Цель - разработать систему накрутки поисковых факторов для SEO в Yandex и Google

Активность каждого из профилей разбивается на 3 блока:  
1) целевая активность по накрутке действий ПФ для SEO по исходным ключевым запросам заданных сайтов в Яндекс и Google с указанием региона (или без указания региона)  
2) балластовая активность имилирующая действия живого человека для прохождения антибот систем  
3) дополнительная активность связанная с накоплением cookies различных систем, заходом на большие популярные сайты, просмотром видео, активностями в сервисах яндекса и google (то что доступно без регистрации), переходам по баннерным, тизерным, рекламным сетям, использование картографических сервисов, использование локальных сервисов - это все необходимо для создания устойчивого пользовательского профиля похожего на реального человека с привязкой к локации, определенные интересы, приносящего деньги поисковым и рекламным системам, для снижения антикапча-активности  
  
Все действия всегда совершаются от имени определенного профиля, история активности, а также cookies и настройки - все это сохраняется и накапливается по каждому профилю

### **Основные функциональные требования:**

| **№** | **Требование** | **Описание** |
| --- | --- | --- |
| 1 | Эмуляция реального пользователя | Все действия должны быть случайными, разнообразными и имитировать живого человека |
| 2 | Fingerprinting | Настройка уникального браузерного отпечатка для каждого профиля (ОС, шрифты, Canvas, гео, язык и т.д.) |
| 3 | Прокси в ротации (RF) | Использование мобильных прокси с российскими IP, вращение между запросами |
| 4 | Playwright + JS/TS | Использовать Playwright для реализации автоматизации |
| 5 | Сохранение cookies и истории | Сохранять состояние сессии, чтобы поддерживать "живость" профилей |
| 6 | Рандомизированные задержки | Временные интервалы между действиями должны быть нелинейными |
| 7 | Плавное движение мыши | Эмуляция движения курсора к элементам перед кликом |
| 8 | Скроллинг страницы | Рандомный скролл вверх/вниз, с разной скоростью |
| 9 | Балластные действия | Ложные клики, перемещения мыши, набор текста без цели |
| 10 | Поисковые действия | Запросы в Яндекс/Google, переход по результатам, поиск нужного домена или URL |
| 11 | Видео и медиа активность | Прогрузка видео, пауза, воспроизведение, длительность просмотра |
| 12 | Региональная специфика | Учет региона: Язык, часовой пояс, координаты, локализация контента |
| 13 | Целевые действия на сайте | Переходы по ссылкам, клики, скролл, прогрев страницы |
| 14 | Работа с вебвизором | Действия должны попадать в статистику аналитики (Yandex.Metrica, Google Analytics) |
| 15 | Прогрев профилей | Равномерная активность, повторные заходы, формирование историй |
| 16 | Мониторинг ошибок и отказов | Автоматический перезапуск упавших задач, логирование |

сделать чтобы в сценарии было открытие нескольких ссылок и при этом Фокус переходил на эти браузеры так чтобы видео начало просматриваться

Рандомные тайминги между действиями

Скроллинг рандомный  
Движение мышки плавные к объекту клика  
Поиск дальше

Набор текста в текстовом поисковом поле

Переходы по ссылкам  
Балластовые действия, например клики в пустое место, заходы на рандомные ссылки которые размещены на странице  
Надо так чтобы активность отображалась в метрике и в вебмастере

Мобильные прокси РФ в ротации, использование нескольких антикапча сервисов  
  
Сервер Ubuntu 22.04, вся автоматизация должна быть произведена с использованием PLAYWRIGHT

важно каждый день ходить по новостям, по яндекс рекламе и по google рекламе и по топовым сайтам

Искать рандомные темы

Замэтчиться с определенной региональной спецификой

Надо уметь искать нужный url или нужный домен в результатах выдачи поисковой системы и до этой позиции делать передвижение мышки чтобы было естественно

fingerprinting

OS

Canvas

Browser

Screen resolution

Browser resolution

Lang

GEO coordinates

Fonts

Js fingerprinting

= какие еще?

Сохранение всех cookies в рамках всех сессий профиля

Накопление истории профиля

Прогрев профилей постепенной активностью и регулярными дополнительными активностями

Повторно заходить на сайт

На сайте ходить по страницам  
Искать ссылки на загруженных страницах и приходить к целям

Так чтобы в вебвизоре отображалось

Генерация реальных фингерпринтов

Используем playwright

## **4. Полная структура затрат**

| **Компонент** | **Описание** | **Стоимость (ежемесячная)** |
| --- | --- | --- |
| Сервер | VPS с 2 vCPU, 4 GB RAM, без учета трафика - что рекомендуешь? | ~500–800 руб (~$6–$10) |
| Прокси | Российские мобильные (200 IP) | ~2000–4000 руб (~$30–$60) |
| Антикапча | API-сервис для решения капчи | ~500–1000 руб (~$7–$15) |
|  |  |  |
| ИТОГО |  | ~3200–6300 руб (~$45–$90) |

## **Антикапча: варианты и сравнение, какие еще предложишь (обязательно должна быть возможность прохождения капчи Яндекса)?**

<https://botlab.me/price>

<https://anti-captcha.com/ru/apidoc> ?  
<https://capsola.space/#price> ?

**CapSolver** - хорошо работает с Яндекс капчей

**NoCaptcha** - российский сервис

**XEvil** - можно развернуть локально

| **Сервис** | **Стоимость за 1000 решений** | **Поддержка API** | **Тип капчи** | **Уровень автоматизации** | **Комментарии** |
| --- | --- | --- | --- | --- | --- |
| CapMonster | ~$1–$2 | ✅ | Локально | Высокая | Работает через собственный софт, дороже при масштабировании |
| Anti-Captcha | ~$0.6–$1.2 | ✅ | Все типы | Высокая | Самый популярный сервис, интегрируется легко |
| 2Captcha | ~$0.8–$1.5 | ✅ | Все типы | Высокая | Хорошая документация, поддерживает Playwright |
| RuCaptcha | ~$0.5–$1 | ✅ | Все типы | Средняя | Русскоязычный аналог, дешевле, но медленнее |
| Manual solving | бесплатно | ❌ | Все типы | Низкая | Не подходит для автоматизации |

РИСКИ и способы их МИТИГАЦИИ

| Бан IP/прокси | Частый парсинг, одинаковые заголовки | Использовать качественные мобильные прокси, менять User-Agent, добавить рандомные паузы |
| --- | --- | --- |
| Обнаружение ботов | Антиботы Яндекса, Google, Cloudflare | Использовать плагины против обнаружения, уникальные отпечатки, human-like поведение |
| Одинаковые сессии | Не меняются cookies, fingerprint | Уникальные профили, ротация прокси, разные часовые пояса |
| Обнаружение в вебвизоре | Отсутствие mousemove, скролла | Реалистичное движение мыши, скролл, клики, время нахождения на странице |
| Видео не считается просмотренным | HTML5 video detection | Имитация начала воспроизведения, установка времени, pause/resume |
| Защита от XSS/JS-футпринтов | JsFingerprinting | Использование плагинов маскировки в playwright |
| Высокая нагрузка на сервер | Одновременная работа 200 агентов | Распределить нагрузку, использовать пулы браузеров, оптимизиров |

# **РИСКИ И СПОСОБЫ ИХ ПРЕОДОЛЕНИЯ (АЛГОРИТМИЧЕСКИЕ РЕШЕНИЯ)**

| **Риск** | **Причина** | **Алгоритмическое решение** |
| --- | --- | --- |
| Обнаружение как бота | Однотипный фингерпринт, поведение | Использовать уникальные профили, рандомизировать действия |
| Бан IP | Много запросов с одного IP | Вращать прокси, менять User-Agent, добавлять паузы |
| Капча | Антиботы требуют решения | Интеграция с Anti-Captcha, автоматическое распознавание капчи |
| Одинаковые сессии | Не меняются cookies, время захода | Сохранять cookies, имитировать разное время входа |
| Обнаружение headless-браузера | navigator.webdriver = true | Отключить детекцию headless, использовать stealth модуль |
| Видео не считается просмотренным | HTML5 video detection | Имитация начала воспроизведения, установка времени, pause/resume |
| Недостаточно активности в вебвизоре | Нет mousemove, скролла | Реалистичное движение мыши, скролл, клики, время нахождения на стра |

1. Используйте пулы браузеров и асинхронность
   * Запускайте 5–10 браузеров одновременно на сервере
   * Используйте Promise.all, p-queue, worker\_threads
2. Сохраняйте состояние сессии после каждого действия
3. Перезапускайте агента с последнего места
4. Используйте Redis/Kafka для очереди задач

## **Полная структура затрат**

| **Компонент** | **Описание** | **Стоимость (ежемесячная)** |
| --- | --- | --- |
| Сервер | VPS с 2 vCPU, 4 GB RAM, 1 ТБ трафика | ~500–800 руб (~$6–$10) |
| Прокси | Российские мобильные (200 IP) | ~2000–4000 руб (~$30–$60) |
| Антикапча | API-сервис для решения капчи | ~500–1000 руб (~$7–$15) |

Эти параметры должны быть уникальными или изменяемыми для каждого профиля , чтобы избежать обнаружения как бота.

### **🔹 Основные параметры:**

| **Группа** | **Параметр** | **Описание** |
| --- | --- | --- |
| User-Agent | navigator.userAgent | Строка User-Agent браузера |
| OS / Platform | navigator.platform,oscpu,deviceMemory | Информация об операционной системе |
| Canvas | Canvas rendering | Подделка рендеринга canvas |
| WebGL / GL Renderer | WebGLRenderingContext | Информация о GPU и драйверах |
| Screen resolution | screen.width,height,availWidth,availHeight | Разрешение экрана |
| Browser resolution | window.innerWidth,innerHeight | Размер окна браузера |
| Languages | navigator.languages,language | Языковые настройки браузера |
| Timezone | Intl.DateTimeFormat().resolvedOptions().timeZone | Часовой пояс |
| Geolocation | navigator.geolocation | Координаты пользователя |
| Fonts | Доступные шрифты | Через Canvas или Flash |
| AudioContext | AudioContext | Звуковой отпечаток |
| Battery | navigator.getBattery() | Уровень заряда устройства |
| DevicePixelRatio | window.devicePixelRatio | Соотношение пикселей устройства |
| Touch support | maxTouchPoints | Поддержка тач-устройств |
| MediaDevices | navigator.mediaDevices.enumerateDevices() | Устройства ввода/вывода |
| Permissions | navigator.permissions.query() | Разрешения сайта |
| Plugins | navigator.plugins | Плагины браузера |
| HardwareConcurrency | navigator.hardwareConcurrency | Количество ядер процессора |
| Do Not Track | navigator.doNotTrack | Настройка "Не отслеживать" |
| JS Engine | Function.prototype.toString() | Отпечаток JS-движка |
| ClientRects | Измерение DOM-элементов | Используется для детекции headless |

✅ Цель : сделать каждый профиль уникальным, но реалистичным (например, сочетание OS + разрешения экрана должно соответствовать реальным устройствам)

## **Дополнительные параметры для fingerprinting:**

javascript

const additionalFingerprints = {

*// WebRTC*

webRTC: {

localIP: 'RTCPeerConnection',

publicIP: 'RTCPeerConnection',

mediaDevices: 'navigator.mediaDevices'

},

*// Network Information*

network: {

connectionType: 'navigator.connection.effectiveType',

downlink: 'navigator.connection.downlink',

rtt: 'navigator.connection.rtt',

saveData: 'navigator.connection.saveData'

},

*// Hardware*

hardware: {

memory: 'navigator.deviceMemory',

cpuClass: 'navigator.cpuClass',

maxTouchPoints: 'navigator.maxTouchPoints',

vibrate: 'navigator.vibrate'

},

*// Browser features*

features: {

cookieEnabled: 'navigator.cookieEnabled',

onLine: 'navigator.onLine',

pdfViewerEnabled: 'navigator.pdfViewerEnabled',

webdriver: 'navigator.webdriver',

bluetooth: 'navigator.bluetooth'

},

*// Storage*

storage: {

localStorage: 'window.localStorage',

sessionStorage: 'window.sessionStorage',

indexedDB: 'window.indexedDB'

},

*// Permissions API*

permissions: {

geolocation: 'permissions.query({name: "geolocation"})',

notifications: 'permissions.query({name: "notifications"})',

camera: 'permissions.query({name: "camera"})',

microphone: 'permissions.query({name: "microphone"})'

},

*// CSS и рендеринг*

rendering: {

colorDepth: 'screen.colorDepth',

pixelDepth: 'screen.pixelDepth',

orientation: 'screen.orientation',

matchMedia: 'window.matchMedia queries'

}

};

установка

npm install playwright

npm install playwright-stealth

npm install puppeteer-extra-plugin-stealth

npm install @puppeteer/browsers

npm install faker-js

npm install axios

npm install redis

npm install dotenv

npm install winston

**Дополнительные библиотеки для установки:**

bash

*# Основные зависимости для работы с браузерами*

npm install playwright-extra

npm install playwright-extra-plugin-stealth

npm install playwright-extra-plugin-recaptcha

*# Работа с прокси*

npm install proxy-chain

npm install https-proxy-agent

npm install socks-proxy-agent

*# Управление задачами и потоками*

npm install p-queue

npm install bull

npm install bee-queue

*# База данных для хранения профилей*

npm install mongoose

npm install @prisma/client prisma

*# Утилиты*

npm install user-agents

npm install randomstring

npm install node-cron

npm install dayjs

npm install lodash

npm install uuid

*# Мониторинг и метрики*

npm install prom-client

npm install @sentry/node

*# Для работы с координатами*

npm install geolib

npm install @turf/random

*# Парсинг и валидация*

npm install joi

npm install yup

npm install cheerio

*# HTTP клиенты*

npm install got

npm install undici

*# TypeScript (если используете)*

npm install -D typescript

npm install -D @types/node

npm install -D ts-node

npm install -D nodemon

**✅ Основные компоненты:**

1. **Серверное приложение** - Node.js/TypeScript с Express API
2. **База данных** - MongoDB для хранения данных, Redis для кеширования
3. **Автоматизация браузера** - Playwright с антидетект функциями
4. **Система очередей** - Bull/Redis для управления задачами
5. **Мониторинг** - Prometheus + Grafana для метрик

### **✅ Инфраструктура:**

1. **Docker** - контейнеризация всех сервисов
2. **PM2** - управление процессами Node.js
3. **Nginx** - реверс-прокси (опционально)

### **✅ Инструменты разработки:**

1. **TypeScript** - типизированный код
2. **Jest** - тестирование
3. **ESLint/Prettier** - линтинг и форматирование
4. **CLI** - командный интерфейс для управления

### **✅ Документация:**

1. **README.md** - общее описание проекта
2. **Installation Guide** - пошаговая установка
3. **Configuration Guide** - настройка системы
4. **API Documentation** - REST API референс
5. **Troubleshooting** - решение проблем

### **✅ Безопасность:**

1. **Шифрование** - чувствительных данных
2. **Аутентификация** - API ключи
3. **Rate Limiting** - защита от перегрузки
4. **Изоляция** - контейнеры Docker

### **📁 Структура каталогов:**

seo-automation/

├── src/ # Исходный код

│ ├── core/ # Основная бизнес-логика

│ ├── services/ # Сервисы (БД, прокси, капча)

│ ├── api/ # REST API

│ ├── cli/ # CLI команды

│ ├── utils/ # Вспомогательные функции

│ └── types/ # TypeScript типы

├── tests/ # Тесты

├── scripts/ # Скрипты автоматизации

├── config/ # Конфигурационные файлы

├── docs/ # Документация

├── logs/ # Логи (генерируется)

└── dist/ # Скомпилированный код (генерируется)

### **🚀 Для начала работы:**

1. Клонируйте репозиторий
2. Запустите ./scripts/setup.sh
3. Настройте .env файл
4. Создайте профили: npm run cli profiles:create --count 50
5. Запустите первую задачу!

## **Возможности и свойства системы**

### **Основные возможности:**

1. **Многопрофильная эмуляция**
   * 200+ уникальных профилей с индивидуальными fingerprints
   * Персонализированное поведение для каждого профиля
   * Накопление истории и cookies
2. **Интеллектуальная автоматизация**
   * Адаптивные алгоритмы поведения
   * Машинное обучение для оптимизации
   * Автоматическое обнаружение и обход защит
3. **Поисковая оптимизация**
   * Работа с Яндекс и Google
   * Региональный таргетинг
   * Улучшение поведенческих факторов
4. **Мониторинг и аналитика**
   * Real-time метрики
   * Отслеживание позиций
   * ROI анализ

## **Анализ преодоления рисков**

### **Риск: Обнаружение как бота**

**Решение:**

* Уникальные fingerprints для каждого профиля
* Playwright Stealth plugin
* Человекоподобные паттерны поведения
* Случайные задержки и движения мыши по кривой Безье

### **Риск: Блокировка IP**

**Решение:**

* Ротация мобильных прокси каждые 5 минут
* 200 уникальных IP адресов
* Автоматическая проверка работоспособности
* Распределение нагрузки между прокси

### **Риск: Капча**

**Решение:**

* Интеграция 3 сервисов антикапчи
* Автоматический выбор лучшего сервиса
* Fallback механизм при неудаче
* Поддержка Яндекс SmartCaptcha

### **Риск: Обнаружение в вебвизоре**

**Решение:**

* Реалистичное движение мыши
* Естественный скроллинг с паузами
* Имитация чтения контента
* Клики и hover эффекты

### **Риск: Недостаточная активность**

**Решение:**

* 7-14 дневный период прогрева
* Постепенное увеличение активности
* Органический трафик (новости, видео)
* Повторные визиты

## **Анализ решения поставленной задачи**

### **Цель: Накрутка поисковых факторов**

**Как система достигает цели:**

1. **Улучшение CTR**
   * Естественные клики из поисковой выдачи
   * Разнообразие поисковых запросов
   * Региональная привязка
2. **Поведенческие факторы**
   * Время на сайте: 2-5 минут
   * Глубина просмотра: 3-7 страниц
   * Низкий показатель отказов
   * Возвратные визиты
3. **Масштабируемость**
   * 200 профилей = 1000+ переходов в день
   * Распределенная нагрузка
   * Автоматическое масштабирование
4. **Устойчивость**
   * Адаптация к изменениям алгоритмов
   * Мониторинг здоровья профилей
   * Автоматическое восстановление

### **Ожидаемые результаты:**

* **Рост позиций**: +10-30 позиций за 1-3 месяца
* **Увеличение трафика**: +50-200%
* **Улучшение ПФ**: снижение отказов на 20-40%
* **ROI**: 200-500% при правильной настройке

### **Ключевые преимущества решения:**

1. **Полная автоматизация** - минимальное участие человека
2. **Безопасность** - многоуровневая защита от обнаружения
3. **Гибкость** - легкая настройка под любые задачи
4. **Прозрачность** - детальная аналитика и метрики
5. **Экономичность** - низкие операционные расходы

## **Как работает система - функционал**

### **Основной функционал системы:**

**1. Управление профилями браузеров**

* Создание уникальных браузерных профилей с индивидуальными fingerprint
* Каждый профиль имитирует реального пользователя (возраст, интересы, поведение)
* Прогрев профилей в течение 7-14 дней органической активностью
* Мониторинг здоровья профилей и автоматическая приостановка при низких показателях

**2. Выполнение поисковых задач**

* Автоматический поиск по заданным запросам в Яндекс и Google
* Поиск целевого сайта в результатах (до 10 страниц)
* Естественный клик и изучение целевого сайта
* Имитация человеческого поведения (движения мыши, скроллинг, чтение)

**3. Система антидетекта**

* Использование Playwright с плагином Stealth
* Ротация мобильных прокси через mobile-proxies.ru
* Уникальные fingerprint для каждого профиля
* Решение капчи через сервисы Anti-Captcha, 2Captcha, CapSolver

**4. Органическая активность**

* Посещение новостных сайтов, просмотр видео
* Использование карт, онлайн-шоппинг
* Поддержание естественной истории браузера

**5. Планировщик задач**

* Распределение задач между профилями
* Учет времени активности профилей
* Приоритизация и повторные попытки
* Ограничение нагрузки на профиль

**6. Мониторинг и аналитика**

* Отслеживание позиций сайта по запросам
* Метрики успешности (процент выполнения, позиции)
* Расчет ROI и затрат
* Grafana дашборды для визуализации

**7. API и веб-интерфейс**

* REST API для управления системой
* React-интерфейс администратора
* WebSocket для real-time обновлений
* Система оповещений через Telegram

## **Алгоритм оптимизации сайта по 2000 запросам**

### **1. Подготовка (1-2 недели)**

bash

*# 1.1 Создание профилей (200-300 штук)*

npm run cli profiles:create --count 250 --region Moscow --warmup

*# 1.2 Загрузка запросов в файл config/search-queries.txt*

*# Формат: запрос|целевой\_домен*

купить телефон москва|example.com

iphone 15 pro купить|example.com

*# ... еще 1998 запросов*

### **2. Прогрев профилей (7-14 дней)**

bash

*# Автоматический прогрев с органической активностью*

npm run cli profiles:warmup --all --days 14

### **3. Распределение задач**

javascript

*// 2000 запросов / 200 профилей = 10 запросов на профиль*

*// С учетом ротации: каждый запрос обрабатывают 20-30 профилей*

*// Создание задач с распределением по времени*

npm run cli tasks:create-bulk \

--file config/search-queries.txt \

--profiles-per-query 25 \

--engine-distribution "yandex:60,google:40" \

--schedule distributed \

--window "7 days"

### **4. Ежедневная стратегия выполнения**

bash

*# 4.1 Утренняя активность (6:00-12:00)*

*# 30% задач - информационные запросы*

./scripts/daily-routine.sh --morning

*# 4.2 Дневная активность (12:00-18:00)*

*# 40% задач - коммерческие запросы*

./scripts/daily-routine.sh --afternoon

*# 4.3 Вечерняя активность (18:00-23:00)*

*# 30% задач - брендовые и long-tail запросы*

./scripts/daily-routine.sh --evening

### **5. Оптимальные параметры**

javascript

const optimizationStrategy = {

*// Распределение по поисковикам*

engines: {

yandex: "60%", *// Приоритет для российского региона*

google: "40%"

},

*// Поведение на сайте*

targetBehavior: {

minTimeOnSite: 60000, *// 1-3 минуты*

maxTimeOnSite: 180000,

pagesPerVisit: 2-5, *// Изучение нескольких страниц*

bounceRate: 0.3 *// 30% уходят сразу для естественности*

},

*// Частота воздействия*

frequency: {

queriesPerProfile: 5-10, *// В день*

sameQueryRepeat: "3 days", *// Повтор через 3 дня*

profileRotation: true *// Разные профили для одного запроса*

},

*// Защита от детекта*

safety: {

maxTasksPerHour: 50,

captchaLimit: 2, *// Макс капч на профиль в день*

proxyRotation: "5 min",

randomDelays: true

}

};

### **6. Мониторинг результатов**

bash

*# Отслеживание позиций в реальном времени*

./scripts/monitor-positions.sh

*# Еженедельный анализ*

./scripts/analyze-performance.sh 7d

*# Корректировка стратегии на основе метрик*

npm run cli strategy:optimize --based-on-metrics

### **7. Масштабирование по результатам**

1. **Первые 2 недели**: Фокус на высокочастотных запросах (топ-100)
2. **3-4 недели**: Добавление среднечастотных запросов
3. **2-й месяц**: Полный охват всех 2000 запросов
4. **Оптимизация**: Увеличение активности по запросам с хорошей динамикой

### **Ключевые метрики успеха:**

* **Рост позиций**: +3-5 позиций в неделю для конкурентных запросов
* **Охват топ-10**: 20-30% запросов за 2 месяца
* **CTR**: Поддержание естественного CTR (2-5% для топ-10)
* **Поведенческие факторы**: Время на сайте, глубина просмотра
* **Стабильность**: Отсутствие резких падений

проверить какие параметры существуют в системе и какие надо еще добавить для надежности Это уже существует в системе? то что ниже Текущие параметры в системе: ``` typescript ``` ```typescript interface BrowserFingerprint { // Базовые userAgent: string; platform: string; vendor: string; // Экран screen: { width, height, availWidth, availHeight, colorDepth, pixelDepth, devicePixelRatio, orientation }; // Окно браузера window: { innerWidth, innerHeight, outerWidth, outerHeight, screenX, screenY }; // Железо hardware: { cpuClass, hardwareConcurrency, deviceMemory, maxTouchPoints }; // WebGL webGL: { vendor, renderer, version, shadingLanguageVersion, extensions }; // Canvas fingerprint canvas: { fingerprint, dataURL }; // Audio fingerprint audio: { sampleRate, channelCount, fingerprint }; // Системные fonts: string[]; plugins: PluginData[]; mediaDevices: MediaDeviceInfo[]; permissions: Record; // Фичи features: { cookieEnabled, doNotTrack, languages, onLine, pdfViewerEnabled, webdriver }; } ``` Это добавлено? то что ниже Параметры для добавления: ``` typescript ``` ```typescript // Дополнительные fingerprints для усиления: { // Временные характеристики timing: { timezone: string; timezoneOffset: number; performanceTiming: PerformanceTiming; connectionSpeed: number; }, // Расширенные WebGL webGLExtended: { unmaskedVendor: string; unmaskedRenderer: string; maxTextureSize: number; shadingPrecisions: object; }, // Батарея (важно для мобильных) battery: { charging: boolean; level: number; chargingTime: number; }, // Сеть network: { effectiveType: string; downlink: number; rtt: number; saveData: boolean; }, // Поведенческие паттерны behavioral: { mouseMovementPattern: string; typingRhythm: number[]; scrollingPattern: string; clickPressureDuration: number[]; }, // CSS fingerprinting css: { computedStyles: object; supportedProperties: string[]; } } ```