CLEAN CODE BY CONVENTION?!

AN EXPERIENCE REPORT BY THE EXAMPLE OF ANDROID LINT.

ANDRÉ DIERMANN

Software Architect @ it-objects

AGENDA

- Introduction
- Android Lint
- Clean Code
  by Convention?

INTRODUCTION

- Motivation
- Clean Code
- Android Lint

OUR MOTIVATION

CODE QUALITY FLAWS DUE TO

- heterogenous team
- working distributed over Europe
- extreme agility
- framework limits
- huge code base

OUR MOTIVATION

HOW TO MAKE THE CODE CLEAN(ER)?

- harmonize the understanding
  → common conventions
- provide automated support
  → Static Code Analysis

CLEAN CODE

- no unique definition
- "School of Thought"

CLEAN CODE

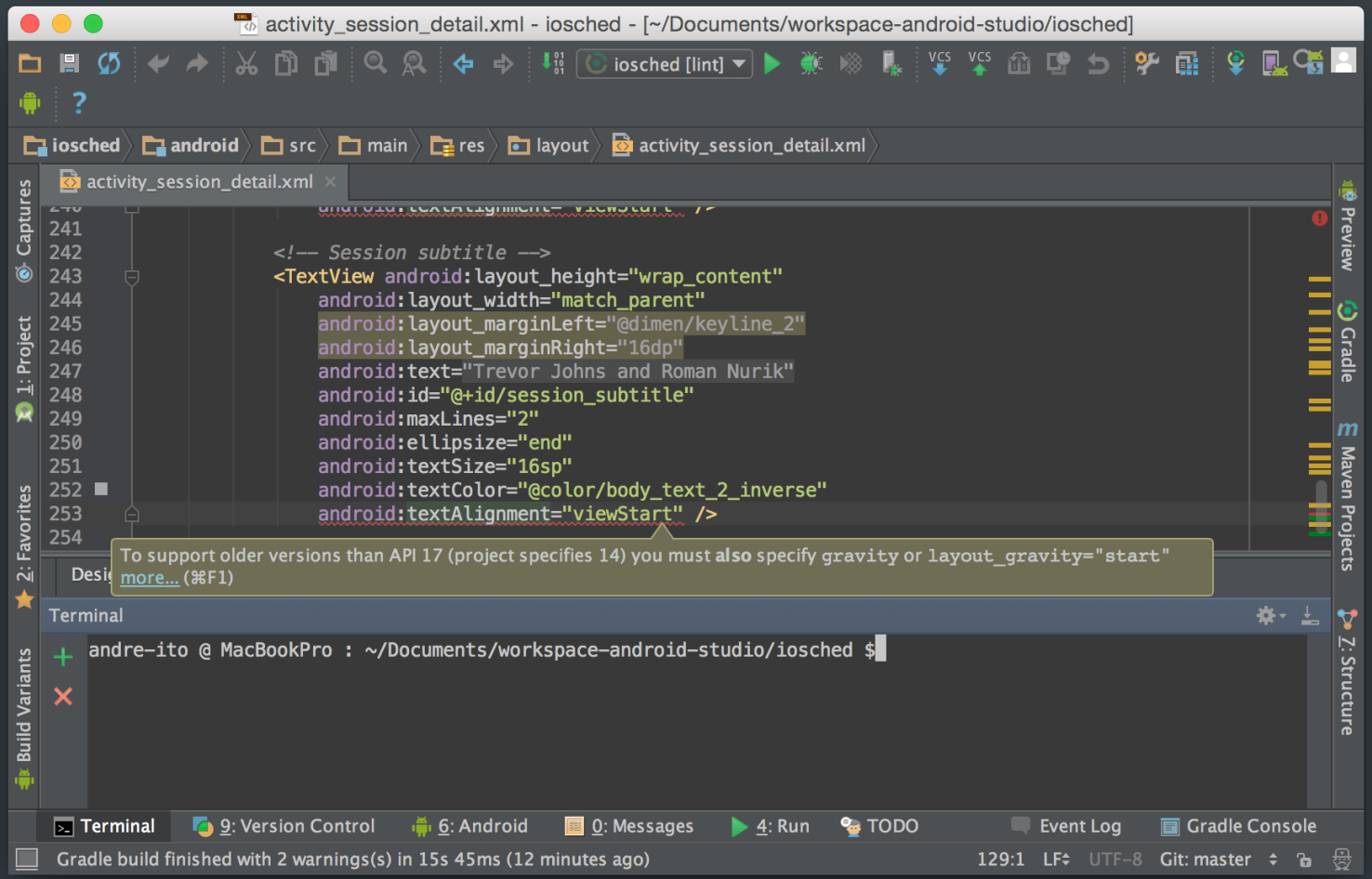
[...] Clean code always looks like it was written by someone who cares. [...]

Michael Feathers

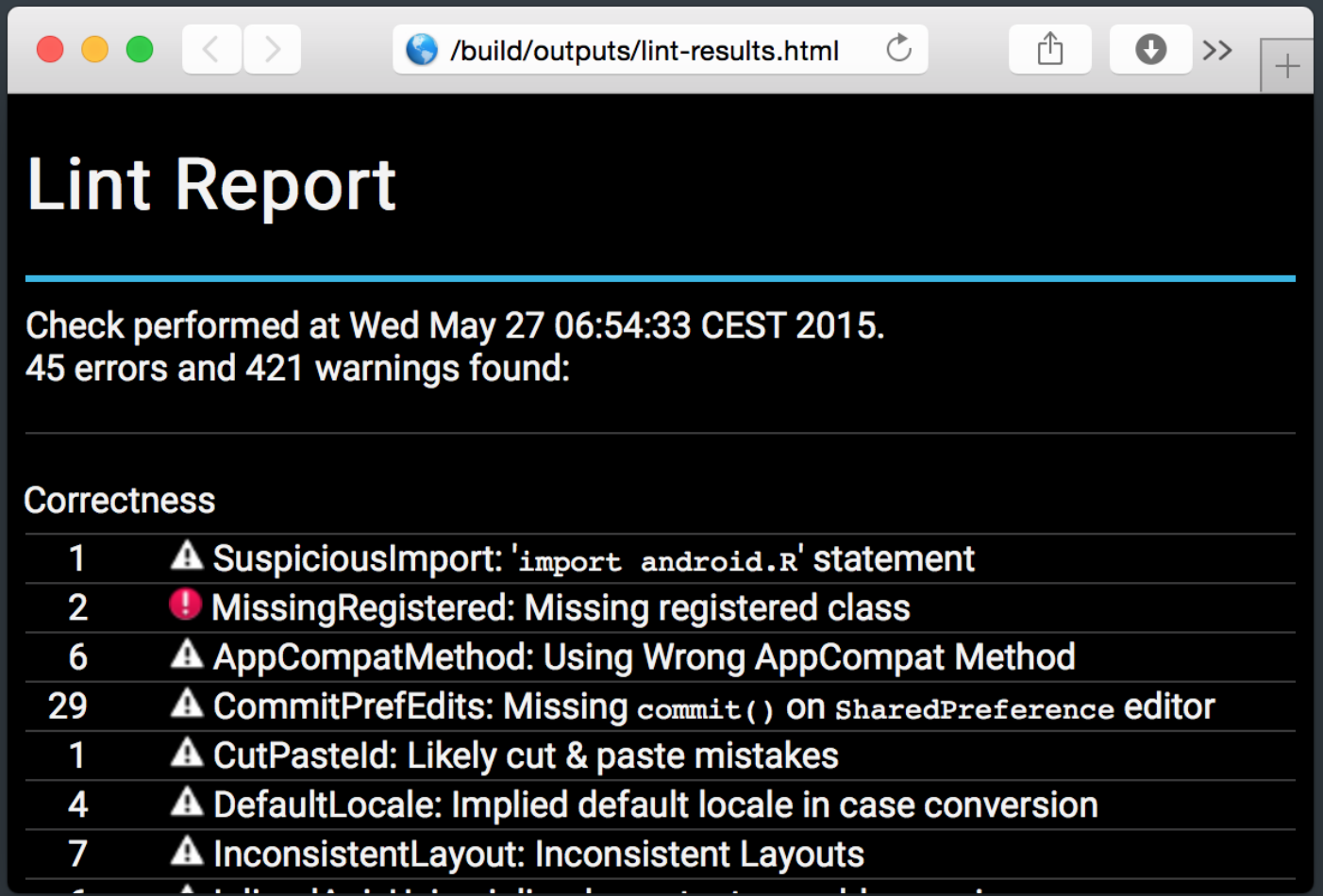
ANDROID LINT

- tool for command-line and IDE
- scans all kind of development artifacts
- reports potential bugs, bad coding habits, broken conventions, ...
- features more than 200 built-in checks (October 2015)

EXAMPLE


activity_session_detail.xml - iosched - [~/Documents/workspace-android-studio/iosched]
iosched [lint]
iosched
android
src
main
res
layout
activity_session_detail.xml
activity_session_detail.xml
241
242 <!-- Session subtitle -->
243 <TextView android:layout_height="wrap_content"
244 android:layout_width="match_parent"
245 android:layout_marginLeft="@dimen/keyline_2"
246 android:layout_marginRight="16dp"
247 android:text="Trevor Johns and Roman Nurik"
248 android:id="@+id/session_subtitle"
249 android:maxLines="2"
250 android:ellipsize="end"
251 android:textSize="16sp"
252 android:textColor="@color/body_text_2_inverse"
253 android:textAlignment="viewStart" />
254
To support older versions than API 17 (project specifies 14) you must also specify gravity or layout_gravity="start"
more... (⌘F1)
Terminal
andre-ito @ MacBookPro : ~/Documents/workspace-android-studio/iosched $
Terminal
9: Version Control
6: Android
0: Messages
4: Run
TODO
Event Log
Gradle Console
Gradle build finished with 2 warnings(s) in 15s 45ms (12 minutes ago)
129:1
LF
UTF-8
Git: master
1: Project
2: Favorites
Build Variants
Captures
Preview
Gradle
Maven Projects
Z: Structure

EXAMPLE

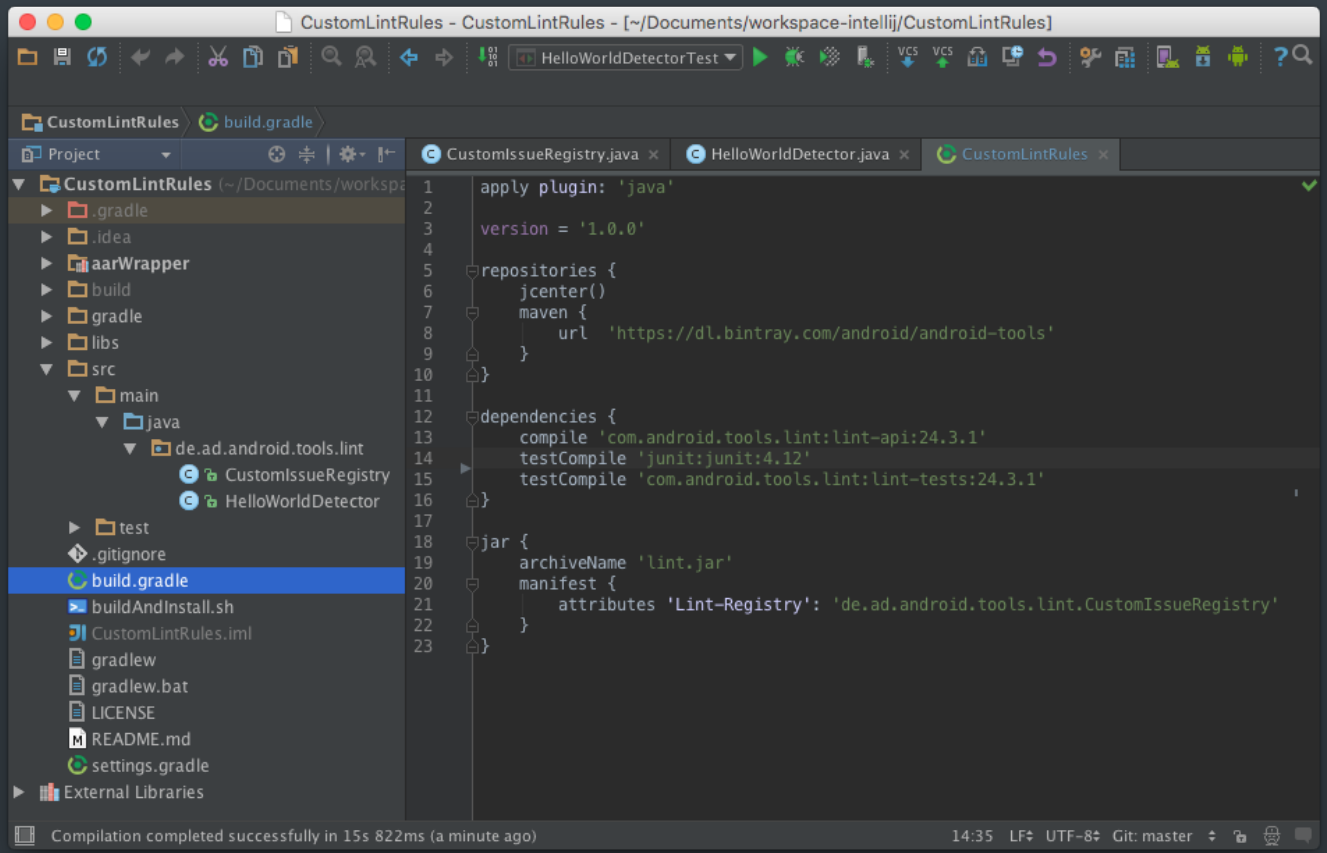

/build/outputs/lint-results.html
Lint Report
Check performed at Wed May 27 06:54:33 CEST 2015.
45 errors and 421 warnings found:
Correctness
1 SuspiciousImport: 'import android.R' statement
2 MissingRegistered: Missing registered class
6 AppCompatMethod: Using Wrong AppCompat Method
29 CommitPrefEdits: Missing commit() on SharedPreference editor
1 CutPasteId: Likely cut & paste mistakes
4 DefaultLocale: Implied default locale in case conversion
7 InconsistentLayout: Inconsistent Layouts

ANDROID LINT

- Creation + Verification
- Operation + Continuous Integration
- Real world examples

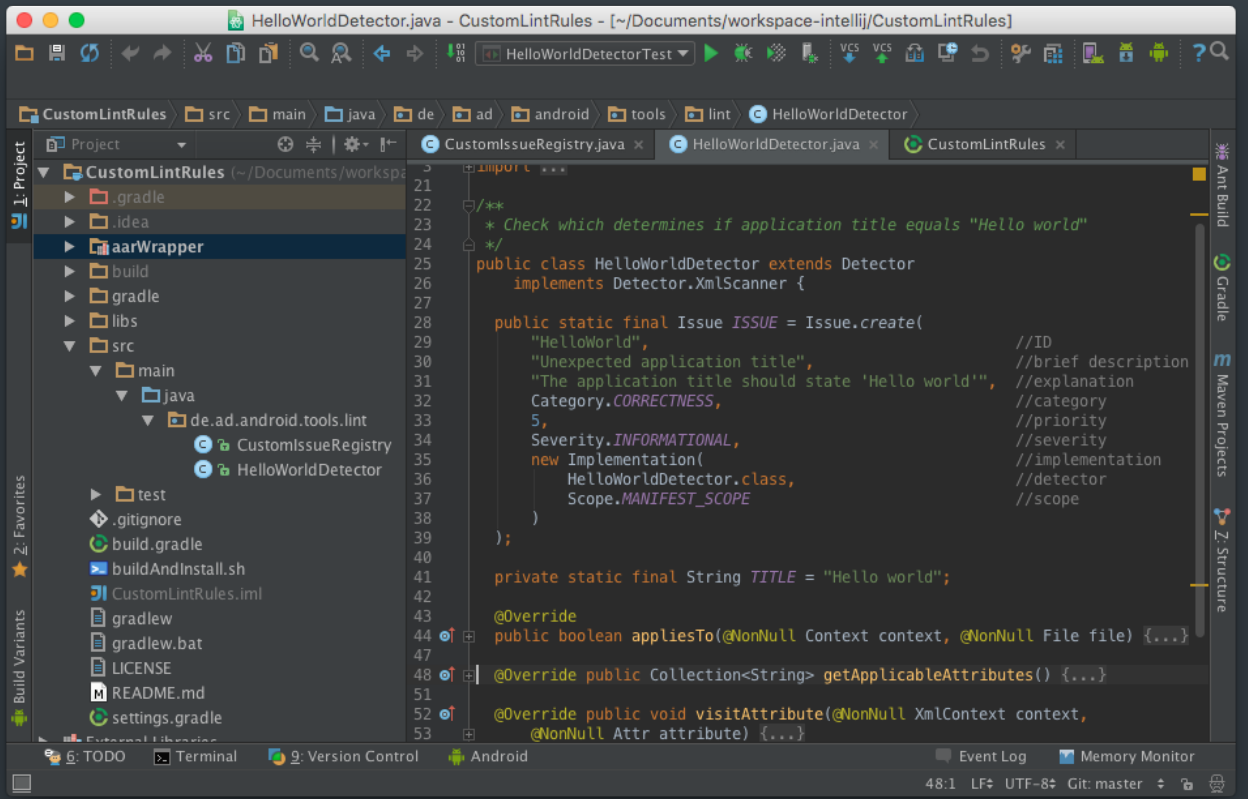
CREATION

https://github.com/a11n/CustomLintRules



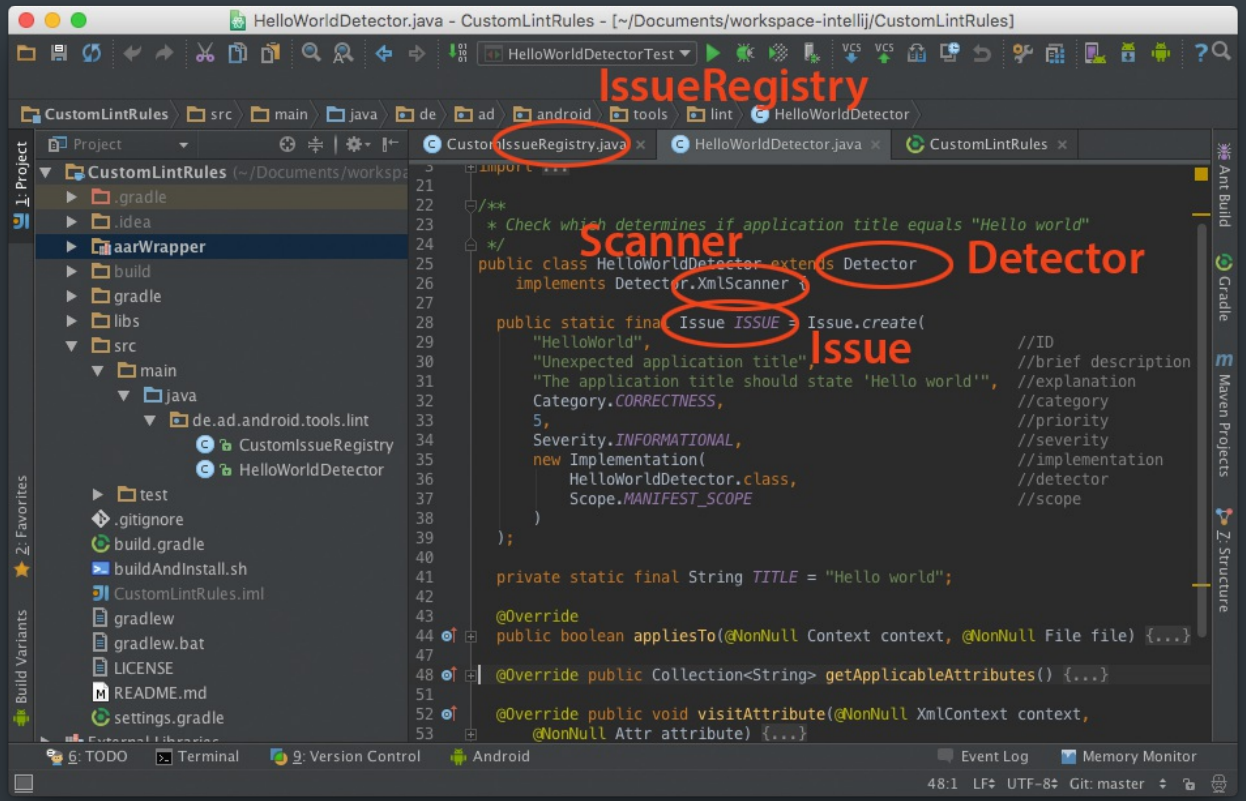
CREATION

CORE PRINCIPLES



CREATION

CORE PRINCIPLES



CREATION

CORE PRINCIPLES OF THE LINT API

- `Issue`
- `Detector`
- `Scanner`
- `IssueRegistry`

Reference guide:
https://github.com/a11n/android-lint

VERIFICATION

TEST

```
public class HelloWorldDetectorTest extends LintDetectorTest {
  //Specify the detector under test
  @Override protected Detector getDetector() { ... }

  //Specify the issues to report
  @Override protected List<Issue> getIssues() { ... }

  //Perform test
  public void test() throws Exception {
    //assert that linting a given set of files
    //returns the expected output
    assertEquals(EXPECTED_OUTPUT, lintFiles(FILES));
  }
}
```

VERIFICATION

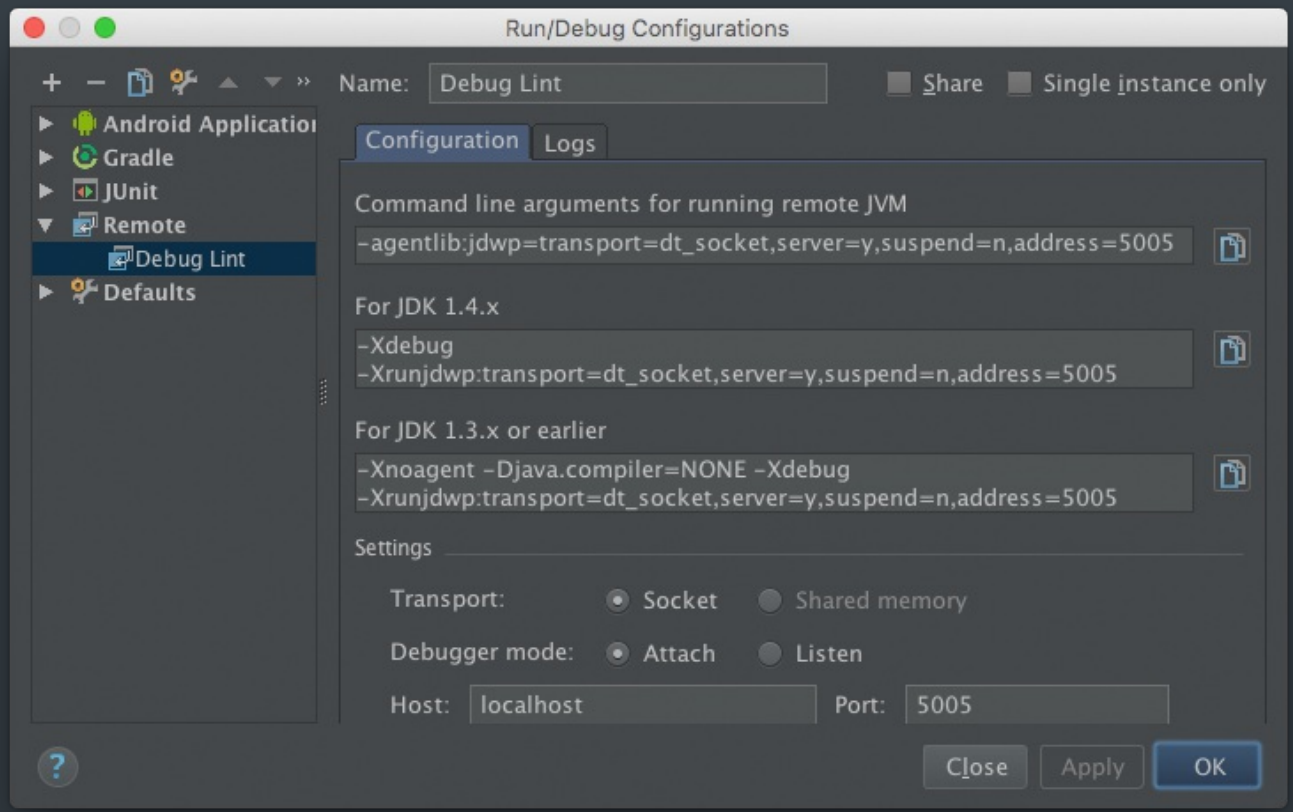
ASSERT

```
public void testShouldDetectWarning() throws Exception {
  assertEquals(
    "AndroidManifest.xml:8: Information: Unexpected title \"@strin
  + "        android:label=\"@string/app_name\"\n"
  + "        ~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~\n"
  + "0 errors, 1 warnings\n",
    lintFiles("InvalidAndroidManifest.xml=>AndroidManifest.xml"));
}
```

JUnit4 & better assertions:
https://github.com/a11n/lint-junit-rule

VERIFICATION

DEBUG



OPERATION

CONFIGURATION

- `build.gradle`

```
android {
  lintOptions { ... }
  }
```

- `lint.xml`

```
<lint>
  <issue id="IconMissingDensityFolder" severity="ignor
</lint>
```

- Java / Resource

```
@SuppressLint("TheIssueYouWantToSuppress")
```

```
tools:ignore="TheIssueYouWantToSuppress"
```

OPERATION

APPLICATION

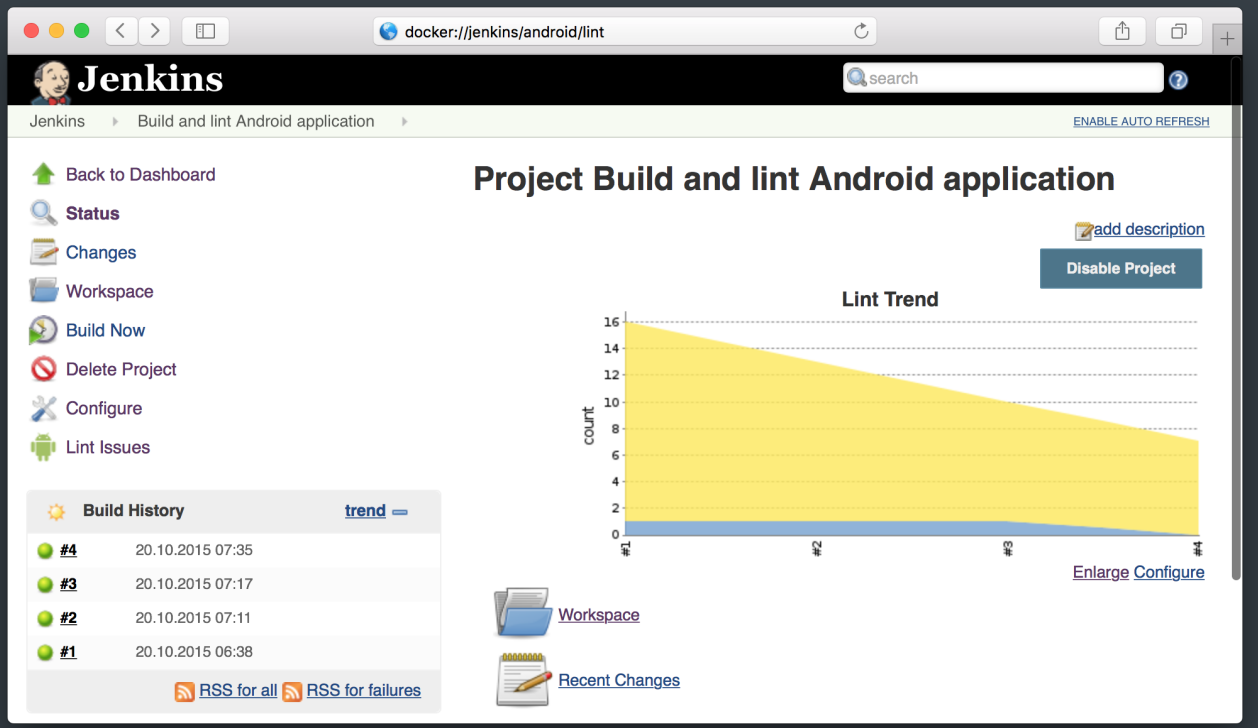
- Have a Java module for the Lint rules and an Android library module as wrapper

```
Android application project
--app          //default Android application module
--lint         //Android library, acts as wrapper for the Lint ru
--lintrules    //Java module with your custom Lint rules
```

```
project.afterEvaluate {
def compileLint = project.tasks.getByPath(':lint:compileLint')
compileLint.dependsOn ':lintrules:jar'
compileLint << {
  copy{
    from '../lintrules/build/libs'
    into 'build/intermediates/lint'
  }
}
}
```

CONTINOUS INTEGRATION

https://github.com/a11n/docker-jenkins-android-lint



REAL WORLD EXAMPLES

- Naming conventions
- Logging
- Smell detection

CLEAN CODE BY CONVENTION?

- Challenges
- Conventions
- Chances + Limits
- Discussion

CHALLENGES

INTRODUCING CLEAN CODE

- Where to start?
- How to convince team members?
- How to support team members?
- How to evolve team culture?
- ...

CONVENTIONS

Conventions are agreements on how to do certain things.

Synonyms: Guidelines, Principles, Best Practices, ...

CONVENTIONS

RELATION TO CLEAN CODE

- "Conventions are part of Clean Code."
- "Clean Code is expressed by conventions."
- "Clean Code is a set of conventions."

CHANCES

- easy to understand, communicate and validate
- trigger awareness for Clean Code

CHANCES

CLEAN CODE

$\Rightarrow$

CONVENTIONS

LIMITS

CONVENTIONS

≠

CLEAN CODE

DISCUSSION

- Do you have dedicated conventions in your team, project or company?
- How do you explain/teach clean code to a Junior Developer?

THANK YOU FOR YOUR ATTENTION.

Q&A

q2ad a11n