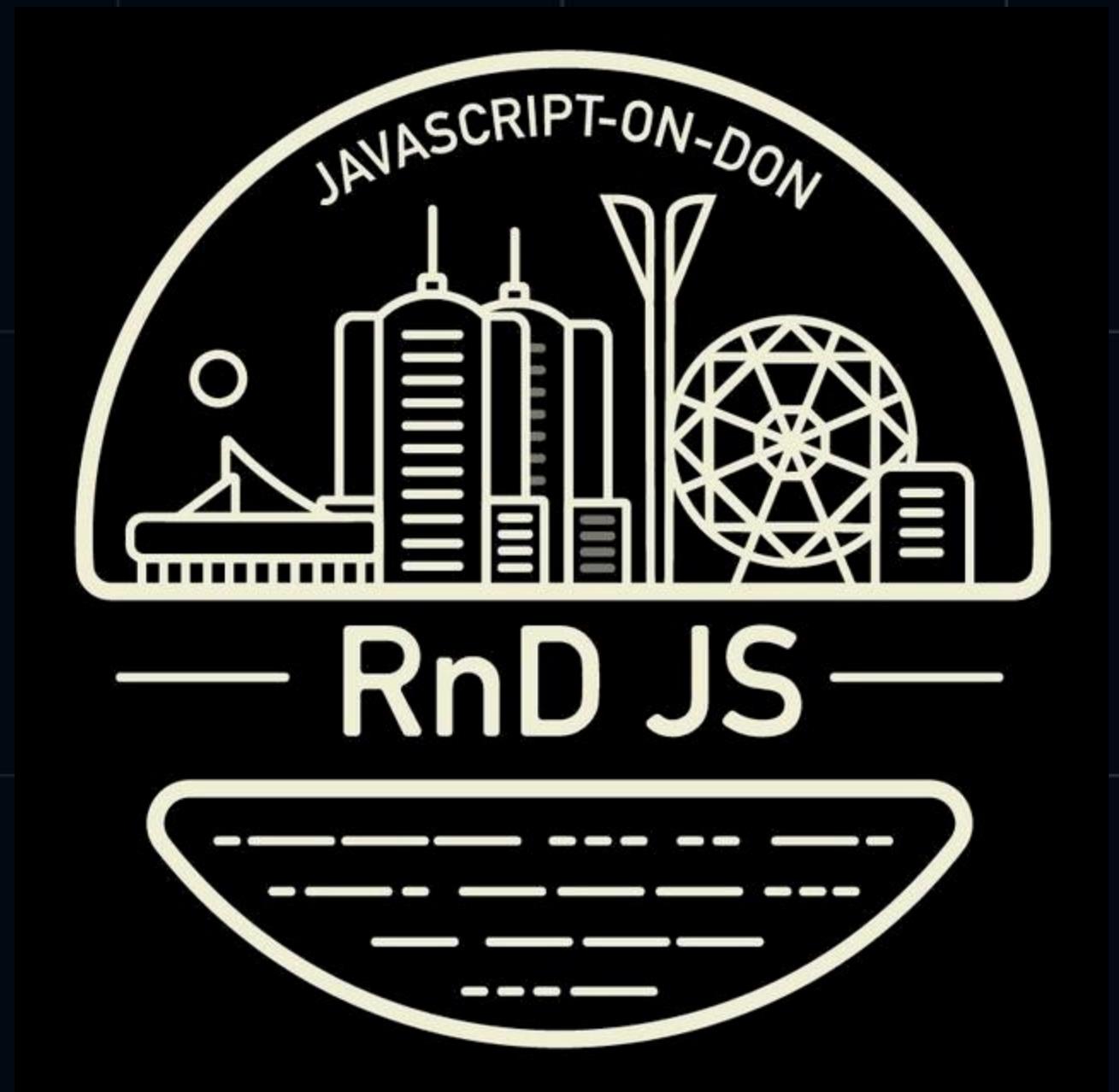


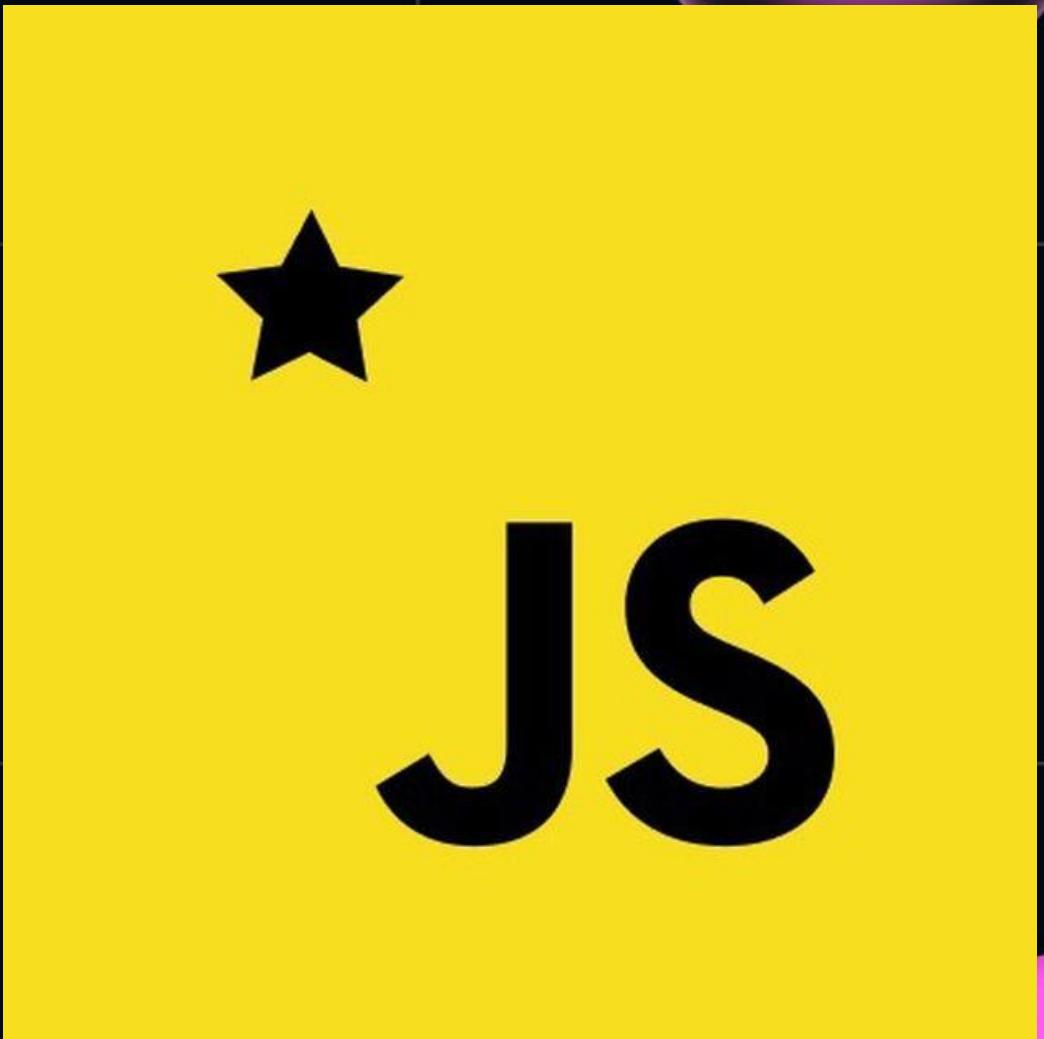
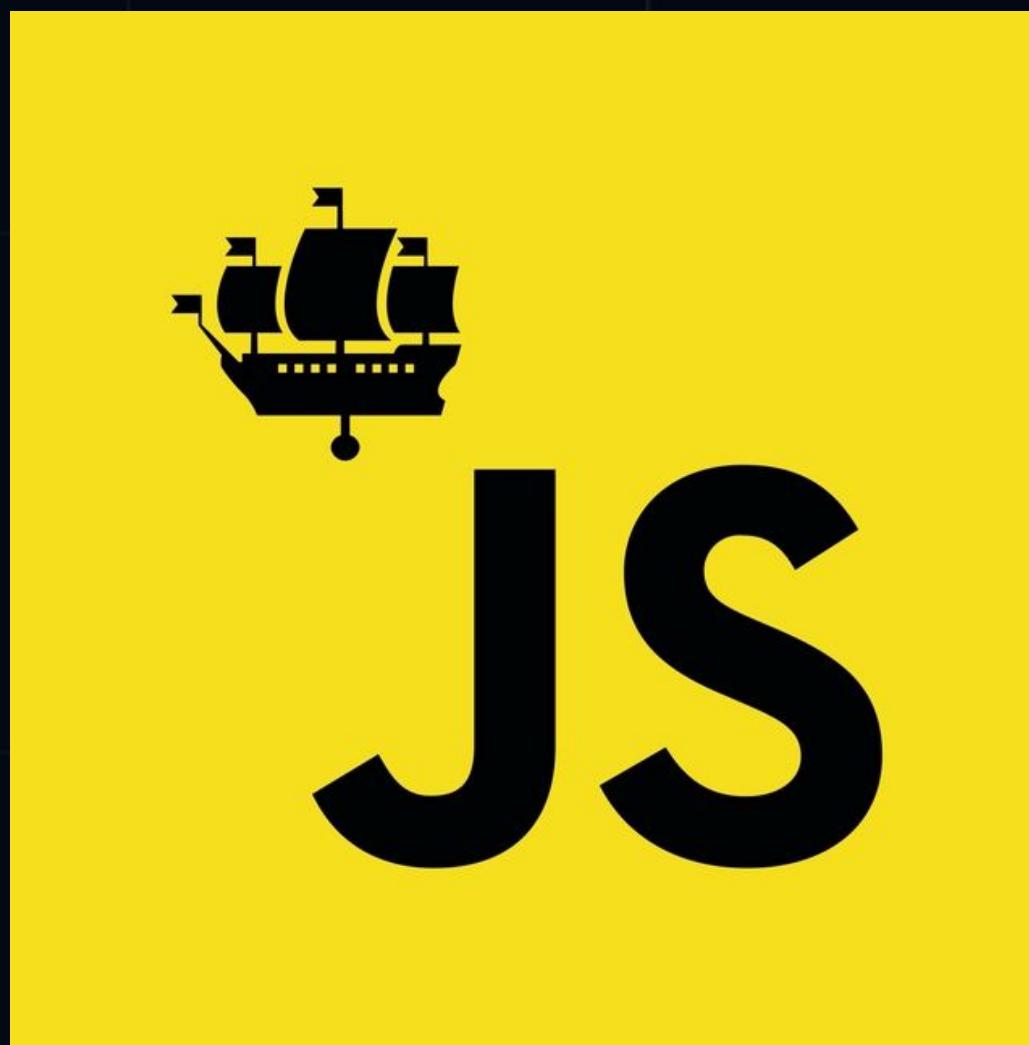
# Real DTO



Виктор  
Вершанский







# Bio

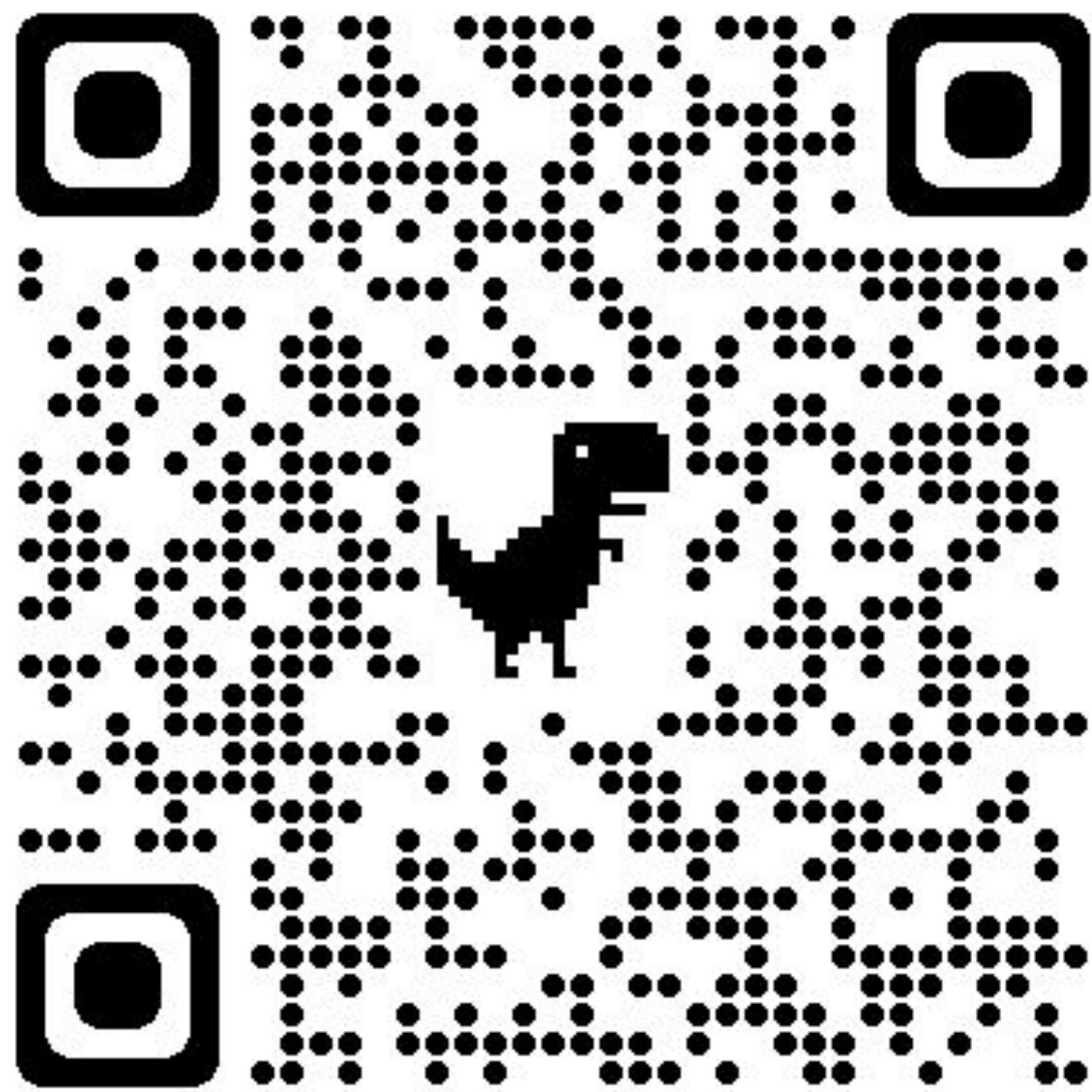


Виктор



wentout

- JS в продакшен 1999
- Back-End на JS в 2000
- Node.js с 2009
- Diagnostics Group
- BUGs Chrome & v8
- PhD in Economy of IT
- PMI PMBoK + Agile



# о чём будет идти речь

- контекст постановки задачи
- формулировка проблематики
- про что уже рассказывал по теме
- как создаётся код для решения
- как развивается решение: что сделано
- практический пример использования
- что ещё можно сделать дальше





# КОНТЕКСТ



# КОНТЕКСТ



1

- Контекст постановки задачи



- **контекст постановки задачи**

Вопросы на собеседовании



Код с продакшена



- **контекст постановки задачи**

Вопросы на собеседовании



Код с продакшена

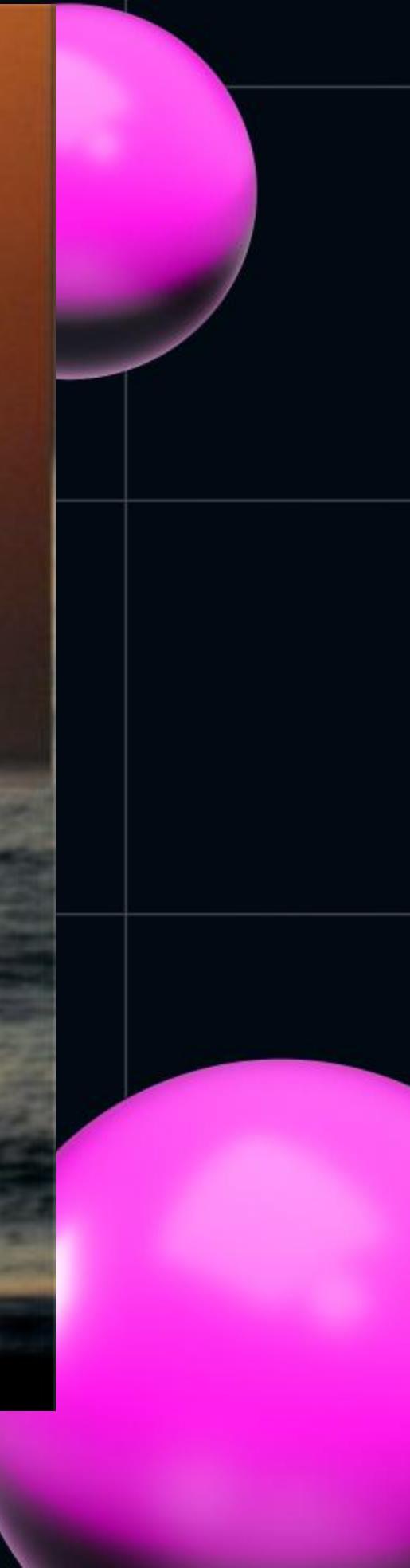
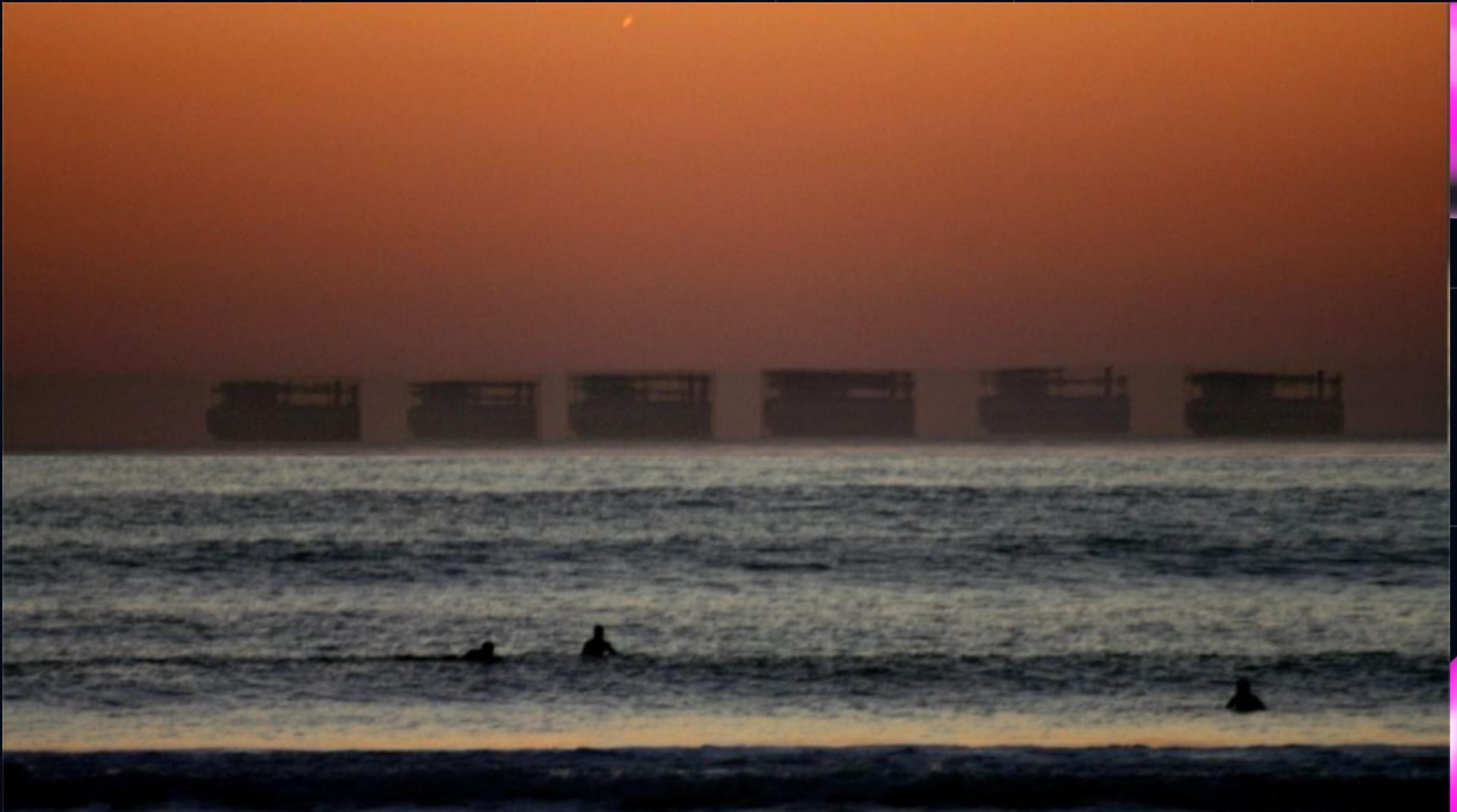




# Fata Morgana



# Fata Morgana



# Fata Morgana (mirage)

文 A 38 languages ▾

Article Talk

Read Edit View history Tools ▾

From Wikipedia, the free encyclopedia

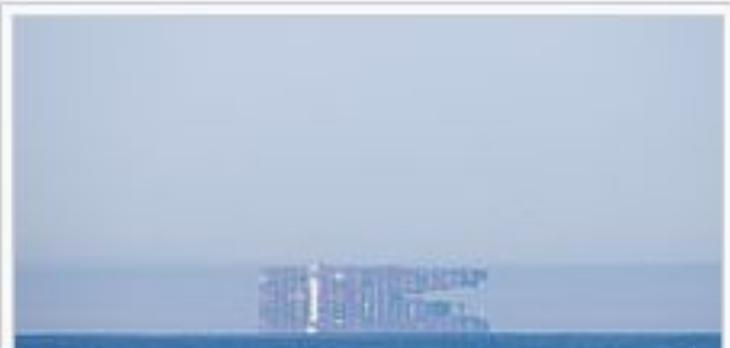
A **Fata Morgana** (Italian: [fata mor'ga:na]) is a complex form of superior mirage visible in a narrow band right above the horizon. The term *Fata Morgana* is the Italian translation of "Morgan the Fairy" (*Morgan le Fay* of Arthurian legend). These mirages are often seen in the Italian Strait of Messina, and were described as fairy castles in the air or false land conjured by her magic.

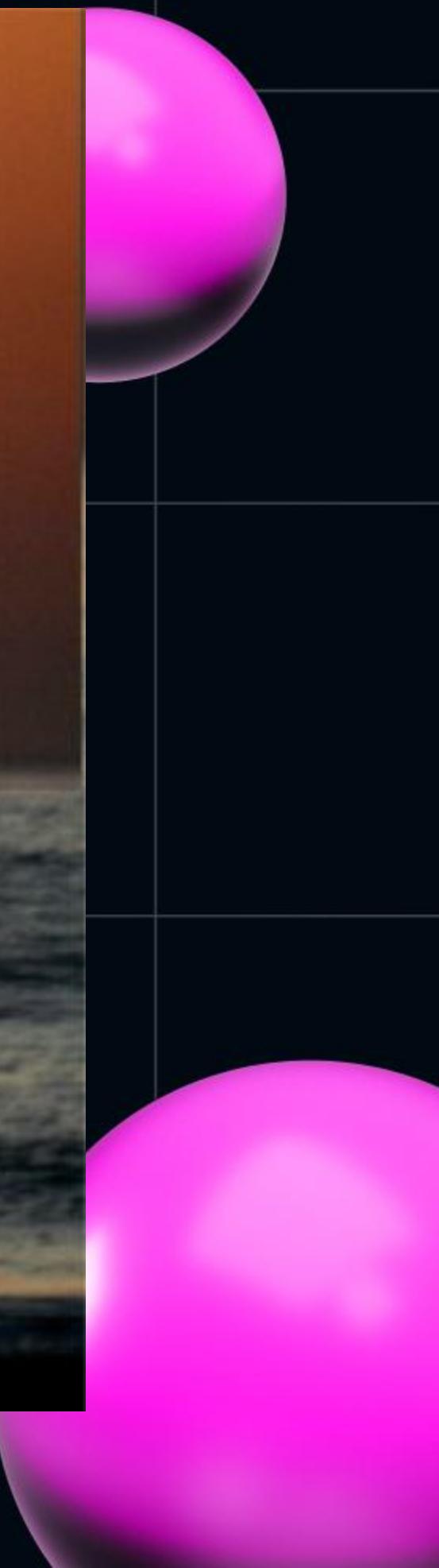
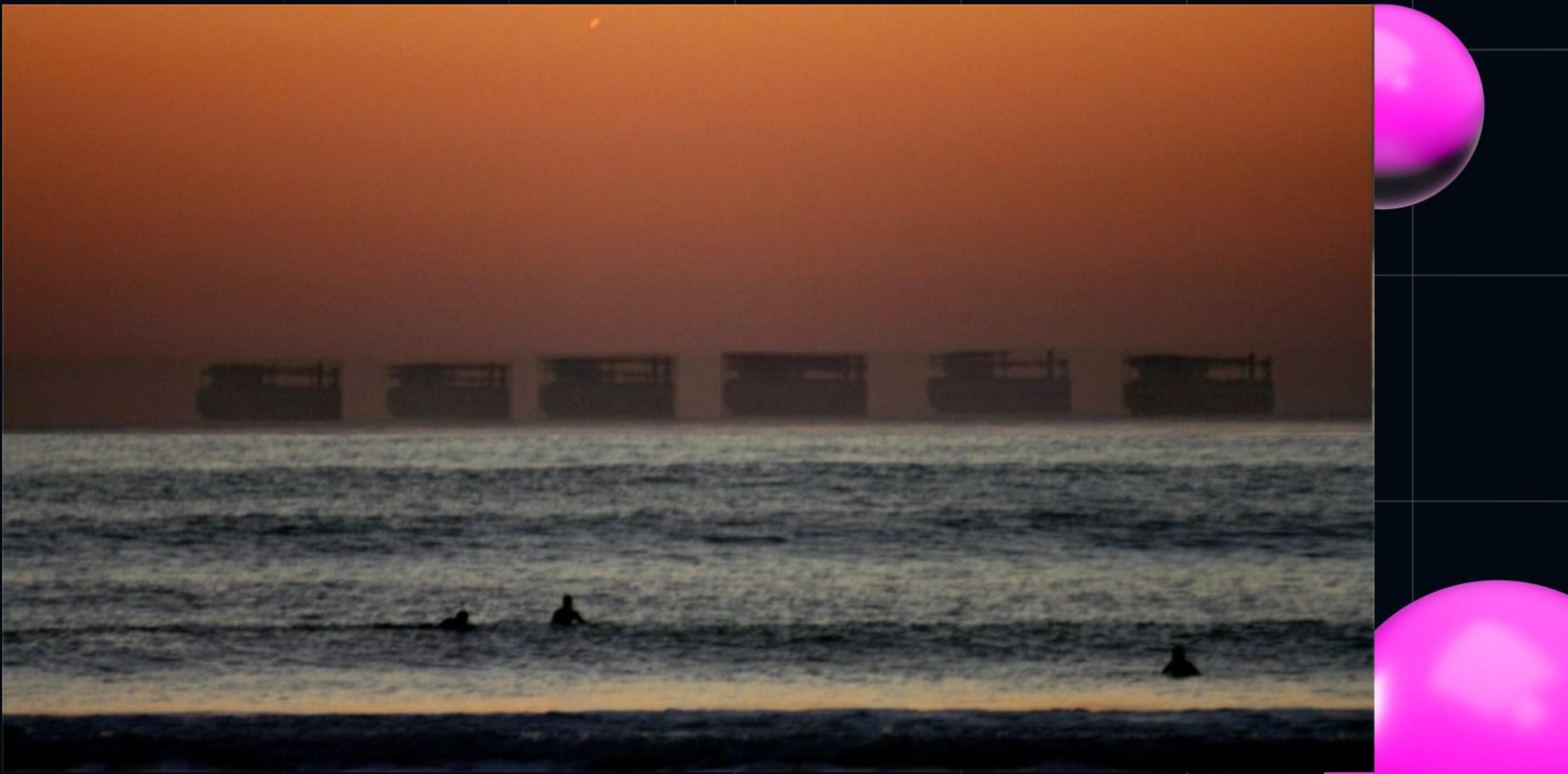
Fata Morgana mirages significantly distort the object or objects on which they are based, often such that the object is completely unrecognizable. A Fata Morgana may be seen on land or at sea, in polar regions, or in deserts. It may involve almost any kind of distant object, including boats, islands, and the coastline. Often, a Fata Morgana changes rapidly. The mirage comprises several inverted (upside down) and erect (right-side up) images that are stacked on top of one another. Fata Morgana mirages also show alternating compressed and stretched zones.<sup>[1]</sup>

The optical phenomenon occurs because rays of light bend when they pass through air layers of different temperatures in a steep thermal inversion where an atmospheric duct has formed.<sup>[1]</sup> In calm weather, a layer of significantly warmer air may rest over colder dense air, forming an atmospheric duct that acts like a refracting lens, producing a series of both inverted and erect images. A Fata Morgana requires a duct to be present; thermal inversion alone is not enough to produce this kind of mirage. While a thermal inversion often takes place without there being an atmospheric duct, an atmospheric duct cannot exist without there first being a thermal inversion.



A Fata Morgana seen over the Baltic Sea, 2016. The mirage consists of multiple upright and inverted images over the original object





**как всё началось ...**



# uKit

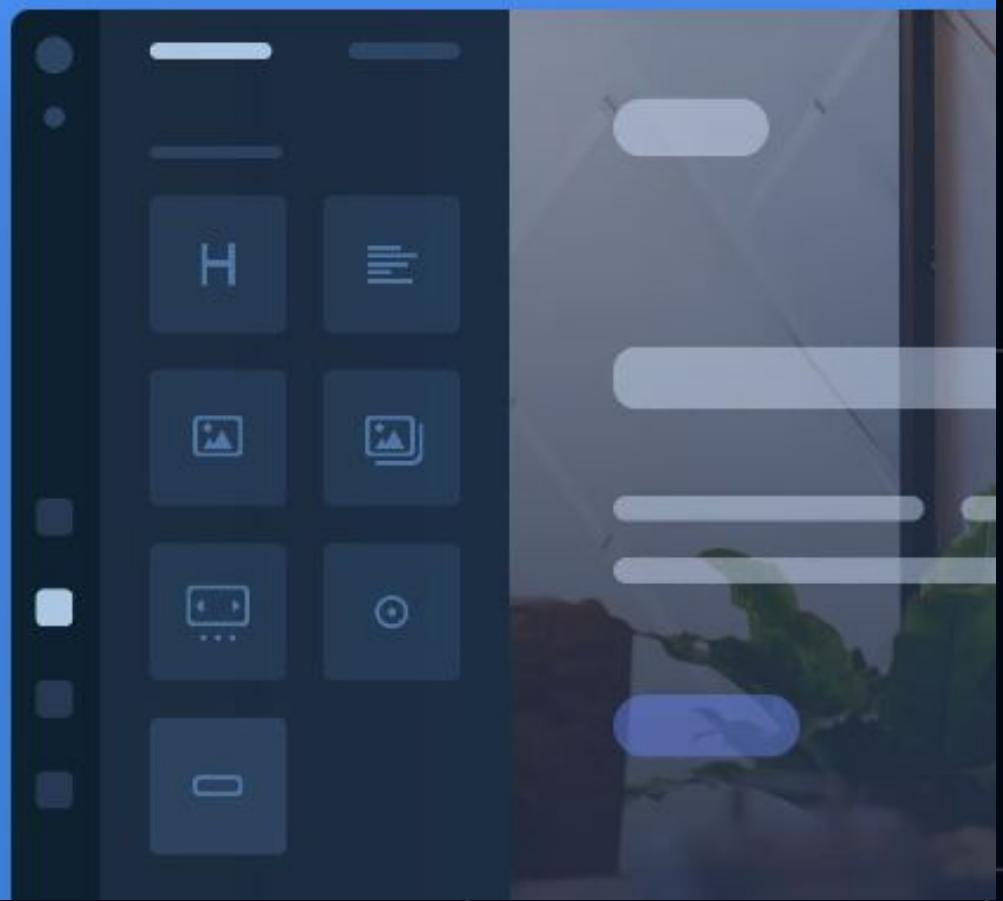
[Почему мы?](#)[Цены](#)

## Сайты uKit доступны без перебоев

Наши серверы находятся на территории России, доступ к сайтам работает без ограничений. Конструктор uKit зарегистрирован в [Едином реестре российского ПО](#)

# Конструктор сайтов для бизнеса

Создайте сайт своими руками

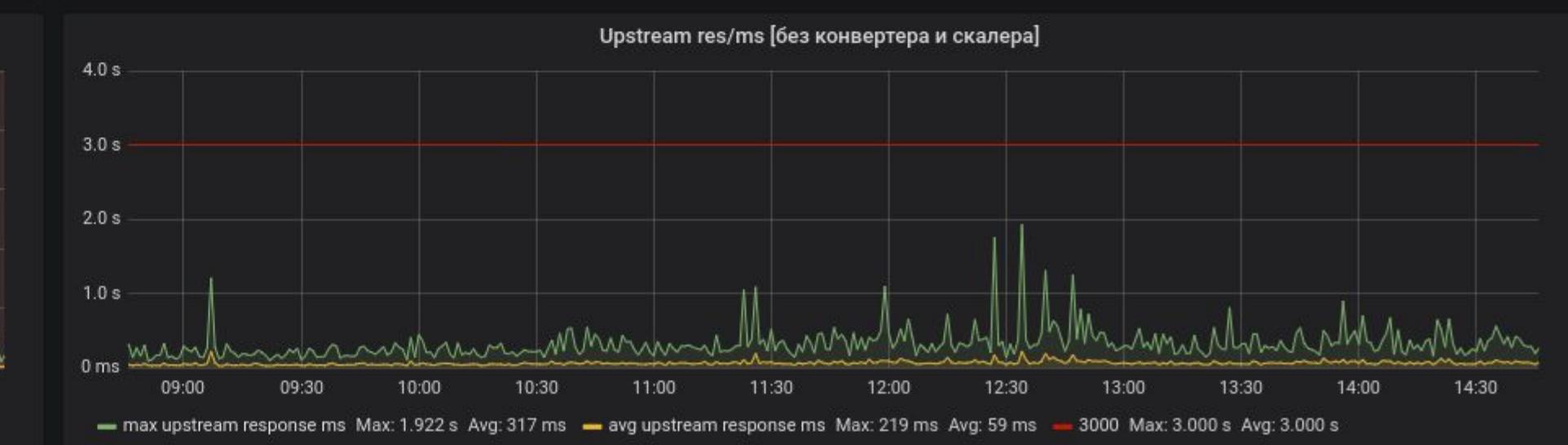
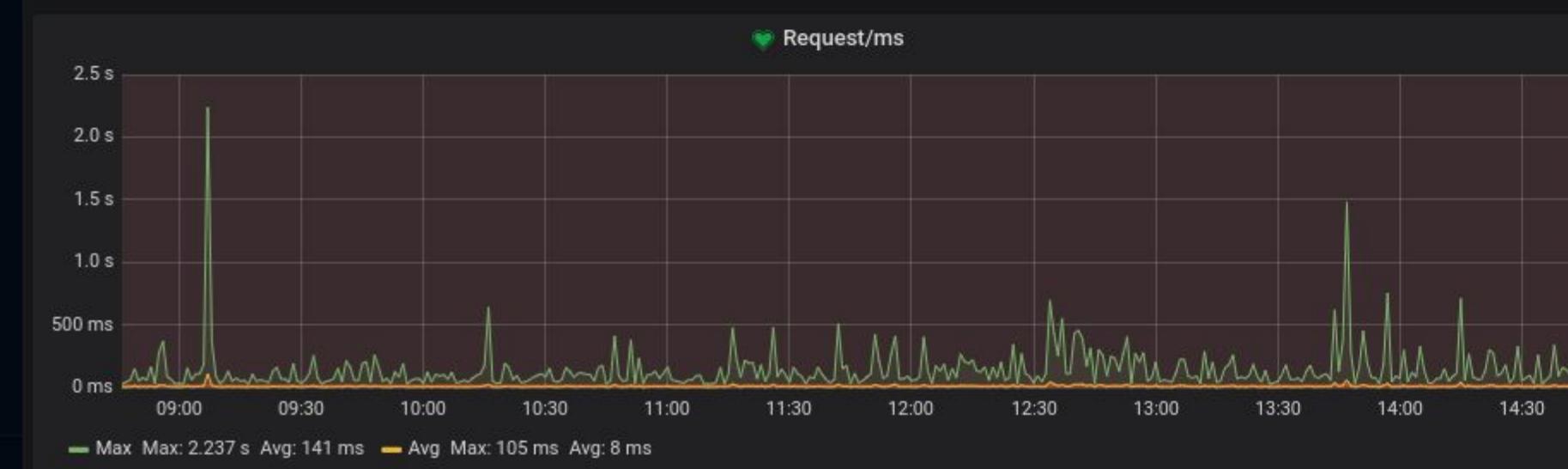
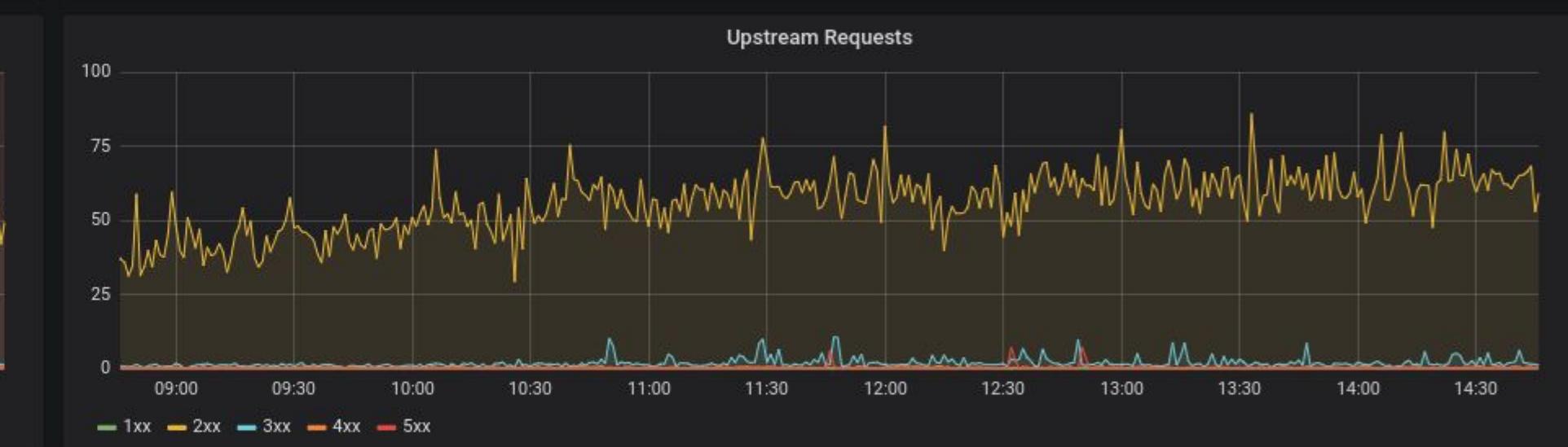
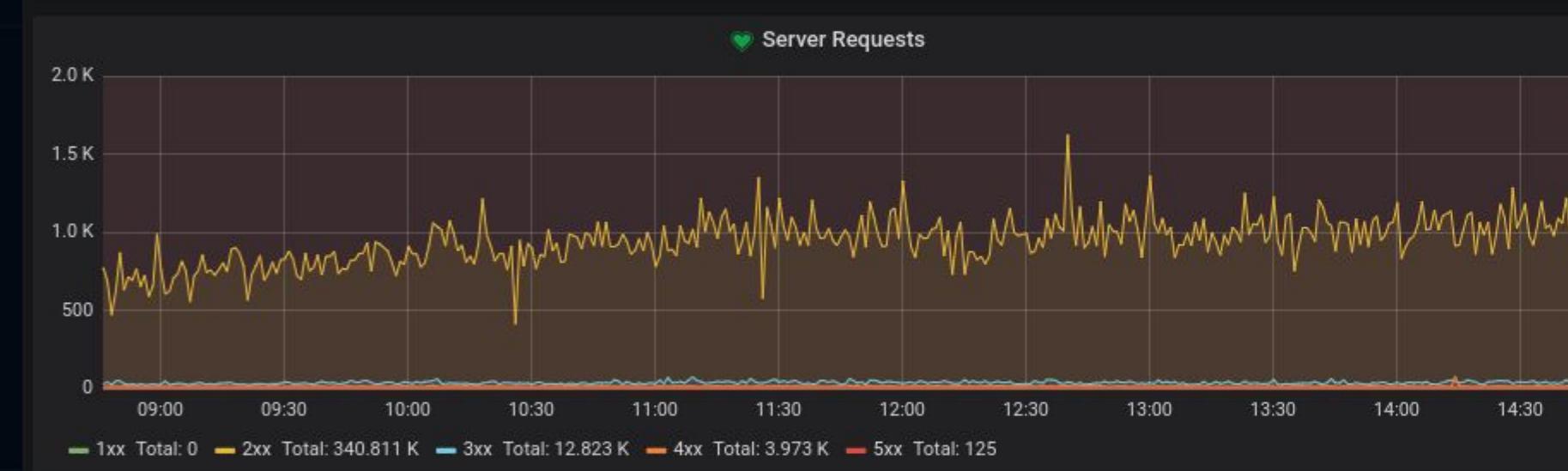
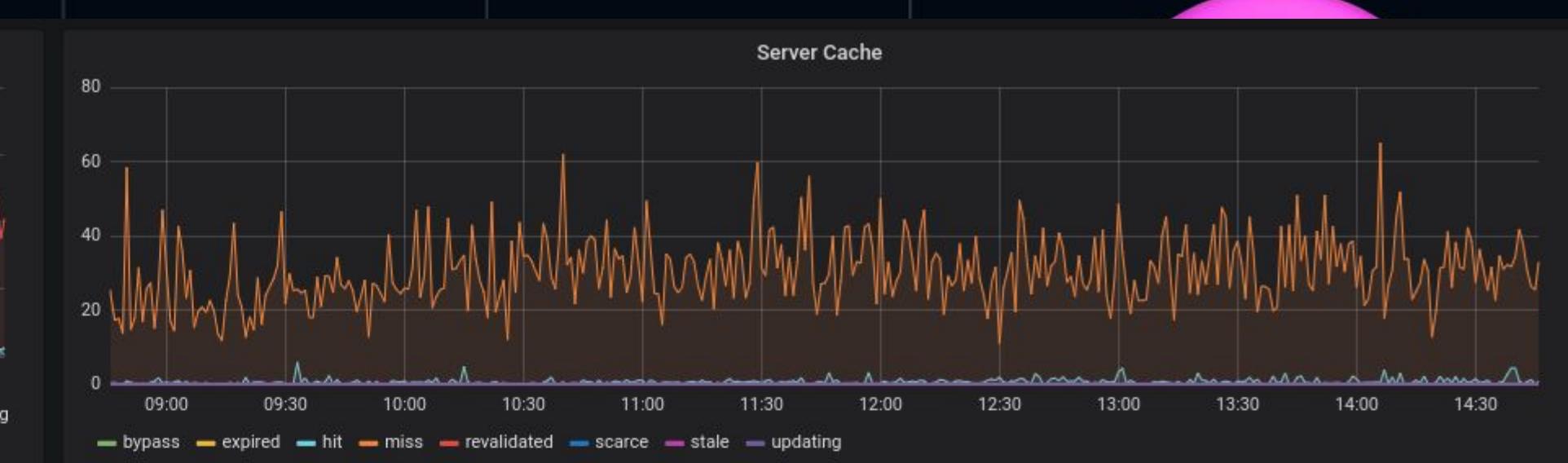
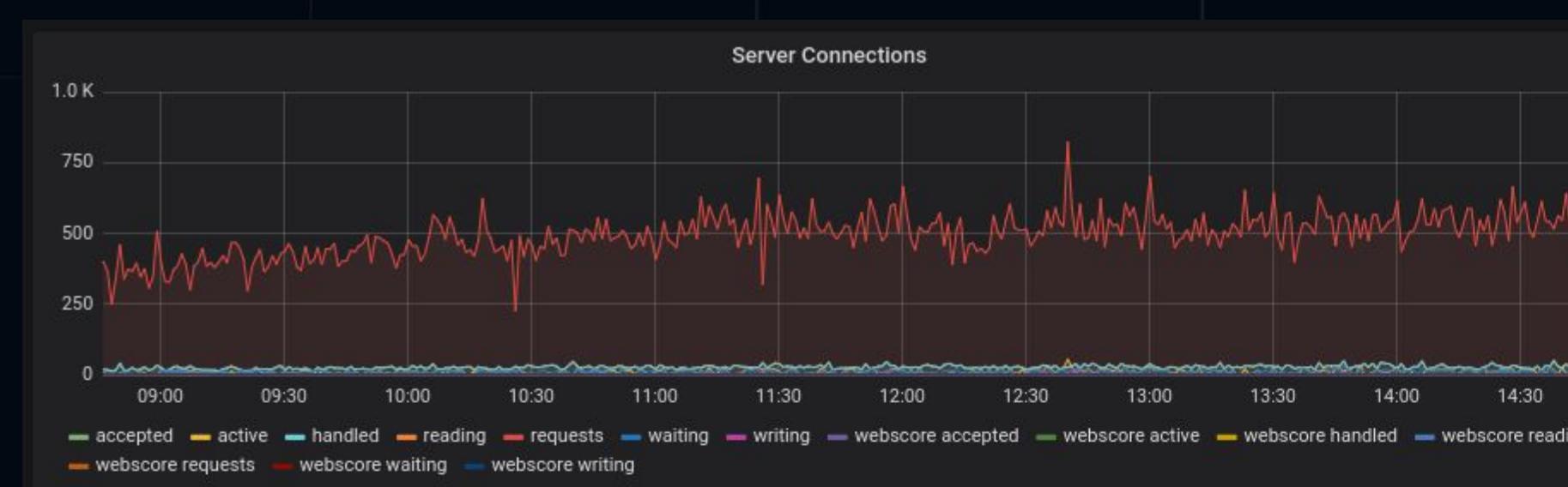
[Создать сайт](#)[▶ Смотреть видео](#)











500

SERVER ERROR

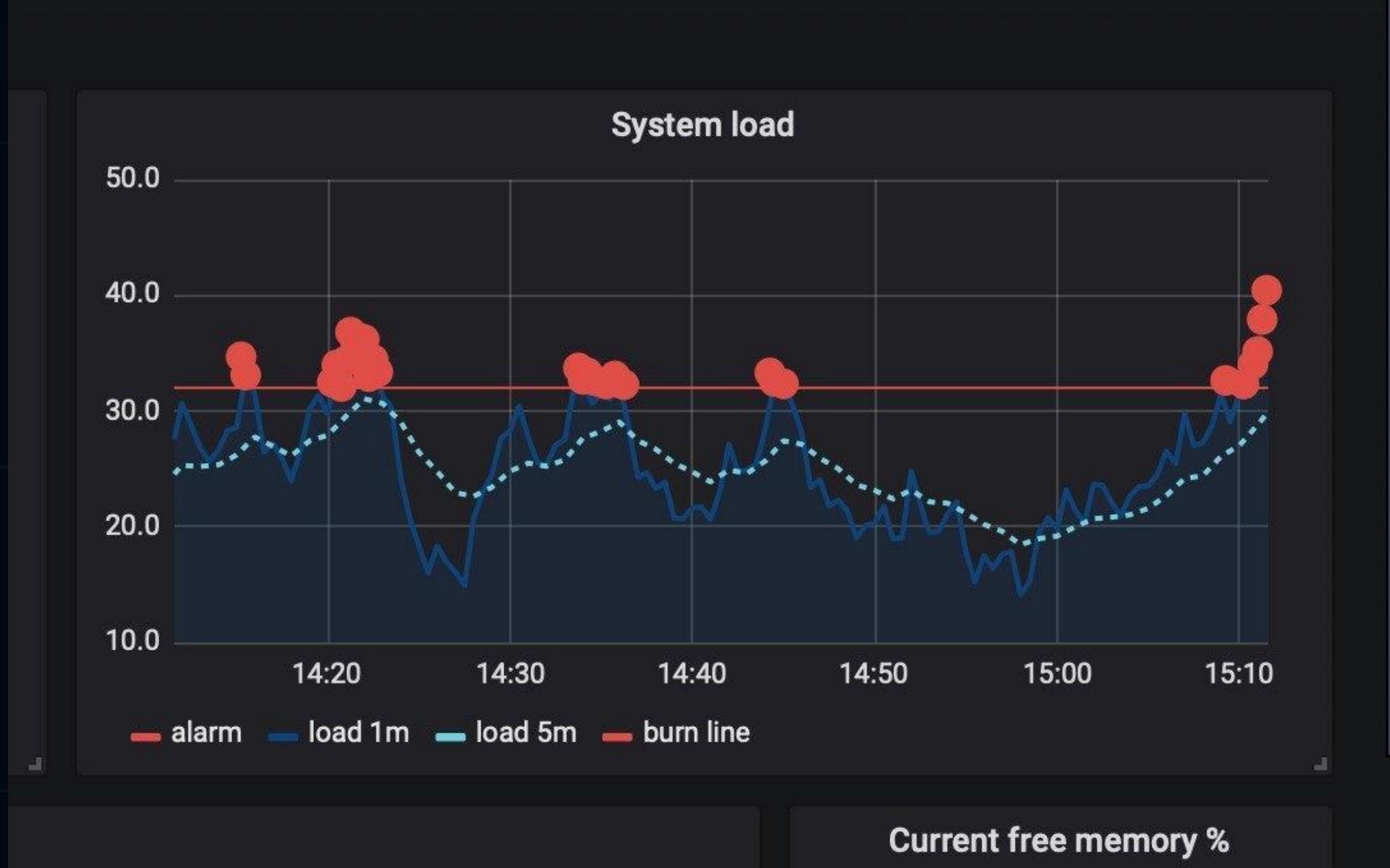
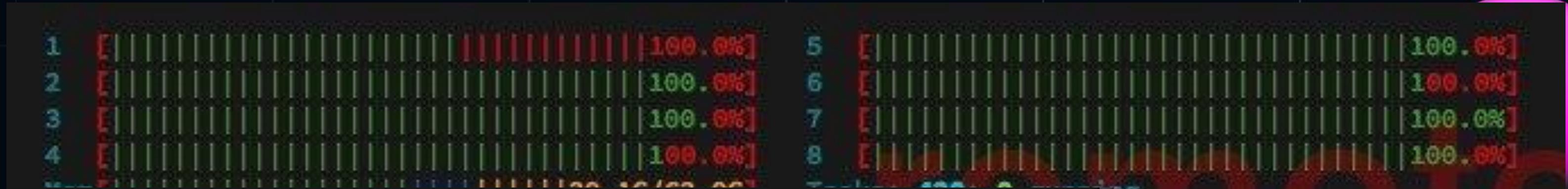
ПЯТИСОТАЯ

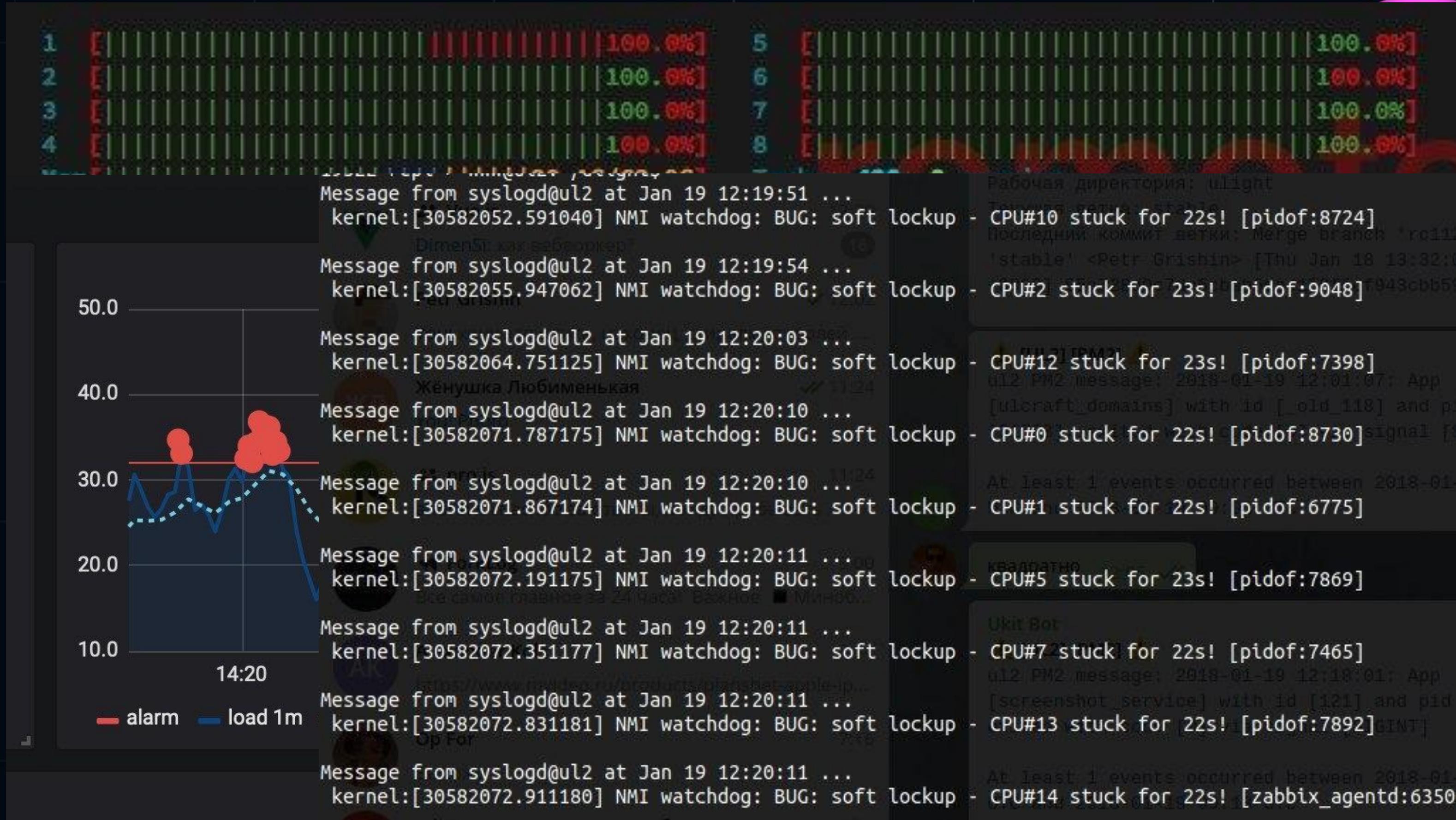
Sticky

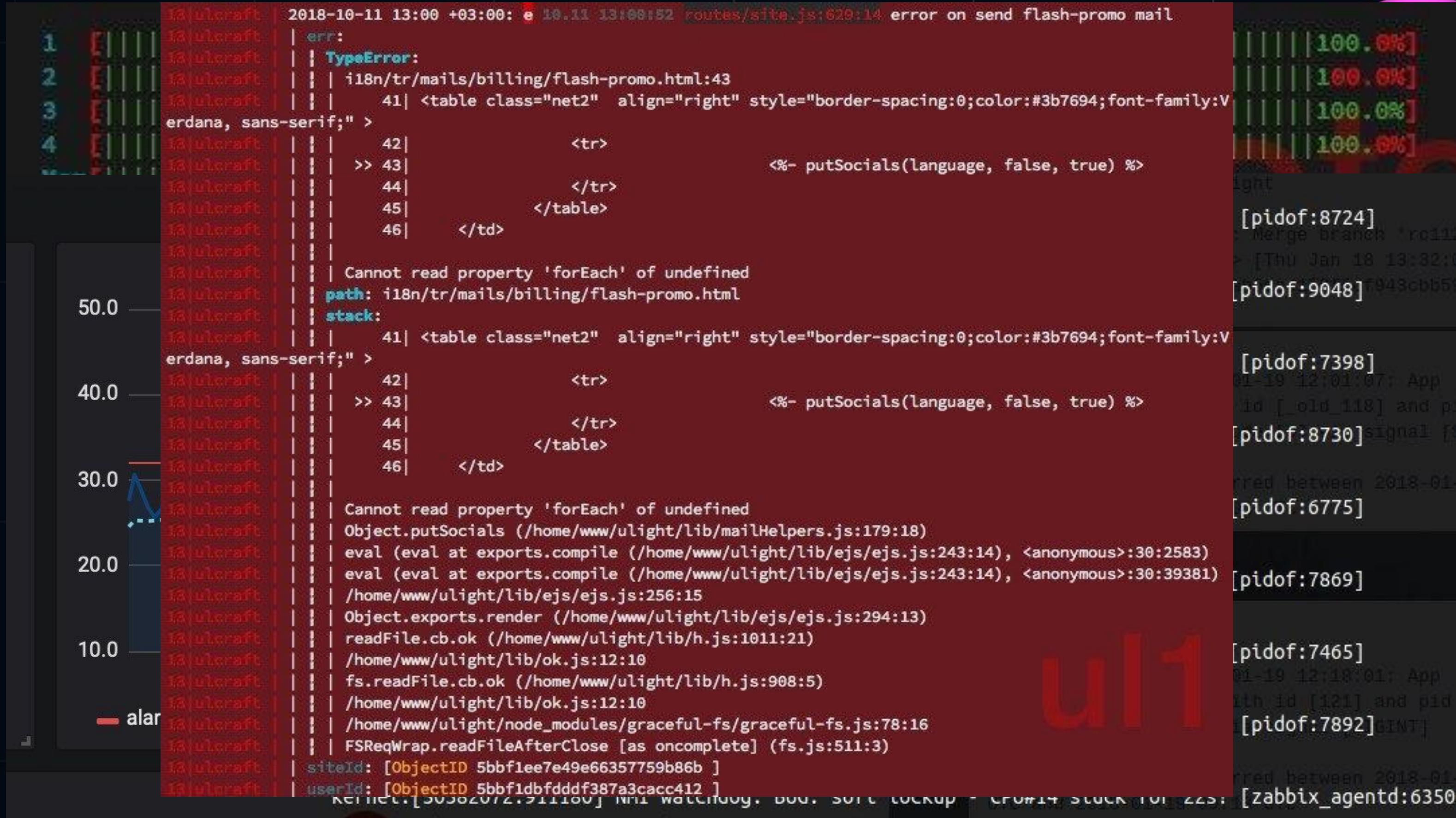




```
1 E [██████████] 100.0% 5 E [██████████] 100.0%  
2 E [██████████] 100.0% 6 E [██████████] 100.0%  
3 E [██████████] 100.0% 7 E [██████████] 100.0%  
4 E [██████████] 100.0% 8 E [██████████] 100.0%  
Mem E [██████████] 29.1G/62.9G Tasks: 420; 9 running  
CPU: 0.0% user, 0.0% system, 0.0% idle
```









# формулировка



# формулировка

2

- Контекст постановки задачи
- **формулировка проблематики**



```
2018-10-11 13:00 +03:00: e 10.11 13:00:52 routes/site.js:620:14 error on send flash-promo mail
| err:
|   | TypeError:
|   |   i18n/tr-mails/billing/flash-promo.html:43
|   |     41| <table class="net2" align="right" style="border-spacing:0;color:#3b7694;font-family:Verdana, sans-serif;" >
|   |       42|           <tr>
|   |         >> 43|             <%- putSocials(language, false, true) %>
|   |           44|           </tr>
|   |         45|           </table>
|   |       46|     </td>
|   |
|   |   Cannot read property 'forEach' of undefined
|   |   path: i18n/tr-mails/billing/flash-promo.html
|   |   stack:
|   |   41| <table class="net2" align="right" style="border-spacing:0;color:#3b7694;font-family:Verdana, sans-serif;" >
|   |       42|           <tr>
|   |         >> 43|             <%- putSocials(language, false, true) %>
|   |           44|           </tr>
|   |         45|           </table>
|   |       46|     </td>
|   |
|   |   Cannot read property 'forEach' of undefined
|   |   Object.putSocials (/home/www/ulight/lib/mailHelpers.js:179:18)
|   |   eval (eval at exports.compile (/home/www/ulight/lib/ejs/ejs.js:243:14), <anonymous>:30:2583)
|   |   eval (eval at exports.compile (/home/www/ulight/lib/ejs/ejs.js:243:14), <anonymous>:30:39381)
|   |   /home/www/ulight/lib/ejs/ejs.js:256:15
|   |   Object.exports.render (/home/www/ulight/lib/ejs/ejs.js:294:13)
|   |   readFile.cb.ok (/home/www/ulight/lib/h.js:1011:21)
|   |   /home/www/ulight/lib/ok.js:12:10
|   |   fs.readFile.cb.ok (/home/www/ulight/lib/h.js:908:5)
|   |   /home/www/ulight/lib/ok.js:12:10
|   |   /home/www/ulight/node_modules/graceful-fs/graceful-fs.js:78:16
|   |   FSReqWrap.readFileAfterClose [as oncomplete] (fs.js:511:3)
|   |
|   | siteId: [ObjectID 5bbf1ee7e49e66357759b86b ]
|   | userId: [ObjectID 5bbf1dbfddd387a3cacc412 ]
```



2018-10-11 13:00 +03:00: e 10.11 13:00:52 routes/site.js:629:14 error on send flash-promo mail  
| err:  
| | TypeError:  
| | i18n/tr-mails/billing/flash-promo.html:43  
| | :0;color:#3b7694;font-family:V  
| | page, false, true) %>  
  
**ЭТО БЫЛО  
НЕВОЗМОЖНО**  
  
**ЗАПЛАНИРОВАТЬ**  
siteId: [ObjectId 5bbf1ee7e49e66357759b86b ]  
userId: [ObjectId 5bbf1dbfdff387a3cacc412 ]  
:0;color:#3b7694;font-family:V  
page, false, true) %>  
13:14), <anonymous>:30:2583)  
13:14), <anonymous>:30:39381)





**ИЗЮМА  
ТЯПНУЛИ**

# А ДАВАЙТЕ ЧТО-НИБУДЬ

# ВНЕДРИМ

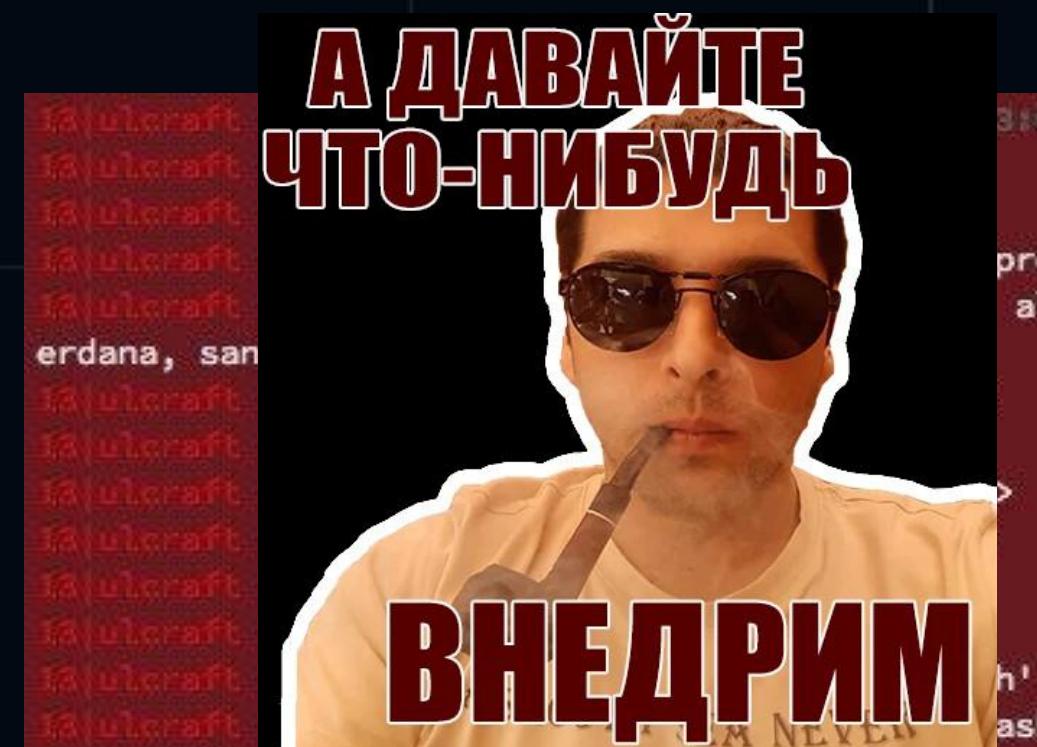


# ЭТО БЫЛО НЕВОЗМОЖНО



**ИЗЮМА  
ТЯПНУЛИ**

Sticky



# А ДАВАЙТЕ ТО-НИБУДЬ

# ВНЕДРИМ



# У ТЕБЯ ПРАВ НЕТУ

Sticky



# ЭТО БЫЛО НЕВОЗМОЖНО



# **ЗАПЛАНИРОВАТЬ**

**ХОТЯ ...**



**хотя ... если немного ...**





# HolyJS // Москва 2016

Thomas Watson

Opbeat

Debugging Node.js  
Performance Issues  
in Production





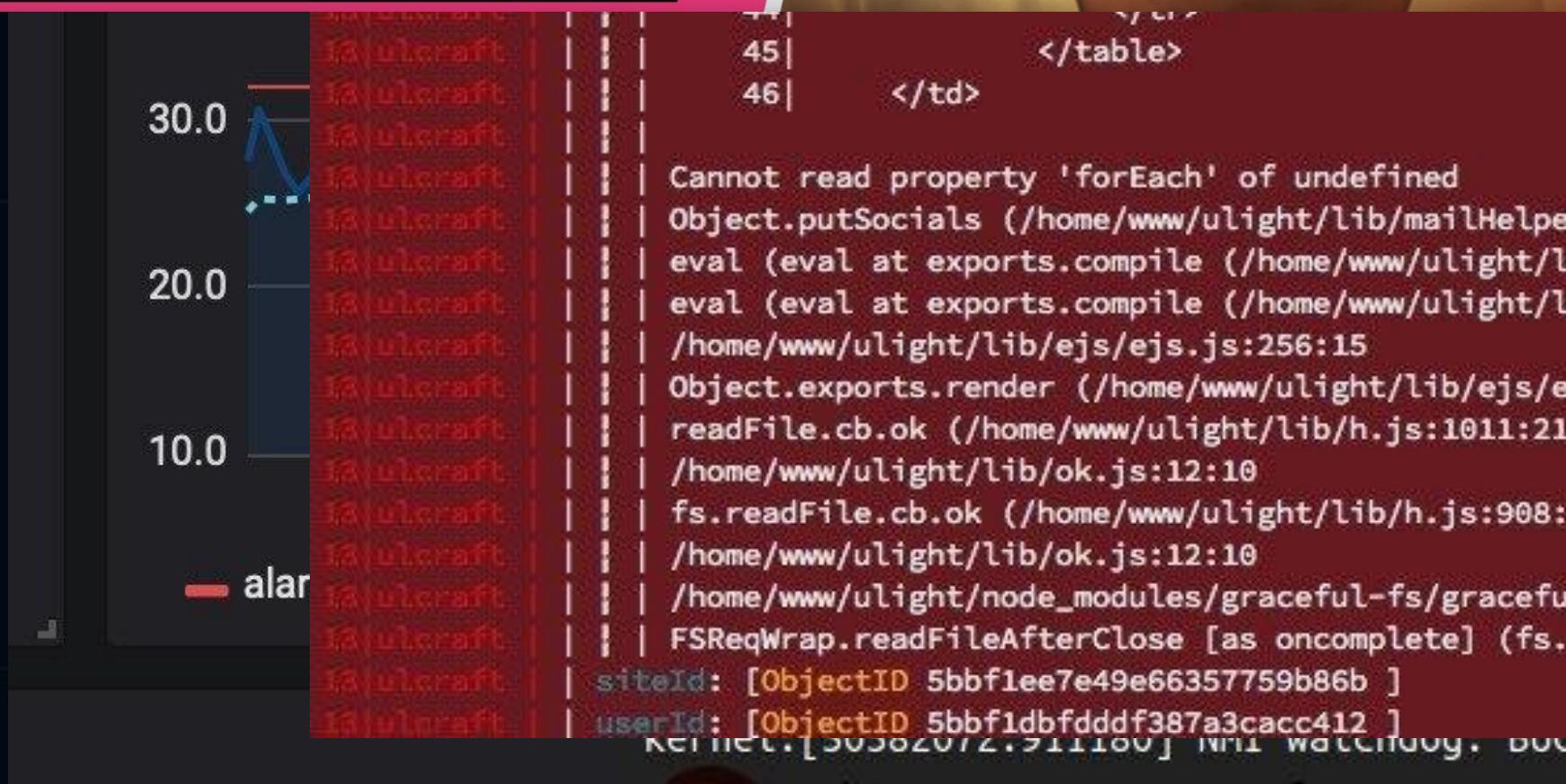
0.0%  
0.0%  
0.0%  
0.0%  
24] 18 13:32:0  
8] 7943cbb59  
98] 1.01: App  
118] and pid  
0] signal fo  
en 2018-01-  
5]  
9] 1.01: App  
5]  
92] GINT  
en 2018-01-  
gentd:6350

# HolyJS // Москва 2016

## Thomas Watson

Opbeat

### Debugging Node.js Performance Issues in Production



```
20:14 error on send flash-promo mail  
border-spacing:0;color:#3b7694;font-family:V  
Socials(language, false, true) %>
```



BoF-CE  
BoF-SES







# HolyJS

// Москва 2016

## Thomas Watson

Opbeat

Debugging Node.js  
Performance Issues  
in Production

Я ЗНАЮ  
КАК БЫСТРЕЕ

НО ЭТО БУДЕТ  
ДОЛЬШЕ



```
11 13:00:52 routes/s
flash-promo.html:43
t2" align="right" s
<tr>
</tr>
ole>
Each' of undefined
g/flash-promo.html
ight" s
```



```
defined
ib/mailHelpers.js:179:18)
www/ulight/lib/ejs/ejs.js:243:14), <anonymous>:30:2583)
www/ulight/lib/ejs/ejs.js:243:14), <anonymous>:30:39381)
5
nt/lib/ejs/ejs.js:294:13)
n.js:1011:21)

ib/h.js:908:5)

L-fs/graceful-fs.js:78:16
complete] (fs.js:511:3)
>
>

lruobj. bvu. sort tookup - cr0#14 stuck for 225: [zabbix_agentd:6350]
```

## Thomas Watson

Opbeat

Debugging Node.js  
Performance Issues  
in Production

50.0

fullcraft | stack:



```
11 13:00:52 routes/s
flash-promo.html:43
t2" align="right" s
<tr>
</tr>
ole>
each' of undefined
calling/flash-promo.html
```



question about `async_hooks` ➔



**system**

Hi Tomas! Sorry for interrupting! Nice to meet you! My name is Victor, I'm from Russia, and I c



**Thomas Watson**

to me ▾

If I understand your proposal correctly, you are advocating for a 5th hook to be added to Async

```
ts@ulcraft: ~ 2018-10-11 13:00 +03:00: e 10.11.13.68:52 routes/site.js:629:14 error on send flash-promo mail
1 |   |
2 |   | err:
3 |   |   | TypeError:
|   |   |   | i18n/tr-mails/billing/flash-promo.html:43
|   |   |   |   41| <table class="net2" align="right" style="border-spacing:0;color:#3b7694;font-family:Verdana, sans-serif;" >
```

## question about async\_hooks ➔



**system**

Hi Tomas! Sorry for



**Thomas Watson**

to me ▾

If I understand your

```
context.js ×
const eid = hooks.eid; eid = 1
const tid = hooks.tid; tid = 0
debugger;
```

The screenshot shows a debugger interface with a variable pane. A variable named 'eid' is highlighted with a red border. A red arrow points from this highlighted variable to a local variable frame containing the value '0'. Another red arrow points from the same variable 'eid' to its declaration line in the code, which is also highlighted with a red border.

```
ts@ulcraft: ~ 2018-10-11 13:00: +03:00: e 10.11 13:00:52 routes/site.js:629:14 error on send flash-promo mail
1 |   E
2 |   E
3 |   E
| err:
|   | TypeError:
|   |   | i18n/tr-mails/billing/flash-promo.html:43
|   |   |   41| <table class="net2" align="right" style="border-spacing:0;color:#3b7694;font-family:Verdana, sans-serif;" >
```

## question about async\_hooks ➔



**system**

Hi Tomas! Sorry for



**Thomas Watson**

to me ▾

If I understand your

context.js ×

```
const eid = hooks.eid; eid = 1
const tid = hooks.tid; tid = 0
```

debugger;



```
ts@ulcraft: ~ 2018-10-11 13:00: +03:00: e 10.11 13:00:52 routes/site.js:629:14 error on send flash-promo mail
1 |   E
2 |   E
3 |   E
| err:
|   | TypeError:
|   |   | i18n/tr-mails/billing/flash-promo.html:43
|   |   |   41| <table class="net2" align="right" style="border-spacing:0;color:#3b7694;font-family:Verdana, sans-serif;" >
```

## question about async\_hooks ➔



system

Hi Tomas! Sorry for



Thomas Watson

to me ▾

If I understand your

context.js ×

```
const eid = hooks.eid; eid = 1
const tid = hooks.tid; tid = 0
```

debugger;



```
10|ulcraft | 2018-10-11 13:00 +03:00: e 10.11 13:00:52 routes/site.js:629:14 error on send flash-promo mail
11|ulcraft |
12|ulcraft |   err:
13|ulcraft |     | TypeError:
14|ulcraft |     |   | i18n/tr-mails/billing/flash-promo.html:43
15|ulcraft |     |   |   41| <table class="net2" align="right" style="border-spacing:0;color:#3b7694;font-family:Verdana, sans-serif;" >
```

# question about async\_hooks ➔

nodejs / diagnostics

Watch 108 Star 281 Fork 54

Code Issues 53 Pull requests 4 Projects 0 Wiki Insights

## Proposal of Polling~Queue: Sync Hook Problem #249

Closed wentout opened this issue 28 days ago · 4 comments



wentout commented 28 days ago · edited

Hi!

First of all I wish to say many thanks for all this happens!  
Really nice API, and works just good enough!

And, I'm very sorry, but I need to ask a question about the issue I'm under and unfortunately unable to solve by myself. Despite everything with **Async** contexts seems to be **working good**, there is a **problem with Sync** instead.

If you can remember @trevnorris described an Idea about Continuation Local Storage: [Here Exactly](#).

### Assignees

No one assigned

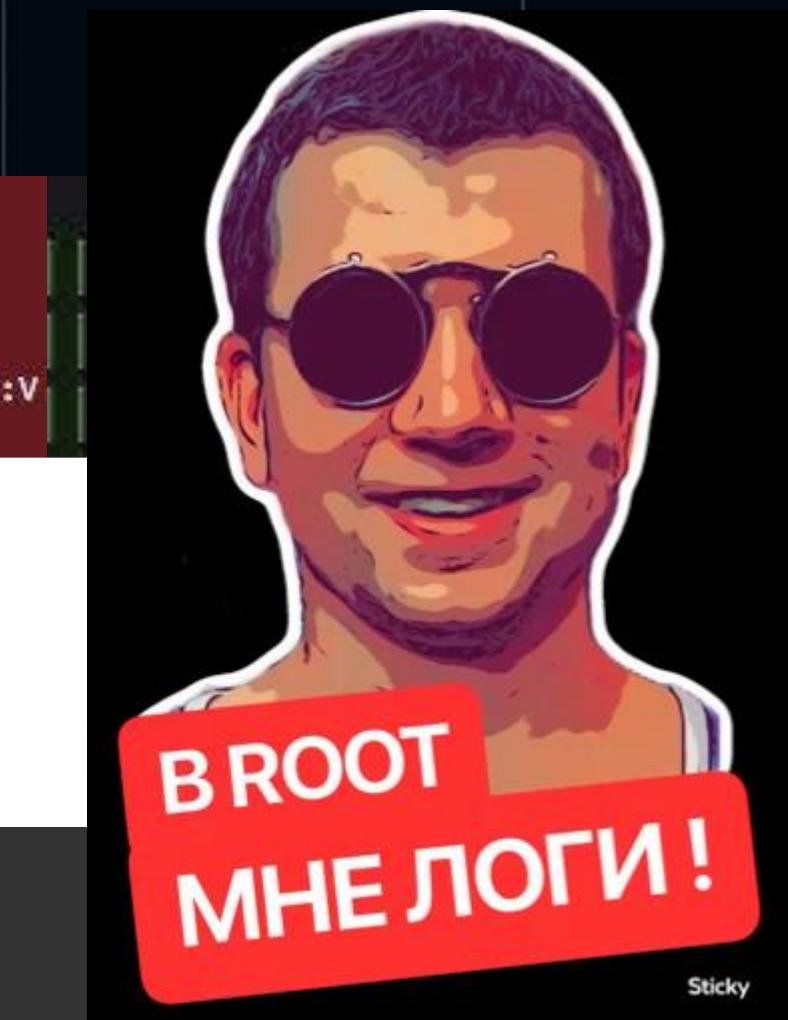
### Labels

None yet

### Projects

None yet

### Milestone



```
10|ulight | 2018-10-11 13:00 +03:00: e 10.11 13:00:52 routes/site.js:620:14 error on send flash-promo mail
11|ulight |   |
12|ulight |   | err:
13|ulight |   |   |
14|ulight |   |     |
15|ulight |   |       i18n/tr-mails/billing/flash-promo.html:43
16|ulight |   |         41| <table class="net2" align="right" style="border-spacing:0;color:#3b7694;font-family:Verdana, sans-serif;" >
```

# question about async\_hooks ➔

nodejs / diagnostics

Watch 108 Star 281 Fork 54

Code Issues 53 Pull requests 4 Projects 0 Wiki Insights

## Proposal of Polling~Queue: Sync Hook Problem #249

Closed wentout

wentout

Error: Callback was already called.

```
at /home/www/elight/node_modules/async/lib/async.js:43:36
at /home/www/elight/node_modules/async/lib/async.js:694:17
at /home/www/elight/node_modules/async/lib/async.js:173:37
at /home/www/elight/node_modules/mongoose/lib/model.js:4506:16
at model.$__save.error (/home/www/elight/node_modules/mongoose/lib/model.js:4506:16)
at /home/www/elight/node_modules/kareem/index.js:315:21
at next (/home/www/elight/node_modules/kareem/index.js:209:27)
at /home/www/elight/node_modules/kareem/index.js:182:9
at process.nextTick (/home/www/elight/node_modules/kareem/index.js:452:38)
at args.(anonymous function) (/home/www/.nvm/versions/node/v10.13.0/lib/node_modules/kareem/index.js:452:38)
at process._tickCallback (internal/process/next_tick.js:61:11)
```

Hi!

First of all, I'm sorry for my English. I'm not native speaker.

Really nice library!

And, I'm very happy to use it by myself.

Sync instances.

If you can help me, I will be very grateful.

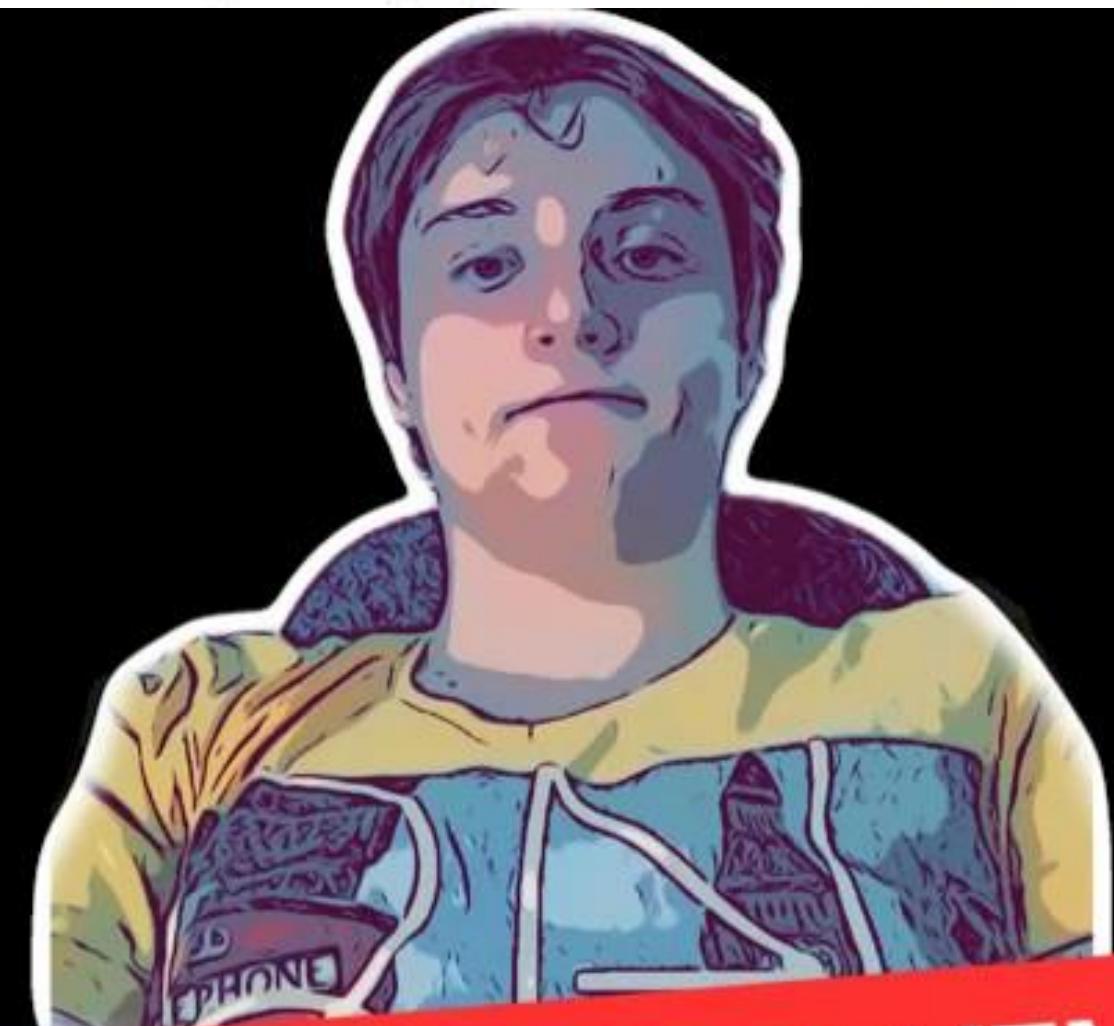
2018-11-14 13:41 +03:00: e 11.14 13:41:26 service/queue.js:69:8 QueueService |

| error:



```
1 | 2018-10-11 13:00 +03:00: e 10.11 13:00:52 routes/site.js:620:14 error on send flash-promo mail
2 |   |
3 |   |   err:
4 |   |   |
5 |   |   |   TypeError:
6 |   |   |   |
7 |   |   |   |   i18n/tr-mails/billing/flash-promo.html:43
8 |   |   |   |   |
9 |   |   |   |   41| <table class="net2"  align="right" style="border-spacing:0;color:#3b7694;font-family:V
10 |   |   |   |   erdana, sans-serif;" >
```

# question about `async_hooks`



**ИНКАСТЫЛИРУЕМ**

nodejs / diagnostics

 Code  Issues

 Issues 53  Pull requests 4

## Proposal of Polling~Queue

 **Closed** went to

wentout

Hi!

First of all

Really nice

And I'm

by myse

## Sync ins

If you can

```
Error: Callback was
      at /home/www/uli
      at /home/www/uli
      at /home/www/uli
      at /home/www/uli
      at model.$__save
      at /home/www/uli
      at next (/home/w
      at /home/www/uli
      at process.nextT
      at args.(anonymo
      at process. tick
```

```
2018-11-14 13:41 +03:00: e 11.14 13:41:26 service/queue.js:69:8 QueueService |  
| error:
```

Fork 54

## New issue

# B ROOT МНЕ ЛОГИ!

## Sticky

$$= 1$$
$$= 0$$

# context-dive

## Dive to async code with Context (v2)

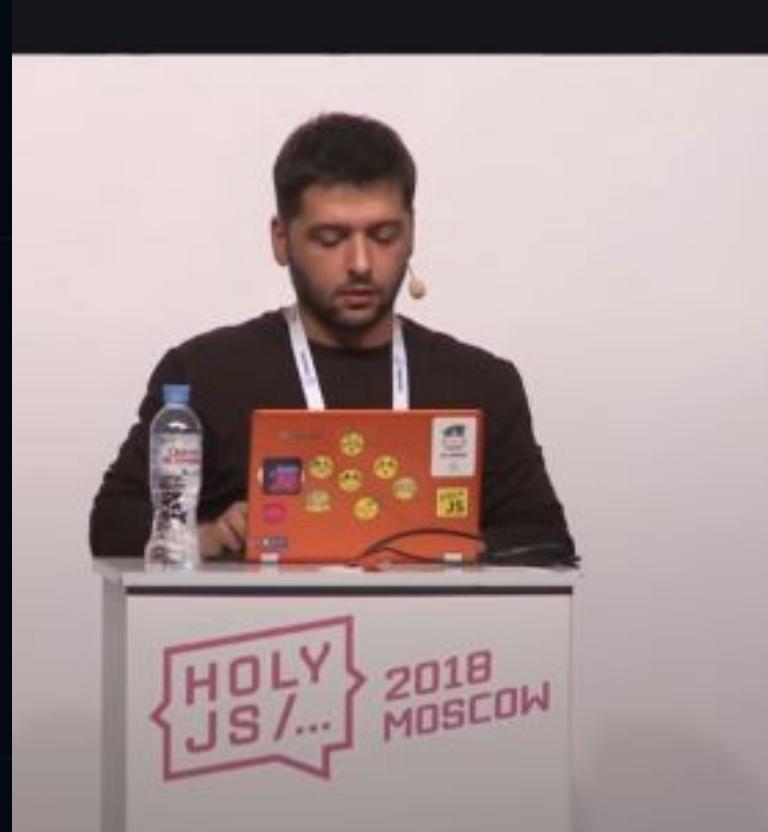
Using this module you are able to achieve the following functionality:

1. CLS/TLS : Continuation/Thread Local Storage ([wikipedia link](#)).
2. Performance measuring, based on the top of **Dived Context** we are running in.
3. Meaningful UncaughtException and UnhandledRejection extras, based on CLS.

... also there is plan to develop bit more rich functionality ...

So this module allows you to wrap some execution context context via **async\_hooks**. The main Idea is the wrapped **execution context** must be a function, cause otherwise we will be unable to handle everything through callbacks. Core concept is about that if we have some attributes of wrapped function as callbacks, so we will wrap them too, and therefore we will be able to track the context back through that callbacks too.

```
I
e Error:
  at process.uncaughtExceptionHandler (/home/went/_dev/context-dive.com/holy/mods/errors.js:6
:6)
    at process.emit (events.js:187:15)
    at process.EventEmitter.emit (domain.js:441:20)
    at process._fatalException (internal/bootstrap/node.js:622:27)
ReferenceError: b is not defined
  at module.exports (/home/went/_dev/context-dive.com/holy/routes/fail.js:13:13)
  at userModel.findOne (/home/went/_dev/context-dive.com/holy/routes/simple.js:30:21)
  at /home/went/_dev/context-dive.com/holy/node_modules/mongoose/lib/model.js:4529:16
  at process.nextTick (/home/went/_dev/context-dive.com/holy/node_modules/mongoose/lib/query.
js:2584:28)
  at process.internalTickCallback (internal/process/next_tick.js:70:11)
HTTP Server info : red socket timeout reached /
HTTP Server info : green GET /demo/find/782cab23cec52a9b9207176c09deb0e395cc63d1eaa55af4 500 5006
ms
5881 repl > printFile('routes/start_a')
{ Error: Cannot find module '../routes/start_a'
  at Function.Module._resolveFilename (internal/modules/cjs/loader.js:587:15)
  at Function.resolve (internal/modules/cjs/helpers.js:32:19)
  at printFile (/home/went/_dev/context-dive.com/holy/mods/repl.js:180:31) code: 'MODULE_NOT_
FOUND' }
5881 repl > printFile('routes/start_a')
```





# HolyJS

2019 Piter

Конференция для  
разработчиков



Мероприятие завершилось

24–25 мая

САНКТ-ПЕТЕРБУРГ





# The NodeConference for All JS Developers

## NODE CONFERENCE NL

I amsterdam

7TH JUNE 2019, THEATER DE MEERVAART, AMSTERDAM



# Speakers



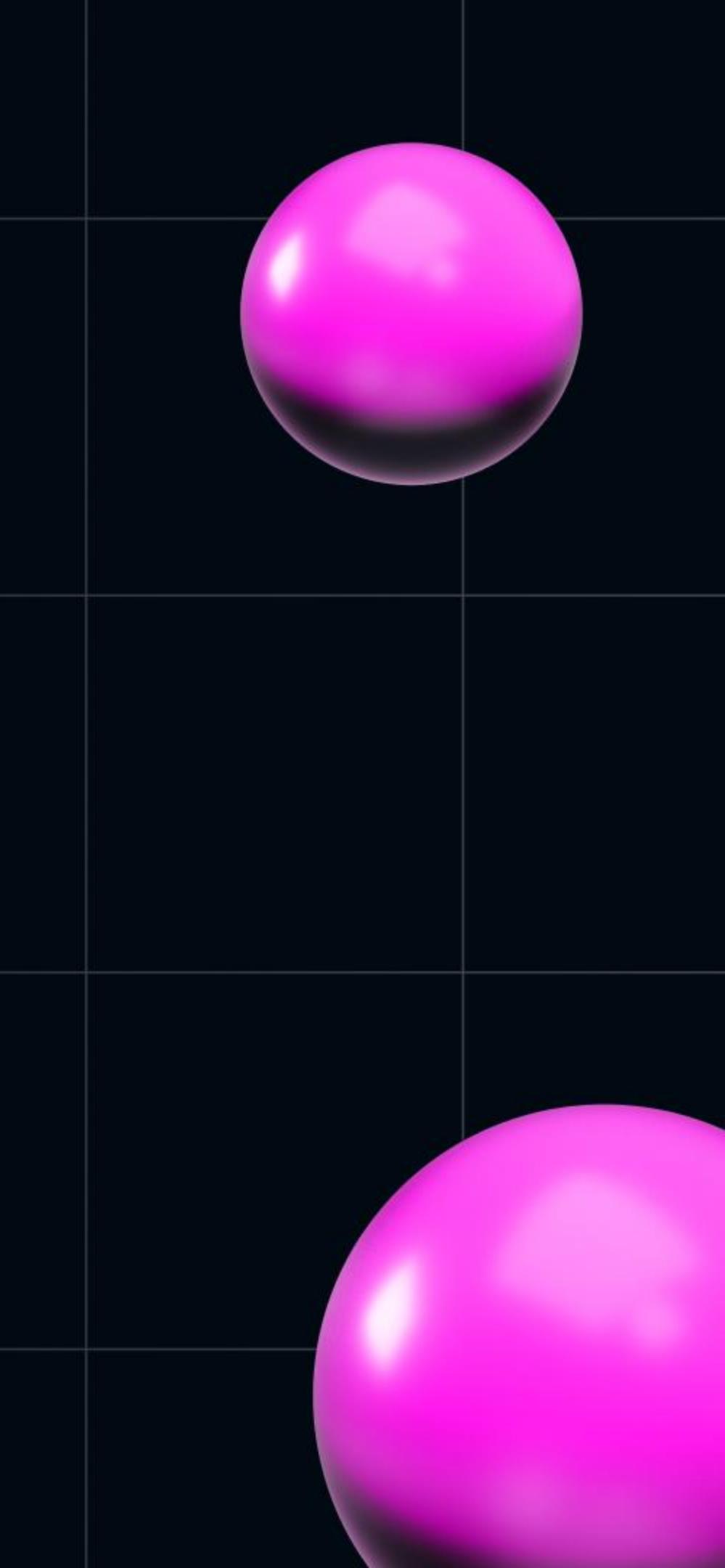
# After Party!

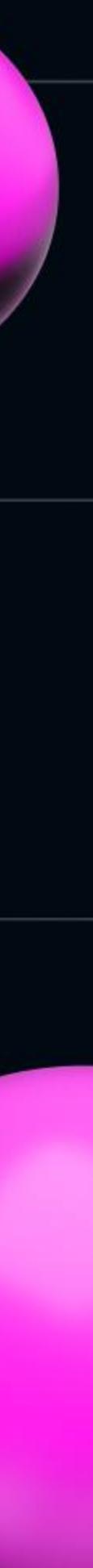
Join us up from 19:30 at Kanarie Club

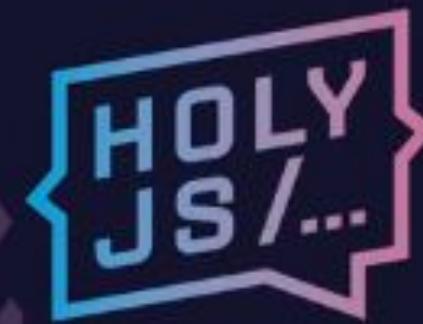
Bellamypark 51, 1053 AT Amsterdam

Direct Tram Line 17 towards Central Station  
Get off at Stop: Ten Katestraat (1 min. walking)









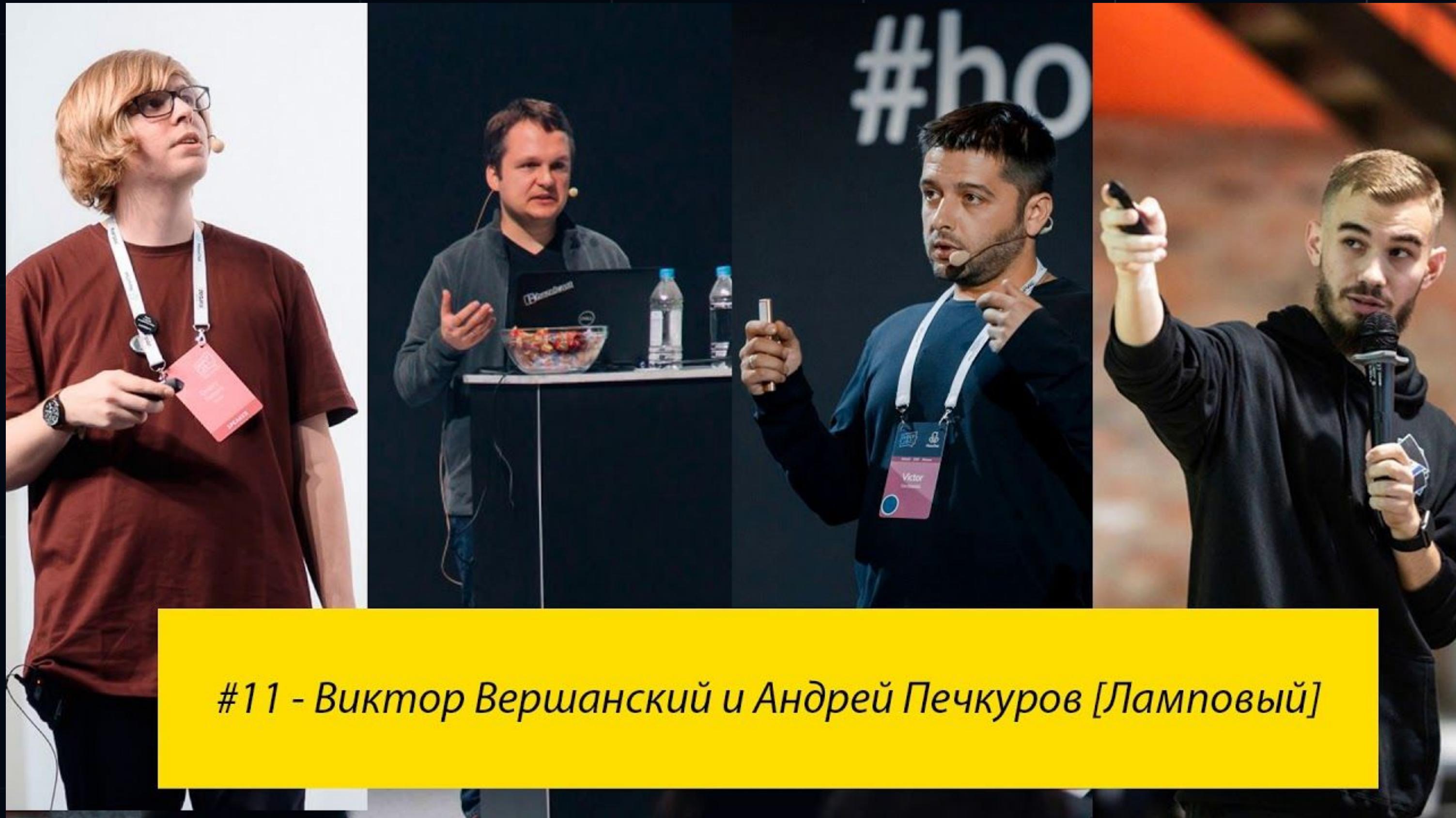
2019  
MOSCOW

# Андрей Печкуров

Hazelcast

История одной оптимизации  
производительности  
Node.js-библиотеки





#11 - Виктор Вершанский и Андрей Печкуров [Ламповый]



 Reviewed 14 pull requests in 3 repositories

[hazelcast/hazelcast](#)

[nodejs/node](#)

 [async\\_hooks: add AsyncLocal API mk2](#)

 [async-hooks: introduce async-storage API](#)

 [async\\_hooks: add executionAsyncResource](#)

[hazelcast/hazelcast-client-protocol](#)



**Andrei Pechkurov**



9 pull requests

2023

3 pull requests

2023

Dec 25

2022

Dec 21

2021

Dec 16

2021

2 pull requests

2020

2019

Dec 18

2018



Created an issue in [nodejs/diagnostics](#) that received 30 comments

## ⌚ [async hooks] proposal for standard CLS API - request for feedback

Hi guys, I believe that upcoming `executionAsyncResource()` function ([nodejs/node#30959](#)) will allow building a simple and robust CLS API as a part of a...

30 comments



C

[github.com/nodejs/node/pull/26540](https://github.com/nodejs/node/pull/26540)

## async-hooks: introduce async-storage API #26540

vdeturckheim wants to merge 1 commit into [nodejs:master](#) from [vdeturckheim:async\\_storage](#) 



**puzpuzpuz** commented on Mar 6, 2020

Any chance this will be backported to v12?

It would be great to have [AsyncLocalStorage](#) in v12, so I'm all for backporting it.

Note: [executionAsyncResource](#) ([#30959](#)) has to be backported first.

cc [@Qard](#) [@vdeturckheim](#)



2

# Class: AsyncLocalStorage

## ► History

This class creates stores that stay coherent through asynchronous operations.

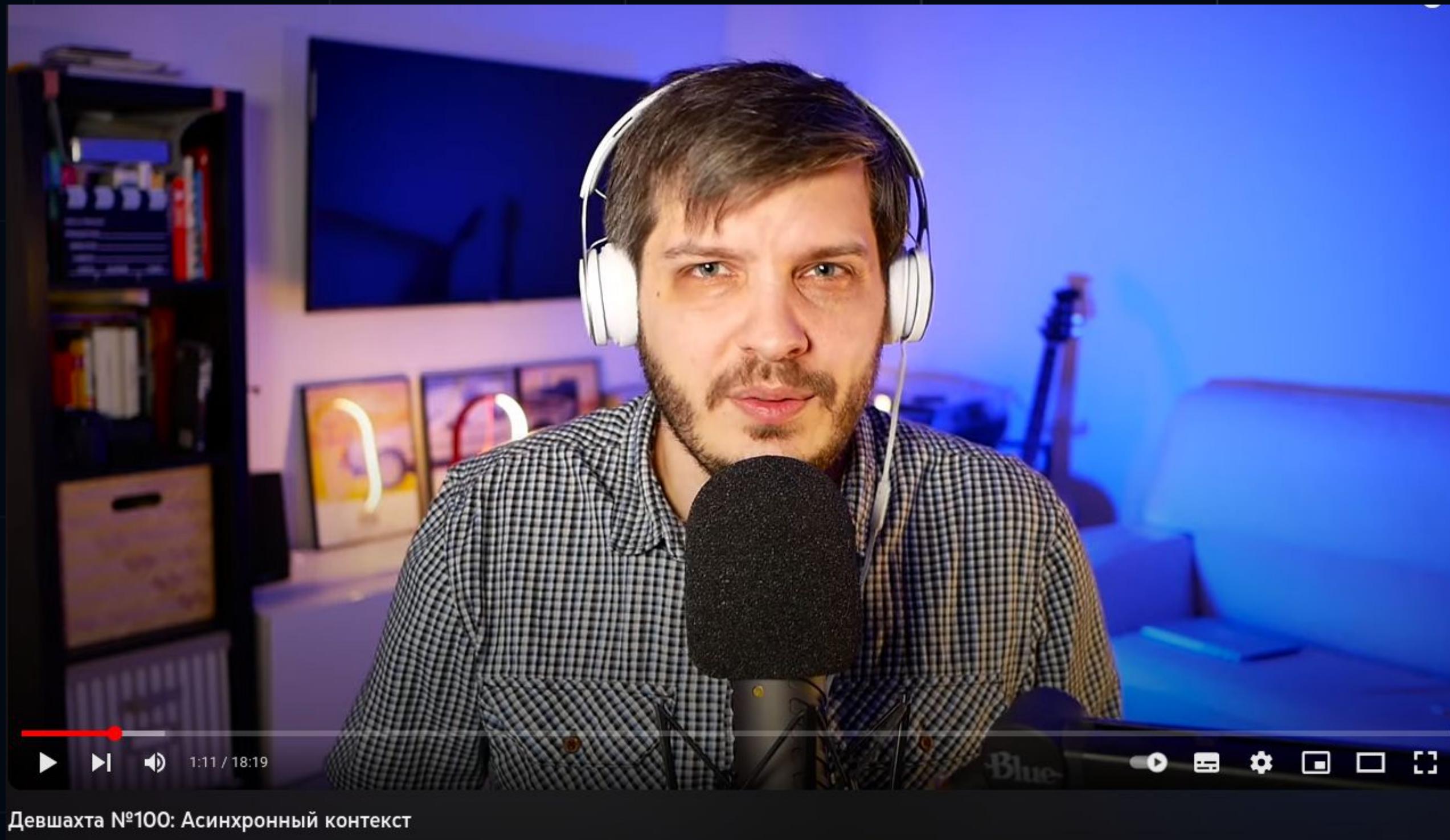
While you can create your own implementation on top of the `node:async_hooks` module, `AsyncLocalStorage` provides a performant and memory safe implementation that involves significant optimizations that are non-obvious to implement.

The following example uses `AsyncLocalStorage` to build a simple logger that assigns IDs to incoming HTTP requests and logs messages logged within each request.

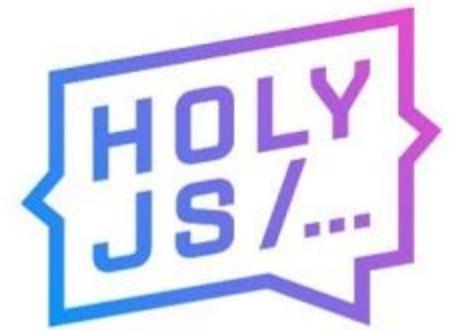
```
const http = require('node:http');
const { AsyncLocalStorage } = require('node:async_hooks');

const asyncLocalStorage = new AsyncLocalStorage();
```

# Девахта №100: Асинхронный контекст



# Алгоритмы консенсуса. При чём тут Node.js?



**Андрей Печкуров**  
Hazelcast

Алгоритмы консенсуса. При чём  
тут Node.js?



2019



**только моя задача так и  
осталась без решения**



прошлое



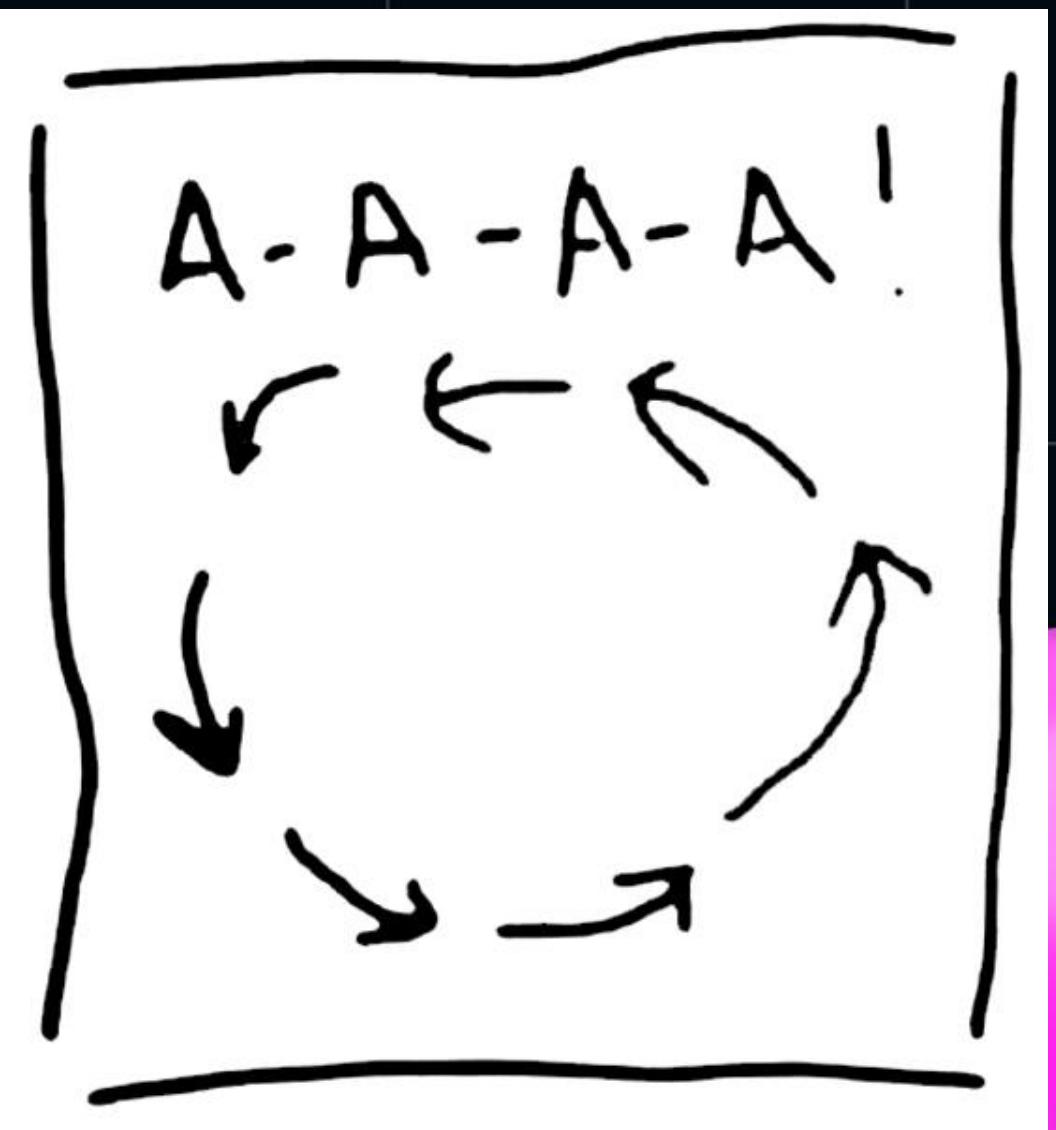
прошлое



- контекст постановки задачи
- формулировка проблематики
- **про что уже рассказывал по теме**

●

- контекст постановки задачи
- формулировка проблематики
- про что уже рассказывал по теме





BrendanEich ✅

@BrendanEich

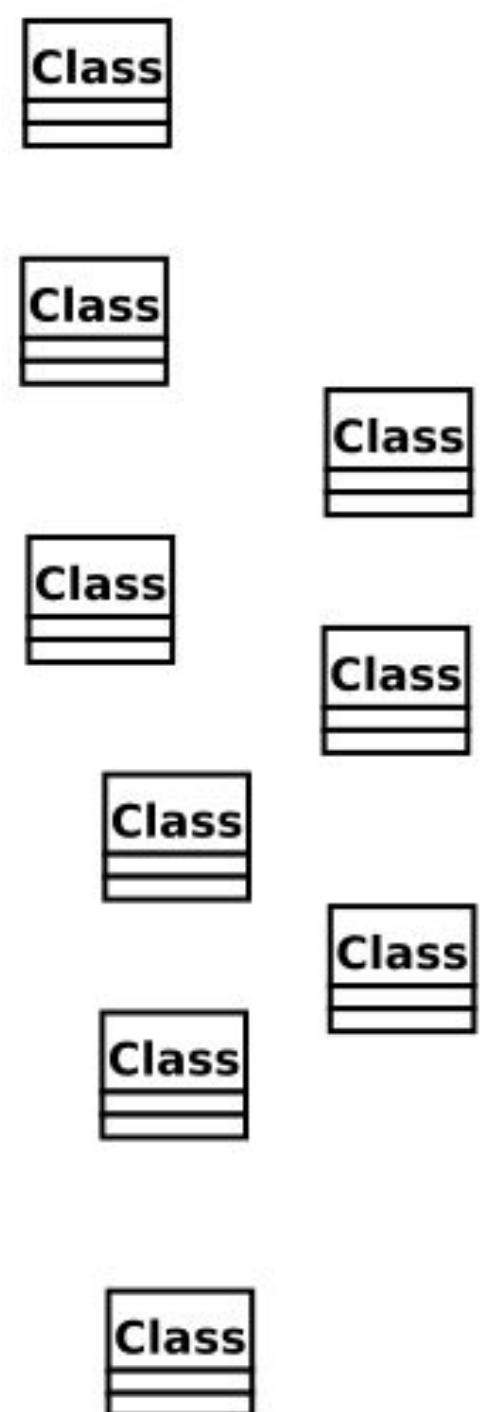
Replying to [@BrendanEich](#) [@rauschma](#) and [@IndieScripter](#)

If I didn't have "Make it look like Java" as an order from management, \*and\* I had more time (hard to unconfound these two causal factors), then I would have preferred a Self-like "everything's an object" approach: no Boolean, Number, String wrappers. No undefined and null. Sigh.

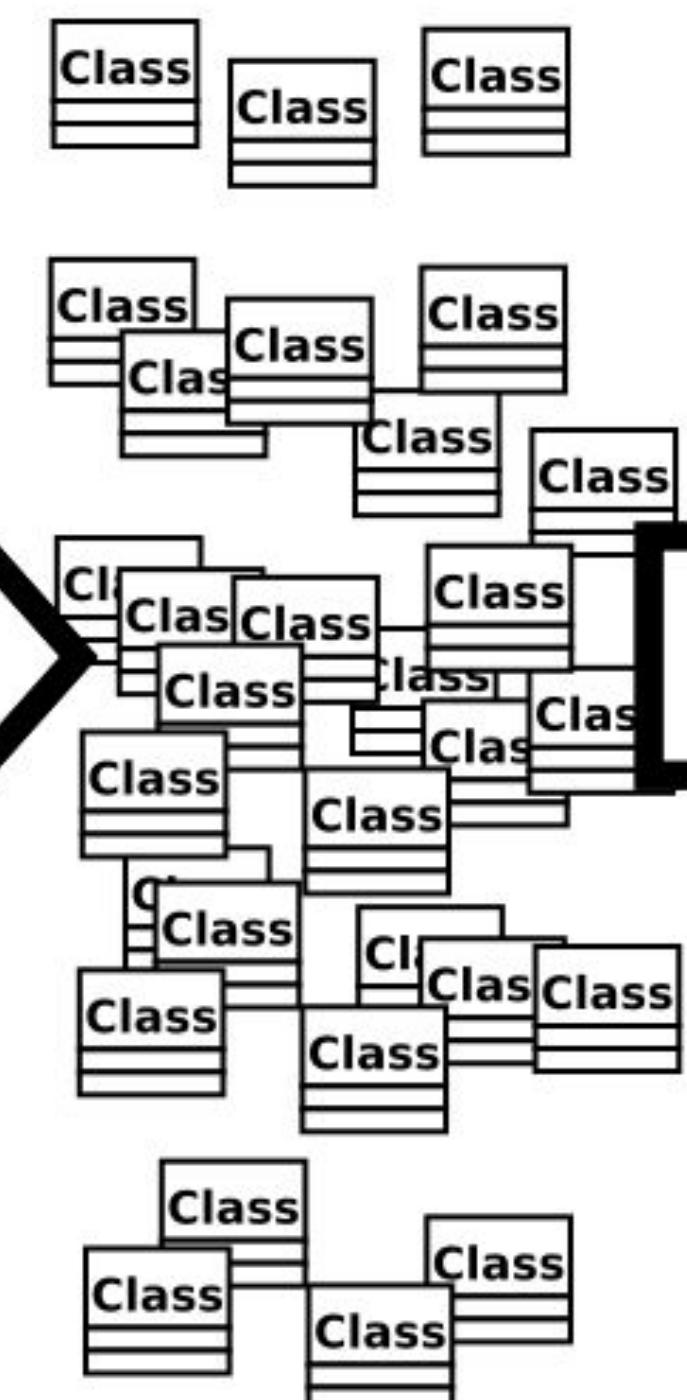
**1 day**



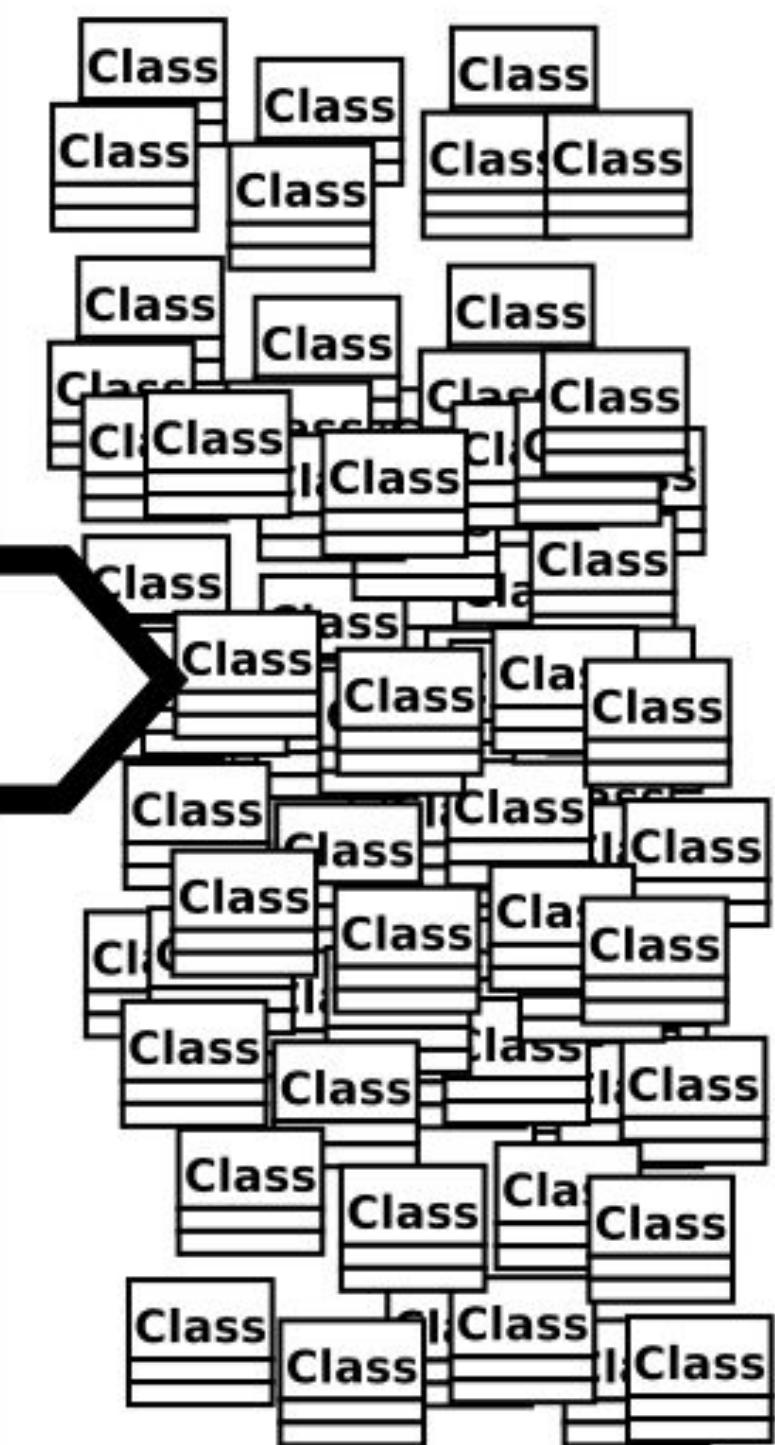
**2 days**



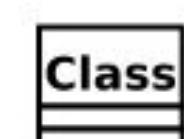
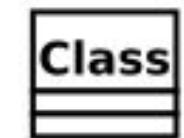
**3 days**



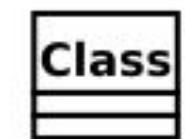
**3 month**



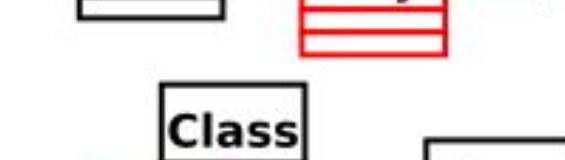
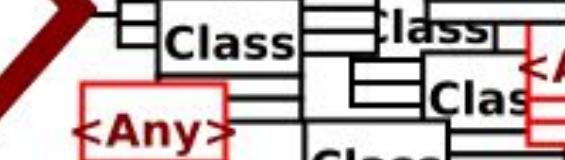
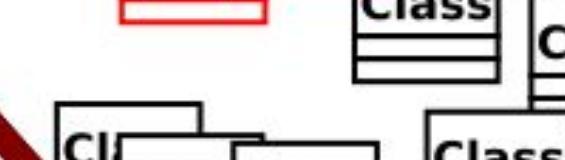
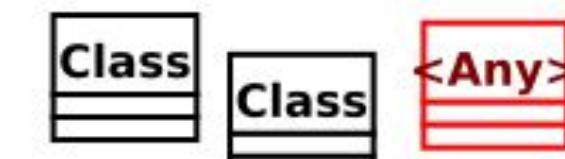
**1 day**



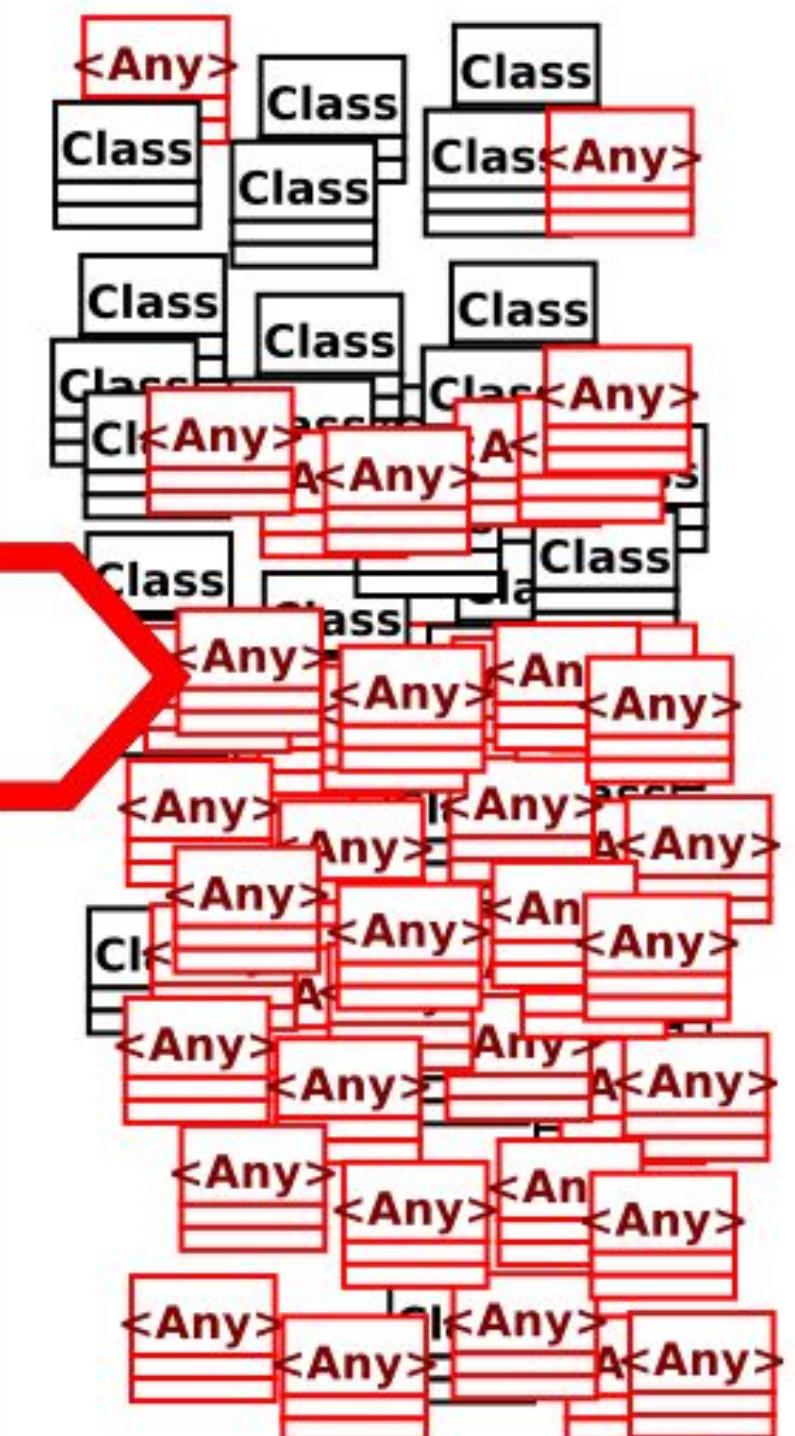
**2 years**



**3 years**



**N years**





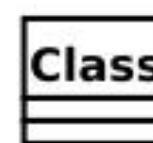
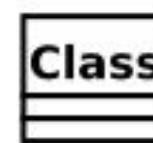
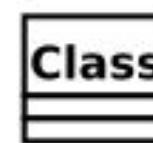
BrendanEich ✅  
@BrendanEich

и всё это объекты

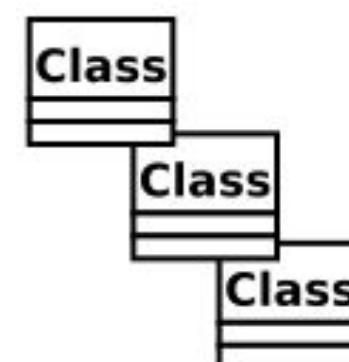
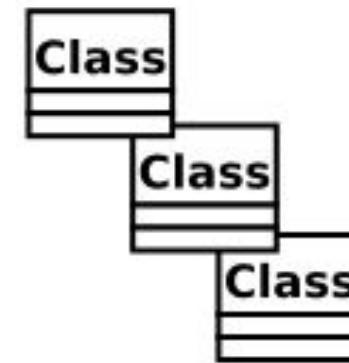
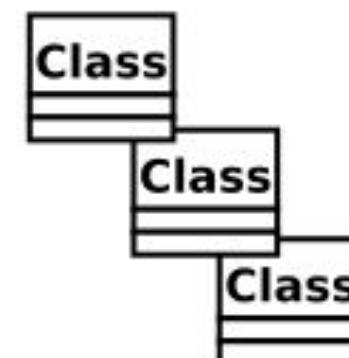
Replying to @BrendanEich @rauschma and @IndieScripter

If I didn't have "Make it look like Java" as an order from management, \*and\* I had more time (hard to unconfound these two causal factors), then I would have preferred a Self-like "everything's an object" approach: no Boolean, Number, String wrappers. No undefined and null. Sigh.

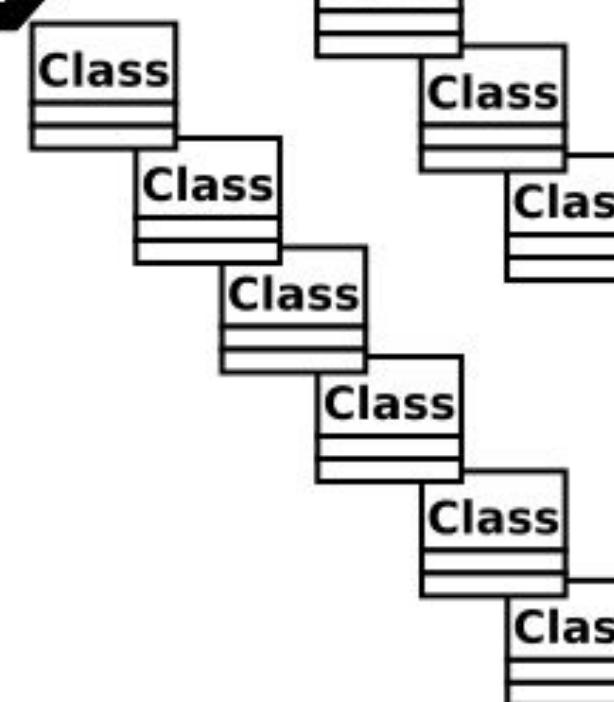
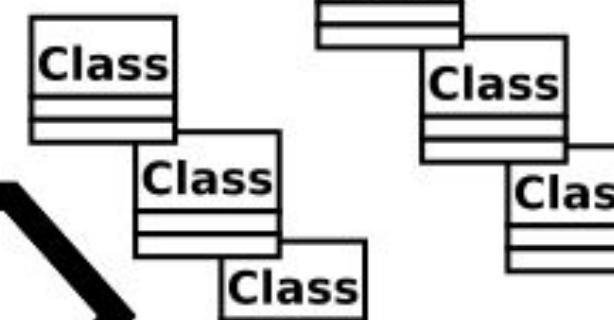
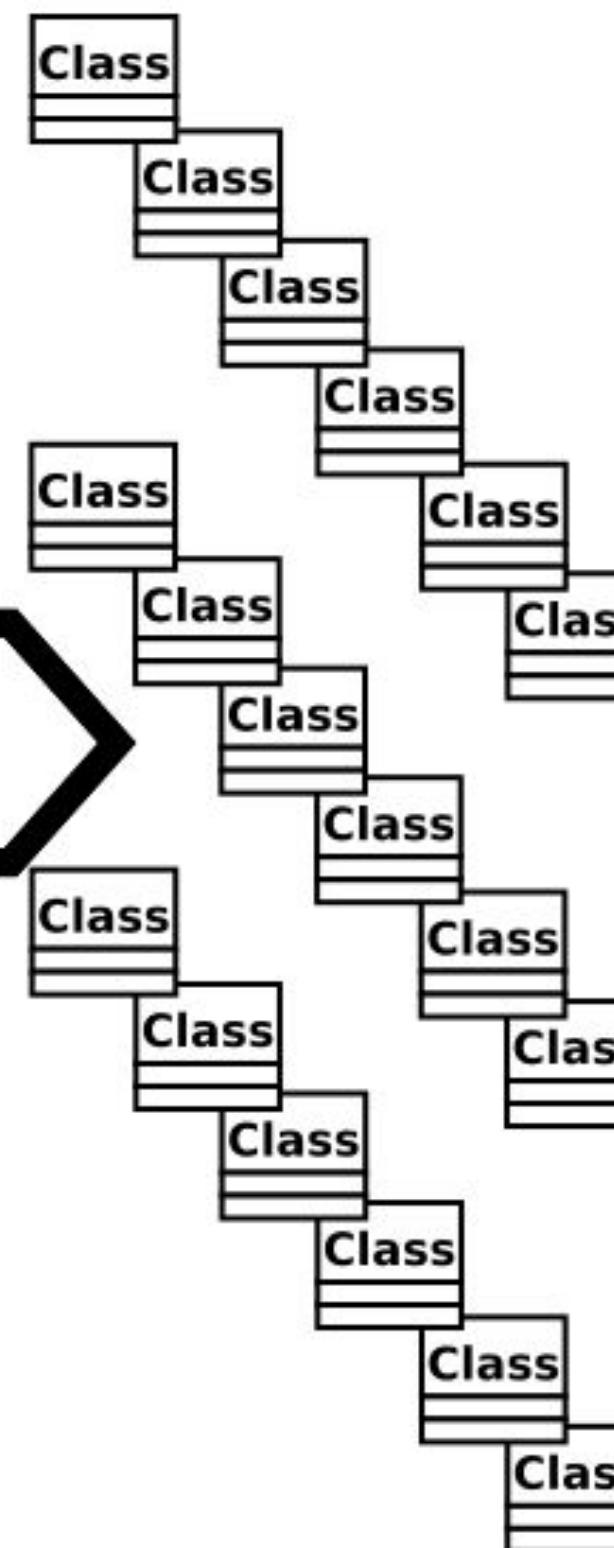
1 year



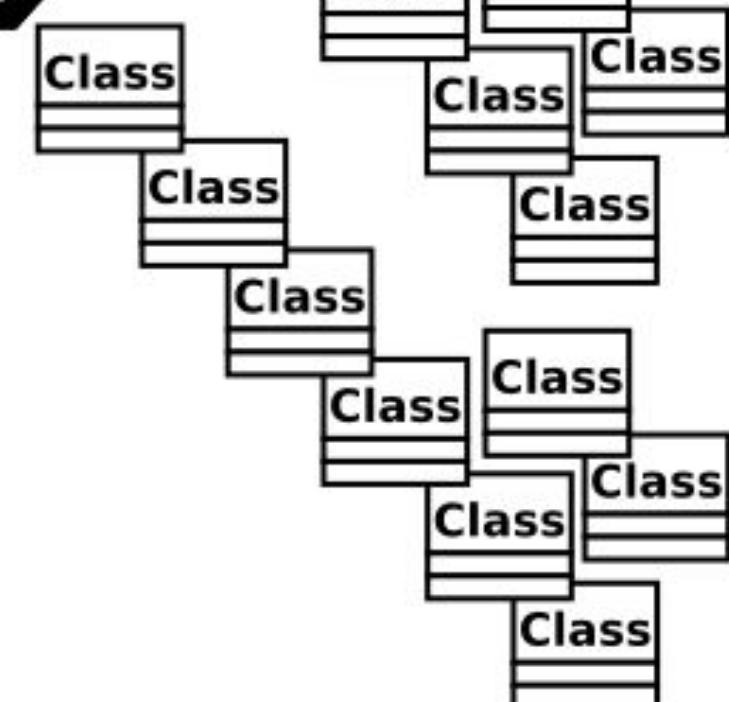
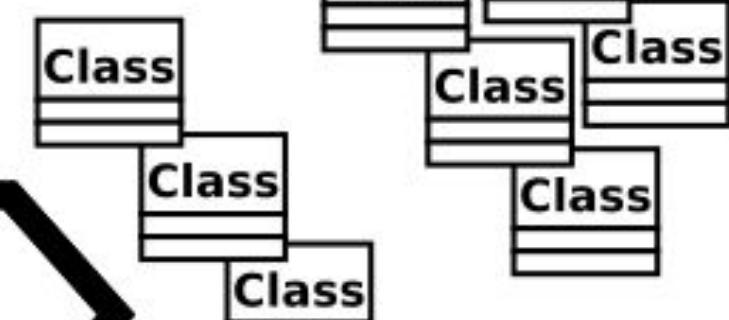
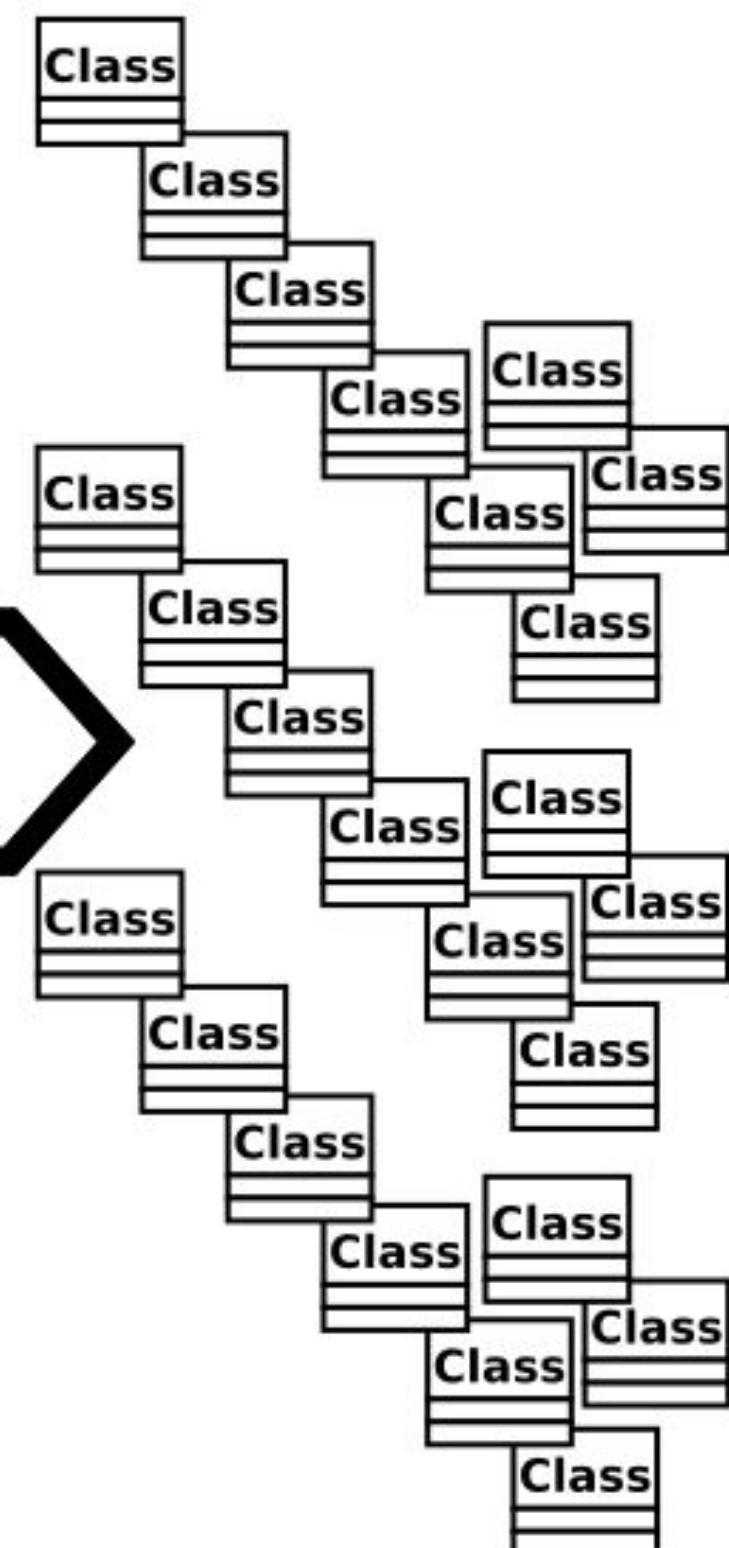
2 years



3 years



N years



> next

< ▼ *MyConstructor {state: 3}* **i**

state: 3

▼ \_\_proto\_\_:

state: 2

▼ \_\_proto\_\_:

state: 1

► \_\_proto\_\_: Object

**так как у меня уже был  
опыт с декораторами**



# context-dive

## Dive to async code with Context (v2)

Using this module you are able to achieve the following functionality:

1. CLS/TLS : Continuation/Thread Local Storage ([wikipedia link](#)).
2. Performance measuring, based on the top of **Dived Context** we are running in.
3. Meaningful UncaughtException and UnhandledRejection extras, based on CLS.

... also there is plan to develop bit more rich functionality ...

So this module allows you to wrap some execution context context via **async\_hooks**. The main Idea is the wrapped **execution context** must be a function, cause otherwise we will be unable to handle everything through callbacks. Core concept is about that if we have some attributes of wrapped function as callbacks, so we will wrap them too, and therefore we will be able to track the context back through that callbacks too.

# context-dive

## into execution flow

### Dive to async code with Context (v2)

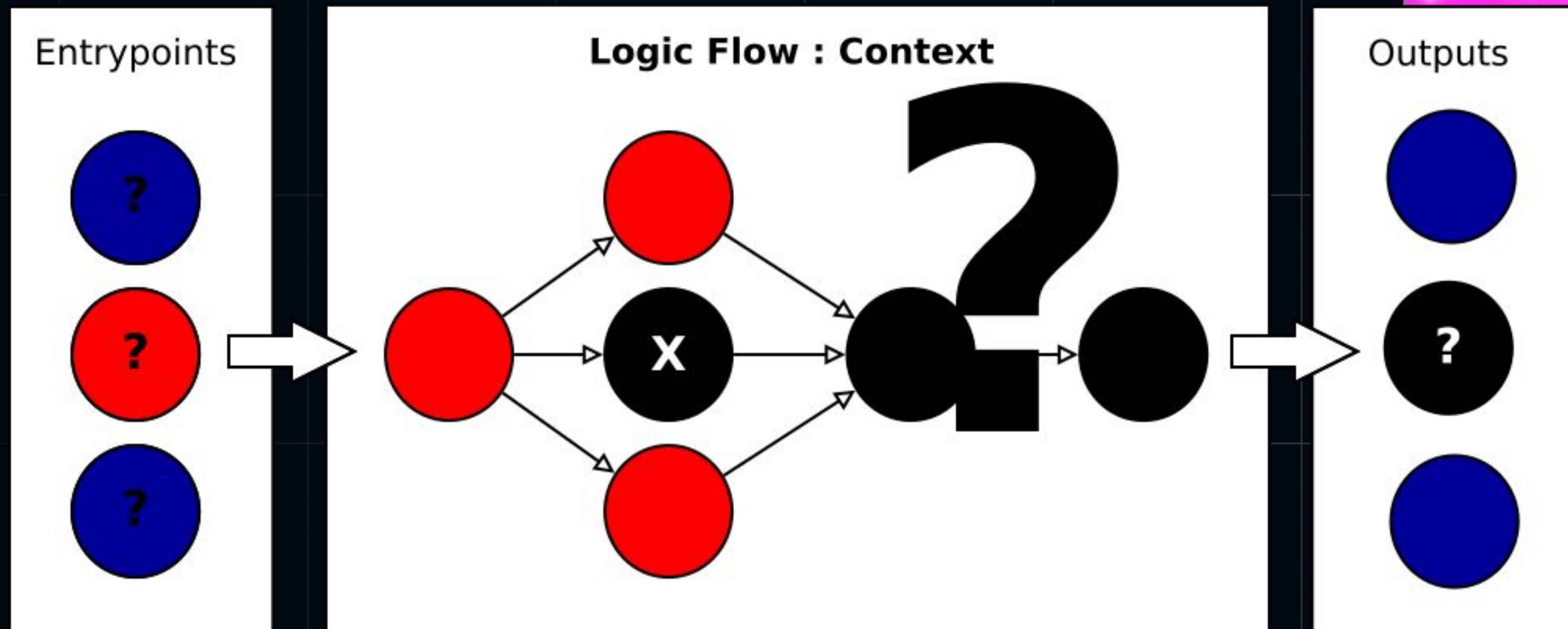
Using this module you are able to achieve the following functionality:

1. CLS/TLS : Continuation/Thread Local Storage ([wikipedia link](#)).
2. Performance measuring, based on the top of **Dived Context** we are running in.
3. Meaningful UncaughtException and UnhandledRejection extras, based on CLS.

... also there is plan to develop bit more rich functionality ...

So this module allows you to wrap some execution context context via **async\_hooks**. The main Idea is the wrapped **execution context** must be a function, cause otherwise we will be unable to handle everything through callbacks. Core concept is about that if we have some attributes of wrapped function as callbacks, so we will wrap them too, and therefore we will be able to track the context back through that callbacks too.

# В различных по потоках данных



**решил упорядочить  
не потоки данных**



решил упорядочить  
не потоки данных  
а сами данные



решил упорядочить  
не потоки данных  
а сами данные



при помощи  
Prototype Chain





# Inheritance in JavaScript : Factory of Constructors with Prototype Chain : point of view from boring nerd

#javascript #inheritance #prototype



we

Feb 25 Originally published at Medium • 18 min read

Edit

## Manage



went

JavaScript enthusiast. Computerman.  
Teapot Systems Administrator.

## Edit profile

JOINED

Jan 8, 2020

## More from went

Dead Simple type checker for JavaScript

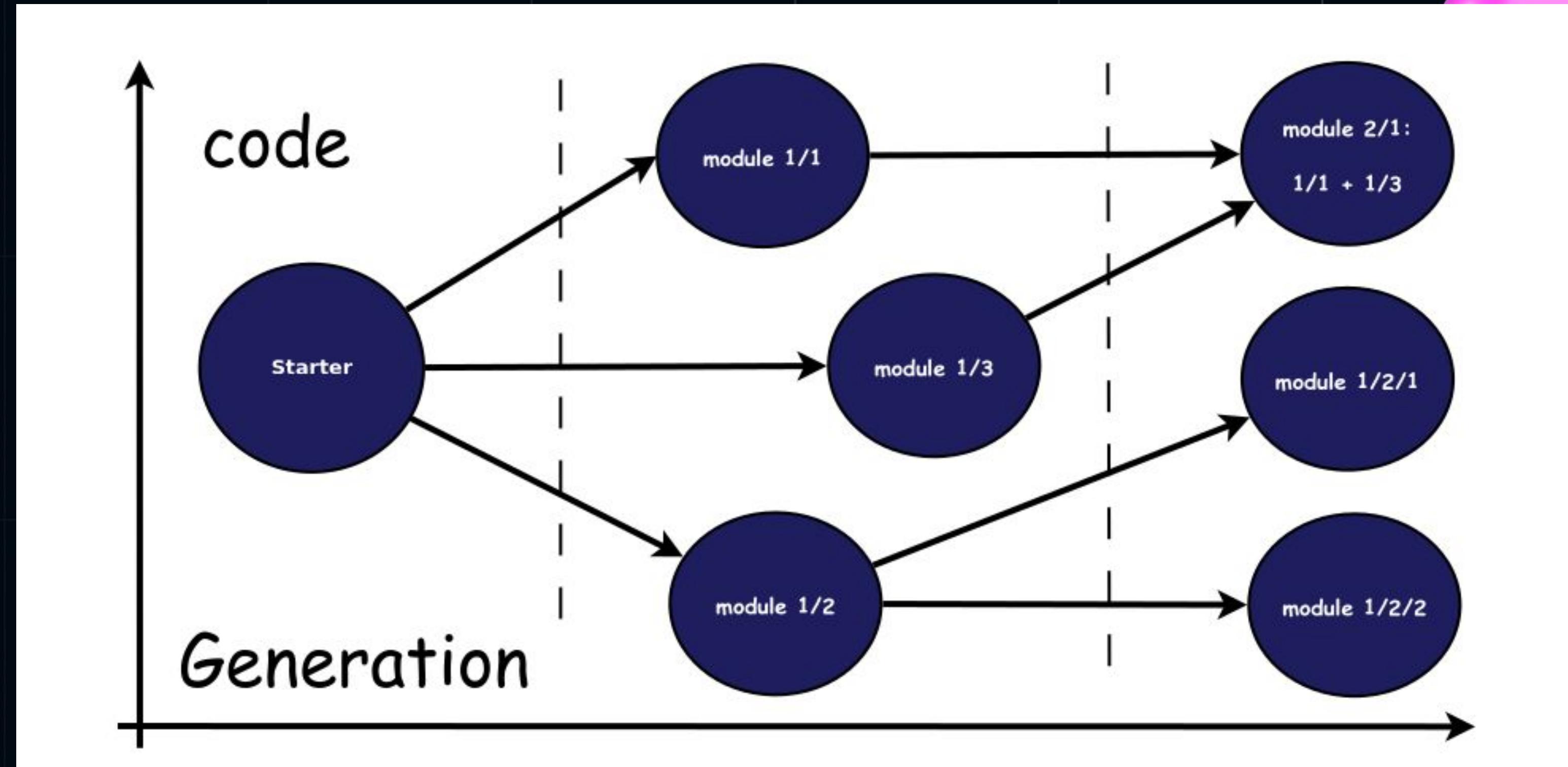
#javascript #typechek #inritance

# Architecture of Prototype Inheritance in JavaScript

#inheritance #architecture  
#javascript

при помощи  
Prototype Chain





# mnemonica TS

0.9.955 • Public • Published 3 days ago

[Readme](#)

[Code](#)

Beta

[0 Dependencies](#)

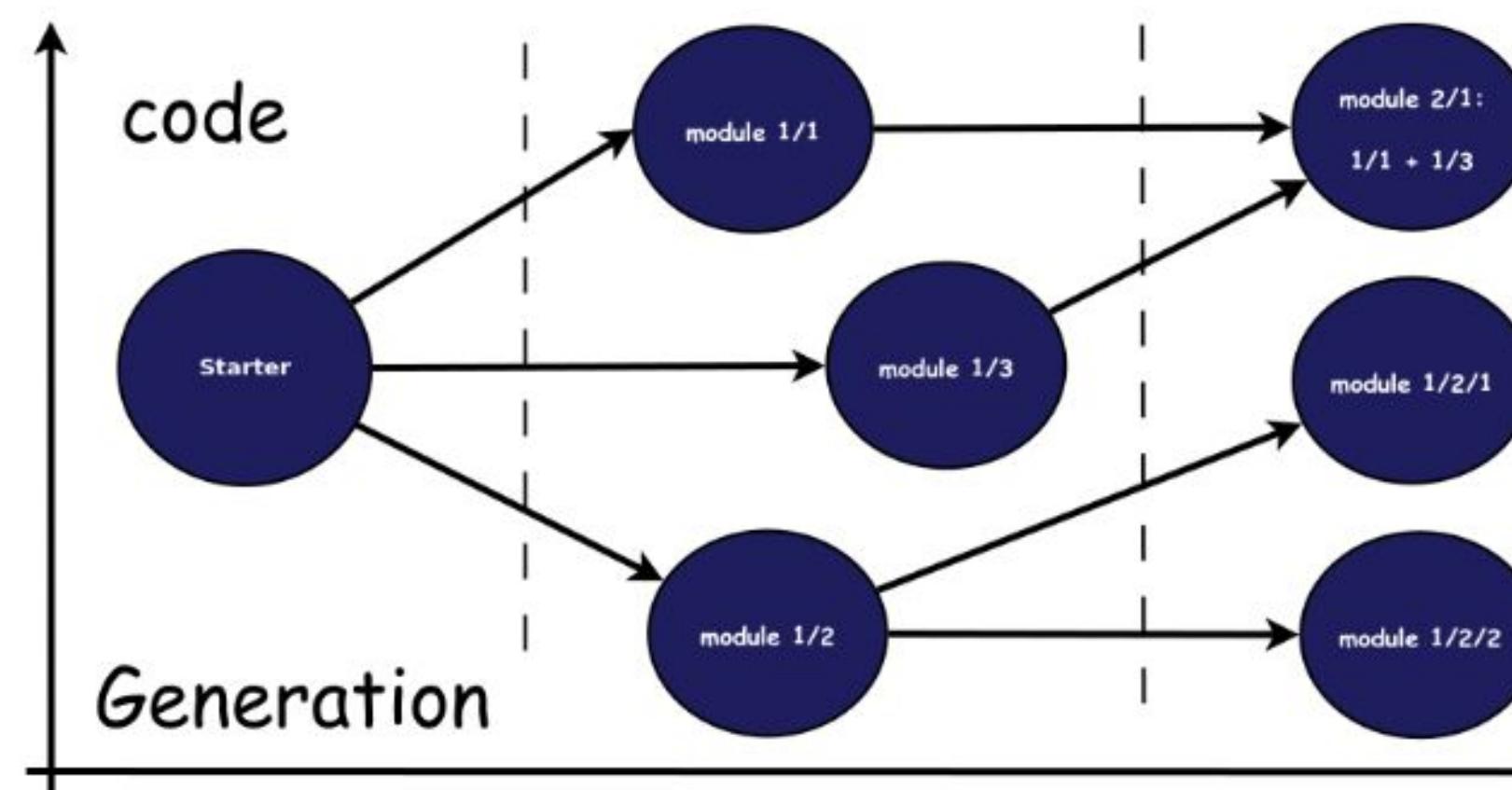
[0 Dependents](#)

[136 Versions](#)

## mnemonica is

abstract technique that aids information retention : instance inheritance system

... allows us to make inherited descriptions of mappings of transformations from predecessor structured data types to the successors, as if it was math  $f(x) \Rightarrow y$  ... and we will use this keyword as a persistent data structure where we will apply that transformations



Install

`> npm i mnemonica`



Repository

[github.com/wentout/mnemonica](https://github.com/wentout/mnemonica)

Homepage

[github.com/wentout/mnemonica#read...](https://github.com/wentout/mnemonica#readme)

Weekly Downloads

26



Version

0.9.955

License

MIT

Unpacked Size

120 kB

Total Files

80

Issues

0

Pull Requests

0



# Chronotope: await Eloquent.Errors

## Приглашенные эксперты



Дмитрий Махнёв  
JetBrains

## Спикеры

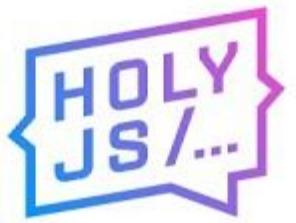


Виктор Вершанский

2020



# Оптимизация синхронной асинхронности



2021 PITER

**Дмитрий Махнёв**

JetBrains

Оптимизация синхронной  
асинхронности



**typeomatica** 

0.3.31 • Public • Published 21 days ago

 [Readme](#)

 [Code](#) Beta

 [0 Dependencies](#)

 [0 Dependents](#)

 [23 Versions](#)

# Type ø matica

 coverage **100%**

license **MIT** version **v0.3.31** last commit **october**

**\$ npm install typeomatica**

This package is a part of **mnemonica** project.

Strict Types checker for objects which represent Data Types.

## how it works

see `test/index.ts`

```
class SimpleBase extends BasePrototype {  
    stringProp = '123';  
};
```

Install

 `> npm i typeomatica`

Repository

 [github.com/wentout/typeomatica](https://github.com/wentout/typeomatica)

Homepage

 [github.com/wentout/typeomatica#read...](https://github.com/wentout/typeomatica#read...)

 Weekly Downloads



Version

**0.3.31**

License

**MIT**

Unpacked Size

**49.2 kB**

Total Files

**42**

# Строгая типизация в JavaScript



2021 PITER

**Виктор Вершанский**

DataArt

Strict Types in JavaScript



2021





**КОД**



код



- контекст постановки задачи
- формулировка проблематики
- про что уже рассказывал по теме
- **как создаётся код для решения**

- контекст постановки задачи
- формулировка проблематики
- про что уже рассказывал по теме
- **как создаётся код для решения**



```
declare function remapKeys  
  <  
    T extends number,  
    RR extends string,  
    O extends Record<string, T>,  
    R extends Record<string, RR>  
  >  
(obj: O, remap: R): {  
  [K in keyof O as K extends keyof R ? R[K] : K]: O[K]  
}  
  
var res = remapKeys({ obj: 1, a: 2 }, { obj: 'newObj' })
```

```
declare function remapKeys  
  <  
    T extends number,  
    RR extends string,  
    O extends Record<string, T>,  
    R extends Record<string, RR>  
  >  
(obj: O, remap: R): {  
  [K in keyof O as K extends keyof R ? R[K] : K]: O[K]  
}  
  
var res = remapKeys({ obj: 1, a: 2 }, { obj: 'newObj' })
```



- контекст постановки задачи
- формулировка проблематики
- про что уже рассказывал по теме
- **как создаётся код для решения**



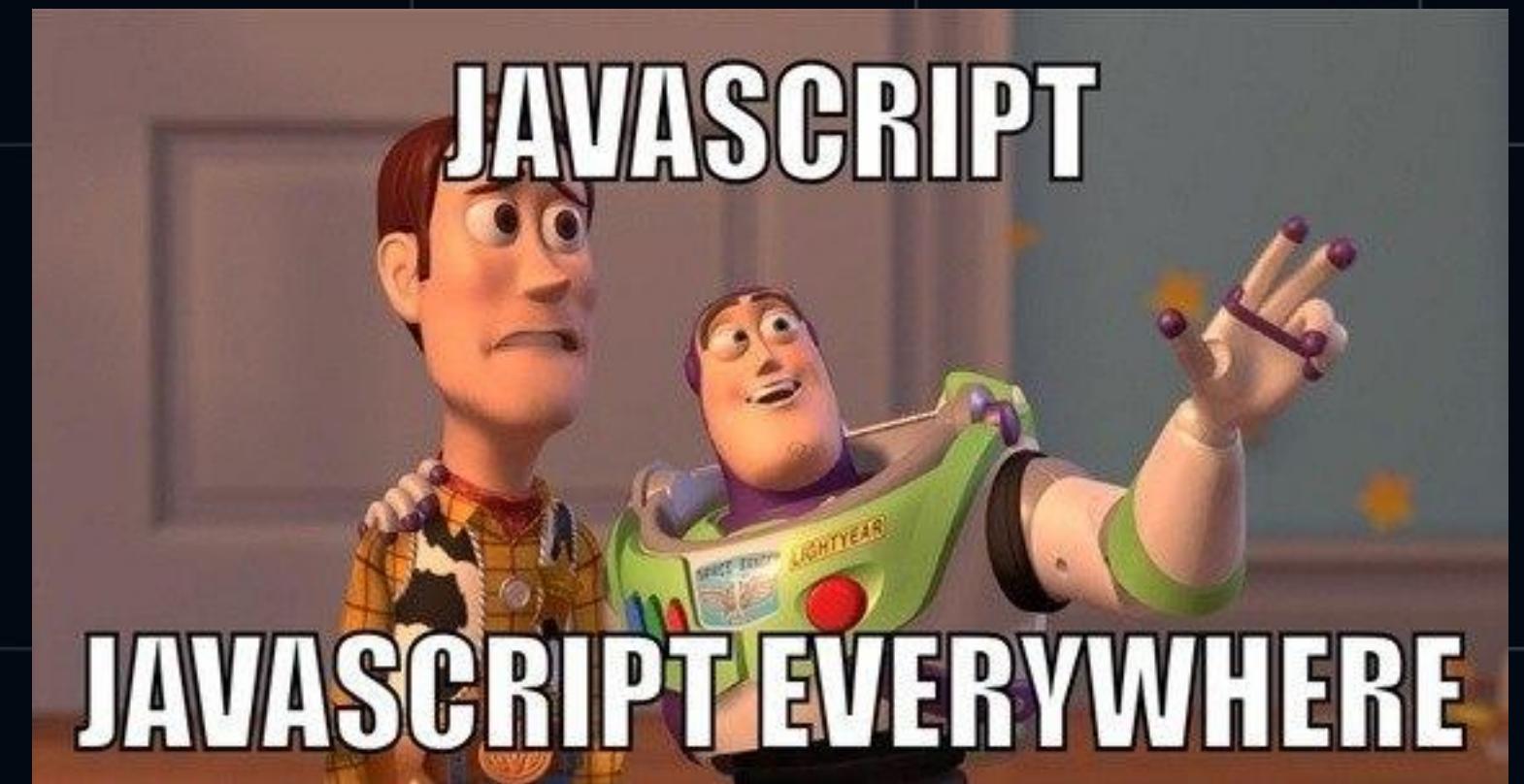
Ща напишу



**для начала нужно типизировать  
функциональный конструктор**



- контекст постановки задачи
- формулировка проблематики
- про что уже рассказывал по теме
- **как создаётся код для решения**



```
1 type Proto<P, T> = Pick<P, Exclude<keyof P, keyof T>> & T;
2
3 const CstrFn = function <R>(param: R) {
4   Object.assign(this, param);
5 } as {
6   (): void
7   new <
8     T extends object,
9     S extends Proto<T, typeof CstrFn.prototype>>(param: T): {
10    [key in keyof S]: S[key]
11  }
12  prototype: {
13    m: number
14  }
15 };
16 CstrFn.prototype.m = 2
17
18 const s = new CstrFn({ m: '1', s: 1 });
19 console.log(s);
```

```
1 type Proto<P, T> = Pick<P, Exclude<keyof P, keyof T>> & T;
2
3 const CstrFn = function <R>(param: R) {
4   Object.assign(this, param);
5 } as {
6   (): void
7   new <
8     T extends object,
9     S extends Proto<T, typeof CstrFn.prototype>>(param: T): {
10    [key in keyof S]: S[key]
11  }
12  prototype: {
13    m: number
14  } const s: {
15    s: number;
16    m: number;
17  }
18 const s = new CstrFn({ m: '1', s: 1 });
19 console.log(s);
```



**то есть да,  
такая вот простиныя**



```
1 type Proto<P, T> = Pick<P, Exclude<keyof P, keyof T>> & T;
2
3 const CstrFn = function <R>(param: R) {
4   Object.assign(this, param);
5 } as {
6   (): void
7   new <
8     T extends object,
9     S extends Proto<T, typeof CstrFn.prototype>>(param: T): {
10    [key in keyof S]: S[key]
11  }
12  prototype: {
13    m: number
14  }
15 };
16 CstrFn.prototype.m = 2
17
18 const s = new CstrFn({ m: '1', s: 1 });
19 console.log(s);
```

**но она в самом деле нужна**



```
1 type Proto<P, T> = Pick<P, Exclude<keyof P, keyof T>> & T;
2
3 const CstrFn = function <R>(param: R) {
4   Object.assign(this, param);
5 } as {
6   (): void
7   new <
8     T extends object,
9     S extends Proto<T, typeof CstrFn.prototype>>(param: T): {
10    [key in keyof S]: S[key]
11  }
12  prototype: {
13    m: number
14  } const s: {
15    s: number;
16    m: number;
17  }
18 const s = new CstrFn({ m: '1', s: 1 });
19 console.log(s);
```



**и ещё инструментарий:**



**mnemonica** TS

0.9.955 • Public • Published 3 days ago

[Readme](#)

[Code](#)

Beta

[0 Dependencies](#)

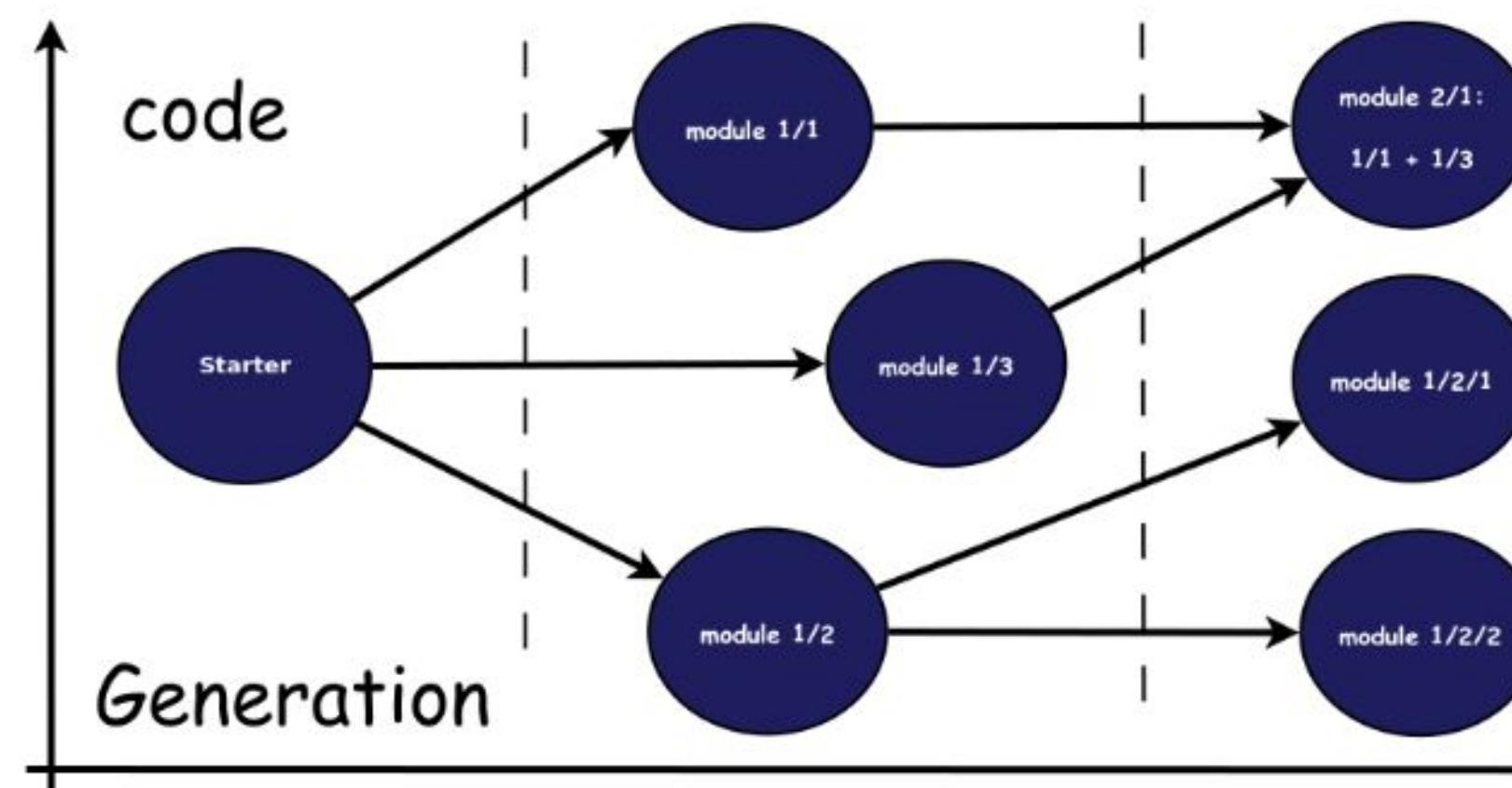
[0 Dependents](#)

[136 Versions](#)

## mnemonica is

abstract technique that aids information retention : instance inheritance system

... allows us to make inherited descriptions of mappings of transformations from predecessor structured data types to the successors, as if it was math  $f(x) \Rightarrow y$  ... and we will use this keyword as a persistent data structure where we will apply that transformations



Install

`> npm i mnemonica`



Repository

[github.com/wentout/mnemonica](https://github.com/wentout/mnemonica)

Homepage

[github.com/wentout/mnemonica#read...](https://github.com/wentout/mnemonica#readme)

Weekly Downloads

26



Version

0.9.955

License

MIT

Unpacked Size

120 kB

Total Files

80

Issues

0

Pull Requests

0

TS test-example.ts X

core > test-ts > TS test-example.ts > ...

```
1 import { define } from '..';
2
3 const FirstType = define('SomeType', function (this: {
4   first: 'FirstType',
5 }) {
6   this.first = 'FirstType';
7 }
8
9 const SecondType = FirstType.define('SecondType', function (this: {
10   first: undefined,
11   second: string,
12 }) {
13   this.first = undefined;
14   this.second = 'SecondType';
15 }) ;
```

TS test-example.ts ×

core > test-ts > TS test-example.ts > ...

```
16
17  const first = new FirstType();
18
19  type TSecondInstance =
20  | InstanceType<typeof SecondType>;
21
22  const second = new first.SecondType() as TSecondInstance;
23
24  // { first: undefined, second: "SecondType" }
25  console.log(second);
26
27
```



что сейчас



что сейчас

5

- контекст постановки задачи
- формулировка проблематики
- про что уже рассказывал по теме
- как создаётся код для решения
- **как развивается решение: что сделано**



2023



2023



# Типы в прототипах



Виктор  
Вершанский



о чём там речь



# о чём там речь

Issue Type \*  Bug ?

Some issue types are unavailable due to incompatible field configuration and/or workflow associations.

Main Additional

Summary \* Злободневный баг

Priority ?

Component/s

Assignee

Story Points

Нормально  
Очень срочно  
Очень важно  
Важно  
Когда-нибудь  
Маловажное возможное улуч...

press down to select.

TS mixWithProto.ts X

□ ...

talks > 2023-05-HolyJS > examples > TS mixWithProto.ts > ...

```
1 type init = {
2   s: number
3   z: number
4 }
5
6 type next = {
7   s?: string
8   m: boolean
9 }
10
```

TS mixWithProto.ts X

...

talks > 2023-05-HolyJS > examples > TS mixWithProto.ts > ...

```
22 type unit = proto & next
23
24 const aggregation: unit = {
25   z: 123,
26   s: 'x',
27   m: true,
28 };
29   type sss = string | undefined
30 type sss = typeof aggregation.s
31
```

TS mixWithProto.ts X

□ ...

talks > 2023-05-HolyJS > examples > TS mixWithProto.ts > ...

```
1 type init = {
2   s: number
3   z: number
4 }
5
6 type next = {
7   s?: string
8   m: boolean
9 }
10
```

TS mixWithProto.ts X

□ ...

talks > 2023-05-HolyJS > examples > TS mixWithProto.ts > ...

```
10
11   type proto = {
12     z: number;
13   }
14   type proto = Pick<
15     init,
16     Exclude<
17       keyof init,
18       keyof next
19     >>
```

ts function\_construct\_typed.ts ×

examples > ts function\_construct\_typed.ts > ...

```
1  function OtherConstruct(this: { field: number }) {}  
2  OtherConstruct.prototype = {  
3    otherField: true  
4  }  
5  
6  type Proto<P, T> = Pick<P, Exclude<keyof P, keyof T>> & T;  
7  
8  const define = function <P extends object, T>(Cstr: { (this: T): void }, proto: P) {  
9    const MyConstructor = function (): Proto<P, T> {  
10      return new Cstr;  
11    };  
12    Object.setPrototypeOf(MyConstructor.prototype, proto);  
13    return MyConstructor;  
14  };  
15  
16  const myConstruct = define(OtherConstruct, { otherField: true });  
17  
18  const myConstructedItem = myConstruct();  
19  
20  console.log(myConstructedItem);  
21
```

ts function\_construct\_typed.ts ×

examples > ts function\_construct\_typed.ts > ...

```
1  function OtherConstruct(this: { field: number }) {}  
2  OtherConstruct.prototype = {  
3    otherField: true  
4  }  
5  
6  type Proto<P, T> = Pick<P, Exclude<keyof P, keyof T>> & T;  
7  
8  const define = function <P extends object, T>(Cstr: { (this: T): void }, proto: P) {  
9    const MyConstructor = function (): Proto<P, T> {  
10      return new Cstr;  
11    };  
12    Object.setPrototypeOf(MyConstructor, prototype, proto);  
13    return const myConstructedItem: Proto<{  
14      otherField: boolean;  
15    }, {  
16      field: number;  
17    }>  
18  const myConstructedItem = myConstruct();  
19  
20  console.log(myConstructedItem);  
21
```

ts function\_construct\_typed.ts ×

examples > ts function\_construct\_typed.ts > ...

```
1  function OtherConstruct(this: { field: number }) {}  
2  OtherConstruct.prototype = {  
3    otherField: true  
4  }  
5  
6  type Proto<P, T> = Pick<P, Exclude<keyof P, keyof T>> & T;  
7  
8  const define = function <P extends object, T>(Cstr: { (this: T): void }, proto: P) {  
9    const MyConstructor = function (): Proto<P, T> {  
10      return new Cstr;  
11    };  
12    Object.setPrototypeOf(MyConstructor, prototype, proto);  
13    return const myConstructedItem: Proto<{  
14      otherField: boolean;  
15    }, {  
16      field: number;  
17    }>  
18  const myConstructedItem = myConstruct();  
19  
20  console.log(myConstructedItem);  
21
```

ts function\_construct\_typed.ts ×

examples > ts function\_construct\_typed.ts > ...

```
1  function OtherConstruct(this: { field: number }) {}  
2  OtherConstruct.prototype = {  
3    otherField: true  
4  }  
5  
6  type Proto<P, T> = Pick<P, Exclude<keyof P, keyof T>> & T;  
7  
8  const myConstruct = function(): Proto<{  
9    field: number;  
10   otherField: boolean; }> {  
11    Object.setPrototypeOf(MyConstructor.prototype, proto);  
12    return const myConstructedItem: Proto<{  
13      otherField: boolean; }> {  
14        const field: number = this.field;  
15        return {  
16          field, otherField: true; };  
17      };  
18    const myConstructedItem = myConstruct();  
19    console.log(myConstructedItem);  
20  
21
```

**но есть проблемка ...**



**mnemonica** TS

0.9.955 • Public • Published 3 days ago

[Readme](#)

[Code](#)

Beta

[0 Dependencies](#)

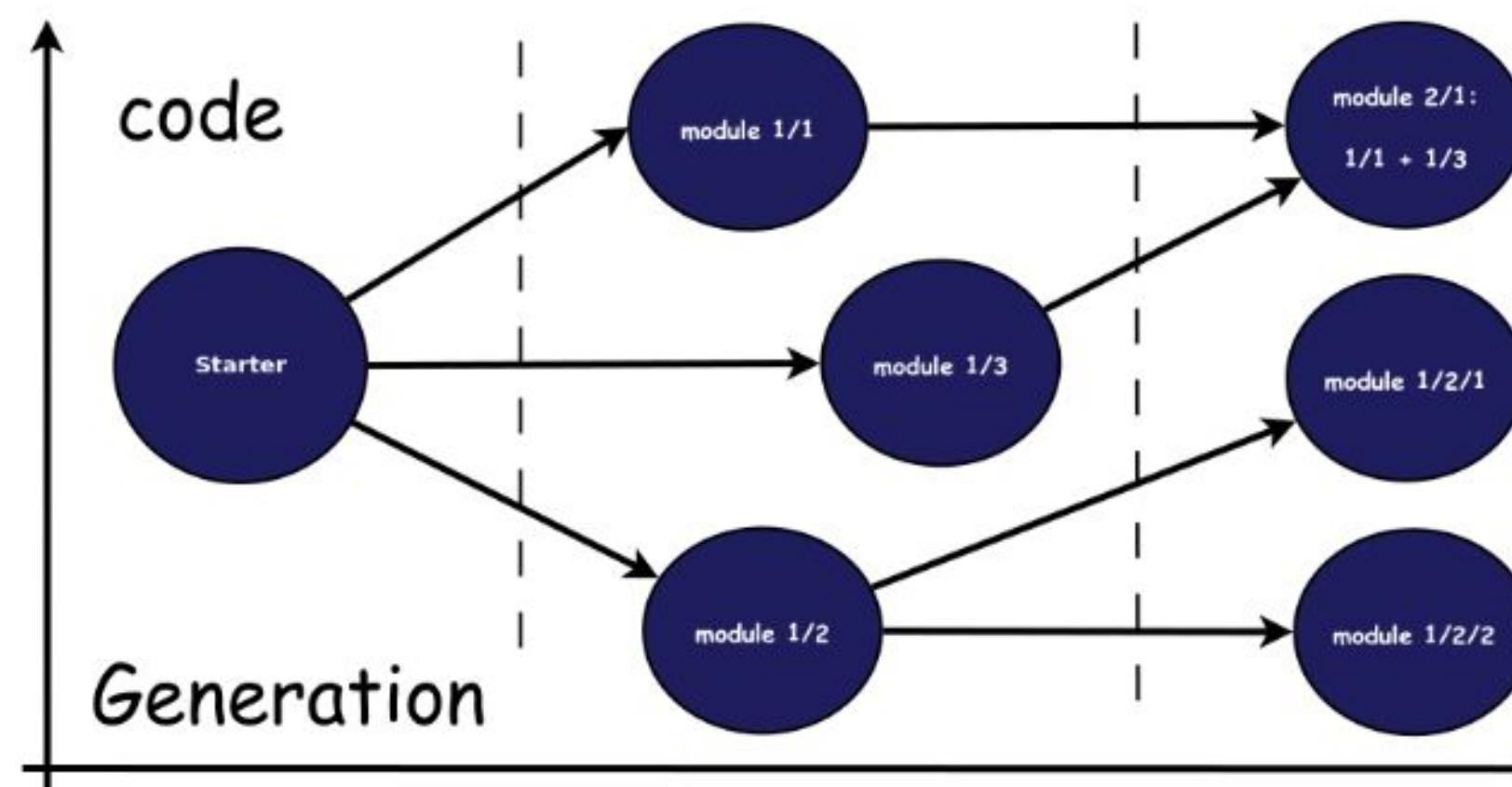
[0 Dependents](#)

[136 Versions](#)

## mnemonica is

abstract technique that aids information retention : instance inheritance system

... allows us to make inherited descriptions of mappings of transformations from predecessor structured data types to the successors, as if it was math  $f(x) \Rightarrow y$  ... and we will use this keyword as a persistent data structure where we will apply that transformations



Install

`> npm i mnemonica`



Repository

[github.com/wentout/mnemonica](https://github.com/wentout/mnemonica)

Homepage

[github.com/wentout/mnemonica#read...](https://github.com/wentout/mnemonica#readme)

Weekly Downloads

26



Version

0.9.955

License

MIT

Unpacked Size

120 kB

Total Files

80

Issues

0

Pull Requests

0

TS test-example.ts ×

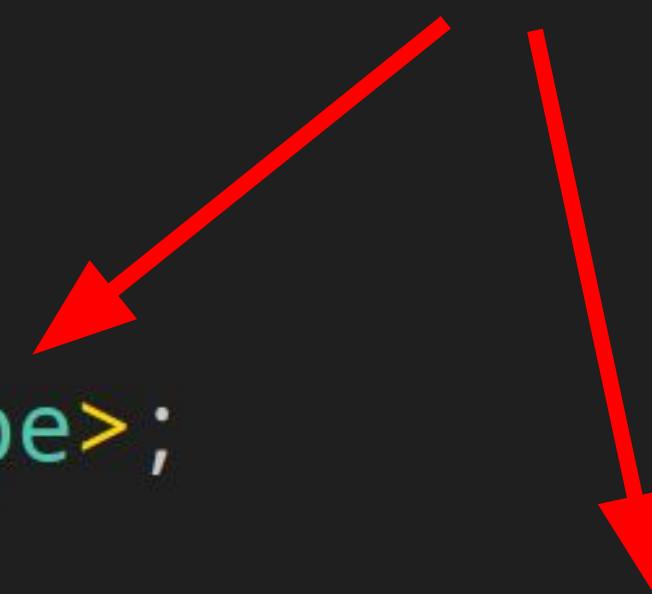
core > test-ts > TS test-example.ts > ...

```
16
17  const first = new FirstType();
18
19  type TSecondInstance =
20  | InstanceType<typeof SecondType>;
21
22  const second = new first.SecondType() as TSecondInstance;
23
24  // { first: undefined, second: "SecondType" }
25  console.log(second);
26
27
```

TS test-example.ts ×

core > test-ts > TS test-example.ts > ...

```
16
17  const first = new FirstType();
18
19  type TSecondInstance =
20  →   → InstanceType<typeof SecondType>;
21
22  const second = new first.SecondType() as TSecondInstance;
23
24  // { first: undefined, second: "SecondType" }
25  console.log(second);
26
27
```



```
TS test-example.ts ×  
core > test-ts > TS test-example.ts > ...  
16  
17 const first = new FirstType();  
18  
19 type TSecondInstance =  
20     InstanceType<typeof SecondType>;  
21  
22 const second = new first.SecondType() as TSecondInstance;  
23  
24 // { first: undefined, second: "SecondType" }  
25 console.log(second);  
26  
27
```



**... решение есть ...**



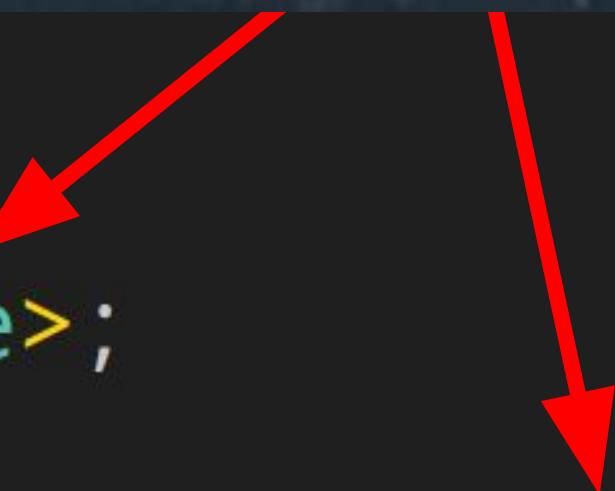
... apply, call, bind ...



TS test-example.ts ×

core > test-ts > TS test-example.ts > ...

```
16
17  const first = new FirstType();
18
19  type TSecondInstance =
20    | InstanceType<typeof SecondType>;
21
22  const second = new first.SecondType() as TSecondInstance;
23
24  // { first: undefined, second: "SecondType" }
25  console.log(second);
26
27
```



TS init.ts

X

TS direct.ts

TS apply.ts

TS call.ts

TS bind.ts

2023-11-HolyJS > examples > TS init.ts > ...

```
1 import { define } from 'mnemonica';
2
3 export const FirstType = define('SomeType', function (this: {
4   first: 'FirstType',
5 }) {
6   this.first = 'FirstType';
7 });
8
9 export const SecondType = FirstType.define('SecondType', function (this: {
10  first: undefined,
11  second: string,
12 }) {
13  this.first = undefined;
14  this.second = 'SecondType';
15 });
16
```

TS init.ts

TS direct.ts X

TS apply.ts

TS call.ts

TS bind.ts

2023-11-HolyJS > examples > TS direct.ts > ...

```
1 import { FirstType, SecondType } from './init';
2
3 const first = new FirstType();
4
5 type TSecondInstance = InstanceType<typeof SecondType>;
6 const second = new first.SecondType() as TSecondInstance;
7
8 // { first: undefined, second: string }
9 console.log(second);
10
11
12
```

TS init.ts

TS apply.ts X

TS call.ts

TS bind.ts

2023-11-HolyJS > examples > TS apply.ts > ...

```
1 import { apply } from 'mnemonica';
2
3 import { FirstType, SecondType } from './init';
4
5 const first = new FirstType();
6
7 // { first: undefined, second: string }
8 const second = apply(first, SecondType);
9 console.log(second);
10
11
```

TS init.ts

TS apply.ts X

TS call.ts

TS bind.ts

2023-11-HolyJS > examples > TS apply.ts > ...

```
1 import { apply } from 'mnemonica';
2
3 import const second: {
4   [x: string]: new () => unknown;
5   first: undefined;
6   second: string;
7   // ...
8 }
9
10 const second = apply(first, SecondType);
11 console.log(second);
```

TS init.ts

TS apply.ts X

TS call.ts

TS bind.ts

2023-11-HolyJS > examples > TS apply.ts > ...

```
1 import { apply } from 'mnemonica';
2
3 import const second: {
4   [x: string]: new () => unknown;
5   first: undefined;
6   second: string;
7   // ...
8   // ...
8 const second = apply(first, SecondType);
9 console.log(second);
10
11
```

TS init.ts

TS apply.ts

TS call.ts

X

TS bind.ts

2023-11-HolyJS > examples > TS call.ts > ...

```
1 import { call } from 'mnemonica';
2
3 import { FirstType, SecondType } from './init';
4
5 const first = new FirstType();
6
7 // { first: undefined, second: string }
8 const second = call(first, SecondType);
9 console.log(second);
10
11
```

TS init.ts

TS apply.ts

TS call.ts

TS bind.ts

code > talks > 2023-11-HolyJS > examples > TS call.ts > ...

```
1 import { call } from 'mnemonica';
2
3 import const second: {
4   [x: string]: new () => unknown;
5   first: undefined;
6   second: string;
7   // { f }
8   const second = call(first, SecondType);
9   console.log(second);
10
```

TS init.ts

TS apply.ts

TS call.ts

TS bind.ts

code > talks > 2023-11-HolyJS > examples > TS call.ts > ...

```
1 import { call } from 'mnemonica';
2
3 import const second: {
4   | [x: string]: new () => unknown;
5   | first: undefined;
6   | second: string;
7   // { f }
8 const second = call(first, SecondType);
9 console.log(second);
10
```

TS init.ts

TS apply.ts

TS call.ts

TS bind.ts

X

2023-11-HolyJS > examples > TS bind.ts > ...

```
1 import { bind } from 'mnemonica';
2
3 import { FirstType, SecondType } from './init';
4
5 const first = new FirstType();
6
7 const boundSecond = bind(first, SecondType);
8
9 // { first: undefined, second: string }
10 const second = boundSecond();
11 console.log(second);
12
```

TS init.ts

TS apply.ts

TS call.ts

TS bind.ts

X

2023-11-HolyJS > examples > TS bind.ts > ...

```
1 import { bind } from 'mnemonica';
2
3 import { FirstType, SecondType } from './init';
4
5 const second: {
6   [x: string]: new () => unknown;
7   first: undefined;
8   second: string;
9   // ...
10} = bind();
11
12const second = boundSecond();
13console.log(second);
```

TS init.ts

TS apply.ts

TS call.ts

TS bind.ts

X

2023-11-HolyJS > examples > TS bind.ts > ...

```
1 import { bind } from 'mnemonica';
2
3 import { FirstType, SecondType } from './init';
4
5 const second: {
6   [x: string]: new () => unknown;
7   first: undefined;
8   second: string;
9   // ...
10} const second = boundSecond();
11 console.log(second);
12
```

... class ...



TS init.ts

X

TS direct.ts

TS apply.ts

TS call.ts

TS bind.ts

2023-11-HolyJS > examples > TS init.ts > ...

```
1 import { define } from 'mnemonica';
2
3 export const FirstType = define('SomeType', function (this: {
4   first: 'FirstType',
5 }) {
6   this.first = 'FirstType';
7 });
8
9 export const SecondType = FirstType.define('SecondType', function (this: {
10  first: undefined,
11  second: string,
12 }) {
13  this.first = undefined;
14  this.second = 'SecondType';
15 });
16
```

```
9  export const SecondType = FirstType.define('SecondType',  
10     function ()  
11       this: {  
12         first: undefined,  
13         second: string,  
14       }){  
15       this.first = undefined;  
16       this.second = 'SecondType';  
17     };  
18  
19  
20  export const ThirdType = SecondType.define('ThirdType', class {  
21    third: string  
22    constructor(){  
23      this.third = 'ThirdType';  
24    }  
25  });  
26
```

```
9  export const SecondType = FirstType.define('SecondType',  
10     function ()  
11       this: {  
12         first: undefined,  
13         second: string,  
14       }){  
15         this.first = undefined;  
16         this.second = 'SecondType';  
17     };  
18  
19  
20  export const ThirdType = SecondType.define('ThirdType', class {  
21    third: string  
22    constructor(){  
23      this.third = 'ThirdType';  
24    }  
25  });  
26
```



TS init.ts

TS third.ts X

TS apply.ts

TS call.ts

TS bind.ts

2023-11-HolyJS > examples > TS third.ts > ...

```
1 import { apply } from 'mnemonica';
2
3 import { FirstType, SecondType, ThirdType } from './init';
4
5 const first = new FirstType();
6
7 // { first: undefined, second: string }
8 const second = apply(first, SecondType);
9 console.log(second);
10
11 const third = apply(second, ThirdType);
12 console.log(third);
13
```

TS init.ts

TS third.ts X

TS apply.ts

TS call.ts

TS bind.ts

2023-11-HolyJS > examples > TS third.ts > ...

```
1 import { apply } from 'mnemonica';
2
3 import { FirstType, SecondType, ThirdType } from './init';
4
5 const third: {
6   [x: string]: new () => unknown;
7   first: undefined;
8   second: string;
9   third: string;
10 }
11 const third = apply(second, ThirdType);
12 console.log(third);
13
```

TS init.ts

TS third.ts X

TS apply.ts

TS call.ts

TS bind.ts

2023-11-HolyJS > examples > TS third.ts > ...

```
1 import { apply } from 'mnemonica';
2
3 import { FirstType, SecondType, ThirdType } from './init';
4
5 const third: {
6   [x: string]: new () => unknown;
7   first: undefined;
8   second: string;
9   third: string; ←
10 }
11 const third = apply(second, ThirdType); ←
12 console.log(third);
13
```



пример



пример

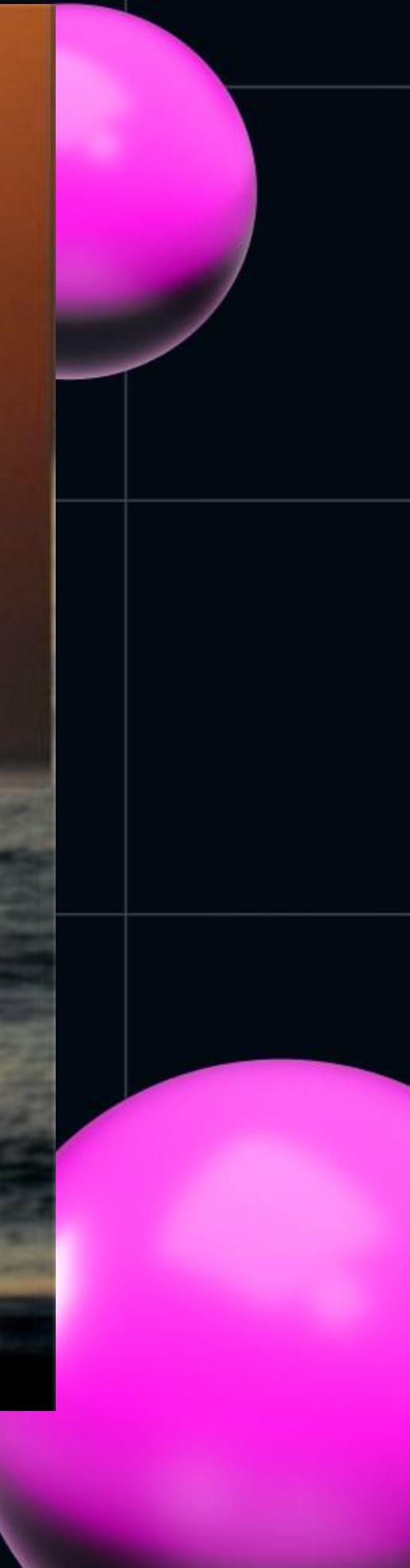
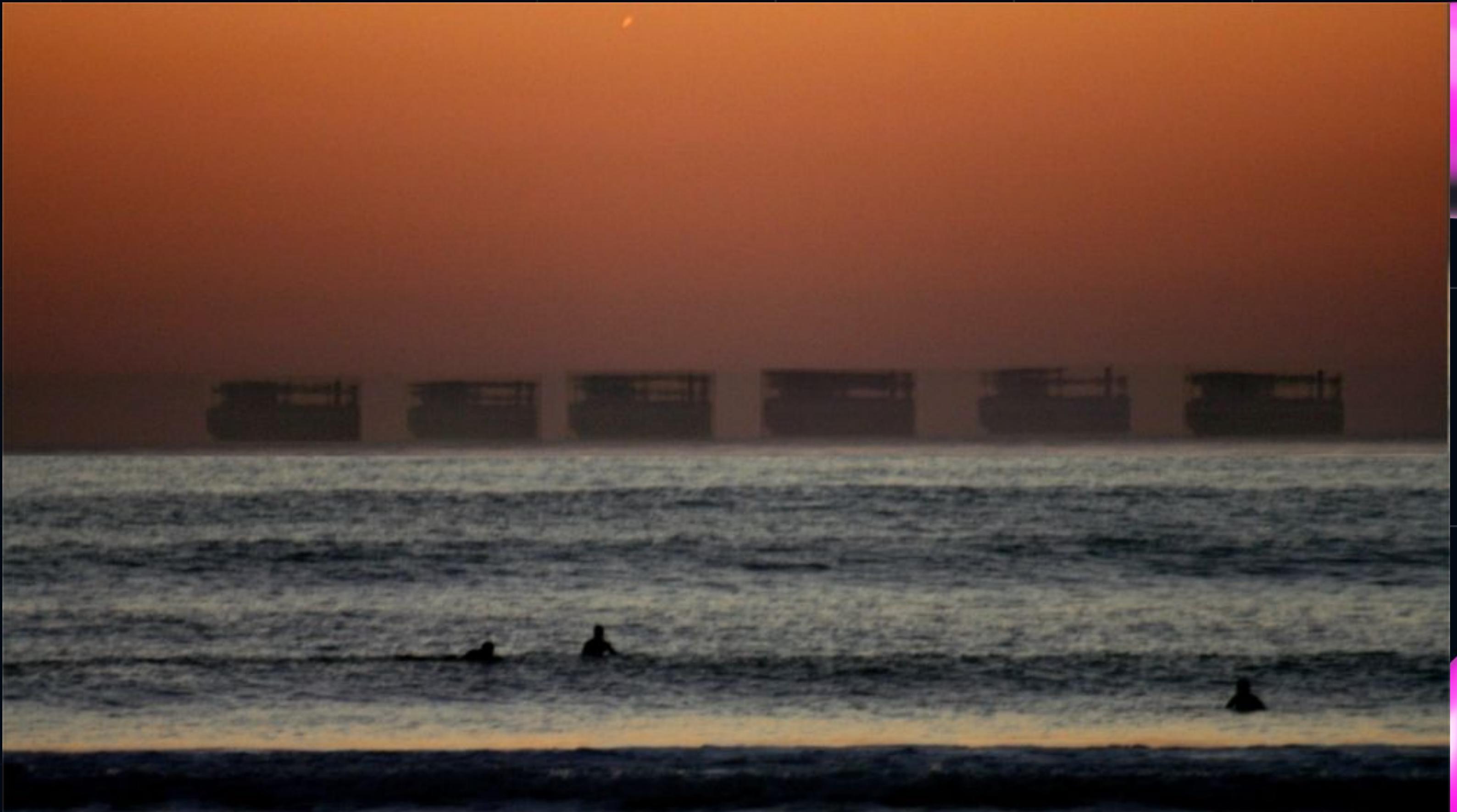


- контекст постановки задачи
- формулировка проблематики
- про что уже рассказывал по теме
- как создаётся код для решения
- как развивается решение: что сделано
- **практический пример использования**

- контекст постановки задачи
- формулировка проблематики
- про что уже рассказывал по теме
- как создаётся код для решения
- как развивается решение: что сделано
- **практический пример использования**



# Fata Morgana



# DEMO





**что дальше**

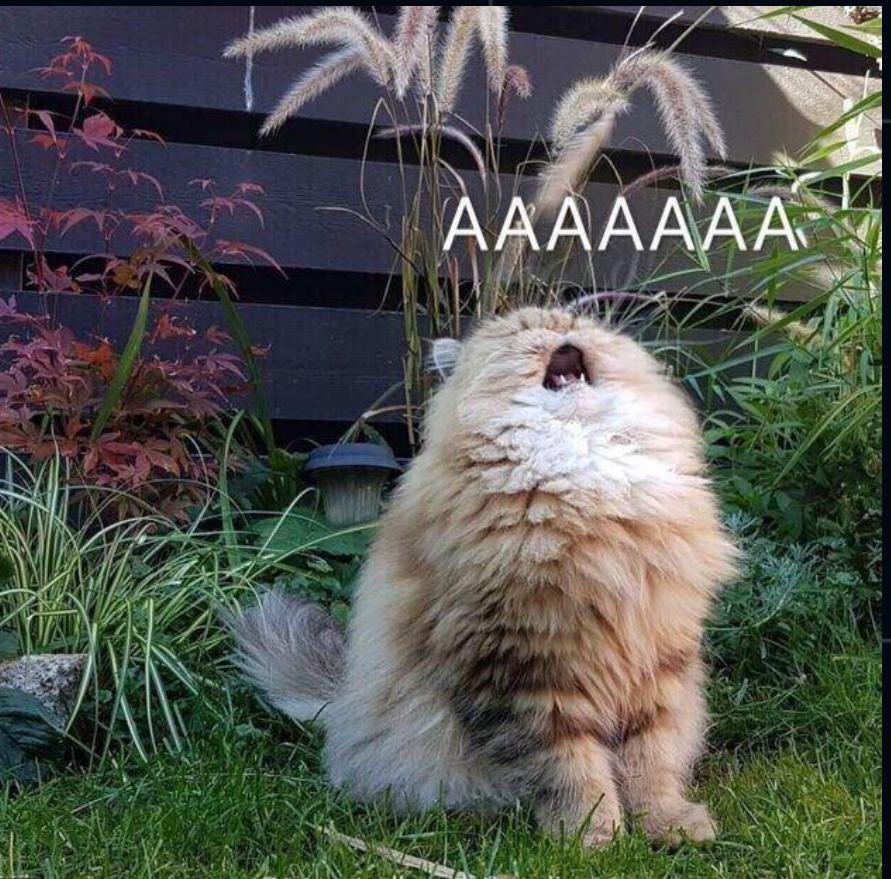


что дальше



7

- контекст постановки задачи
- формулировка проблематики
- про что уже рассказывал по теме
- как создаётся код для решения
- как развивается решение: что сделано
- практический пример использования
- **что ещё можно сделать дальше**



... decorators ...



ts init.ts

ts decorator.ts ×

ts third.ts

ts apply.ts

ts call.ts

ts bind.ts

2023-11-HolyJS > examples > ts decorator.ts > ...

```
1 import { define } from 'mnemonica';
2
3 function defined<T extends { new(): unknown }> (cstr: T, s: ClassDecoratorContext<T>) {
4   constTypeDef = define(s.name, cstr);
5   Object.setPrototypeOf(cstr.prototype, newTypeDef);
6 }
7
8 @defined
9 class MyClass {
10   z: number;
11   constructor() {
12     this.z = 123;
13   }
14 }
15
16 const myInstance = new MyClass();
17 console.log(myInstance.z);
18
```

TS init.ts

TS decorator.ts X

TS third.ts

TS apply.ts

TS call.ts

TS bind.ts

2023-11-HolyJS > examples > TS decorator.ts > ...

```
1 import { define } from 'mnemonica';
2
3 function defined<T extends { new(): unknown }>(cstr: T, s: ClassDecoratorContext<T>) {
4   constTypeDef = define(s.name, cstr);
5   Object.setPrototypeOf(cstr.prototype, newTypeDef);
6 }
7
8 @defined
9 class MyClass {
10   z: number;
11   constructor() {
12     this.z = 123;
13   }
14 }
15
16 const myInstance = new MyClass();
17 console.log(myInstance.z);
```



... HO ...



TS init.ts

TS decorator.ts 1 X

TS third.ts

TS apply.ts

TS call.ts

TS bind.ts

2023-11-HolyJS > examples > TS decorator.ts > ...

```
1 import { define } from 'mnemonica';
2
3 function defined<T extends { new(): unknown }>(cstr: T, s: ClassDecoratorContext<T>) {
4   constTypeDef = define(s.name, cstr);
5   Object.setPrototypeOf(cstr.prototype, newTypeDef);
6 }
7
8 @defined
9 class MyClass {
10   z: number;
11   constructor() {
12     this.z = 123;
13   }
14 }
15
16 const myInstance = new MyClass;
17 console. Property 'define' does not exist on type 'typeof MyClass'. ts(2339)
18
19 any
20
21 MyClass.define()
```

TS init.ts

TS decorator.ts 1 X

TS third.ts

TS apply.ts

TS call.ts

TS bind.ts

2023-11-HolyJS > examples > TS decorator.ts > ...

```
1 import { define } from 'mnemonica';
2
3 function defined<T extends { new(): unknown }>(cstr: T, s: ClassDecoratorContext<T>) {
4   constTypeDef = define(s.name, cstr);
5   Object.setPrototypeOf(cstr.prototype, newTypeDef);
6 }
7
8 @defined
9 class MyClass {
10   z: number;
11   constructor() {
12     this.z = 123;
13   }
14 }
15
16 const myInstance = new MyClass;
17 console. Property 'define' does not exist on type 'typeof MyClass'. ts(2339)
18
19 any
20
21 MyClass.define()
```



и **ещё на самом деле**  
**myInstance** **хоть и как бы**  
**является экземпляром того,**  
**что создаётся от define,**  
**только вот ...**



TS init.ts

TS decorator.ts 1 X

TS third.ts

TS apply.ts

TS call.ts

TS bind.ts

2023-11-HolyJS > examples > TS decorator.ts > ...

```
1 import { define } from 'mnemonica';
2
3 function defined<T extends { new(): unknown }>(cstr: T, s: ClassDecoratorContext<T>) {
4   constTypeDef = define(s.name, cstr);
5   Object.setPrototypeOf(cstr.prototype, newTypeDef);
6 }
7
8 @defined
9 class MyClass {
10   z: number;
11   constructor() {
12     this.z = 123;
13   }
14 }
15
16 const myInstance = new MyClass;
17 console. Property 'define' does not exist on type 'typeof MyClass'. ts(2339)
18
19 any
20
21 MyClass.define()
```



**... так что ...**





Спасибо !





следующий доклад

# Mnemonica Project



Виктор  
Вершанский