

Automation for Android

Calaba.sh

Continuous Integration

Calaba.sh



Jenkins



Build step for Android

1. Build with Gradle
2. Create APK file (Artifact)
3. UI testing with Calabash
4. Generate report

Build step for Android

1. Build with Gradle

- Unit test
- Android Unit test
- Espresso

2. Create APK file (Artifact)

3. UI testing with Calabash

4. Generate report

1. Build with Gradle

```
$ ./gradlew assembleDebug
```

```
$ ./gradlew assembleRelease
```

1. Build with Gradle

\$./gradlew assembleRelease

```
signingConfigs {  
    release {  
        storeFile file("keystores/release.keystore")  
        storePassword "your-keystore-password"  
        keyAlias "your-alias"  
        keyPassword "your-alias-password"  
    }  
}  
  
buildTypes {  
    release {  
        signingConfig signingConfigs.release  
    }  
}
```

2. Create APK file

In app/build/outputs/apk

```
app/build/outputs/apk/  
├─ app-debug-unaligned.apk  
└─ app-debug.apk
```

3. Run calabash

Copy APK file to Calabash project

\$calabash-android run <APK>

4. Generate report

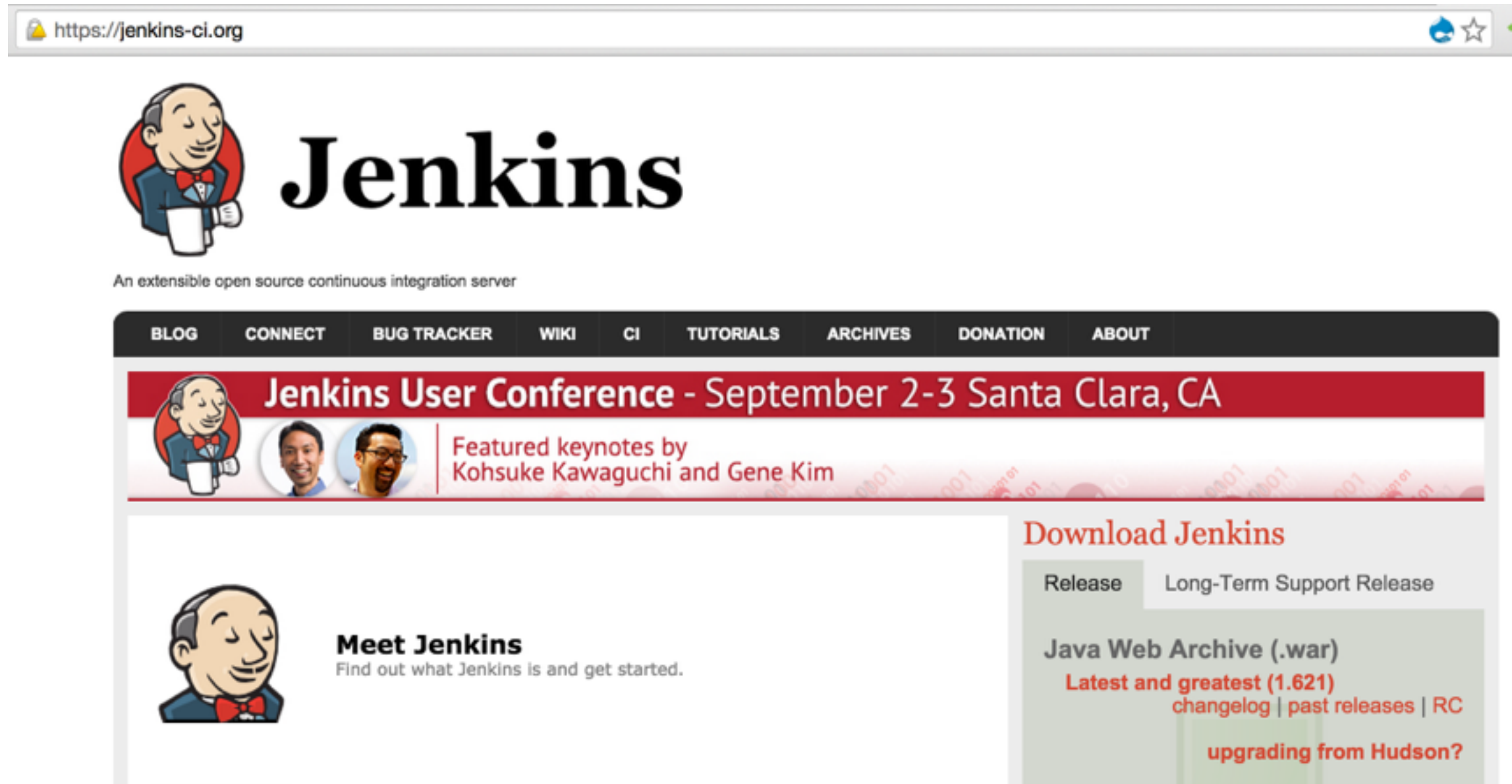
\$scalabash-android run <APK>

--format json --out report.json --format pretty

We need build script


```
export ANDROID_HOME=<ANDROID SDK>
./gradlew assembleDebug
cd calabash
cp ../app/build/outputs/apk/<APK file> .
calabash-android resign <APK file>
calabash-android run <APK file> --format json --out report.json
```

Start with Jenkins



The screenshot shows the Jenkins website homepage. At the top, the browser address bar displays "https://jenkins-ci.org". The main header features the Jenkins logo (a cartoon man in a suit) and the word "Jenkins" in a large, bold, serif font. Below the logo, a tagline reads "An extensible open source continuous integration server". A dark navigation bar contains links: "BLOG", "CONNECT", "BUG TRACKER", "WIKI", "CI", "TUTORIALS", "ARCHIVES", "DONATION", and "ABOUT". A prominent red banner announces the "Jenkins User Conference - September 2-3 Santa Clara, CA", featuring keynotes by Kohsuke Kawaguchi and Gene Kim. Below the banner, the left sidebar has a "Meet Jenkins" section with the tagline "Find out what Jenkins is and get started." The right sidebar, titled "Download Jenkins", offers two tabs: "Release" (selected) and "Long-Term Support Release". Under the "Release" tab, it lists "Java Web Archive (.war)" with the "Latest and greatest (1.621)" version, including links for "changelog", "past releases", and "RC". A link for "upgrading from Hudson?" is also present.

https://jenkins-ci.org




Jenkins

An extensible open source continuous integration server

[BLOG](#) [CONNECT](#) [BUG TRACKER](#) [WIKI](#) [CI](#) [TUTORIALS](#) [ARCHIVES](#) [DONATION](#) [ABOUT](#)

Jenkins User Conference - September 2-3 Santa Clara, CA

Featured keynotes by Kohsuke Kawaguchi and Gene Kim



Meet Jenkins

Find out what Jenkins is and get started.

Download Jenkins

Release Long-Term Support Release

Java Web Archive (.war)

Latest and greatest (1.621)
changelog | past releases | RC

upgrading from Hudson?

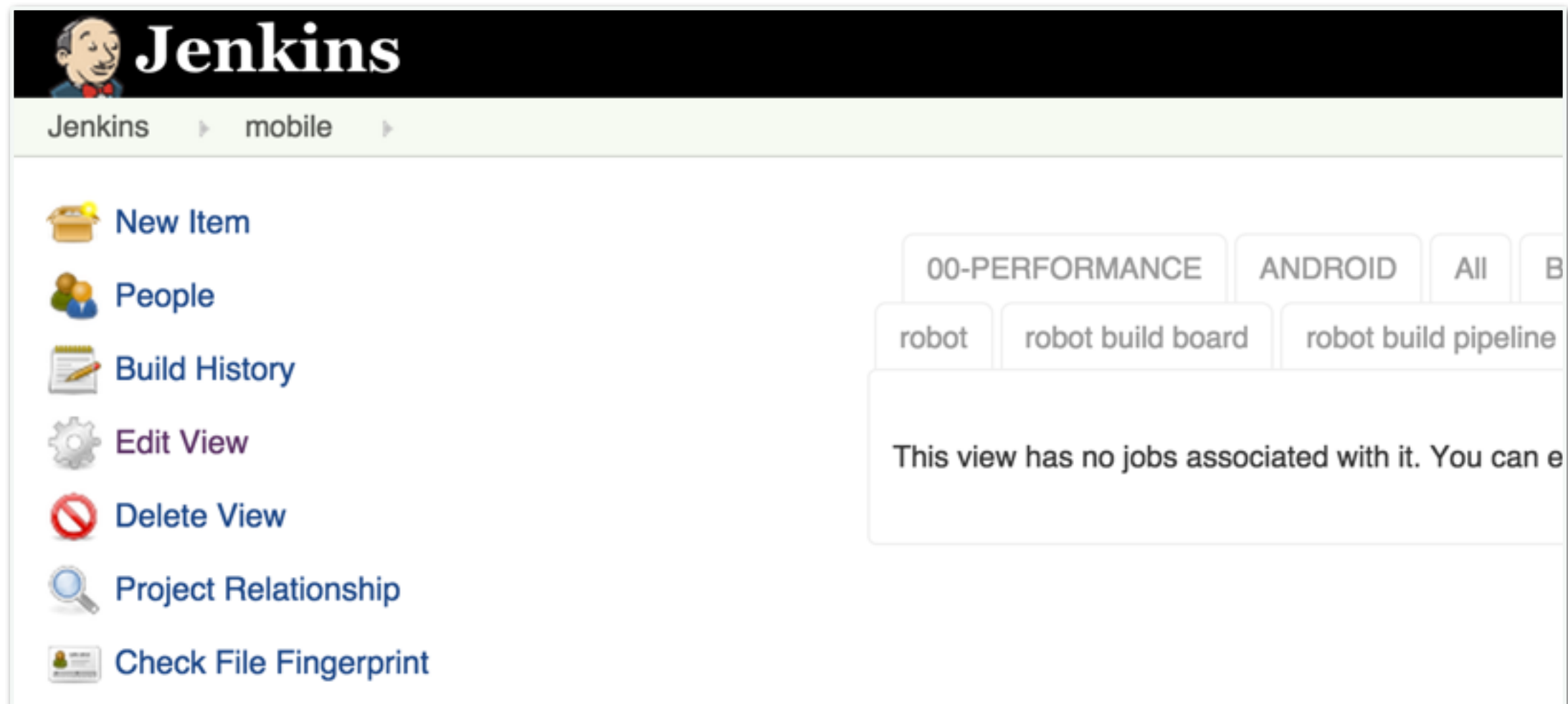
Start Jenkins server

```
$java -jar jenkins.war
```

INFO: Jenkins is fully up and running

Start Jenkins server

<http://localhost:8080>



New job

Item name

01-BUILD-AND-TESTING

☒ **Freestyle project**

This is the central feature of Jenkins. Jenkins will build your project, combining any SCM with any build system for something other than software build.

☐ **Maven project**

Build a maven project. Jenkins takes advantage of your POM files and drastically reduces the configuration.

☐ **External Job**

This type of job allows you to record the execution of a process run outside Jenkins, even on a remote machine; you can use Jenkins as a dashboard of your existing automation system. See [the documentation for more details](#).

☐ **Multi-configuration project**

Suitable for projects that need a large number of different configurations, such as testing on multiple environments etc.

☐ **Copy existing Item**

Copy from

OK

Source code management

Source Code Management

☐ None

☐ CVS

☐ CVS Projectset

☒ Git


Repositories

Repository URL

git@github.com:up1/android_mockapi_exmample.git

Credentials

- none -

 Add

Auto build in every minute

Build Triggers

☐ Build after other projects are built


☐ Build periodically

☒ Poll SCM

Schedule

Add build process

Build

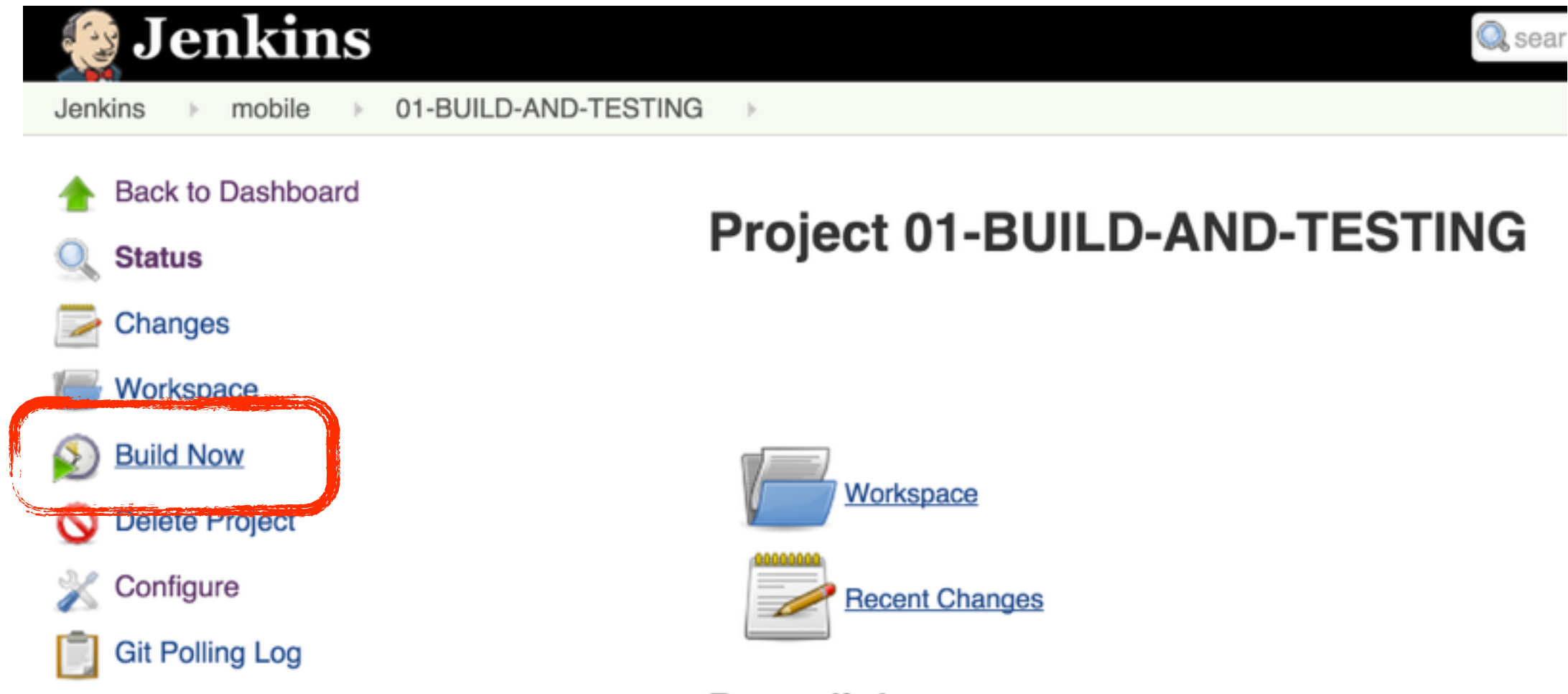
 **Execute shell**

Command

`sh android.sh`

[See the list of available environment variables](#)

Manual build



The image shows the Jenkins web interface for a project named "01-BUILD-AND-TESTING". The top navigation bar includes the Jenkins logo, a search bar, and a breadcrumb trail: "Jenkins > mobile > 01-BUILD-AND-TESTING >". On the left sidebar, a list of actions is provided: "Back to Dashboard", "Status", "Changes", "Workspace", "Build Now" (highlighted with a red rounded rectangle), "Delete Project", "Configure", and "Git Polling Log". The main content area is titled "Project 01-BUILD-AND-TESTING" and contains links for "Workspace" and "Recent Changes", each accompanied by an icon.

Jenkins

Jenkins > mobile > 01-BUILD-AND-TESTING >

Back to Dashboard

Status

Changes

Workspace

Build Now

Delete Project

Configure

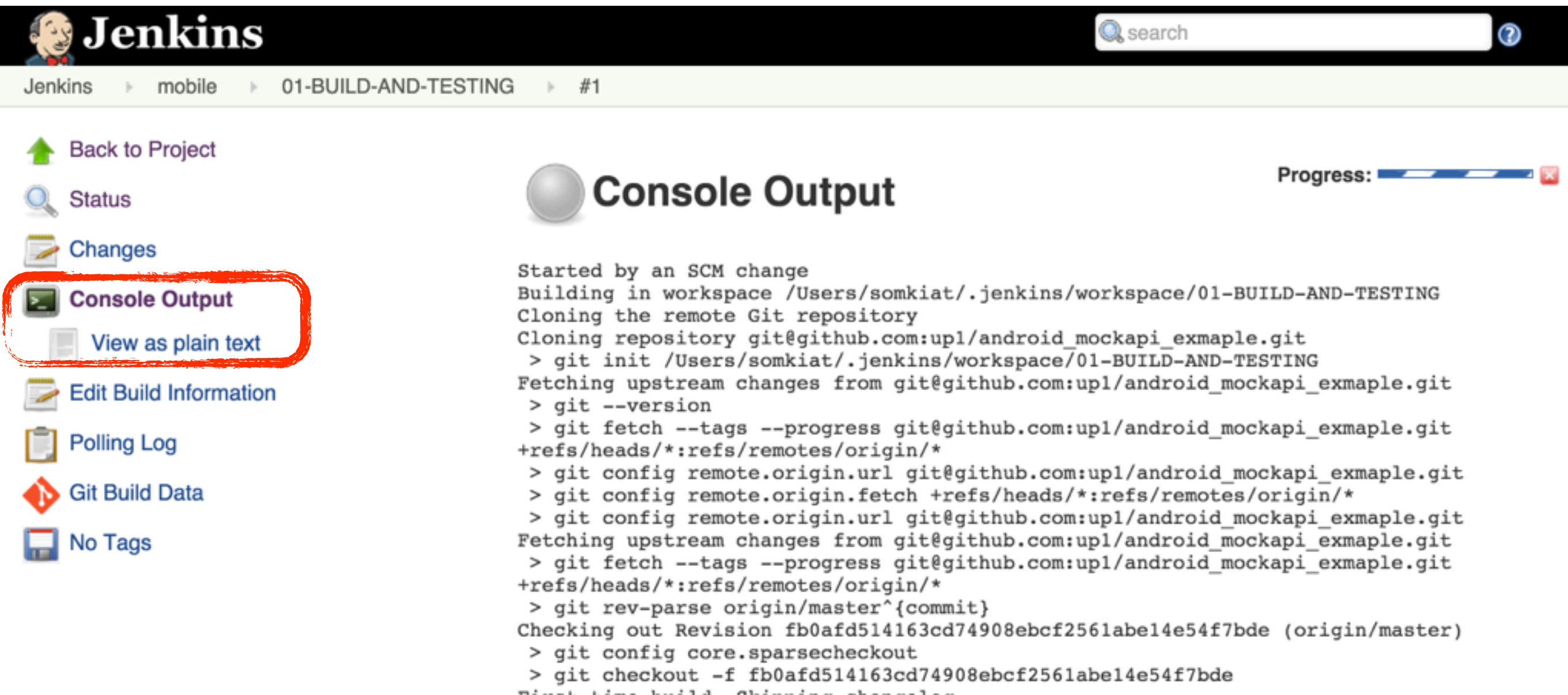
Git Polling Log

Project 01-BUILD-AND-TESTING

Workspace

Recent Changes

See result



The image is a screenshot of the Jenkins web interface. At the top, the Jenkins logo is on the left, and a search bar is on the right. Below the logo, the breadcrumb navigation shows 'Jenkins' > 'mobile' > '01-BUILD-AND-TESTING' > '#1'. On the left sidebar, there are several links: 'Back to Project' (with a green arrow icon), 'Status' (with a magnifying glass icon), 'Changes' (with a document icon), 'Console Output' (with a terminal icon and highlighted by a red rounded rectangle), 'View as plain text' (with a document icon), 'Edit Build Information' (with a pencil icon), 'Polling Log' (with a clipboard icon), 'Git Build Data' (with a Git logo icon), and 'No Tags' (with a folder icon). The main content area is titled 'Console Output' with a large grey sphere icon. To the right of the title is a 'Progress' bar. The console output text shows the following commands and their results:

```
Started by an SCM change
Building in workspace /Users/somkiat/.jenkins/workspace/01-BUILD-AND-TESTING
Cloning the remote Git repository
Cloning repository git@github.com:upl/android_mockapi_exmaple.git
> git init /Users/somkiat/.jenkins/workspace/01-BUILD-AND-TESTING
Fetching upstream changes from git@github.com:upl/android_mockapi_exmaple.git
> git --version
> git fetch --tags --progress git@github.com:upl/android_mockapi_exmaple.git
+refs/heads/*:refs/remotes/origin/*
> git config remote.origin.url git@github.com:upl/android_mockapi_exmaple.git
> git config remote.origin.fetch +refs/heads/*:refs/remotes/origin/*
> git config remote.origin.url git@github.com:upl/android_mockapi_exmaple.git
Fetching upstream changes from git@github.com:upl/android_mockapi_exmaple.git
> git fetch --tags --progress git@github.com:upl/android_mockapi_exmaple.git
+refs/heads/*:refs/remotes/origin/*
> git rev-parse origin/master^{commit}
Checking out Revision fb0afd514163cd74908ebcf2561abel14e54f7bde (origin/master)
> git config core.sparsecheckout
> git checkout -f fb0afd514163cd74908ebcf2561abel14e54f7bde
```

Publish test report

Publish Robot Framework test results

Publish cucumber results as a report

Publish xUnit test result report

Record JaCoCo coverage report

Record fingerprints of files to track usage

Run Tests on AppThwack

Upload to HockeyApp


▼

Post-build Actions

Add post-build action ▼

Publish test report

Post-build Actions

 **Publish cucumber results as a report**

Json Reports Path

The path relative to the workspace of the json reports generated by cucumber-jvm e.g. target - leave empty to let the plugin find them automagically

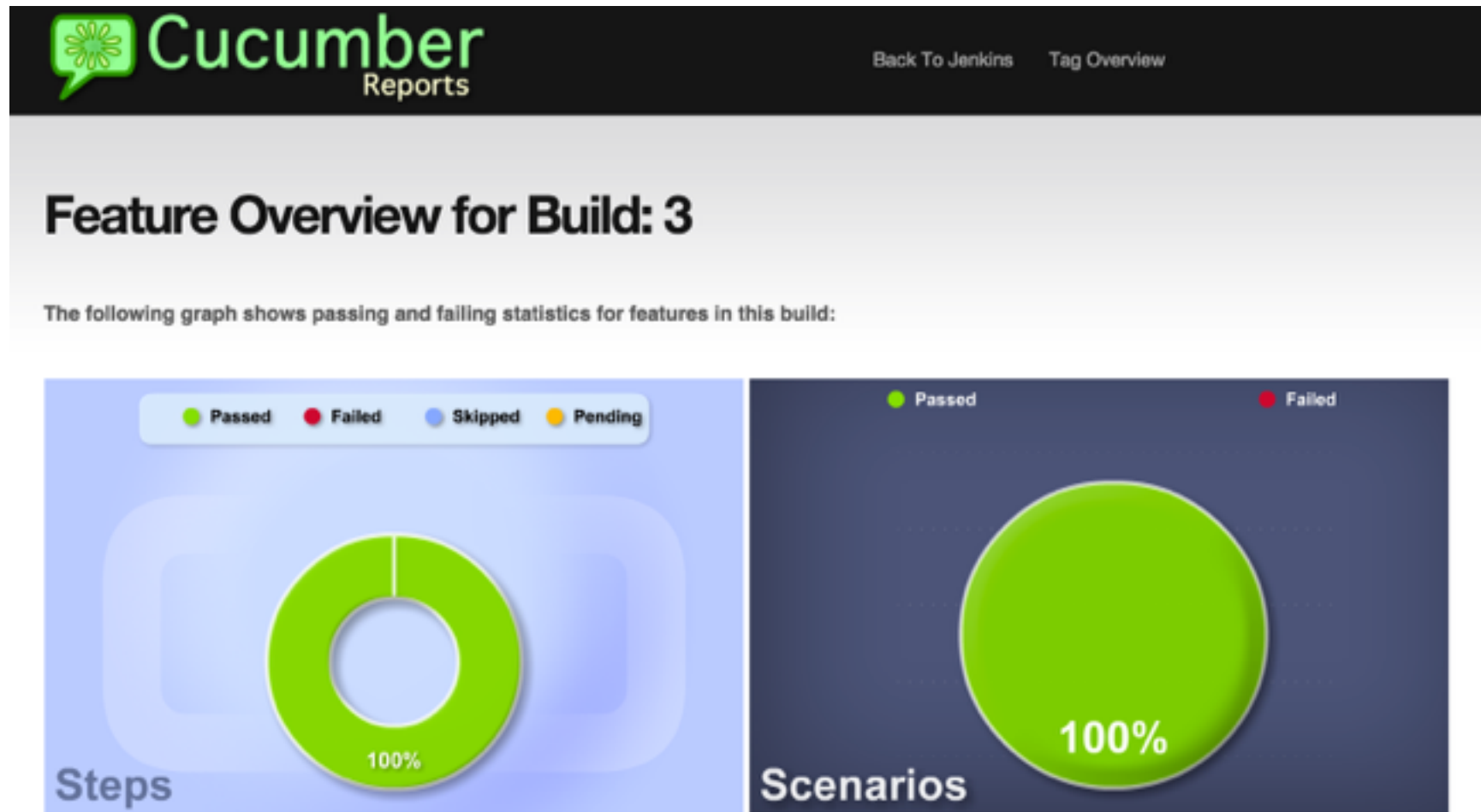
Plugin Url Path

The path to the jenkins user content url e.g. http://host:port[/jenkins/]plugin - leave empty if jenkins url root is host:port

Skipped Steps Fail the Build ☐

Tick this if you want skipped steps to cause the build to fail

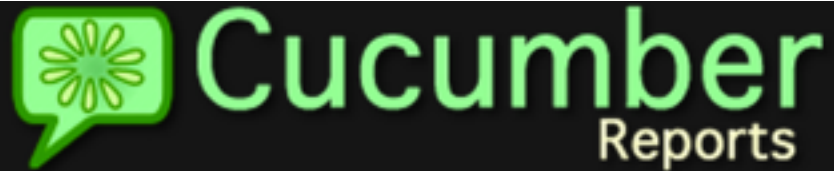
Cucumber report



Feature Statistics

Feature	Scenarios			Steps					Duration	Status
	Total	Passed	Failed	Total	Passed	Failed	Skipped	Pending		
First feature	1	1	0	1	1	0	0	0	84 ms	passed
1	1	1	0	1	1	0	0	0	84 ms	Totals

Cucumber report detail

[Back To Jenkins](#)[Tag Overview](#)[Feature Overview](#)

Feature Result for Build: 3

	Scenarios			Steps						
Feature	Total	Passed	Failed	Total	Passed	Failed	Skipped	Pending	Duration	Status
First feature	1	1	0	1	1	0	0	0	84 ms	passed

Feature: First feature

Scenario: Welcome to main page

Then I see "Angular"

84 ms