

СИТИДРАЙВ

Кодовая эволюция:
избавляемся от
CoffeeScript
с помощью нейросети





Эксперт

Сорокин Иван

Руководитель группы разработки
системных сервисов

ivan.sorokin@citydrive.ru | +7 999 530-37-30



Спикер

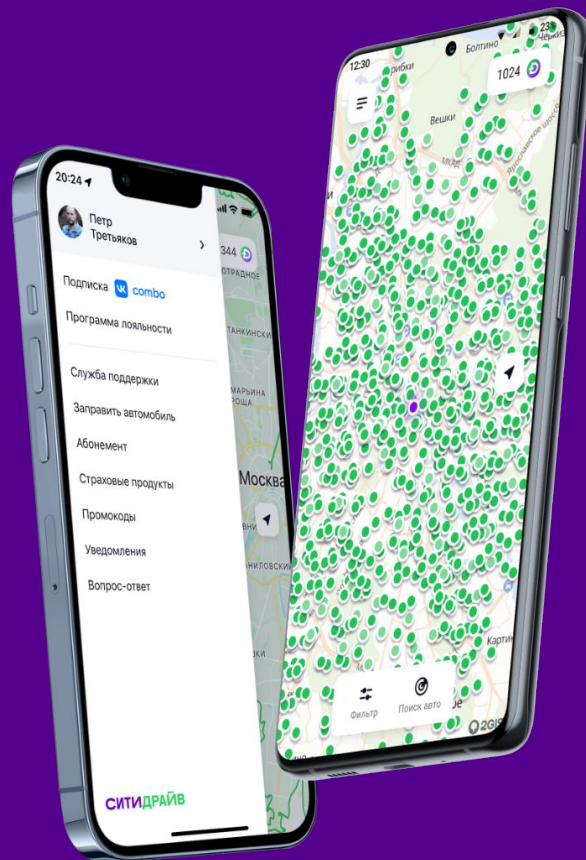
Долгавин Никита

Руководитель группы разработки
функциональных сервисов

n.dolgavin@citydrive.ru | +7 999 970-96-96

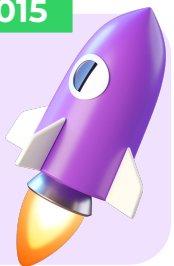
Ситидрайв — это каршеринг

- ★ Аренда автомобилей **от 1 минуты**
- ★ Всё взаимодействие с пользователем — **онлайн** через приложение и службу поддержки
- ★ Единая платформа **в трёх городах:** Москве, Санкт-Петербурге и Сочи
- ★ **2 место** на рынке каршеринга по размерам парка и количеству поездок



История

2015



Появились
на рынке
(YouDrive)

2019



Вошли
в «O2O Холдинг»
VK и Сбера

2021



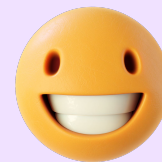
Провели
ребрендинг и стали
Ситидрайвом

2021



Въехали в топ-3
сервисов
каршеринга

2023



Рост автопарка
на 40%, запуск
шоурума и СТО

В цифрах

> 15 000 автомобилей

≈ 80 000 поездок в сутки

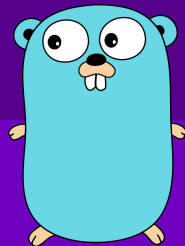
6 поездок на 1 авто в сутки 



IT команда



80 человек IT
команда



все новое
пишем на Go



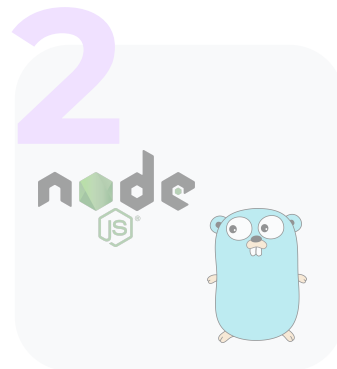
более 100 backend-
сервисов

Монолит

Из чего состоит backend каршеринга?



Монолит

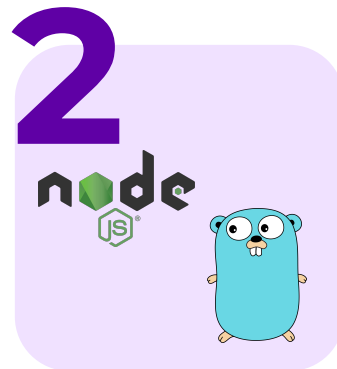


Микросервисы

Из чего состоит backend каршеринга?



Монолит



Микросервисы

Монолит в молодости

Запуск в PM2



NodeJS 12

>600k строк JS

CoffeeScript 1

> 30к coffee

Пакеты версий 2015 г



Отсутствие метрик, трассировок

Монолит в молодости

Запуск в PM2



NodeJS-12

>600k строк JS

CoffeeScript 1

> 30к coffee

Пакеты версий 2015-г



Отсутствие метрик, трассировок

Какие проблемы в CoffeeScript?

coffeascript

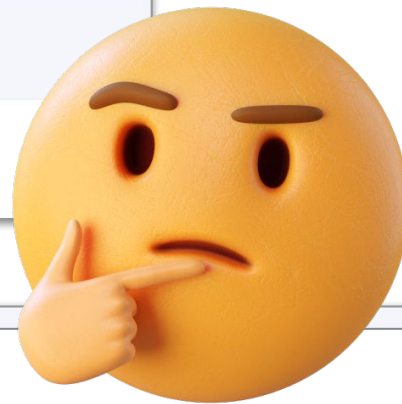


Вакансии

Резюме

Компании

Найдено 775 резюме у 678 соискателей



erlang



Вакансии

Резюме

Компании

Найдено 2 137 резюме у 1 681 соискателя

Вдруг откуда не ВОЗЬМИСЬ, ПОЯВИЛСЯ ...

```
penaltyList = (req, callback)->  
    new PenaltiesAdminModel()  
    .fetchAll callback
```

Ад обратных вызовов



```
getDataFromAPI (data1) ->  
  processData data1, (processedData1) ->  
    uploadData processedData1, (response1) ->  
      getDataFromDb response1, (data2) ->  
        processDataFromDb data2, (prcData2) ->  
          sendDataToAPI prcData2, (response2) ->  
            logger.info "success", response2
```

А бывало и вот так



```
.then (data, cb) =>
  if data
    accountingProperties.bill_qr_id = data.bill_qr_id
    utils.addFileToImageServer(
      checkFileName,
      checkId + path.extname(checkFileName),
      false,
      'other_liquids_replenish'
    )
  .then (data, cb) =>
```

Вроде бы все ок ...

Но здесь утечка памяти
из-за пропущенного
вызова cb

Ну линтер хоть можно поставить?



НЕТ :)



Варианты решения проблемы:

- ▶ Обновиться
- ▶ Проверять транспирированный код



CONFUSED



Подсветка IDE

Пытаемся понять, какой `@_process` вызывается в данном месте

```
invoke (data, cb) :void =>
  @_process false, cb
.then (r
  user =
# FIXM
if !us
  util
  retu
  @_process (mocha)
  @_process (mocha)
  @_process (mocha)
  @_process (mocha)
```

Choose Declaration

- Ⓟ `_process (ejs)`
- Ⓡ Ⓢ `_process (Farm, jest-worker)`
- Ⓟ `_process (mocha)`
- Ⓟ `_process (mocha)`
- Ⓟ `_process (mocha)`
- Ⓟ `_process (mocha)`



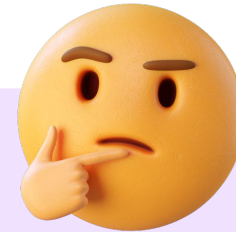
А он в этом же файле несколькими строками выше

```
_process: (onlyNegative, callback) :void
  invoke (data, cb) :void =>
    @_fetchUsers cb
    .and (data, cb) :void =>
```

Что еще?



**Увеличенное
время запуска
сервиса
(плюс ~3 секунды)**



**Полифилы
потребляют
память
(об этом позже)**

**Перейдем к решению
проблем**

Как
пробовали
подойти
к задаче?



Переписать руками



Декофенаторы



ИИ

Как переписывает код декофенатор?

```
Untitled-1

fetchData = (callback) ->
  setTimeout(() ->
    data = { name: 'John', age: 30 }
    callback(data)
    , 1000)
processData = (data) ->
  console.log("Name: #{data.name}, Age: #{data.age}")
fetchData(processData)
```

Исходный

```
(function() {  
  var fetchData, processData;  
  fetchData = function(callback) {  
    return setTimeout(function() {  
      var data;  
      data = {  
        name: 'John',  
        age: 30  
      };  
      return callback(data);  
    }, 1000);  
  };  
  processData = function(data) {  
    return console.log("Name: " + data.name + ", Age: " + data.age);  
  };  
  fetchData(processData);  
}).call(this);
```

```
const fetchData = (callback) => {
  setTimeout(() => {
    const data = { name: 'John', age: 30 };
    callback(data);
  }, 1000);
};

const processData = (data) => {
  console.log(`Name: ${data.name}, Age: ${data.age}`);
};

fetchData(processData);
```

Как**обрабатывает ИИ**

Выбор модели и различные варианты

- ▶ text-davinci-003
- ▶ LaMDA
- ▶ code-davinci-002



Демонстрация работы ИИ

Usage

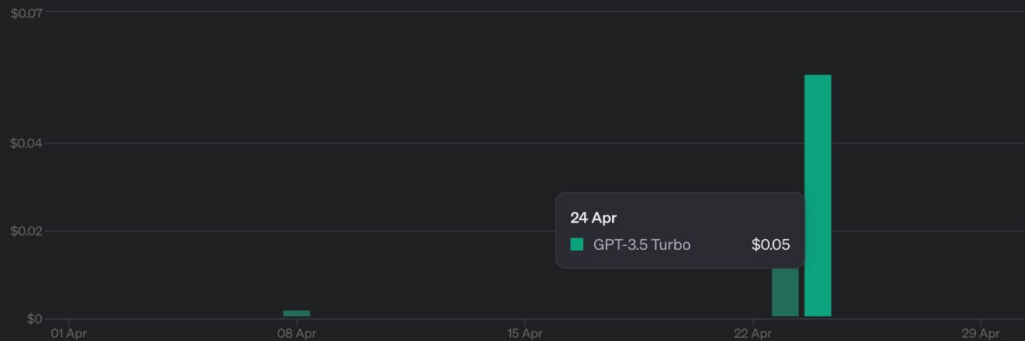
Cost

Activity

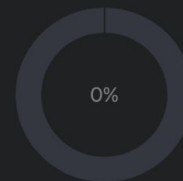
< April >

Export

Monthly Spend \$0,07



Project's Monthly Bill apr. 1 - 30



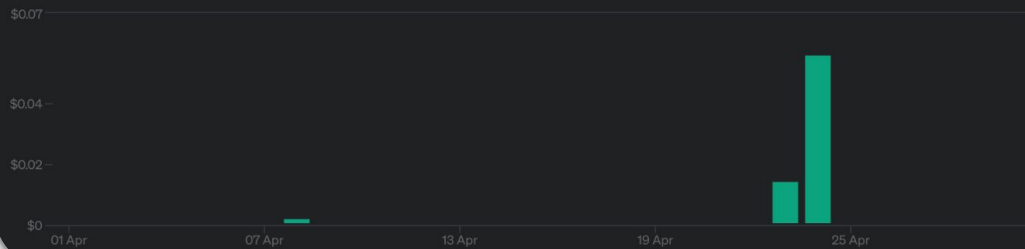
\$0.07

/ \$120.00 limit

Increase limit

View monthly bill for the whole organization →

GPT-3.5 Turbo \$0,07



Model	Input	Output
gpt-4	\$30.00 / 1M tokens	\$60.00 / 1M tokens
gpt-4-32k	\$60.00 / 1M tokens	\$120.00 / 1M tokens

GPT-3.5 Turbo models are capable and cost-effective.

`gpt-3.5-turbo-0125` is the flagship model of this family, supports a 16K context window and is optimized for dialog.

`gpt-3.5-turbo-instruct` is an Instruct model and only supports a 4K context window.

[Learn about GPT-3.5 Turbo](#) ↗

Model	Input	Output
gpt-3.5-turbo-0125	\$0.50 / 1M tokens	\$1.50 / 1M tokens
gpt-3.5-turbo-instruct	\$1.50 / 1M tokens	\$2.00 / 1M tokens

Достоинства



- ▶ Экономия времени и ресурсов
- ▶ Оптимизированный код
- ▶ Экономия нервов

Достоинства



- ▶ Экономия времени и ресурсов
- ▶ Оптимизированный код
- ▶ Экономия нервов

Недостатки



- ▶ Необходимо иметь тесты
- ▶ Необходимо иметь метрики
- ▶ Нужно посмотреть код

Ну, и как всё прошло?



Всего
>10 релизов



6 багов

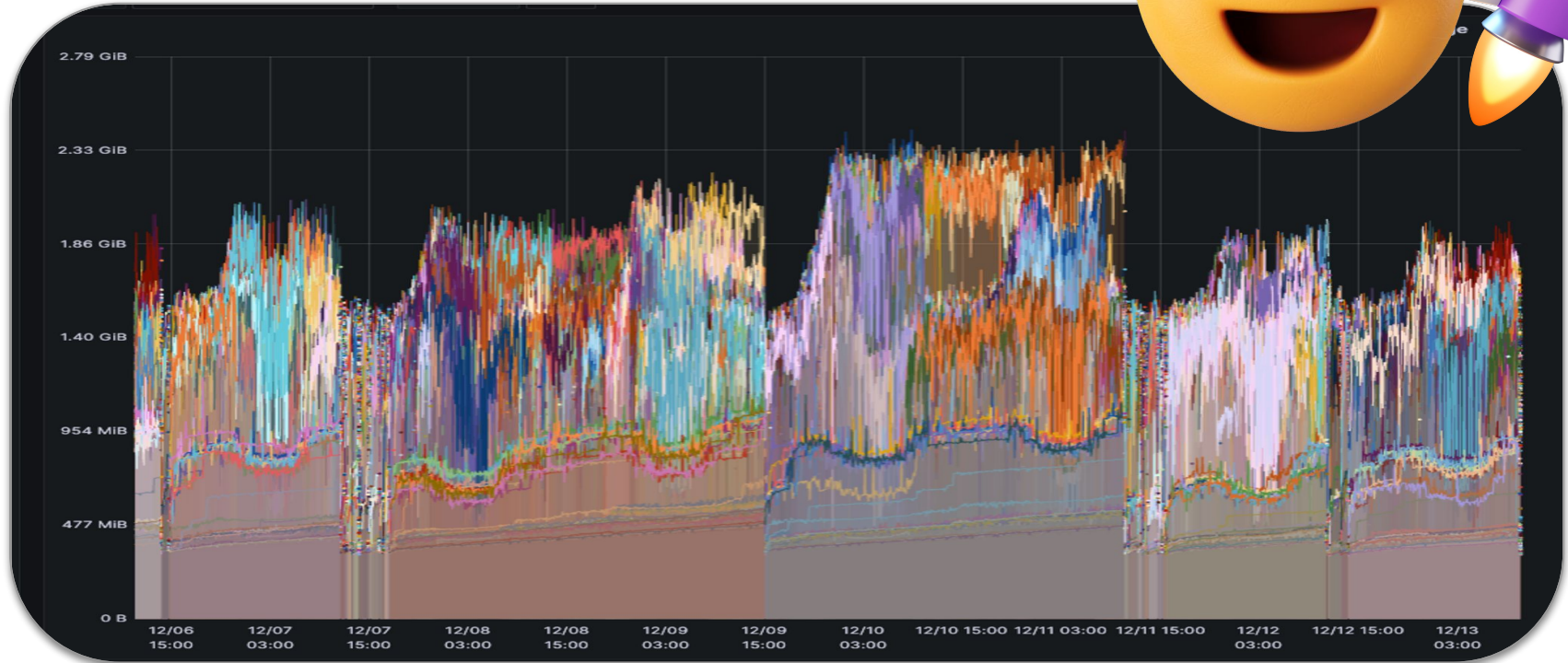
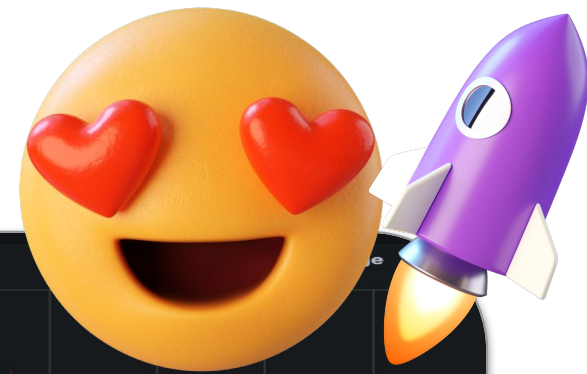


1 релиз
откатился
канарейкой



1 инцидент

Результаты



СИТИДРАЙВ

**Спасибо
за внимание!**

