Code Talks

Как количество тестов производительности переходит в качество

Спустя годы попыток

Спустя годы попыток отчеты по нагрузке

Спустя годы попыток отчеты по нагрузке интересны читателям

Code Talks

Спустя годы попыток

отчеты по нагрузке

интересны читателям

Автотесты

Автотесты производительности.

Автотесты

производительности,

которых много.

Автотесты

производительности,

которых много,

и это хорошо

Смирнов Вячеслав

Ускоряю miro.com

Развиваю @qa_load

План

1. Чем сложна нагрузка?
2. Как понять результат теста за секунду?
3. Как понять сотню результатов тестов за минуту?
4. 4️⃣ Что дает Test Management System (TMS)?
5. Что дает Test Framework (JUnit) писателю?
6. Почему больше тестов — лучше?

Code Talks

1 Чем сложна нагрузка?

Просто ли

написать

нагрузочные тесты?

А все запросы учтены?

А это пиковый час?

Профиль совпадает?

Мониторинг настроен?

Что по ресурсам?

Места хватит?

Инструмент тянет?

Лимиты настроены?

Предупредили безопасников?

Персональных данных нет?

Обработка ошибок есть?

А это 100к пользователей?

А все запросы учтены?

А это пиковый час?

Профиль совпадает?

Мониторинг настроен?

Что по ресурсам?

Места хватит?

Инструмент тянет?

Лимиты настроены?

Предупредили безопасников?

Персональных данных нет?

Обработка ошибок есть?

А это 100к пользователей?

А все запросы учтены?

А это пиковый час?

Профиль совпадает?

Мониторинг настроен?

Что по ресурсам?

Места хватит?

Инструмент тянет?

Лимиты настроены?

Предупредили безопасников?

Персональных данных нет?

Обработка ошибок есть?

А это 100к пользователей?

А все запросы учтены?

А это пиковый час?

Профиль совпадает?

Мониторинг настроен?

Что по ресурсам?

Места хватит?

Инструмент тянет?

Лимиты настроены?

Предупредили безопасников?

Персональных данных нет?

Обработка ошибок есть?

А это 100к пользователей?

А все запросы учтены?

А это пиковый час?

Профиль совпадает?

Мониторинг настроен?

Что по ресурсам?

Места хватит?

Инструмент тянет?

Лимиты настроены?

Предупредили безопасников?

Персональных данных нет?

Обработка ошибок есть?

А это 100к пользователей?

Непростые

тесты

Просто ли

понять отчет

по нагрузке?

100 графиков

100 ГБайт логов

10 ГБайт трейсов

5 попыток запуска теста

5 попыток запуска теста

10000 ошибок

1000 отличий

10 гипотез

100 графиков

100 ГБайт

10 ГБайт

5 п

ток

10000 ошибок

1000 отличий

10 гипотез

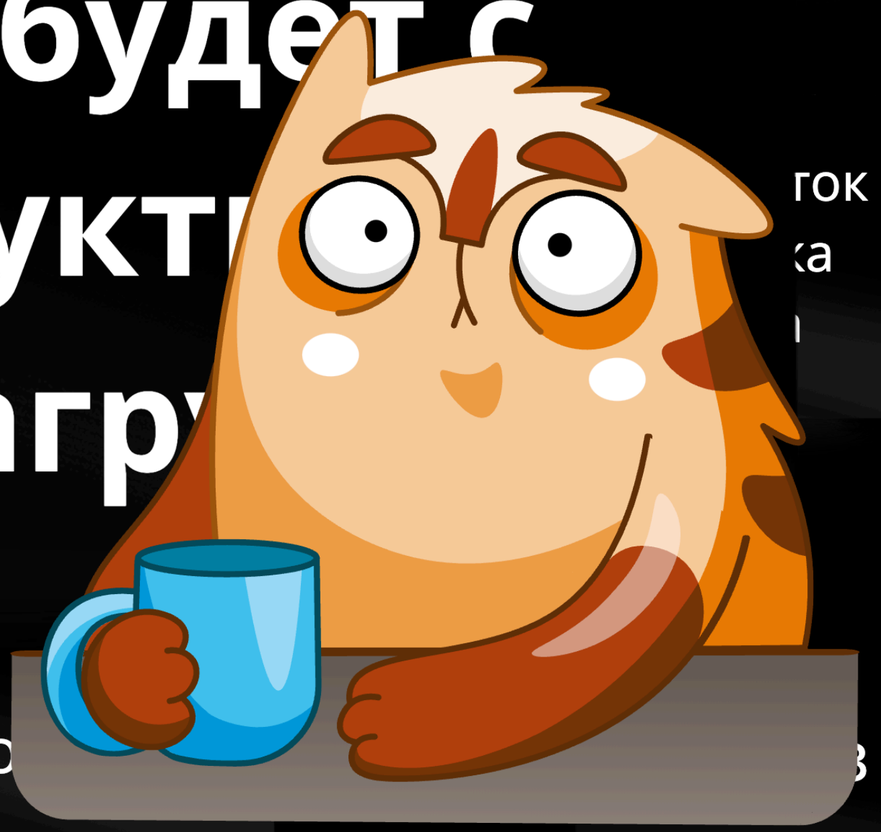
Что будет с продуктивом под нагрузкой?

100
графиков

100
ГБайт

10
ГБайт
ов

Что будет с продукт
под нагр



5 п опыток
за пуска
т

гок
ка

10000
ошибок

о

в

Непростые тесты и результаты

Непростые

тесты и

результаты

Code Talks

Людям

трудно понимать

результаты нагрузки

Code Talks

2 Как понять результат теста за секунду?

Результат одного теста:

✅ OK

🛑 Fail

Результат одного теста:

✅ OK

🛑 Fail

⚠️ не получилось проверить

Результат набора тестов:

✅ 15

✅ 10 🛑 5

🛑 15

⚠️ 15

Результат набора тестов:

✅ 15 — все хорошо

✅ 10 🛑 5

🛑 15

⚠️ 15

Результат набора тестов:

✅ 15

✅ 10 🛑 5 — кое-что сломалось

🛑 15

⚠️ 15

Результат набора тестов:

✅ 15

✅ 10 🛑 5

🛑 15 — все плохо

⚠️ 15

Результат набора тестов:

✅ 15

✅ 10 🛑 5

🛑 15

⚠️ 15 — не удалось проверить

100 графиков

100 ГБайт

10 ГБайт

5 п

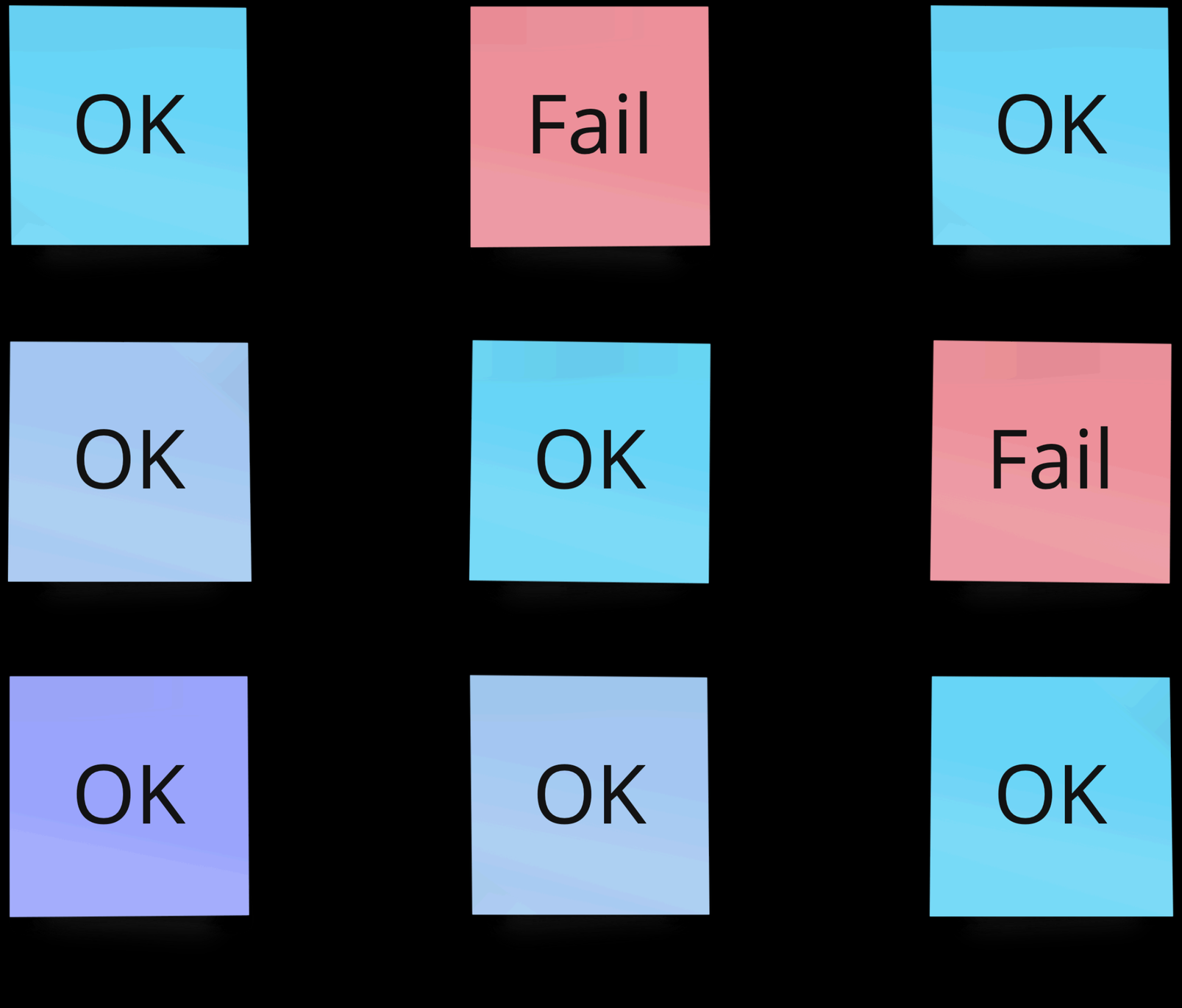
ток

10000 ошибок

1000 отличий

10 гипотез

Что будет с продуктивом под нагрузкой?


OK
Fail
OK
OK
OK
Fail
OK
OK
OK


OK
Fail
OK
OK
OK
OK
Мои изменения ничего не сломали?

OK

Fail

OK

OK

Fail

OK

В результате важны

статус: ✅ 🛑

и **описание**: сервис, версия, ...

Code Talks

3 Как сделать анализ сотни результатов?

Что с сервисами?

Service	Version 1.2.3	Version 1.2.4
API Service	✅250	✅242 🛑8

Что с сервисами?

Service	Version 1.2.3	Version 1.2.4
API Service	✅250	✅242 🛑8

В API Service что-то сломалось в версии 1.2.4

Кликаем по Service API Service

Service	Version 1.2.3	Version 1.2.4
API Service	✅250	✅242 🛑8

и переходим в детали

Что с API?

Endpoint	Version 1.2.3	Version 1.2.4
GET /project	✅50	✅42 🛑8
GET /users	✅100	✅100
GET /account	✅100	✅100

Сбой в GET /project

Endpoint	Version 1.2.3	Version 1.2.4
GET /project	✅50	✅42 🛑8
GET /users	✅100	✅100
GET /account	✅100	✅100

Кликаем по GET /project

Endpoint	Version 1.2.3	Version 1.2.4
GET /project	✅50	✅42 🛑8
GET /users	✅100	✅100
GET /account	✅100	✅100

Что не так с GET /project?

Endpoint	Payload	Version 1.2.3	Version 1.2.4
GET /project	?fields=name	✅10	✅2 🛑8
GET /project	?fields=items	✅20	✅20
GET /project	?fields=permissions	✅20	✅20

Поле name замедлилось

Endpoint	Payload	Version 1.2.3	Version 1.2.4
GET /project	?fields=name	✅10	✅2 🛑8
GET /project	?fields=items	✅20	✅20
GET /project	?fields=permissions	✅20	✅20

Кликаем по payload name ...

Endpoint	Payload	Version 1.2.3	Version 1.2.4
GET /project	?fields=name	✅10	✅2 🛑8
GET /project	?fields=items	✅20	✅20
GET /project	?fields=permissions	✅20	✅20

Проанализировали результаты

- Service: API Service
- Endpoint: GET /project
- Payload: ?fields=name
- Version: 1.2.4

Проанализировали результаты

- Service: API Service
- Endpoint: GET /project
- Payload: ?fields=name
- Version: 1.2.4

за несколько кликов

Code Talks

Таблицы

преобразуют статусы в

аналитический отчет

Code Talks

4 Что дает

Test Management System (TMS)?

Это подход

для разных TMS:

Allure, Qase, TestRail,

TestIT, самодельных ...

Это подход

для разных TMS:

Allure, Qase, TestRail,

TestIT, самодельных ...

OK

Fail

OK

OK

Fail

OK

OK

Fail

OK

OK

Fail

OK

OK

Можно сделать сводный отчет

Endpoint	Payload	Version 1.2.3	Version 1.2.4
GET /project	?fields=name	✅10	✅2 🛑8
GET /project	?fields=items	✅20	✅20
GET /project	?fields=permissions	✅20	✅20

Где есть группировка по

- Service
- Endpoint
- Payload
- Version
- ...

Organize tests

As described in Improving navigation in your test report, Allure supports multiple ways to organize tests into hierarchical structures. Allure JUnit 5 provides functions to assign the

To specify a test's location in the behavior-based hierarchy:

Annotations API | Runtime API

```
import io.qameta.allure.Epic;
import io.qameta.allure.Feature;
import io.qameta.allure.Story;
import org.junit.jupiter.api.Test;

class TestMyWebsite {

    @Test
    @Epic("Web interface")
    @Feature("Essential features")
    @Story("Authentication")
    void testAuthentication() {
        // ...
    }
}
```

To specify a test's location in the suite-based hierarchy:

java

```
import io.qameta.allure.Allure;
import org.junit.jupiter.api.Test;

class TestMyWebsite {

    @Test
    void testAuthentication() {
        Allure.label("parentSuite", "Tests for web interface");
```

Organize tests

As described in Improving navigation in your test report, Allure supports multiple ways to organize tests into hierarchical structures. Allure JUnit 5 provides functions to assign the

allurereport.org/docs/junit5/#organize-tests

Behavior: Epic / Feature / Story

To specify a test's location in the behavior-based hierarchy:

Annotations API

```
import io.qameta.allure.Epic;
import io.qameta.allure.Feature;
import io.qameta.allure.Story;
import org.junit.jupiter.api.Test;

class TestMyWebsite {

    @Test
    @Epic("Web interface")
    @Feature("Essential features")
    @Story("Authentication")
    void testAuthentication() {
        // ...
    }
}
```

To specify a test's location in the suite-based hierarchy:

java

```
import io.qameta.allure.Allure;
import org.junit.jupiter.api.Test;

class TestMyWebsite {

    @Test
    void testAuthentication() {
```

On this page

Organize tests

As described in Improving navigation in your test report, Allure supports multiple ways to organize tests into hierarchical structures. Allure JUnit 5 provides functions to assign the

allurereport.org/docs/junit5/#organize-tests

To specify a test's location in the behavior-based hierarchy:

Annotations API Runtime API

```
import io.qameta.allure.Epic;
import io.qameta.allure.Feature;
import io.qameta.allure.Story;
import org.junit.jupiter.api.Test;

class TestMyWebsite {

    @Test
    @Epic("Web interface")
    @Feature("Essential features")
    @Story("Authentication")
    void testAuthentication() {
        // ...
    }
}
```

To specify a test's location in the suite-based hierarchy:

java

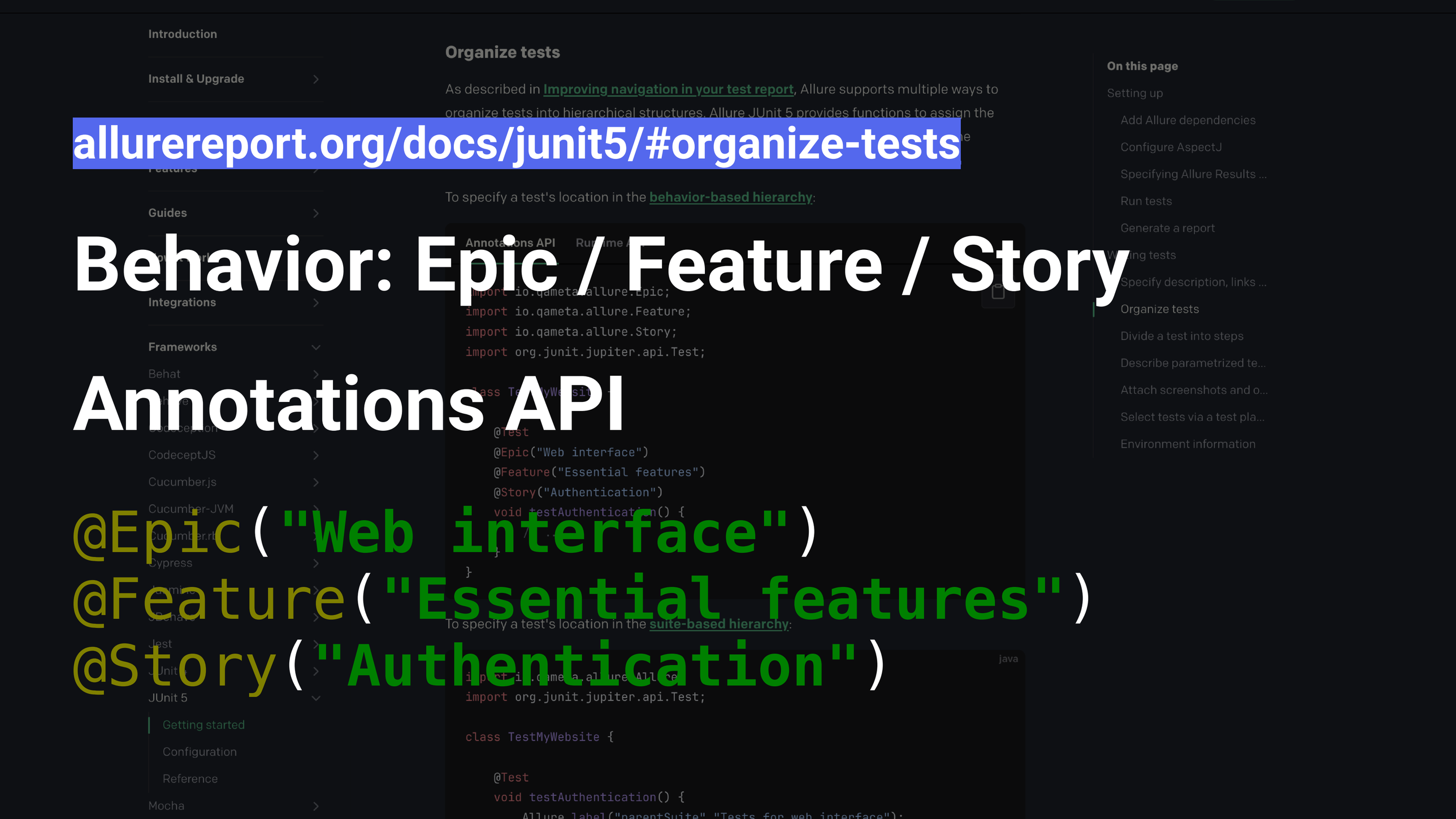
```
import io.qameta.allure.Allure;
import org.junit.jupiter.api.Test;

class TestMyWebsite {

    @Test
    void testAuthentication() {
        Allure.label("parentSuite", "Tests for web interface");
```

Behavior: Epic / Feature / Story

Annotations API



allurereport.org/docs/junit5/#organize-tests

Behavior: Epic / Feature / Story

Annotations API

```
@Epic("Web interface")
@Feature("Essential features")
@Story("Authentication")
```

Organize tests

As described in Improving navigation in your test report, Allure supports multiple ways to organize tests into hierarchical structures. Allure JUnit 5 provides functions to assign the relevant fields to tests either by adding annotations or "dynamically" (same as for the metadata fields).

To specify a test's location in the behavior-based hierarchy:

Annotations API | Runtime API

```
import io.qameta.allure.Epic;
import io.qameta.allure.Feature;
import io.qameta.allure.Story;
import org.junit.jupiter.api.Test;

class TestMyWebsite {

    @Test
    @Epic("Web interface")
    @Feature("Essential features")
    @Story("Authentication")
    void testAuthentication() {
        // ...
    }
}
```

To specify a test's location in the suite-based hierarchy:

java

```
import io.qameta.allure.Allure;
import org.junit.jupiter.api.Test;

class TestMyWebsite {

    @Test
    void testAuthentication() {
        Allure.label("parentSuite", "Tests for web interface");
```

Organize tests

As described in Improving navigation in your test report, Allure supports multiple ways to organize tests into hierarchical structures. Allure JUnit 5 provides functions to assign the

To specify a test's location in the behavior-based hierarchy:

Annotations API | Runtime API

```
import io.qameta.allure.Allure;
import org.junit.jupiter.api.Test;

class TestMyWebsite {

    @Test
    void testAuthentication() {
        Allure.epic("Web interface");
        Allure.feature("Essential features");
        Allure.story("Authentication");
        // ...
    }
}
```

To specify a test's location in the suite-based hierarchy:

java

```
import io.qameta.allure.Allure;
import org.junit.jupiter.api.Test;

class TestMyWebsite {

    @Test
    void testAuthentication() {
        Allure.label("parentSuite" "Tests for web interface");
        Allure.suite("Tests for essential features");
        Allure.label("subSuite", "Tests for authentication");
```

Organize tests

As described in Improving navigation in your test report, Allure supports multiple ways to organize tests into hierarchical structures. Allure JUnit 5 provides functions to assign the

allurereport.org/docs/junit5/#organize-tests

Behavior: Epic / Feature / Story

To specify a test's location in the behavior-based hierarchy:

Annotations API | Runtime API

```
import io.qameta.allure.Allure;
import org.junit.jupiter.api.Test;

class TestMyWebsite {

    @Test
    void testAuthentication() {
        Allure.epic("Web interface");
        Allure.feature("Essential features");
        Allure.story("Authentication");
        // ...
    }
}
```

To specify a test's location in the suite-based hierarchy:

java

```
import io.qameta.allure.Allure;
import org.junit.jupiter.api.Test;

class TestMyWebsite {

    @Test
    void testAuthentication() {
        Allure.label("parentSuite" "Tests for web interface");
        Allure.suite("Tests for essential features");
        Allure.label("subSuite", "Tests for authentication");
```

On this page

Organize tests

As described in Improving navigation in your test report, Allure supports multiple ways to organize tests into hierarchical structures. Allure JUnit 5 provides functions to assign the

allurereport.org/docs/junit5/#organize-tests

To specify a test's location in the behavior-based hierarchy:

Annotations API | Runtime API

```
import org.junit.jupiter.api.Test;

class TestMyWebsite {

    @Test
    void testAuthentication() {
        Allure.epic("Web interface");
        Allure.feature("Essential features");
        Allure.story("Authentication");
        // ...
    }
}
```

Behavior: Epic / Feature / Story

Runtime API

To specify a test's location in the suite-based hierarchy:

java

```
import io.qameta.allure.Allure;
import org.junit.jupiter.api.Test;

class TestMyWebsite {

    @Test
    void testAuthentication() {
        Allure.label("parentSuite" "Tests for web interface");
        Allure.suite("Tests for essential features");
```

allurereport.org/docs/junit5/#organize-tests

Behavior: Epic / Feature / Story

Runtime API

```
Allure.epic("Web interface");
Allure.feature("Essential features");
Allure.story("Authentication");
```

Organize tests

As described in Improving navigation in your test report, Allure supports multiple ways to organize tests into hierarchical structures. Allure JUnit 5 provides functions to assign the relevant fields to tests either by adding annotations or "dynamically" (same as for the metadata fields).

To specify a test's location in the behavior-based hierarchy:

Annotations API | Runtime API

```
import io.qameta.allure.Allure;
import org.junit.jupiter.api.Test;

class TestMyWebsite {

    @Test
    void testAuthentication() {
        Allure.epic("Web interface");
        Allure.feature("Essential features");
        Allure.story("Authentication");
        // ...
    }
}
```

To specify a test's location in the suite-based hierarchy:

java

```
import io.qameta.allure.Allure;
import org.junit.jupiter.api.Test;

class TestMyWebsite {

    @Test
    void testAuthentication() {
        Allure.label("parentSuite" "Tests for web interface");
        Allure.suite("Tests for essential features");
        Allure.label("subSuite", "Tests for authentication");
```

allurereport.org/docs/gettingstarted-navigation

Epics, **features** and **user stories** are widely-used terms for describing a software's features and organizing tests for said features. Usually, an epic is a large set of features that the team aims to develop and test, and a feature is described as a set of user stories that describe how the software is expected to behave in different scenarios.

Allure adapters for all test frameworks provide ways to indicate a test's epic, feature and user story. If a test framework lets the author define some of these terms natively, e.g., in its own language-agnostic file format, then Allure tries its best to incorporate the provided data into the generated tree structure. See the specific Allure adapter's documentation for more details.

Поля по умолчанию

docs.qameta.io/allure-testops/briefly/test-cases/labels/

- parentSuite, suite, subSuite
- testClass, testMethod
- epic, feature, story
- owner, lead
- thread
- layer

Поля по умолчанию

docs.qameta.io/allure-testops/briefly/test-cases/labels/

- parentSuite, suite, subSuite
- testClass, testMethod
- epic, feature, story
- owner, lead
- thread
- layer

List of standard labels used in Allure Framework

Label	Occurrence*	Allure Report v.2	Allure TestOps	Comment
ALLURE_ID	first	✗	✓	
[illegible]	[illegible]	✗	✓	Deprecated, use `ALLURE_ID` instead
[illegible]	[illegible]	✓	✗	
testClass	first	✓	✓	
testMethod	first	✓	✓	
parentSuite	all	✓	✓	
suite	all	✓	✓	
subSuite	all	✓	✓	
epic	all	✓	✓	
feature	all	✓	✓	
story	all	✓	✓	
framework	first	✓	✗	No default processing for Allure TestOps, but often used in user-defined mappings
language	first	✓	✗	No default processing for Allure

Задать можно любые метки (label)

service (сервис)

★ `Allure.label("service", "API");`

endpoint (метод сервиса)

```
   Allure.label("service",  "API");
★  Allure.label("endpoint", "GET /projects");
```

payload (параметры)

```
   Allure.label("service",  "API");
   Allure.label("endpoint", "GET /projects");
★  Allure.label("payload",  "fields=item");
```

dataset (тестовые данные)

```
    Allure.label("service",  "API");
    Allure.label("endpoint", "GET /projects");
    Allure.label("payload",  "fields=item");
★   Allure.label("dataset",  "enterprise 10000 users");
```

version (версию сервиса)

```
    Allure.label("service",  "API");
    Allure.label("endpoint", "GET /projects");
    Allure.label("payload",  "fields=item");
    Allure.label("dataset",  "enterprise 10000 users");
★   Allure.label("version",  getServerVersion());
```

environment (стенд)

```
Allure.label("service",  "API");
Allure.label("endpoint", "GET /projects");
Allure.label("payload",  "fields=item");
Allure.label("dataset",  "enterprise 10000 users");
Allure.label("version",  getServerVersion());
Allure.label("environment",     "production");
```

server-location (размещение)

```
Allure.label("service",  "API");
Allure.label("endpoint", "GET /projects");
Allure.label("payload",  "fields=item");
Allure.label("dataset",  "enterprise 10000 users");
Allure.label("version",  getServerVersion());
Allure.label("environment",     "production");
Allure.label("server-location", "USA");
```

client-location (параметры клиента)

```
Allure.label("service",  "API");
Allure.label("endpoint", "GET /projects");
Allure.label("payload",  "fields=item");
Allure.label("dataset",  "enterprise 10000 users");
Allure.label("version",  getServerVersion());
Allure.label("environment",     "production");
Allure.label("server-location", "USA");
Allure.label("client-location", "Europe");
```

Задать можно любые метки (label)

```
Allure.label("service",  "API");
Allure.label("endpoint", "GET /projects");
Allure.label("payload",  "fields=item");
Allure.label("dataset",  "enterprise 10000 users");
Allure.label("version",  getServerVersion());
Allure.label("environment",     "production");
Allure.label("server-location", "USA");
Allure.label("client-location", "Europe");
```

★ ...

В Annotation API

значения всегда задаются в коде

```
@Endpoint("GET /projects");
```

В Runtime API

значения могут быть константами

```
Allure.label("version", "1.2.3");
```

В Runtime API

значения могут быть константами

```
Allure.label("version", "1.2.3");
```

и функциями тоже

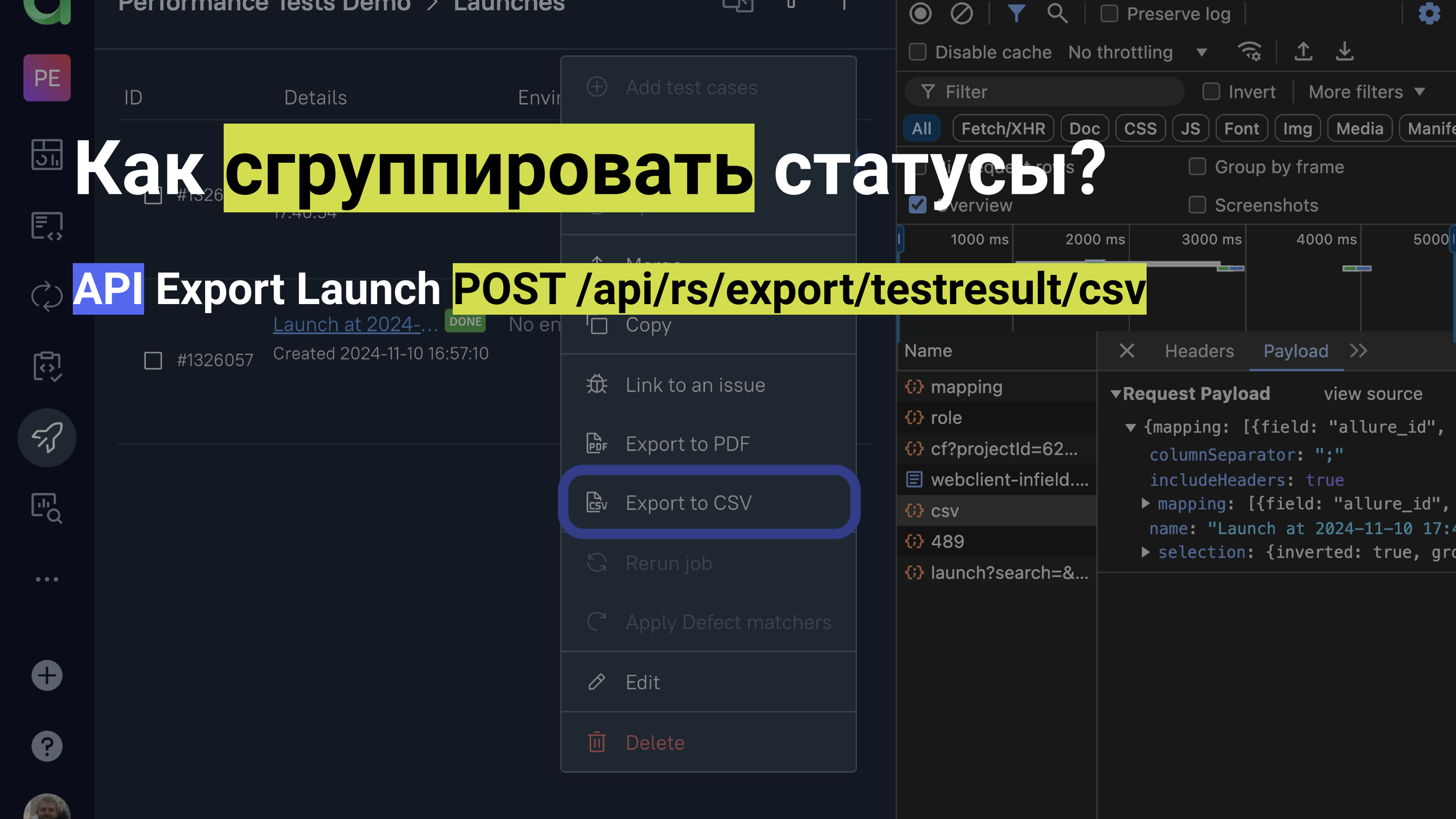
```
Allure.label("version",  getServerVersion());
```

Как сгруппировать статусы?

Endpoint	Payload	Version 1.2.3	Version 1.2.4
GET /project	?fields=name	✅10	✅2 🛑8
GET /project	?fields=items	✅20	✅20
GET /project	?fields=permissions	✅20	✅20

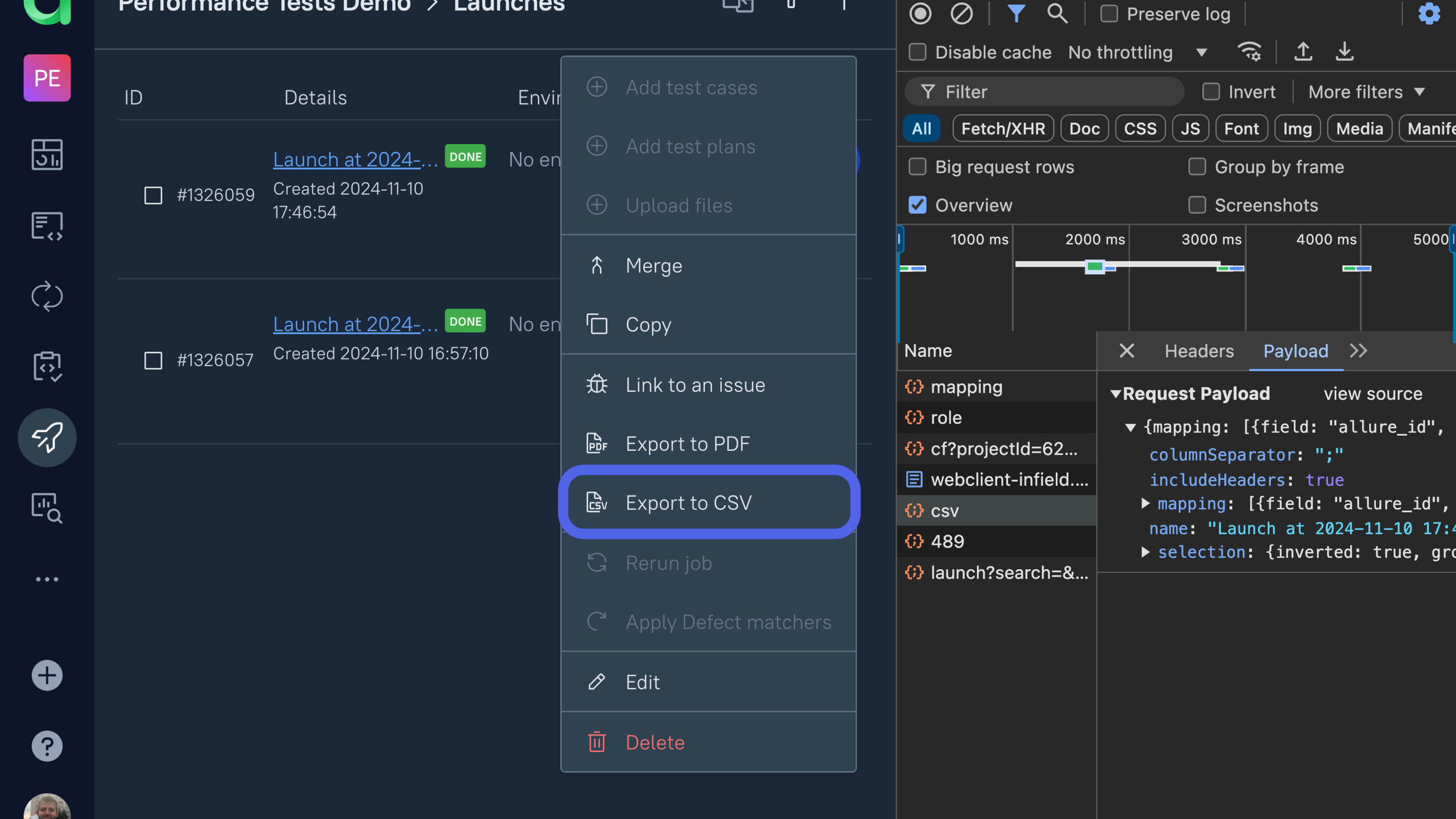
Как сгруппировать статусы?

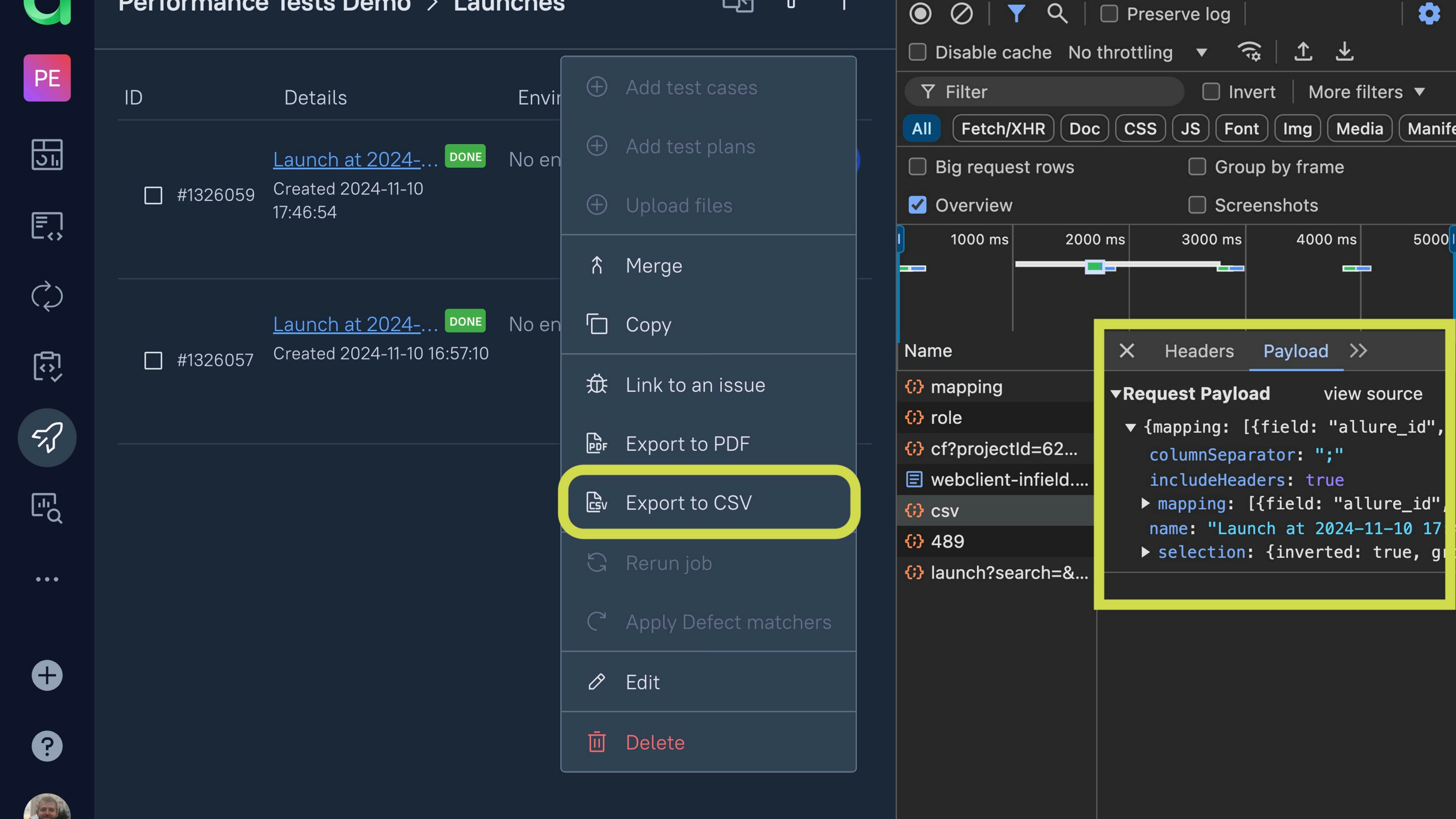
API Export Launch POST /api/rs/export/testresult/csv

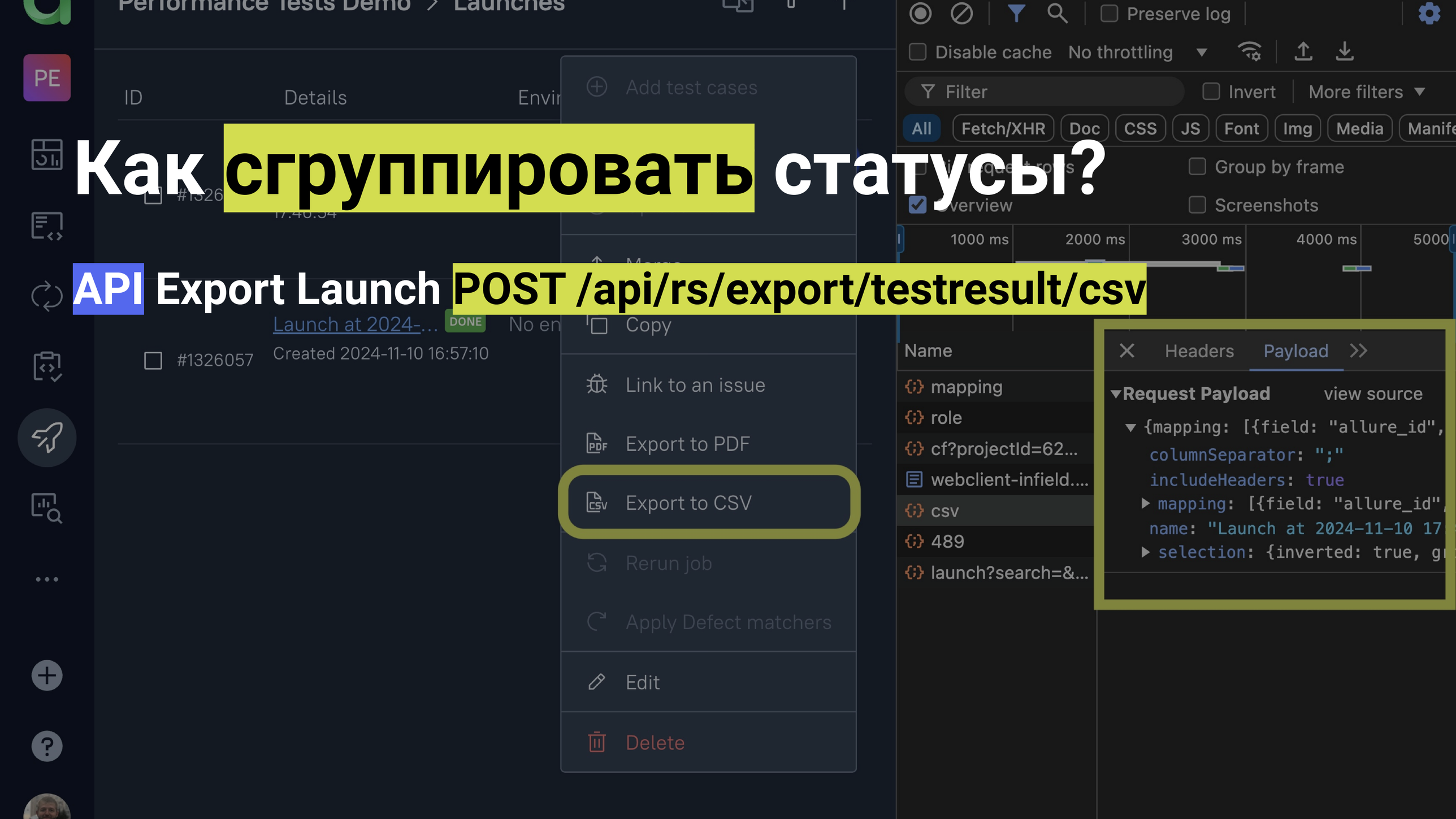


Как сгруппировать статусы?

API Export Launch POST /api/rs/export/testresult/csv


Performance Tests Demo > Launches
ID
Details
Envi
Launch at 2024-...
DONE
No en
#1326059
Created 2024-11-10 17:46:54
Launch at 2024-...
DONE
No en
#1326057
Created 2024-11-10 16:57:10
Add test cases
Add test plans
Upload files
Merge
Copy
Link to an issue
Export to PDF
Export to CSV
Rerun job
Apply Defect matchers
Edit
Delete
Preserve log
Disable cache
No throttling
Filter
Invert
More filters
All
Fetch/XHR
Doc
CSS
JS
Font
Img
Media
Big request rows
Group by frame
Overview
Screenshots
1000 ms
2000 ms
3000 ms
4000 ms
Name
Headers
Payload
mapping
role
cf?projectId=62...
webclient-infield....
csv
489
launch?search=&...
Request Payload
view source
{mapping: [{field: "allure_id",
columnSeparator: ";"
includeHeaders: true
mapping: [{field: "allure_id",
name: "Launch at 2024-11-10 17:
selection: {inverted: true, gro


Performance Tests Demo > Launches
ID
Details
Envi
#1326059
Launch at 2024-...
DONE
No en
Created 2024-11-10 17:46:54
#1326057
Launch at 2024-...
DONE
No en
Created 2024-11-10 16:57:10
Add test cases
Add test plans
Upload files
Merge
Copy
Link to an issue
Export to PDF
Export to CSV
Rerun job
Apply Defect matchers
Edit
Delete
Preserve log
Disable cache
No throttling
Filter
Invert
More filters
All
Fetch/XHR
Doc
CSS
JS
Font
Img
Media
Big request rows
Group by frame
Overview
Screenshots
1000 ms
2000 ms
3000 ms
4000 ms
Name
mapping
role
cf?projectId=62...
webclient-infield....
csv
489
launch?search=&...
Headers
Payload
Request Payload
view source
{mapping: [{field: "allure_id",
columnSeparator: ";"
includeHeaders: true
mapping: [{field: "allure_id"
name: "Launch at 2024-11-10 17
selection: {inverted: true, gr



Как сгруппировать статусы?

API Export Launch **POST /api/rs/export/testresult/csv**

Как сгруппировать статусы?

API Export Launch POST /api/rs/export/testresult/csv

CSV загружается в InfluxDB / VictoriaMetrics / ...

Как сгруппировать статусы?

API Export Launch POST /api/rs/export/testresult/csv

CSV загружается в InfluxDB / VictoriaMetrics / ...

Grafana строит аналитический отчет

Как сгруппировать статусы?

API Export Launch POST /api/rs/export/testresult/csv

CSV загружается в InfluxDB / VictoriaMetrics / ...

Grafana строит аналитический отчет

Grafana показывает детали

Как сгруппировать статусы?

API Export Launch POST /api/rs/export/testresult/csv

CSV загружается в InfluxDB / VictoriaMetrics / ...

Grafana строит аналитический отчет

Grafana показывает детали

Анализ обычного отчета: 2+ часа

Анализ обычного отчета: 2+ часа

Анализ аналитики: от 10-минут

Анализ обычного отчета: 2+ часа

Анализ аналитики: от 10-минут

Выгодно с 12-го теста

Таблицы в Grafana дают читателю легкость анализа

А также в Allure TestOps легко

А также в Allure TestOps легко создать дефект по Fail-тесту

А также в Allure TestOps легко создать дефект по Fail-тесту и за-Mute-ить (игнорировать) сбой

А также в Allure TestOps легко создать дефект по Fail-тесту и за-Mute-ить (игнорировать) сбой

Читатели

не игнорируют

тесты по нагрузке

Читатели

правильно игнорируют

тесты по нагрузке

Читатели

правильно (Issue)

игнорируют (Mute)

Code Talks

Читателям

нравится

простота

Code Talks

5 Что дает JUnit писателю?

Allure интегрирован с JUnit 5

- работает хорошо
- документация
- примеры

В JUnit понятная структура проекта

В JUnit понятная структура проекта

@BeforeAll для подачи нагрузки

- Запуск инструмента (k6/gatling/jmeter/...)
- Ожидание метрик (summary.json)

В JUnit понятная структура проекта

@BeforeAll для подачи нагрузки

- Запуск инструмента (k6/gatling/jmeter/...)
- Ожидание метрик (summary.json)

@Test для проверок

- Проверка метрик (summary.json)
- Добавление деталей (ссылки на Grafana)

В JUnit понятная структура проекта

@BeforeAll

- Запуск инструмента
- Ожидание метрик

@Test

- Проверка метрик
- Добавление деталей

JUnit написан на Java

JUnit написан на Java

Некоторые инструменты запустить легко

- JMeter-Java-DSL
- Gatling

JUnit написан на Java

Некоторые инструменты запустить легко

- JMeter-Java-DSL
- Gatling

Любые инструменты запустить возможно

- В Java есть java.testcontainers.org
- Через Test Containers можно запустить Docker-контейнеры
- Внутри Docker можно запустить: k6, Taurus, ...

JUnit может запустить

- JMeter-Java-DSL
- Gatling
- Taurus
- k6
- любой инструмент

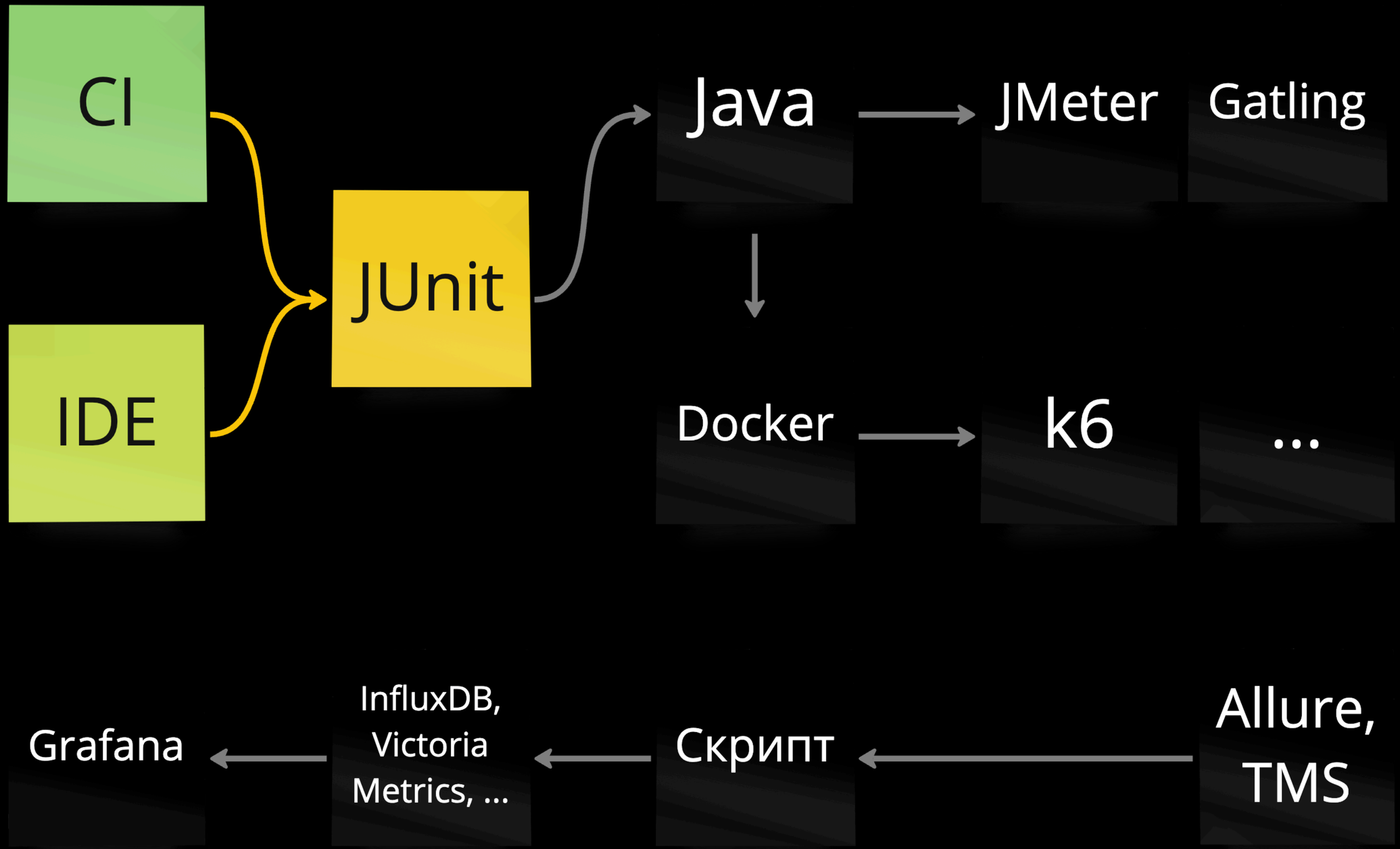
JUnit

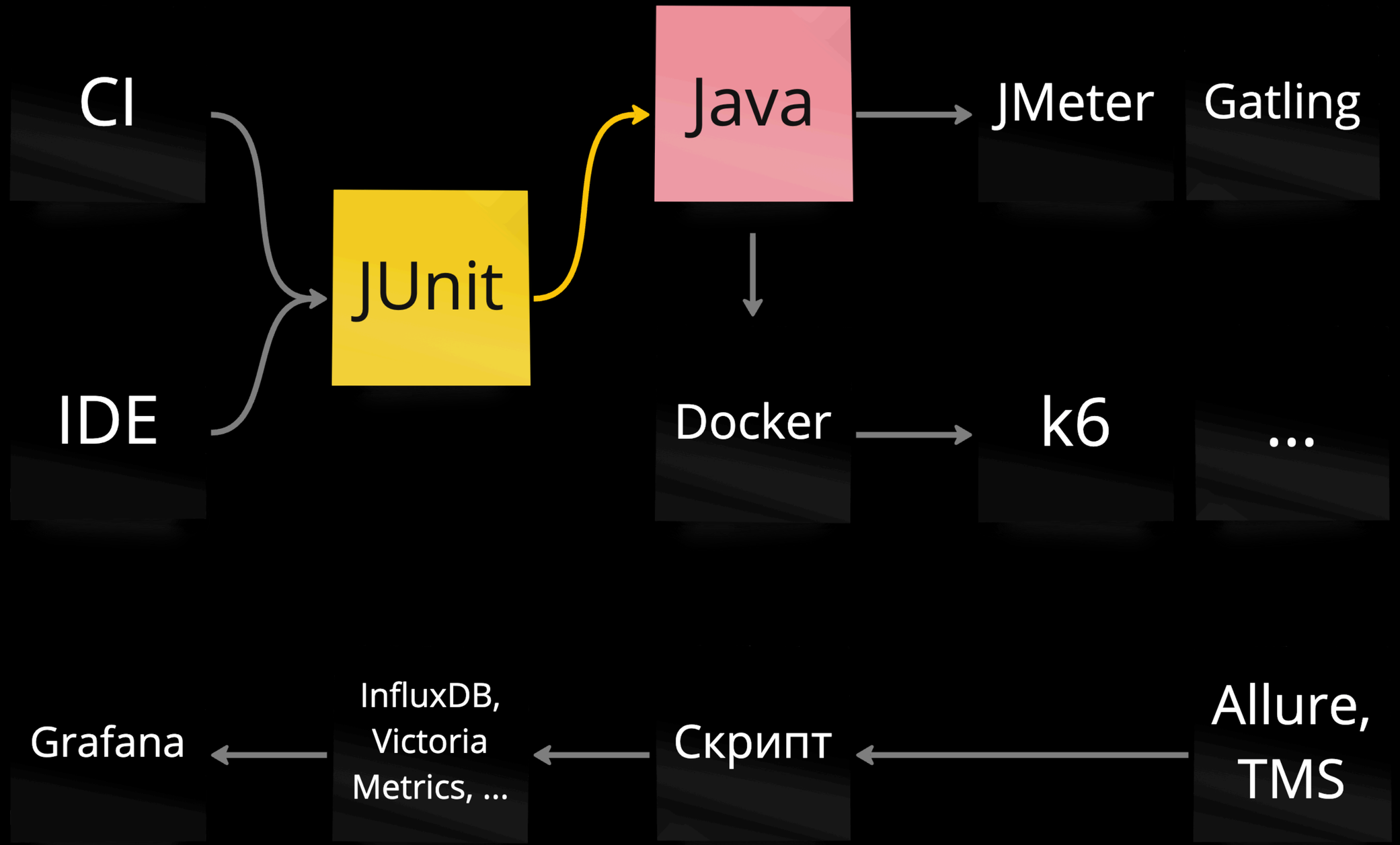
интегрирован в

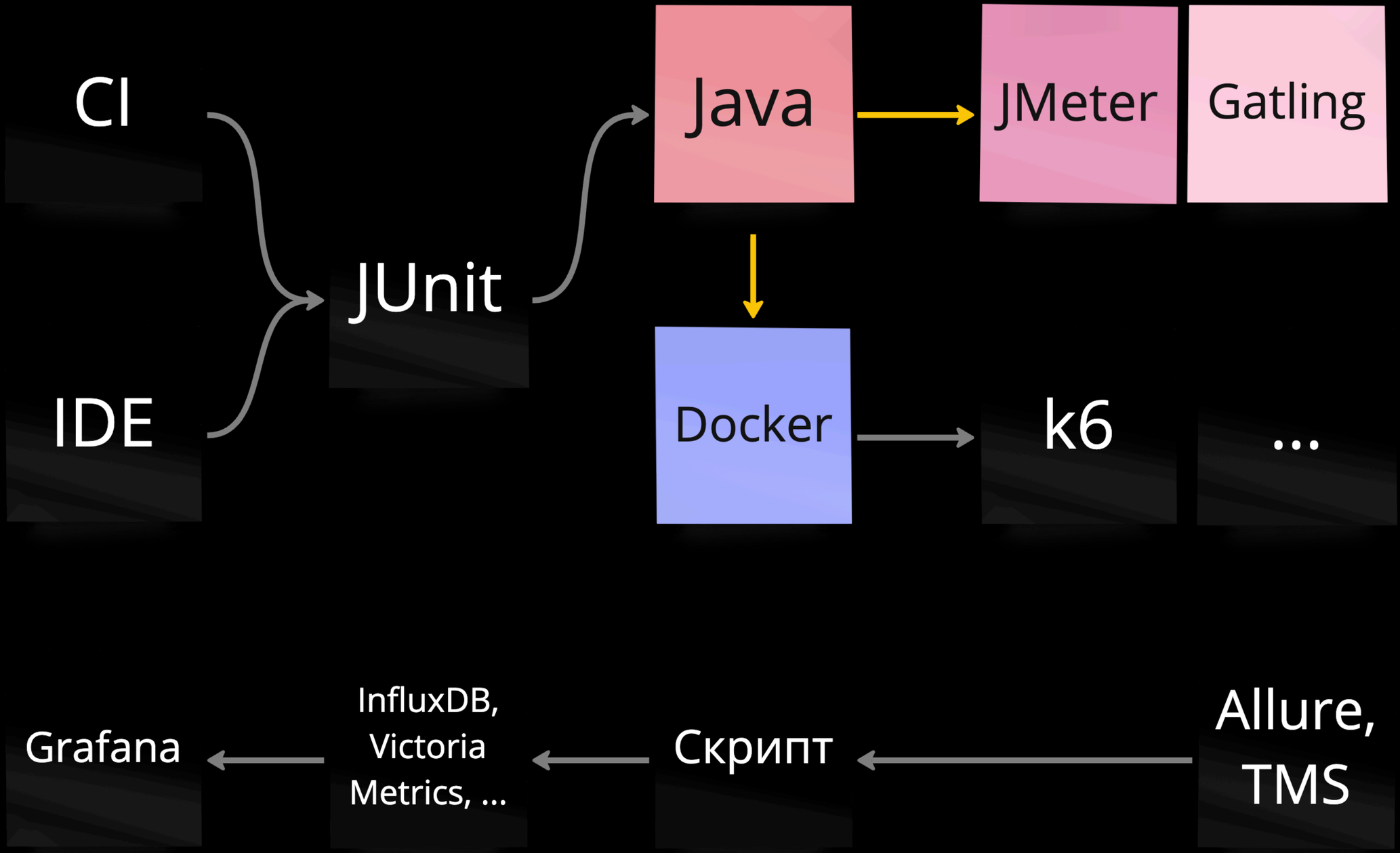
среды разработки IDE

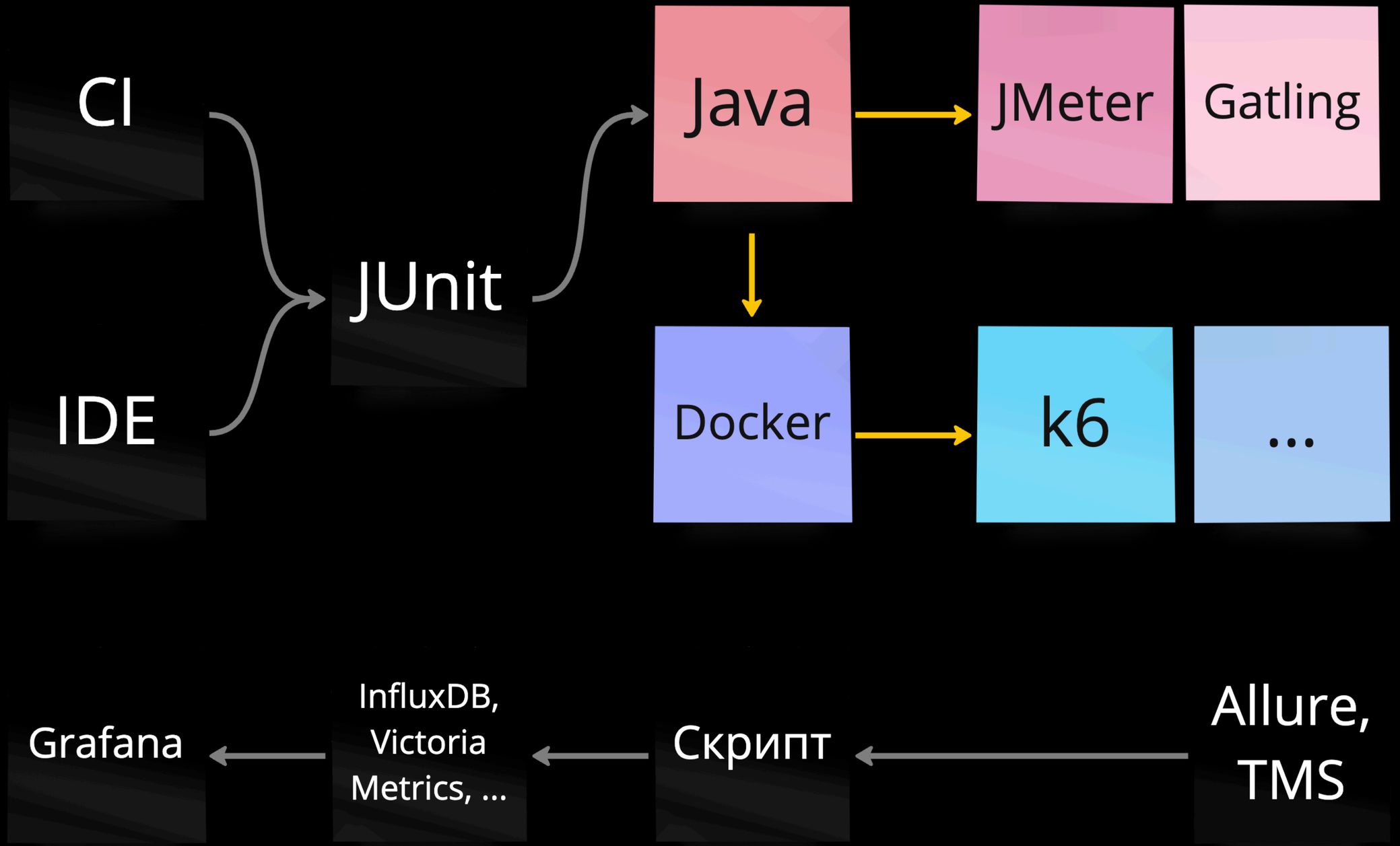
JUnit

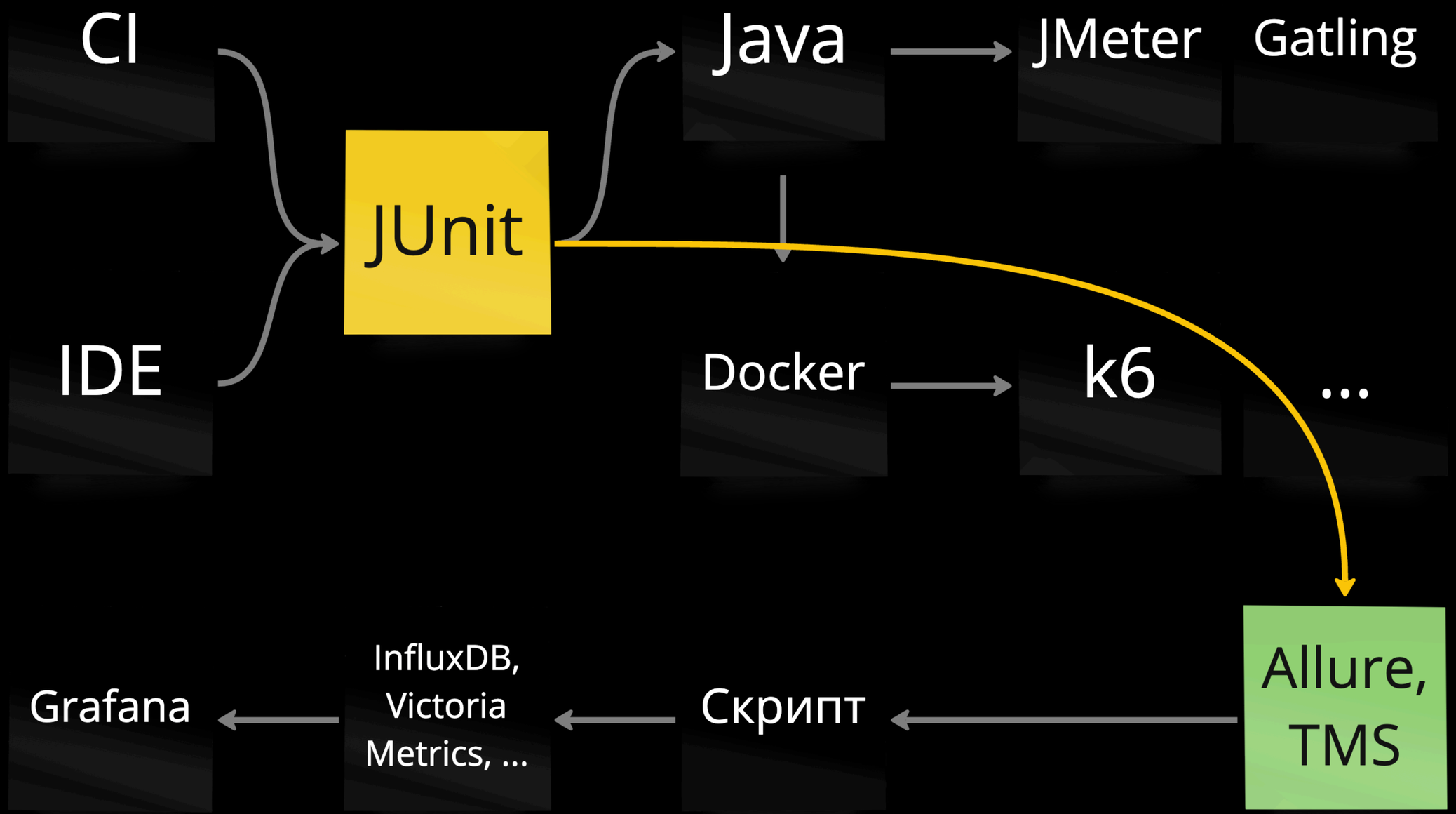
интегрирован в системы сборки CI

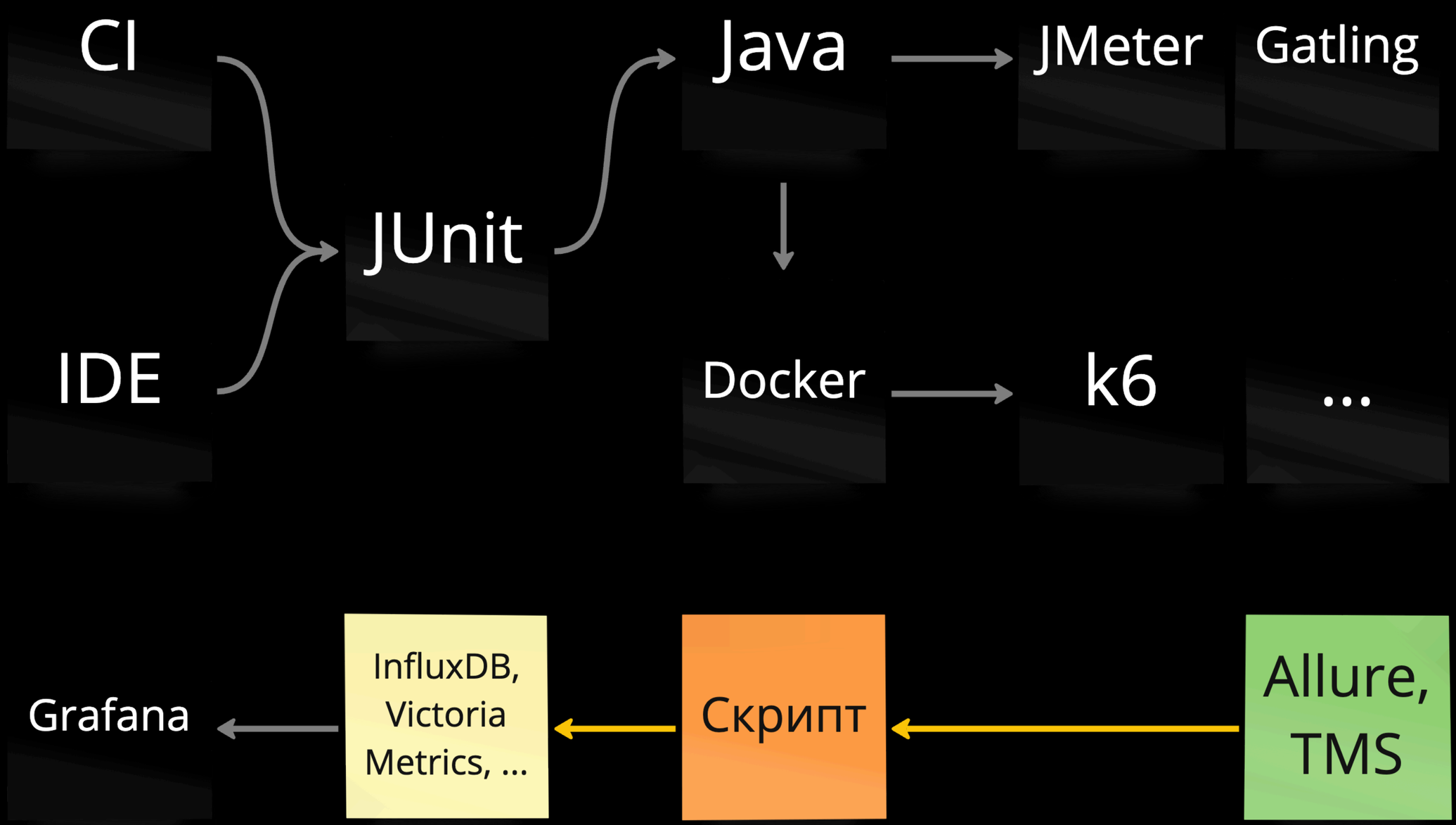

CI
IDE
JUnit
Java
JMeter
Gatling
Docker
k6
...
Grafana
InfluxDB,
Victoria
Metrics, ...
Скрипт
Allure,
TMS

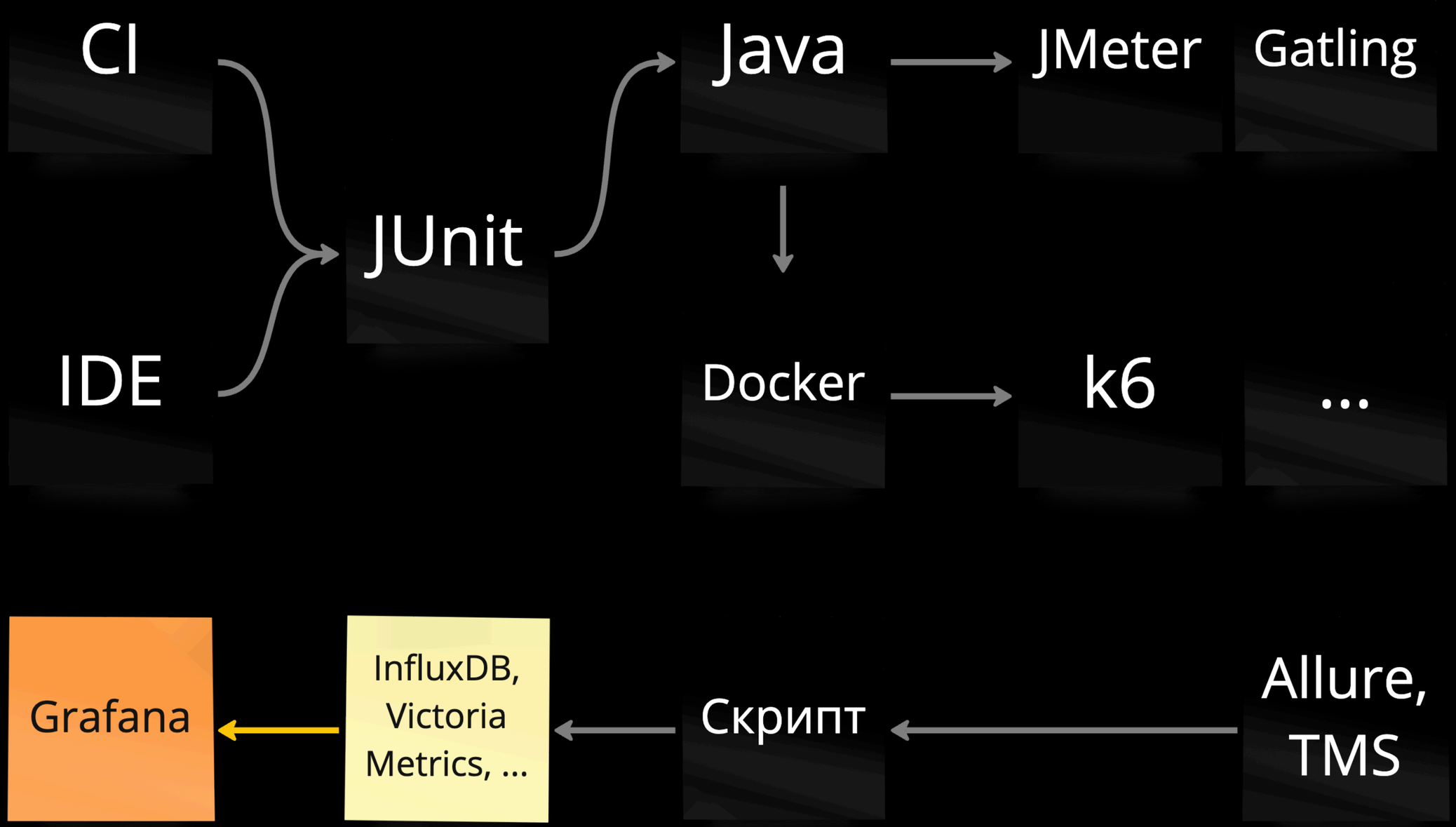

CI
IDE
JUnit
Java
JMeter
Gatling
Docker
k6
...
Grafana
InfluxDB,
Victoria
Metrics, ...
Скрипт
Allure,
TMS

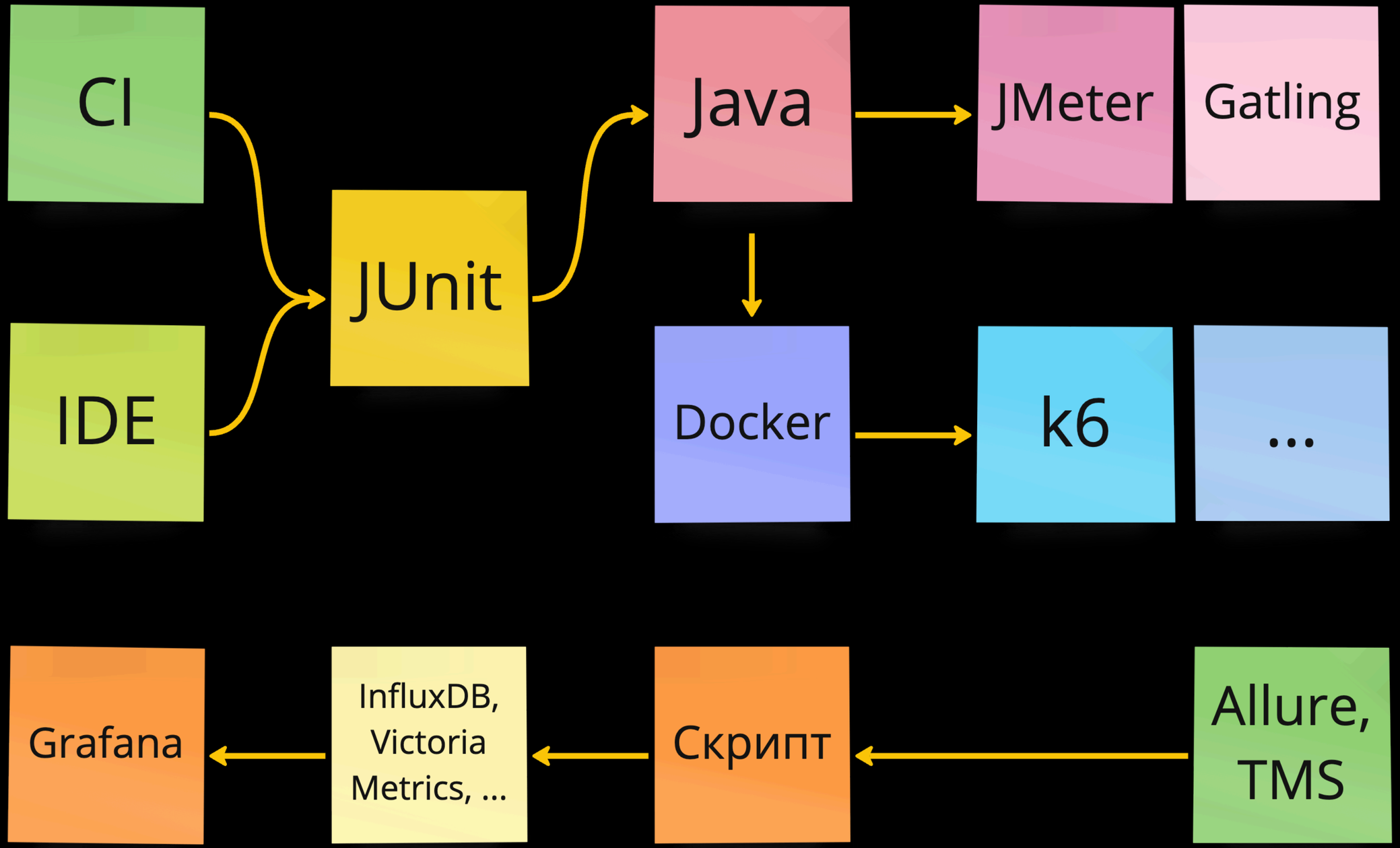

CI
IDE
JUnit
Java
JMeter
Gatling
Docker
k6
...
Grafana
InfluxDB,
Victoria
Metrics, ...
Скрипт
Allure,
TMS


CI
IDE
JUnit
Java
JMeter
Gatling
Docker
k6
...
Grafana
InfluxDB,
Victoria
Metrics, ...
Скрипт
Allure,
TMS


CI
IDE
JUnit
Java
JMeter
Gatling
Docker
k6
...
Allure,
TMS
Скрипт
InfluxDB,
Victoria
Metrics, ...
Grafana


CI
IDE
JUnit
Java
JMeter
Gatling
Docker
k6
...
Grafana
InfluxDB,
Victoria
Metrics, ...
Скрипт
Allure,
TMS


CI
IDE
JUnit
Java
JMeter
Gatling
Docker
k6
...
Grafana
InfluxDB,
Victoria
Metrics, ...
Скрипт
Allure,
TMS


CI
IDE
JUnit
Java
JMeter
Gatling
Docker
k6
...
Grafana
InfluxDB,
Victoria
Metrics, ...
Скрипт
Allure,
TMS

Фреймворк задает структуру

JUnit задает структуру

JUnit задает структуру

Статистика тестируется

JUnit задает структуру

k6 summary.json тестируется

JUnit задает структуру

k6 summary.json тестируется

Шлем результаты в хранилище

JUnit задает структуру

k6 summary.json тестируется

Шлем CSV в InfluxDB

JUnit задает структуру

k6 summary.json тестируется

Шлем CSV в InfluxDB

Grafana строит отчет

JUnit задает структуру

k6 summary.json тестируется

Шлем CSV в InfluxDB

Grafana строит отчет

Code Talks

6 Почему больше тестов — лучше?

Много тестов

увеличивают

тестовое покрытие

Много тестов

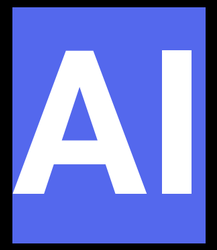
стандартизируют

подачу нагрузки

Шаблонные тесты

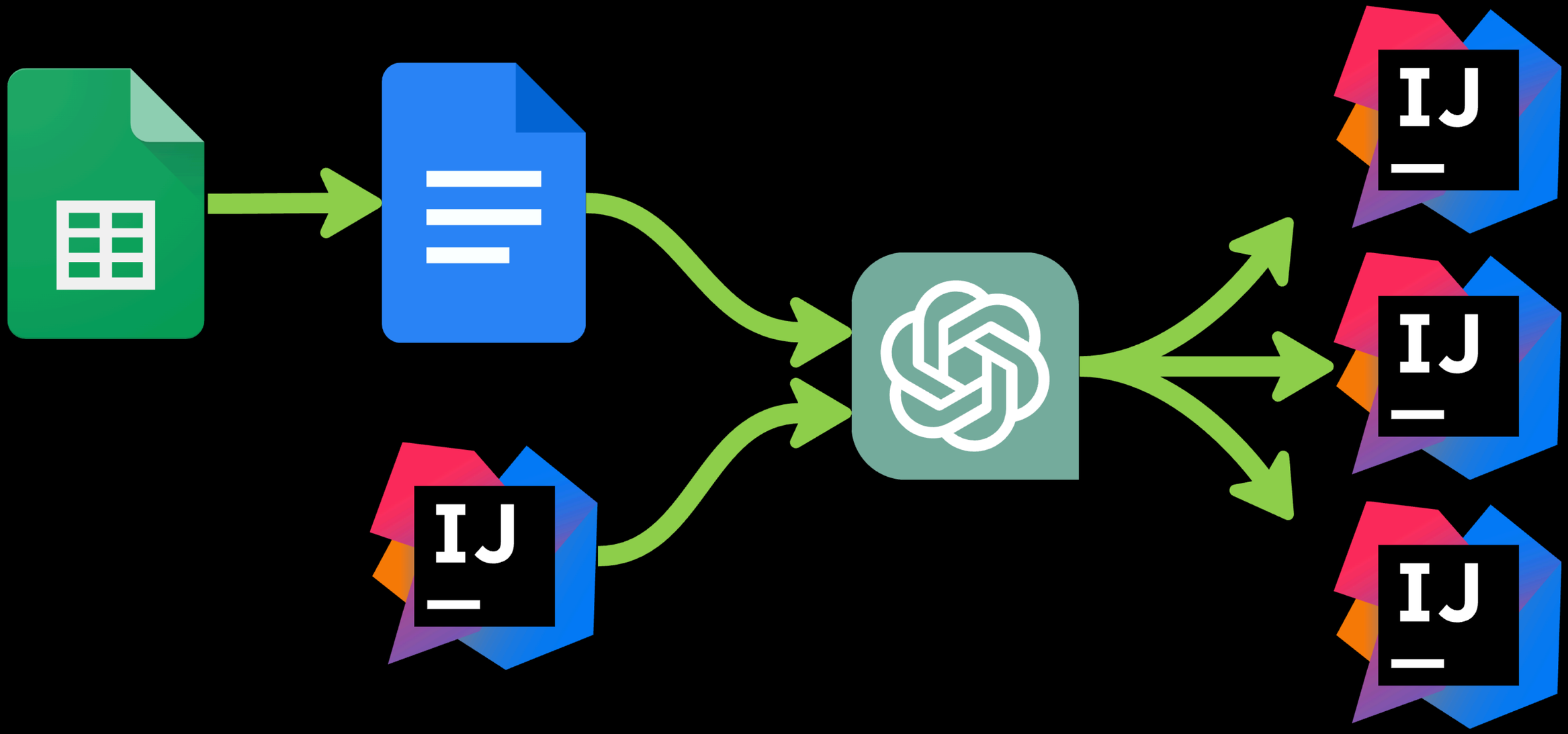
генерируются

в 2024-м году



легко генерирует код

из списка параметров



Code Talks

Сотня простых тестов

долго выполняется,

да быстро понимается

Понравилось? ✅OK 🛑Fail

@qa_load

@smirnovqa

github.com/polarnik/k6-junit-integration

polarnik.github.io/k6-junit-integration/codetalks