# Automated Detection and Reporting of Build Cache Misses

## DPE Summit 2023

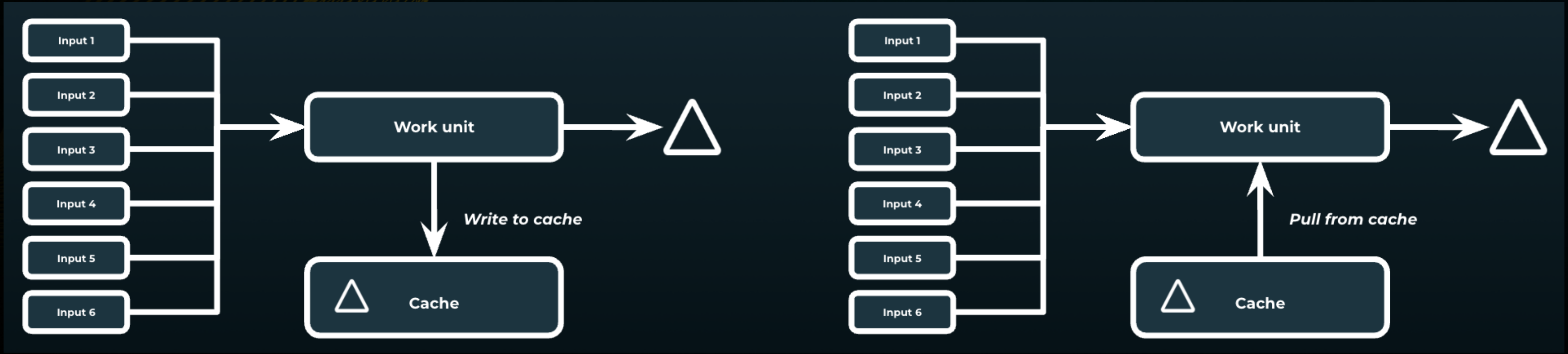
Etienne Studer, SVP of Engineering, Develocity

## Increased DevProd through Build Caching

**Build caching is a powerful technology to accelerate build performance on CI and locally.**

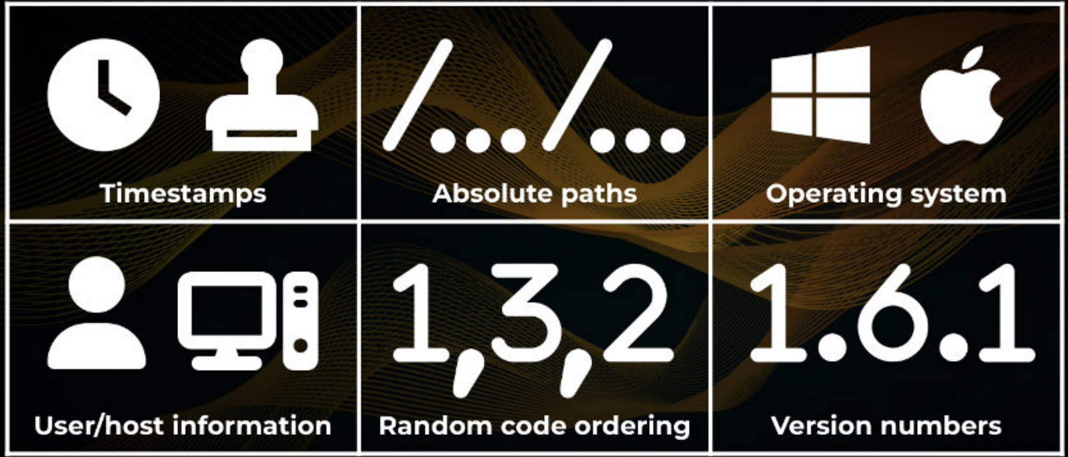
## How Build Caching works

**Build Caching avoids doing the same work already done before by reusing the work unit output stored in a local or remote build cache.**



## Volatility in the inputs of work units

**Volatile inputs of work units cause unnecessary execution of work units.**



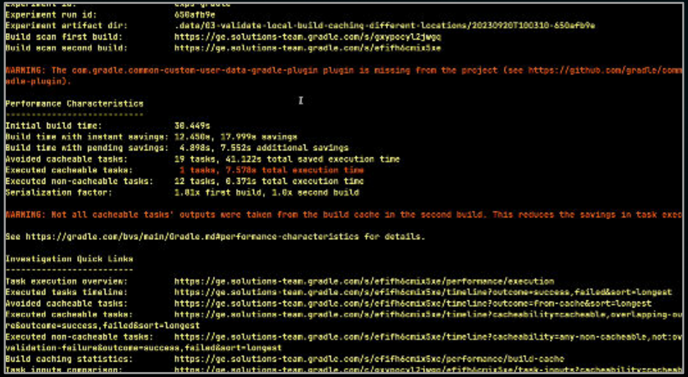
## Verification if a build is fully cacheable

When a build is executed twice under the same conditions, all of the second build’s cache-compatible work units should take their output from the build cache.

## Automation of the verification process

**Use the Build Validation Scripts to make the process reliable, repeatable, automated, and productive.**

https://github.com/gradle/gradle-enterprise-build-validation-scripts

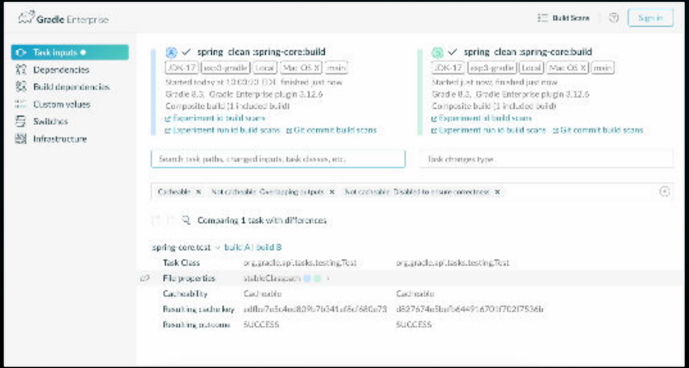


```
Experiment id:                exp3-gradle
Experiment run id:            650afb9e
Experiment artifact dir:      .data/03-validate-local-build-caching-different-locations/20230920T100310-650afb9e
Build scan first build:       https://ge.solutions-team.gradle.com/s/gxypocyl2jwgq
Build scan second build:      https://ge.solutions-team.gradle.com/s/efifh6cmix5xe

WARNING: The com.gradle.common-custom-user-data-gradle-plugin plugin is missing from the project (see https://github.com/gradle/comm
adle-plugin).

Performance Characteristics
---------------------------
Initial build time:            30.449s
Build time with instant savings: 12.450s, 17.999s savings
Build time with pending savings:  4.898s, 7.552s additional savings
Avoided cacheable tasks:       19 tasks, 41.122s total saved execution time
Executed cacheable tasks:       1 tasks, 7.578s total execution time
Executed non-cacheable tasks:  12 tasks, 0.371s total execution time
Serialization factor:          1.01x first build, 1.0x second build

WARNING: Not all cacheable tasks' outputs were taken from the build cache in the second build. This reduces the savings in task exec

See https://gradle.com/bvs/main/Gradle.md#performance-characteristics for details.

Investigation Quick Links
-------------------------
Task execution overview:       https://ge.solutions-team.gradle.com/s/efifh6cmix5xe/performance/execution
Executed tasks timeline:       https://ge.solutions-team.gradle.com/s/efifh6cmix5xe/timeline?outcome=success,failed&sort=longest
Avoided cacheable tasks:       https://ge.solutions-team.gradle.com/s/efifh6cmix5xe/timeline?outcome=from-cache&sort=longest
Executed cacheable tasks:      https://ge.solutions-team.gradle.com/s/efifh6cmix5xe/timeline?cacheability=cacheable,overlapping-ou
re&outcome=success,failed&sort=longest
Executed non-cacheable tasks:  https://ge.solutions-team.gradle.com/s/efifh6cmix5xe/timeline?cacheability=any-non-cacheable,not:ov
validation-failure&outcome=success,failed&sort=longest
Build caching statistics:      https://ge.solutions-team.gradle.com/s/efifh6cmix5xe/performance/build-cache
Task inputs comparison:        https://ge.solutions-team.gradle.com/c/gxypocyl2jwgq/efifh6cmix5xe/task-inputs?cacheability=cacheab
```

## Investigation of build cache misses

**Use Develocity Build Scan Comparison to investigate changes in task inputs that caused a build cache miss.**

https://ge.solutions-team.gradle.com/c/gxypocyl2jwgq/efifh6cmix5xe/task-inputs



spring clean :spring-core:build
JDK-17 | [illegible]-gradle | Local | Mac OS X | main
Started today at 13:03:03 EDT, finished just now
Gradle 8.3, Gradle Enterprise plugin 3.12.6
Composite build (1 included build)
Experiment id build scans
Experiment run id build scans | Git commit build scans

spring clean :spring-core:build
JDK-17 | [illegible]-gradle | Local | Mac OS X | main
Started just now, finished just now
Gradle 8.3, Gradle Enterprise plugin 3.12.6
Composite build (1 included build)
Experiment id build scans
Experiment run id build scans | Git commit build scans

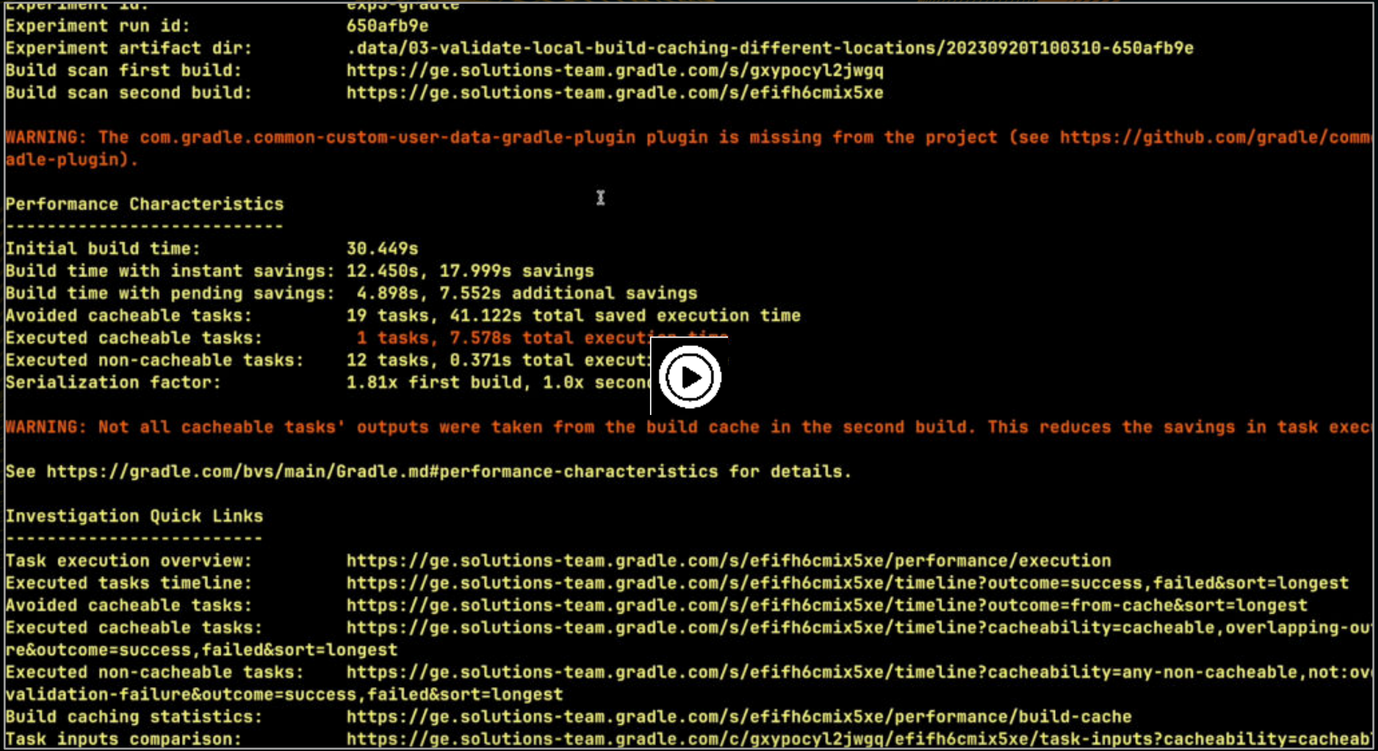
Search task paths, changed inputs, task classes, etc.
Task changes type
Cacheable × | Not cacheable: Overlapping outputs × | Not cacheable: Disabled to ensure correctness ×

Comparing 1 task with differences

spring-core:test | build A | build B

| | | |
|---|---|---|
| Task Class | org.gradle.api.tasks.testing.Test | org.gradle.api.tasks.testing.Test |
| File properties | stableClasspath ● ● | |
| Cacheability | Cacheable | Cacheable |
| Resulting cache key | [illegible] | [illegible] |
| Resulting outcome | SUCCESS | SUCCESS |

# Demo: BVS-driven build cacheability verification



```
Experiment id:                exp3-gradle
Experiment run id:            650afb9e
Experiment artifact dir:      .data/03-validate-local-build-caching-different-locations/20230920T100310-650afb9e
Build scan first build:       https://ge.solutions-team.gradle.com/s/gxypocyl2jwgq
Build scan second build:      https://ge.solutions-team.gradle.com/s/efifh6cmix5xe

WARNING: The com.gradle.common-custom-user-data-gradle-plugin plugin is missing from the project (see https://github.com/gradle/comm
adle-plugin).

Performance Characteristics
---------------------------
Initial build time:           30.449s
Build time with instant savings: 12.450s, 17.999s savings
Build time with pending savings:  4.898s, 7.552s additional savings
Avoided cacheable tasks:      19 tasks, 41.122s total saved execution time
Executed cacheable tasks:      1 tasks, 7.578s total executi
Executed non-cacheable tasks:  12 tasks, 0.371s total execut
Serialization factor:         1.81x first build, 1.0x secon

WARNING: Not all cacheable tasks' outputs were taken from the build cache in the second build. This reduces the savings in task exec

See https://gradle.com/bvs/main/Gradle.md#performance-characteristics for details.

Investigation Quick Links
-------------------------
Task execution overview:      https://ge.solutions-team.gradle.com/s/efifh6cmix5xe/performance/execution
Executed tasks timeline:      https://ge.solutions-team.gradle.com/s/efifh6cmix5xe/timeline?outcome=success,failed&sort=longest
Avoided cacheable tasks:      https://ge.solutions-team.gradle.com/s/efifh6cmix5xe/timeline?outcome=from-cache&sort=longest
Executed cacheable tasks:     https://ge.solutions-team.gradle.com/s/efifh6cmix5xe/timeline?cacheability=cacheable,overlapping-ou
re&outcome=success,failed&sort=longest
Executed non-cacheable tasks: https://ge.solutions-team.gradle.com/s/efifh6cmix5xe/timeline?cacheability=any-non-cacheable,not:ov
validation-failure&outcome=success,failed&sort=longest
Build caching statistics:     https://ge.solutions-team.gradle.com/s/efifh6cmix5xe/performance/build-cache
Task inputs comparison:       https://ge.solutions-team.gradle.com/c/gxypocyl2jwgq/efifh6cmix5xe/task-inputs?cacheability=cacheab
```

## Build caching regressions are costly

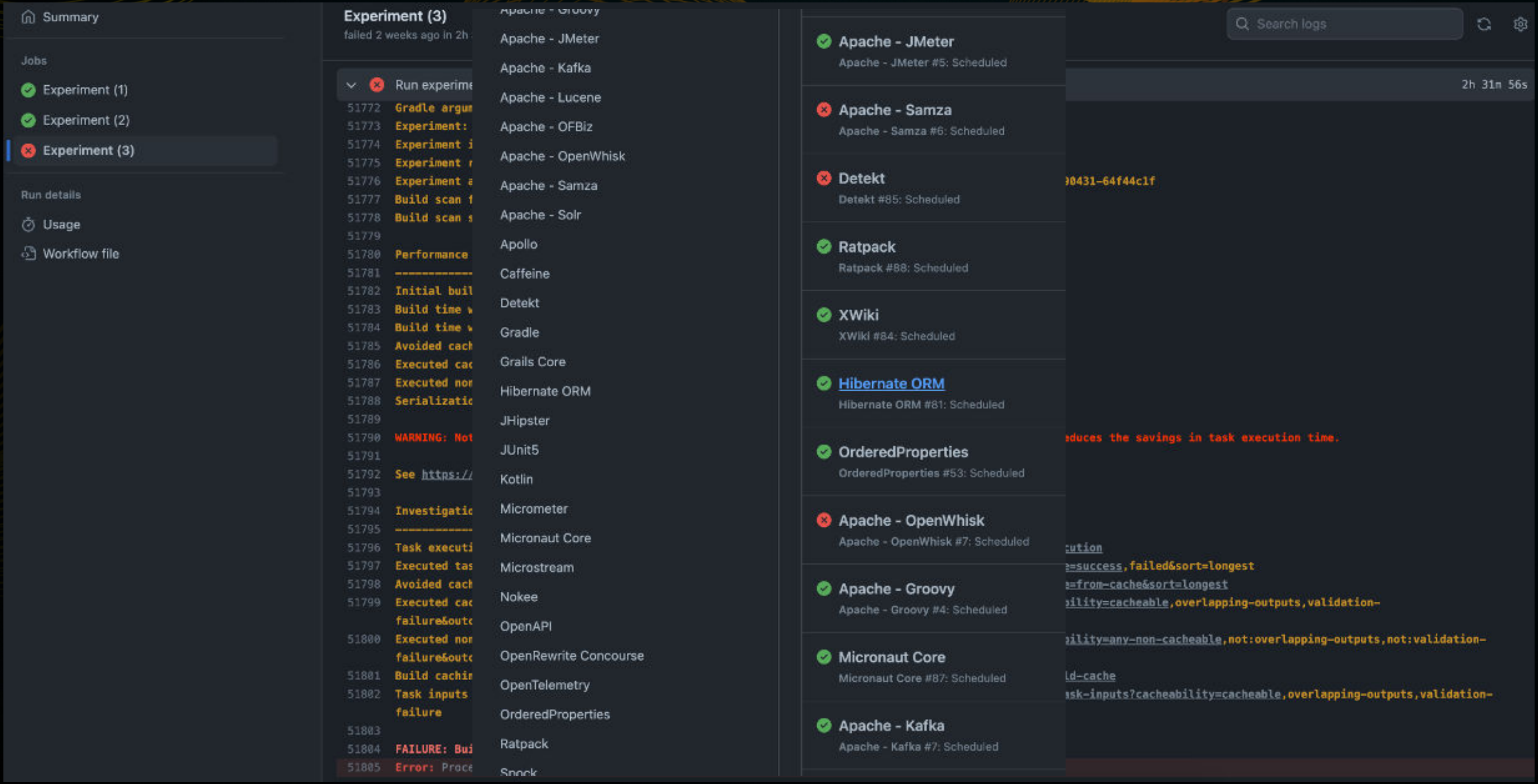
**Build changes that make a build no longer fully cacheable must be detected and notifications issued so the regression can be promptly investigated and fixed.**

## Automation of the regression detection process

**The Build Validation Scripts should be run on a continuous base in CI and cause the CI run to fail if a build is no longer fully cacheable.**

```
./03-validate-local-build-caching-different-locations.sh -r https://github.com/spring-projects/spring-framework --fail-if-not-fully-cacheable
```

# Example: Continuously verified OSS projects


Summary

Jobs
- Experiment (1)
- Experiment (2)
- Experiment (3)

Run details
- Usage
- Workflow file

**Experiment (3)**
failed 2 weeks ago in 2h

Run experime

```
Gradle argum
Experiment:
Experiment i
Experiment r
Experiment a
Build scan f
Build scan s

Performance
------------
Initial buil
Build time w
Build time w
Avoided cach
Executed cac
Executed non
Serializatio

WARNING: Not

See https://

Investigatio
------------
Task executi
Executed tas
Avoided cach
Executed cac
failure&outc
Executed non
failure&outc
Build cachin
Task inputs
failure

FAILURE: Bui
Error: Proce
```

Apache - Groovy
Apache - JMeter
Apache - Kafka
Apache - Lucene
Apache - OFBiz
Apache - OpenWhisk
Apache - Samza
Apache - Solr
Apollo
Caffeine
Detekt
Gradle
Grails Core
Hibernate ORM
JHipster
JUnit5
Kotlin
Micrometer
Micronaut Core
Microstream
Nokee
OpenAPI
OpenRewrite Concourse
OpenTelemetry
OrderedProperties
Ratpack
Spock

- ✓ **Apache - JMeter** — Apache - JMeter #5: Scheduled
- ✗ **Apache - Samza** — Apache - Samza #6: Scheduled
- ✗ **Detekt** — Detekt #85: Scheduled
- ✓ **Ratpack** — Ratpack #88: Scheduled
- ✓ **XWiki** — XWiki #84: Scheduled
- ✓ **Hibernate ORM** — Hibernate ORM #81: Scheduled
- ✓ **OrderedProperties** — OrderedProperties #53: Scheduled
- ✗ **Apache - OpenWhisk** — Apache - OpenWhisk #7: Scheduled
- ✓ **Apache - Groovy** — Apache - Groovy #4: Scheduled
- ✓ **Micronaut Core** — Micronaut Core #87: Scheduled
- ✓ **Apache - Kafka** — Apache - Kafka #7: Scheduled

Search logs

2h 31m 56s

```
90431-64f44c1f

educes the savings in task execution time.

cution
e=success,failed&sort=longest
e=from-cache&sort=longest
bility=cacheable,overlapping-outputs,validation-

bility=any-non-cacheable,not:overlapping-outputs,not:validation-

ld-cache
ask-inputs?cacheability=cacheable,overlapping-outputs,validation-
```

https://github.com/gradle/gradle-enterprise-oss-projects/actions

## Cross-machine regression detection

**The verification process can be extended by orchestrating the execution of two builds of the same version on different machines and submitting them to the Build Validation Scripts for verification of full cacheability.**

[./04-validate-remote-build-caching-ci-ci.sh -1 https://ge.solutions-team.gradle.com/s/gxypocyl2jwgq -2 https://ge.solutions-team.gradle.com/s/efifh6cmix5xe --fail-if-not-fully-cacheable]

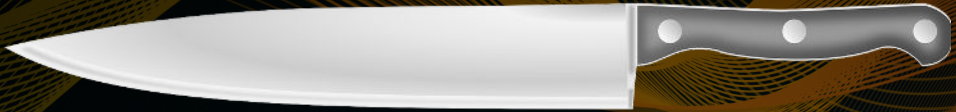
## Regression detection generalization

**The verification process can be further generalized by automatically finding – amongst all builds – similar builds where full cacheability is expected and verifying them
for full cacheability.**

# Regression detection generalization

| | | | |
|---|---|---|---|
| **spring-authorization-server** | :spring-authorization-server-docs:asciidoctor | https://ge.spring.io/c/kgzhygoftphtq/yf64zqfxta5xe | **Volatile input** |
| **spring-boot-build** | :spring-boot-project:spring-boot-actuator-autoconfigure:asciidoctor | https://ge.spring.io/c/6skln6qftnq2g/d6b2kc3yfgir4 | **Volatile input** |
| **spring-boot-build** | :spring-boot-project:spring-boot-actuator-autoconfigure:asciidoctorPdf | https://ge.spring.io/c/6skln6qftnq2g/d6b2kc3yfgir4 | **Volatile input** |
| **spring-boot-build** | :spring-boot-project:spring-boot-docs:asciidoctor | https://ge.spring.io/c/dbi36jrigy7im/ha3s5a3rlltj2 | **Volatile input** |
| **spring-boot-build** | :spring-boot-project:spring-boot-docs:asciidoctorMultipage | https://ge.spring.io/c/dbi36jrigy7im/ha3s5a3rlltj2 | **Volatile input** |
| **spring-boot-build** | :spring-boot-project:spring-boot-docs:asciidoctorPdf | https://ge.spring.io/c/dbi36jrigy7im/ha3s5a3rlltj2 | **Volatile input** |
| **javadoc-plugin** | :validatePlugins | https://ge.spring.io/c/vzkmbme7boyuy/6d7mcunkposza | **Eviction** |
| **spring-authorization-server** | :spring-authorization-server-docs:api | https://ge.spring.io/c/ejldvuvw7ushw/plvsiuinoiekw | **Eviction** |

Make your investment of achieving a fully cacheable build worthwhile by automating the continuous detections of caching regressions.



Keep the knife sharp!

# THANKS

etienne@gradle.com
gradle.com