

Три кита web-автоматизации



Простота

Доверие

КОНТРОЛЬ

Три кита web-автоматизации



Артем

Ерошенко

Автоматизация web-тестирования



A wooden table with a spiral-bound notebook and a card. The card has a diagram with red arrows. Three white rounded rectangles with black borders are overlaid on the image, containing the text 'Простота', 'Доверие', and 'Контроль' from top to bottom.

Простота

Доверие

КОНТРОЛЬ

A wooden table with a map and a document. The map is on the left, showing a city layout with red and yellow markers. The document is on the right, with some text and a logo. The background is a plain wall.

Простота

Доверие

КОНТРОЛЬ

A collection of various pliers and wire cutters is arranged on a light-colored, textured surface. The tools include: a pair of long-handled pliers with yellow handles and a red grip section; a pair of wire cutters with blue and red handles; a pair of side cutters with red handles; a pair of side cutters with black handles and a green grip section; and a pair of side cutters with silver handles. A white rectangular box with a black border is centered over the tools, containing the text "Инструменты разработки".

Инструменты разработки

Инструменты разработки

Работа с браузером

Подготовка данных

Конфигурация тестов

Инъекция зависимостей

Инструменты разработки

A person wearing a blue shirt and a tan tool belt is using a red power drill on a wall. The tool belt is filled with various tools like screwdrivers, pliers, and a hammer. The background is a light-colored wall and a wooden surface.

Работа с браузером

Подготовка данных

Конфигурация тестов

Инъекция зависимостей

[Новости](#) в Санкт-Петербурге в блогах 29 марта, пятница 13 04

1. Путин говорит, что перестал подслушивать [разговоры после ухода из КГБ](#)
2. Пилотируемый [корабль "Союз" успешно причалил к МКС](#)
3. СК возбудил дело о мошенничестве [с электронными браслетами](#)
4. 19 пострадавших в аварии [под Череповцом остаются в больнице](#)
5. В ГАБТе отказались комментировать [информацию об увольнении Цискаридзе](#)



ООО НКО «Яндекс.Деньги»

[Бесплатная MasterCard Gold](#)

на три года

[Почта](#) [Завести ящик](#)

 чужой компьютер
 [В](#) [f](#) [т](#) [...](#)
[вспомнить пароль](#) [↑](#)

Яндекс

Найдётся всё

[Поиск](#) [Карты](#) [Маркет](#) [Новости](#) [Словари](#) [Картинки](#) [Видео](#) [Музыка](#) [Перевод](#) [Блоги](#) [ещё](#)

Например, [центр Солнца пересекает небесный экватор](#)

Удобный [Яндекс.Браузер](#)

VESTEL **То, что вы ищете!** Холодильник VNF 386 LWE **18 990 руб.** Стиральная машина ARWM 1041 L **9 900 руб.**

[Карта Санкт-Петербурга](#)

[Погода](#) -1 °C
вечером 0, ночью -5

[Пробки](#) 5 баллов
[Движение плотное](#)

- [Маркет](#) — смартфоны до 9 тыс.
 - [Авто](#) — фото новинок
 - [Работа](#) — зарплата от 50 000
 - [М² Недвижимость](#) — купить участок
- [Диск](#) [Деньги](#) [Фотки](#) [Игры](#) [Расписания](#)

[Телепрограмма](#) [Афиша](#)

09:40 [Красота...](#) Невский канал - Домашний

11:50 [Иллюзия охоты](#) ТВ Центр - ОТВ

12:40 [Наука 2.0. Большой...](#) Россия 2

	сегодня	завтра
USD ЦБ	30,9962 +0,0872	31,0834
EUR ЦБ	39,6627	
Нефть	-0,14%	109,77 29/03

[Новости](#) в Санкт-Петербурге в блогах 29 марта, пятница 13 04

1. Путин говорит, что перестал подслушивать [разговоры после ухода из КГБ](#)
2. Пилотируемый [корабль "Союз" успешно причалил к МКС](#)
3. СК возбудил дело о мошенничестве с [электронными браслетами](#)
4. 19 пострадавших в аварии [под Череповцом](#) остаются в больнице
5. В ГАБТе отказались комментировать [информацию об увольнении Цискаридзе](#)



ООО НКО «Яндекс.Деньги»

Бесплатная MasterCard Gold

на три года

Почта [Завести ящик](#)

логин

пароль

чужой компьютер

Войти



[вспомнить пароль](#)



Яндекс

Найдётся всё

[Поиск](#) [Карты](#) [Маркет](#) [Новости](#) [Словари](#) [Картинки](#) [Видео](#) [Музыка](#) [Перевод](#) [Блоги](#) [ещё](#)

Например, [центр Солнца пересекает небесный экватор](#)

Удобный [Яндекс.Браузер](#)



Найти



VESTEL

То, что вы ищете!

Холодильник VNF 386 LWE

18 990 руб.

Стиральная машина ARWM 1041 L

9 900 руб.

[Карта Санкт-Петербурга](#)



[Погода](#) -1 °C

вечером 0, ночью -5

[Пробки](#) 5 баллов

[Движение плотное](#)

[Маркет](#) — смартфоны до 9 тыс.

[Авто](#) — фото новинок

[Работа](#) — зарплата от 50 000

[м² Недвижимость](#) — купить участок

[Диск](#) [Деньги](#) [Фотки](#) [Игры](#) [Расписания](#)

[Телепрограмма](#) [Афиша](#)

09:40 [Красота...](#) Невский канал - Домашний

11:50 [Иллюзия охоты](#) ТВ Центр - ОТВ

12:40 [Наука 2.0. Большой...](#) Россия 2

сегодня

завтра

[USD ЦБ](#) 30,9962 +0,0872 **31,0834**

[EUR ЦБ](#) **39,6627**

[Нефть](#) -0,14% **109,77** 29/03

[Life News](#) [Рыбалка GdeKluet.ru](#) [Комсомольская правда](#)

[520 виджетов для России](#)

A large, gnarled, leafless tree stands in a green field at sunset. The sky is a mix of purple, blue, and orange, with a bright sun low on the horizon behind a line of trees. The tree's branches are intricate and dark against the colorful sky.

Древоподобная структура

A close-up photograph of a person's hands using a red and black power drill to work on a light-colored wall. The person is wearing a blue shirt and a tan leather tool belt filled with various tools like screwdrivers, pliers, and a hammer. The background shows a wooden floor and a white trim.

Page elements

[Новости](#) в Санкт-Петербурге в блогах 29 марта, пятница 13 04

1. Путин говорит, что перестал подслушивать [разговоры после ухода из КГБ](#)
2. Пилотируемый [корабль "Союз" успешно причалил к МКС](#)
3. СК возбудил дело о мошенничестве [с электронными браслетами](#)
4. 19 пострадавших в аварии [под Череповцом остаются в больнице](#)
5. В ГАБТе отказались комментировать [информацию об увольнении Цискаридзе](#)



ООО НКО «Яндекс.Деньги»

[Бесплатная MasterCard Gold](#)

на три года

[Почта](#) [Завести ящик](#)

логин

пароль

чужой компьютер

[Войти](#) [В](#) [f](#) [т](#) [...](#)

[вспомнить пароль](#) [↑](#)

Яндекс
Найдётся всё

[Поиск](#) [Карты](#) [Маркет](#) [Новости](#) [Словари](#) [Картинки](#) [Видео](#) [Музыка](#) [Перевод](#) [Блоги](#) [ещё](#)

Например, [центр Солнца пересекает небесный экватор](#)

[Найти](#)

Например, [центр Солнца пересекает небесный экватор](#)

[Удобный Яндекс.Браузер](#)

VESTEL **То, что вы ищете!** Холодильник VNF 386 LWE **18 990 руб.** Стиральная машина ARWM 1041 L **9 900 руб.**

[Карта Санкт-Петербурга](#)

[Погода](#) -1 °C
вечером 0, ночью -5

[Пробки](#) 5 баллов
[Движение плотное](#)

- [Маркет](#) — смартфоны до 9 тыс.
- [Авто](#) — фото новинок
- [Работа](#) — зарплата от 50 000
- [М² Недвижимость](#) — купить участок
- [Диск](#) [Деньги](#) [Фотки](#) [Игры](#) [Расписания](#)

[Телепрограмма](#) [Афиша](#)

09:40 [Красота...](#) Невский канал - Домашний
11:50 [Иллюзия охоты](#) ТВ Центр - ОТВ
12:40 [Наука 2.0. Большой...](#) Россия 2

	сегодня	завтра
USD ЦБ	30,9962 +0,0872	31,0834
EUR ЦБ	39,6627	
Нефть	-0,14% 109,77	29/03

[Life News](#) [Рыбалка GdeKluet.ru](#) [Комсомольская правда](#)

[520 виджетов для России](#)

Структура элемента

```
public class SearchArrow extends HTMLElement {  
  
    @FindBy(xpath = "//*[id='text']")  
    private HTMLElement requestInput;  
  
    @FindBy(xpath = "//input[@type='submit']")  
    private HTMLElement searchButton;  
  
    public HTMLElement getRequestInput() {  
        return requestInput;  
    }  
  
    public HTMLElement getSearchButton() {  
        return searchButton;  
    }  
  
}
```

Структура элемента

```
public class SearchArrow extends HTMLElement {  
  
    @FindBy(xpath = "//*[id='text']")  
    private HTMLElement requestInput;  
  
    @FindBy(xpath = "//input[@type='submit']")  
    private HTMLElement searchButton;  
  
    public HTMLElement getRequestInput() {  
        return requestInput;  
    }  
    public HTMLElement getSearchButton() {  
        return searchButton;  
    }  
}
```

Структура элемента

```
public class SearchArrow extends HtmlElement {  
  
    @FindBy("xpath = //*[id='text']")  
    private HtmlElement requestInput;  
  
    @FindBy("xpath = //input[@type='submit']")  
    private HtmlElement searchButton;  
  
    public HtmlElement getRequestInput() {  
        return requestInput;  
    }  
    public HtmlElement getSearchButton() {  
        return searchButton;  
    }  
}
```


Структура страницы

```
public class SearchPage {  
  
    @FindBy(" //div[class='search-arrow'] ")  
    private SearchArrow searchArrow;  
  
    public SearchPage(WebDriver driver) {  
        PageFactory.init(driver, this);  
    }  
  
    public SearchArrow getSearchArrow() {  
        return searchArrow;  
    }  
}
```

Структура страницы

```
public class SearchPage {  
  
    @FindBy("//div[class='search-arrow']")  
    private SearchArrow searchArrow;  
  
    public SearchPage(WebDriver driver) {  
        PageFactory.init(driver, this);  
    }  
  
    public SearchArrow getSearchArrow() {  
        return searchArrow;  
    }  
}
```

Структура страницы

```
public class SearchPage {  
  
    @FindBy(" //div[class='search-arrow'] ")  
    private SearchArrow searchArrow;  
  
    public SearchPage(WebDriver driver) {  
        PageFactory.init(driver, this);  
    }  
  
    public SearchArrow getSearchArrow() {  
        return searchArrow;  
    }  
  
}
```

Как ВЫГЛЯДИТ ТЕСТ?

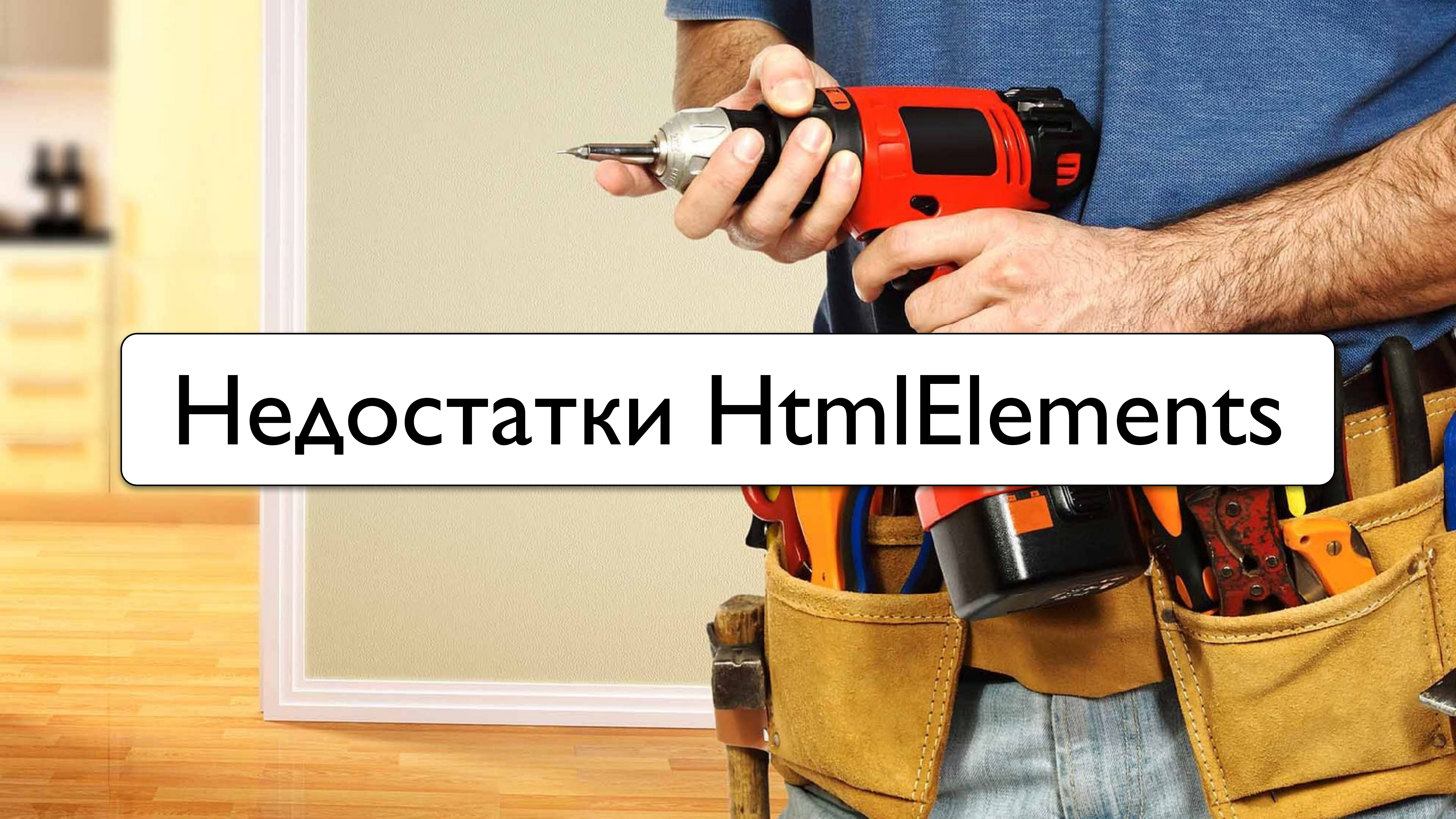
```
public class SimpleTest {  
  
    private SearchPage searchPage = new SearchPage(...);  
  
    @Test  
    public void testUserPage() {  
        searchPage.searchArrow().requestInput()  
            .sendKeys("Heisenbug")  
        searchPage.searchArrow().searchButton()  
            .submit();  
  
        HtmlElement element = searchPage.getSearchResultsBlock();  
        assertThat(element, should(exists()).withWaitFor(10))  
    }  
}
```

Как ВЫГЛЯДИТ ТЕСТ?

```
public class SimpleTest {  
  
    private SearchPage searchPage = new SearchPage(...);  
  
    @Test  
    public void testUserPage() {  
        searchPage.searchArrow().requestInput()  
            .sendKeys("Heisenbug")  
        searchPage.searchArrow().searchButton()  
            .submit();  
  
        HtmlElement element = searchPage.getSearchResultsBlock();  
        assertThat(element, should(exists()).withWaitFor(10))  
    }  
}
```

Как ВЫГЛЯДИТ ТЕСТ?

```
public class SimpleTest {  
  
    private SearchPage searchPage = new SearchPage(...);  
  
    @Test  
    public void testUserPage() {  
        searchPage.searchArrow().requestInput()  
            .sendKeys("Heisenbug")  
        searchPage.searchArrow().searchButton()  
            .submit();  
  
        HtmlElement element = searchPage.getSearchResultsBlock();  
        assertThat(element, should(exists()).withWaitFor(10))  
    }  
}
```

A close-up photograph of a person's hands using a red and black cordless power drill to work on a light-colored wall. The person is wearing a blue shirt and a tan leather tool belt filled with various tools like pliers, a hammer, and a screwdriver. The background shows a wooden floor and a white door frame.

Недостатки HtmlElements

Боль с Ajax страницами

```
public class SimpleTest {  
  
    private SearchPage searchPage = createSearchPage();  
  
    @Test  
    public void testUserPage() {  
        searchPage.searchArrow().requestInput()  
            .sendKeys("Heisenbug")  
        searchPage.searchArrow().searchButton()  
            .submit();  
    }  
}
```

StaleElementReferenceException

Проверки сбоку

```
public class SimpleTest {  
  
    @Test  
    public void testUserPage() {  
        searchPage.open();  
        searchPage.searchArrow().requestInput()  
            .sendKeys("Heisenbug");  
        searchPage.searchArrow().searchButton()  
            .submit();  
  
        HtmlElement element = searchPage.getSearchResultsBlock();  
        assertThat(element, should(exists()).withWaitFor(10))  
    }  
}
```

Отсутствие параметризации

```
public class SearchArrow extends HTMLElement {  
  
    @FindBy(xpath = "//a[contains(text(), 'Поиск')]")  
    private HTMLElement searchLink;  
  
    @FindBy(xpath = "//a[contains(text(), 'Карты')]")  
    private HTMLElement mapsLink;  
  
    @FindBy(xpath = "//a[contains(text(), 'Маркет')]")  
    private HTMLElement marketLink;  
  
}
```

Отсутствие логирования

```
public class BaseSteps {  
  
    @Step("Check {element} is {matcher}")  
    public void checkElement(HtmlElement element, Matcher matcher) {  
        assertThat(element, matcher) ;  
    }  
  
    @Step("Send keys {element} is {matcher}")  
    public void sendKeysToElement(HtmlElement element, String value) {  
        element.sendKeys(value);  
    }  
  
    @Step("Click on {element}")  
    public void clickOnElement(HtmlElement element) {  
        element.click();  
    }  
}
```

eroshenkoam/htmllements

This repository Search Pull requests Issues Marketplace Explore

[eroshenkoam / htmllements](#) Unwatch 6 Unstar 5 Fork 7

Code Issues 10 Pull requests 4 Insights Settings

The next version of HtmlElements [Edit](#)

[Add topics](#)

150 commits 5 branches 16 releases 5 contributors Apache-2.0

Branch: master New pull request Create new file Upload files Find file Clone or download

qameta-ci [maven-release-plugin] prepare for next development iteration Latest commit 7008379 on 18 Sep

.mvn/wrapper	add jenkinsfile	5 months ago
docs/2.0	features	9 months ago
src	fix global parameters for custom blocks (via #21)	3 months ago
.gitignore	add project meta and license information	a year ago
AUTHORS	Update authors	a year ago
Jenkinsfile	update jenkinsfile	5 months ago
LICENSE	Update LICENSE	a year ago
README.md	Remove unused docs	a year ago
mvnw	add jenkinsfile	5 months ago
mvnw.cmd	add jenkinsfile	5 months ago
pom.xml	[maven-release-plugin] prepare for next development iteration	3 months ago

Интерфейсы вместо классов


```
public interface SearchArrow extends HTMLElement {  
  
    @FindBy("//*[@id='text']")  
    HTMLElement requestInput();  
  
    @FindBy("//input[@type='submit']")  
    HTMLElement searchButton();  
}  
  
public interface SearchResult extends HTMLElement {  
  
    @FindBy("//div[class='title']")  
    HTMLElement title();  
}
```

Параметризация

Поиск [Карты](#) [Маркет](#) [Новости](#) [Словари](#) [Картинки](#) [Видео](#) [Музыка](#) [Перевод](#) [Блоги](#) [ещё](#)

Найти

Например, центр Солнца пересекает небесный экватор

 Удобный [Яндекс.Браузер](#)

```
public interface Menu extends HtmlElement {  
  
    @FindBy("//a[contains(text(), '{name}')]")  
    HtmlElement link(@Param("name") String name);  
  
}
```

Проверки с перепопытками

```
public void simpleTest {  
  
    searchPage.searchArrow().requestInput()  
        .sendKeys("Heisenbug")  
    searchPage.searchArrow().searchButton()  
        .submit();  
  
    searchPage.searchResultsBlock()  
        .waitUntil(displayed())  
        .should(hasText(contains("Heisenbug")));  
}
```

Шаблоны страниц

```
public interface WithHeader {  
  
    @FindBy(" //div[@class='header'] ")  
    Header header();  
  
}
```

```
public interface WithFooter {  
  
    @FindBy(" //div[@class='footer'] ")  
    Footer footer();  
  
}
```


Дескрипшен элемента

```
public interface SearchPage extends WebPage {  
  
    @FindBy( "//*[@id='text']" )  
    @Description( "Поисковая строка" )  
    SearchArrow searchArrow();  
  
    @FindBy( "//input[@class='results']" )  
    @Description( "Результаты поиска" )  
    List<SearchResult> searchResults();  
  
}
```

Шаги практически не нужны

- Открываем страницу - [https://\[redacted\]avto.ru/cars/used/sale/10\[redacted\]14-2d5c9](https://[redacted]avto.ru/cars/used/sale/10[redacted]14-2d5c9) 1 parameter 6s 285ms
- ✔ Ждем пока элемент 'Кнопка «Смотреть отчет диагностики»' будет в состоянии [element is displayed on page] 217ms
- ✔ Кликаем на элемент 'Кнопка «Смотреть отчет диагностики»' 4s 189ms
- ✔ Ждем пока элемент 'Краткое содержание сертификата' будет в состоянии [element is displayed on page] 292ms
- ✔ Скроллинг: 2000 px 1 parameter 37ms
- ✔ Ждем пока элемент 'Выползающее меню навигации по сертификату' будет в состоянии [element is displayed on page] 132ms
- ✔ Кликаем на элемент 'Кнопка «Открыть попап»' 229ms
- ✔ Проверяем что элемент 'Кнопка «Закреть попап» в попапе навигации сертификата' в состоянии [element is displayed on page] 168ms
- ✔ Тушим веб драйвер 5 attachments 781ms

Шаги практически не нужны

- Открываем страницу - [https://\[redacted\]avto.ru/cars/used/sale/10\[redacted\]14-2d5c9](https://[redacted]avto.ru/cars/used/sale/10[redacted]14-2d5c9) 1 parameter 6s 285ms
- Ждем пока элемент 'Кнопка «Смотреть отчет диагностики»' будет в состоянии [element is displayed on page] 217ms
- Кликаем на элемент 'Кнопка «Смотреть отчет диагностики»' 4s 189ms
- Ждем пока элемент 'Краткое содержание сертификата' будет в состоянии [element is displayed on page] 292ms
- Скроллинг: 2000 px 1 parameter 37ms
- Ждем пока элемент 'Выползающее меню навигации по сертификату' будет в состоянии [element is displayed on page] 132ms
- Кликаем на элемент 'Кнопка «Открыть попап»' 229ms
- Проверяем что элемент 'Кнопка «Закрыть попап» в попапе навигации сертификата' в состоянии [element is displayed on page] 168ms
- Тушим веб драйвер 5 attachments 781ms

Шаги практически не нужны

- Открываем страницу - <https://avto.ru/cars/used/sale/1014-2d5c9> 1 parameter 6s 285ms
- ✔ Ждем пока элемент 'Кнопка «Смотреть отчет диагностики»' будет в состоянии [element is displayed on page] 217ms
- ✔ Кликаем на элемент 'Кнопка «Смотреть отчет диагностики»' 4s 189ms
- ✔ Ждем пока элемент 'Краткое содержание сертификата' будет в состоянии [element is displayed on page] 292ms
- ✔ Скроллинг: 2000 px 1 parameter 37ms
- ✔ Ждем пока элемент 'Выползающее меню навигации по сертификату' будет в состоянии [element is displayed on page] 132ms
- ✔ Кликаем на элемент 'Кнопка «Открыть попап»' 229ms
- ✔ Проверяем что элемент 'Кнопка «Закрыть попап» в попапе навигации сертификата' в состоянии [element is displayed on page] 168ms
- ✔ Тушим веб драйвер 5 attachments 781ms

Работа с коллекциями

```
@Test
public void testUserPage() {

    // ...
    searchPage.searchResults()
        .filter(WebElement::isDisplayed())
        .convert(SearchResult::getTitle())
        .should(containsInAnyOrder(...));
}
```

Тест ВЫГЛЯДИТ ОЧЕНЬ ПРОСТО

```
public void simpleTest {  
  
    searchPage.open();  
    searchPage.searchArrow().requestInput()  
        .sendKeys("Heisenbug");  
    searchPage.searchArrow().searchButton()  
        .submit();  
  
    searchPage.searchResults()  
        .filter(WebElement::isDisplayed())  
        .convert(SearchResult::getTitle())  
        .should(containsInAnyOrder(...));  
}
```

eroshenkoam/htmllements

This repository Search Pull requests Issues Marketplace Explore

[eroshenkoam / htmllements](#) Unwatch 6 Unstar 5 Fork 7

Code Issues 10 Pull requests 4 Insights Settings

The next version of HtmlElements [Edit](#)

[Add topics](#)

150 commits 5 branches 16 releases 5 contributors Apache-2.0

Branch: master New pull request Create new file Upload files Find file Clone or download

qameta-ci [maven-release-plugin] prepare for next development iteration Latest commit 7008379 on 18 Sep

.mvn/wrapper	add jenkinsfile	5 months ago
docs/2.0	features	9 months ago
src	fix global parameters for custom blocks (via #21)	3 months ago
.gitignore	add project meta and license information	a year ago
AUTHORS	Update authors	a year ago
Jenkinsfile	update jenkinsfile	5 months ago
LICENSE	Update LICENSE	a year ago
README.md	Remove unused docs	a year ago
mvnw	add jenkinsfile	5 months ago
mvnw.cmd	add jenkinsfile	5 months ago
pom.xml	[maven-release-plugin] prepare for next development iteration	3 months ago

Инструменты разработки

Работа с браузером

Подготовка данных

Конфигурация тестов

Инъекция зависимостей

Подготовка данных

Пользователь

Счет

Объявление

Test

Карта

Договор

Профиль



Обычный HTTP запрос

```
curl -X POST http://<host>/service/users \  
-H "Content-Type: application/json" \  
-d "email=***" \  
-d "password=***"
```

http://square.github.io/retrofit

Retrofit

Download v2.3.0



A type-safe HTTP client for Android and Java

Introduction

Retrofit turns your HTTP API into a Java interface.

```
public interface GitHubService {  
    @GET("users/{user}/repos")  
    Call<List<Repo>> listRepos(@Path("user") String user);  
}
```

The `Retrofit` class generates an implementation of the `GitHubService` interface.

```
Retrofit retrofit = new Retrofit.Builder()  
    .baseUrl("https://api.github.com/")  
    .build();  
  
GitHubService service = retrofit.create(GitHubService.class);
```

Each `Call` from the created `GitHubService` can make a synchronous or asynchronous HTTP request to the remote webserver.

[Introduction](#)

[API Declaration](#)

[Retrofit Configuration](#)

[Download](#)

[Contributing](#)

[License](#)

[Javadoc](#)

[StackOverflow](#)

@Path параметры

```
curl -X GET http://<host>/service/users/123
```

```
public interface UserService {  
  
    @GET("/service/users/{id}")  
    Response<User> getUser(@Path("id") String id);  
  
}
```

@Query параметры

```
curl -X GET http://<host>/service/users?email=person@mail.com
```

```
public interface UserService {  
  
    @GET("/service/users")  
    Response<User> getUser(@Query("email") String email);  
  
}
```

Тело запроса

```
curl -X PUT http://<host>/service/users  
-d '{"email': 'person@mail.com', 'password': '***'}"
```

```
public interface UserService {  
  
    @PUT("/service/users")  
    Response<User> getUser(@Body User user);  
  
}
```


Работа с заголовками

```
curl -X GET http://<host>/service/me  
-H "Content-Type: application/json"  
-H "User-Agent: Retrofit-Sample-App"
```

```
public interface UserService {  
  
    @Headers({  
        "Content-Type: application/json",  
        "User-Agent: Retrofit-Sample-App"  
    })  
    @GET("/service/me")  
    Response<User> getMe();  
  
}
```

Описание UserService

```
public interface UserService {  
    @FormUrlEncoded  
    @POST( "/service/users" )  
    Response<User> createUser( @Field( "email" ) String email,  
                               @Field( "password" ) String pass );  
  
    @GET( "/service/users/{id}" )  
    Response<User> getUser( @Path( "id" ) String id );  
  
    @DELETE( "/service/users/{id}" )  
    Response<Void> deleteUser( @Path( "id" ) String id );  
}
```

Создание UserService

```
public static UserService createUserService() {  
  
    Retrofit retrofit = new Retrofit().Builder()  
        .addConverterFactory(GsonConverterFactory.create());  
        .baseUrl("https://api.company.com/")  
        .build();  
  
    return retrofit.create(UserService.class)  
  
}
```

ИСПОЛЬЗОВАНИЕ В ТЕСТЕ

```
public class SimpleTest {
    private UserService userService = createUserService();
    private User user;

    @Before
    public void createUser() {
        user = userService.createUser(randomEmail(), randomPassword());
    }
    @Test
    public void testUserPage() {...}
    @After
    public void deleteUser() {
        userService.deleteUser(user.getId());
    }
}
```

И много чего еще :)

Retrofit

Download v2.3.0



A type-safe HTTP client for Android and Java

Introduction

Retrofit turns your HTTP API into a Java interface.

```
public interface GitHubService {  
    @GET("users/{user}/repos")  
    Call<List<Repo>> listRepos(@Path("user") String user);  
}
```

The `Retrofit` class generates an implementation of the `GitHubService` interface.

```
Retrofit retrofit = new Retrofit.Builder()  
    .baseUrl("https://api.github.com/")  
    .build();  
  
GitHubService service = retrofit.create(GitHubService.class);
```

Each `Call` from the created `GitHubService` can make a synchronous or asynchronous HTTP request to the remote webserver.

[Introduction](#)

[API Declaration](#)

[Retrofit Configuration](#)

[Download](#)

[Contributing](#)

[License](#)

[Javadoc](#)

[StackOverflow](#)

Инструменты разработки

Работа с браузером

Подготовка данных

Конфигурация тестов

Инъекция зависимостей

Запуск тестов

```
mvn clean test -Denv=local
```

```
mvn clean test -Denv=testing
```

```
mvn clean test -Denv=stable
```

```
gradle clean test -Penv=local
```

```
gradle clean test -Penv=testing
```

```
gradle clean test -Penv=stable
```

Файл `testing.properties`

```
# Web config
```

```
service.web.url=https://testing.service.ru
```

```
service.web.cookies=...
```

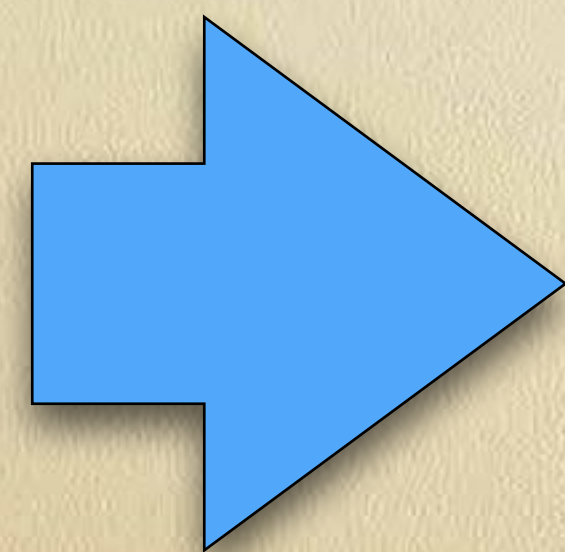
```
# Api config
```

```
service.api.url=https://testing.service.ru/api
```

```
service.api.username=admin
```

```
service.api.password=admin
```


Конфигурация тестов



local.properties

stable.properties

testing.properties

System.properties

<http://owner.aeonbits.org>

OWNER

Java™ properties reinvented.

OVERVIEW

DOCUMENTATION

NEWS

VIEW ON GITHUB

Fork me on GitHub

Get rid of the boilerplate code in properties based configuration.

Simple

No more code to load, convert and manage properties files, just use *your properties* in the simplest possible way.

[See how OWNER works →](#)

Powerful

Annotation based, automatic type conversion, variables expansion, load strategies, hot reload, event notifications, and *much* more...

[See OWNER features list →](#)

Flexible

Pick the features you need and forget what you don't need. Easy to adopt, scales on complexity, fully documented.

[Why should I use OWNER? →](#)

Описание конфигурации

```
public interface ProxyConfig extends Config {
```

```
    @Key("proxy.enabled")
```

```
    boolean isEnabled();
```

```
    @Key("proxy.host")
```

```
    String getHost();
```

```
    @Key("proxy.port")
```

```
    int getPort();
```

```
}
```

```
proxy.enabled=true
```

```
proxy.host=https://proxy.compnay.ru
```

```
proxy.port=4242
```

Примитивные типы

```
public interface ProxyConfig extends Config {  
  
    @Key("proxy.enabled")  
    boolean isEnabled();  
  
    @Key("proxy.host")  
    String getHost();  
  
    @Key("proxy.port")  
    int getPort();  
  
}
```

Расширенные ТИПЫ

```
public interface ProxyConfig extends Config {  
  
    @Key("service.url")  
    URL getUrl();  
  
    @Key("service.workspace")  
    File getWorkspace();  
  
    @Key("service.timeUnit")  
    TimeUnit getTimeUnit();  
  
}
```

СПИСКИ И МАССИВЫ

```
public interface ProxyConfig extends Config {  
  
    @Separator( "\n" )  
    @Key( "service.lines" )  
    List<File> getLines();  
  
    @Separator( ";" )  
    @Key( "service.words" )  
    String[] getWords();  
  
}
```

Дефолтные значения

```
public interface ProxyConfig extends Config {  
  
    @Key("proxy.host")  
    @DefaultValue("proxy.company.com")  
    String getHost();  
  
    @Key("proxy.port")  
    @DefaultValue("4242")  
    int getPort();  
  
}
```

Несколько источников

```
@Config.Sources({ "file:~/testing.properties",  
                  "file:/etc/testing.properties",  
                  "classpath:testing.properties" })  
public interface ProxyConfig extends Config {  
  
    //...  
  
}
```


Переопределение значений

```
public interface MyConfig extends Config {
```

```
    @Key("proxy.enabled")  
    boolean isEnabled();
```

```
    @Key("proxy.host")  
    String getHost();
```

```
    @Key("proxy.port")  
    int getPort();
```

```
}
```

```
mvn clean test \  
-Dproxy.enabled=true \  
-Dproxy.host=some.host.com \  
-Dproxy.port=4343 \  
-Dproxy.enabled=true \  
-Dproxy.host=some.host.com \  
-Dproxy.port=4343
```

Конфигурация WebConfig

```
@Config.Sources("classpath:{env}.properties")
public interface WebConfig extends Config {

    @Key("service.web.url")
    URL getUrl();

    @Key("service.web.cookies")
    List<String> getCookies();

}
```

```
mvn clean test -Denv=testing
```

Конфигурация ApiConfig

```
@Config.Sources("classpath:{env}.properties")
public interface ApiConfig extends Config {

    @Key("service.api.url")
    URL getUrl();

    @Key("service.api.username")
    String getUsername();

    @Key("service.api.password")
    String getPassword();

}
```


Использование в тестах

```
public class SimpleTest {  
  
    private WebConfig webConfig = createWebConfig();  
  
    private WebDriver webDriver = createWebDriver(webConfig);  
  
    private User user;  
  
    @Test  
    void testSomething() { ... }  
  
}
```

<http://owner.aeonbits.org>

OWNER

Java™ properties reinvented.

OVERVIEW

DOCUMENTATION

NEWS

VIEW ON GITHUB

Fork me on GitHub

Get rid of the boilerplate code in properties based configuration.

Simple

No more code to load, convert and manage properties files, just use *your properties* in the simplest possible way.

[See how OWNER works →](#)

Powerful

Annotation based, automatic type conversion, variables expansion, load strategies, hot reload, event notifications, and *much* more...

[See OWNER features list →](#)

Flexible

Pick the features you need and forget what you don't need. Easy to adopt, scales on complexity, fully documented.

[Why should I use OWNER? →](#)

Инструменты разработки

A person wearing a blue shirt and a tan tool belt is using a red power drill on a wall. The tool belt is filled with various tools, including screwdrivers, pliers, and a hammer. The background is a light-colored wall and a wooden surface.

Работа с браузером

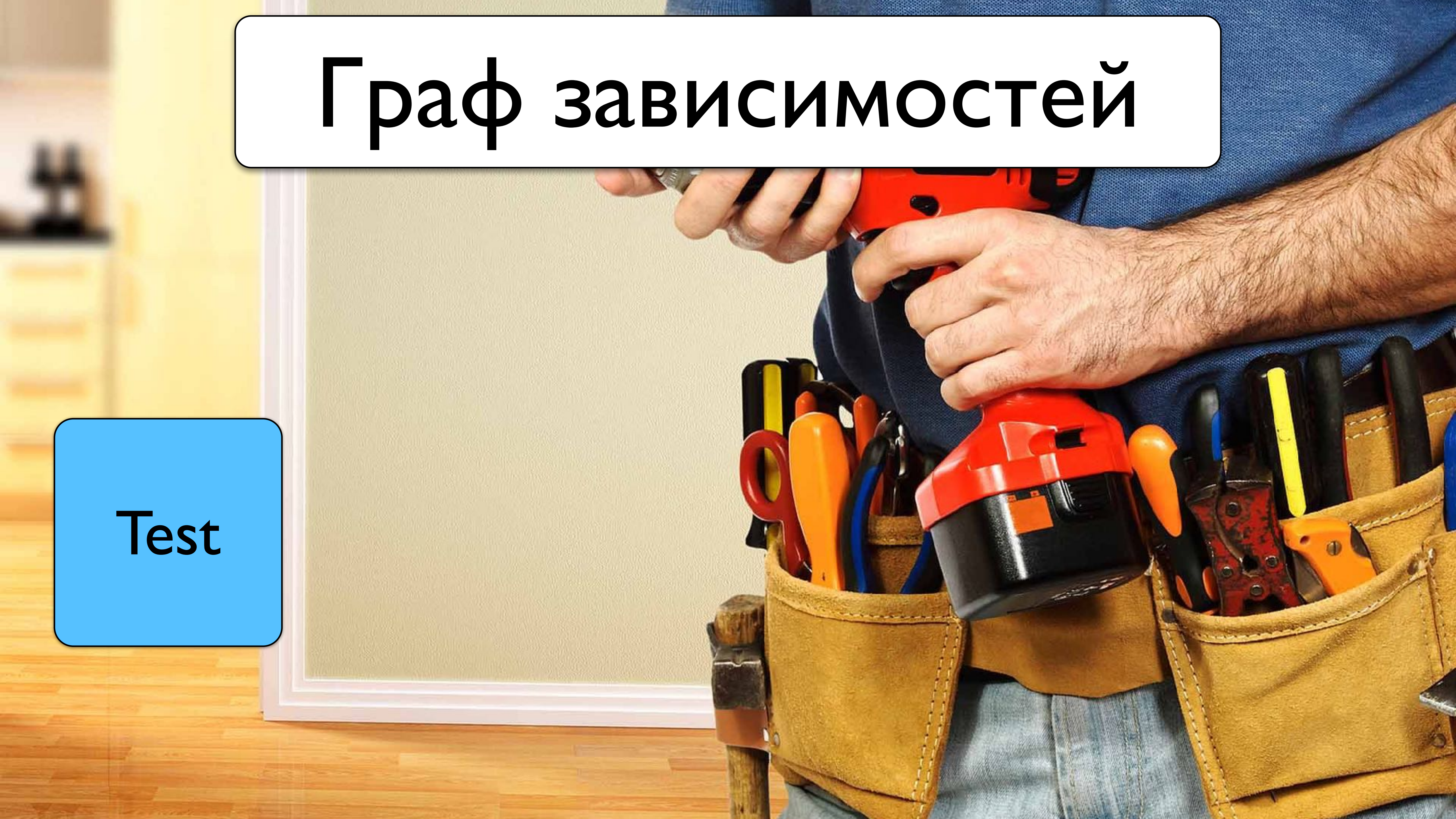
Подготовка данных

Конфигурация тестов

Инъекция зависящих

Граф зависимостей

Test



Граф зависимостей

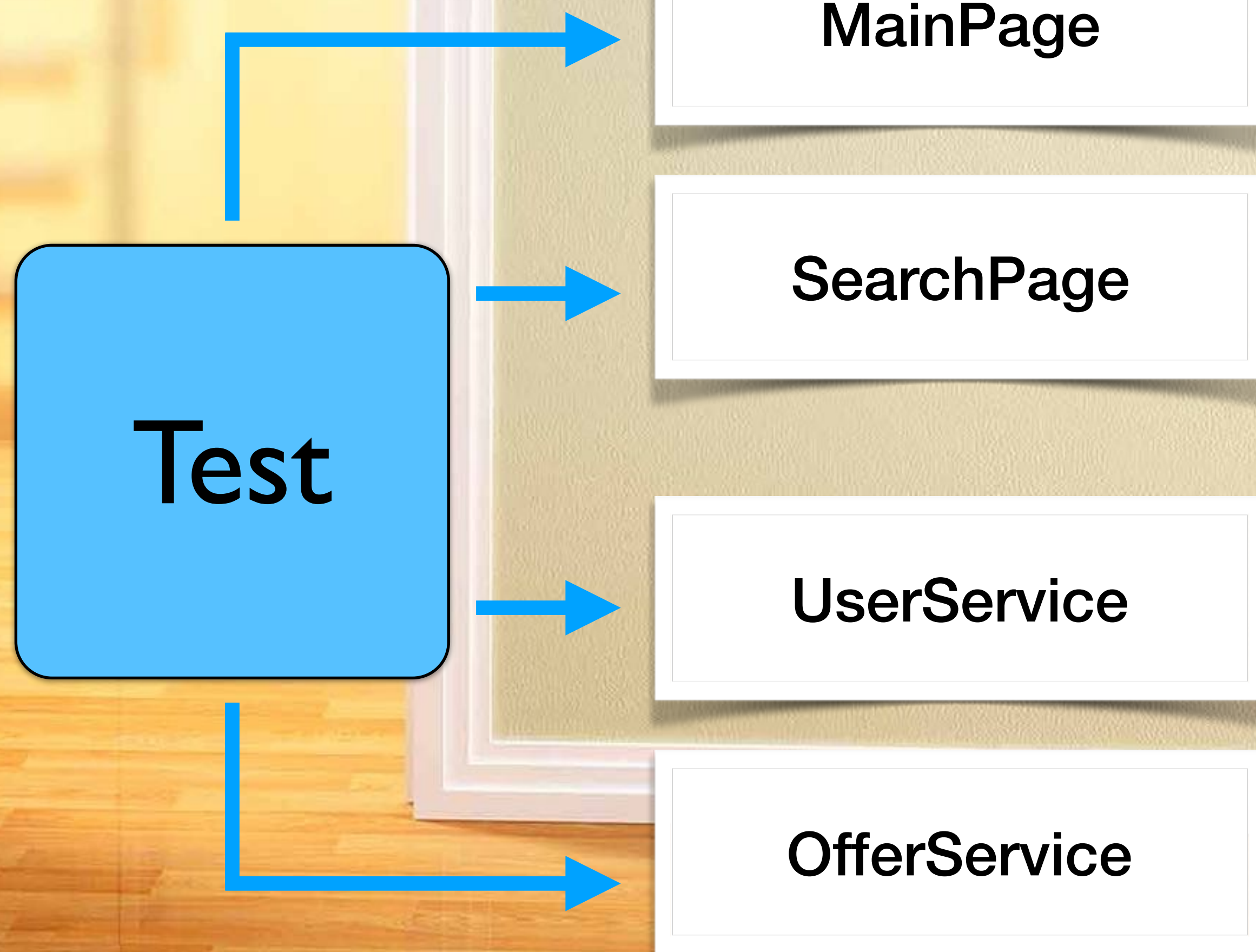
MainPage

SearchPage

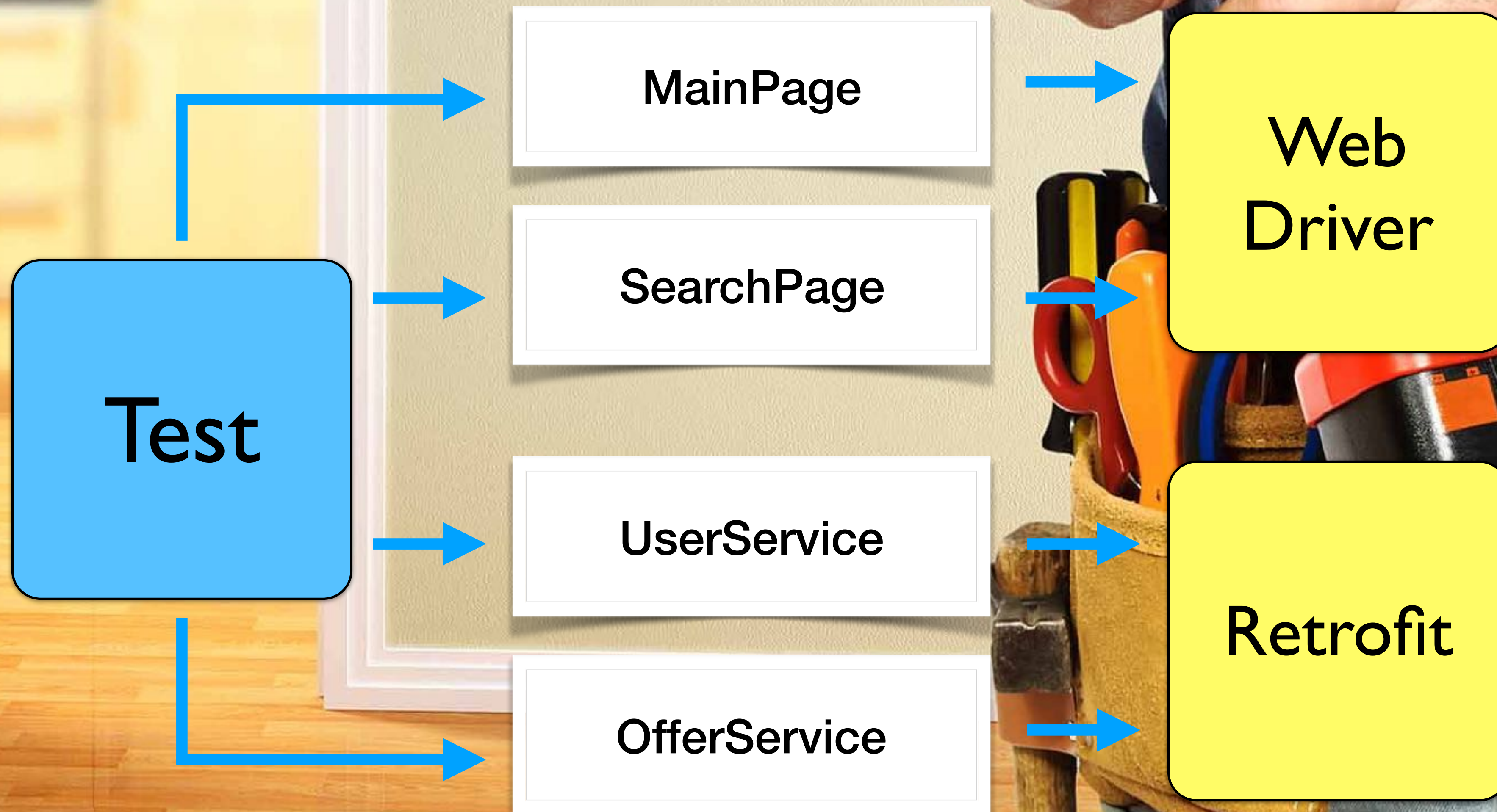
UserService

OfferService

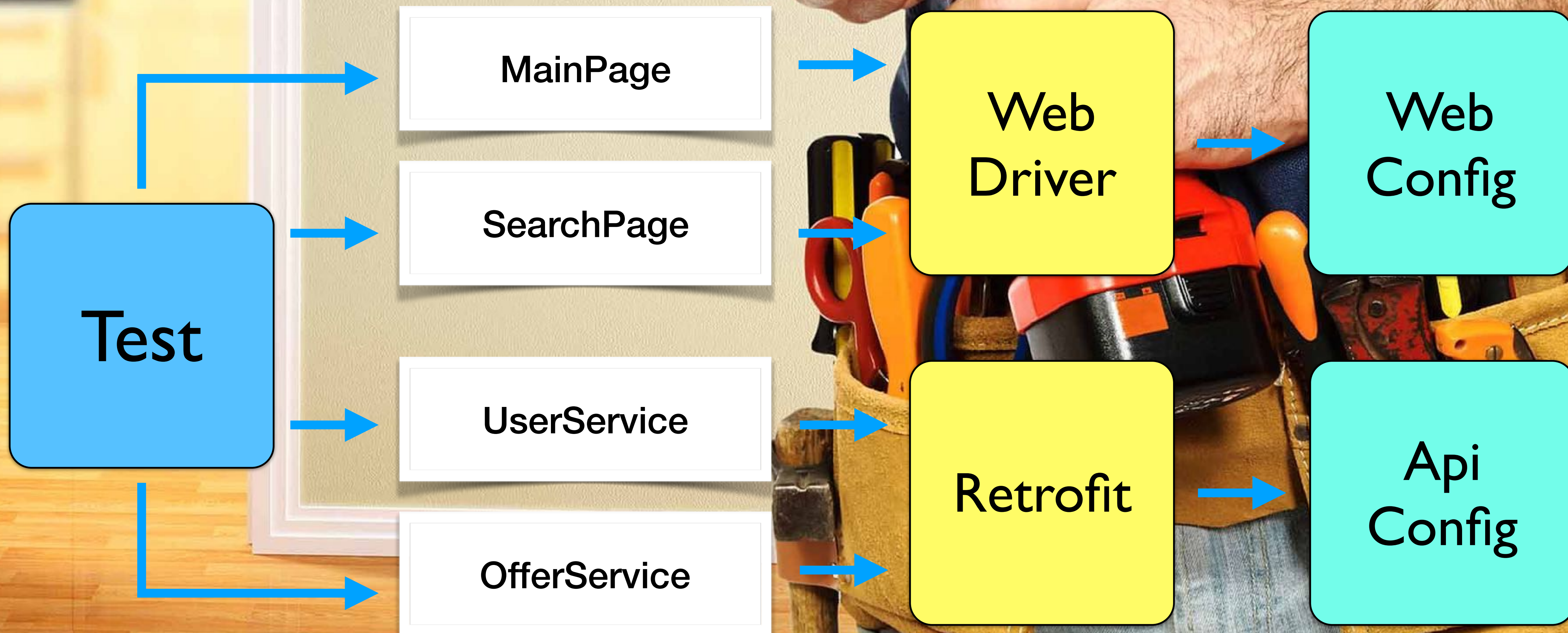
Test



Граф зависимостей



Граф зависимостей



Как ВЫГЛЯДИТ ТЕСТ?

```
public class SimpleTest {  
  
    private User user;  
  
    private WebConfig webConfig = createWebConfig();  
    private WebDriver webDriver = createWebDriver(webConfig);  
    private MainPage mainPage = createMainPage(webDriver);  
  
    private ApiConfig apiConfig = createApiConfig();  
    private UserService userService =  
        createUserService(apiConfig);  
  
    void testSomething() { ... }  
}
```

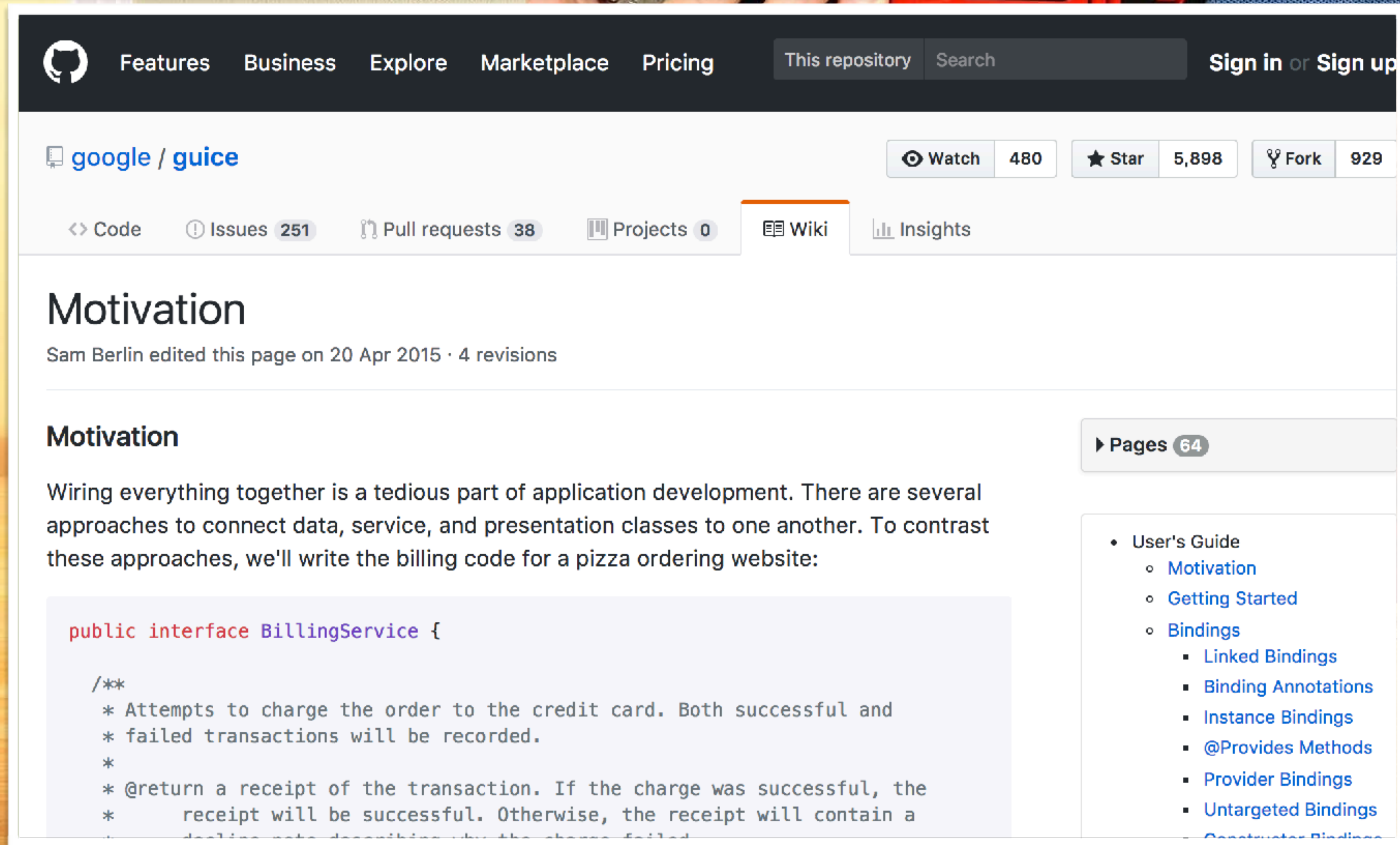
Базовый класс - плохо

```
public class BaseClass {  
  
    private WebConfig webConfig = createWebConfig();  
    private WebDriver webDriver = createWebDriver(webConfig);  
    private MainPage mainPage = createMainPage(webDriver);  
  
    private ApiConfig apiConfig = createApiConfig();  
    private UserService userService =  
        createUserService(apiConfig);  
  
}
```

Статические поля - плохо

```
public class Configuration {  
  
    private static WebConfig webConfig;  
  
    private static WebDriver webDriver;  
  
    private static MainPage mainPage;  
  
    private static ApiConfig apiConfig;  
  
    private static UserService userService;  
  
    static {...}  
  
}
```

http://github.com/google/guice



The screenshot shows the GitHub repository page for `google/guice`. The repository has 480 watchers, 5,898 stars, and 929 forks. The Wiki page for `Motivation` is displayed, showing the title, author (Sam Berlin), and a paragraph of text. A code block for `BillingService` is also visible. On the right side, there is a navigation menu for the Wiki pages, including `User's Guide`, `Motivation`, `Getting Started`, and `Bindings`.

Features Business Explore Marketplace Pricing This repository Search Sign in or Sign up

google / guice Watch 480 Star 5,898 Fork 929

Code Issues 251 Pull requests 38 Projects 0 Wiki Insights

Motivation

Sam Berlin edited this page on 20 Apr 2015 · 4 revisions

Motivation

Wiring everything together is a tedious part of application development. There are several approaches to connect data, service, and presentation classes to one another. To contrast these approaches, we'll write the billing code for a pizza ordering website:

```
public interface BillingService {  
  
    /**  
     * Attempts to charge the order to the credit card. Both successful and  
     * failed transactions will be recorded.  
     *  
     * @return a receipt of the transaction. If the charge was successful, the  
     *         receipt will be successful. Otherwise, the receipt will contain a  
     *         decline note describing why the charge failed.  
     */  
}
```

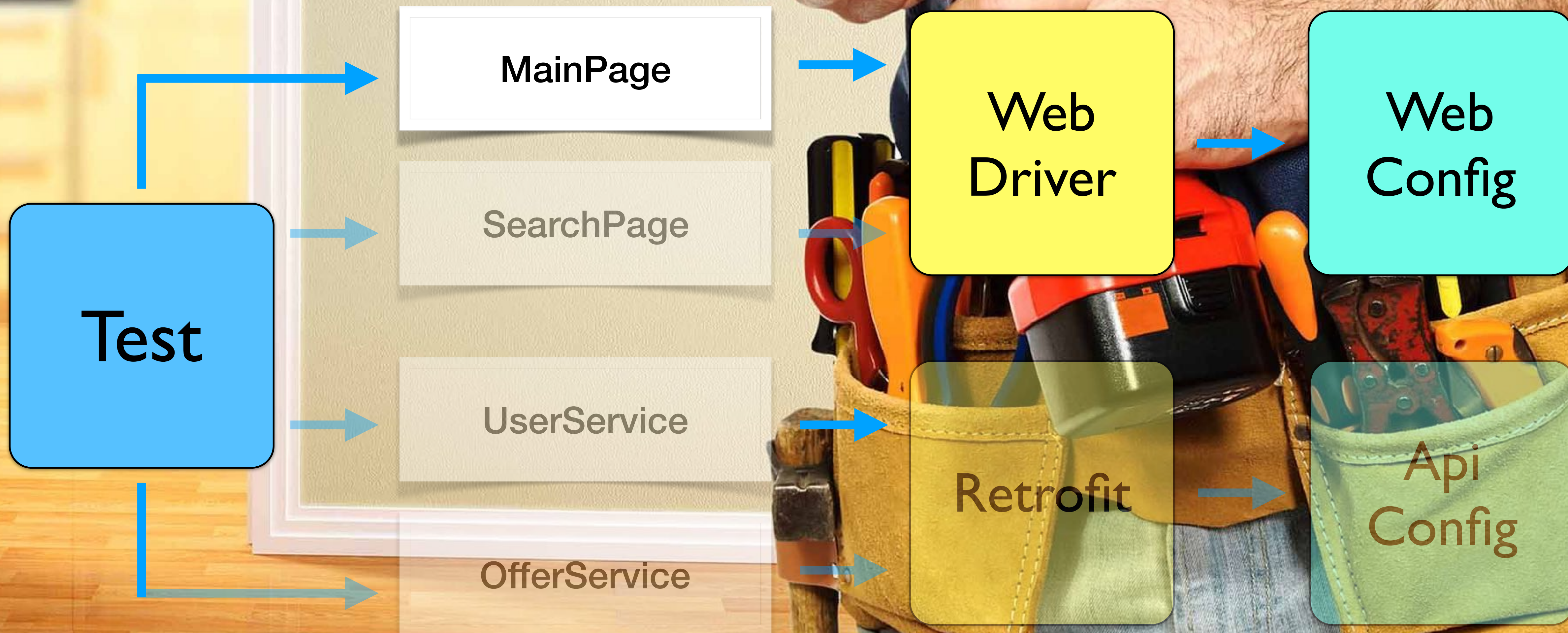
Pages 64

- User's Guide
 - Motivation
 - Getting Started
 - Bindings
 - Linked Bindings
 - Binding Annotations
 - Instance Bindings
 - @Provides Methods
 - Provider Bindings
 - Untargeted Bindings
 - Constructor Bindings

Что нам реально нужно?

```
public class SimpleTest {  
  
    private User user;  
  
    private WebConfig webConfig = createWebConfig();  
    private WebDriver webDriver = createWebDriver(webConfig);  
    private MainPage mainPage = createMainPage(webDriver);  
  
    private ApiConfig apiConfig = createApiConfig();  
    private UserService userService =  
        createUserService(apiConfig);  
  
    void testSomething() { ... }  
  
}
```


Граф зависимостей



Поставщик WebConfig

```
class WebConfigProvider implements Provider<WebConfig> {  
  
    @Override  
    protected WebConfig get() {  
        return ConfigFactory.create(WebConfig.class,  
                                   System.properties);  
    }  
  
}
```

Поставщик WebDriver

```
class WebDriverProvider implements Provider<WebDriver> {  
  
    private WebConfig config;  
  
    @Inject  
    public WebDriverProvider(WebConfig config) {  
        this.config = config;  
    }  
  
    @Override  
    protected WebDriver get() {  
        return new WebDriver(config);  
    }  
}
```

Поставщик MainPage

```
class MainPageProvider implements Provider<MainPage> {  
  
    private WebDriver driver;  
  
    @Inject  
    public MainPageProvider(WebDriver driver) {  
        this.driver = driver;  
    }  
  
    @Override  
    protected MainPage get() {  
        return new WebPageFactory(driver).create(MainPage.class);  
    }  
}
```

Конфигурация модуля

```
public class TestModule extends AbstractModule {  
  
    @Override  
    protected void configure() {  
        bind(WebConfig.class).to(WebConfigProvider.class);  
        bind(WebDriver.class).to(WebDriverProvider.class);  
        bind(MainPage.class).to(MainPageProvider.class);  
  
        bind(ApiConfig.class).to(ApiConfigProvider.class);  
        bind(Retrofit.class).to(RetrofitProvider.class);  
        //...  
    }  
}
```

Как ВЫГЛЯДИТ ТЕСТ?

```
public class SimpleTest {  
  
    private User user;  
  
    @Inject  
    private MainPage mainPage;  
  
    @Inject  
    private UserService userService;  
  
    void testSomething() { ... }  
}
```


Как ВЫГЛЯДИТ ТЕСТ?

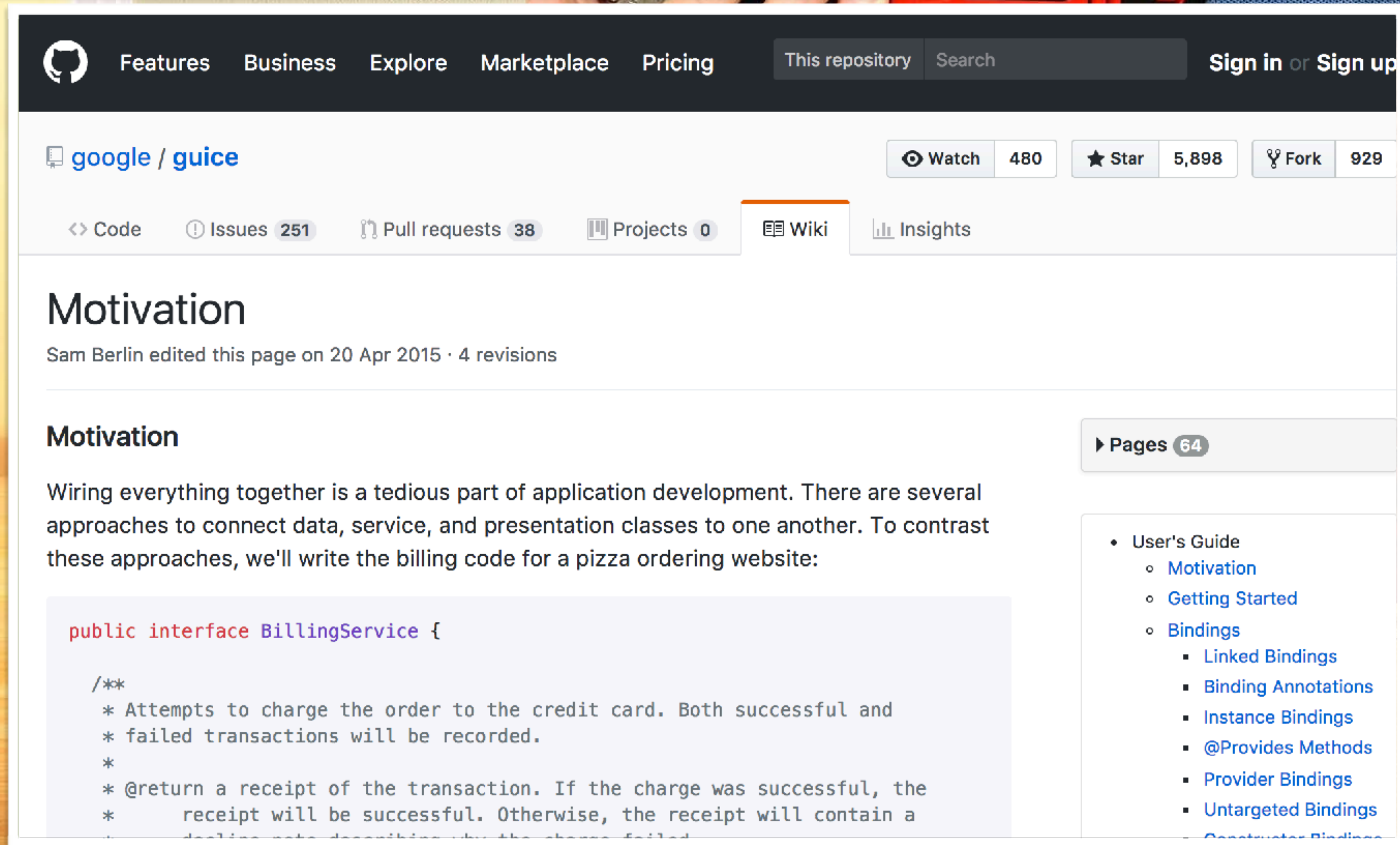
```
public class SimpleTest {  
  
    @Inject  
    private User user;  
  
    @Inject  
    private MainPage mainPage;  
  
    void testSomething() { ... }  
  
}
```


Интеграция с JUnit

<https://github.com/marcolamberto/guice-junit-runner>

<https://github.com/google/acai>

http://github.com/google/guice



The screenshot shows the GitHub repository page for `google/guice`. The repository has 480 watchers, 5,898 stars, and 929 forks. The Wiki page for `Motivation` is displayed, showing the title, author (Sam Berlin), and a paragraph of text. A code block for `BillingService` is also visible. On the right side, there is a navigation menu for the Wiki pages, including `User's Guide`, `Motivation`, `Getting Started`, and `Bindings`.

Features Business Explore Marketplace Pricing This repository Search Sign in or Sign up

google / guice Watch 480 Star 5,898 Fork 929

Code Issues 251 Pull requests 38 Projects 0 Wiki Insights

Motivation

Sam Berlin edited this page on 20 Apr 2015 · 4 revisions

Motivation

Wiring everything together is a tedious part of application development. There are several approaches to connect data, service, and presentation classes to one another. To contrast these approaches, we'll write the billing code for a pizza ordering website:

```
public interface BillingService {  
  
    /**  
     * Attempts to charge the order to the credit card. Both successful and  
     * failed transactions will be recorded.  
     *  
     * @return a receipt of the transaction. If the charge was successful, the  
     *         receipt will be successful. Otherwise, the receipt will contain a  
     *         decline note describing why the charge failed.  
     */  
}
```

Pages 64

- User's Guide
 - Motivation
 - Getting Started
 - Bindings
 - Linked Bindings
 - Binding Annotations
 - Instance Bindings
 - @Provides Methods
 - Provider Bindings
 - Untargeted Bindings
 - Constructor Bindings

Инструменты разработки

Работа с браузером

HtmlElements2

Подготовка данных

Retrofit

Конфигурация тестов

Owner

Инъекция зависимостей

Guice

A wooden table with a map and a notebook. The map is on the left, showing a city layout with red and yellow markers. The notebook is on the right, with some text visible. The background is a plain wall.

Простота

Доверие

КОНТРОЛЬ

Доверие команды к тестам



Короткие тесты

Сервер "заглушек"

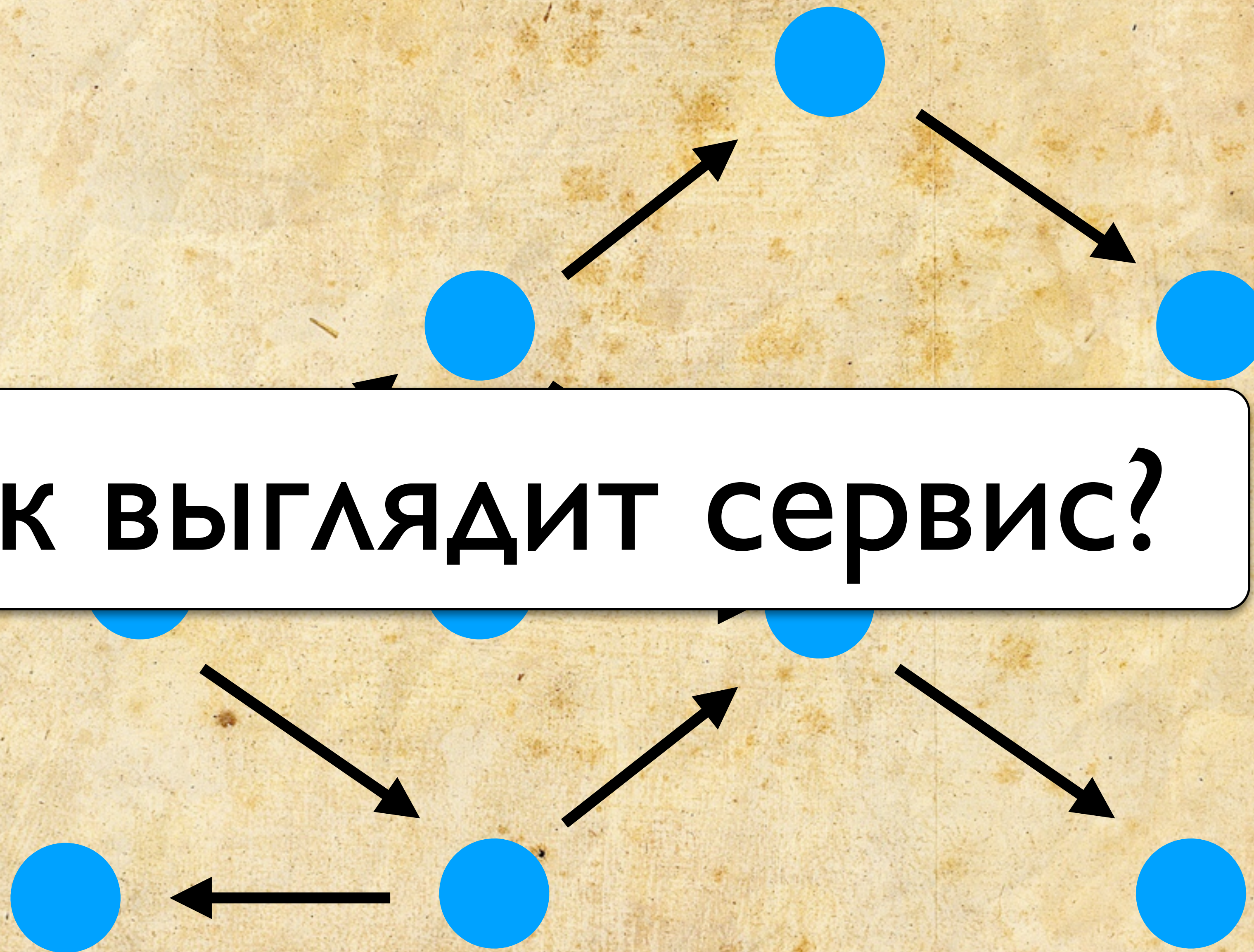
Доверие команды к тестам

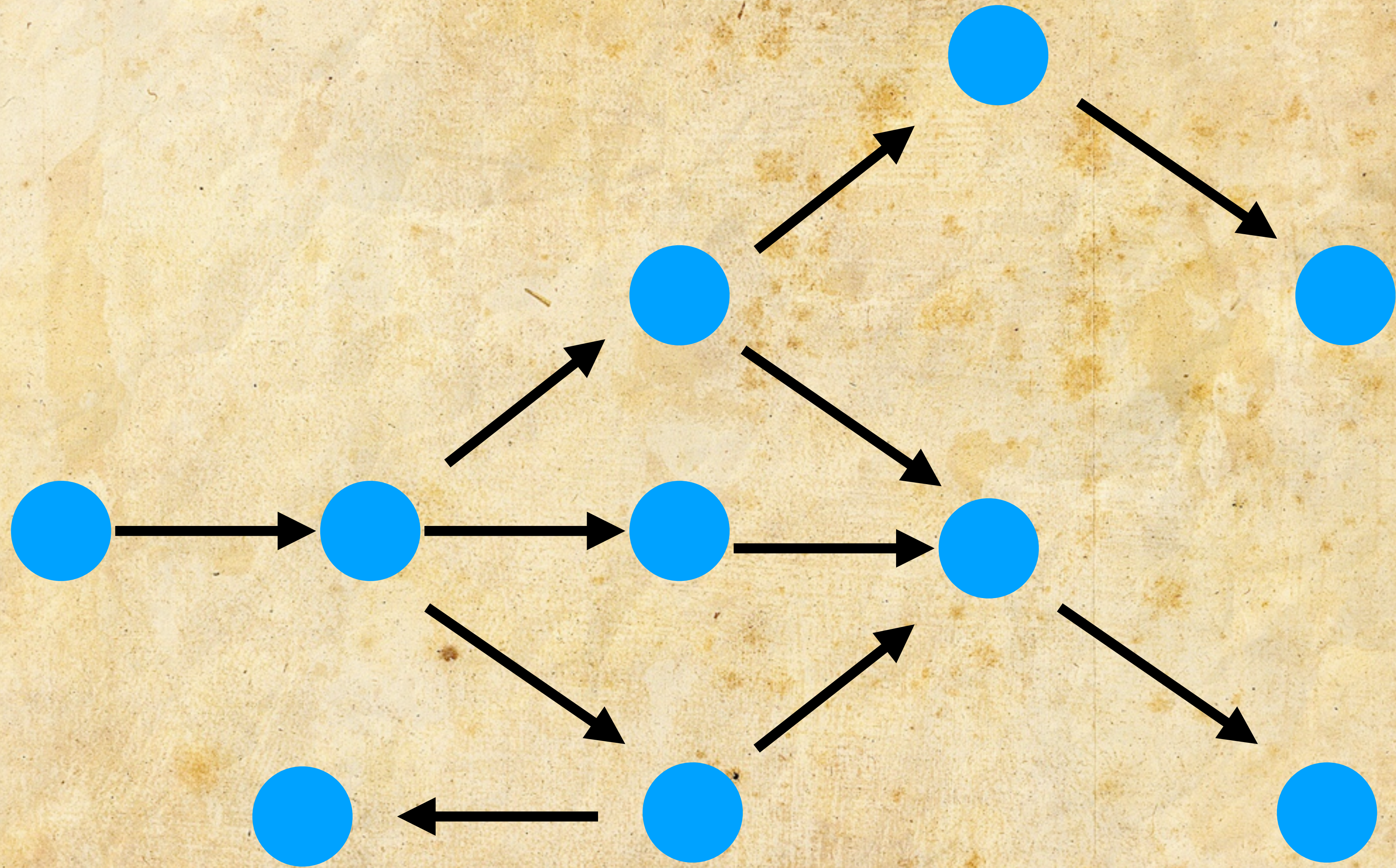


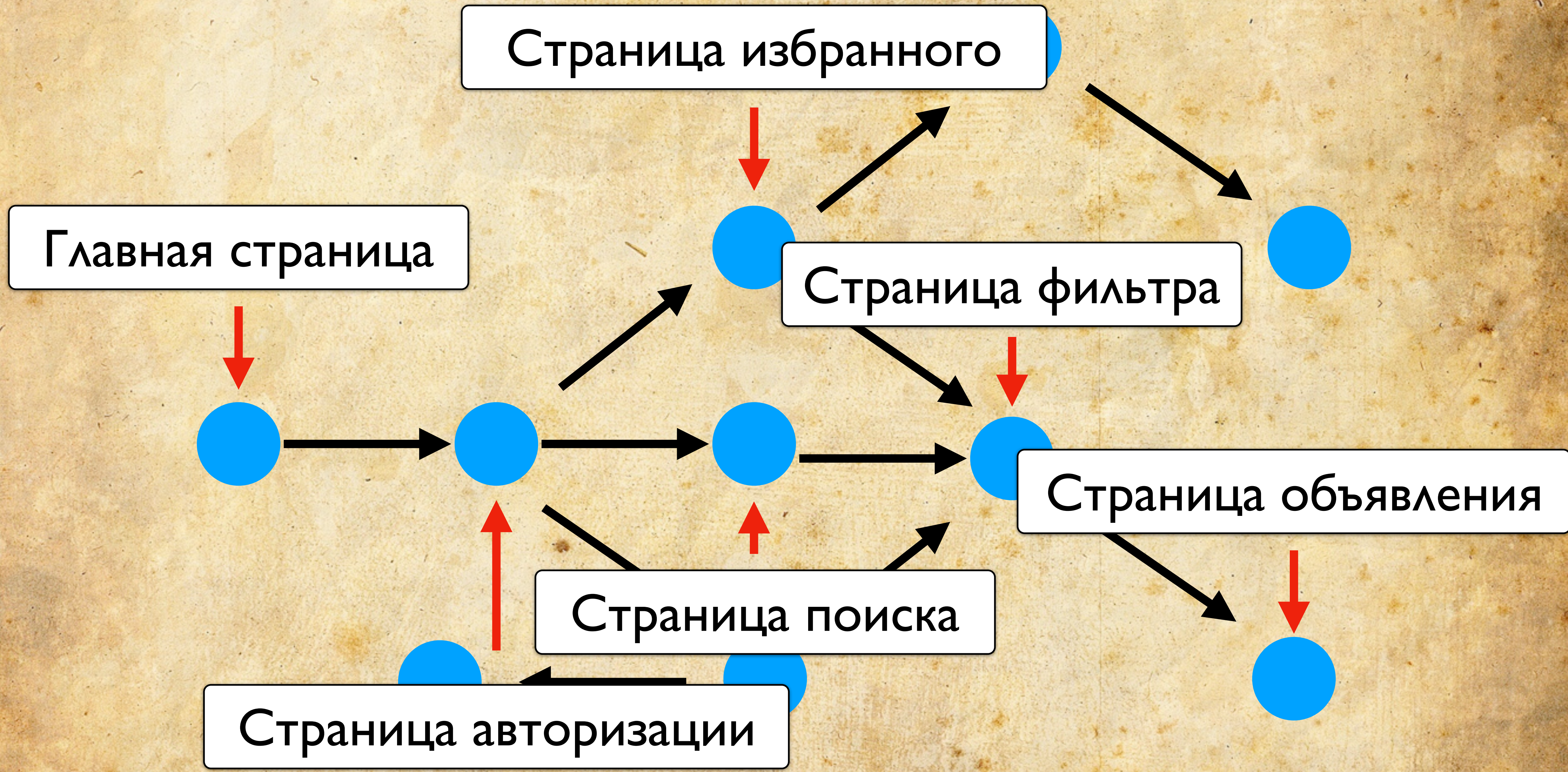
Короткие тесты

Сервер "заглушек"

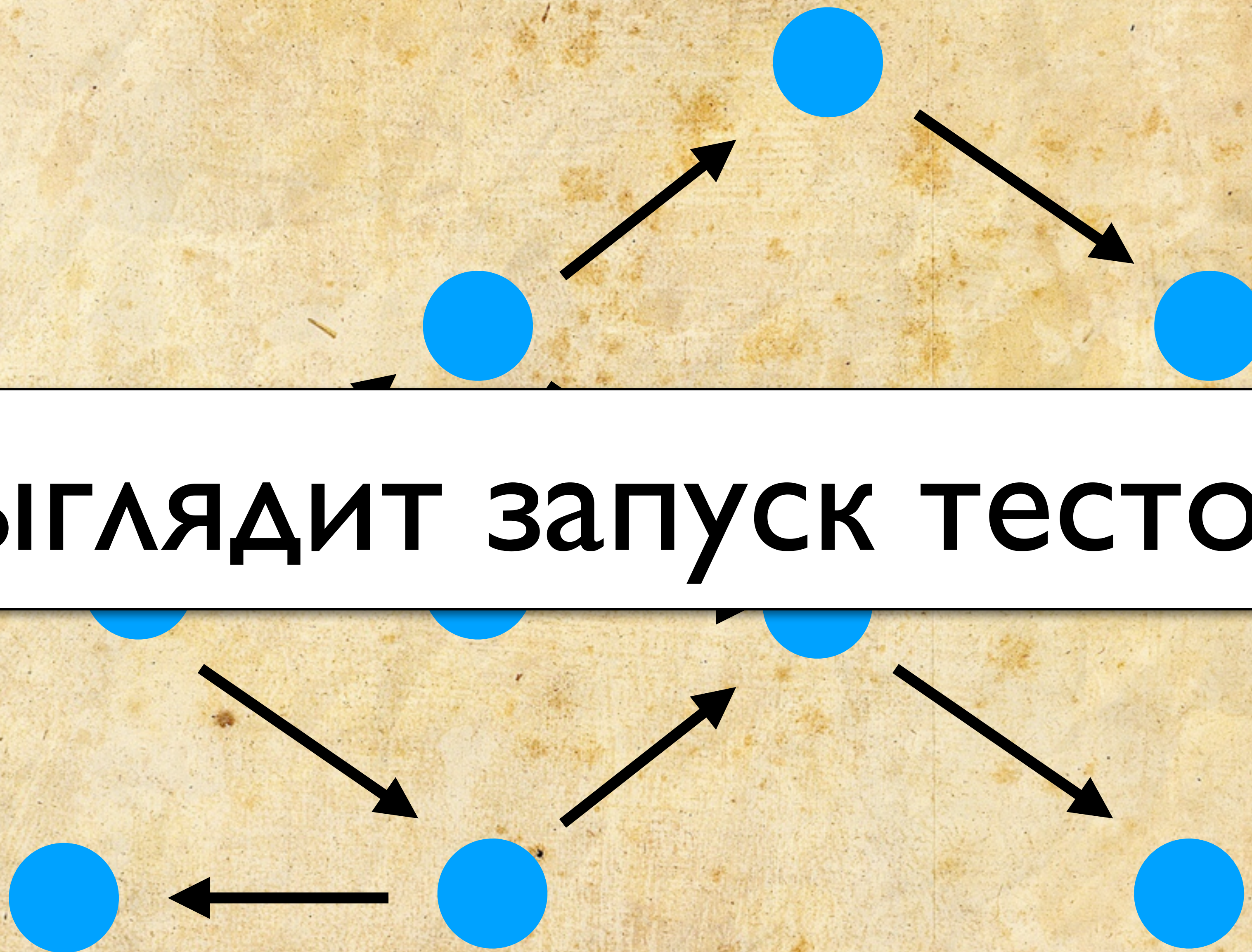
Как выглядит сервис?



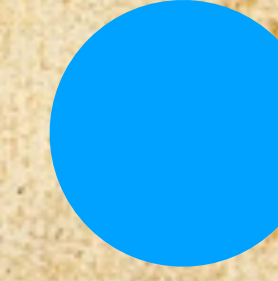
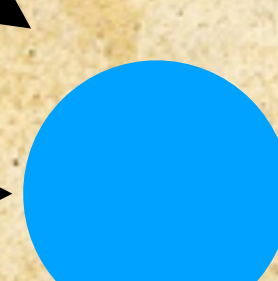


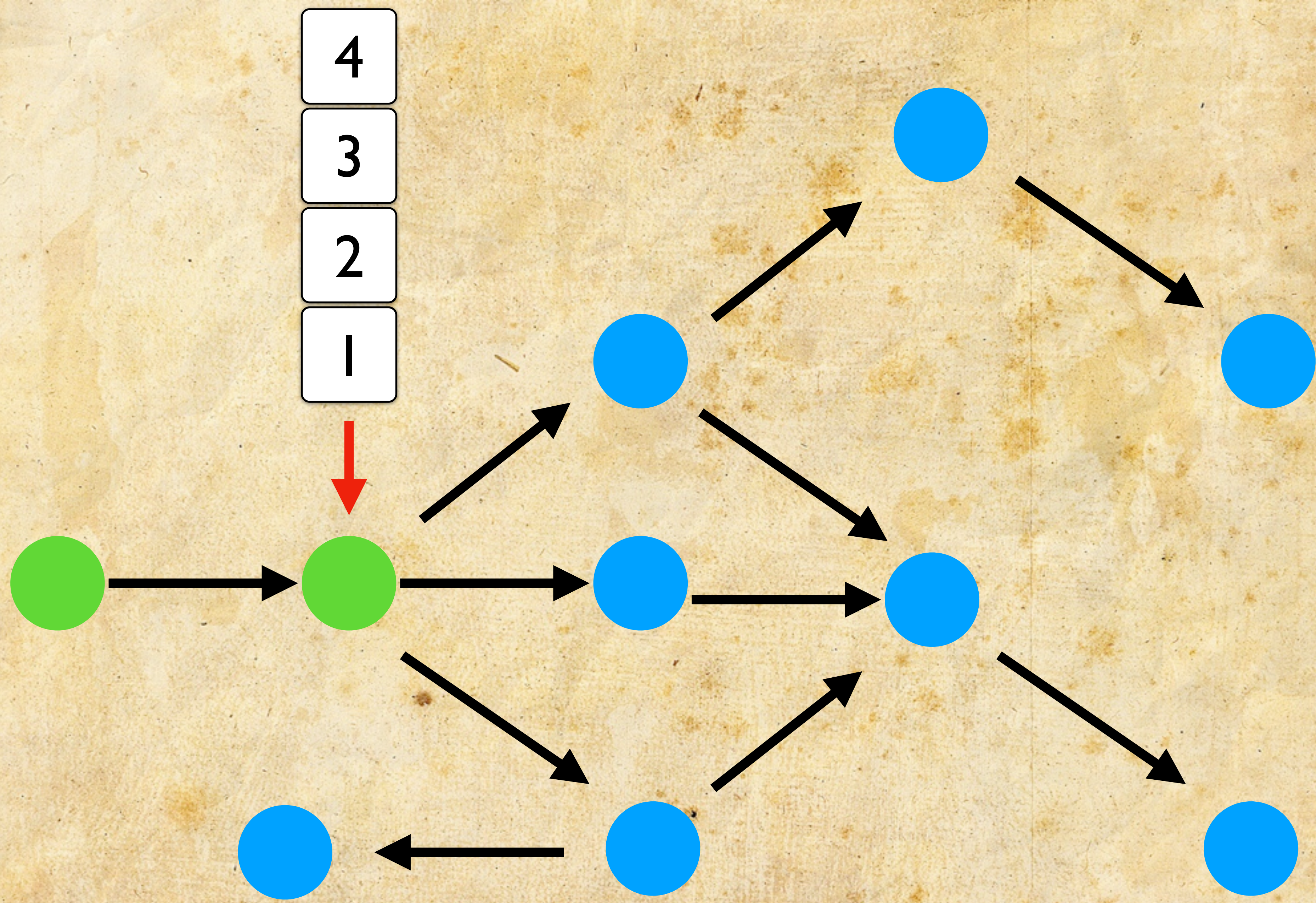


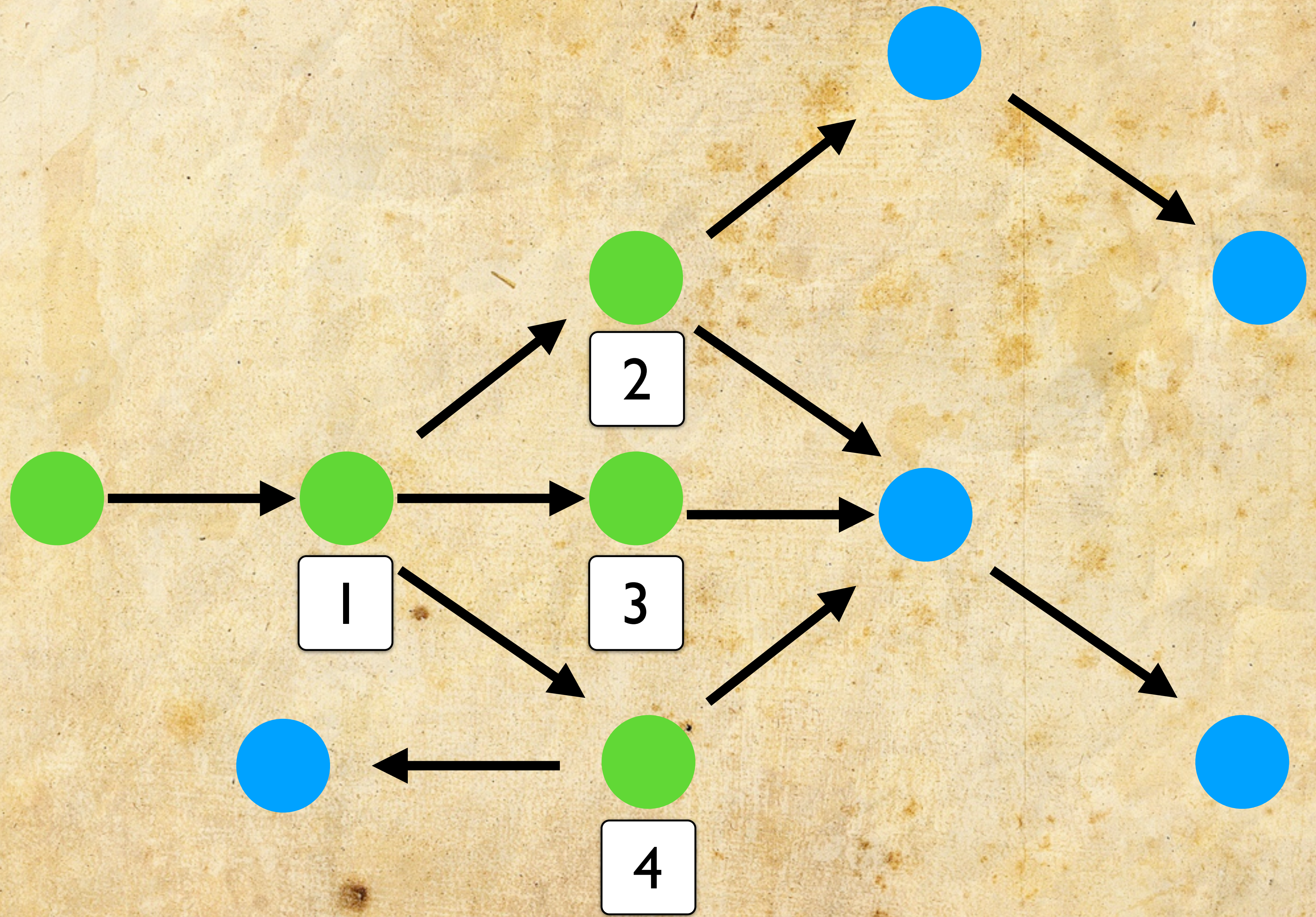
Как выглядит запуск тестов

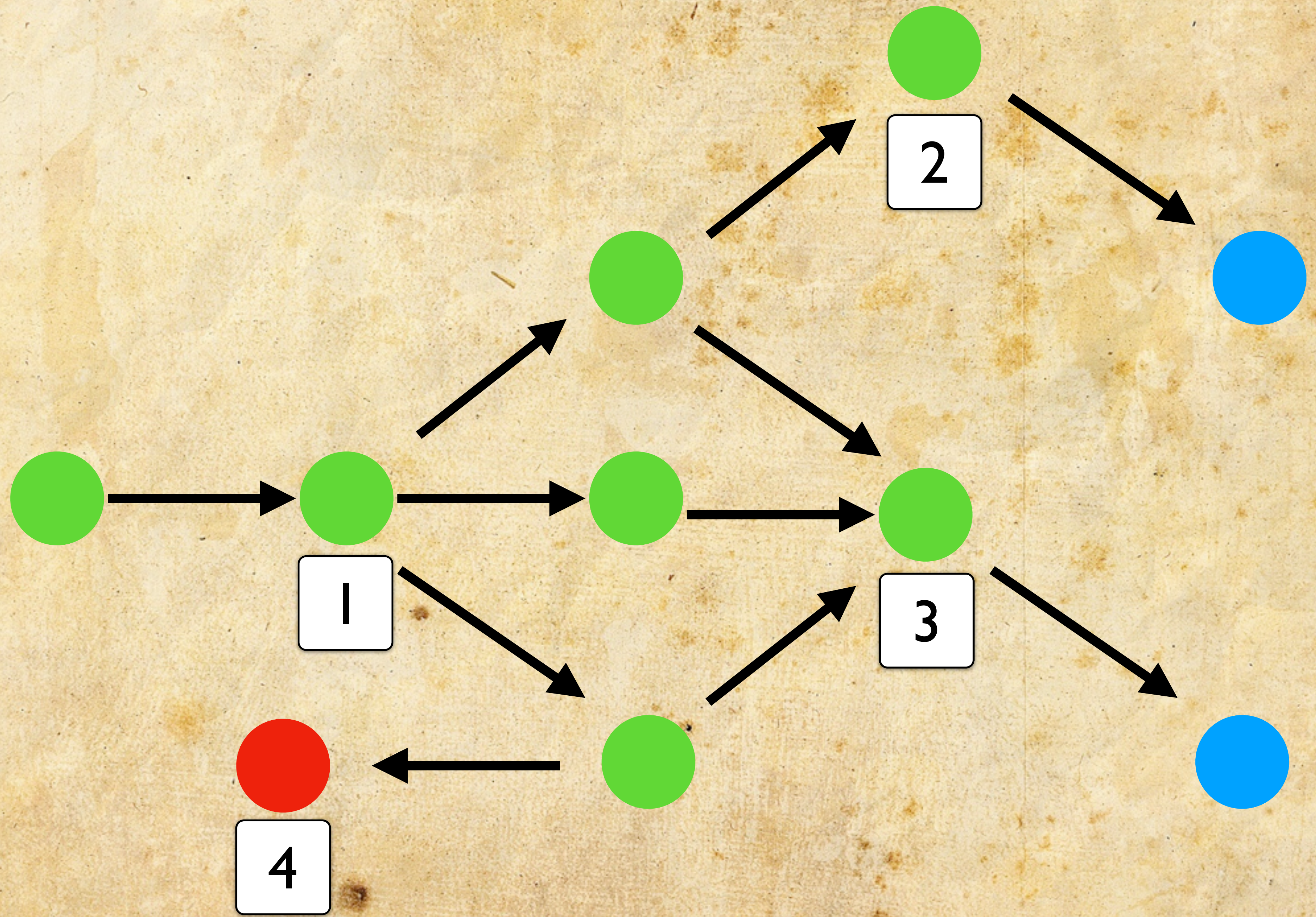


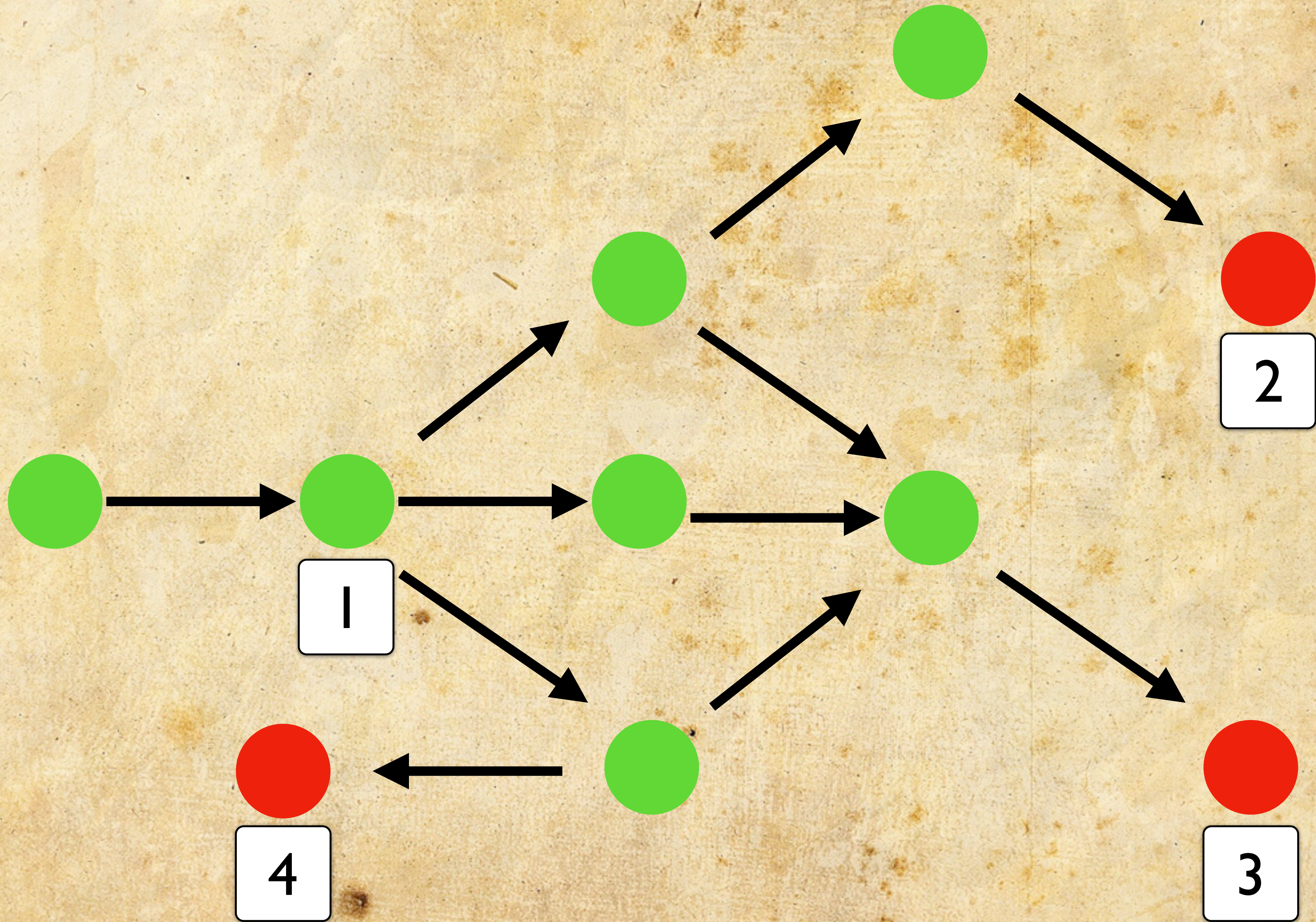
- 4
- 3
- 2
- 1









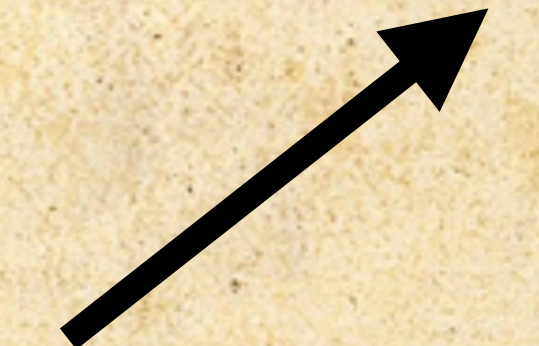
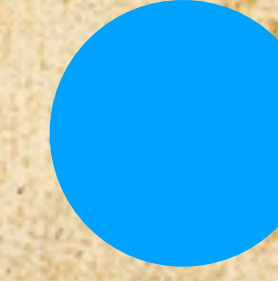
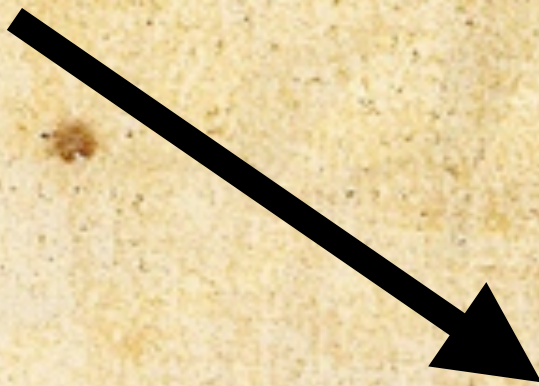
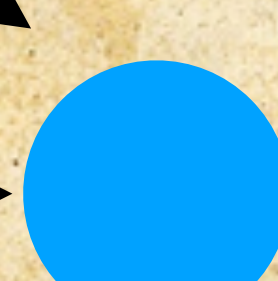


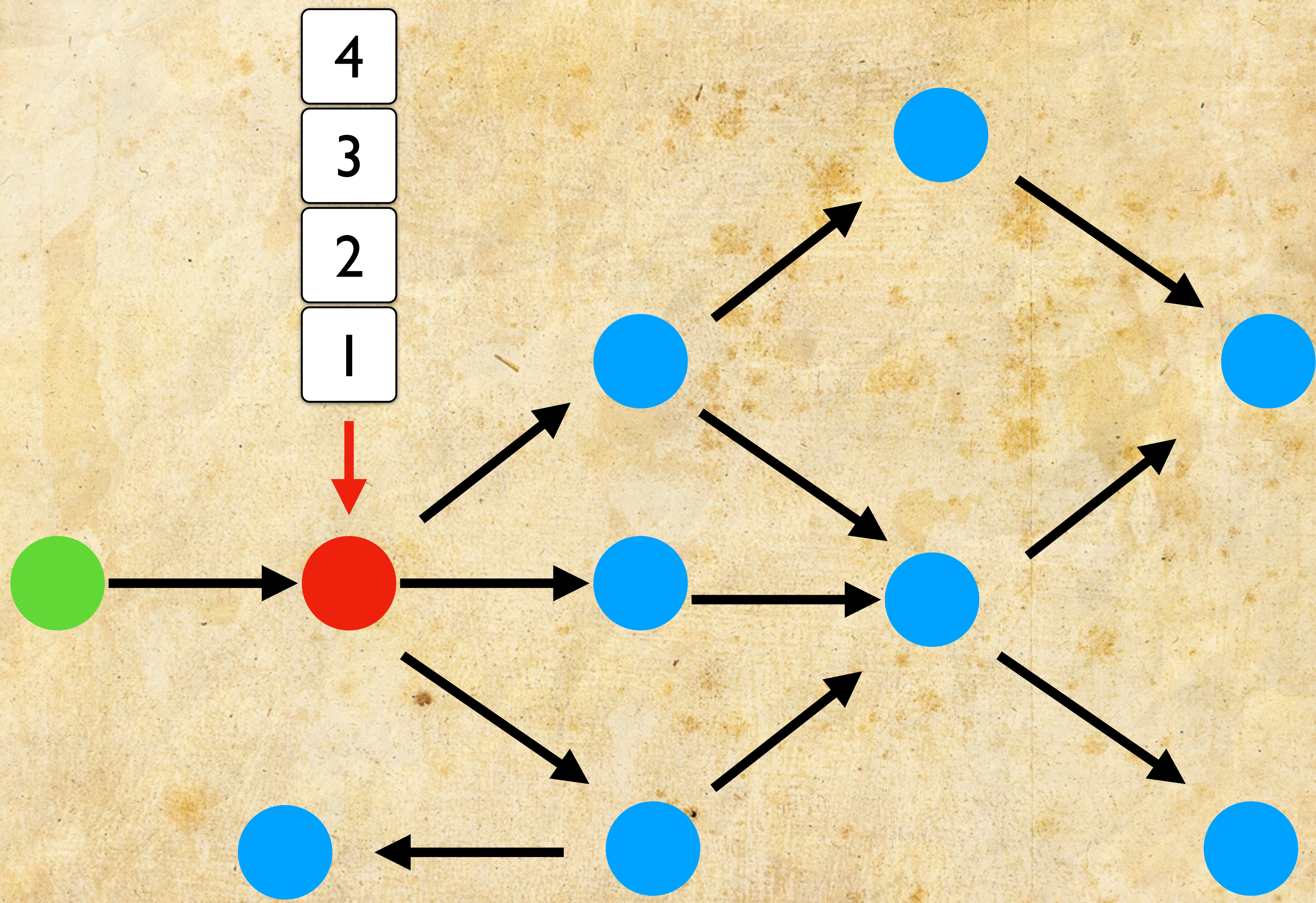


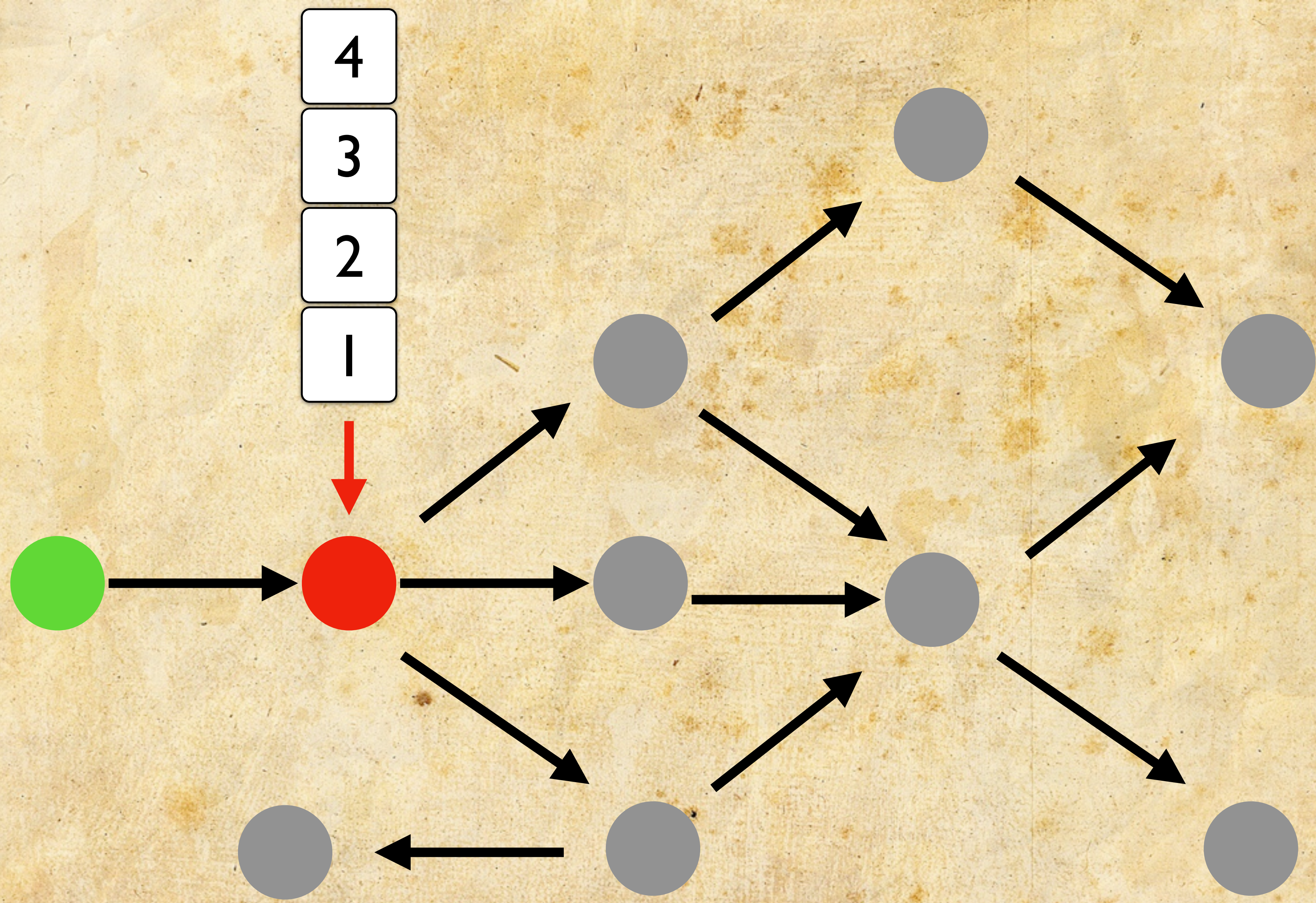
Как на самом деле?

The image features a central white rounded rectangle with a black border containing the Russian text "Как на самом деле?". Surrounding this rectangle are several blue circular nodes. Some nodes are partially obscured by the rectangle. Black arrows connect the nodes in a network, with some pointing towards the central text box and others pointing away from it. The background is a textured, aged, light brown paper.

- 4
- 3
- 2
- 1





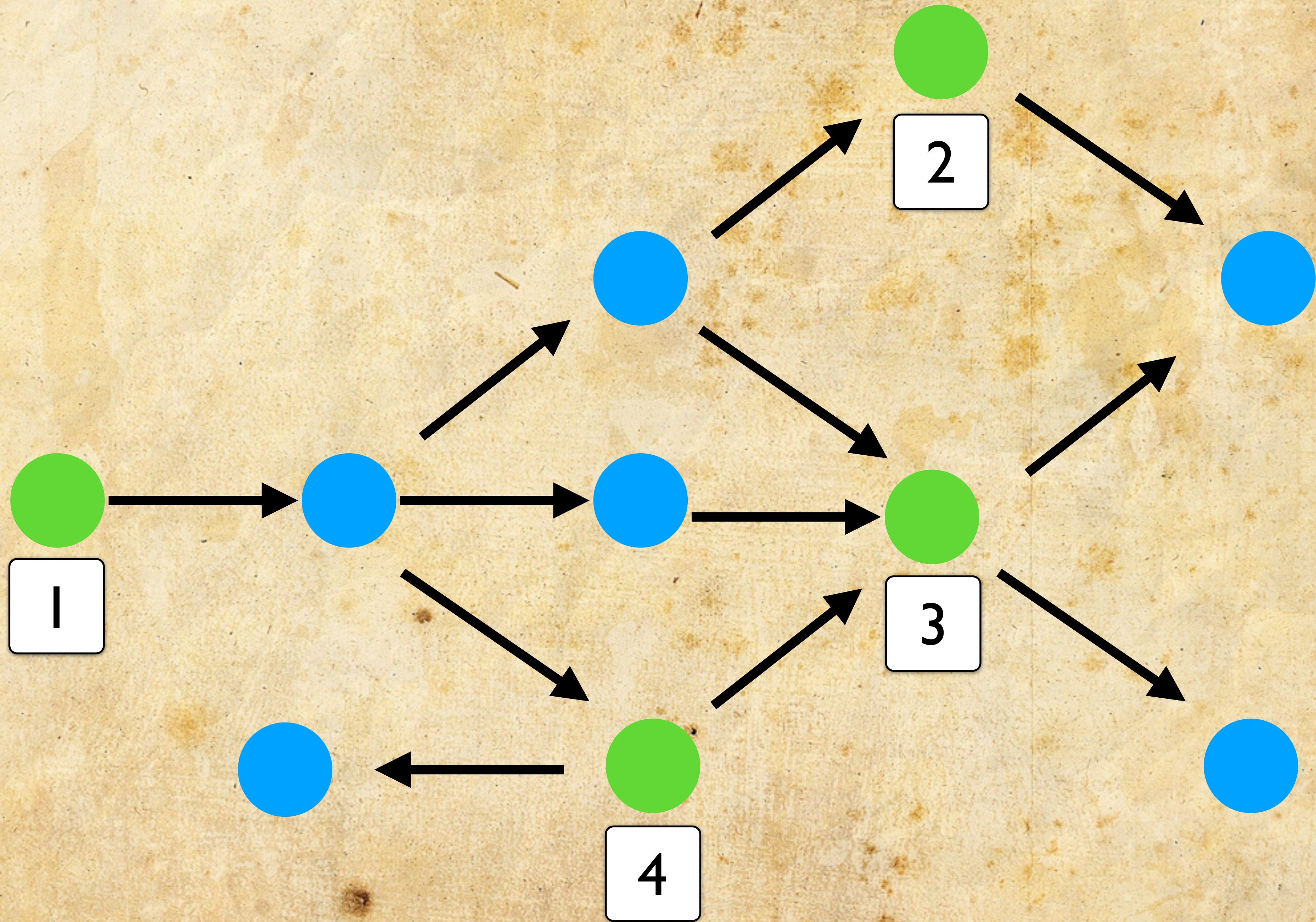


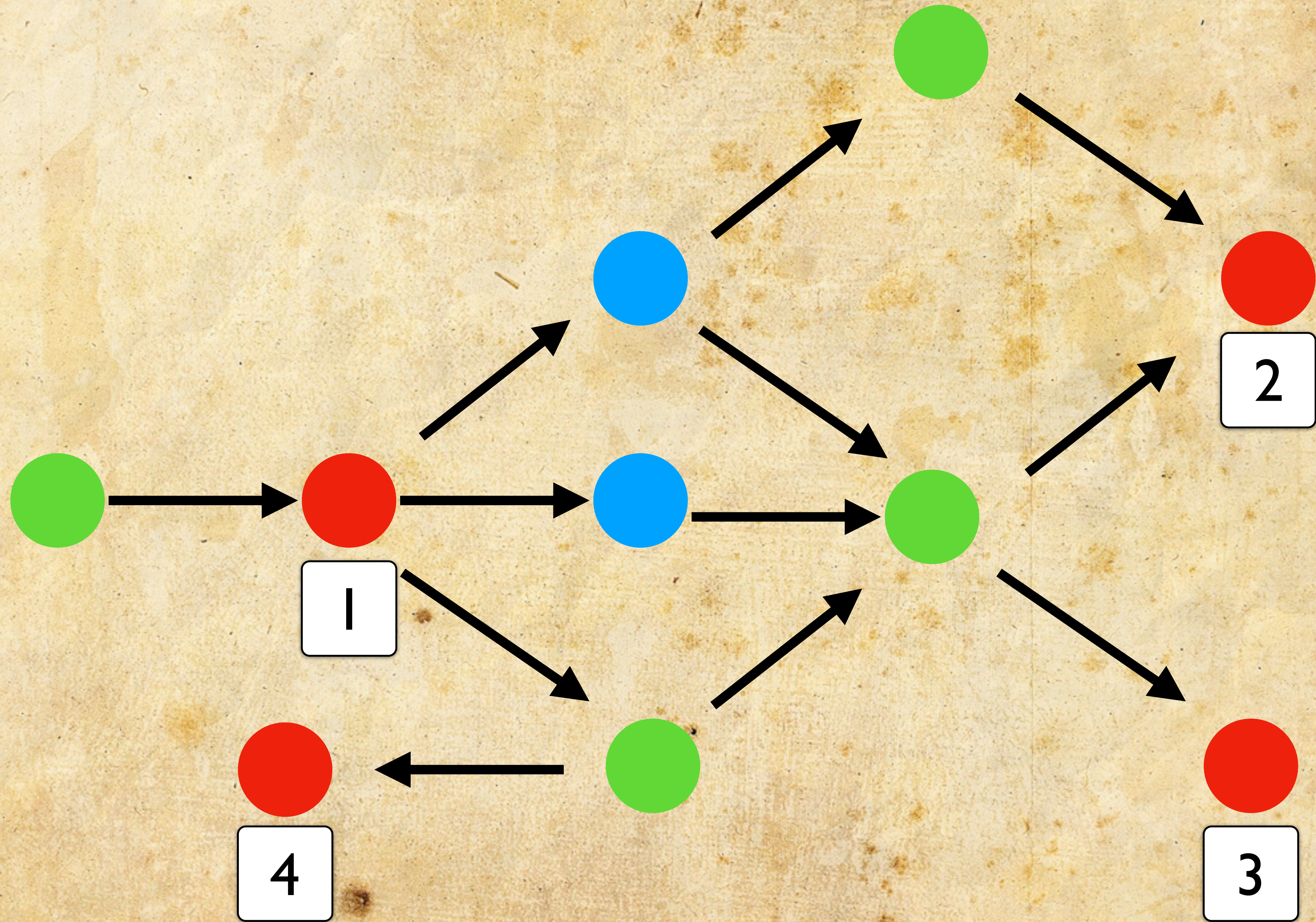


Почему так происходит?

Разные точки входа







Как действовать?



Как действовать?

Общение с ручными тестированием

Как действовать?

Общение с ручными тестированием

Подготовка всех данных через API

Как действовать?

Общение с ручными тестированием

Подготовка всех данных через API

Урлы быстрого доступа для страниц

Доверие команды к тестам




Короткие тесты

Сервер "заглушек"



Как проверить объявление?

Volvo в Санкт-Петербурге 612 объявлений [Сохранить поиск](#)

	Volvo XC60 I Рестайлинг 2.4 AT (181 л.с.) дизель, полный привод, внедорожник 5 дв., бежевый Webasto Кожаный салон 👑 Санкт-Петербург	1 700 000 ₹	2015	75 000 км
	Volvo C30 I 2.4 AT (170 л.с.) бензин, передний привод, хэтчбек 3 дв., пурпурный 👑 Санкт-Петербург	325 000 ₹ 340 000 ₹ ↓ 15 000 ₹ ↑	2007	165 000 км ↓ Проверено Авто.ру
	Volvo S40 II Рестайлинг 1.6 MT (100 л.с.) бензин, передний привод, седан, чёрный 👑 Санкт-Петербург	419 000 ₹	2011	125 000 км

Как проверить фильтры?

Все Новые С пробегом [Сохранить поиск](#)

Volvo Модель Поколение +

Кузов Коробка Двигатель Привод Объем от, л до


Год от до Пробег от, км до Цена от, Р до

[Санкт-Петербург + 200 км](#) [Все параметры](#) [Сбросить](#) [Показать 612 объявлений](#)

850	9	S40	43	V40 Cross Country	17	XC70	76
960	8	S60	92	V50	14	XC90	159
C30	13	S80	72	XC60	73	Все модели – 39	

Продажа Volvo

По актуальности За все время Проверенные авто [?](#)



Volvo XC60 I Рестайлинг **1 700 000 Р** 2015 75 000 км

2.4 AT (181 л.с.) дизель, полный привод, внедорожник 5 дв., бежевый

Сервер "заглушек"

Какие есть реализации?

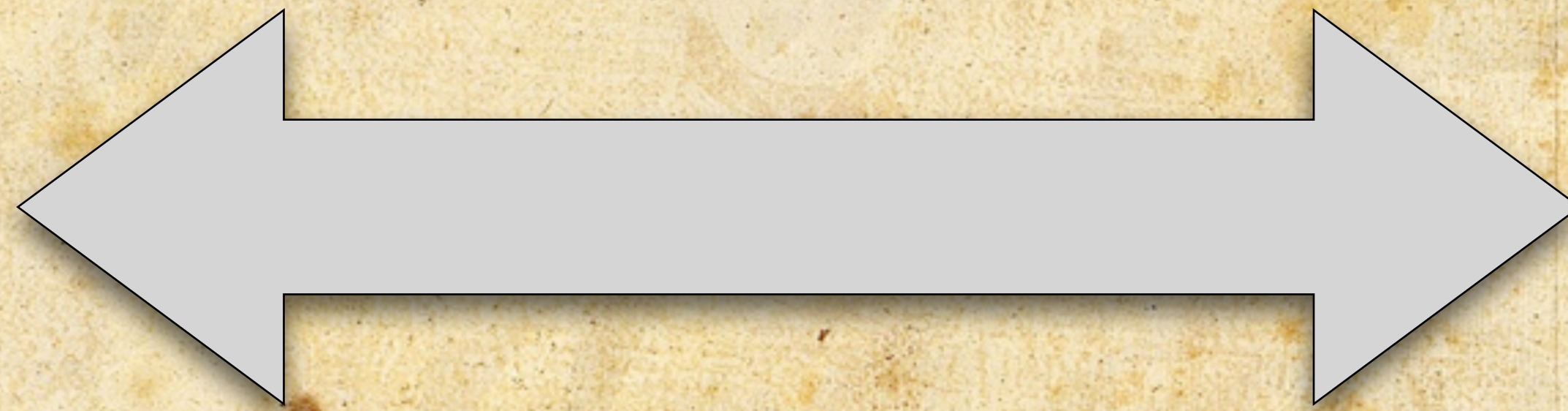
<https://www.citrusframework.org>

<http://www.mock-server.com>

<http://wiremock.org>

Как устроен сервис?

Frontend

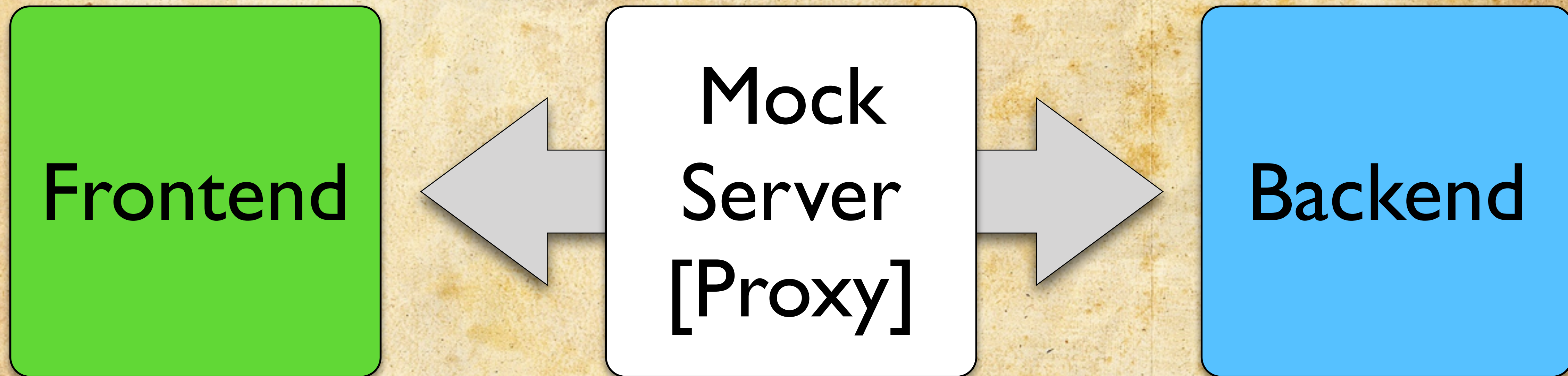


Backend

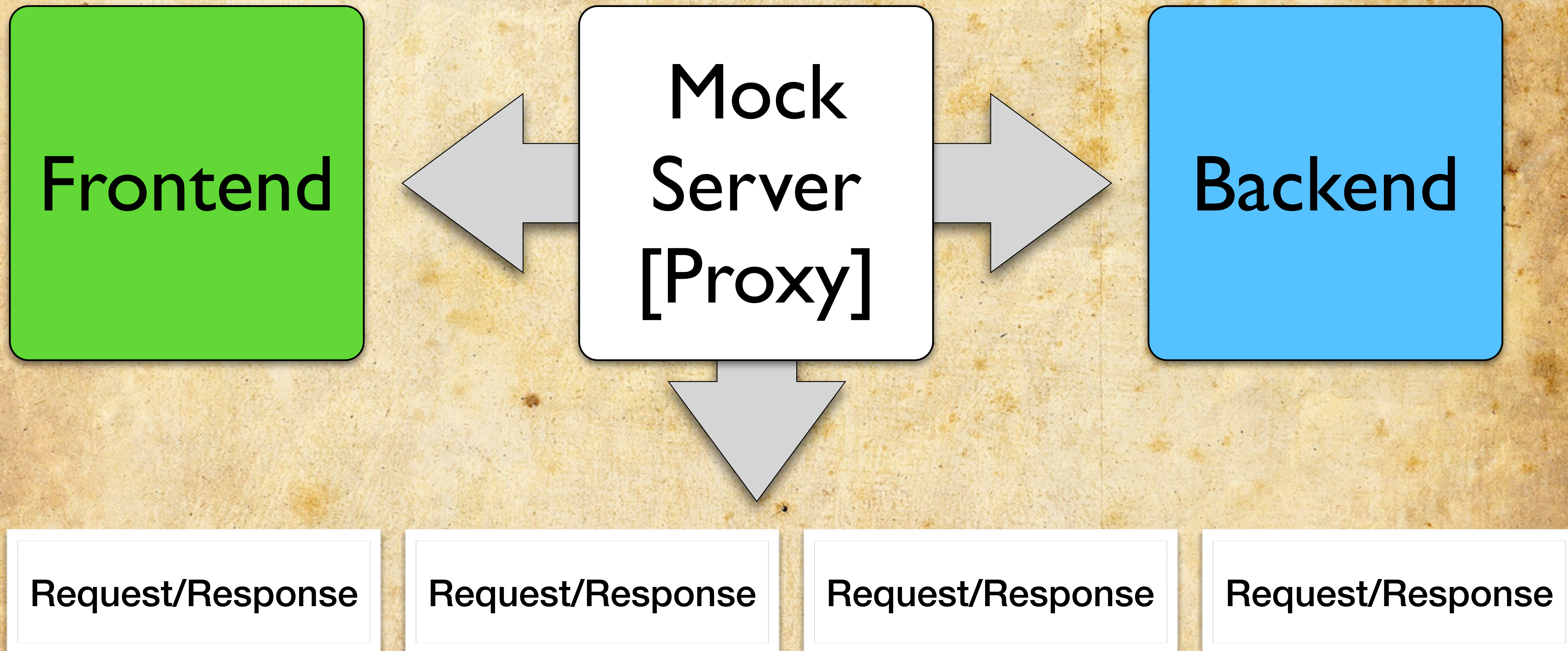
Сервер "заглушек"



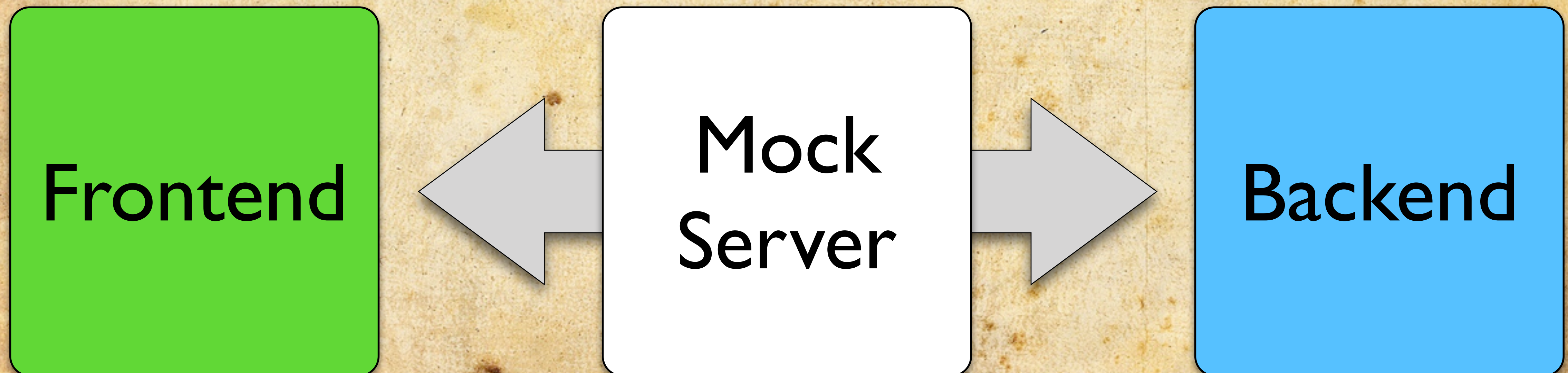
Режим прокси



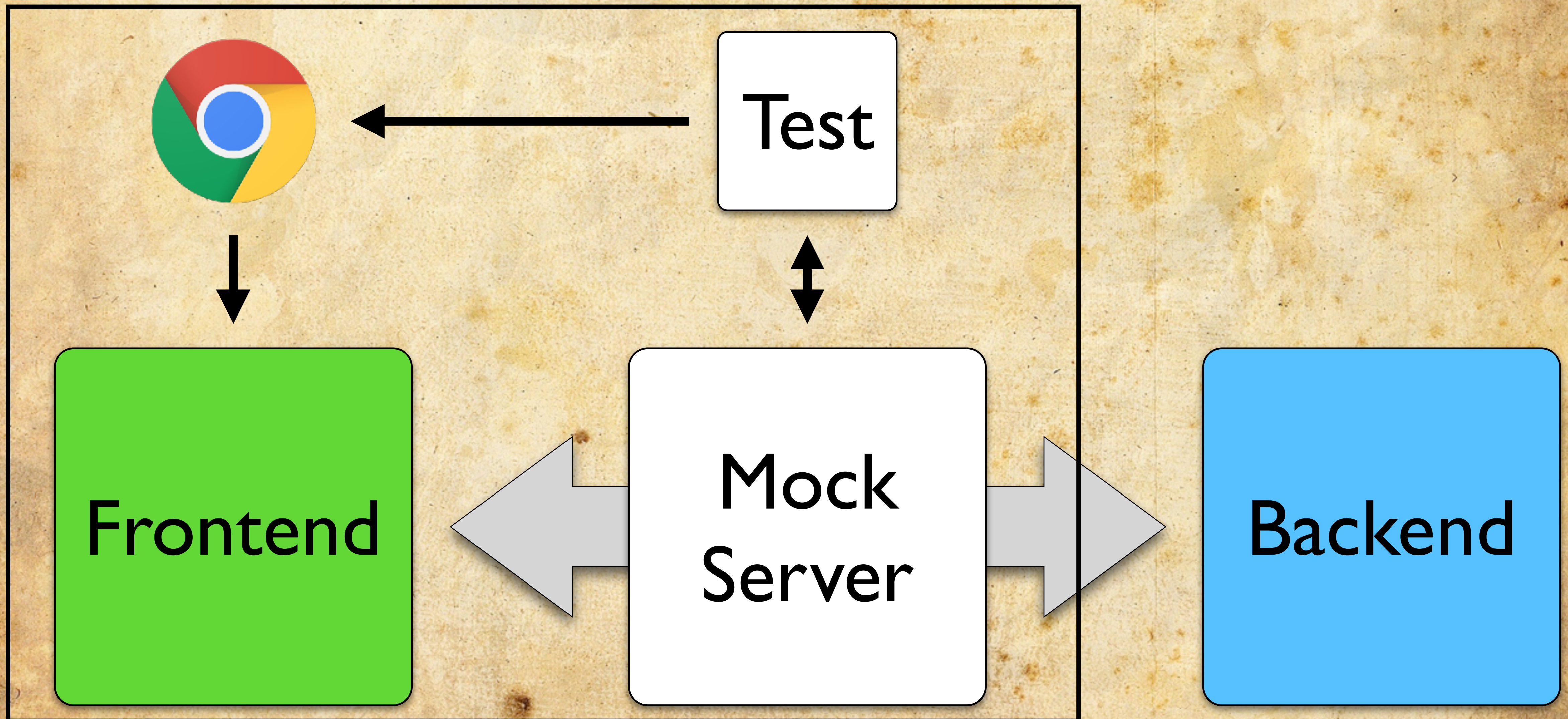
Режим прокси



Как устроено тестирование?



Как устроено тестирование?



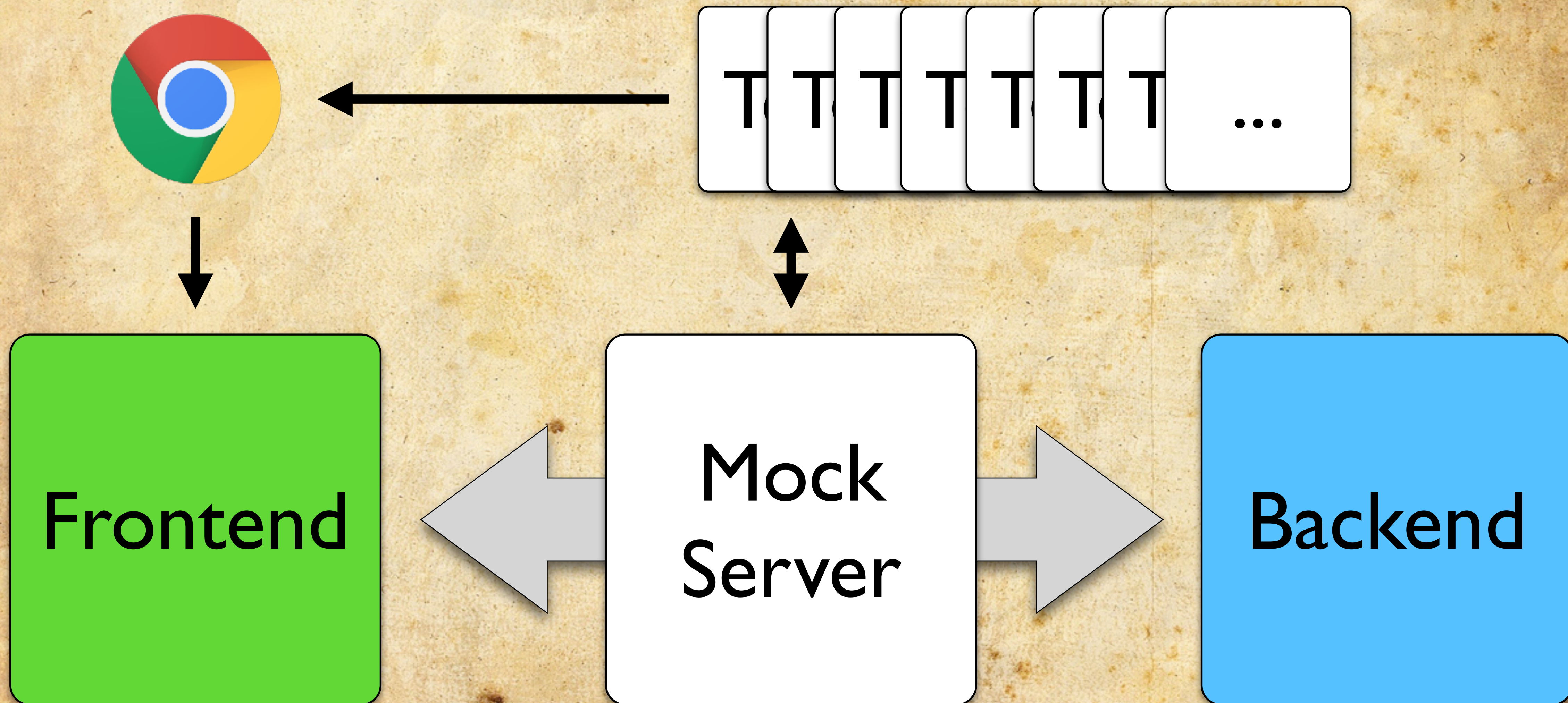
Как ВЫГЛЯДИТ КОД В ТЕСТЕ?

```
new MockServerClient("mock.company.com", 1080)
    .when(
        request()
            .withMethod("GET")
            .withPath("/search"),
        TimeToLive.exactly(TimeUnit.MINUTES, 5L)
    )
    .respond(
        response()
            .withStatusCode(200)
            .withBody("{ ... }")
    );
```

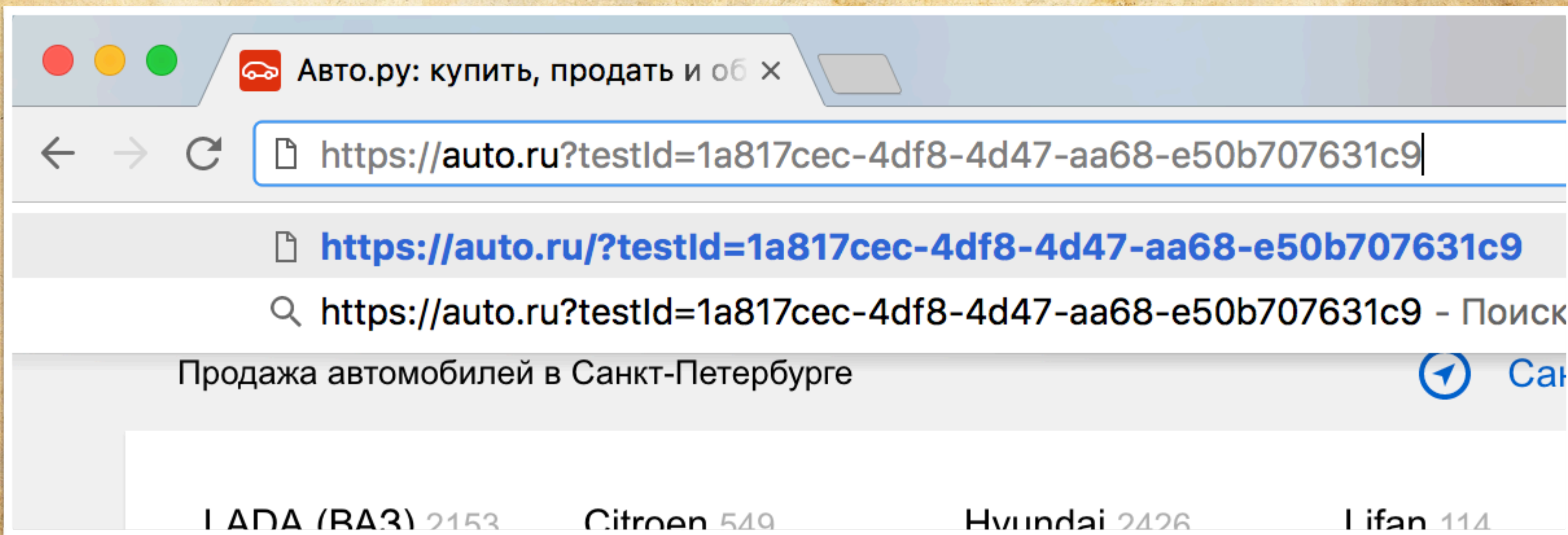
Request/Response

Request/Response

А если в несколько потоков?

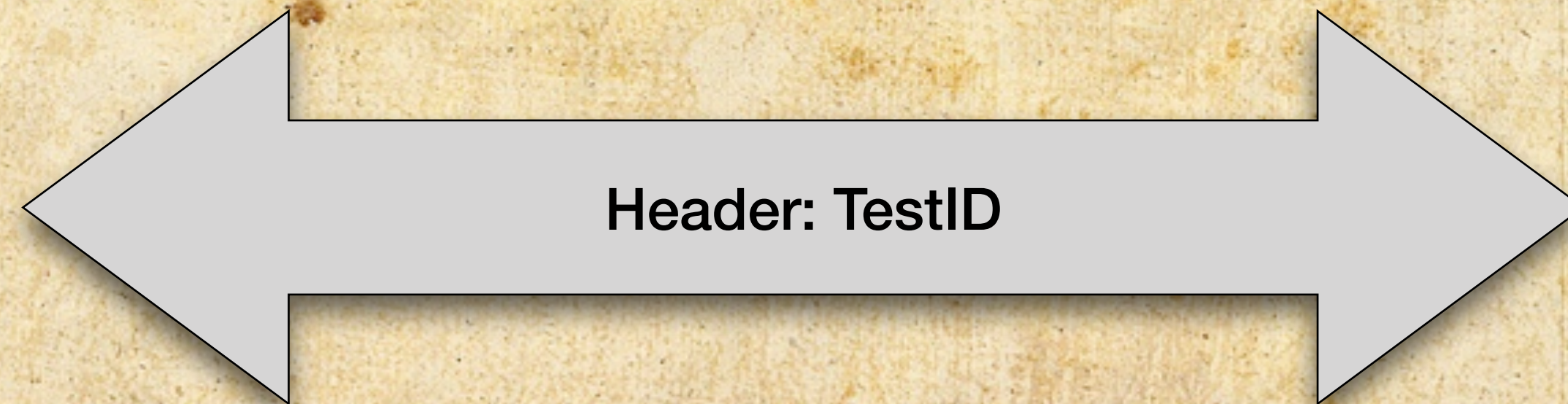
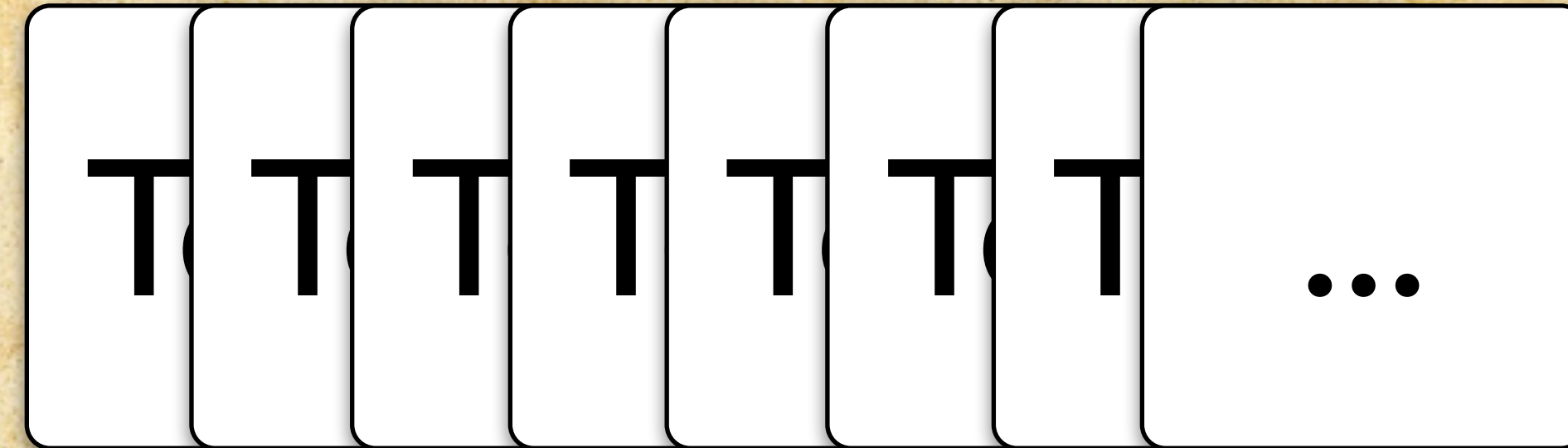


Идентификатор теста

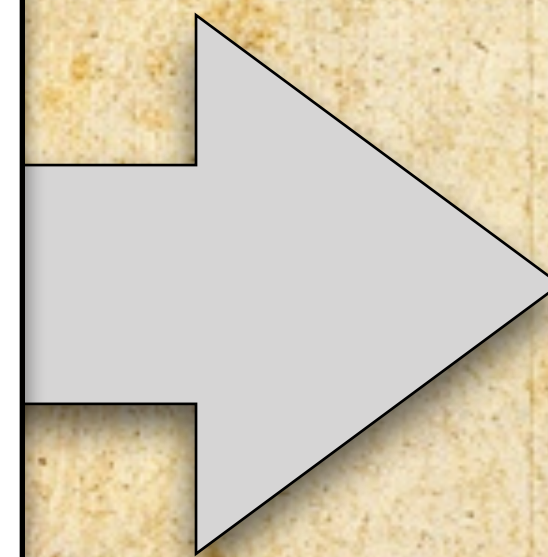
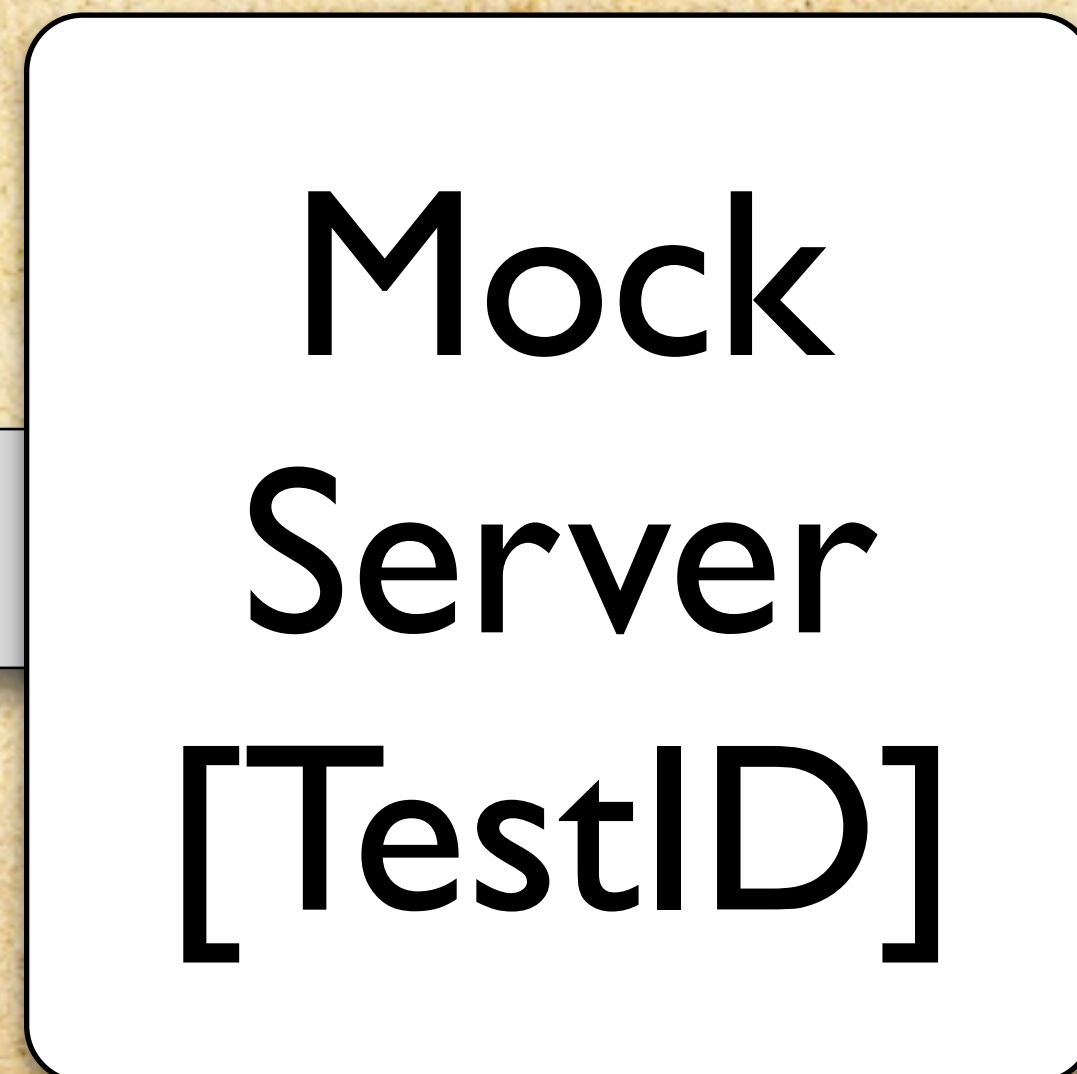
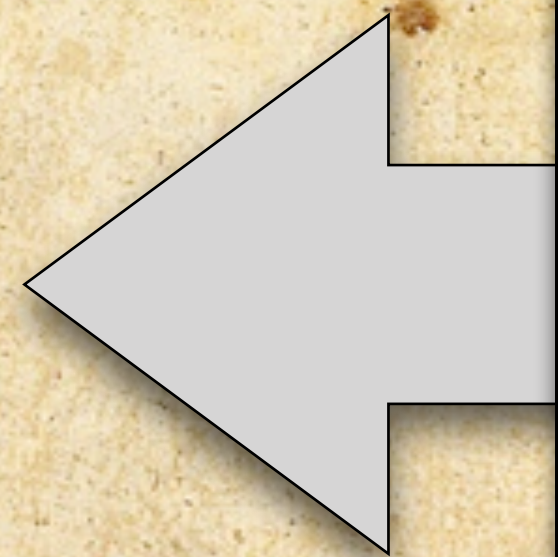
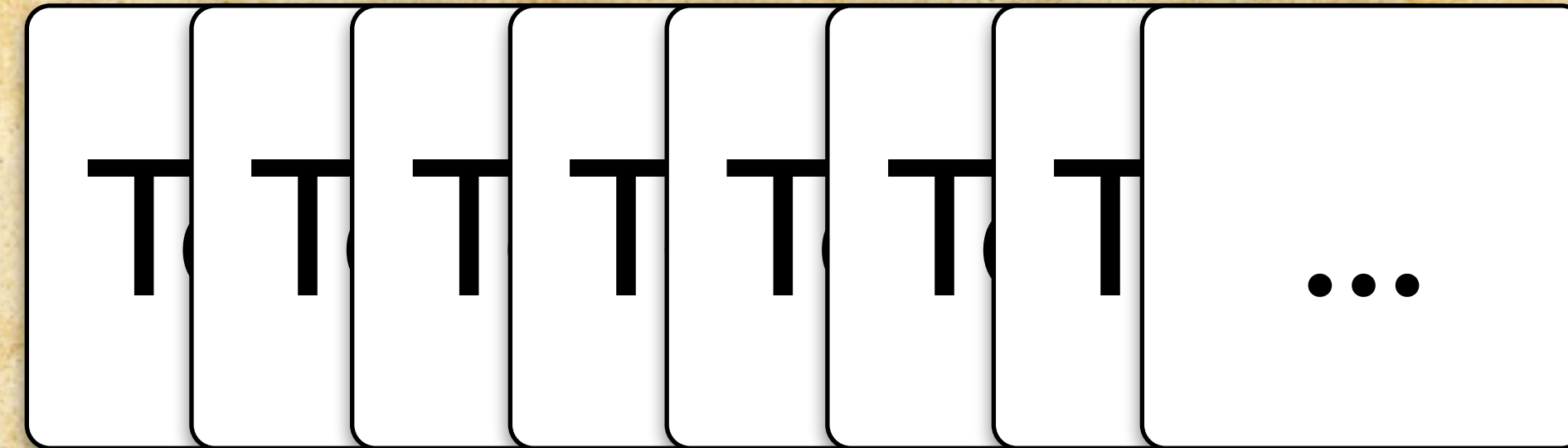


TestID: 1a817cec-4df8-4d47-aa68-e50b707631c9

Заголовок в HTTP запросе



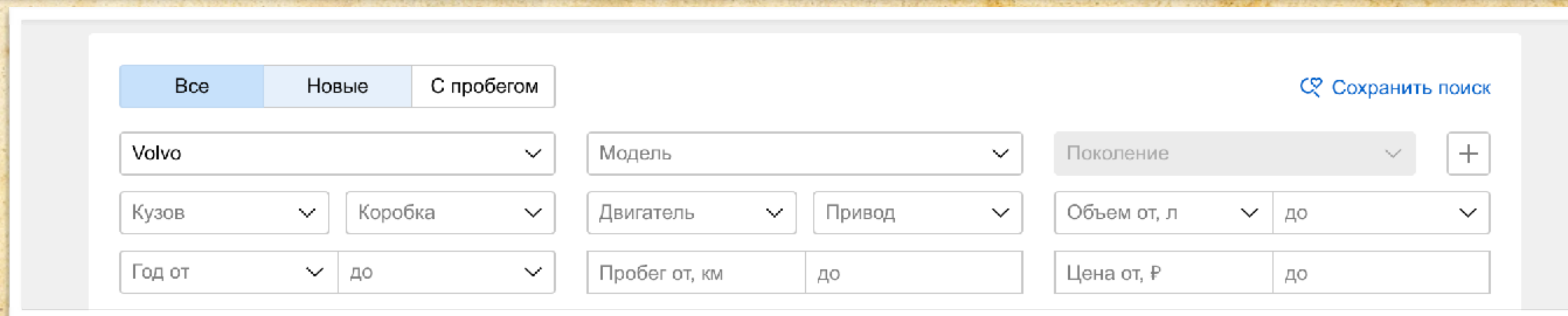
Заголовок в HTTP запросе



Как ВЫГЛЯДИТ КОД В ТЕСТЕ?

```
new MockServerClient("mock.company.com", 1080)
    .when(
        request()
            .withMethod("GET")
            .withPath("/search")
            .withHeader("TestID", testID),
        TimeToLive.exactly(TimeUnit.MINUTES, 5L)
    )
    .respond(
        response()
            .withStatusCode(200)
            .withBody("{ ... }")
    );
```

Как ВЫГЛЯДИТ КОД В ТЕСТЕ?



The screenshot shows a car search interface with the following filters and options:

- Buttons: Все (selected), Новые, С пробегом
- Search: Сохранить поиск
- Brand: Volvo
- Model: Модель
- Generation: Поколение
- Body: Кузов
- Transmission: Коробка
- Engine: Двигатель
- Drive: Привод
- Volume: Объем от, л до
- Year: Год от до
- Mileage: Пробег от, км до
- Price: Цена от, Р до

```
new MockServerClient("mock.company.com", 1080)
    .verify(
        request()
            .withMethod("GET")
            .withPath("/search")
            .withHeader("TestID", testID)
            .withQueryStringParameter("model", "Volvo"),
        VerificationTimes.exactly(1)
    );
```

Сервер "заглушек"

Доверие команды к тестам



Короткие тесты

Сервер "заглушек"

Allure

Suites

name duration status Filter by status: 0 0 455 0 0

- > Коммерческие - Автобусы - Базовые фильтры поиска 7
- > Коммерческие - Грузовики - Базовые фильтры поиска 8
- > Коммерческие - Лёгкие коммерческие - Базовые фильтры поиска 1
- > Коммерческие - Лёгкие коммерческие - Базовые фильтры поиска - Грузоподъёмность 4
- > Коммерческие - Лёгкие коммерческие - Базовые фильтры поиска - Прицепы 4
- > Коммерческие - Прицепы - Базовые фильтры поиска



Filter by status: 0 0 455 0 0

КА 7

КА 8

Message

A close-up, slightly blurred photograph of a wooden table. On the table, there is a laptop with a light-colored cover and a notebook with a dark cover. The notebook is open, showing some pages with red and yellow markings. The background is a plain, light-colored wall.

Простота

Доверие

КОНТРОЛЬ



Частые вопросы

Launches types

Project

Component

Env

first

web

docker

second

api

testing

third

db

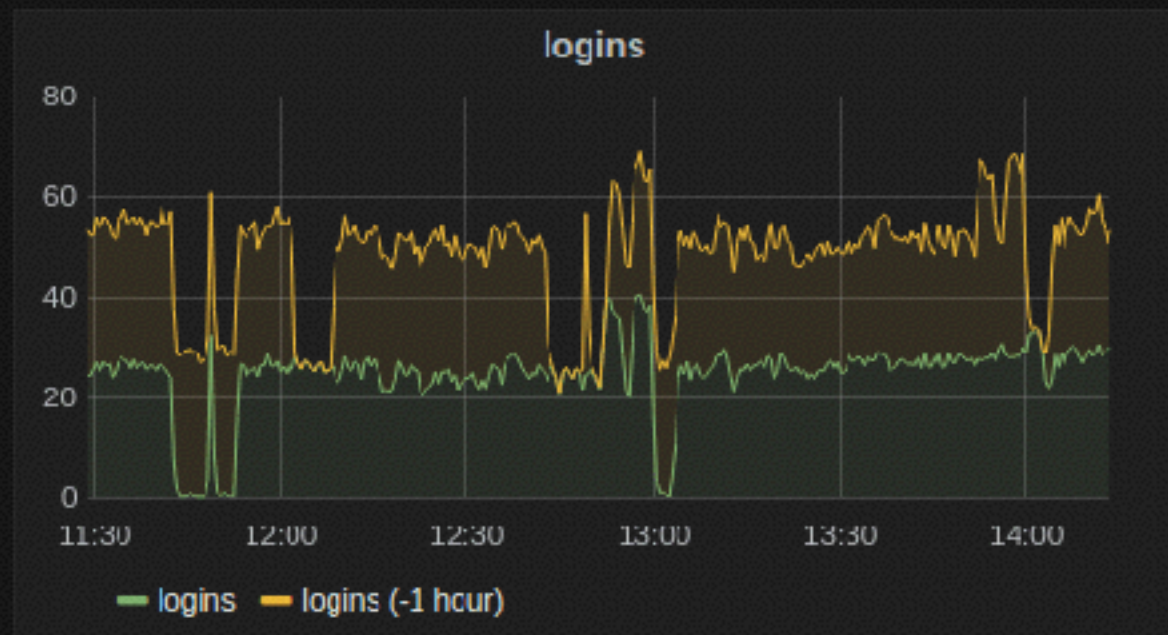
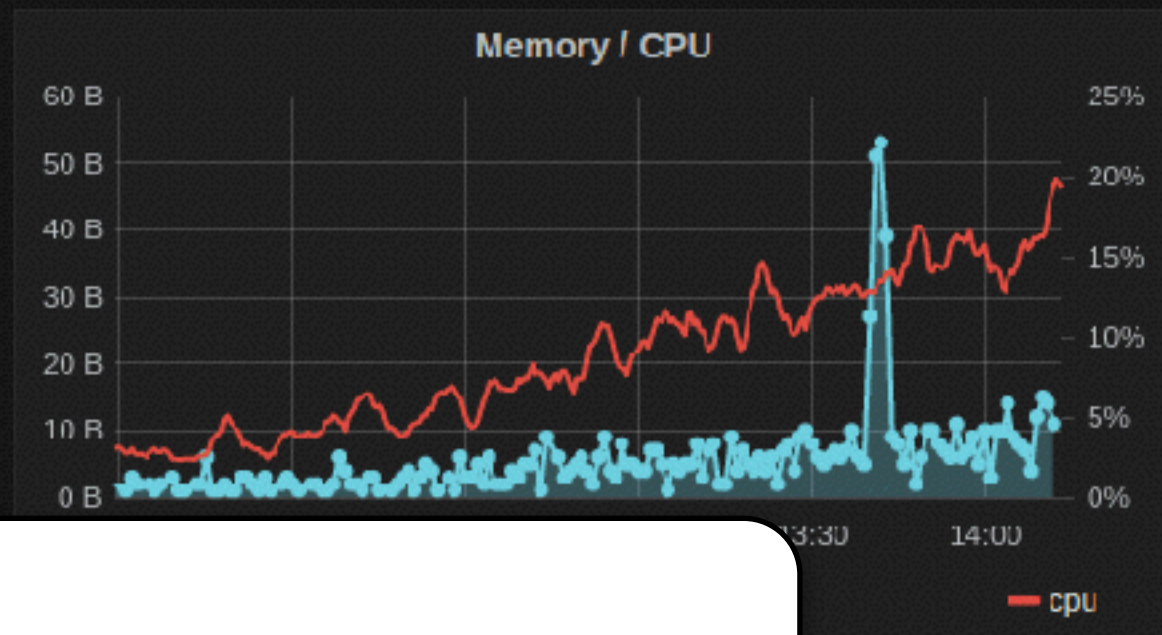
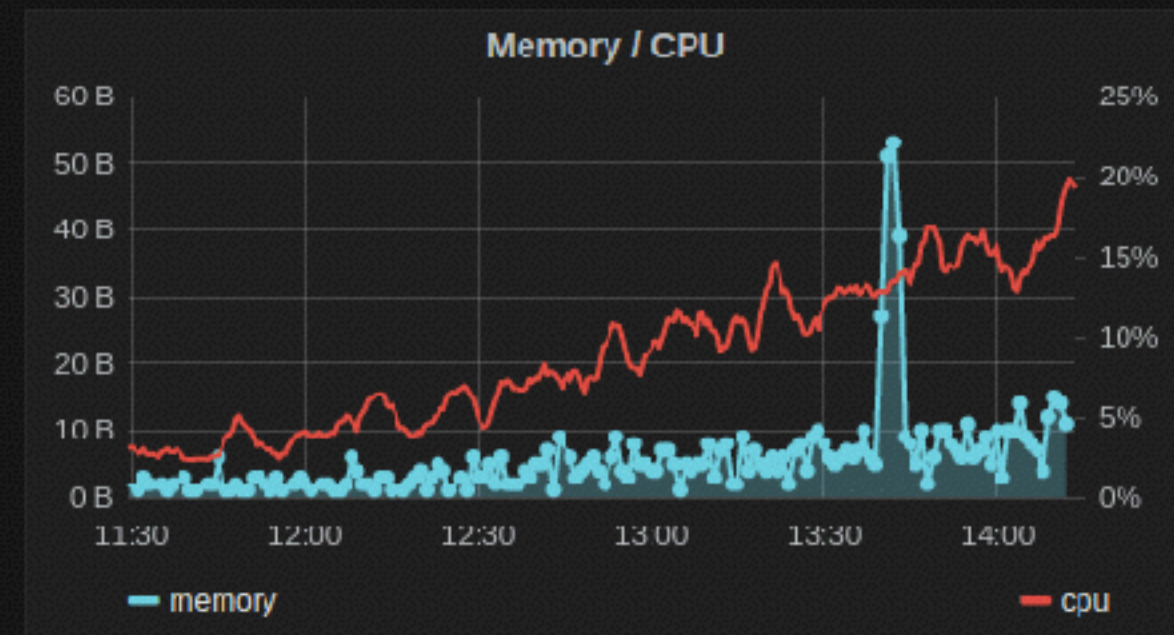
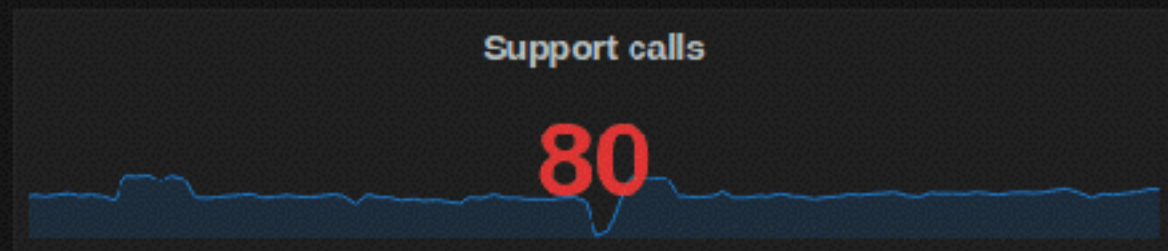
prestable



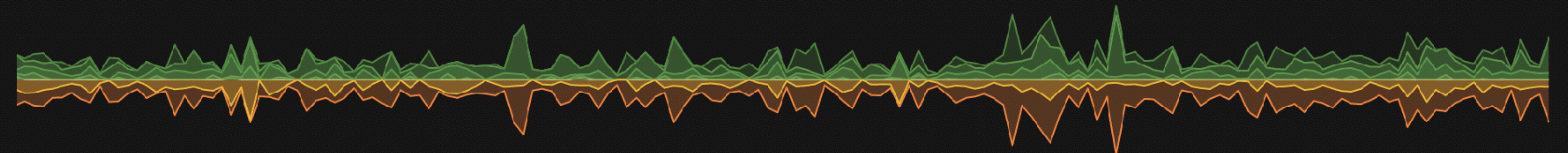
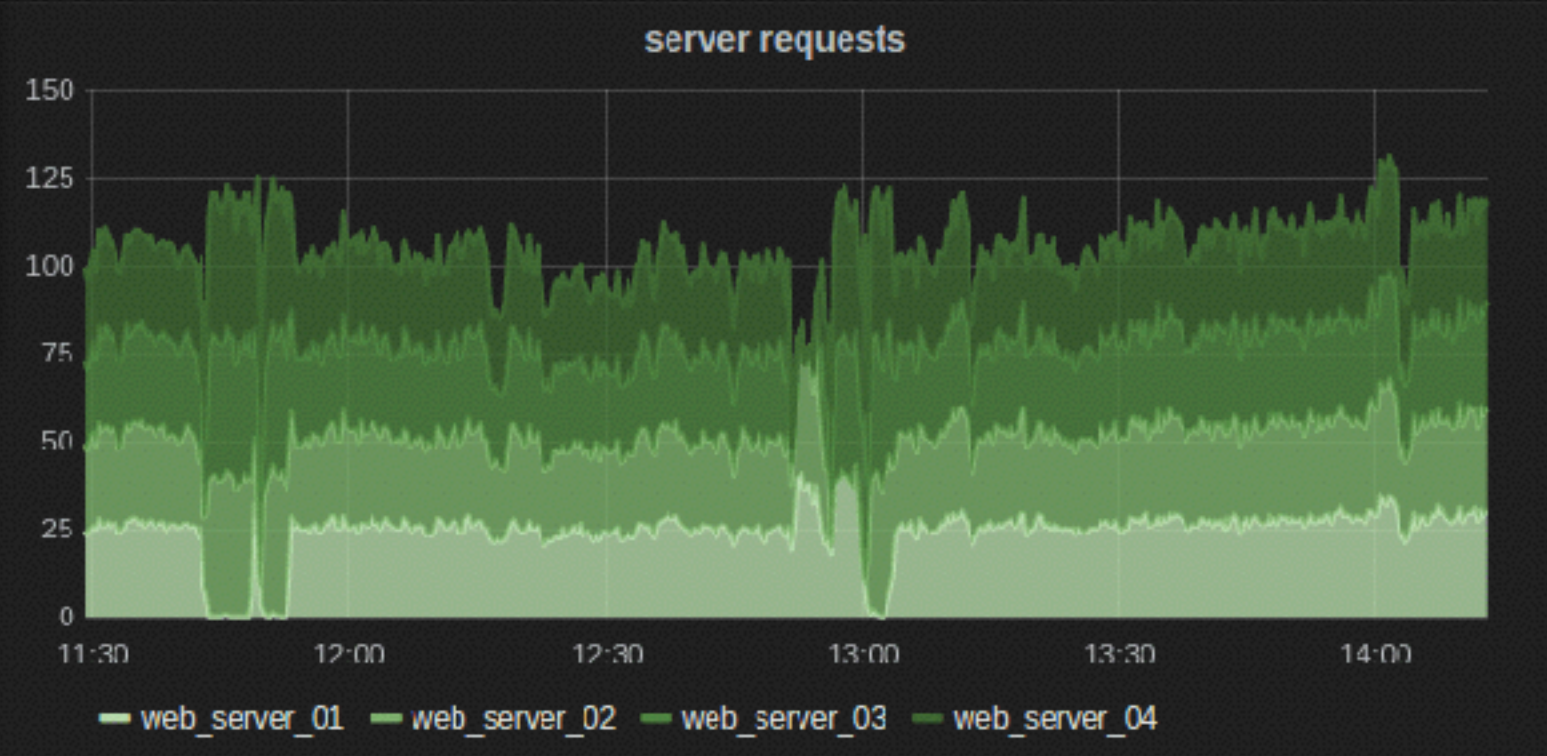
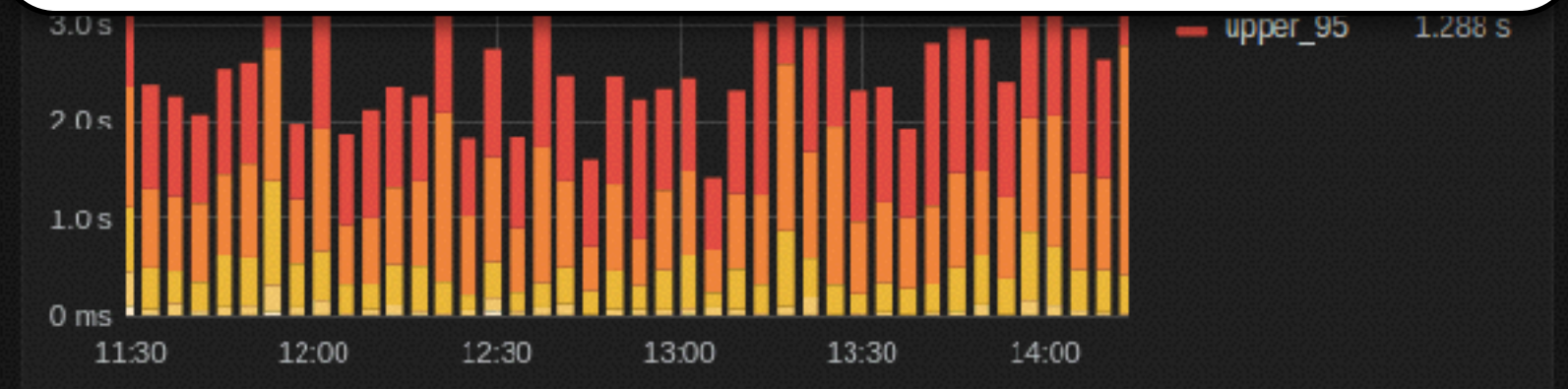
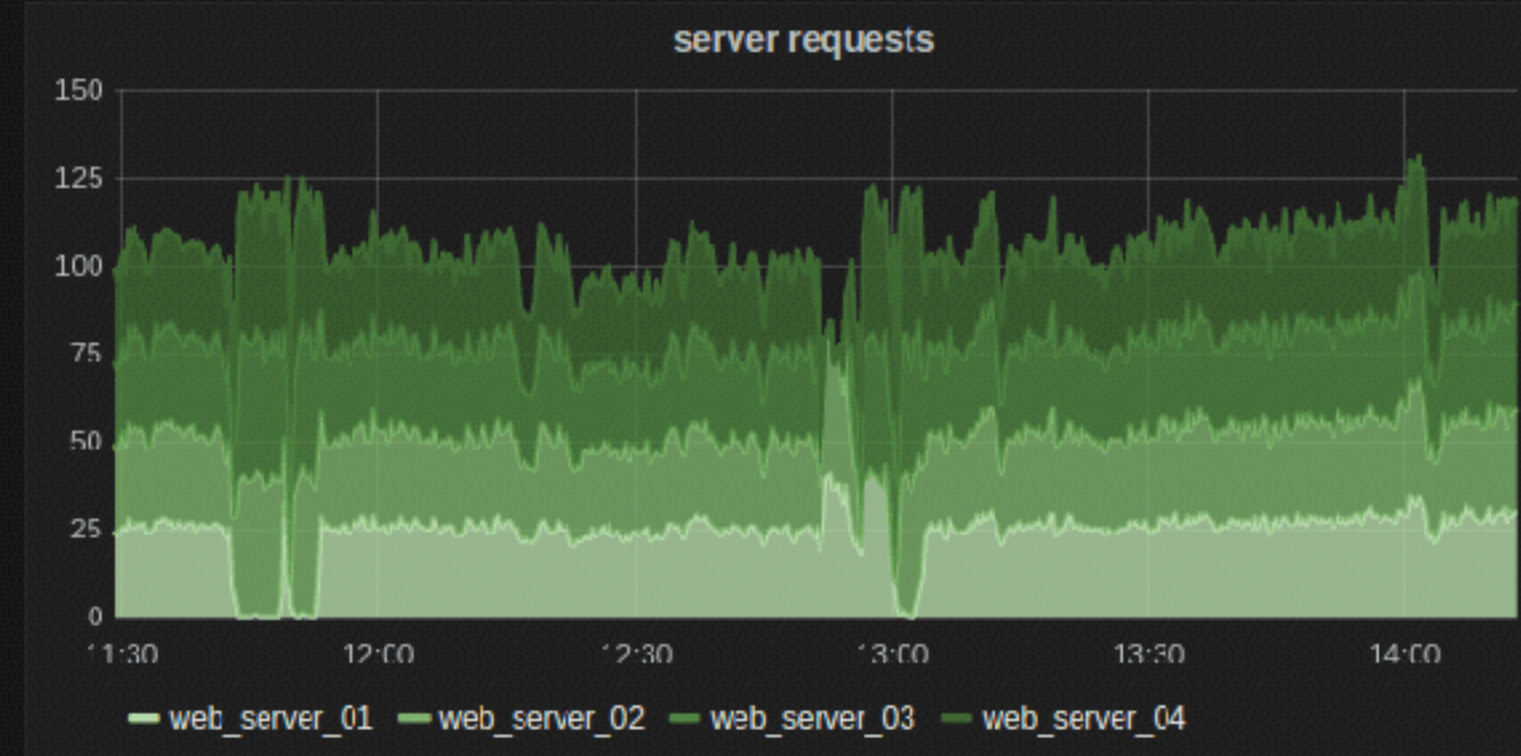
Big Dashboard



Zoom Out Oct 4, 2015 11:29:09 to Oct 4, 2015 14:13:23 UTC



Grafana

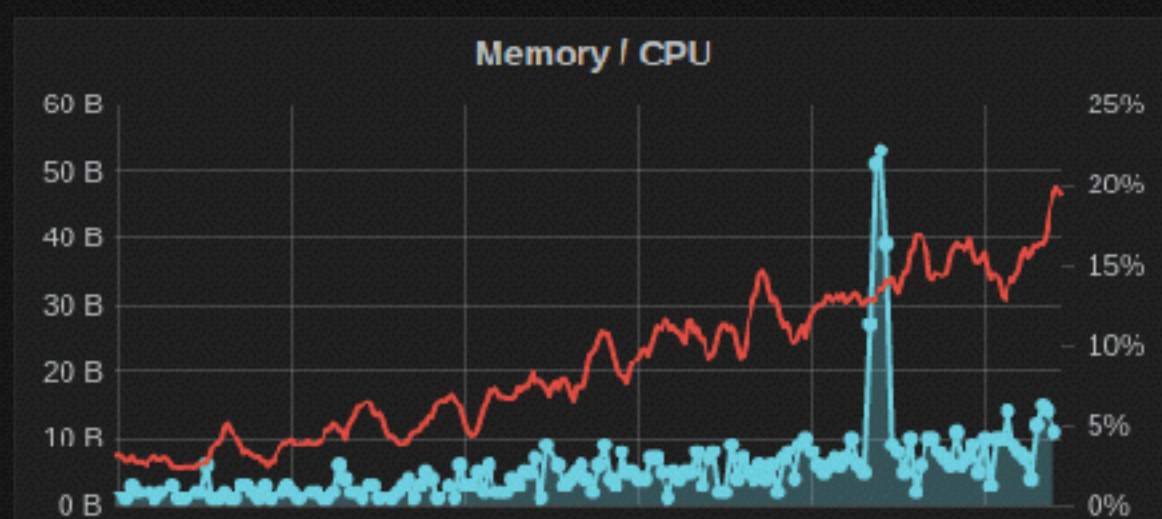
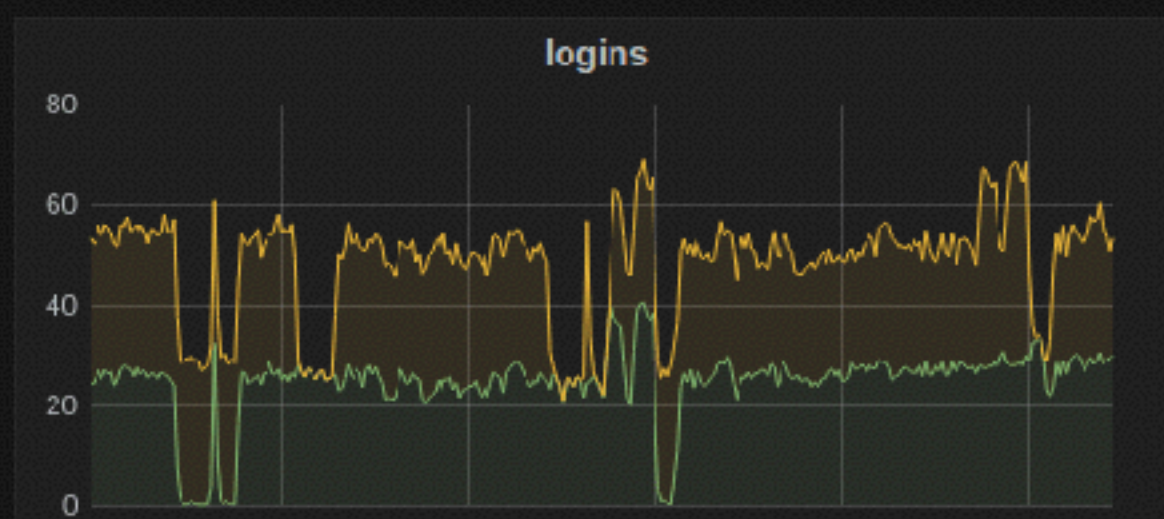
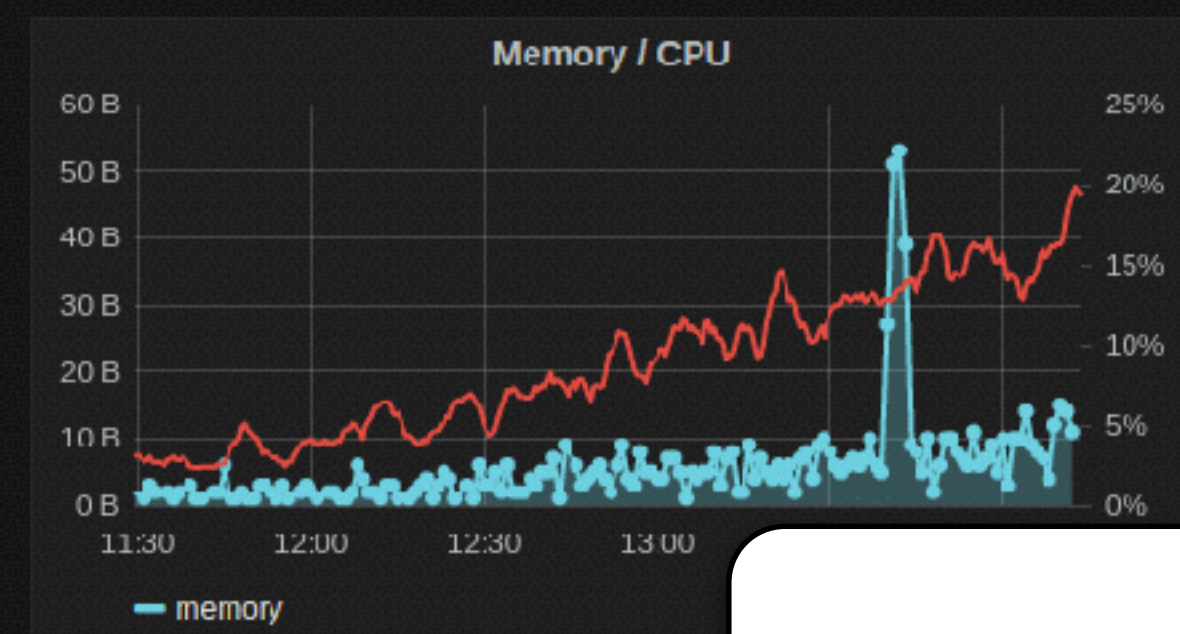
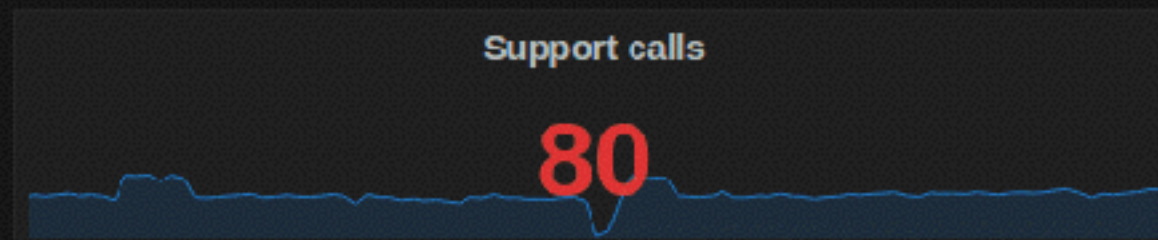




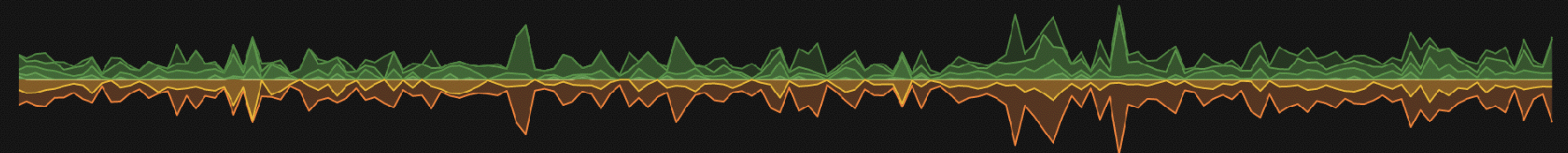
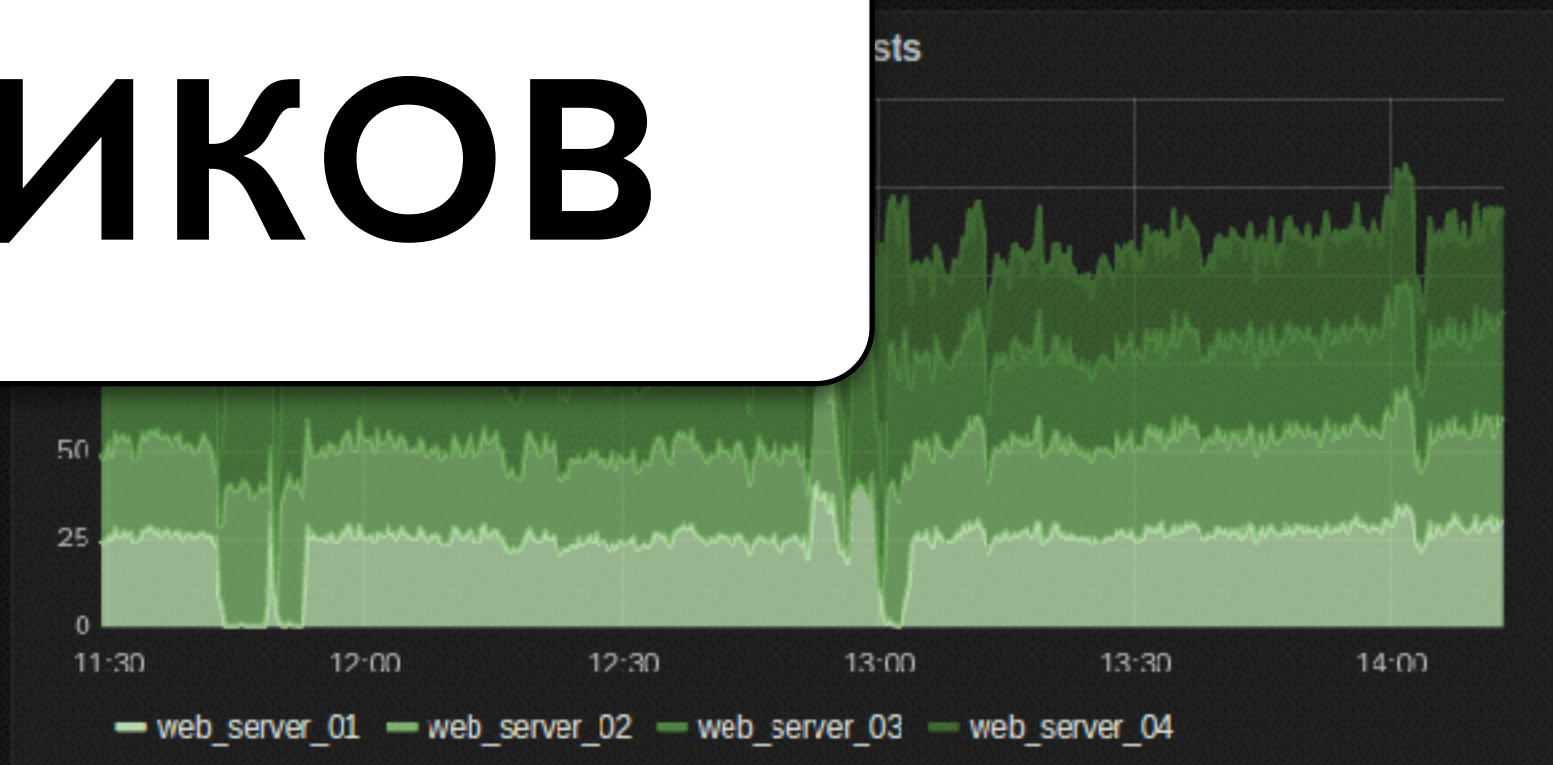
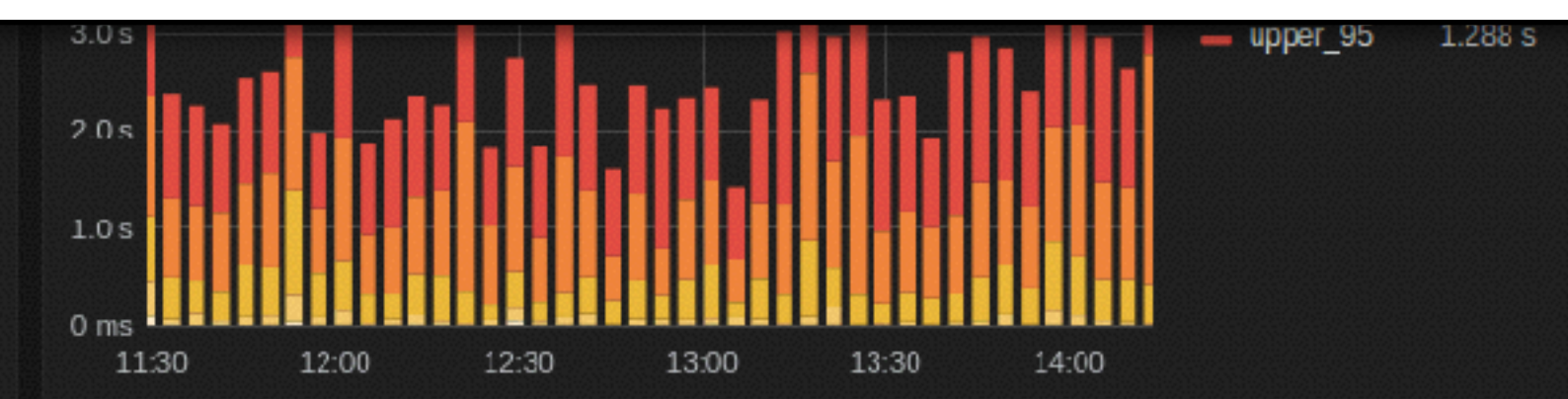
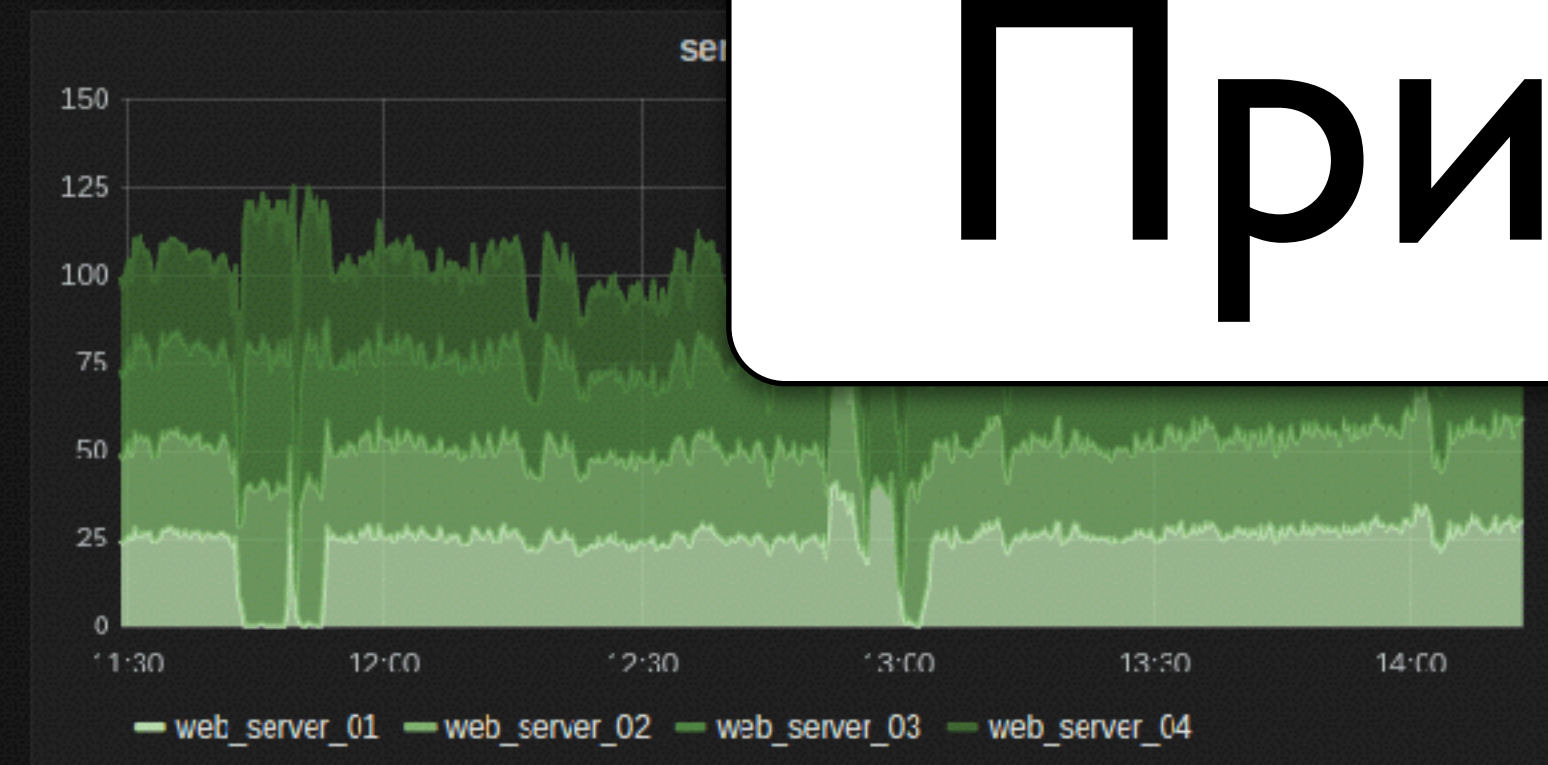
Big Dashboard



Zoom Out Oct 4, 2015 11:29:09 to Oct 4, 2015 14:13:23 UTC



Примеры графиков

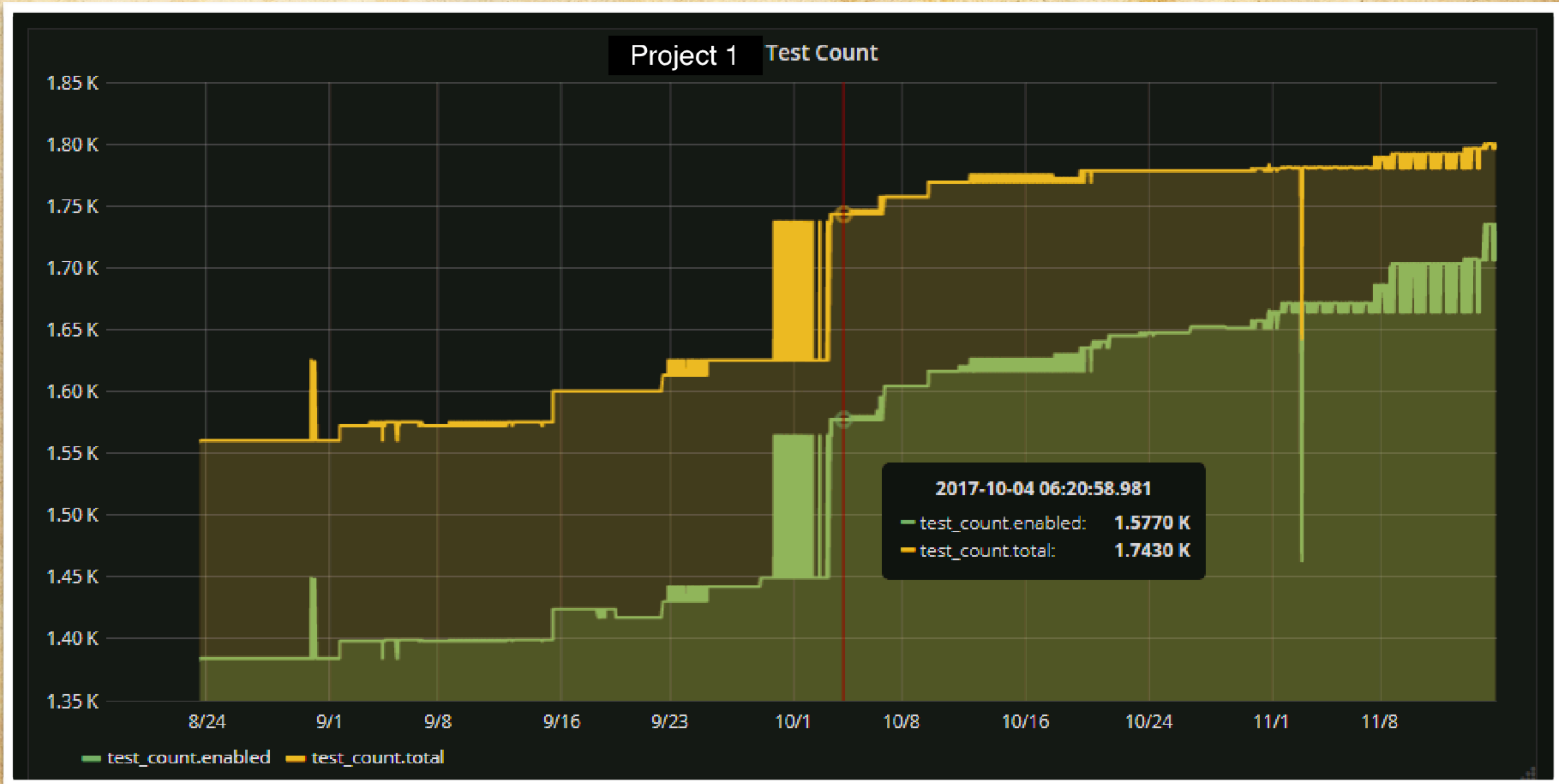


Project 1 Test Count

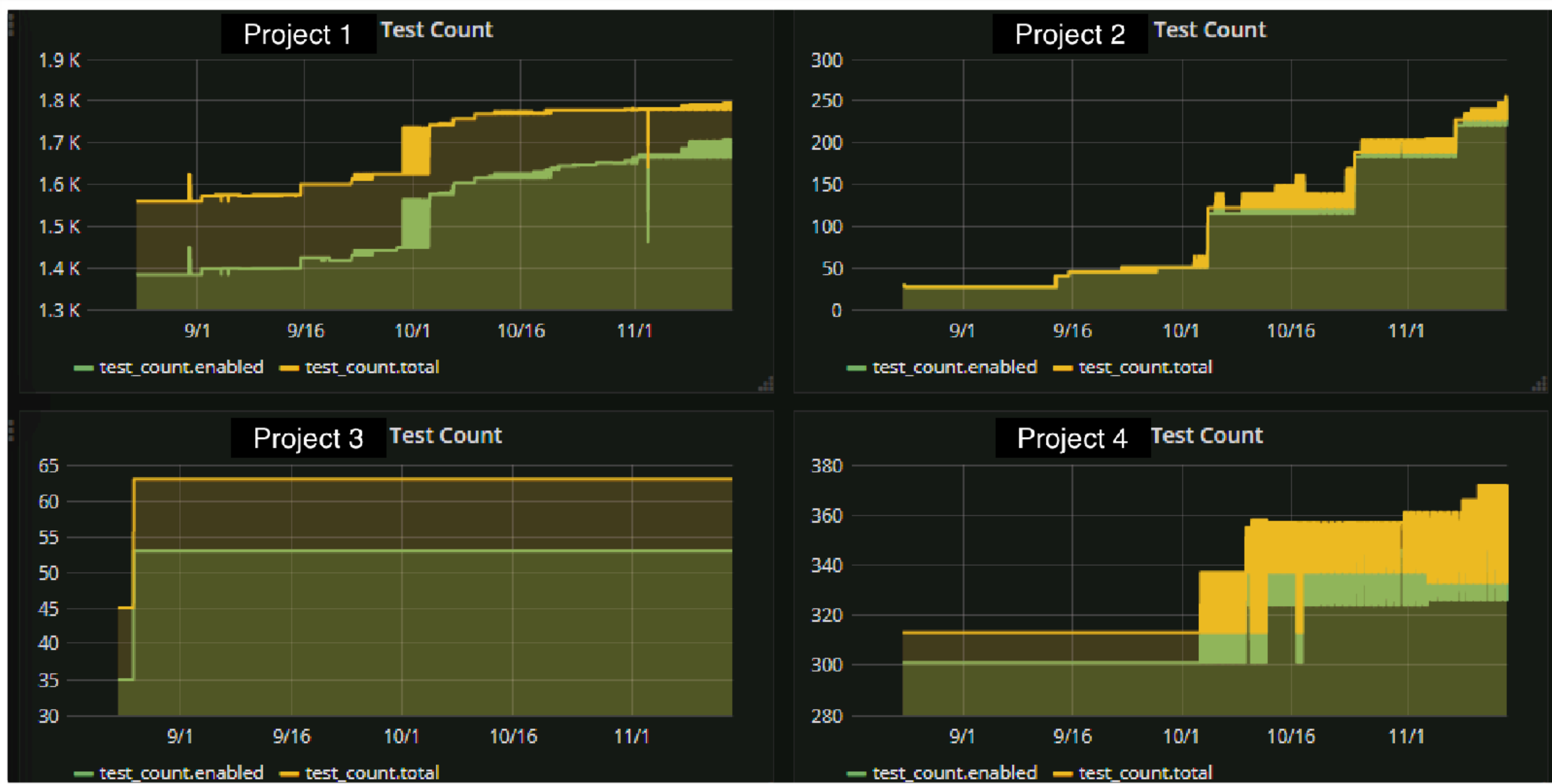
Количество тестов



Количество тестов



Количество тестов



Project 4 - smoke - test

Статус запуска тестов



Статусы теста

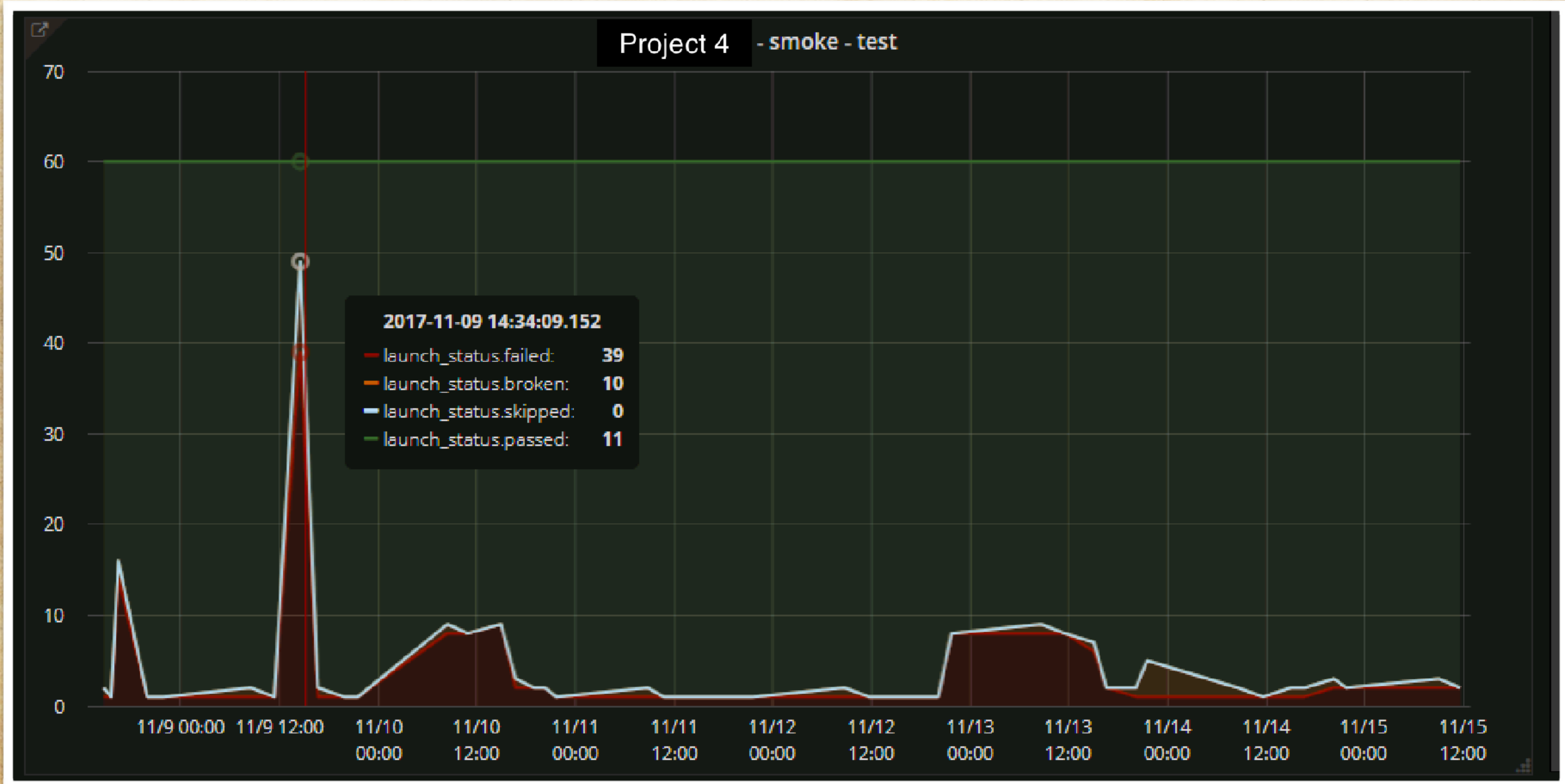
skipped

broken

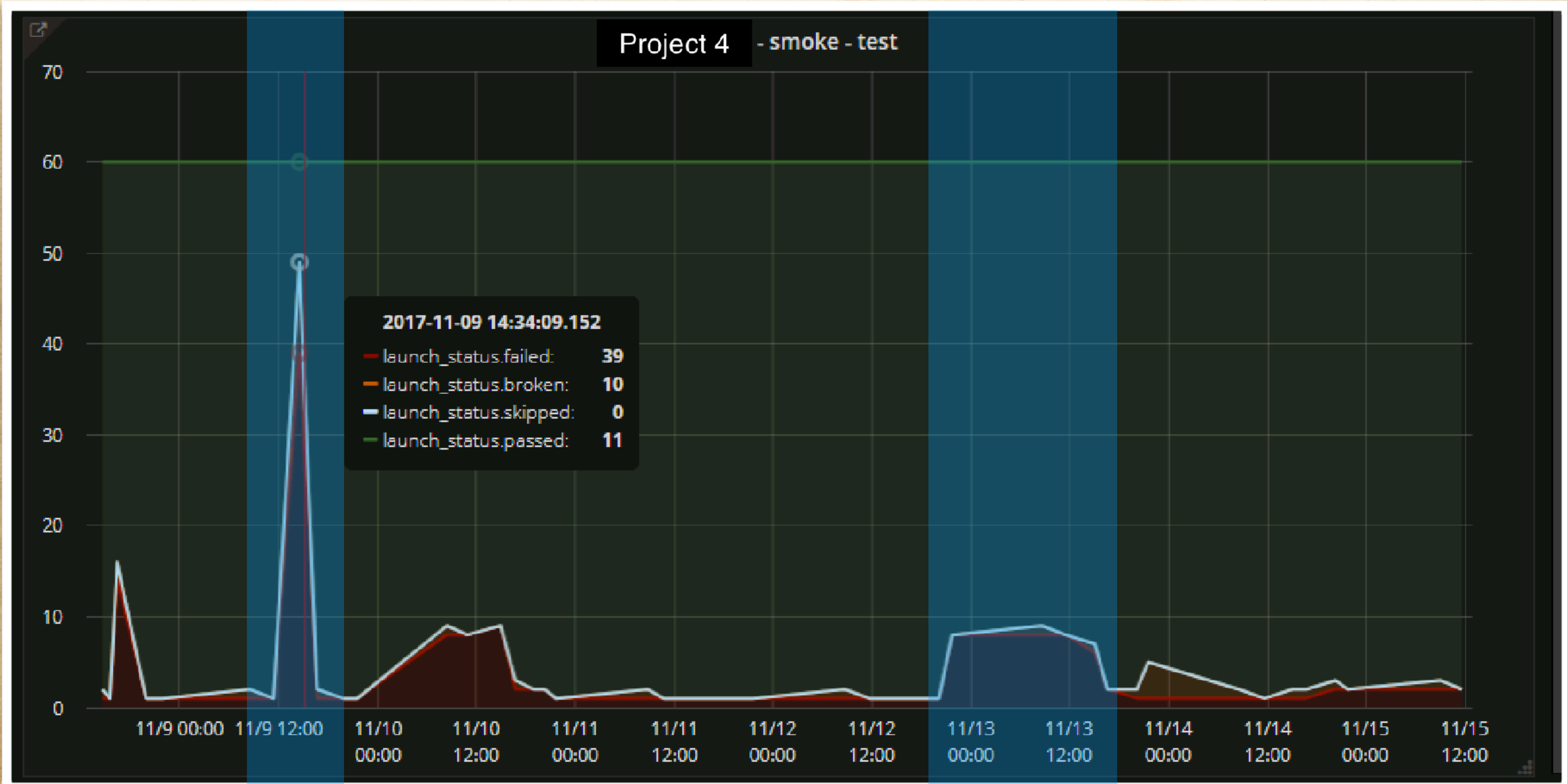
failed

passed

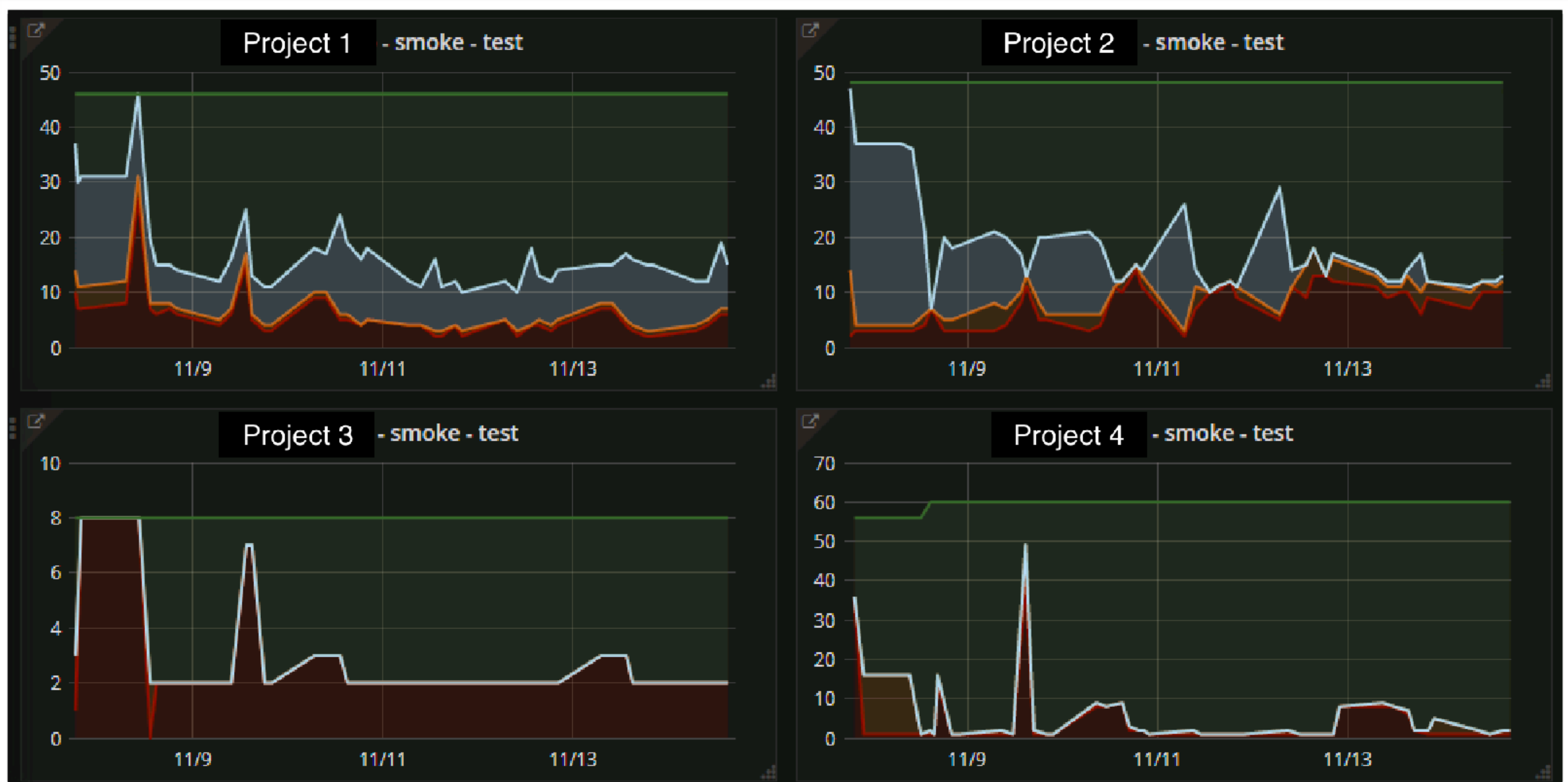
Статус запуска тестов



Статус запуска тестов



Статус запуска тестов

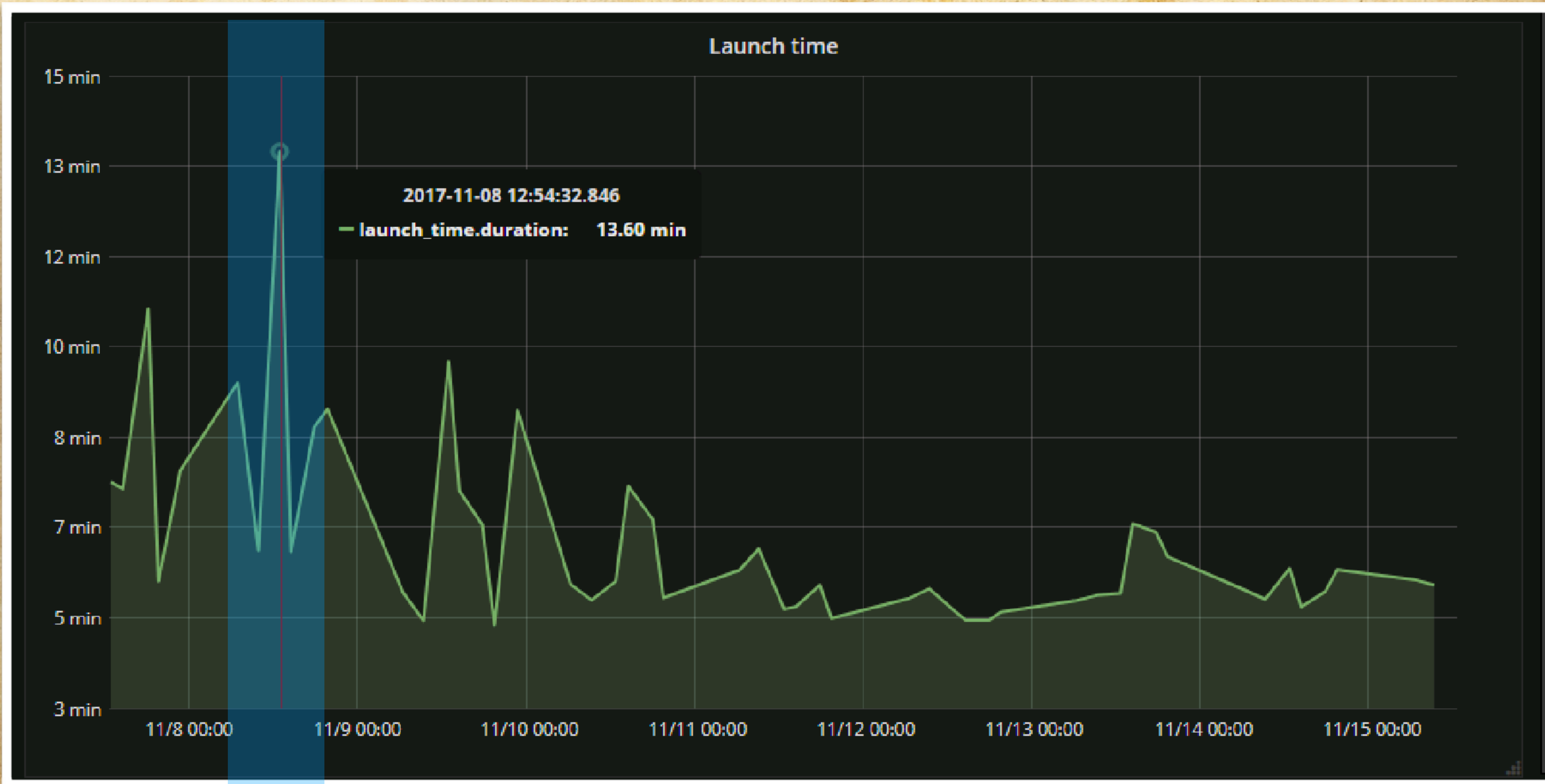


Launch time

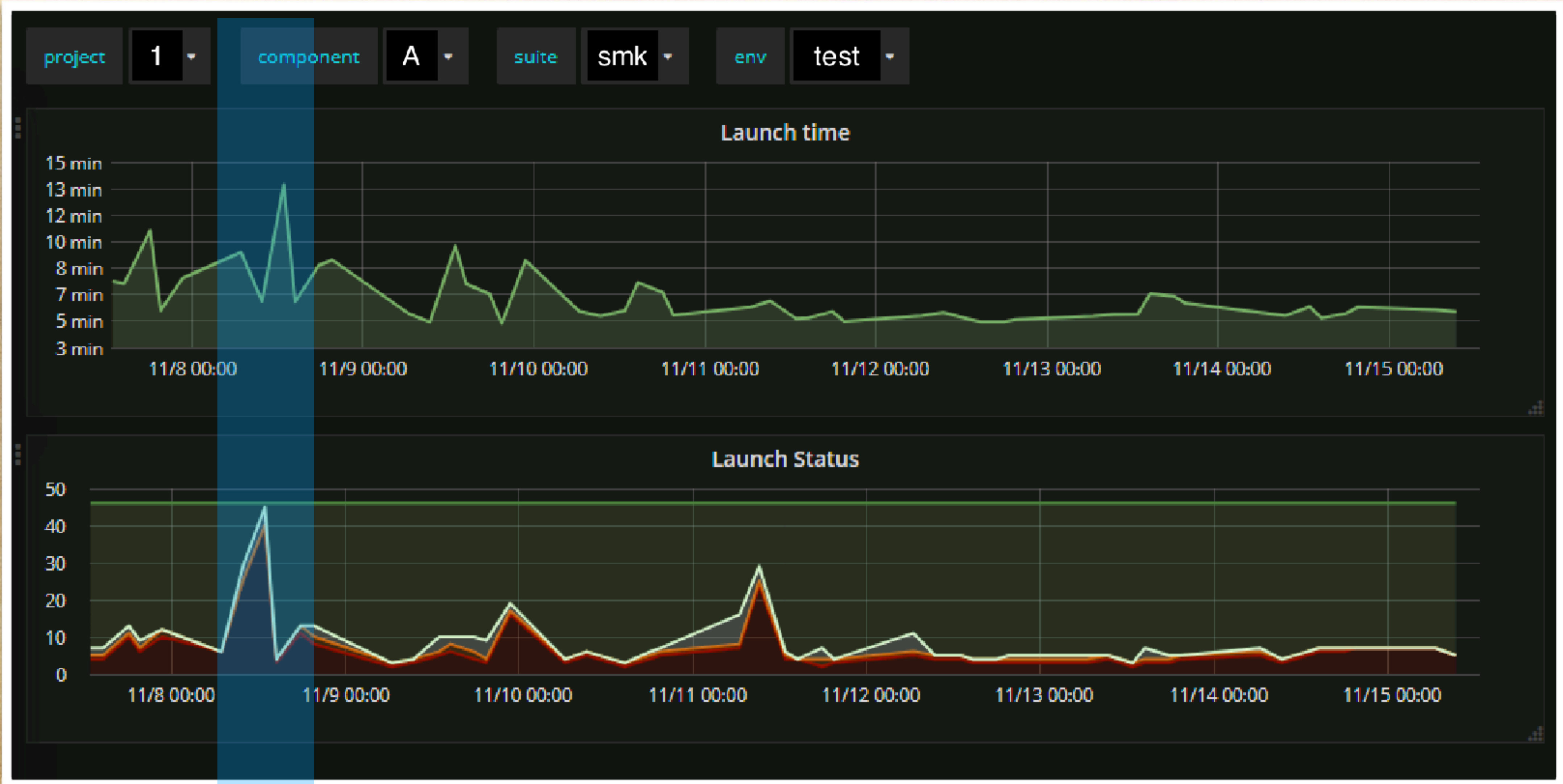


Время запуска тестов

Время запуска тестов



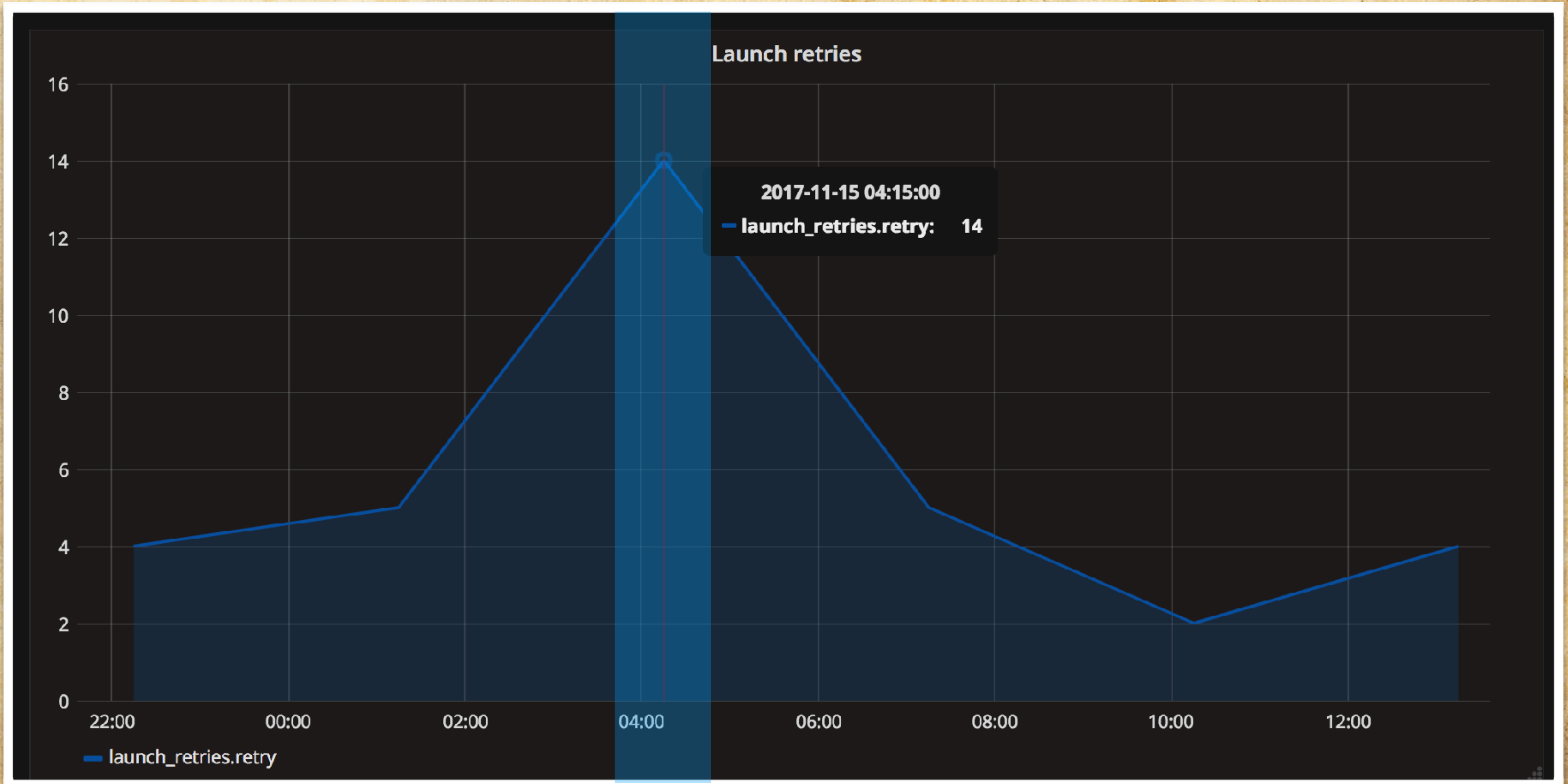
Время запуска тестов



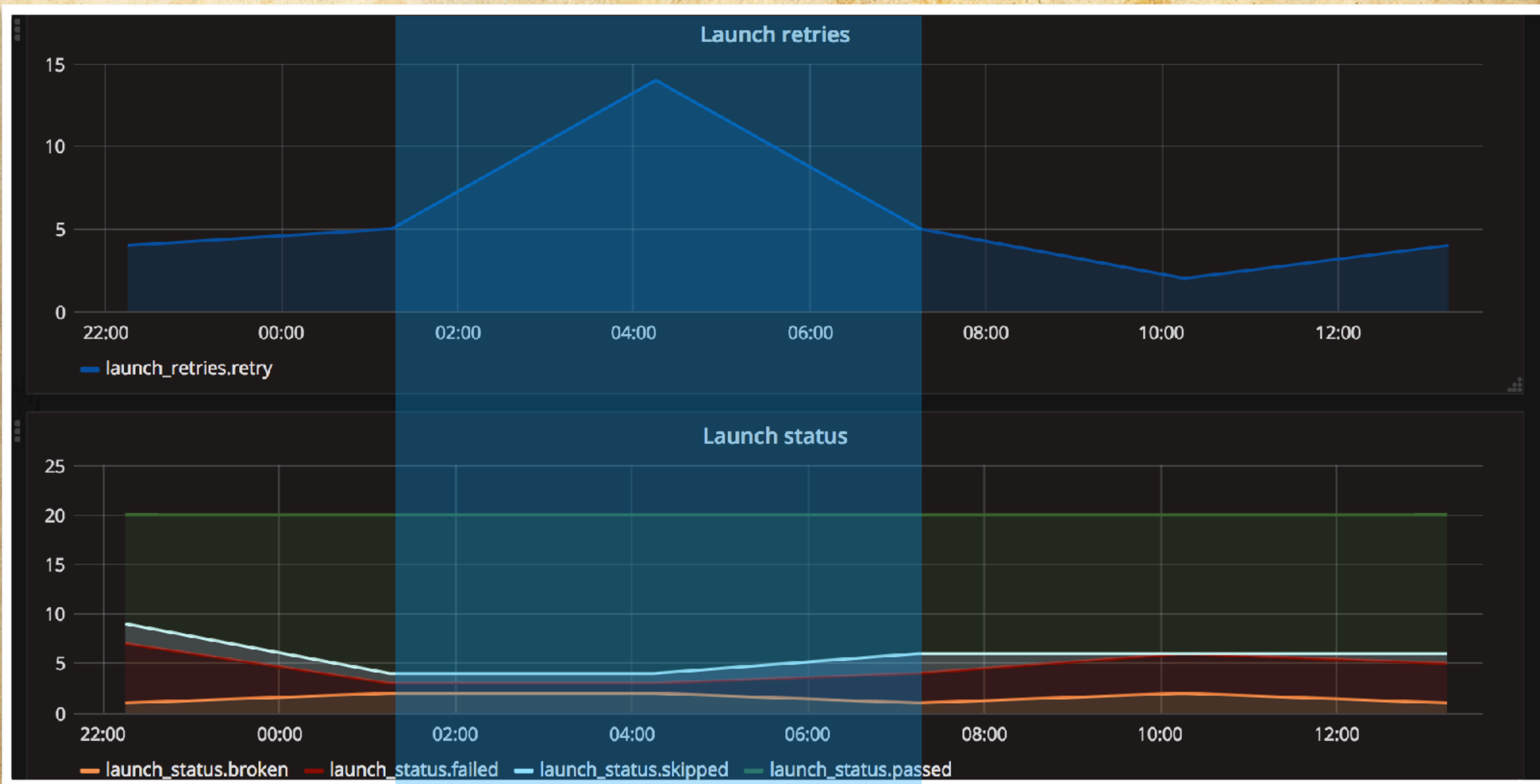
Launch retries



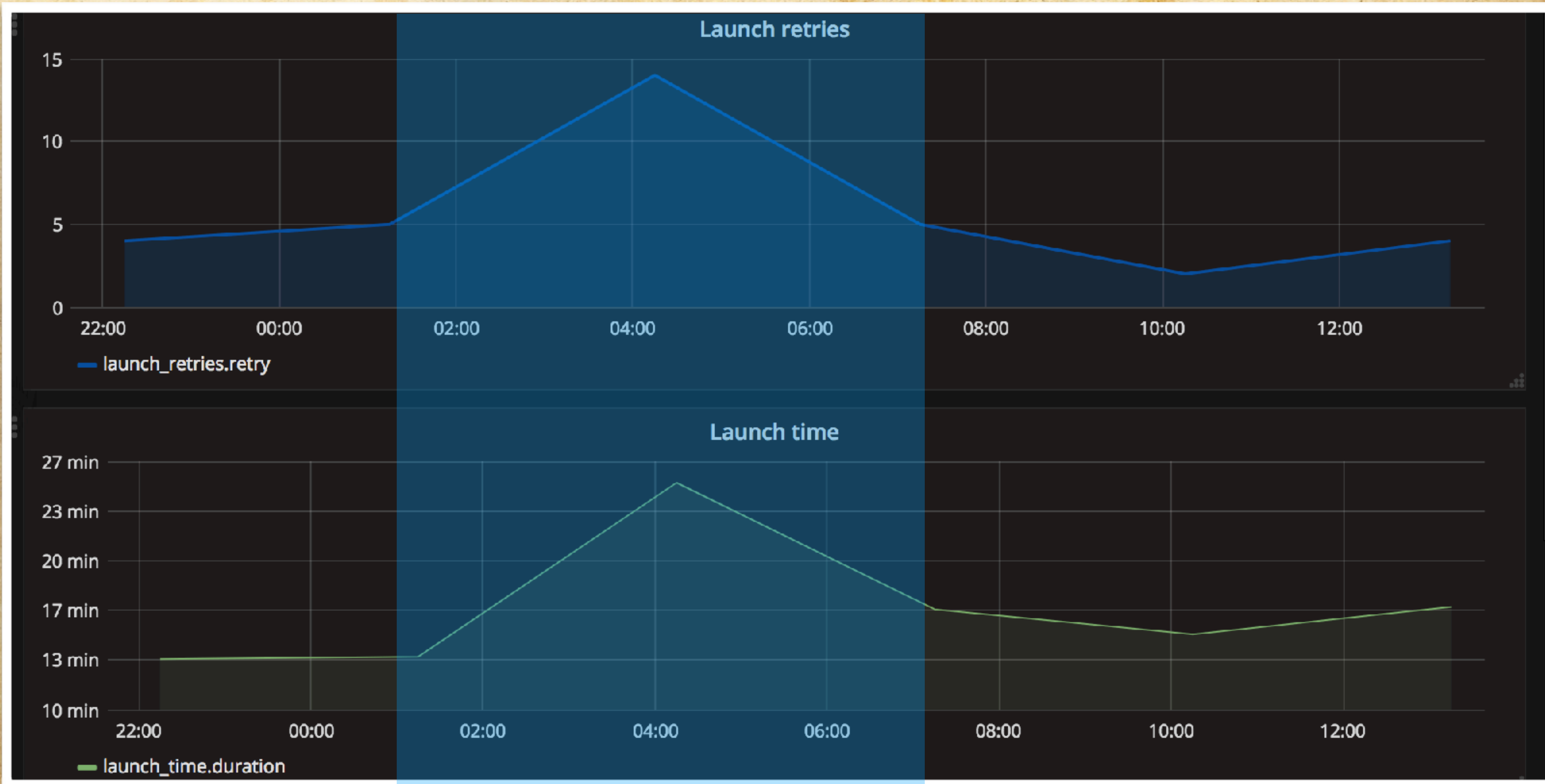
Количество перезапусков



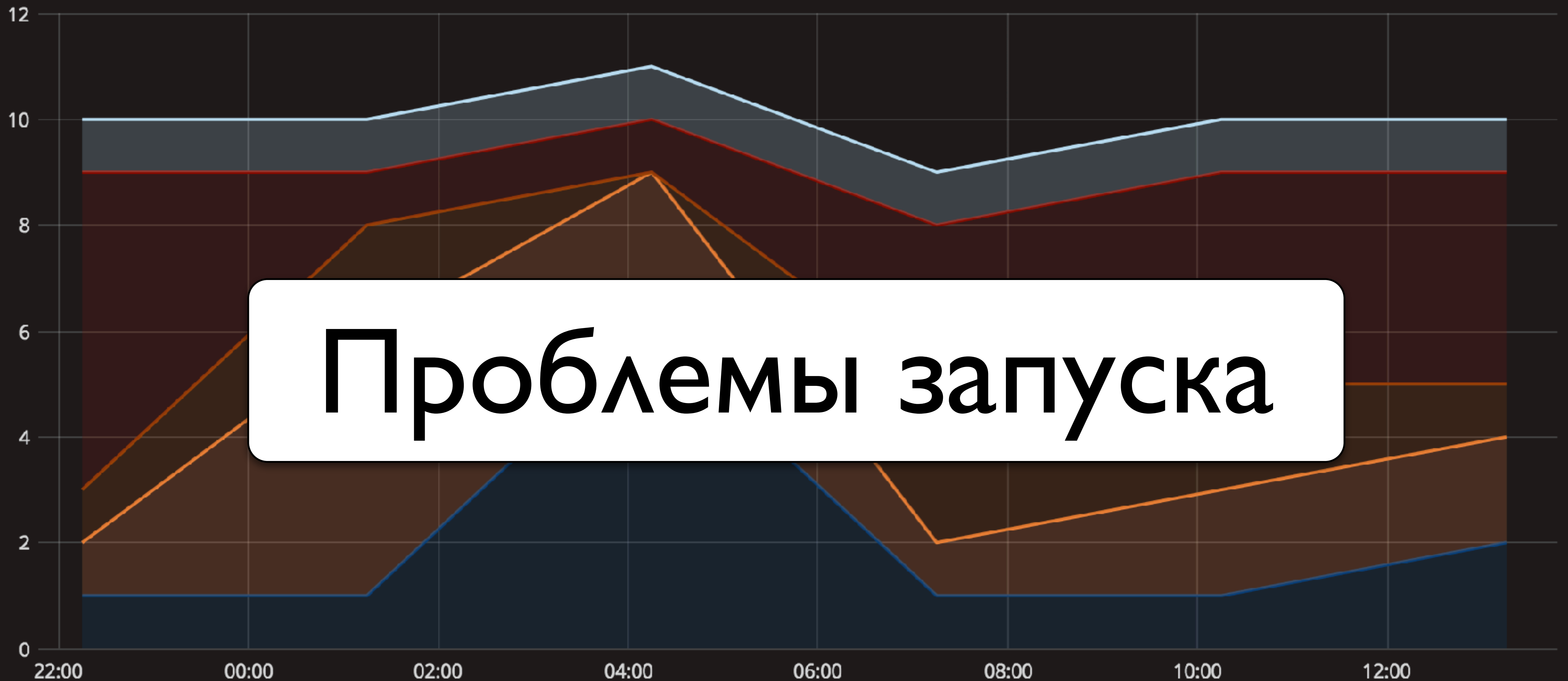
Перезапуски и статус



Перезапуски и время



Launch problems



Проблемы запуска

— launch_problems.Infrastructure problems — launch_problems.Outdated tests — launch_problems.Test defects — launch_problems.Product defects
— launch_problems.Ignored tests

Статусы теста

skipped

broken

failed

passed

Файл 'categories.json'

```
[
  {
    "name": "Outdated selectors",
    "messageRegex": ".*Unable to locate element.*",
    "matchedStatuses": ["broken"]
  },
  {
    "name": "Selenium problems",
    "messageRegex": ".*Timed out .* from renderer.*",
    "matchedStatuses": ["broken"]
  }
]
```

Категории проблем

Categories

order ▾ name ▾ duration ▾ status ▾

Filter by status: 2 5 0 0 0

- Configuration problems 1
 - FileNotFoundException: Configuration problem: config.xml is missing 1
- Outdated selectors 3
 - NoSuchElementException: no such element: Unable to locate element: {"method":"xpath","selector":"//*[contains(@class,'card__footer')]"} 1
 - NoSuchElementException: no such element: Unable to locate element: {"method":"xpath","selector":"//*[contains(@class,'card__header')]"} 1
 - NoSuchElementException: no such element: Unable to locate element: {"method":"xpath","selector":"//*[contains(@class,'wall__title')]"} 1
- Product defects 2
 - AssertionError: Expected: <2147483653L> but: was <-2147483643L> 1
 - AssertionError: разница в пикселях между скриншотом тестинга и продакшена Expected: a value less than <5> but: <47365> was greater than <5> 1
- Selenium problems 1
 - TimeoutException: timeout: cannot determine loading status from timeout: Timed out receiving message from renderer: -0.029 1

my.company.Offers.offersListValidation

Broken Offers list validation

Overview History Retries

FileNotFoundException: Configuration problem: config.xml is missing

Categories: Configuration problems

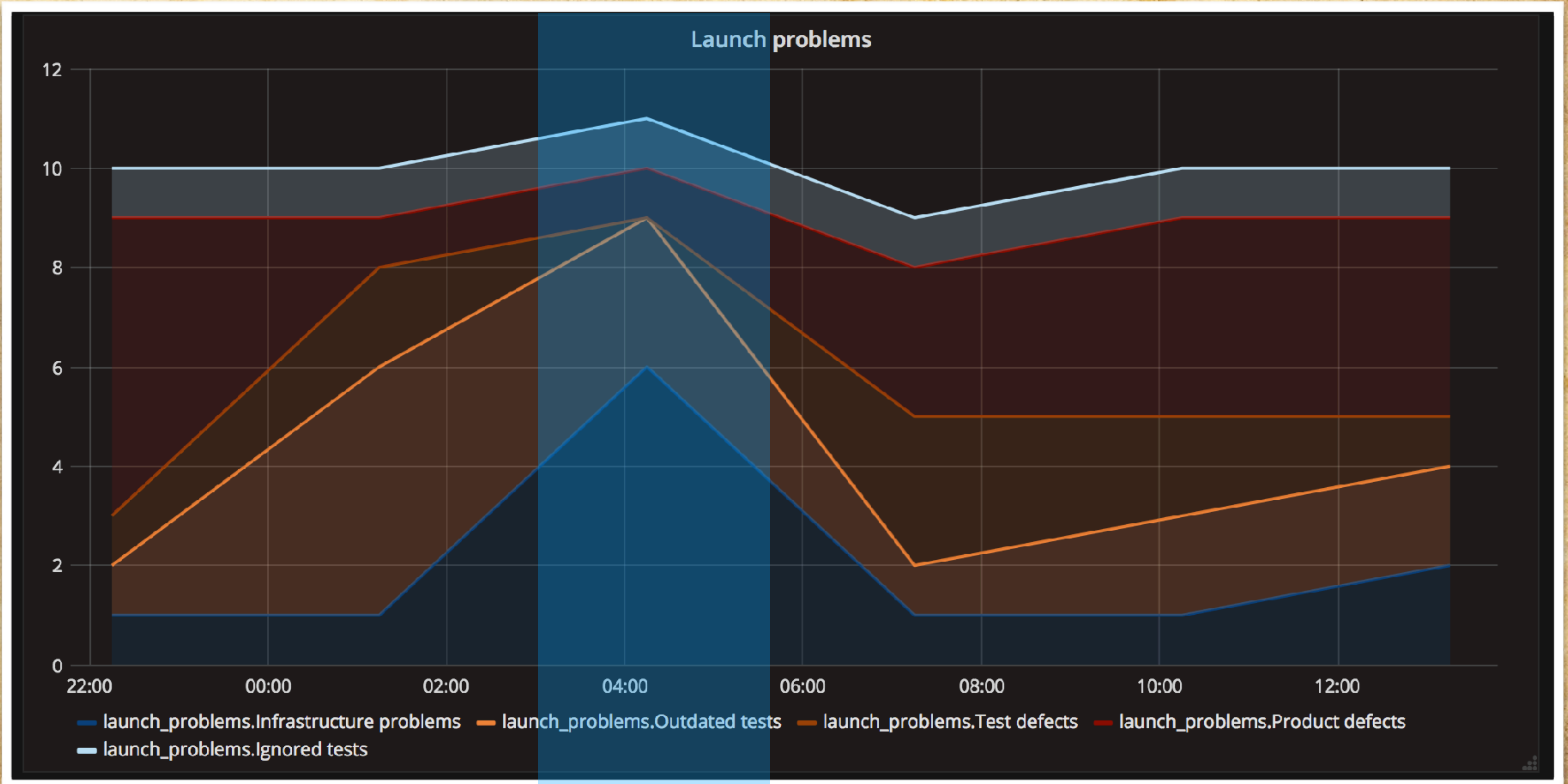
Severity: normal

Duration: ⌚ 2ms

Execution

No information about test execution is available.

Проблемы запуска

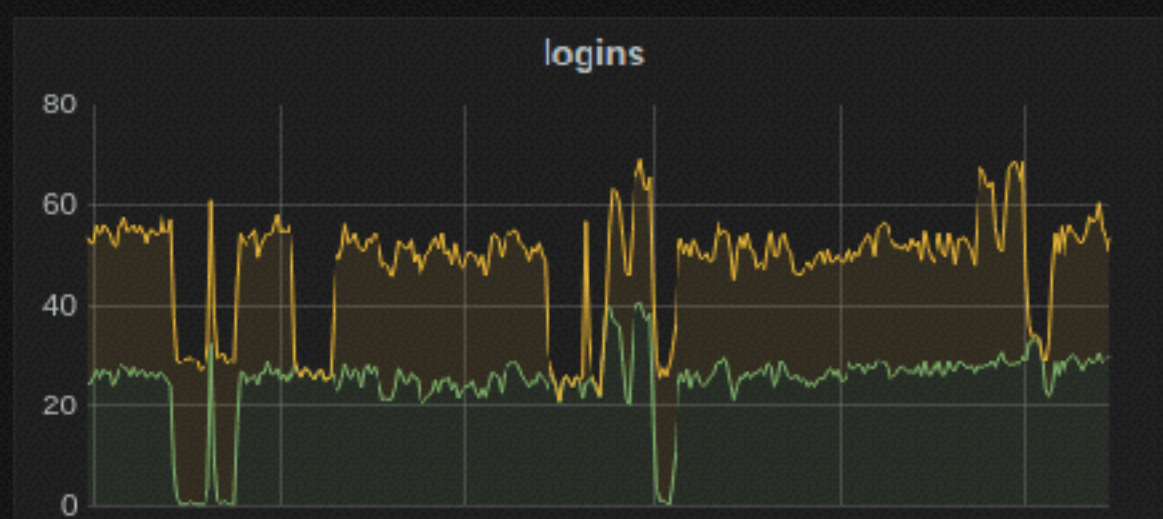
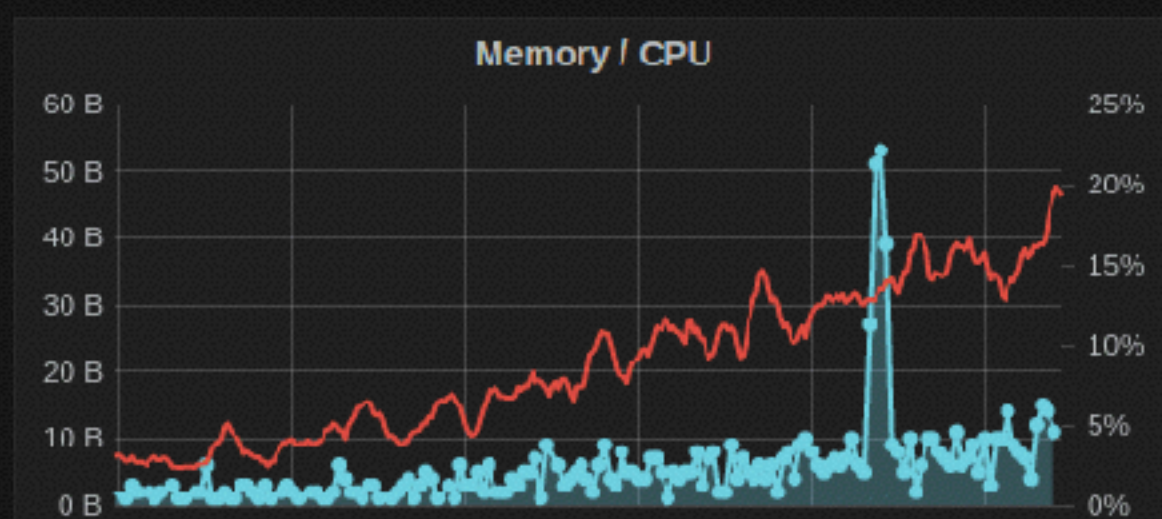
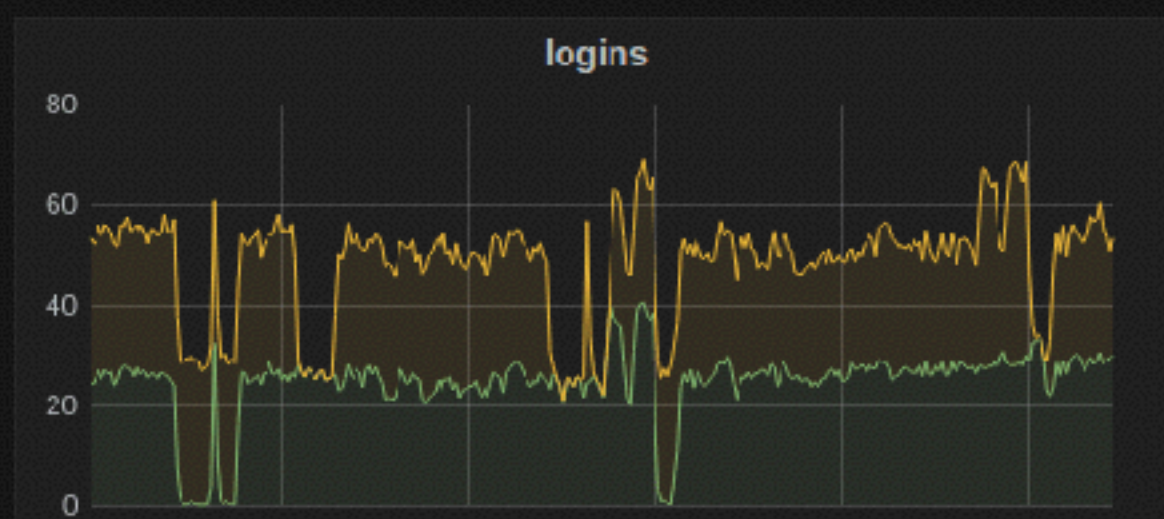
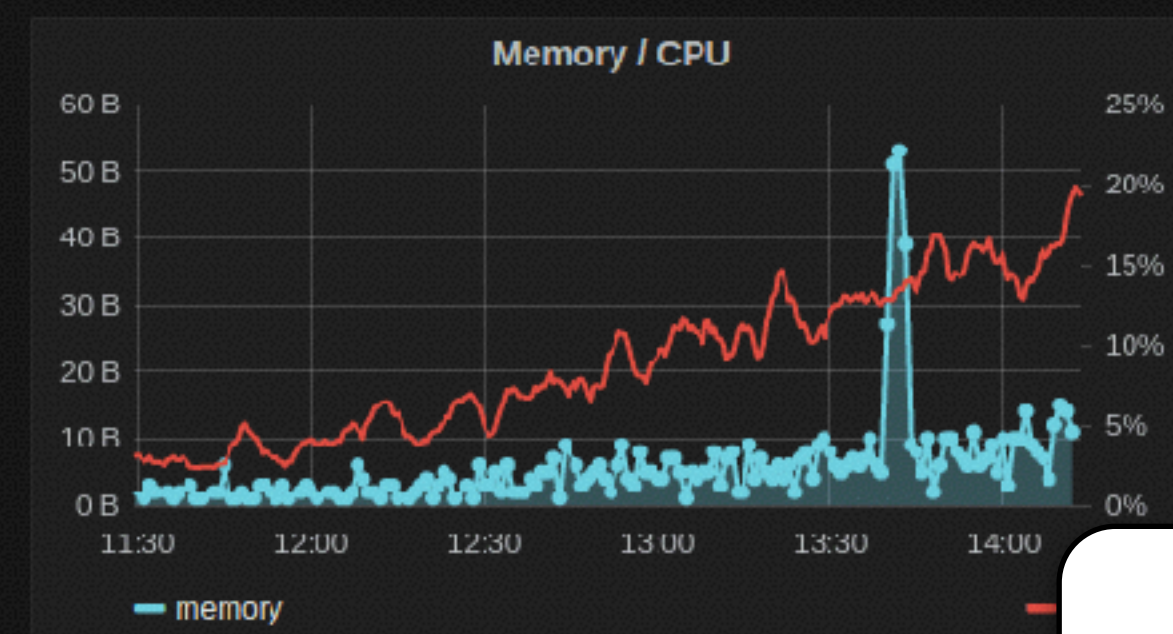
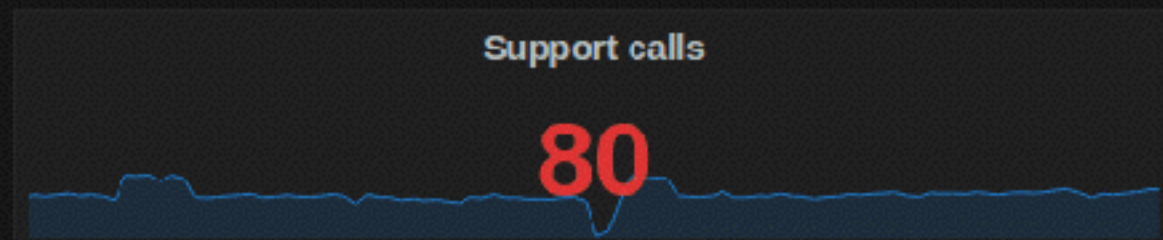




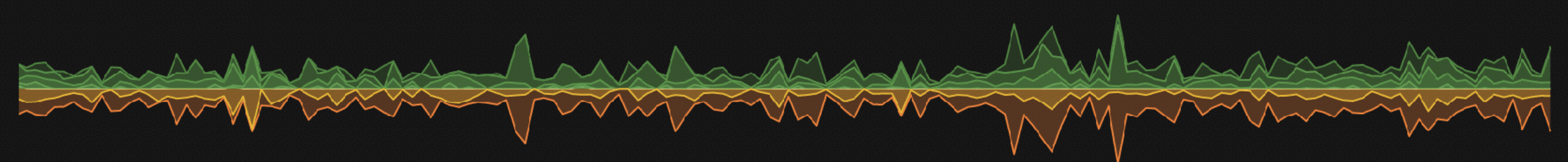
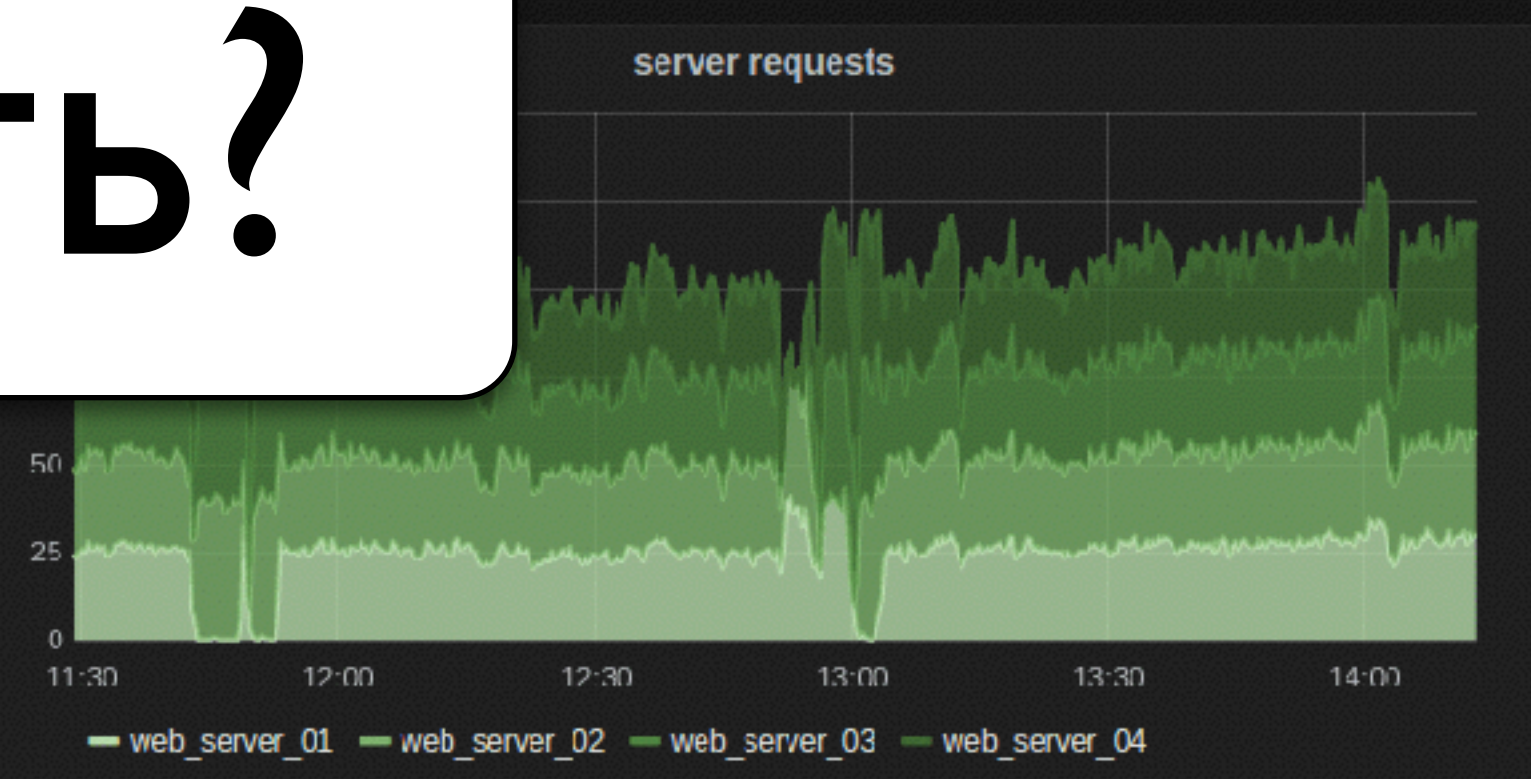
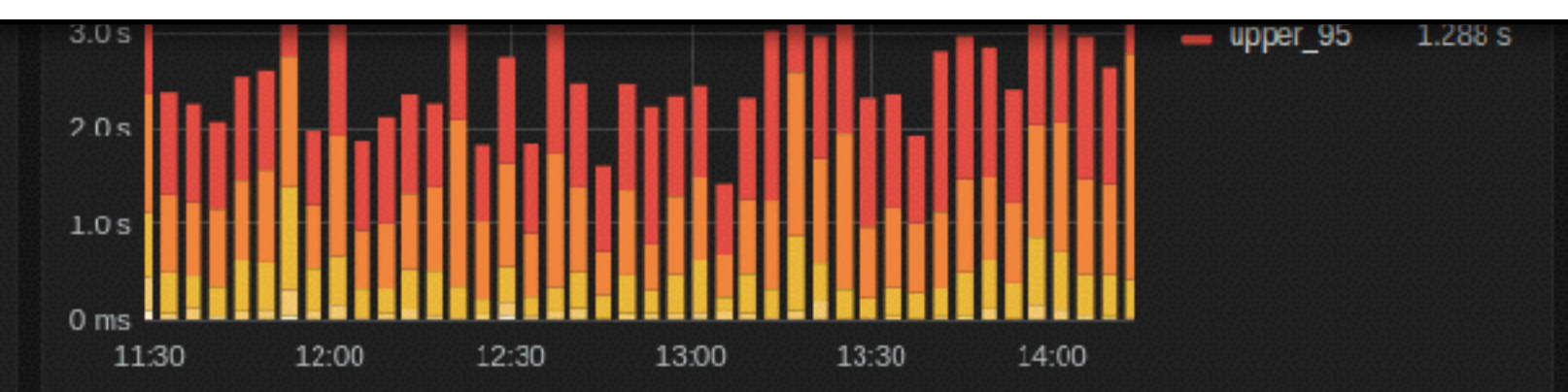
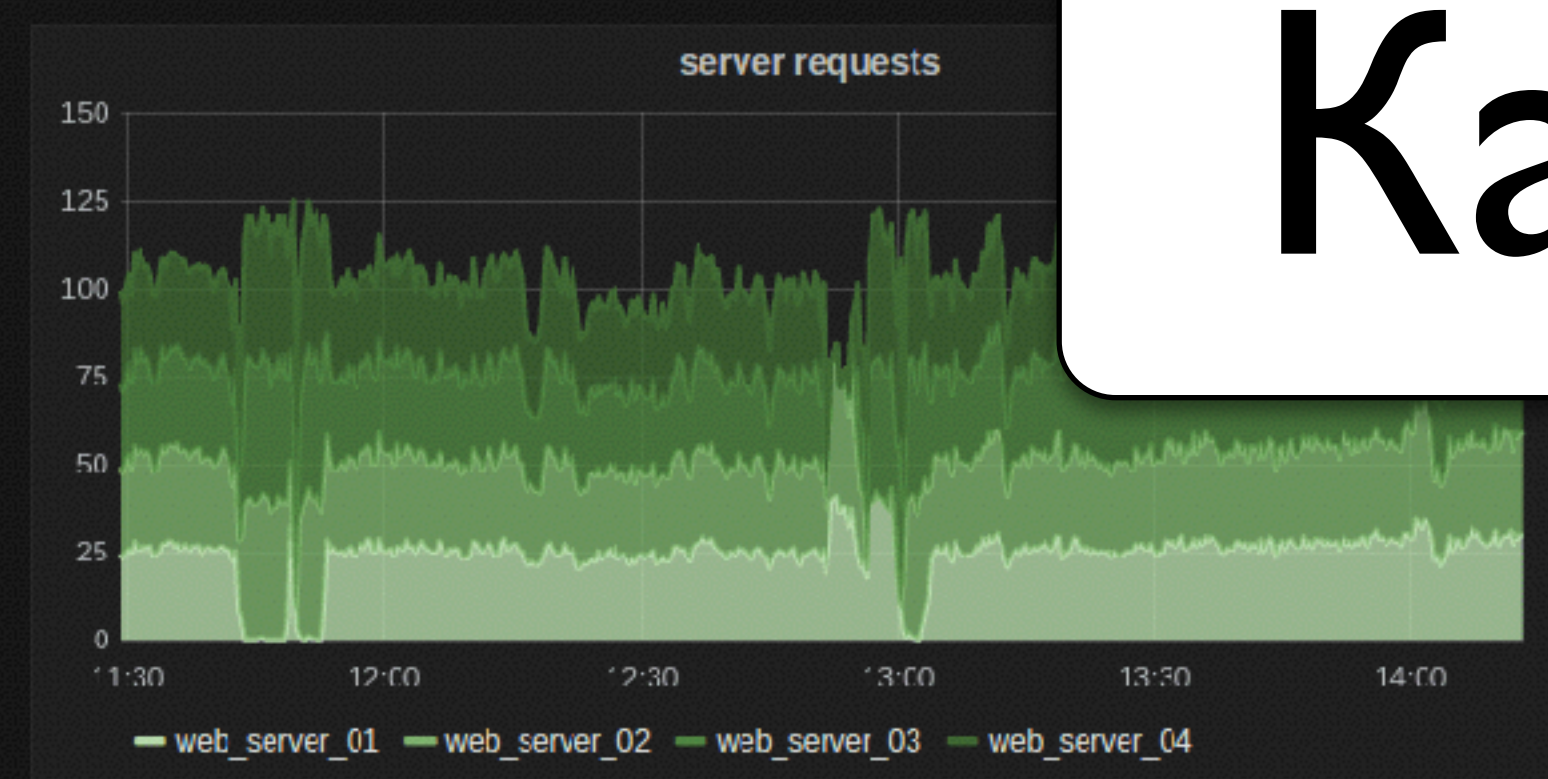
Big Dashboard



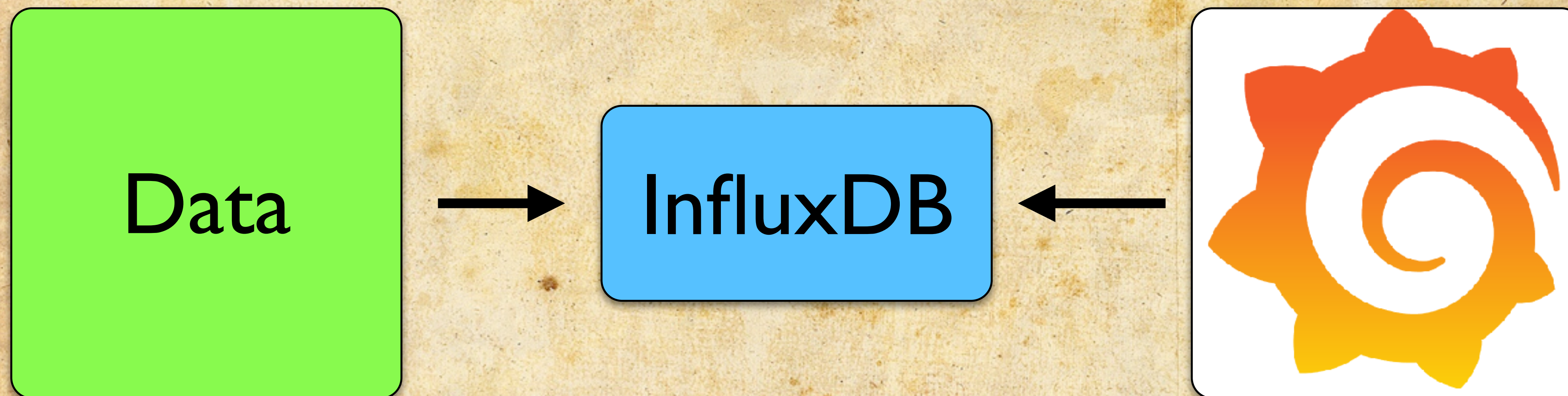
Zoom Out Oct 4, 2015 11:29:09 to Oct 4, 2015 14:13:23 UTC



Как настроить?



Как это работает?



Экспорт в InfluxDB



```
curl -XPOST 'http://influx/write?db=auto' --data-binary @data.txt
```

Формат данных InfluxDB

launch_status failed=10

launch_status broken=5

launch_status skipped=5

launch_status passed=100

launch_time duration=813839

launch_retries retry=15

launch_problems selenium=7

launch_problems outdated=4

Формат данных InfluxDB

```
launch_status failed=10  
launch_status broken=5  
launch_status skipped=5  
launch_status passed=100
```

```
launch_time duration=813839
```

```
launch_retries retry=15
```

```
launch_problems selenium=7  
launch_problems outdated=4
```

Формат данных InfluxDB

launch_status failed=10

launch_status broken=5

launch_status skipped=5

launch_status passed=100

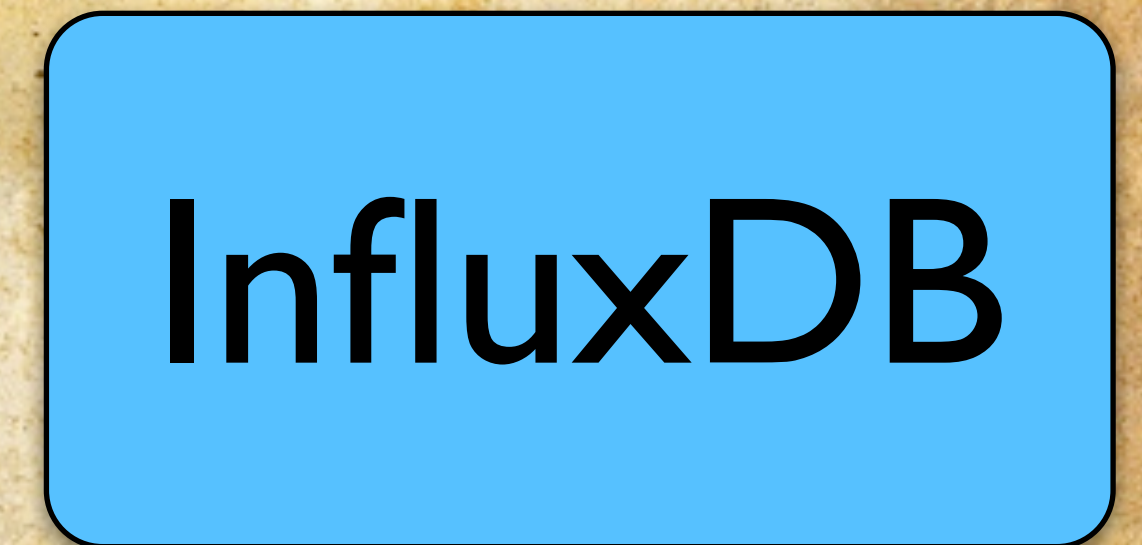
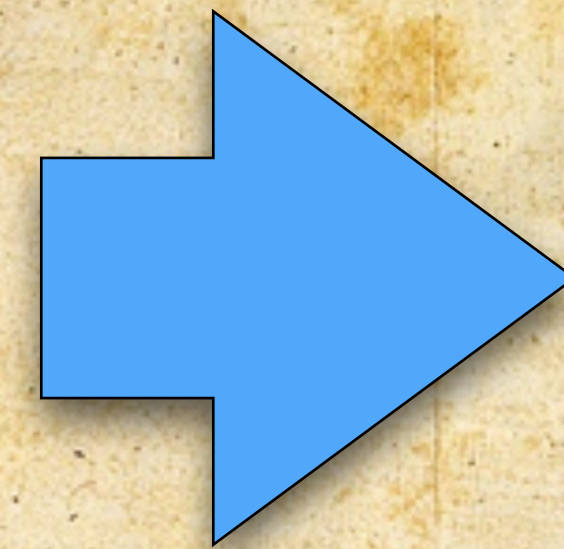
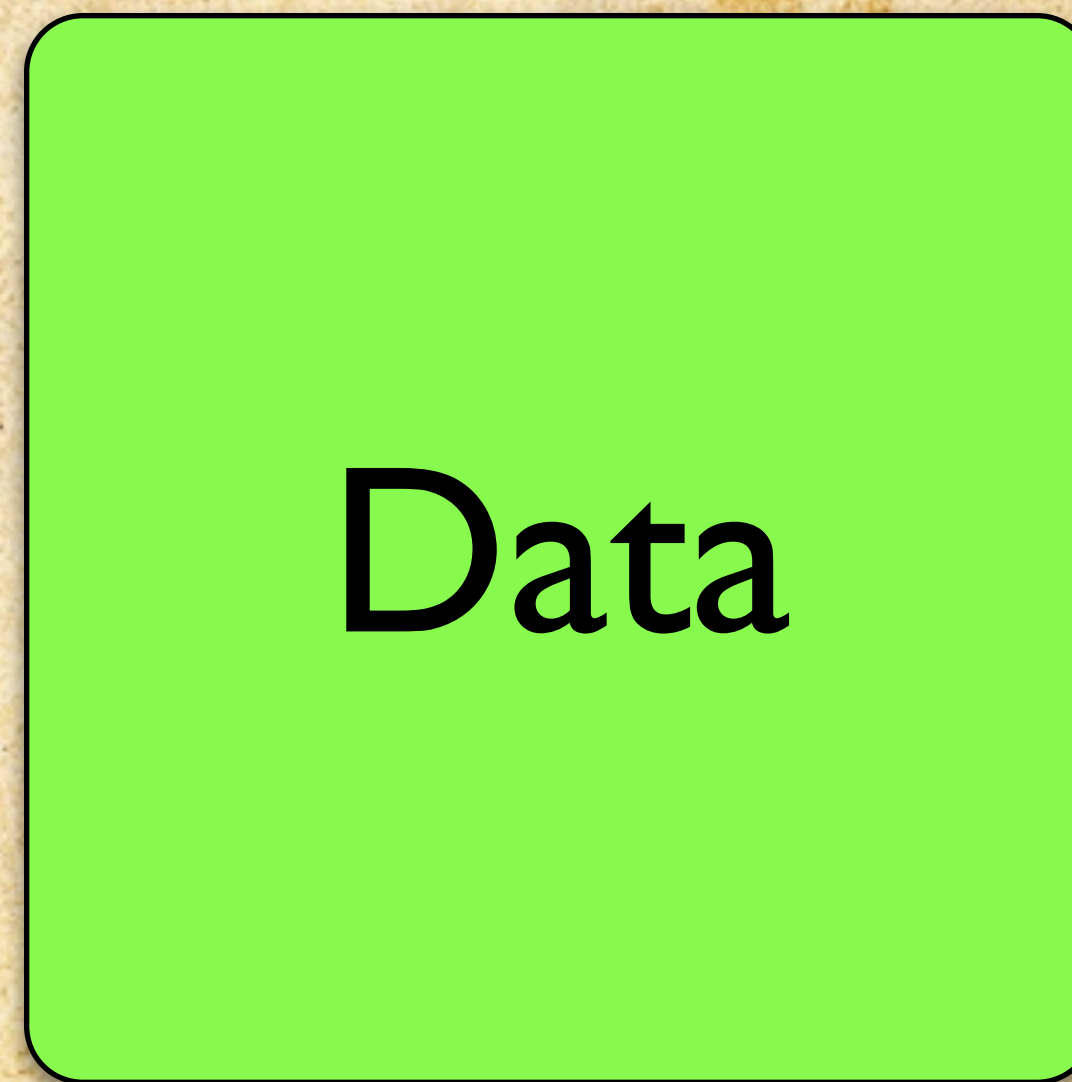
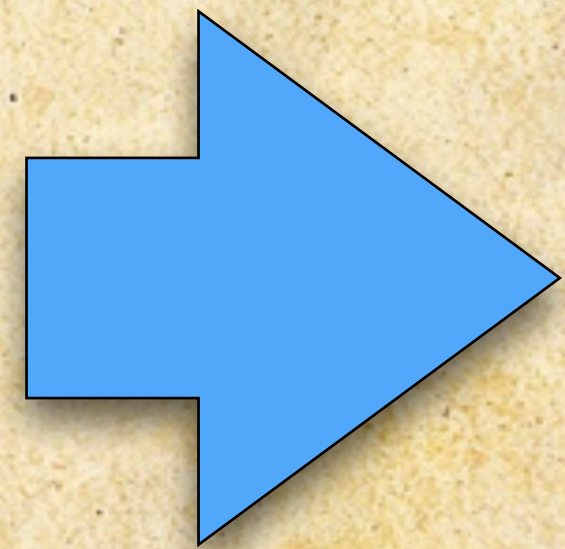
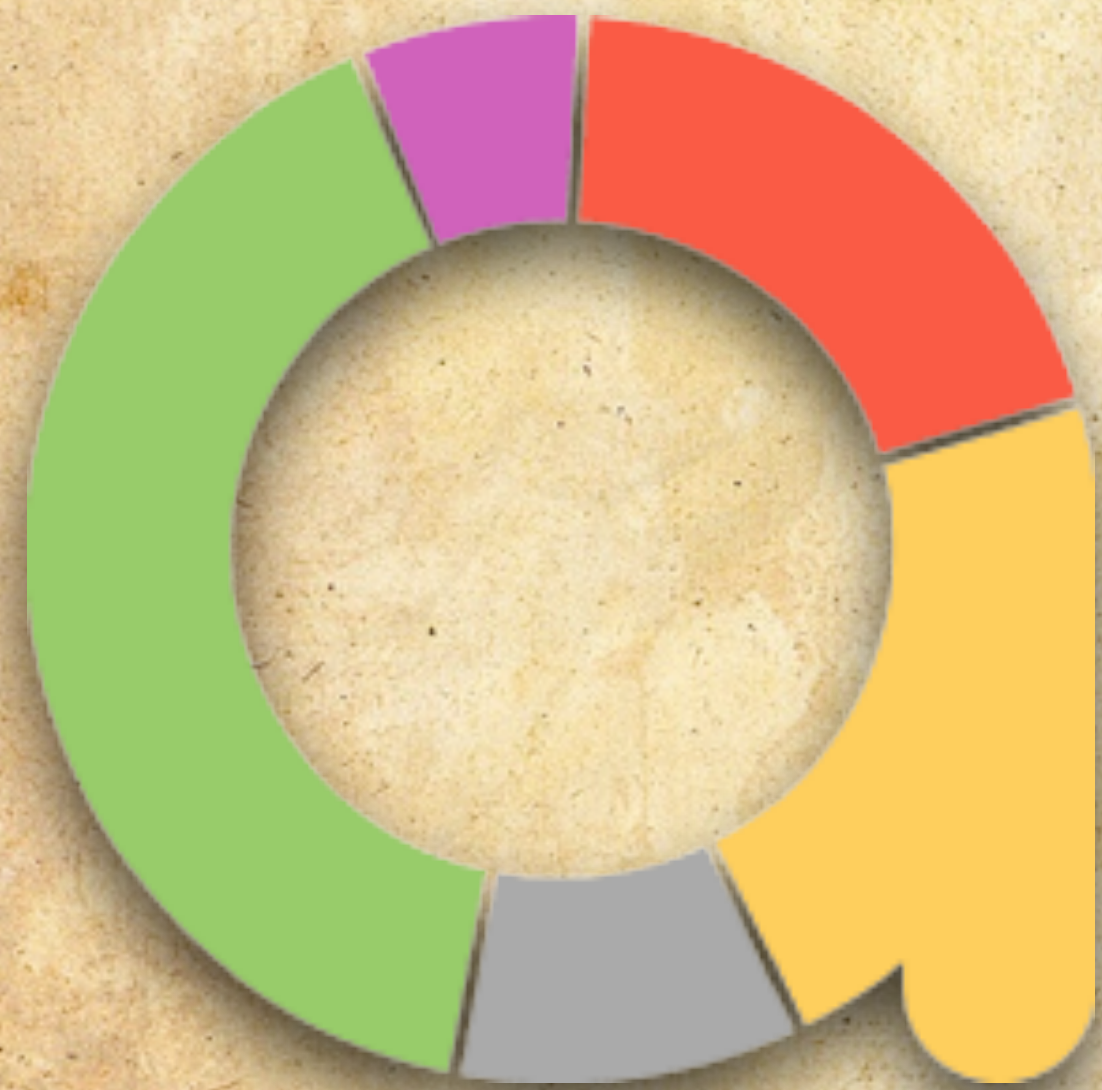
launch_time duration=813839

launch_retries retry=15

launch_problems selenium=7

launch_problems outdated=4

Allure Plugin



```
curl -XPOST 'http://influx/write?db=auto' --data-binary @data.txt
```




Частые вопросы

A wooden table with a spiral-bound notebook and a card. The card has a diagram with red arrows. Three white rounded rectangles with black borders are overlaid on the image, containing the text 'Простота', 'Доверие', and 'Контроль' from top to bottom.

Простота

Доверие

КОНТРОЛЬ



@QametaSoftware

@eroshenkoam

Вопросы?