



Владислав Лайков

Senior PHP Developer

Symfony / Laravel · распределённые системы и высокие нагрузки

t.me/vladisdev

[+375 29 163 36 69](tel:+375291633669)

vladlikedev@gmail.com

linkedin.com/in/vlad-laikov-php

Москва · удалённо

Senior backend-инженер, **7+ лет на PHP / Symfony**. Последние **3,5 года** профильная зона — **денежный контур продукта**: платежи, балансы, интеграции; за плечами **10+ платёжных провайдеров** в проде.

Проектирую денежный контур так, чтобы платёж нельзя было потерять, задвоить или списать дважды: уникальность гарантирует сама база, а не код приложения, порядок блокировок снимает дедлоки вместо ретраев поверх симптома, а внутренние и внешние статусы никогда не расходятся. Тяжёлые операции выношу из синхронного цикла в очереди, а способ доставки результата подбираю под задачу.

Меняю живой прод без окна простоя: мажорные апгрейды СУБД и фреймворков, смену диалекта и вынос кусков монолита в сервисы веду поэтапно и обратимо, на кластере **~50 ТБ**. Инфраструктуру при этом не переключиваю на других — **Docker** и **Kubernetes**, **RabbitMQ** и **Kafka**, репликация и шардирование, пайплайны **GitLab CI / GitHub Actions**, алерты в **Grafana**: эпик веду от архитектурного решения до графиков в проде.

ПРОФИЛЬ

КОМПЕТЕНЦИИ: платёжные интеграции и биллинг · **PHP / Symfony · Laravel · PostgreSQL · high-load** и **распределённые системы**

ДОМЕНЫ: **платежи и биллинг** · **финтех-core** · SaaS · MedTech · классифайды · аутсорс

НАВЫКИ

Языки и фреймворки: **PHP 5.6–8.4, Symfony 3.4–7.4, Laravel 6–13**, Doctrine, API Platform, Symfony Messenger, **Go**, Composer / PSR, **ООП, SOLID** · GRASP, паттерны GoF, DDD, CQRS, Hexagonal

Данные и производительность БД: **PostgreSQL 14–18**, MySQL 5.7–8, **Redis, MongoDB, ClickHouse**, Elasticsearch, PgBouncer · планы запросов и составные индексы, партиционирование растущих таблиц, шардирование, логическая репликация, апгрейды под нагрузкой

Инфраструктура и эксплуатация: **Docker, Kubernetes, Linux, AWS** (ECS, EC2, S3), Nginx, **GitLab CI**, GitHub Actions, **Grafana / Loki**, Prometheus, Sentry, ELK · метрики, логи, алерты, разбор инцидентов до первопричины

Распределённые системы и real-time: **Kafka, RabbitMQ**, Symfony Messenger, **Mercure, Centrifugo**, WebSocket, микросервисы, REST API / OpenAPI · приём и оркестрация вебхуков, идемпотентные обработчики, ретрай при сбоях

Качество и тестирование: **PHPUnit, Codeception, Selenium, PHPStan** (level 9), Rector, CS Fixer, GrumPHP · code review, поэтапный рефакторинг легаси без переписывания, тесты на реальных БД

ОПЫТ РАБОТЫ

Апр 2024 — наст. время

Senior PHP Developer · платёжная команда

systeme.io — международная SaaS-платформа · удалённо

Международная SaaS-платформа для онлайн-бизнеса «всё-в-одном» (email-маркетинг, воронки продаж, онлайн-курсы, приём платежей): **15+ млн** зарегистрированных аккаунтов и **500K+** активных бизнесов в **170+ странах**. Высоконагруженная система: порядка **1 млрд** email-отправок в месяц (**12+ млрд** в год), **~2 млрд** записей контактов в базе, миллионы опубликованных лендингов и воронок с трафиком в сотни млн просмотров в месяц. Вынос кусков монолита в сервисы — поэтапно и обратимо.

Стек: **PHP 8.3–8.5, Symfony 6.4–7.4, Doctrine, PostgreSQL 14–18, MongoDB, Redis, RabbitMQ, Kafka, Mercure, Docker, AWS, Grafana / Loki, PHPStan level 9, GrumPHP**

- Заменял клиентский polling на push через **Mercure (SSE)**: тяжёлые операции ушли из синхронного запроса в очереди, а интерфейс получает событие в момент готовности, а не опрашивает сервер по таймеру. Спроектировал под это реестр наблюдателей: новое место опроса подключается одним классом, без правок в бизнес-обработчиках, событие уходит только при реальном изменении ответа. Результат: из системы ушло **~200 тыс. лишних запросов в час** — **~1,5 млрд в год**, около **90% опросного трафика**; самый нагруженный роут: было **~70 тыс. запросов в час**, стало **~4 тыс.**
- Владел **всеми 10+ платёжными интеграциями** платформы. Реализовал **Apple Pay через Stripe** end-to-end и вынес одного из провайдеров **из монолита в отдельный микросервис** — своя база, свой деплой, общий bundle контрактов: новый провайдер подключается, не трогая ядро.
- Спроектировал **движок частичных возвратов**: бизнес дал требование одной строкой — сам определил гранулярность (уровень позиции заказа), четыре сценария маршрутизации вебхука, гвардированную машину статусов и защиту от переплаты.
- Провёл мажорный апгрейд **PostgreSQL 14 → 18** на кластере **~50 ТБ**, не останавливая приём платежей: новая база работала рядом со старой и держалась в синхроне логической репликацией, переключались частями и на любом шаге могли вернуться назад.
- Миграции **PHP 8.3 → 8.5** и **Symfony 6.4 → 7.4 LTS** на живом проде. Дисциплина через тулинг: **PHPStan level 9**, жёсткий CI-гейт на каждый PR, реальные PostgreSQL и Redis в тестах вместо моков.

PHP Developer · команда CORE

Octopays (Mservis) — финтех-платформа · удалённо

Международный платёжный провайдер (PSP-агрегатор) для high-risk сегмента: приём платежей и выплаты на карты и кошельки в **10+ странах**. Высоконагруженная система: ~**5 млн** транзакций в месяц (**60+ млн** в год), **40+** платёжных методов и провайдеров, **200+** активных мерчантов. Core-команда, прод-БД **2 ТБ+**, обработка транзакций в реальном времени, событийная архитектура (**Kafka**, миллионы событий в сутки).

Стек: PHP 7.4—8.2, Laravel 9—10, PostgreSQL, MySQL, Redis, Kafka, Docker, ELK, PHPUnit, PHPStan

- **Балансы с резервом:** пока операция не завершена, её сумма заморожена на балансе — одни и те же деньги нельзя списать дважды при параллельных запросах, на потоке ~**150–200 тыс. операций в день**. Идемпотентность выплат — **уникальным индексом в БД**: дубль отклоняет сама база, а не код приложения; **ни одной двойной выплаты после внедрения**.
- **Deadlock**'и при параллельном пересчёте балансов устранил **порядком блокировок и точечными FOR UPDATE** под READ COMMITTED — предотвращение на уровне схемы доступа, а не retry-обёртка поверх симптома. Ошибки конкурентного доступа в пиковые часы сошли **почти на нет**, ручные разборы по «зависшим» операциям практически исчезли.
- Финансовый модуль: транзакция проходит около десяти внутренних статусов, наружу мерчанту — простое pending / success / fail; внутреннее и внешнее состояние никогда не расходятся. Результат — предсказуемые колбэки при **40+ интеграциях** и **~30–40%** обращений по «непонятым» статусам.
- Миграция **Laravel 9 → 10** и рефакторинг критичных платёжных пайплайнов: тесты **PHPUnit** на финансовые сценарии (покрытие до ~**80%**), **PHPStan level 6+**, observability на **ELK** — диагностика платёжных инцидентов сократилась **с часов до минут**. Участвовал в **смене СУБД MySQL → PostgreSQL** на критичном контуре (**2 ТБ+**): схема под PG-специфичные типы и индексы, поэтапная миграция без остановки приёма платежей, синхронизация легаси-кода с новым диалектом; ключевые запросы по транзакциям стали **быстрее в 2–3 раза**, включая самые медленные.

PHP Developer

Basis33 — аутсорс-разработка для клиентов из ЕС · удалённо

Заказная разработка полного цикла для клиентов из ЕС, команда **25 инженеров**: high-load сервисы с большой кодовой базой, автоматизация бизнес-процессов, **медицинская SaaS-платформа**. Вёл два продукта в несмежных доменах, оба доведены до прода.

Стек: PHP 7.4, Symfony 5.3, API Platform, MySQL, RabbitMQ, Redis, Docker, GitLab CI/CD

- Развивал ядро медплатформы и выпускал новые функции. Закрыв запрос на удалённый приём — **интеграция Zoom API**: врач и пациент попадают в консультацию прямо из календаря записи, без ручной рассылки ссылок.
- Разгонял тяжёлые экраны: планы запросов, **N+1** в явные join'ы, составные индексы под реальные фильтры, нормализация схемы и точечная денормализация горячих выборок, кеш в **Redis** с инвалидацией по доменному событию. Прод-кластер жил с **репликацией и партиционированием** — чтение уходило на реплику.
- Закрывал критичное тестами: **юнит-тесты** на доменные правила, **интеграционные** на связку с БД и внешними сервисами, **функциональные** на API-сценарии — регресс в записи и календаре ловился до релиза, а не на проде.

PHP Developer

Like Systems — классифайды и сбор данных · удалённо

Поисковая SPA-платформа объявлений и сервис автоматического сбора данных.

Стек: PHP 7.1—7.4, Symfony 3.4 → 4.4 LTS, Doctrine, PostgreSQL, Docker, Swagger, GitLab CI

- Участвовал в переводе платформы с легаси-стека на актуальный: **PHP 7.1 → 7.4, Symfony 3.4 → 4.4 LTS** — модули переносили по частям, без остановки сервиса.
- Собрал backend платформы объявлений: доменная модель под поиск и фильтры в **PostgreSQL**, версионизируемый **REST-контракт** в Swagger, автосборка и деплой через **GitLab CI**, сборщик данных с внешних площадок.

ПОМИМО РАБОТЫ

- Веду собственные продукты от идеи до продакшена, без команды за спиной: **DDD / CQRS**, real-time на **Centrifugo**, устанавливаемое PWA.
- **Разработка с AI-агентами**. Предпочитаю **Claude Code** и держу его на коротком поводке: вместе выстроен флоу от документации и спецификации до имплементации бэкэнда, фронтенда и деплоя.

ОБРАЗОВАНИЕ

Минск, Беларусь

Белорусский государственный университет информатики и радиоэлектроники

Факультет компьютерных систем и сетей — Computer Science

Английский — **B1**