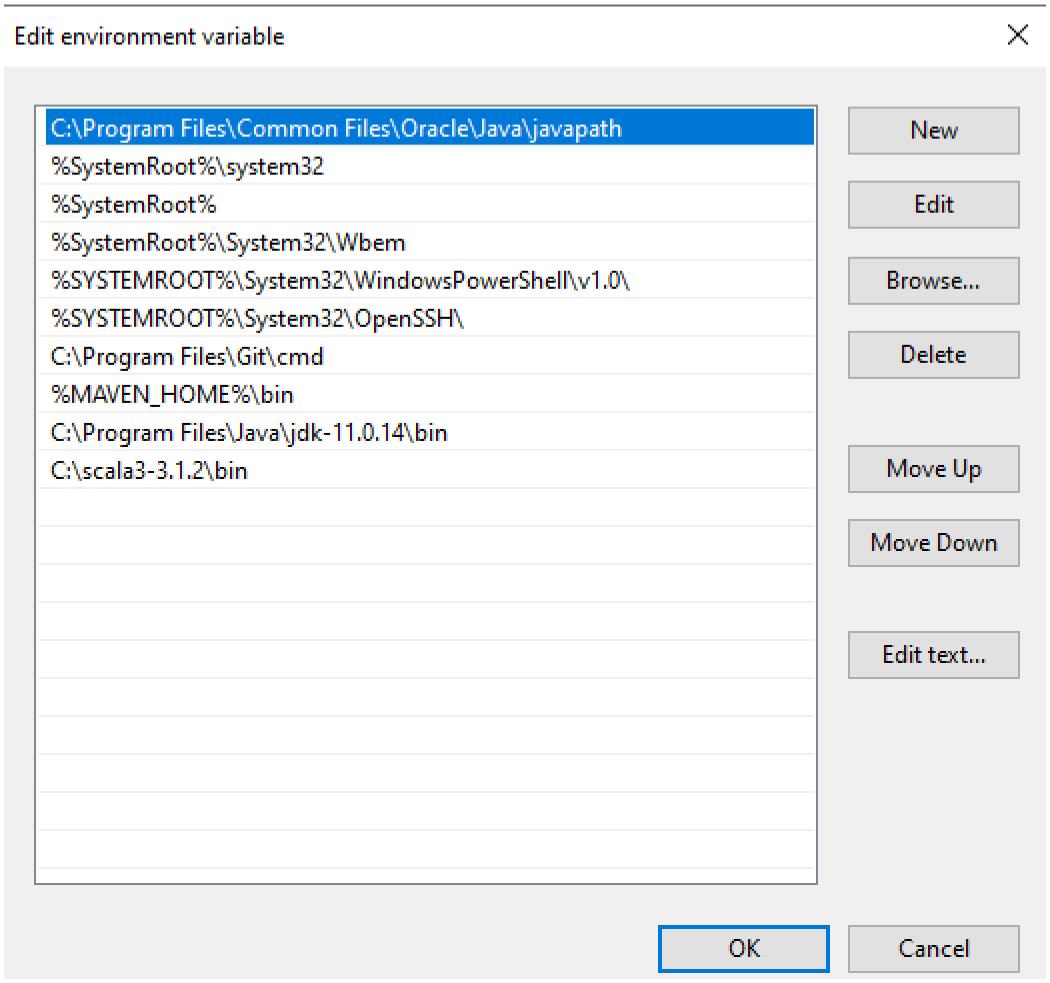
Environment Variables

User variables for Owner

Variable	Value
IntelliJ IDEA Community Edit...	C:\Program Files\JetBrains\IntelliJ IDEA Community Edition 2021.3.2...
JAVA_HOME	C:\Program Files\Java\jdk-11.0.14
OneDrive	C:\Users\Owner\OneDrive
Path	C:\Users\Owner\AppData\Local\Microsoft\WindowsApps;;C:\Users...
TEMP	C:\Users\Owner\AppData\Local\Temp
TMP	C:\Users\Owner\AppData\Local\Temp

System variables

Variable	Value
NUMBER_OF_PROCESSORS	4
OS	Windows_NT
Path	C:\Program Files\Common Files\Oracle\Java\javapath;C:\Windows...
PATHEXT	.COM;.EXE;.BAT;.CMD;.VBS;.VBE;.JS;.JSE;.WSF;.WSH;.MSC
PROCESSOR_ARCHITECTURE	AMD64
PROCESSOR_IDENTIFIER	Intel64 Family 6 Model 60 Stepping 3, GenuineIntel
PROCESSOR_LEVEL	6



```
C:\Program Files\Common Files\Oracle\Java\javapath
%SystemRoot%\system32
%SystemRoot%
%SystemRoot%\System32\Wbem
%SYSTEMROOT%\System32\WindowsPowerShell\v1.0\
%SYSTEMROOT%\System32\OpenSSH\
C:\Program Files\Git\cmd
%MAVEN_HOME%\bin
C:\Program Files\Java\jdk-11.0.14\bin
C:\scala3-3.1.2\bin
```

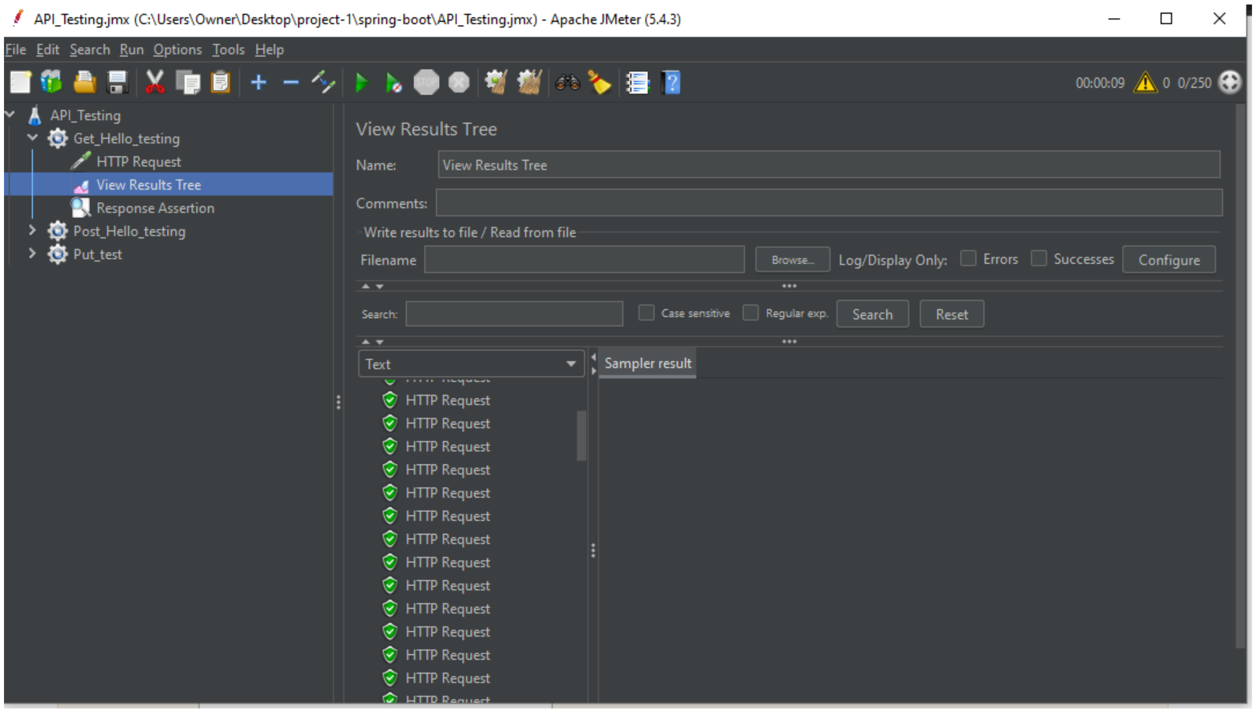
Open the apache jmeter from jmeter.bat under the Apache jmeter folder.Then write the test cases for following project:

https://github.com/TechPrimers/spring-boot-swagger-example

In Jmeter IDE following steps have taken for testing the project

add>thread group>sampler>listener>assertion

The result in the IDE will be shown in the following



<span style="color:red">For Gatling Installation :</span>

https://gatling.io/

This is the gatling website from there the gatling zip file has been uploaded.

unzip the gatling folder.

For running the gatling we also need scala,maven and java.

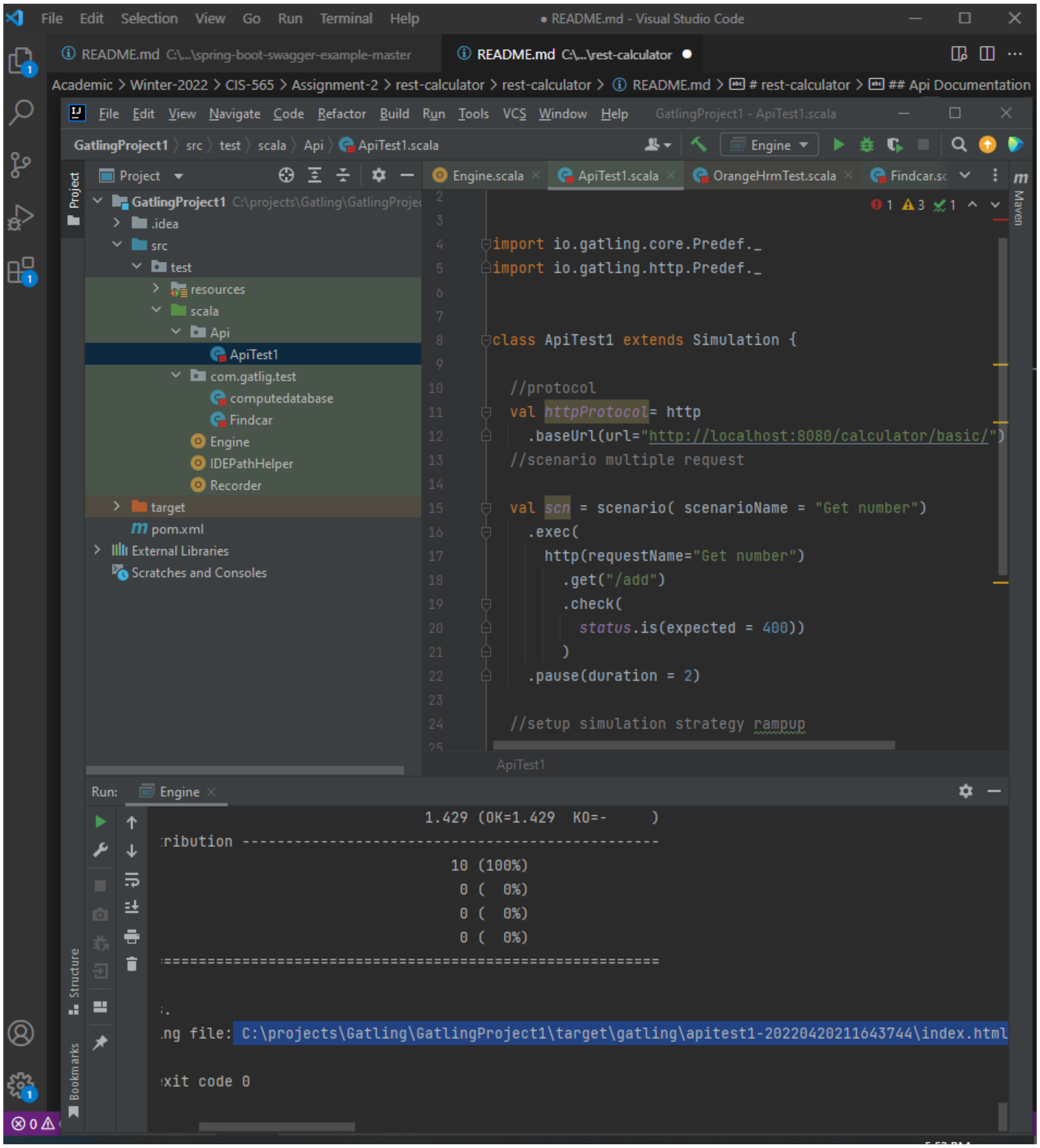
After installing all the prerequisites tools for gatling I created a maven project from the following command prompt:

Create a folder>Open CMD>navigate to folder>run command maven Create a fold

er>Open CMD>navigate to folder>run command maven archetype:generate

It will create a project then open it in the Intellij IDE

It will give the following output:


README.md - Visual Studio Code
Academic > Winter-2022 > CIS-565 > Assignment-2 > rest-calculator > rest-calculator > README.md > # rest-calculator > ## Api Documentation
GatlingProject1 - ApiTest1.scala
GatlingProject1 > src > test > scala > Api > ApiTest1.scala
GatlingProject1 C:\projects\Gatling\GatlingProje
.idea
src
test
resources
scala
Api
ApiTest1
com.gatlig.test
computedatabase
Findcar
Engine
IDEPathHelper
Recorder
target
pom.xml
External Libraries
Scratches and Consoles
Engine.scala
ApiTest1.scala
OrangeHrmTest.scala
Run: Engine

```
import io.gatling.core.Predef._
import io.gatling.http.Predef._


class ApiTest1 extends Simulation {

  //protocol
  val httpProtocol= http
    .baseUrl(url="http://localhost:8080/calculator/basic/")
  //scenario multiple request

  val scn = scenario( scenarioName = "Get number")
    .exec(
      http(requestName="Get number")
        .get("/add")
        .check(
          status.is(expected = 400))
        )
    .pause(duration = 2)

  //setup simulation strategy rampup
```

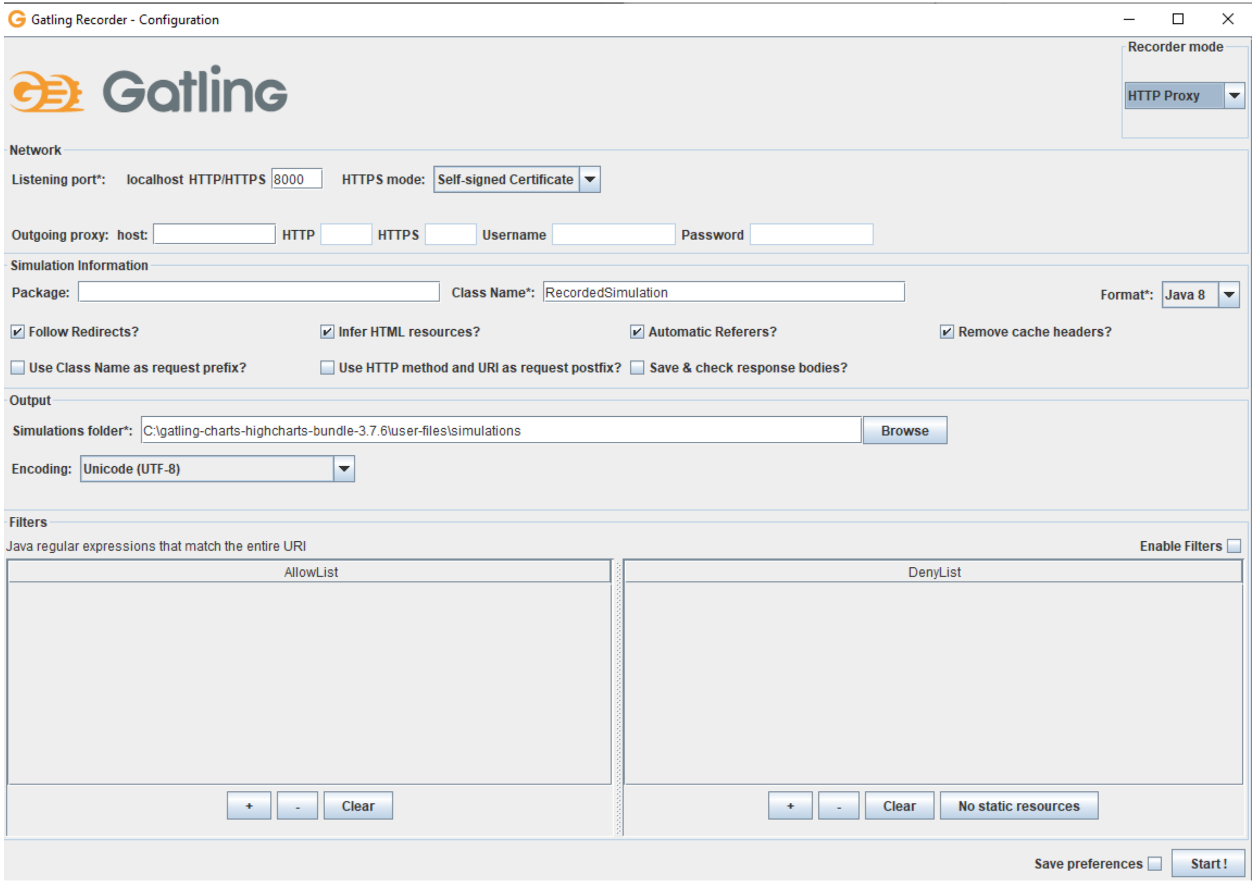
```
                              1.429 (OK=1.429   KO=-      )
:ribution ---------------------------------------------------
                                 10 (100%)
                                  0 (  0%)
                                  0 (  0%)
                                  0 (  0%)
:==========================================================

:.
.ng file: C:\projects\Gatling\GatlingProject1\target\gatling\apitest1-20220420211643744\index.html

:xit code 0
```

For Recording, here a Open source project is used and after recording it will create a HAR file then open it to a gatling web recorder.

for open a gatling web recorder window :

Goto >project>recorder class>run

It will show the following window



Gatling Recorder - Configuration

Recorder mode: HTTP Proxy

Network

Listening port*: localhost HTTP/HTTPS 8000   HTTPS mode: Self-signed Certificate

Outgoing proxy: host: HTTP   HTTPS   Username   Password

Simulation Information

Package: Class Name*: RecordedSimulation   Format*: Java 8

☑ Follow Redirects?   ☑ Infer HTML resources?   ☑ Automatic Referers?   ☑ Remove cache headers?

☐ Use Class Name as request prefix?   ☐ Use HTTP method and URI as request postfix?   ☐ Save & check response bodies?

Output

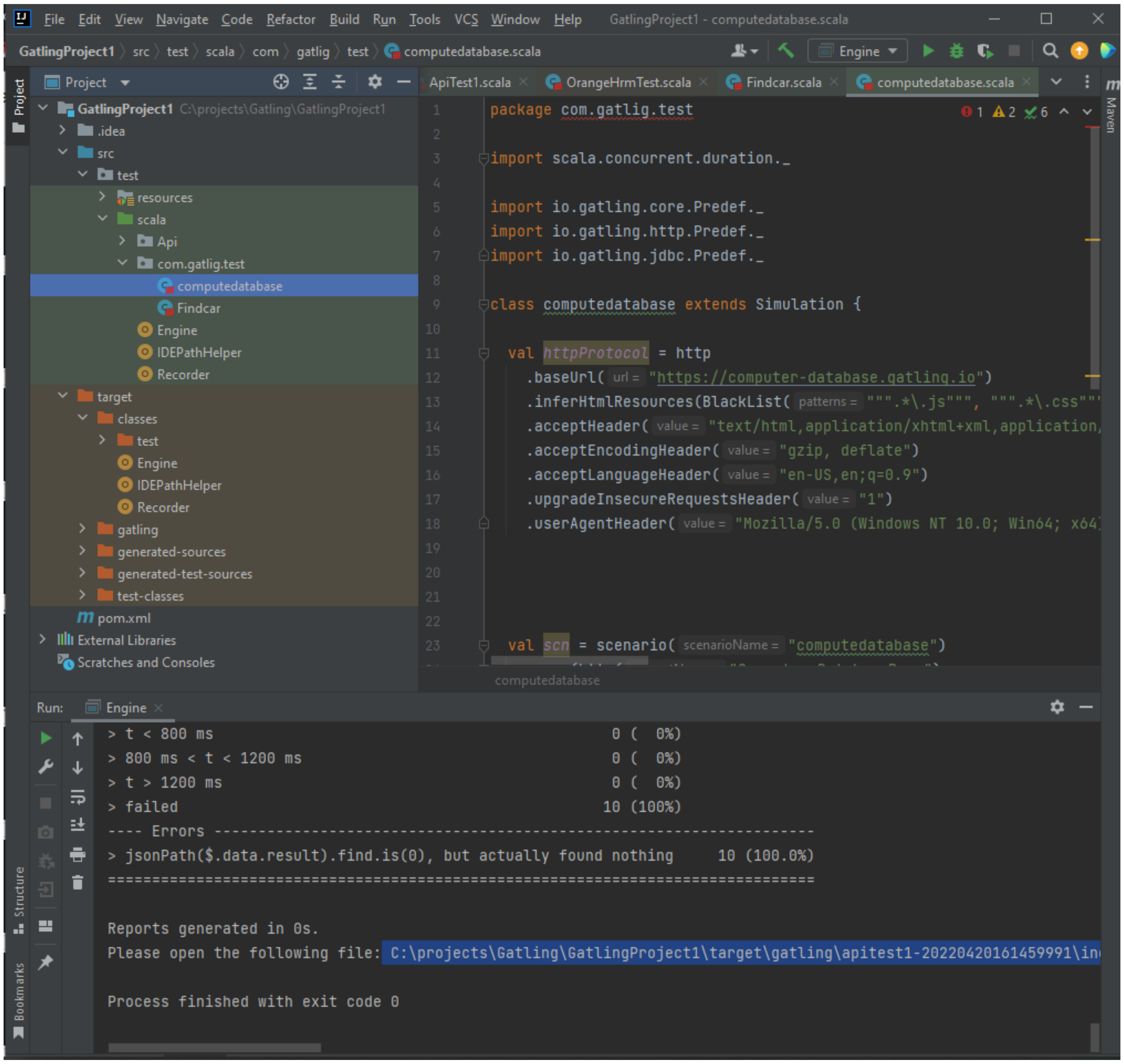
Simulations folder*: C:\gatling-charts-highcharts-bundle-3.7.6\user-files\simulations

Encoding: Unicode (UTF-8)

Filters

Java regular expressions that match the entire URI   Enable Filters ☐

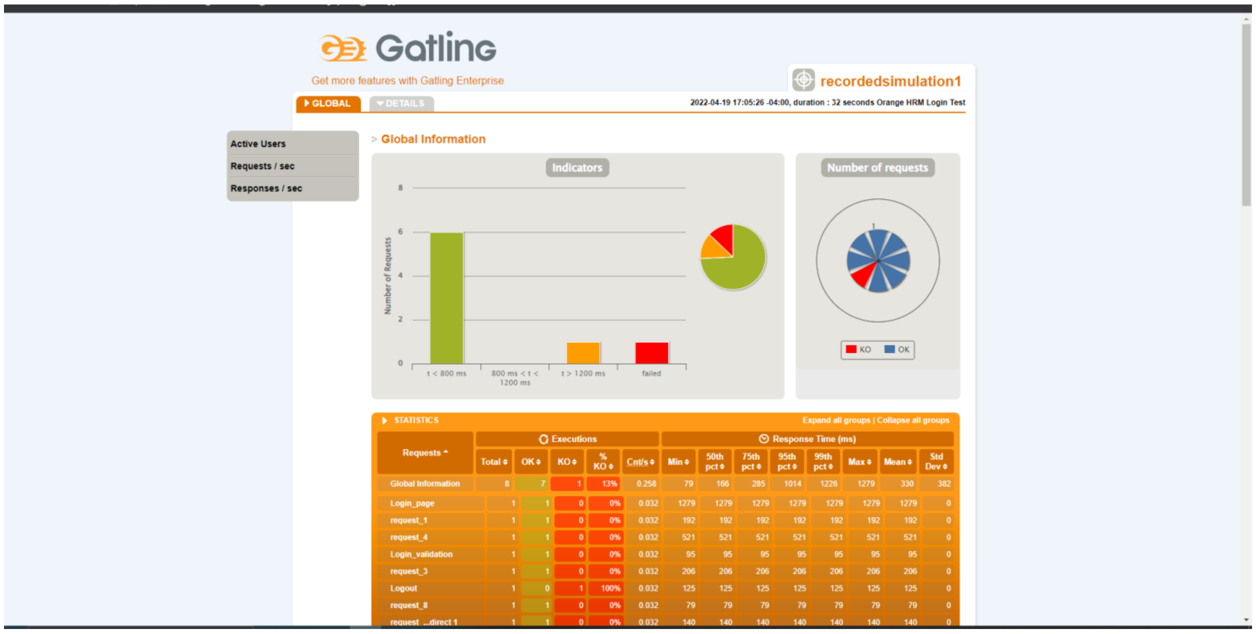
Import the HAR file in this gatling recorder and click the start button so it will generate a simulation script in the IDE.



```
package com.gatlig.test

import scala.concurrent.duration._

import io.gatling.core.Predef._
import io.gatling.http.Predef._
import io.gatling.jdbc.Predef._

class computedatabase extends Simulation {

  val httpProtocol = http
    .baseUrl( url = "https://computer-database.gatling.io")
    .inferHtmlResources(BlackList( patterns = """.*\.js""", """.*\.css""
    .acceptHeader( value = "text/html,application/xhtml+xml,application,
    .acceptEncodingHeader( value = "gzip, deflate")
    .acceptLanguageHeader( value = "en-US,en;q=0.9")
    .upgradeInsecureRequestsHeader( value = "1")
    .userAgentHeader( value = "Mozilla/5.0 (Windows NT 10.0; Win64; x64



  val scn = scenario( scenarioName = "computedatabase")
```

```
> t < 800 ms                                           0 (  0%)
> 800 ms < t < 1200 ms                                 0 (  0%)
> t > 1200 ms                                          0 (  0%)
> failed                                              10 (100%)
---- Errors --------------------------------------------------------------------
> jsonPath($.data.result).find.is(0), but actually found nothing     10 (100.0%)
================================================================================

Reports generated in 0s.
Please open the following file: C:\projects\Gatling\GatlingProject1\target\gatling\apitest1-20220420161459991\in

Process finished with exit code 0
```

Then Run the engine class and execute the report



2022-04-19 17:05:26 -04:00, duration : 32 seconds Orange HRM Login Test

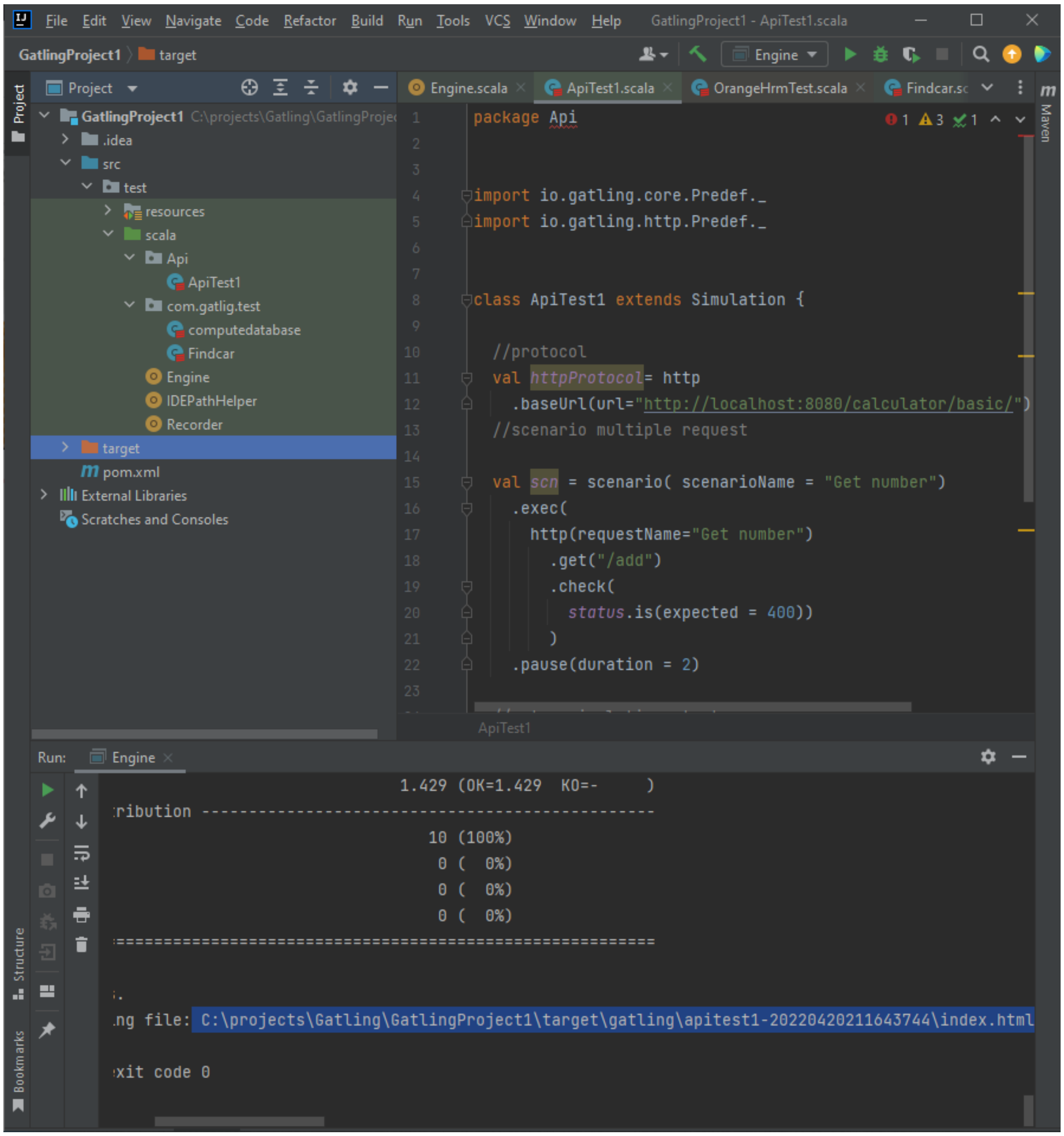
> Global Information

STATISTICS

Expand all groups | Collapse all groups

Requests	Executions					Response Time (ms)							
	Total	OK	KO	% KO	Cnt/s	Min	50th pct	75th pct	95th pct	99th pct	Max	Mean	Std Dev
Global Information	8	7	1	13%	0.258	79	166	285	1014	1226	1279	330	382
Login_page	1	1	0	0%	0.032	1279	1279	1279	1279	1279	1279	1279	0
request_1	1	1	0	0%	0.032	192	192	192	192	192	192	192	0
request_4	1	1	0	0%	0.032	521	521	521	521	521	521	521	0
Login_validation	1	1	0	0%	0.032	95	95	95	95	95	95	95	0
request_3	1	1	0	0%	0.032	206	206	206	206	206	206	206	0
Logout	1	0	1	100%	0.032	125	125	125	125	125	125	125	0
request_8	1	1	0	0%	0.032	79	79	79	79	79	79	79	0
request_...direct 1	1	1	0	0%	0.032	140	140	140	140	140	140	140	0

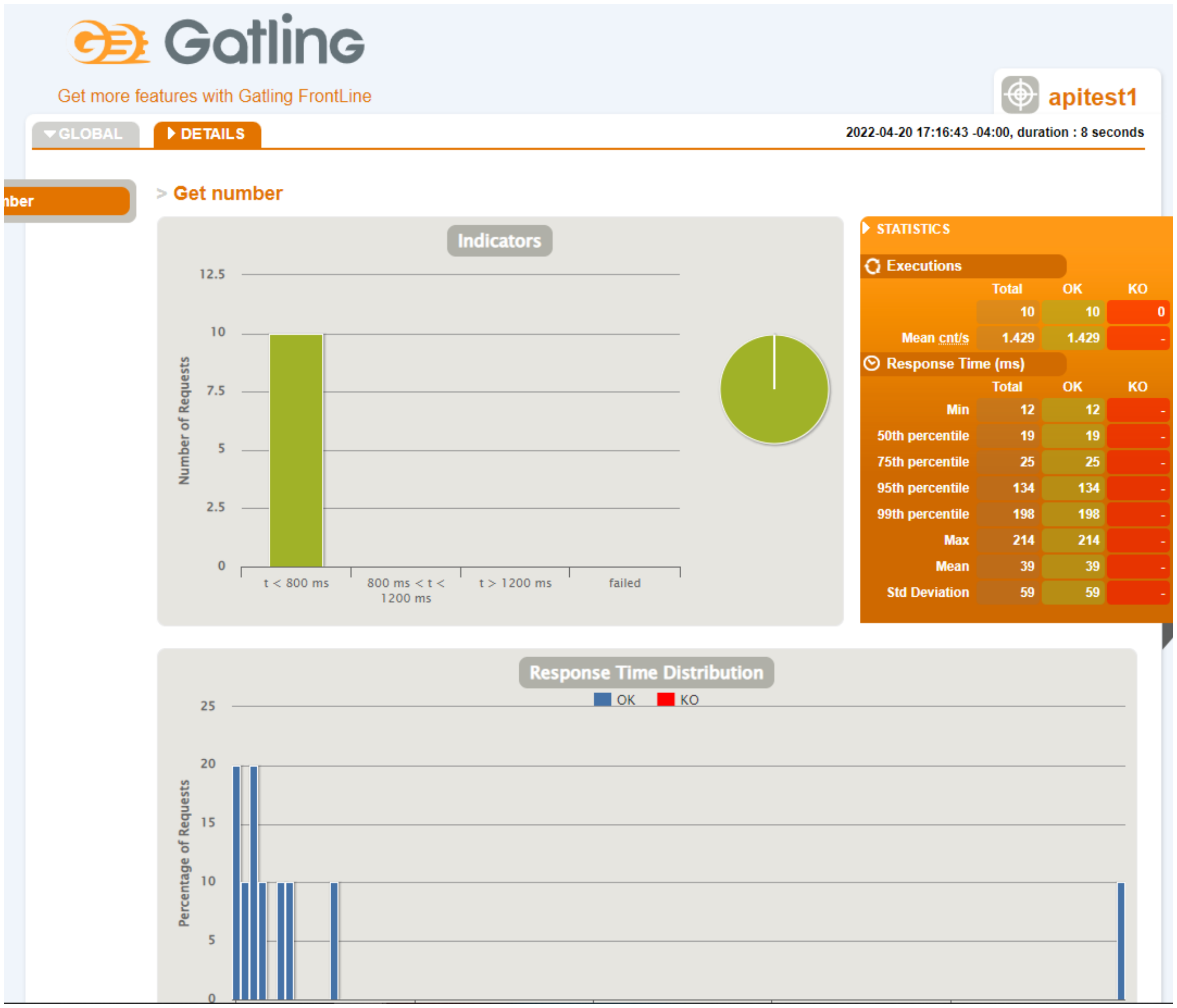
Then we can use various API method and scenarios,ramp up, protocol under this simulation script:



```scala
package Api


import io.gatling.core.Predef._
import io.gatling.http.Predef._


class ApiTest1 extends Simulation {

  //protocol
  val httpProtocol= http
    .baseUrl(url="http://localhost:8080/calculator/basic/")
  //scenario multiple request

  val scn = scenario( scenarioName = "Get number")
    .exec(
      http(requestName="Get number")
        .get("/add")
        .check(
          status.is(expected = 400))
        )
    .pause(duration = 2)
```

```
                              1.429 (OK=1.429  KO=-     )
:ribution ---------------------------------------------------
                                 10 (100%)
                                  0 (  0%)
                                  0 (  0%)
                                  0 (  0%)
==========================================================

:.
.ng file: C:\projects\Gatling\GatlingProject1\target\gatling\apitest1-20220420211643744\index.html

:xit code 0
```

After modifying the simulation Run the engine class and get report:



Get more features with Gatling FrontLine

apitest1

GLOBAL   DETAILS

2022-04-20 17:16:43 -04:00, duration : 8 seconds

nber

> Get number

STATISTICS

Executions

	Total	OK	KO
	10	10	0
Mean cnt/s	1.429	1.429	-

Response Time (ms)

	Total	OK	KO
Min	12	12	-
50th percentile	19	19	-
75th percentile	25	25	-
95th percentile	134	134	-
99th percentile	198	198	-
Max	214	214	-
Mean	39	39	-
Std Deviation	59	59	-