Unlocking build analytics
Getting started with the Gradle Enterprise API

Gabriel Feo, Staff Engineer at iFood
Gradle Enterprise API
Gradle Enterprise REST API
Gradle Enterprise REST API

A few ideas
Gradle Enterprise REST API

A few ideas

1. Benchmarks
A few ideas

1. Benchmarks
2. Monitoring
A few ideas

1. Benchmarks
2. Monitoring
3. Investigations
A few ideas

1. Benchmarks
2. Monitoring
3. Investigations
gradle-enterprise-api-kotlin
gradle-enterprise-api-kotlin

A library to use the Gradle Enterprise REST API in Kotlin
gradle-enterprise-api-kotlin

A library to use the Gradle Enterprise REST API in Kotlin

1. Generated from OpenAPI spec
gradle-enterprise-api-kotlin

A library to use the Gradle Enterprise REST API in Kotlin

1. Generated from OpenAPI spec
2. Common utilities
gradle-enterprise-api-kotlin

A library to use the Gradle Enterprise REST API in Kotlin

1. Generated from OpenAPI spec
2. Common utilities
3. Easy to add
val api = GradleEnterpriseApi.newInstance()
val api = GradleEnterpriseApi.newInstance()

api.buildsApi = Gradle Enterprise

- Builds
- Build Cache
- Meta
- Test Distribution
val api = GradleEnterpriseApi.newInstance()

api.buildsApi.getBuilds()
API for monitoring
API for monitoring
Automating flaky test tickets
Automating flaky test tickets

18 builds (1% of 1.33K builds that executed tests)
Automating flaky test tickets

val api = GradleEnterpriseApi.newInstance()

api.testsApi.getTestContainers()
Automating flaky test tickets

```kotlin
val api = GradleEnterpriseApi.newInstance()

api.testsApi.getTestContainers(
    startTimeMin = yesterday,
)```
Automating flaky test tickets

```kotlin
val api = GradleEnterpriseApi.newInstance()

api.testsApi.getTestContainers(
    startTimeMin = yesterday,
    startTimeMax = now,
)```
Automating flaky test tickets

val api = GradleEnterpriseApi.newInstance()

api.testsApi.getTestContainers(
    startTimeMin = yesterday,
    startTimeMax = now,
    testOutcomes = listOf(TestOutcome.flaky),
)
Automating flaky test tickets

Response

```json
{
  "content": [
    {
      "name": "br.com.ifood.acquisition.usecase.SendEventPurchasedTest",
      "outcomeDistribution": {
        "passed": 154,
        "failed": 0,
        "skipped": 0,
        "flaky": 1,
        "notSelected": 0,
        "total": 155
      }
    }
  ]
}
```
Automating flaky test tickets

Response

```json
{
  "content": [
    {
      "name": "br.com.ifood.acquisition.ubcase.SendEventPurchasedTest",
      "outcomeDistribution": {
        "passed": 154,
        "failed": 0,
        "skipped": 0,
        "flaky": 1,
        "notSelected": 0,
        "total": 155
      }
    }
  ]
}
```
Automating flaky test tickets

```kotlin
val api = GradleEnterpriseApi.newInstance()
api.testsApi.getTestContainers(
    startTimeMin = yesterday,
    startTimeMax = now,
    testOutcomes = listOf(TestOutcome.flaky),
)
```
Automating flaky test tickets

```kotlin
val api = GradleEnterpriseApi.newInstance()

val response = api.testsApi.getTestContainers(
    startTimeMin = yesterday,
    startTimeMax = now,
    testOutcomes = listOf(TestOutcome.flaky),
)

openTickets(response)
```
Jupyter notebooks
Jupyter notebooks

Benefits
Jupyter notebooks

Benefits

1. Re-run a line of code
Jupyter notebooks

1. Re-run a line of code

```scala
val builds = fetchBuilds()
val errorBuilds = builds.filter { ... }
prettyPrint(errorBuilds)
```
1. Re-run a line of code

```scala
val builds = fetchBuilds()
val errorBuilds = builds.filter { ... }
prettyPrint(errorBuilds)
```
1. Re-run a line of code
Jupyter notebooks

1. Re-run a line of code

```kotlin
suspend fun main() {
    val builds = fetchBuilds()
    val errorBuilds = builds.filter { ... }
    prettyPrint(errorBuilds)
}
```
Jupyter notebooks

1. Re-run a line of code

```scala
val builds = fetchBuilds()
val errorBuilds = builds.filter{ ... }
prettyPrint(errorBuilds)
```
Jupyter notebooks

Benefits

1. Re-run a line of code
2. Easy libraries
2. Easy libraries

```python
%use \_
```

- biokotlin
- combinatoricskt
- coroutines
- dataframe
- datetime
- deplearning4j
- deplearning4j-cuda
- default
Jupyter notebooks

2. Easy libraries

In [1]: %use gradle-enterprise-api-kotlin
Jupyter notebooks

Benefits

1. Re-run a line of code
2. Easy libraries
3. Rendering
3. Rendering

In 2
1 JSON("...")
2 MIME("image/jpeg" to "...")
3 HTML("<h3>I'm rendered HTML!</h3>")

Executed at 2023.09.01 17:20:00 in 50ms

Out 2
I'm rendered HTML!
Jupyter notebooks

2. Easy libraries
3. Rendering

In _ 1 %use dataframe
## Jupyter notebooks

2. Easy libraries
3. Rendering

```python
%use dataframe
def builds.toDataFrame()
```

Executed at 2023.09.19 00:23:53 in 13s 427ms

<table>
<thead>
<tr>
<th>id</th>
<th>duration</th>
<th>startTime</th>
<th>tags</th>
</tr>
</thead>
<tbody>
<tr>
<td>cuk6ujuhrt3gy</td>
<td>PT7M26.294S</td>
<td>2023-09-18T19:12:48.015Z</td>
<td>[Cl, ...]</td>
</tr>
<tr>
<td>chnzaxp6w4knc</td>
<td>PT8M13.01S</td>
<td>2023-09-18T19:12:36.066Z</td>
<td>[Cl, ...]</td>
</tr>
<tr>
<td>cjsqidnwdigzy</td>
<td>PT1M45.662S</td>
<td>2023-09-18T19:19:47.199Z</td>
<td>[Cl, ...]</td>
</tr>
</tbody>
</table>
Jupyter notebooks

2. Easy libraries
3. Rendering

```python
%use kandy
plot {
    histogram(errorDates)
}
```

Executed at 2023.09.19 00:47:36 in 28s 90ms
Jupyter notebooks

2. Easy libraries
3. Rendering

```
In 35 1 %use kandy
2 plot {
3    histogram(errorDates)
4 }

Executed at 2023.09.19 00:47:36 in 28s 90ms
```
Jupyter notebooks

Benefits

1. Re-run a line of code
2. Easy libraries
3. Rendering
4. Documentation
Jupyter notebooks

4. Documentation
Jupyter notebooks

Benefits

1. Re-run a line of code
2. Easy libraries
3. Rendering
4. Documentation
API for investigations
API for investigations
Finding remote cache errors
Finding remote cache errors

⚠️ The remote build cache was disabled during the build due to errors.
Finding remote cache errors

```kotlin
val api = GradleEnterpriseApi.newInstance()
api.buildsApi.getBuilds(
```
Finding remote cache errors

```kotlin
val api = GradleEnterpriseApi.newInstance()

api.buildsApi.getBuilds(since = start)
```
Finding remote cache errors

```kotlin
val api = GradleEnterpriseApi.newInstance()
api.buildsApi.getBuilds(since = start)

✗ Paging logic required
```
Finding remote cache errors

val api = GradleEnterpriseApi.newInstance()

api.buildsApi.getBuildsFlow(since = start)

✓ Paging stream
Finding remote cache errors

```kotlin
val api = GradleEnterpriseApi.newInstance()

api.buildsApi.getBuildsFlow(since = start)
    .map { api.buildsApi.getGradleBuildCachePerformance(it.id) }
```
Finding remote cache errors

```kotlin
val api = GradleEnterpriseApi.newInstance()

val builds = api.buildsApi.getBuilds_since = start
.map { api.buildsApi.getGradleBuildCachePerformance { it.id } }
toList()

builds.\textit{filter} \{ perfInfo ->
```
Finding remote cache errors

```kotlin
val api = GradleEnterpriseApi.newInstance()

val builds = api.buildsApi.getBuilds(since = start)
    .map { api.buildsApi.getGradleBuildCachePerformance(it.id) }
    .toList()

builds.filter { perfInfo ->
    perfInfo.buildCaches.remote.isDisabledDueToError
}
```
Finding remote cache errors

```kotlin
val api = GradleEnterpriseApi.newInstance()

val builds = api.buildsApi.getBuilds(since = start)
    .map { api.buildsApi.getGradleBuildCachePerformance(it.id) }
    .toList()

val errorBuilds = builds.filter { perfInfo ->
    perfInfo.buildCaches.remote.isDisabledDueToError
}

errorBuilds.toDataFrame()```
Finding remote cache errors

```kotlin
val api = GradleEnterpriseApi.newInstance()

val builds = api.buildsApi.getBuilds(since = start)
    .map { api.buildsApi.getGradleBuildCachePerformance(it.id) }
    .toList()

val errorBuilds = builds.filter { perfInfo ->
    perfInfo.buildCaches.remote.isDisabledDueToError
}

errorBuilds.toDataFrame()
```

<table>
<thead>
<tr>
<th>id</th>
<th>duration</th>
<th>startTime</th>
<th>tags</th>
<th>customValues</th>
</tr>
</thead>
<tbody>
<tr>
<td>chnzaxp6w4knc</td>
<td>PT8M13.01S</td>
<td>2023-09-18T19:12:36.066Z</td>
<td>[CI, doctor-negative-savings, f..., ...]</td>
<td>[:feature:about-screen:impl:tes..., ...]</td>
</tr>
<tr>
<td>cjsqidnwdigzy</td>
<td>PT1M45.662S</td>
<td>2023-09-18T19:19:47.199Z</td>
<td>[CI, doctor-fresh-daemon, fresh..., ...]</td>
<td>[Cl job=code_coverage, Cl job l..., ...]</td>
</tr>
<tr>
<td>ussqjqvskjzrq</td>
<td>PT1M59.723S</td>
<td>2023-09-18T19:19:47.600Z</td>
<td>[CI, doctor-fresh-daemon, fresh..., ...]</td>
<td>[Cl job=android_lint, Cl job ID..., ...]</td>
</tr>
</tbody>
</table>
Finding remote cache errors

```
val api = GradleEnterpriseApi.newInstance()

val builds = api.buildsApi.getBuilds(since = start)
    .map { api.buildsApi.getGradleBuildCachePerformance(it.id) }
    .toList()

val errorBuilds = builds.filter { perfInfo ->
    perfInfo.buildCaches.remote.isDisabledDueToError
}

errorBuilds.toDataFrame().toCsv()

id,startTime
   cuk6ujhrt3gy,2023-09-18T19:12:48.015Z
  chnzaxp6w4knc,2023-09-18T19:12:36.066Z
     cjsqidnwdigzy,2023-09-18T19:19:47.199Z
   ussqjqvskjzrq,2023-09-18T19:19:47.600Z
     lt5cge2gf2pwg,2023-09-18T19:12:34.600Z
```
Finding remote cache errors

```kotlin
val api = GradleEnterpriseApi.newInstance()

val builds = api.buildsApi.getBuilds(since = start)
    .map { api.buildsApi.getGradleBuildCachePerformance(it.id) }
    .toList()

val errorBuilds = builds.filter { perfInfo ->
    perfInfo.buildCaches.remote.isDisabledDueToError
}

val dates = datesOf(errorBuilds)
plot {
    histogram(dates)
}
```
Finding remote cache errors

```kotlin
val api = GradleEnterpriseApi.newInstance()

val builds = api.buildsApi.getBuilds(since = start).
    map { api.buildsApi.getGradleBuildCachePerformance(it.id) }
    .toList()

val errorBuilds = builds.filter { perfInfo ->
    perfInfo.buildCaches.remote.isDisabledDueToError
}

val dates = datesOf(errorBuilds)
plot {
    histogram(dates)
}
```
Unlocking build analytics
Getting started with the Gradle Enterprise API

Gabriel Feo, Staff Engineer at iFood
gabriel@gabrielfeo.com