CUSTOM LINT RULES

A JOURNEY TOWARDS CLEANER CODE

ANDRÉ DIERMANN

Software Architect @ it-objects

AGENDA

- Introduction
- Creation
- Verification
- Operation
- Q & A

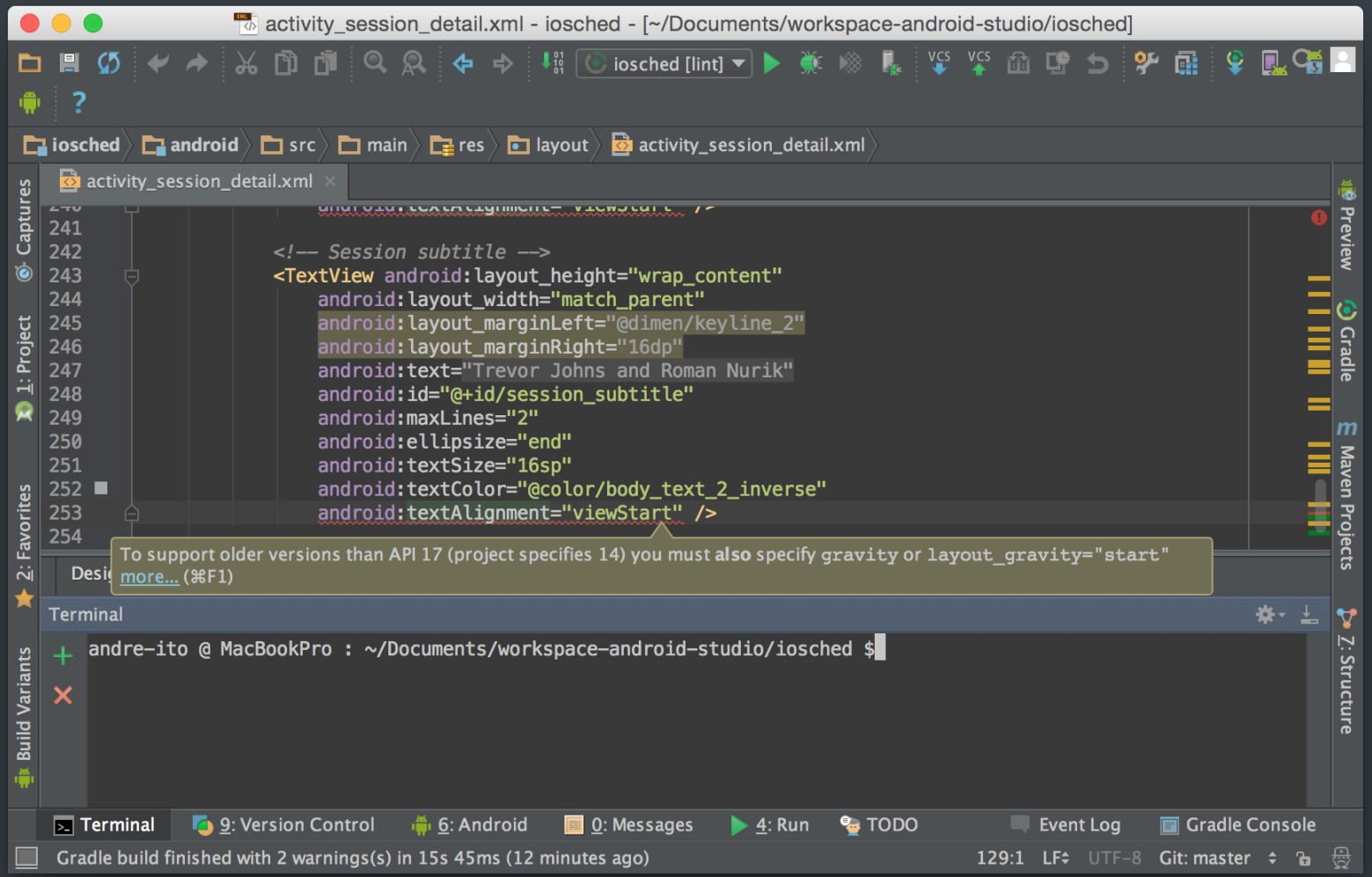
INTRODUCTION

- Lint
- Custom rules

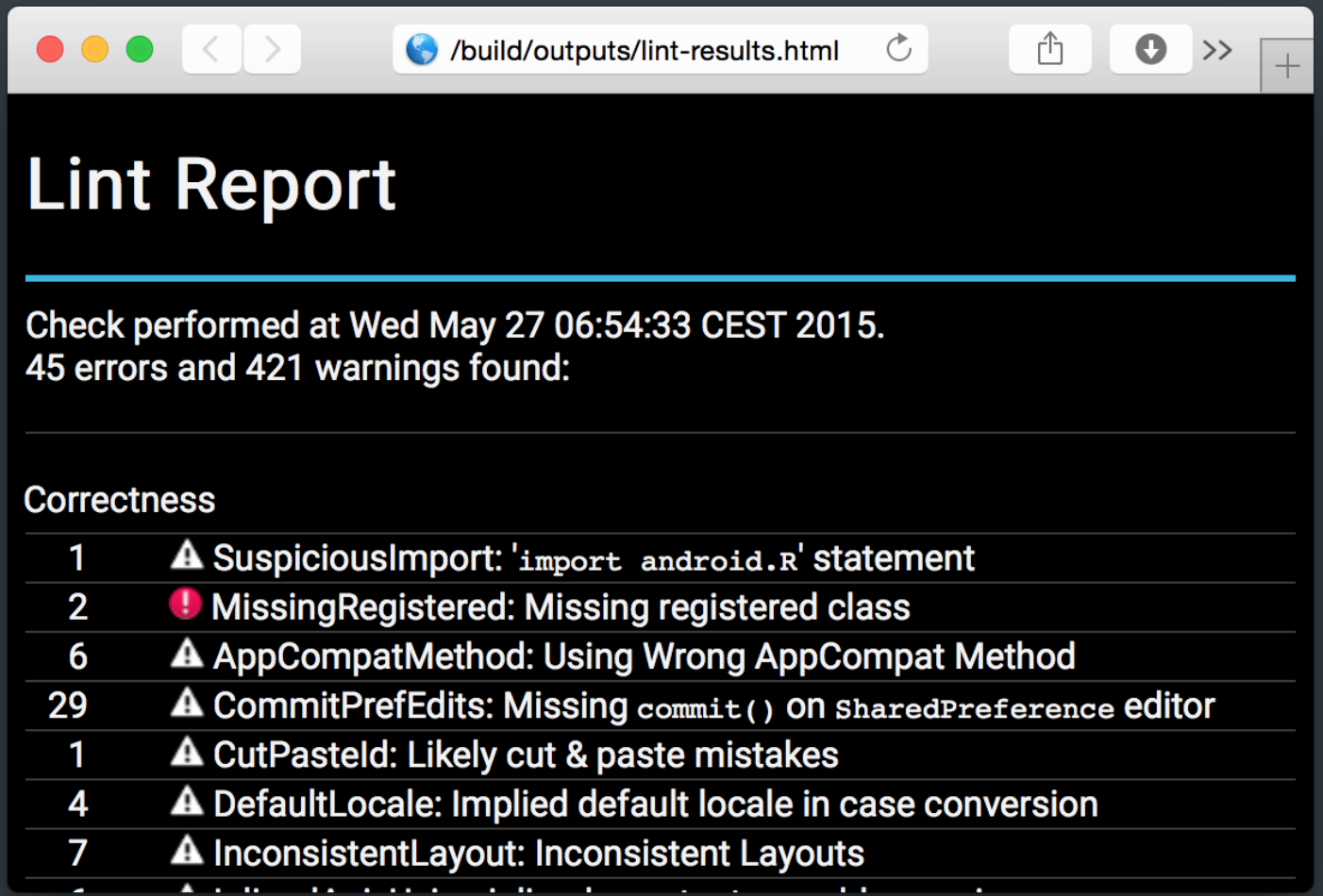
LINT

- 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
Captures
1: Project
2: Favorites
Build Variants
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

CUSTOM RULES

MOTIVATION

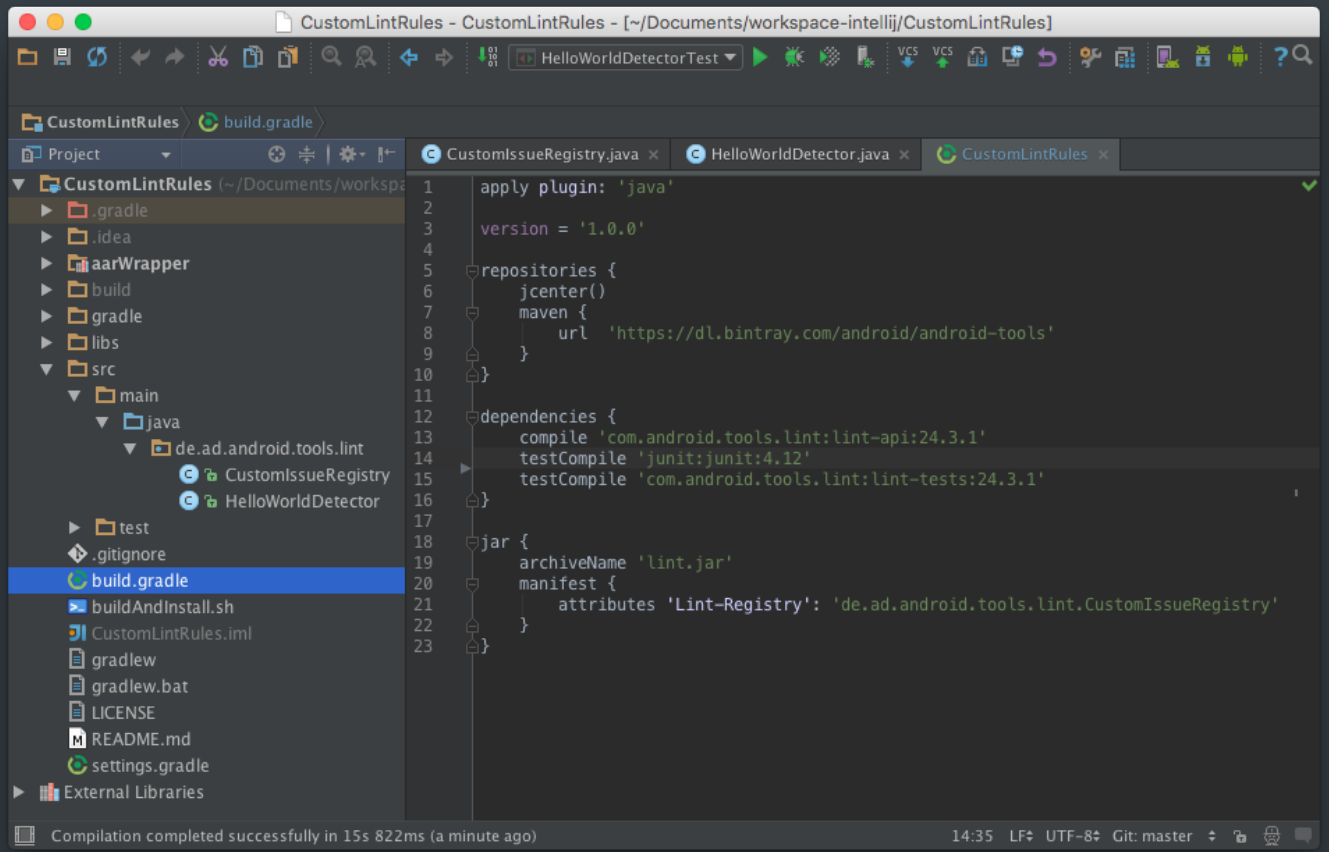
- work in large and distributed teams requires dedicated conventions
- huge code bases require automated checks
- need for 'Android specific' validations (vs. checkstyle, FindBugs, ...)
- ...

CREATION

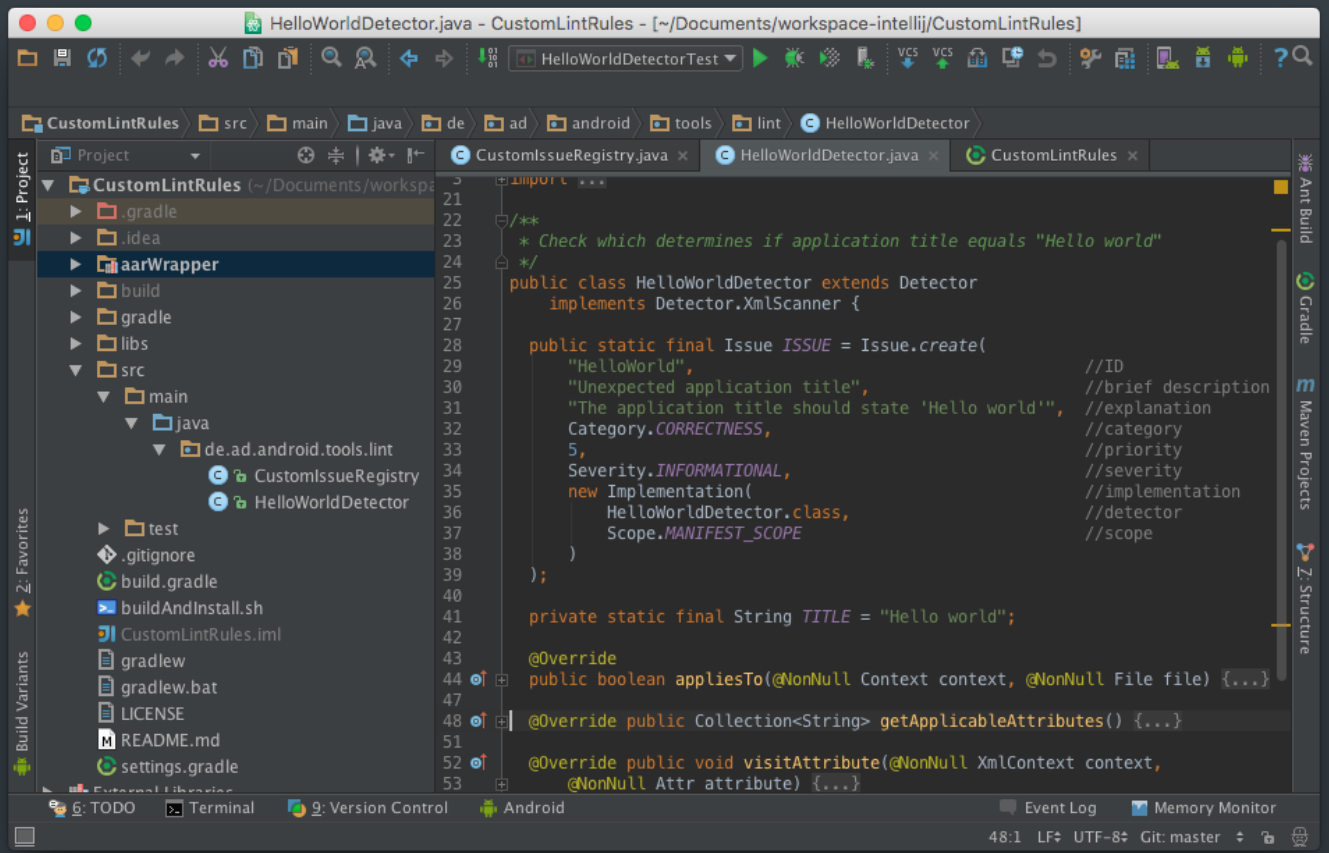
- Project setup
- Lint API

PROJECT SETUP

https://github.com/a11n/CustomLintRules

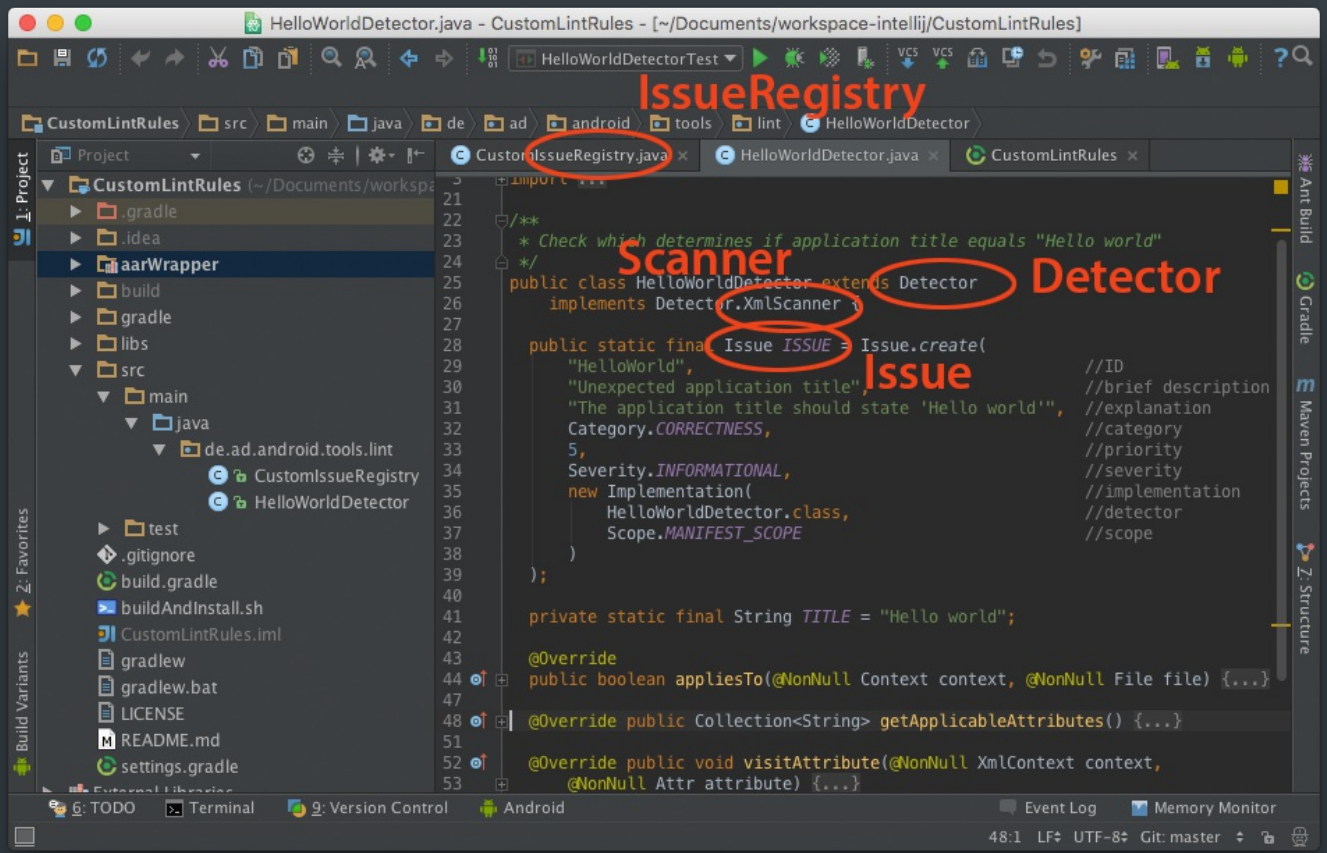


LINT API


HelloWorldDetector.java - CustomLintRules - [~/Documents/workspace-intellij/CustomLintRules]
HelloWorldDetectorTest
CustomLintRules
src
main
java
de
ad
android
tools
lint
HelloWorldDetector
Project
CustomIssueRegistry.java
HelloWorldDetector.java
CustomLintRules
.gradle
.idea
aarWrapper
build
gradle
libs
src
main
java
de.ad.android.tools.lint
CustomIssueRegistry
HelloWorldDetector
test
.gitignore
build.gradle
buildAndInstall.sh
CustomLintRules.iml
gradlew
gradlew.bat
LICENSE
README.md
settings.gradle
/**
* Check which determines if application title equals "Hello world"
*/
public class HelloWorldDetector extends Detector
implements Detector.XmlScanner {
public static final Issue ISSUE = Issue.create(
"HelloWorld", //ID
"Unexpected application title", //brief description
"The application title should state 'Hello world'", //explanation
Category.CORRECTNESS, //category
5, //priority
Severity.INFORMATIONAL, //severity
new Implementation(//implementation
HelloWorldDetector.class, //detector
Scope.MANIFEST_SCOPE //scope
)
);
private static final String TITLE = "Hello world";
@Override
public boolean appliesTo(@NonNull Context context, @NonNull File file) {...}
@Override public Collection<String> getApplicableAttributes() {...}
@Override public void visitAttribute(@NonNull XmlContext context,
@NonNull Attr attribute) {...}
6: TODO
Terminal
9: Version Control
Android
Event Log
Memory Monitor
48:1
LF
UTF-8
Git: master
Ant Build
Gradle
Maven Projects
Z: Structure
1: Project
2: Favorites
Build Variants

LINT API

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



LINT API

ISSUE

```
public static final Issue ISSUE = Issue.create(
    "HelloWorld",                                   //ID
    "Unexpected application title",                 //brief descript
    "The application title should"                  //explanation
      + "state 'Hello world'",
    Category.CORRECTNESS,                           //category
    5,                                              //priority
    Severity.INFORMATIONAL,                         //severity
    new Implementation(                             //implementation
        HelloWorldDetector.class,                   //detector
        Scope.MANIFEST_SCOPE                        //scope
    )
);
```

LINT API

DETECTOR

```
public class HelloWorldDetector extends Detector
 implements XmlScanner {

  public static final Issue ISSUE = Issue.create(...);

  @Override public Collection<String> getApplicableElements() {...

  @Override public Collection<String> getApplicableAttributes() {.

  @Override public void visitElement(@NonNull XmlContext context,
      @NonNull Element element) {...}

  @Override public void visitAttribute(@NonNull XmlContext context
      @NonNull Attr attribute) {...}
}
```

LINT API

SCANNER

- `JavaScanner`
- `ClassScanner`
- `BinaryResourceScanner`
- `ResourceFolderScanner`
- `XmlScanner`
- `GradleScanner`
- `OtherFileScanner`

LINT API

SCANNER

JavaScanner	XmlScanner
`applicableSuperClasses()`	`getApplicableEle`
`checkClass(...)`	`visitElement(..`
`getApplicableMethodNames()`	`getApplicableAtt`
`visitMethod(...)`	`visitAttribute(`
…	…

LINT API

ISSUEREGISTRY

```
public class CustomIssueRegistry extends IssueRegistry {
  @Override
  public List<Issue> getIssues() {
    return Arrays.asList(                //Note:
      MyCustomCheck.ISSUE,               //A check actually is a dete
      MyAdvancedCheck.AN_ISSUE,          //One detector can report
      MyAdvancedCheck.ANOTHER_ISSUE      //multiple types of issues.
    );
  }
}
```

VERIFICATION

- Testing
- Debugging

TESTING

- Test
- Assert
- JUnit4

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));
  }
}
```

ASSERT

Lint test library only allows assertion of `Strings`.

```
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"));
}
```

***#ProTip:** Let the test fail first and copy the assertion error to get the expected output.*

ASSERT

Two options for test data

- Inline resources

```
assertEquals("No warnings.",
  lintProject(
    xml("AndroidManifest.xml", "..."),
    java("src/main/java/test/MainActivity.java", "..."),
    ...));
```

- External resources

```
assertEquals("No warnings.",
  lintFiles(
    "ValidAndroidManifest.xml=>AndroidManifest.xml", ...));
```

JUNIT4

The official Android Lint test library only supports JUnit3 style.

- You have to extend `LintDetectorTest` and
- prefix your test methods with `test`.

LINT-JUNIT-RULE

MOTIVATION

- usage of JUnit4
- better assertions

https://github.com/a11n/lint-junit-rule

EXAMPLE

```
//Setup Detector and Issues
@Rule public Lint lint =
    new Lint(new MyCustomDetector(), MyCustomDetector.ISSUE);

@Test
public void test() throws Exception {
  //Lint the specified files from your test resources
  List<Warning> lintResult =
      lint.files("AndroidManifest.xml", "res/values/strings.xml");

  //Assertions
  //...
}
```

EXAMPLE

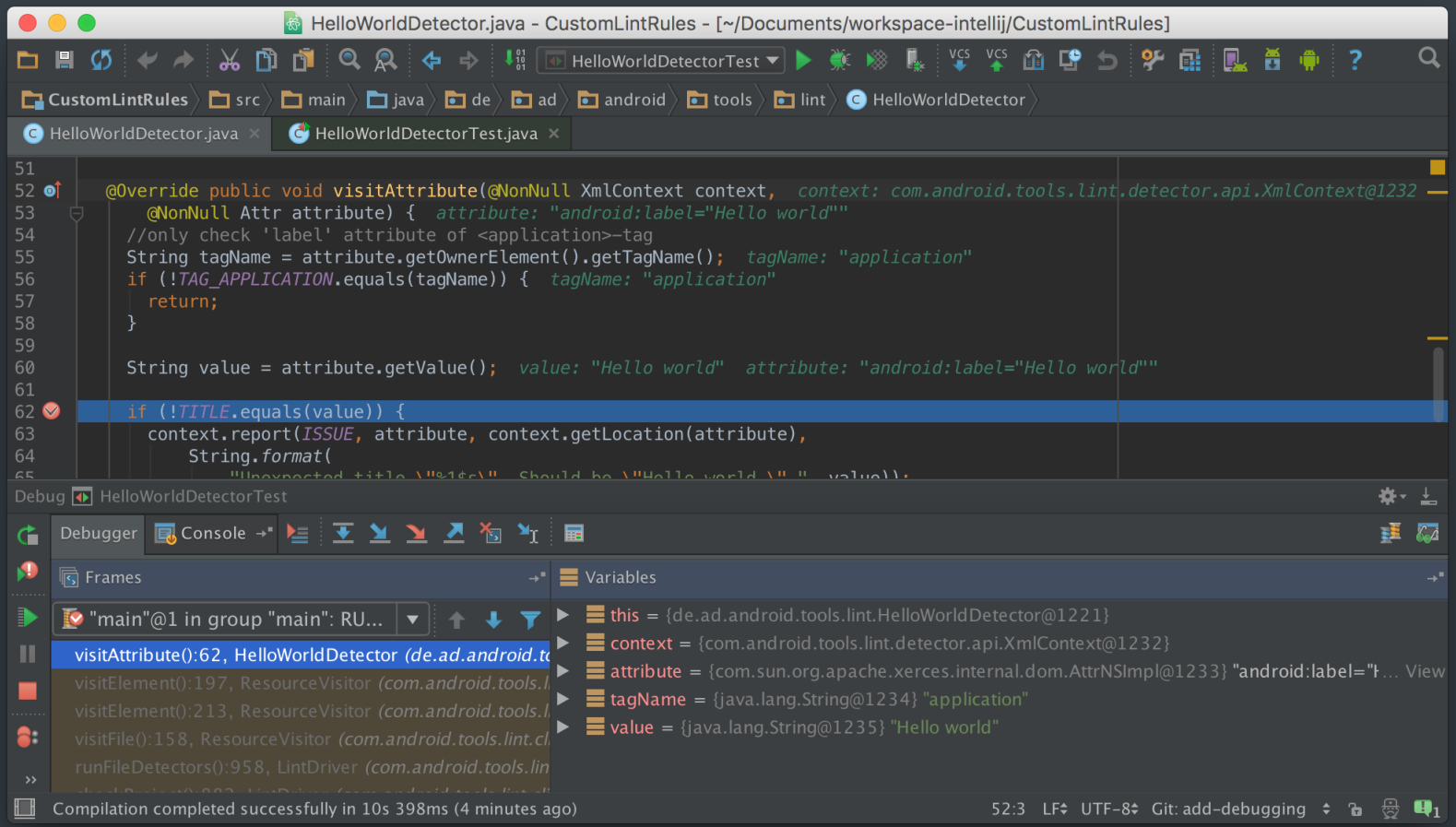
```
//AssertJ
assertThat(lintResult)
  .hasWarnings(2)
  .in("AndroidManifest.xml", "strings.xml")
  .atLine(8, 14)
  .withMessage("MyCustomDetector warning message.",
               "MyCustomDetector warning message.");

//Hamcrest
assertThat(lintResult,
           hasWarnings(
             in("AndroidManifest.xml", "strings.xml"),
             atLine(8, 14),
             withMessage("MyCustomDetector warning message.",
                         "MyCustomDetector warning message.")));
```

DEBUGGING

- during testing
- during Gradle run

DEBUGGING DURING TESTING

Works out of the box in IntelliJ and Android Studio.



DEBUGGING DURING GRADLE RUN

Credits to Marc Prengemann

- Add to `gradle.properties`

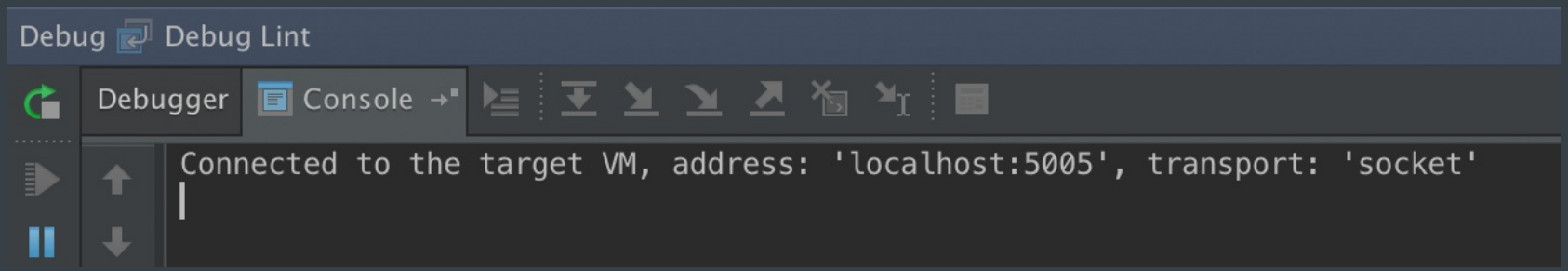
```
org.gradle.jvmargs=
  '-agentlib:jdwp=transport=dt_socket,server=y,suspend=n,addres
```

- Start Gradle Daemon

```
./gradlew --daemon
```

DEBUGGING DURING GRADLE RUN

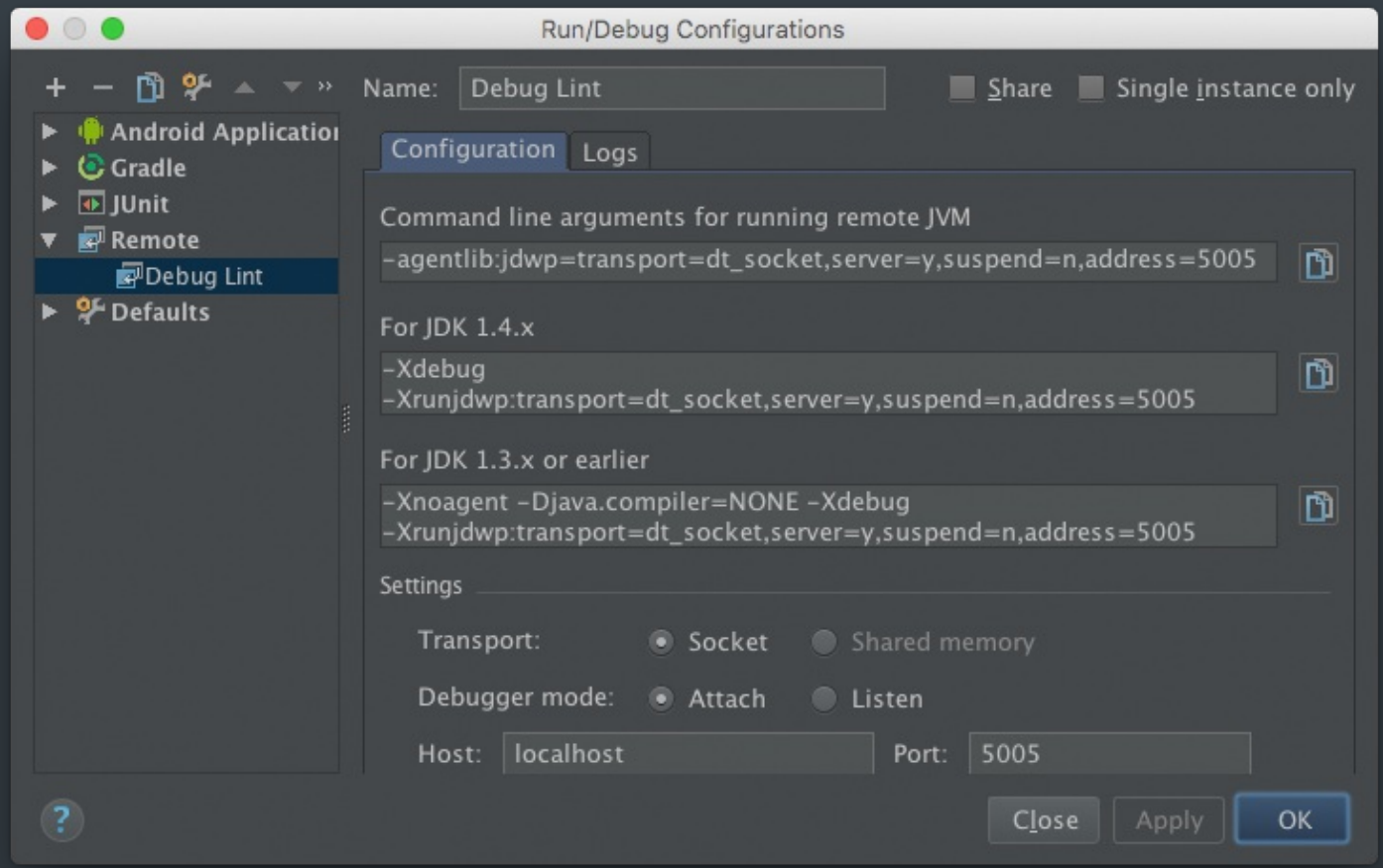
- Add and run new remote debug configuration



- Run Lint on your application project

```
./gradlew lint
```

DEBUGGING DURING GRADLE RUN



OPERATION

- Configuration
- Application
- Continuous Integration
- Discovery

CONFIGURATION

Lint can be configured in different ways and at different levels using:

- Gradle
- File
- Other

GRADLE-BASED CONFIGURATION

```
android {
    lintOptions {
        // set to true to turn off analysis progress reporting by l
        quiet true
        // if true, stop the gradle build if errors are found
        abortOnError false
        // if true, only report errors
        ignoreWarnings true
        }
        ...
    }
```

Complete list of Lint options.

FILE-BASED CONFIGURATION

`lint.xml` in the root directory of the Android project

```
<?xml version="1.0" encoding="UTF-8"?>
<lint>
    <!-- Disable the given check in this project -->
    <issue id="IconMissingDensityFolder" severity="ignore" />

    <!-- Ignore the ObsoleteLayoutParam issue in the specified fil
    <issue id="ObsoleteLayoutParam">
        <ignore path="res/layout/activation.xml" />
        <ignore path="res/layout-xlarge/activation.xml" />
    </issue>

    <!-- Change the severity of hardcoded strings to "error" -->
    <issue id="HardcodedText" severity="error" />
</lint>
```

OTHER

- in Java

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

- in XML

```
xmlns:tools="http://schemas.android.com/tc
tools:ignore="TheIssueYouWantToSuppress"
```

APPLICATION

- Basic approach
- Integrated approach

BASIC APPROACH

- utilizes basic Lint extension feature
- two steps setup
  1. assemble custom Lint rules into JAR
  2. copy JAR to `~/.android/lint/`

INTEGRATED APPROACH

Credits to Cheng Yang

- uses AAR bundle as wrapper
- two steps setup
  1. wrap custom Lint rules into an AAR
  2. make application project depend on that AAR

CONTINUOUS INTEGRATION

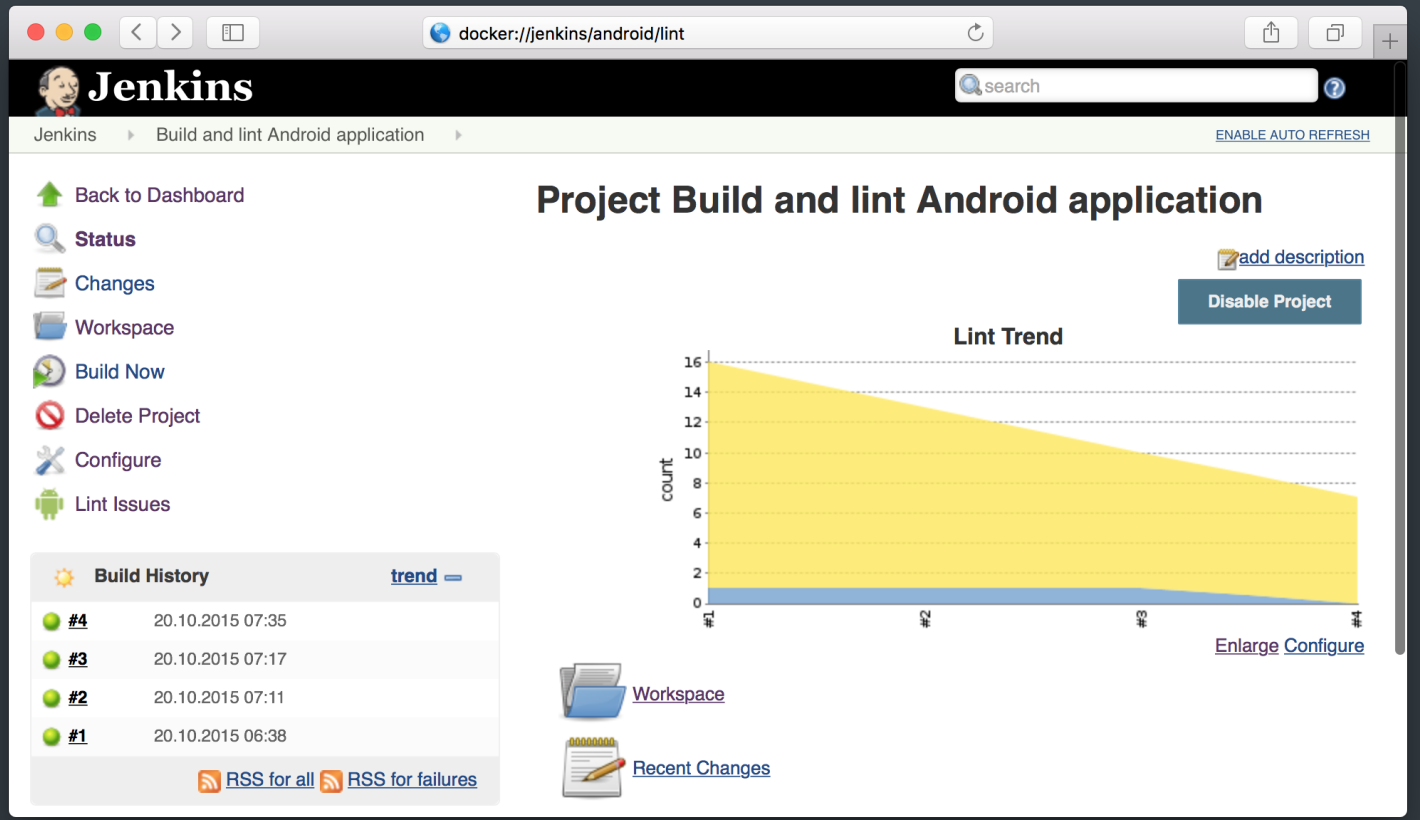
Lint should be integrated into the Continuous Integration process!

RECOMMENDATIONS

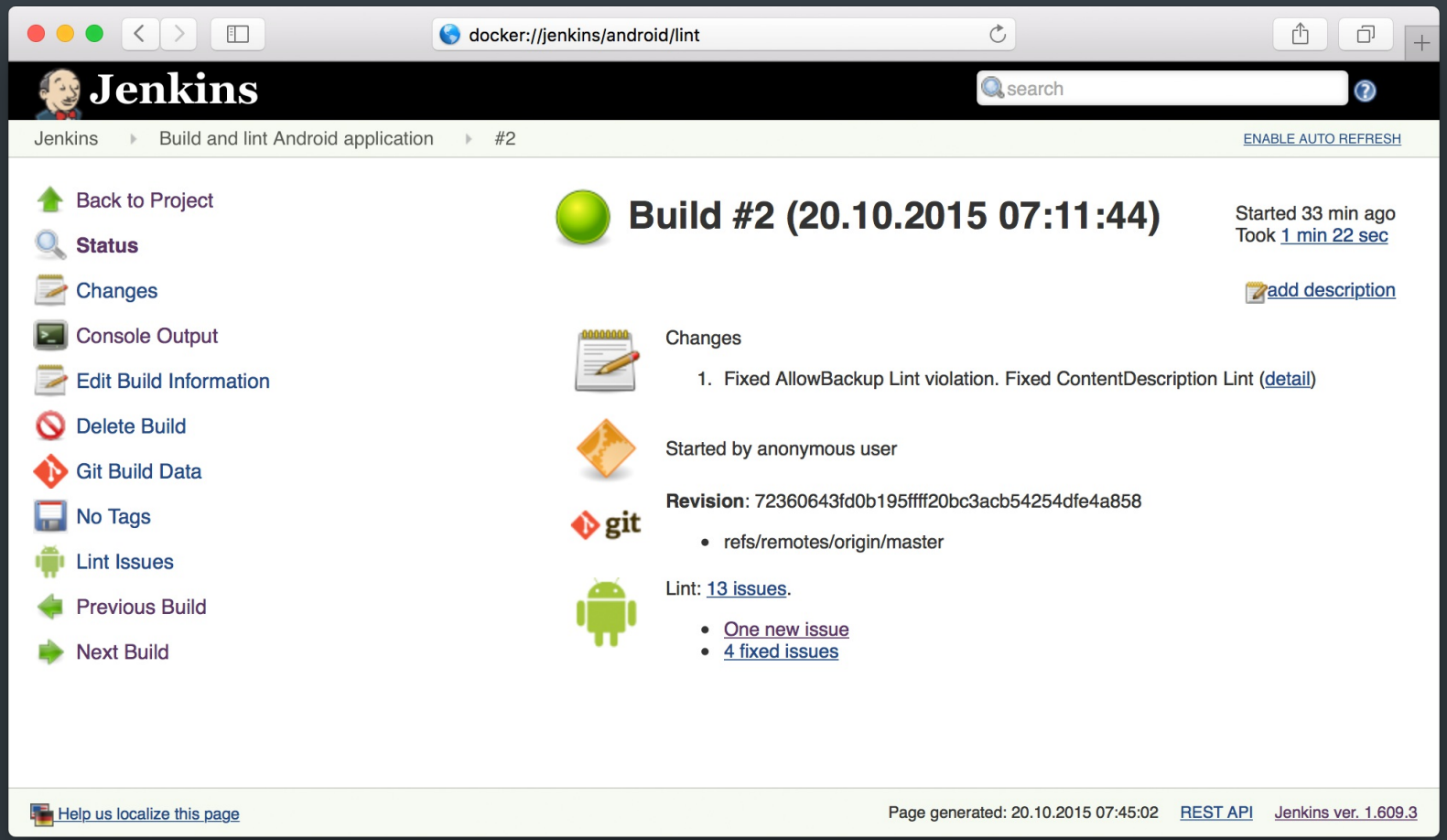
- Be strict
- Be verbose
- Evolve team culture

EXAMPLE

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



EXAMPLE


docker://jenkins/android/lint
Jenkins
search
Jenkins
Build and lint Android application
#2
ENABLE AUTO REFRESH
Back to Project
Status
Changes
Console Output
Edit Build Information
Delete Build
Git Build Data
No Tags
Lint Issues
Previous Build
Next Build
Build #2 (20.10.2015 07:11:44)
Started 33 min ago
Took 1 min 22 sec
add description
Changes
1. Fixed AllowBackup Lint violation. Fixed ContentDescription Lint (detail)
Started by anonymous user
Revision: 72360643fd0b195ffff20bc3acb54254dfe4a858
refs/remotes/origin/master
Lint: 13 issues.
One new issue
4 fixed issues
Help us localize this page
Page generated: 20.10.2015 07:45:02
REST API
Jenkins ver. 1.609.3

DISCOVERY

BACKGROUND

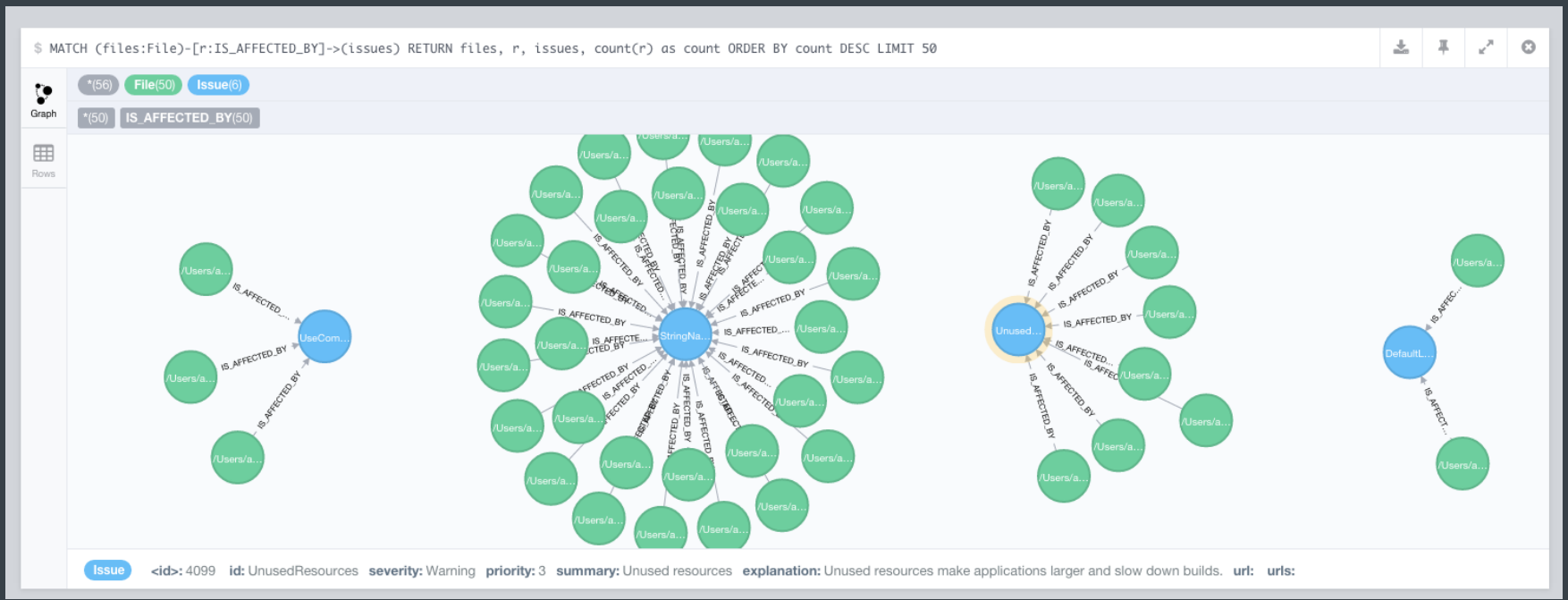
- Linting a pure scaffolded application project discovers **~50 Lint violations**
- Running Lint on a real world project *discovers* **>5000 Lint violations**

Challenge:
Discover relevant information.

DISCOVERY

LINT-GRAPH

https://github.com/a11n/lint-graph



DISCOVERY

LINT-BROWSER

☹ *Not open-sourced yet.* ☹

- Angular Material front-end
- parses `lint-results.xml`
- allows grouping, filtering, sorting, ...
- supports discovery of relations, potential sources, ... of issues.

FUTURE WORK

- Custom `SupportAnnotations`
- Custom `QuickFixes`
- Kotlin support
- ...

THANK YOU FOR YOUR ATTENTION.

Q&A

q2ad a11n