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

Gradle Enterprise REST API

A few ideas

1. Benchmarks
2. Monitoring

Gradle Enterprise REST API

A few ideas

1. Benchmarks
2. Monitoring
3. Investigations

Gradle Enterprise REST API

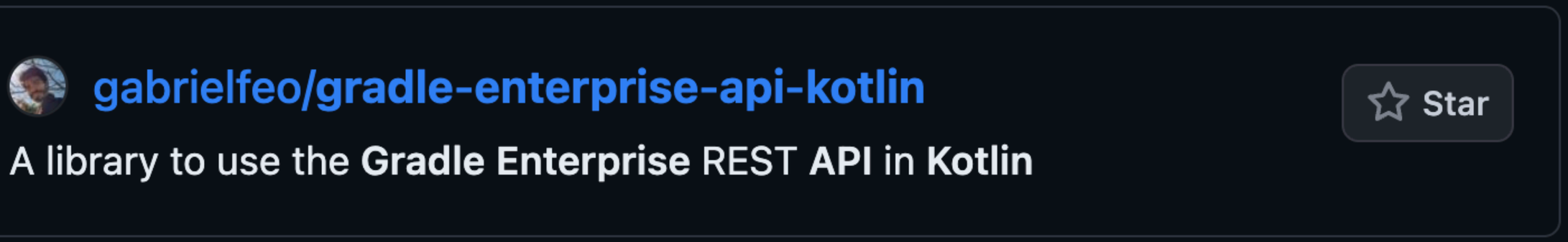
A few ideas

1. Benchmarks
2. Monitoring
3. Investigations

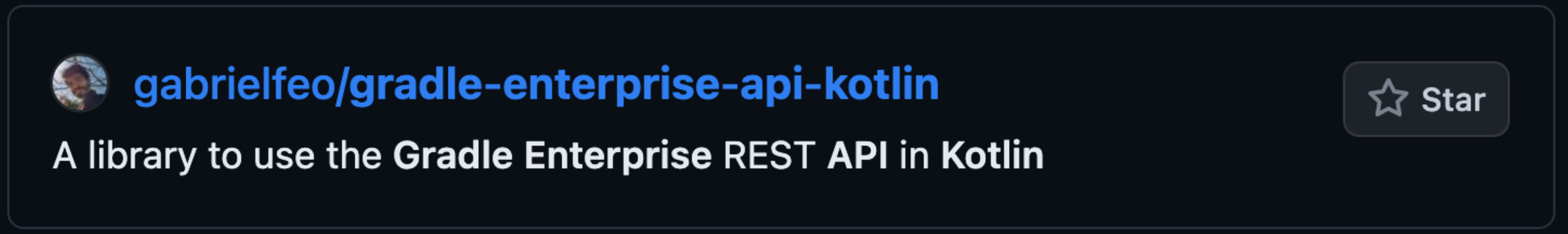
gradle-enterprise-api-kotlin

gradle-enterprise-api-kotlin

gradle-enterprise-api-kotlin

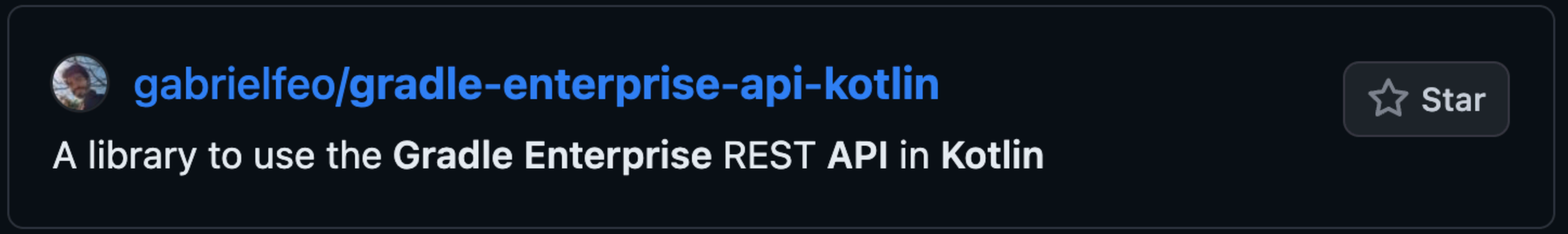


gradle-enterprise-api-kotlin



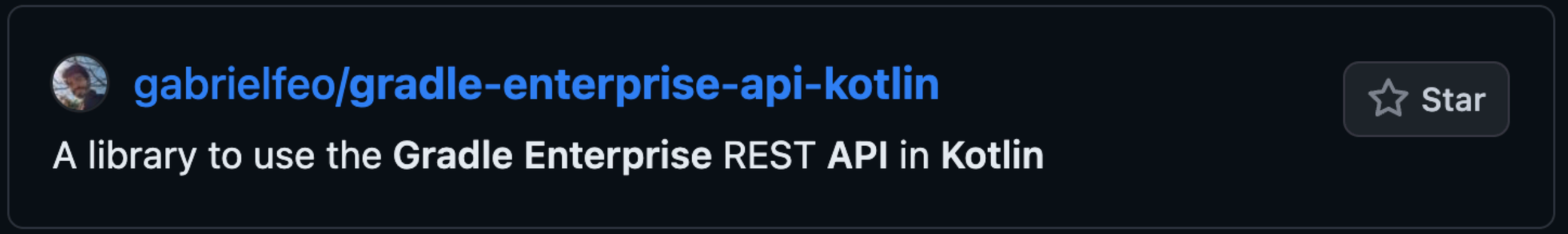
1. Generated from OpenAPI spec

gradle-enterprise-api-kotlin



1. Generated from OpenAPI spec
2. Common utilities

gradle-enterprise-api-kotlin



1. Generated from OpenAPI spec
2. Common utilities
3. Easy to add

gradle-enterprise-api-kotlin

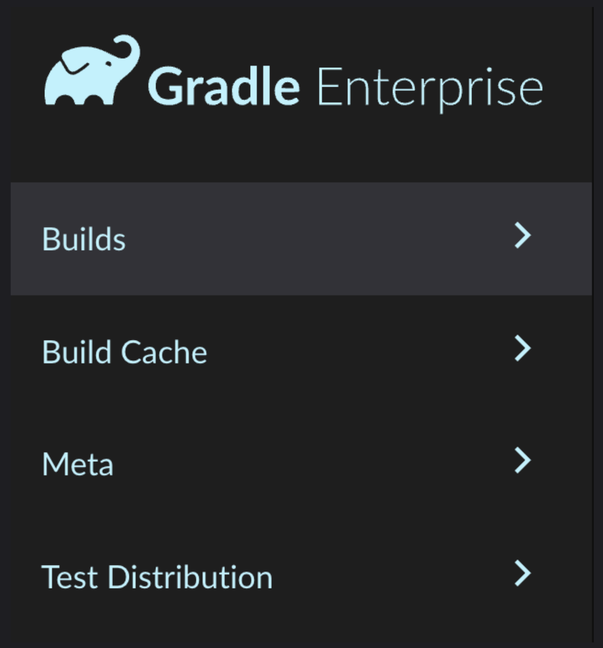
```
val api = GradleEnterpriseApi.newInstance()
```

gradle-enterprise-api-kotlin

```
val api = GradleEnterpriseApi.newInstance()

api.buildsApi
```

=



gradle-enterprise-api-kotlin

```
val api = GradleEnterpriseApi.newInstance()

api.buildsApi.getBuilds(
```

```
@GET
public abstract suspend fun getBuilds(
    @Query since: Long?,
    @Query sinceBuild: String?,
    @Query fromInstant: Long?,
    @Query fromBuild: String?,
    @Query reverse: Boolean?,
    @Query maxBuilds: Int?,
    @Query maxWaitSecs: Int?
): List<Build>
```

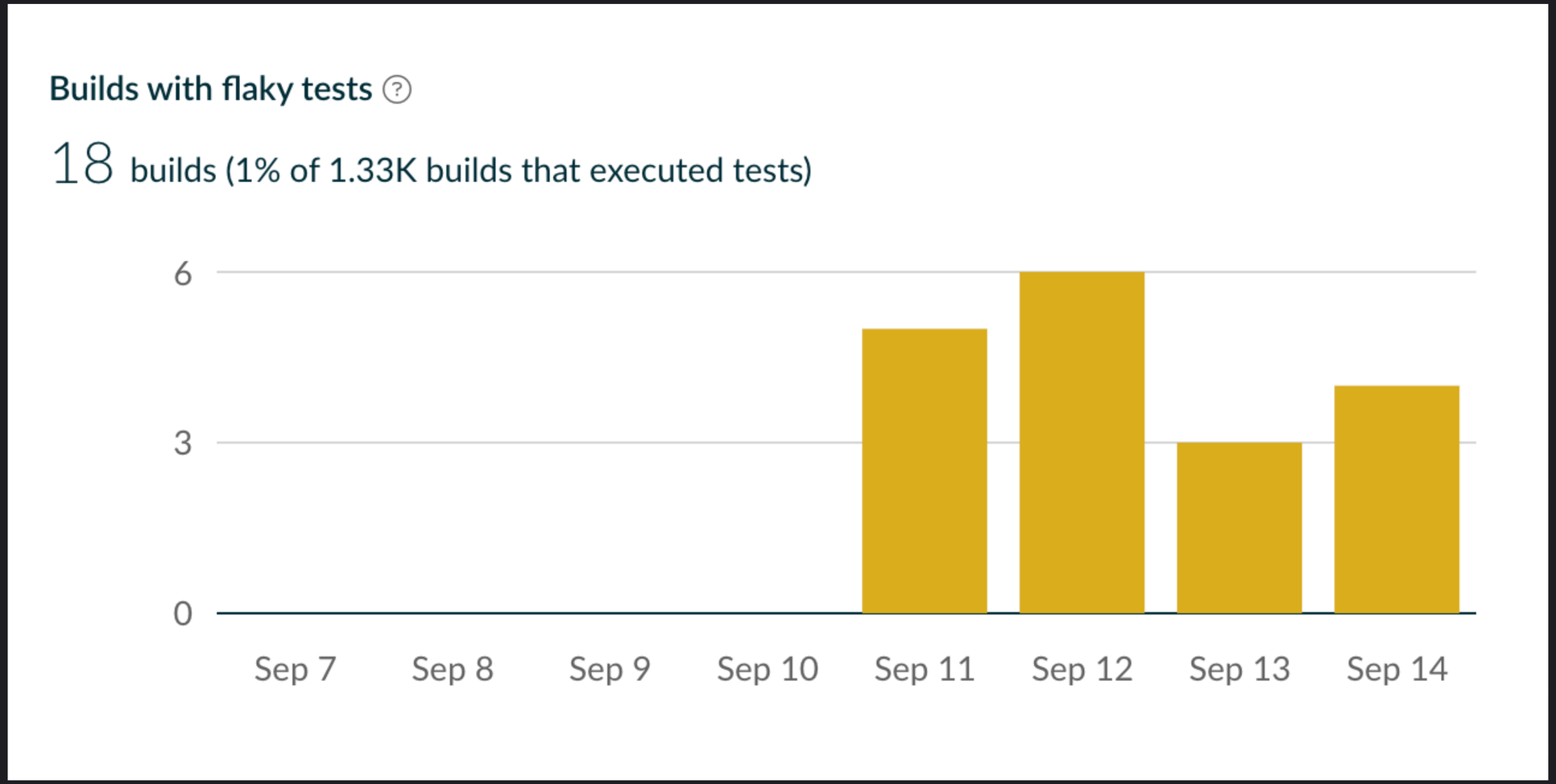
Get a list of builds with the common attributes of a Build Scan. The contained attributes are build tool agnostic. Responses:

API for monitoring

API for monitoring

Automating flaky test tickets

Automating flaky test tickets



Automating flaky test tickets

```
val api = GradleEnterpriseApi.newInstance()

api.testsApi.getTestContainers(
```

Automating flaky test tickets

```
val api = GradleEnterpriseApi.newInstance()

api.testsApi.getTestContainers(
    startTimeMin = yesterday,
```

Automating flaky test tickets

```
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


```
{
  "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


```
{
  "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

```
val api = GradleEnterpriseApi.newInstance()

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

Automating flaky test tickets

```
val api = GradleEnterpriseApi.newInstance()

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

openTickets(response)
```

Jupyter notebooks

Jupyter notebooks

+

Jupyter notebooks

Benefits

Jupyter notebooks

Benefits

1. Re-run a line of code

Jupyter notebooks

1. Re-run a line of code

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

Jupyter notebooks

1. Re-run a line of code

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

Jupyter notebooks

1. Re-run a line of code

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

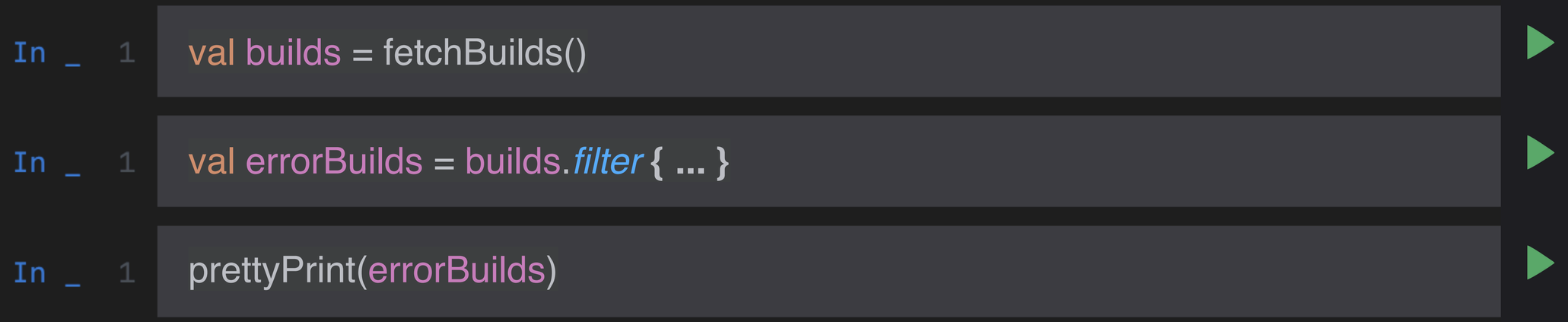
Jupyter notebooks

1. Re-run a line of code

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

Jupyter notebooks

1. Re-run a line of code



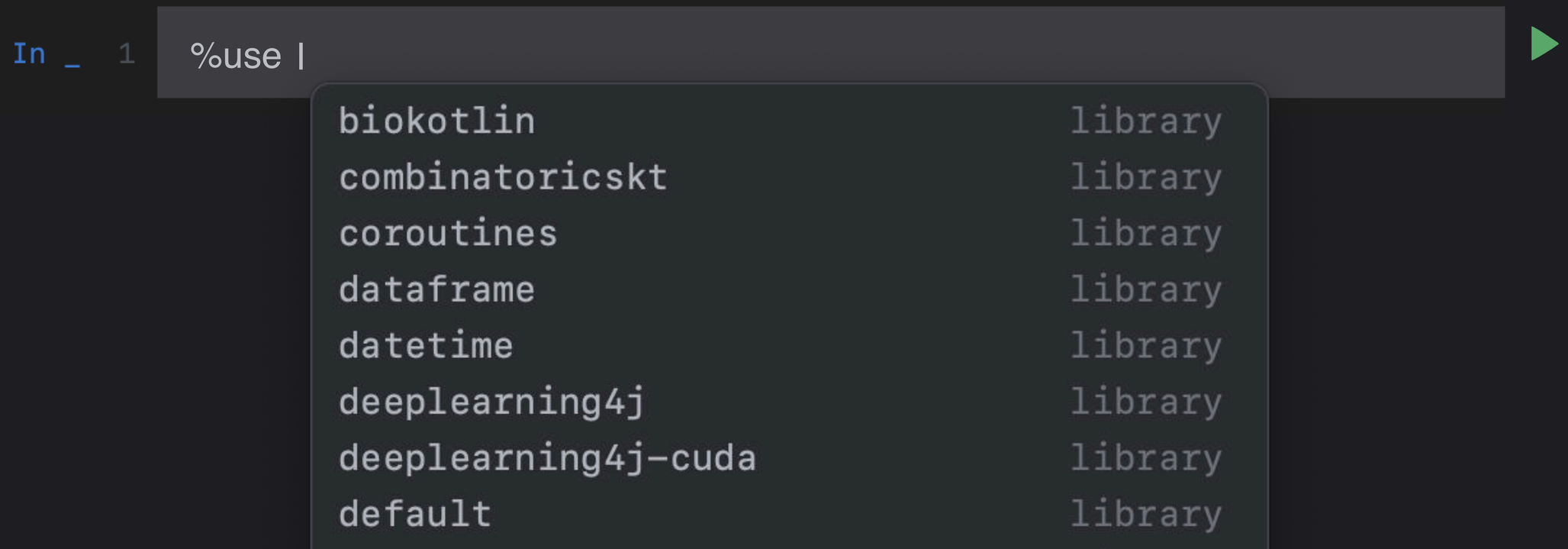
Jupyter notebooks

Benefits

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

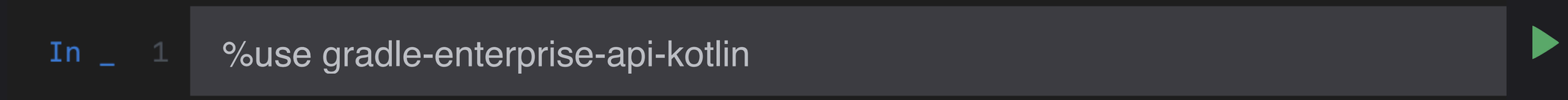
Jupyter notebooks

2. Easy libraries



Jupyter notebooks

2. Easy libraries



Jupyter notebooks

Benefits

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

Jupyter notebooks

3. Rendering

In 2

```
1  JSON("...")
2  MIME("image/jpeg" to "...")
3  HTML("<h3>I'm rendered HTML!</h1>")
```

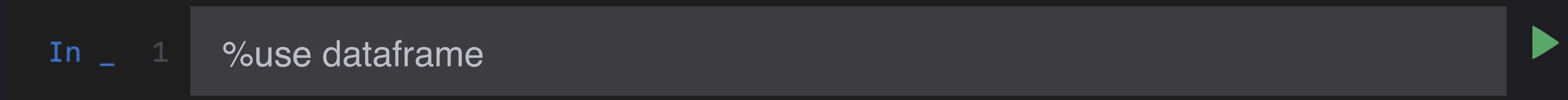
Executed at 2023.09.01 17:20:00 in 50ms

Out 2

I'm rendered HTML!

Jupyter notebooks

2. Easy libraries
3. Rendering



Jupyter notebooks

2. Easy libraries
3. Rendering

In 21

```
1  %use dataframe
2  builds.toDataFrame()
```

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

id	duration	startTime	tags
cuk6ujuhrt3gy	PT7M26.294S	2023-09-18T19:12:48.015Z	[CI, d
chnzaxp6w4knc	PT8M13.01S	2023-09-18T19:12:36.066Z	[CI, d
cjsqidnwdigzy	PT1M45.662S	2023-09-18T19:19:47.199Z	[CI, d

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


5
4
3
2
1
0
Count
Sep 12
Sep 13
Sep 14
Sep 15
Sep 16
Sep 17
Sep 18
Sep 19
Date

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

In [1]:
```
%useLatestDescriptors
%use gradle-enterprise-api-kotlin
%use coroutines, dataframe

val api = GradleEnterpriseApi.newInstance()
```

In [25]:
```
import java.time.*

val start = LocalDate.now().minusWeeks(1).atStartOfDay(ZoneId.of("UTC")).toInstant().toEpochMilli()
data class Build(val attrs: GradleAttributes, val localErrors: Boolean, val remoteErrors: Boolean)
val builds = runBlocking {
    api.buildsApi.getGradleAttributesFlow(since = start).map { attrs ->
        async {
            val cacheInfo = api.buildsApi.getGradleBuildCachePerformance(attrs.id)
            Build(
                attrs,
                localErrors = cacheInfo.buildCaches?.local?.isDisabledDueToError ?: false,
                remoteErrors = cacheInfo.buildCaches?.remote?.isDisabledDueToError ?: false,
            )
        }
    }.toList().awaitAll()
}
```

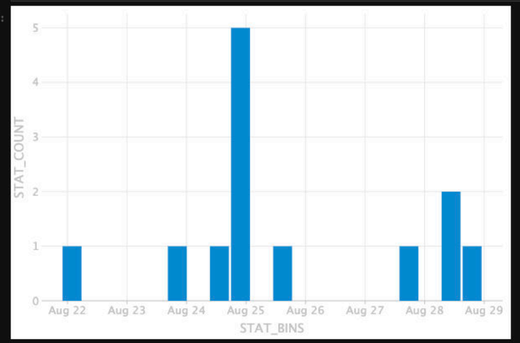
In [26]:
```
val remoteErrorCount = builds.count { it.remoteErrors }
val remoteErrorRatio = remoteErrorCount.toDouble() / builds.size
val localErrorCount = builds.count {it.localErrors }
val localErrorRatio = localErrorCount.toDouble() / builds.size
println("Builds with remote error: $remoteErrorCount / ${builds.size} ($remoteErrorRatio)")
println("Builds with local error: $localErrorCount / ${builds.size} ($localErrorRatio)")
```

```
Builds with remote error: 13 / 7030 (0.001849217638691323)
Builds with local error: 1 / 7030 (1.422475106685633E-4)
```

In [27]:
```
%use kandy

plot {
    histogram(errorDatesList)
}
```

Out[27]:



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

Finding remote cache errors

```
val api = GradleEnterpriseApi.newInstance()

api.buildsApi.getBuilds(
```

Finding remote cache errors

```
val api = GradleEnterpriseApi.newInstance()

api.buildsApi.getBuilds(since = start)
```

Finding remote cache errors

```
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

```
val api = GradleEnterpriseApi.newInstance()

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

Finding remote cache errors

```
val api = GradleEnterpriseApi.newInstance()

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

builds.filter { perfInfo ->
```

Finding remote cache errors

```
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

```
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

```
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()
```

id	duration	startTime	tags	customValues
cuk6ujuhrt3gy	PT7M26.294S	2023-09-18T19:12:48.015Z	[CI, doctor-negative-savings, f..., ...]	[:feature:about-screen:impl:tes...,
chnzaxp6w4knc	PT8M13.01S	2023-09-18T19:12:36.066Z	[CI, doctor-negative-savings, f..., ...]	[:feature:about-screen:impl:tes...,
cjsqidnwdigzy	PT1M45.662S	2023-09-18T19:19:47.199Z	[CI, doctor-fresh-daemon, fresh..., ...]	[CI job=code_coverage, CI job I...
ussqjqvskjzrq	PT1M59.723S	2023-09-18T19:19:47.600Z	[CI, doctor-fresh-daemon, fresh..., ...]	[CI job=android_lint, CI job ID...,

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
cuk6ujuhrt3gy,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
```

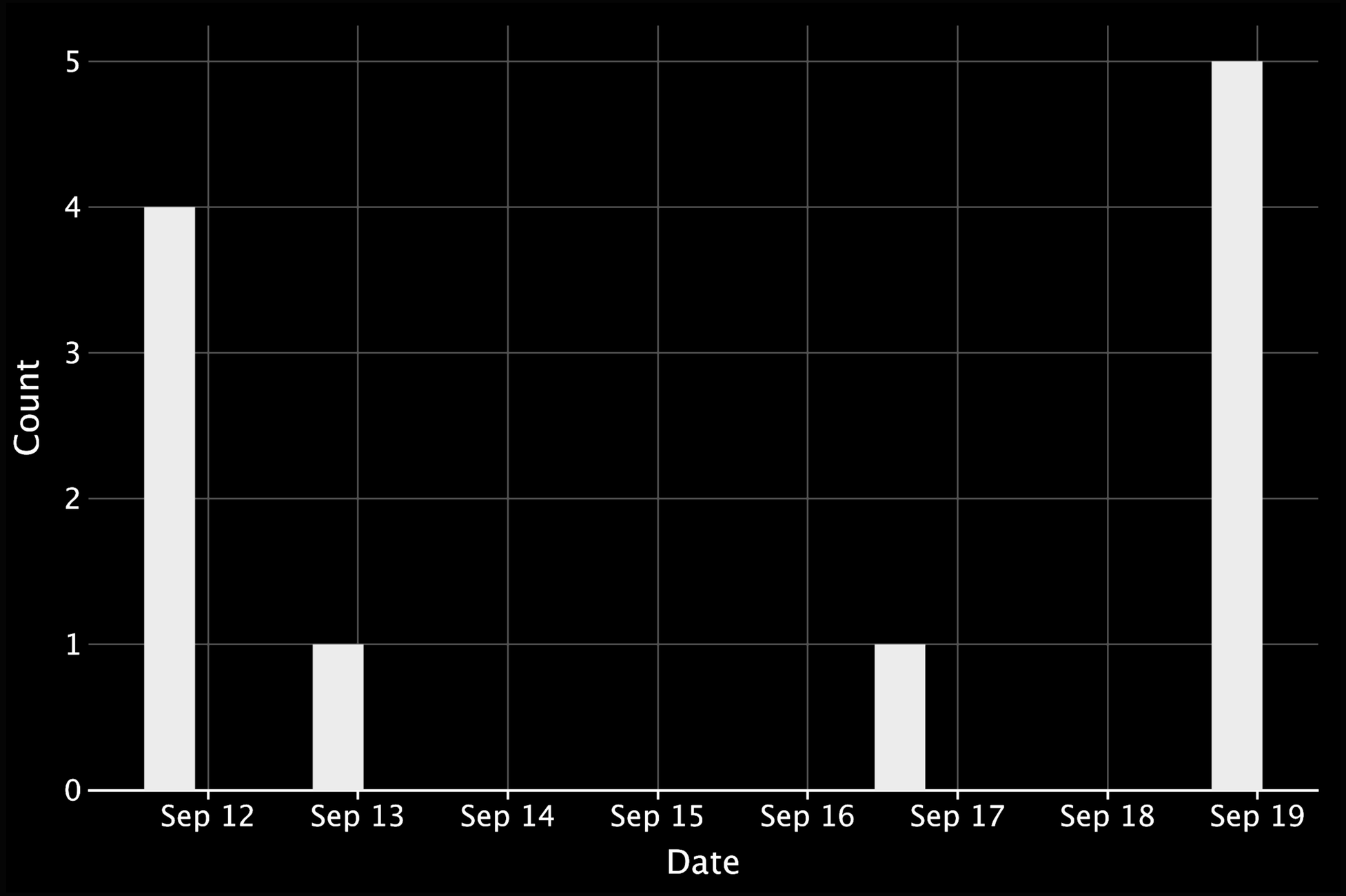
Finding remote cache errors

```
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)
}
```


5
4
3
2
1
0
Count
Sep 12
Sep 13
Sep 14
Sep 15
Sep 16
Sep 17
Sep 18
Sep 19
Date

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
}

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

https://bit.ly/unlocking-build-analytics