

# Zero Error Policy

Узнаём про ошибки  
быстрее пользователей

Максим Земсков

Ведущий разработчик интерфейсов

# Внедряем МОНИТОРИНГ ОШИБОК

- ★ Выбираем систему мониторинга  
Sentry, LogRocket, Bugsnag, etc
- ⚙ Подключаем библиотеку в проект  
Готовый клиент к выбранной  
системе мониторинга
- 🕒 Настраиваем уведомления  
В почту, чаты или телефон

```
<!doctype html>
<html>
  <head>
    <title>Example</title>
  </head>
  <body>
    <h1>hello error</h1>
    <script src="https://js.sentry-cdn.com/browser.min.js" ></script>
  </body>
</html>
```

# Сразу после релиза

SP

## Issues ?

Unresolved 2024 For Review 1766 Regressed Escalating 574 Archived

javascript-react

All Envs

24H

Custom Search

is:unresolved

Resolve  Archive  ...  Last Seen





GRAPH:

24h

EVENTS

USERS

ASSIGNEE

<input type="checkbox"/> <b>TypeError</b> globalThis.fetch(src/sentry/fetch-api) Failed to fetch <b>Escalating</b> JAVASCRIPT-REACT-4M <b>Unhandled</b> 16hr ago   19hr old		280K	43K	 <input type="checkbox"/>
<input type="checkbox"/> <b>Error</b> ?(<anonymous>) gtm inline script (timeout) <b>New</b> JAVASCRIPT-REACT-2E <b>Unhandled</b> 16hr ago   6d old		194K	39K	 <input type="checkbox"/>
<input type="checkbox"/> <b>Error</b> HTMLDocument.click_handler(<anonymous>) click error (dynamic) <b>New</b> JAVASCRIPT-REACT-4Z <b>Unhandled</b> 17hr ago   17hr old		119K	31K	 <input type="checkbox"/>
<input type="checkbox"/> <b>TypeError</b> initApp(src/app/main) undefined is not a function <b>Escalating</b> JAVASCRIPT-REACT-4M <b>Unhandled</b> 16hr ago   19hr old		65K	2.1K	 <input type="checkbox"/>

# Смотрим на ошибку

view full trace

## Tags

browser Chrome 123.0.0 browser.name Chrome device Mac device.family Mac environment production handled no level error  
mechanism onunhandledrejection os Mac OS X >=10.15.7 os.name Mac OS X release 1.0.1 url https://super-secret.app/

## Stack Trace

↑↓ Newest ▾ ...

## NetworkError

The network failed or the CORS configuration is incorrect

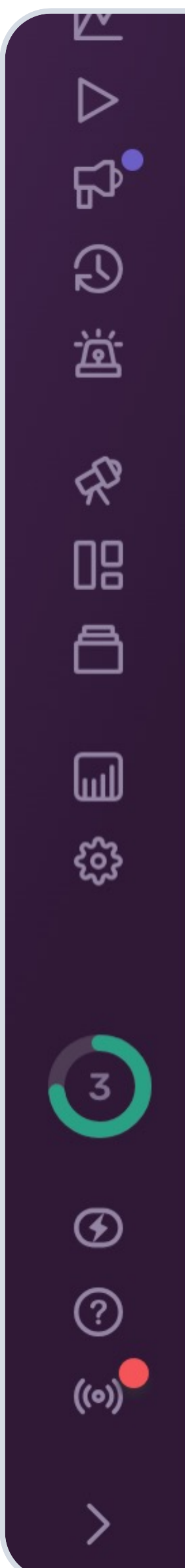
mechanism generic handled false

```
JS fetch-api.ts in globalThis.fetch at line 44:19 In App ▾  
main.tsx in test3 at line 60:29 In App ▲  
58     async function test3() {  
60         const res = await fetch("https://ya.ru");  
61         const data = await res.json();  
63     }
```

## Breadcrumbs

Filter By ▾ Search breadcrumbs ↑↓ Newest ▾

TYPE	CATEGORY	DESCRIPTION	LEVEL	↔ TIME
------	----------	-------------	-------	--------



3

# В результате

Чем заканчивается внедрение мониторинга

- > Игнорируем или отключаем уведомления
- > Игнорируем ошибки
- > Отключаем сбор ошибок из-за высоких расходов или бесполезности



## Что будет в докладе

- 01 Концепция Zero Error Policy
- 02 Устранение шума в потоке ошибок
- 03 Детализация ошибок
- 04 Полнота логирования
- 05 Конфигурация уведомлений
- 06 Итоги

# Максим Земсков

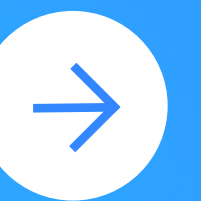
Ведущий разработчик  
интерфейсов

- > Делаю frontend с 2009 года
- > Неоднократно внедрял мониторинг ошибок в крупные проекты
- > Собрал грабли и готов делиться опытом



спикер

# Что такое Zero Error Policy





# Правила Zero Error Policy

- Все ошибки должны попадать в систему мониторинга
- Все ошибки должны быть проанализированы
- Любая программная ошибка должна быть устранена

# Основа для Zero Error Policy

Для Zero Error Policy нужно  
создать условия в проекте



# Устраняем шум в потоке ошибок

2



# Основные источники шума



## Операционные ошибки

Сетевые ошибки,  
ошибки доступа к API



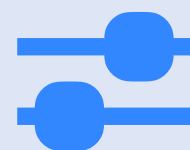
## False-positive ошибки

Лишнее логирование,  
логика на исключениях



## Инъекции 3rd-party кода

Вирусы и антивирусы,  
google tag manager



## Браузерные расширения

Выполнение кода  
в контексте страницы



## Устаревшие браузеры

Ошибки синтаксиса,  
отсутствие API



## Ошибки старых релизов

Хвост старых версий  
после исправления ошибки

# Программные ошибки

## Неожиданное поведение

Не должны происходить,  
когда все идет по плану

- Получение свойства у null или undefined
- Некорректные параметры функции
- Получение данных от бекенда в неожиданном формате

# Операционные ошибки

## Обработанные ошибки

Происходят при нормальном  
выполнении программы

- Отсутствие сети, недоступность API
- Отсутствие прав доступа
- Некорректный пользовательский ввод

# Программные ошибки

level = error

```
console.error(err)
```

```
Sentry.captureException(err, {  
  level: 'error'  
})
```

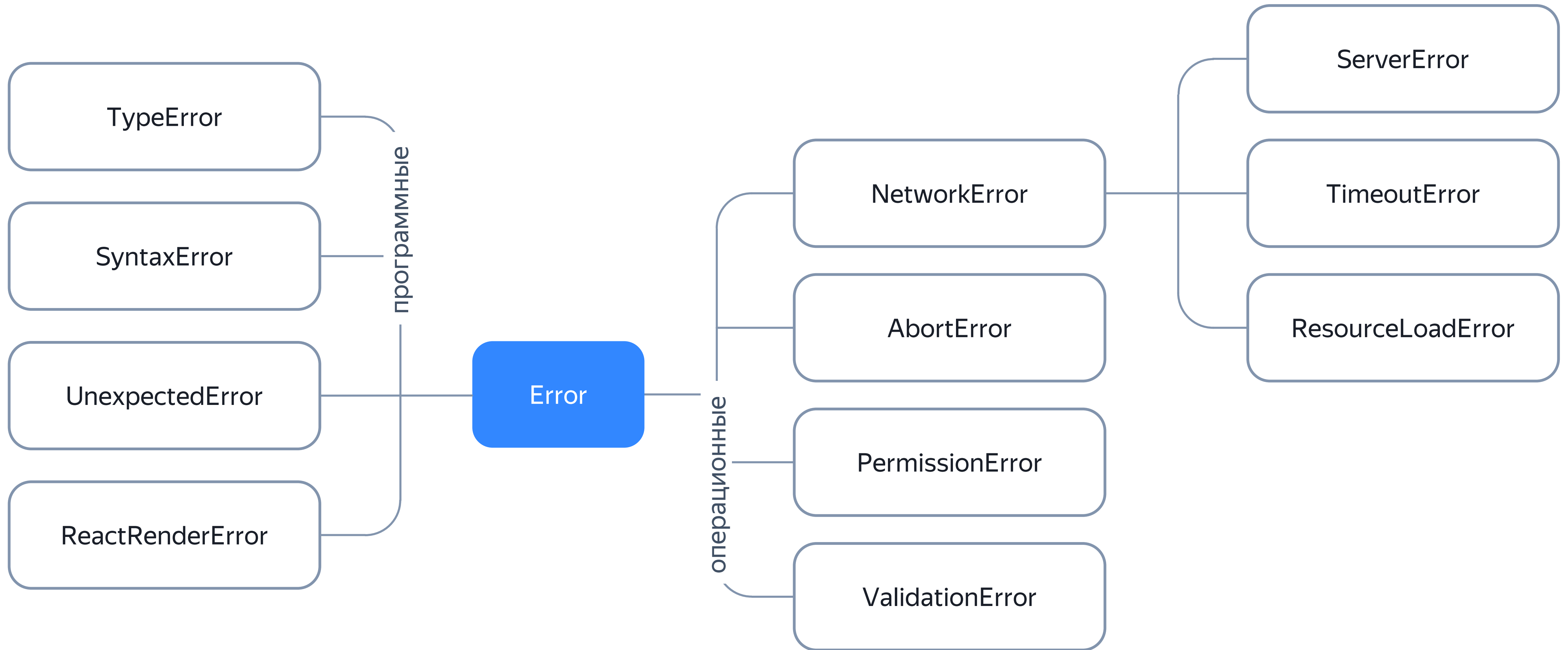
# Операционные ошибки

level = warning

```
console.warn(err)
```

```
Sentry.captureException(err, {  
  level: 'warning'  
})
```

# Примеры классов для ошибок



## Пример логирования в Sentry

```
async function getCoordinates() {  
    // ...  
    throw new PermissionError();  
}  
  
try {  
    await getCoordinates();  
} catch(err: unknown) {  
    // Logs error  
    Sentry.captureException(error);  
}
```



## Пример логирования в Sentry

```
● ● ●  
  
Sentry.addEventProcessor((event, hint) => {  
  const error = hint.originalException;  
  
  for (const cause of iterateErrorCause(error)) {  
    if (cause instanceof PermissionError) {  
      event.level = "warning";  
      break;  
    }  
  }  
  
  return event;  
});
```

# Ошибки загрузки кода

Lazy load, js и css файлы

## ✘ Бандлер создаёт обычный Error

Потому что ничего не знает про нашу разметку ошибок

## ✔ Заменяем на ResourceLoadError

Логируем как операционную ошибку

```
Error: Loading chunk 2177 failed.  
(https://my.app/static/chunk.d932fbfc71592a154d83.  
  at wf (react.production.min.js:152:178)  
  at f.componentDidCatch.c.callback (react.produ  
  at Fh (react.production.min.js:123:7)  
  at Ki (react.production.min.js:211:186)  
  at Ii (react.production.min.js:210:278)  
  at Ii (react.production.min.js:210:175)  
  at Ik (react.production.min.js:209:329)  
  at Sk (react.production.min.js:228:312)
```

## Ошибки загрузки кода

```
Sentry.addEventProcessor(async (event, hint) => {
  const error = hint.originalException;
  const chunkLoadRegExp = /Error: Loading (css)?chunk/i;

  if (chunkLoadRegExp.test(error.message)) {
    const url = extractChunkUrl(error);

    if (await isNetworkError(url)) {
      event.level = "warning";
    } else {
      Sentry.captureException(new ResourceLoadError(url));
    }
  }

  return event;
});
```

# Ошибки загрузки данных

Fetch API, XMLHttpRequest, WebSocket

- ✘ Браузер создаёт `TypeError`  
Сообщение ошибки зависит от браузера  
Теряется стектрейс в Firefox и Safari
- ✔ Заменяем на `NetworkError`  
Делаем обертку над Fetch API  
Восстанавливаем стектрейс

```
[chrome]
TypeError: Failed to fetch
  at test3(main.tsx:60:35)
  at test1_immediate(main.tsx:64:13)
```

```
[safari]
TypeError: Load failed
  <empty stacktrace>
```

```
[firefox]
TypeError: NetworkError when attempting to fetch
  <empty stacktrace>
```

## Ошибки загрузки данных

```
function patchFetchApi() {  
  const { fetch: origFetch, Request } = globalThis;  
  globalThis.fetch = async function (this: unknown, ...args) {  
    // Validates the args via Request constructor  
    const request = new Request(...args);  
    const stack = new Error().stack;  
  
    try {  
      return await origFetch.call(this, request);  
    } catch (err) {  
      // Fixes empty stack  
      err.stack = stack;  
  
      if (err.name === "AbortError") throw err;  
      throw new NetworkError(request, { cause: err });  
    }  
  };  
}
```

# Браузерные расширения

Web extensions, user scripts

## ✘ Ошибки из файлов расширений

Инъекция в страницу кода, который мы не контролируем

## ✔ Фильтруем по стектрейсу

Логировать такие ошибки бессмысленно

```
Error: Clover: Disconnected from Clover back
    at CloverInpageProvider._handleDisconnec
    at CloverInpageProvider._handleStreamDis
    at f (chrome-extension://nhnkbkgjkgciga
    at a.emit (chrome-extension://nhnkbkgjik
    at E (chrome-extension://nhnkbkgjkgciga
```

# Фильтрация браузерных расширений

The screenshot shows the Sentry web interface. On the left is a dark sidebar with navigation icons and menu items. The main area is dark grey and displays a 'FILTERS' section. A white tooltip is open over the 'Inbound Filters' menu item, and another white tooltip is open over the 'Filter out errors known to be caused by browser extensions' toggle, which is currently turned on.

Tags

Environments

Ownership Rules

Data Forwarding

**PROCESSING**

Inbound Filters

Security & Privacy

Issue Grouping

Processing Issues

Debug Files

ProGuard

Source Maps

Performance

Metrics

Replays

User Feedback

SDK SETUP

Client Keys (DSN)

10


0


**FILTERS**


**Filter out errors known to be caused by browser extensions**  
Certain browser extensions will inject inline scripts and are known to cause errors.


**Filter out health check transactions**  
Filter transactions that match most common naming patterns for health checks.

**Filter out legacy browsers: All None**  
The browser versions filtered out will be periodically evaluated and updated.

 **Chrome**  
Version 62 and lower

 **Safari**  
Version 11 and lower

 **Firefox**  
Version 66 and lower

 **Android**  
Version 3 and lower

## Фильтрация браузерных расширений

```
function isBrowserExtension(err: Error): boolean {  
  const lines = err.stack.split('\n');  
  for (const line of lines) {  
    const hasSourceFile = line.includes('/://');  
    if (hasSourceFile) {  
      return /^(?:moz|chrome|safari)-extension:\\/\\/\\.test(line);  
    }  
  }  
  return false;  
}
```



# Браузерные расширения

Ошибки без понятного стектрейса

- ✘ **Ошибки с пустым стеком**
  - При использовании `eval`
  - При логировании строк, вместо `Error`
- ✔ **Фильтруем по сообщению**
  - Достаточно списка из 15-20 регулярных выражений
  - Иногда нужно добавлять новые

```
TypeError: Cannot redefine property: ethereum
    at Function.defineProperty (<anonymous>)
    at <anonymous>:2:8
```

```

const filters = [
  /fb_xd_fragment/,
  /Cannot redefine property:
(solana|ethereum|googletag)/,
  // ... 18 more rules ...
];

function isBrowserExtension(err: Error): boolean {
  return filters.some(regExp => {
    return regExp.test(err.message);
  });
}

```

relay / relay-filter / src / browser\_extensions.rs

Code

Blame

301 lines (275 loc) · 11.3 KB · 

```

9     static EXTENSION_EXC_VALUES: Lazy<Regex> = Lazy::new(|| {
10         Regex::new(
11             r#"(?ix)
12             # Random plugins/extensions
13             top\.GLOBALS|
14             # See: http://blog.errorception.com/2012/03/tale-of-unfindable-js-e
15             originalCreateNotification|
16             canvas.contentDocument|
17             MyApp_RemoveAllHighlights|
18             http://tt\.epicplay\.com|
19             Can't\sfind\svariable:\sZiteReader|
20             jigsaw\s\s\snot\sdefined|
21             ComboSearch\s\s\snot\sdefined|
22             http://loading\.retry\.widdit\.com/|
23             atomicFindClose|
24             # Facebook borked
25             fb_xd_fragment|
26             # ISP "optimizing" proxy - `Cache-Control: no-transform` seems to
27             # reduce this. (thanks @acdha)
28             # See http://stackoverflow.com/questions/4113268
29             bmi_SafeAddOnload|
30             EBCallbackMessageReceived|
31             # See https://groups.google.com/a/chromium.org/forum/#!topic/chromi
32             _gCrWeb|
33             # See http://toolbar.conduit.com/Developer/HtmlAndGadget/Methods/
34             conduitPage|
35             # Google Search app (iOS)
36             # See: https://github.com/getsentry/raven-js/issues/756
37             null\s\s\snot\s\s\sobject\s\s(evaluating\s'elt.parentNode'\s)|
38             # Dragon Web Extension from Nuance Communications          26
39             # See: https://forum.sentry.io/t/error-in-raven-js-plugin-setsuspen
40             plugin\.setSuspendState\s\s\snot\s\s\sfunction|

```

# Устаревшие браузеры

## ✘ Ошибки парсинга или доступа к API

Приходят из браузеров, которые проект осознанно не поддерживает

## ✔ Фильтруем устаревшие браузеры

Проверяем по browserslist и наличие нужных API

```
$> cat package.json
{
  "name": "my-app",
  "version": "0.0.0",
  "private": true,
  "dependencies": {

  },
  "browserslist": [
    "> 0.5% and not dead",
    "Firefox ESR",
    "ios_saf>=15"
  ]
}
```

## browserslist-useragent-regexp



```
// Generated by the following command:  
// $> npx browserslist-useragent-regexp --allowHigherVersions  
const regExp = /Edge?\/(10[5-9]|1[1-9]\d|[2-9]\d\d|\d{4,}).../;  
  
function isBrowserSupported(): boolean {  
  return regExp.test(window.navigator.userAgent);  
}
```

## Фильтруем устаревшие браузеры

```
const isSupported =  
  // Check User-Agent  
  isBrowserSupported() &&  
  // Additional Feature Detection  
  Boolean(window.IntersectionObserver);  
  
Sentry.addEventProcessor(event => {  
  if (!isSupported) {  
    event.level = "warning";  
  }  
  return event;  
});
```

Делаем ошибки  
ПОНЯТНЫМИ



# Что есть в понятной ошибке



## Stacktrace

Указывает на исходные файлы в проекте



## Осмысленное сообщение

Помогает понять, что именно пошло не так



## Дополнительный контекст

Окружение и условия, в которых произошла ошибка

Нет Error → Нет стека



```
// ❌ Incorrect  
throw 'test error';
```

```
// ❌ Incorrect  
throw {  
  message: 'test error'  
};
```

```
// ✅ Correct  
throw new Error('test error');
```



Нет Error → Нет стека

```
● ● ●  
  
// ❌ Incorrect  
Promise.reject();  
  
// ❌ Incorrect  
Promise.reject('test error');  
  
// ❌ Incorrect  
Promise.reject({  
  message: 'test error'  
});  
  
// ✅ Correct  
Promise.reject(new Error('test error'));
```

## ESLint и TypeScript помогают не терять стек

```
● ● ●  
  
// .eslintrc.json  
{  
  "rules": {  
    "@typescript-eslint/no-throw-literal": ["error", {  
      "allowThrowingAny": false,  
      "allowThrowingUnknown": false  
    }],  
    "@typescript-eslint/prefer-promise-reject-errors": "error"  
  }  
}
```

## Потеря контекста в try-catch блоках

```
try {  
  await processPayment(transactionId);  
} catch (err: unknown) {  
  if (isAborted(err)) return;  
  
  // ESLint: Expected an error object to be thrown.  
  throw err;  
}
```

```
NetworkError: Failed to fetch  
  at api.ts:98:1  
  at payment.ts:47:1  
  at index.ts:18:1  
  at index.html:12:25
```

## Используем Error.cause

```
try {  
  await processPayment(transactionId);  
} catch (err) {  
  if (isAbortError(err)) return;  
  
  throw new Error(`Payment failed. Transaction ID:  
${transactionId}`, {  
    // More context with cause  
    cause: err  
  });  
}
```

```
Error: Payment failed. Transaction ID  
  at api.ts:98:1  
  at payment.ts:47:1  
[caused by]  
  NetworkError: Failed to fetch  
    at api.ts:98:1  
    at payment.ts:47:1  
    at index.ts:18:1  
    at index.html:12:25
```

# Script Error

- > Возникает в cross-origin скриптах
- > Маскирует настоящие ошибки
- > Часто фильтруется в системах мониторинга, что плохо

```
<!doctype html>
<html>
  <head>
    <script src="https://browser.sentry-cdn.com/7.1
    <script src="https://api.my-app.dev/user-data.
  </head>
  <body>
    content
  </body>
</html>
```

## Добавляем crossorigin атрибут для скриптов

```
● ● ●  
<!doctype html>  
<html>  
  <head>  
    <script crossorigin="anonymous" src="https://browser.sentry-cdn.com/7.108.0/bundle.min.js">  
    </script>  
    <script crossorigin="use-credentials" src="https://api.my-app.dev/user-data.js">  
    </script>  
  </head>  
  <body>  
    content  
  </body>  
</html>
```

## Мониторинг crossorigin в скриптах

```
function assertScriptCrossorigin(el: HTMLElement) {  
  if (el.nodeName !== "SCRIPT") return;  
  if (!el.src) return;  
  if (isSameOrigin(el.src)) return;  
  
  if (!el.crossOrigin) {  
    Sentry.captureException(  
      new Error(`Cross-origin script w/o crossorigin attr: ${el.src}`)  
    );  
  }  
}
```

## Мониторинг crossorigin в скриптах

```
● ● ●  
  
// Already loaded scripts  
document.querySelectorAll("script").forEach(assertScriptCrossorigin);  
  
// Future scripts  
new MutationObserver((mutations) => {  
  for (const mutation of mutations) {  
    for (const node of mutation.addedNodes) {  
      assertScriptCrossorigin(node);  
    }  
  }  
}).observe(document.documentElement, { childList: true, subtree: true });
```



# Source Maps

- > Удобнее анализировать ошибки в мониторинге
- > Лучше работает группировка одинаковых ошибок
- > Легко настраивается в системе сборки

Stack Trace

Most Relevant

## Error

gtm inline script (reject)

mechanism onunhandledrejection handled **false**

JS

<anonymous> in inline\_script\_id\_f3dfc5eefe2136cba81ee46e8d4081ac at line 2:81

```
1 (function inline_script_id_f3dfc5eefe2136cba81ee46e8d4081ac() {
2   '{snip} og("hello from gtm",document.currentScript);Promise.reject(Error("gtm inline script (reject)")
   ("gtm inlin {snip}
3     })());
```

<anonymous> at line 3:11

/gtm.js in f.aa at line 71:229

```
66 {snip} this.l,this.i);this.I(b));f.prototype.D=function(a){this.o.append(a);for(var b,c=[],d,k;(b=this
b?!~b.tagName.t {snip}
67 {snip} {if(!/^noscript$/i.test(g.tagName)){var r=b++;d.push(g.text.replace(/(\/?>)/," data-ps-id="+r+"
style"!==g.g {snip}
68 {snip} e(a.parentNode,"proxyof");d&&this.u[d].appendChild(a)}b.unshift.apply(b,A(a.childNodes))};f.pr
b=this.o.clear();b& {snip}
69 {snip} a.content:b.appendChild(this.i.createTextNode(a.content))};f.prototype.T=function(a){var
b=this.i.createElement(a.tagName);b.setAttribute(" {snip}
70 {snip} finished twice"}):(this.j.shift(),this.write.apply(this,a.ca),!this.j.length&&this.m&&
(this.A(this.m),this.m=null));f.prototype.ja=function {snip}
71 {snip} />');var b=this.i.getElementById("ps-script");b.parentNode.replaceChild(a,b);f.prototype.ga=fun
{a=a.onload=a.onready {snip}
72 {snip} "));return f}();m.postscribe=function(){function e(){var d=b.shift(),k;d&&(k=d[d.length-
1],k.K(),d.stream=f.apply(null,d),k.L())}function f {snip}
73 {snip} :x+" - "+G+": "+H);p.apply(c.l,arguments);c.write(k,function(){t(u,l);c.l.onerror=p;g.done();c
[],c=null;retu {snip}
74 {snip} n=window.google_tag_manager;n&&(n.postscribe||
(n.postscribe=window.postscribe||F));D("google_tag_manager_external.postscribe.getPostscribe {snip}
75 {snip} function(a,b,c){if(a==Array.prototype||a==Object.prototype)return a;a[b]=c.value;return a},ea=f
["object"]==typeof global {snip}
76 {snip} ull!=m&&ca(c,g,{configurable:!0,writable:!0,value:m})),ka=function(a){return a.raw=a},la=funct
a),ma=function(a){var {snip}
41
```

# Source Maps

- > Удобнее анализировать ошибки в мониторинге
- > Лучше работает группировка одинаковых ошибок
- > Легко настраивается в системе сборки

## Stack Trace

### ReferenceError

lines is not defined

mechanism generic handled true

```
JS inline-scripts.ts in <anonymous> at line 102:24
97     return frame;
98   }
99
100   console.log("INLINE FRAME", { frame });
101
102   const maxLines = lines.length;
103   const sourceLine = Math.max(Math.min(maxLines - 1, frame.lineno - 1), 0);
104
105   frame.pre_context = lines
106     .slice(Math.max(0, sourceLine - linesOfContext), sourceLine)
107     .map((line: string) => this.snipLine(line, 0));
```

<anonymous> in Array.map

```
inline-scripts.ts in InlineScriptIntegration.addContextLines at line 95:19
90   }
91   }
92   }
93
94   private addContextLines(frames, scriptElement) {
95     return frames.map((frame) => {
96       if (frame.filename !== "<anonymous>" || !frame.lineno) {
97         return frame;
98       }
99
100       console.log("INLINE FRAME", { frame });
```

inline-scripts.ts in <anonymous> at line 75:34

<anonymous> in Array.forEach

inline-scripts.ts in InlineScriptIntegration.processEvent at line 44:44

## Пример инлайн скрипта

```
● ● ●  
  
<!doctype html>  
<html>  
  <head>  
    <title>Inline script example</title>  
    <script>  
      // e.g. throws TypeError: init is not a function  
      window.myApp.init({ data });  
    </script>  
  </head>  
  <body>  
    <h1>hello error</h1>  
  </body>  
</html>
```

## Пример инлайн скрипта

```

<!doctype html>
<html>
  <head>
    <title>Inline script example</title>
    <script>
      // e.g. throws TypeError: init is not a function
      window.myApp.init({ data });
    </script>
  </head>
  <body>
    <h1>hello error</h1>
  </body>
</html>

```

TypeError: `init` is not a function  
at <unknown>(http://localhost/index.html:8:5)

Совпадает с адресом страницы и номером строки в html документе

## Получаем содержимое инлайн скрипта по стекейсу

```
● ● ●  
  
// Сохраняем копию html как можно раньше  
const html = 'doctype\n' + document.documentElement.outerHTML;  
  
function getStackWithInlineScripts(error: Error) {  
  const stack = parseStack(error.stack);  
  
  for (const frame of stack) {  
    if (isSameUrl(frame.filename, window.location.href)) {  
      frame.scriptContent = getScriptByLine(html, frame.line, frame.column);  
    }  
  }  
  
  return stack;  
}
```

## Собираем содержимое инлайн скриптов в Sentry

```
import * as Sentry from "@sentry/browser";
import { contextLinesIntegration } from "@sentry/integrations";

Sentry.init({
  dsn: "...",
  integrations: [
    // Support for inline scripts content
    contextLinesIntegration(),
  ],
});
```

# Пример инлайн скрипта в Sentry

Stack Trace

Most Relevant

Full Stack Trace

↑↓ Newest

...

## Error

in project inline script

mechanism

onunhandledrejection

handled

false

JS

index.html in <unknown> at line 32:22

In App

```
27     </head>
28     <body>
29       <div id="root"></div>
30       <script type="module" src="/src/main.tsx"></script>
31       <script type="module" nonce="test">
32         Promise.reject(new Error('in project inline script'));
33
34       (function(w,d,s,l,i){
35         w[l]=w[l]||[];
36         w[l].push({ cspNonce: 'test' });
37         w[l].push({'gtm.start':new Date().getTime(),event:'gtm.js'});
```

# Контролируем полноту логов





## Теряем ошибки в бизнес-логике

```
try {  
    await fetch();  
} catch(e) {  
    showError('Failed to load');  
}
```

## Теряем ошибки в бизнес-логике

```
try {  
    await fetch();  
} catch(e) {  
    if (e instanceof NetworkError) {  
        showError('Network error');  
    } else {  
        throw e;  
    }  
}
```

## Теряем ошибки в бизнес-логике

```
fetch().catch(e => {  
  showError('Failed to load');  
  return null;  
});
```

# Линтер для catch блоков



Пример eslint правила: <https://tinyurl.com/err-lint-rule>

## Логирование ошибки внутри try-catch

```
try {  
    await fetch();  
} catch(e) {  
    if (e instanceof NetworkError) {  
        showError('Network error');  
        Sentry.captureException(e);  
        return;  
    }  
    throw e;  
}
```

# Перехват КОНСОЛИ

- > Не всегда логирование находится в нашем коде
- > Единственный стандартный способ логирования – console
- > Часто встречается в библиотеках

```
console.error(new Error('Failed to render'))
```

```
console.warn(new Error('Minified React error'))
```

## Логирование для console.warn и console.error

```
const originalWarn = console.warn;
console.warn = (...args: unknown[]) => {
  Sentry.captureException(new ConsoleCallError(args), { level: 'warning' });
  originalWarn.apply(console, args);
};

const originalError = console.error;
console.error = (...args: unknown[]) => {
  Sentry.captureException(new ConsoleCallError(args), { level: 'error' });
  originalError.apply(console, args);
};
```

## Перехват консоли вместе с Sentry

```
import * as Sentry from "@sentry/browser";
import { captureConsoleIntegration } from "@sentry/integrations";

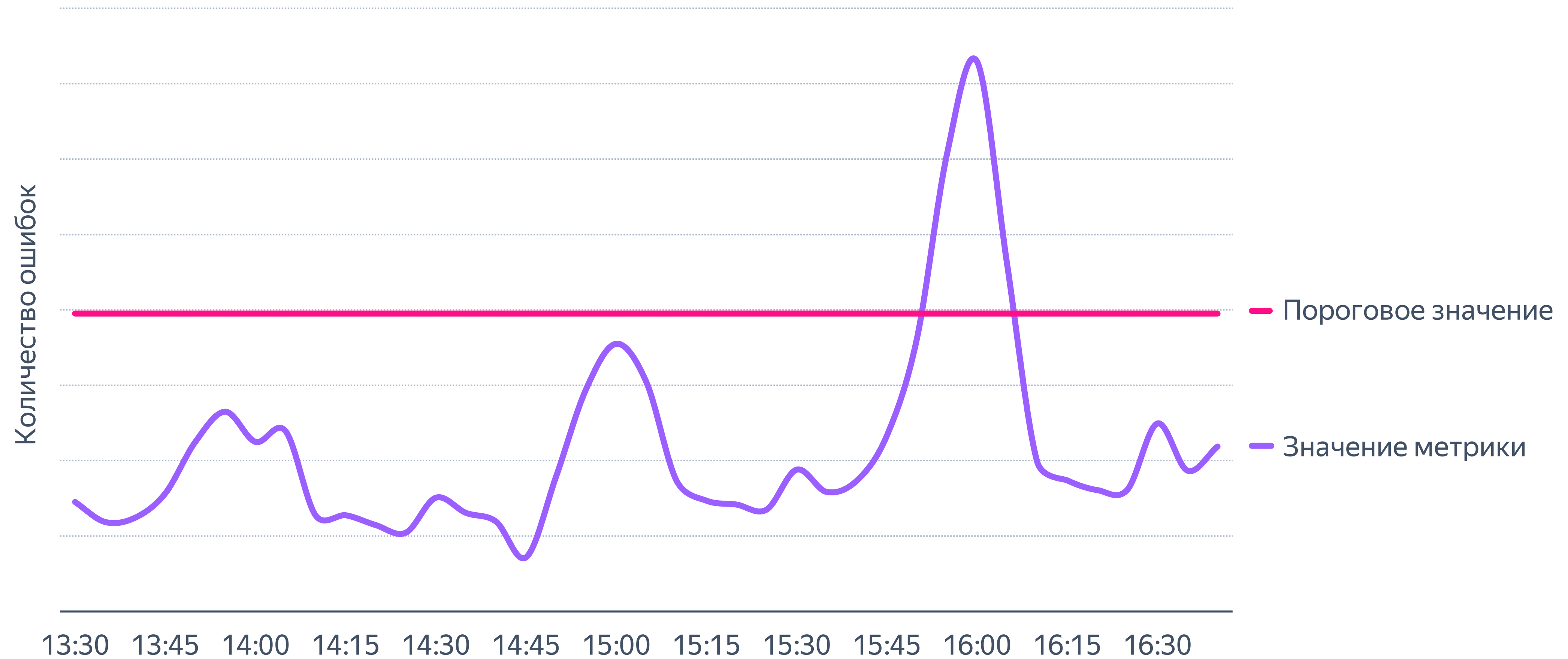
Sentry.init({
  dsn: "...",
  integrations: [
    captureConsoleIntegration(),
  ],
});
```



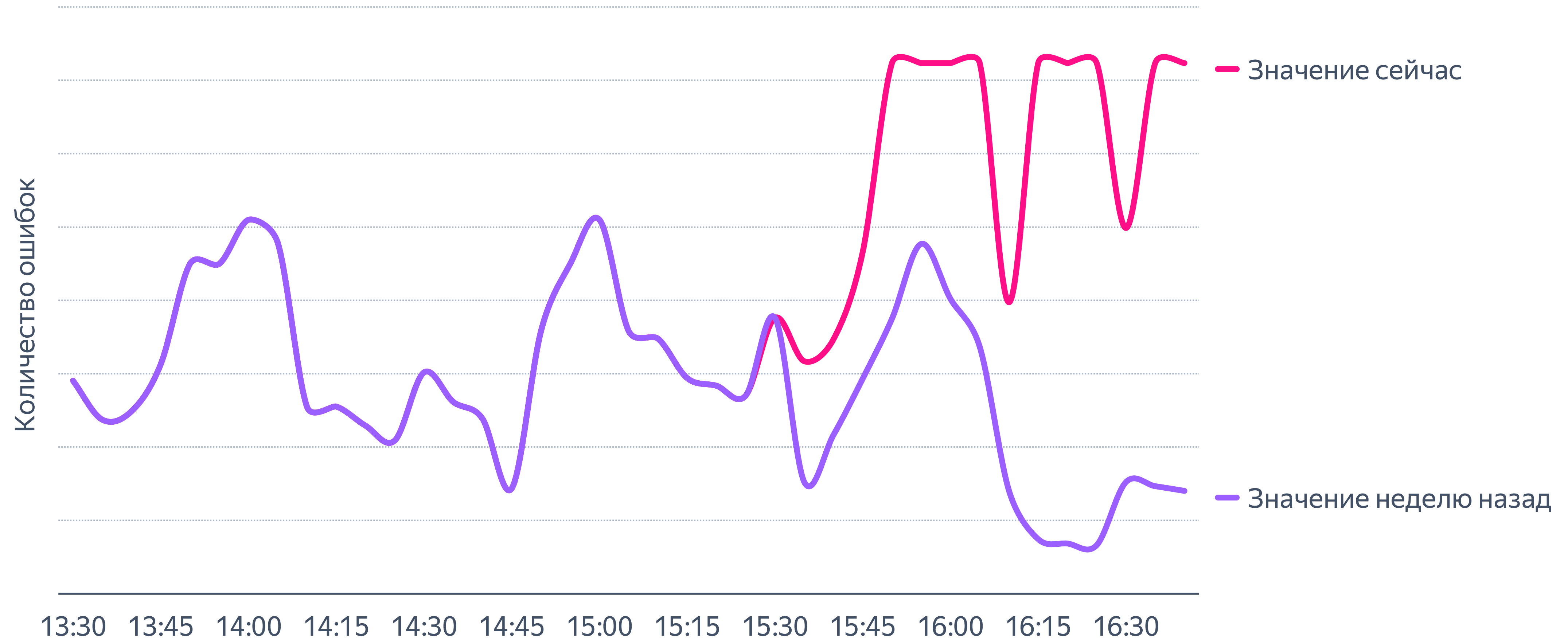
# Настраиваем уведомления



# Пороговые алерты



# Трендовые алерты



# Пороговые

## Новые программные ошибки

Быстро находить и анализировать новые ошибки

## Пользователи с ошибками

Узнавать про массовые проблемы с большим количеством пользователей

# Трендовые

## Известные программные ошибки

Следить за резким ростом уже известных ошибок

## Операционные ошибки

Следить за аномалиями в ожидаемых ошибках

# Примеры дополнительных алертов

- > Ошибки валидации в формах
- > Медленные запросы за данными
- > Показы уведомлений об ошибках
- > Dead clicks
- > Ошибки и отмены оплаты
- > Rage clicks

# ПОДВОДИМ ИТОГИ



# Как внедрить Zero Error Policy

## Подготовка

- 01 Подключить систему мониторинга
- 02 Подготовить код приложения
- 03 Настроить уведомления и процесс разбора

## Процесс

- > Анализ ошибок из уведомлений
- > Исправление багов
- > Устранение новых источников шума

# Результат внедрения Zero Error Policy

**1M → 1.5K**

Ошибок в сутки

**5 минут**

Время реакции  
на новую ошибку

**300+**

Обнаружили  
и починили багов



Хороший мониторинг  
ошибок приносит пользу

... а не боль для команды

Максим Земсков  
Ведущий разработчик интерфейсов

Ⓜ Реклама

Спасибо за внимание!