Engineer

Кодова база

увесь вихідний код одного застосунку в одному репозиторії

Що це таке

Кодова база, англійською codebase, — це сукупність коду, яку команда редагує, рецензує і випускає як одне ціле. Коли інженери кажуть «кодова база», вони мають на увазі місце, куди потрапляє кожна зміна застосунку: сам код, його тести, скрипти, які його збирають і перевіряють, зміни бази даних і документи, що пояснюють, як усе це складається докупи.

The Twelve-Factor App, методологія побудови вебсервісів, дає найкорисніше робоче визначення, бо перетворює межу на те, що можна перевірити. У централізованій системі контролю версій, як-от Subversion, кодова база — це один репозиторій. У розподіленій, як-от Git, — будь-який набір репозиторіїв зі спільним кореневим комітом, тобто першим комітом в історії, тим, що не має батьківського. Два репозиторії, що виросли з того самого першого коміту, — одна кодова база, хоч би як далеко вони розійшлися. Два, що почалися окремо, — дві кодові бази, навіть якщо файли в них майже однакові.

Поруч із цим визначенням той самий документ ставить друге правило: одна кодова база, багато розгортань. Розгортання — це запущений екземпляр застосунку, і зазвичай їх кілька одночасно. Продакшн — одне, staging — інше, і локальна копія кожного розробника теж рахується. Усі вони працюють на тій самій кодовій базі, часто на різних комітах: у розробника є зміни, до яких staging ще не дійшов, а на staging — зміни, яких продакшн поки не бачив.

Що має бути всередині, випливає з цього. Код і все, що потрібно, щоб зробити з нього робочу програму, належать до кодової бази, зокрема маніфест, який називає кожну залежність і її версію. Самі залежності зазвичай лежать зовні, і їх завантажує менеджер пакетів. Значення, що

змінюються від одного розгортання до іншого, теж лишаються зовні, а понад усе — облікові дані: пароль, закомічений один раз, далі мандрує з кожною копією історії.

Кодова база може містити кілька програм. Монорепозиторій тримає фронтенд, API і код бази даних поруч в одному репозиторії, тож один коміт може змінити все це разом. За правилом кореневого коміту це все одно одна кодова база. Протилежний вибір розносить застосунок по репозиторіях з окремими історіями, і тоді кожен репозиторій — окрема кодова база. Twelve-factor каже про наслідок прямо: якщо кодових баз кілька, це розподілена система, і кожен її компонент — окремий застосунок.

Застосуйте поняття до рішення

Застосуйте «Що таке кодова база?» до конкретного рішення. Ми можемо пов'язати поняття із системою, яку ви створюєте або купуєте. Структура системи, спершу продумана на папері, щоб частини пасували одна до одної, а вартість була відома до першого рядка коду.

Переглянути наші послуги Зв'язатися з нами →

Перевірте припущення за терміном

Використайте пояснення та пов'язані роботи, щоб сформулювати запитання для проєкту. Ми можемо перевірити припущення разом до вибору інструмента. Один інфраструктурний план для 14 підрозділів — 14 дослідницьких підрозділів, охоплених спільним інфраструктурним планом

Один інфраструктурний план для 14 підрозділів Зв'язатися з нами →

Чому це важливо

Межа визначає, що може змінюватися разом. У межах однієї кодової бази зміну API і екрана, який його викликає, можна написати, відрецензувати і злити як одну одиницю. Між двома кодовими базами та сама зміна перетворюється на два pull requests, два релізи й період, коли один бік випереджає інший. Обидва варіанти робочі; ціну визначає те, де провели межу.

Кодова база — це ще й одиниця, на яку поширюються стандарти. Набір тестів, лінтер, форматер, правило для повідомлень комітів і task runner беруть «кодову базу» за свою область дії, а все, що лежить поза нею, лишається поза їхньою увагою.

Ранні рішення застигають у кодовій базі швидше, ніж будь-де. Кожен рядок, написаний після вибору, спирається на цей вибір, тож правило шарів чи планка якості, встановлені в перші тижні, стають формою всього,

що з'являється далі. Скасувати їх пізніше означає переписати код, який наріс довкола.

Для людей кодова база — це те, що отримує новачок. Якщо в ній є все потрібне, щоб зібрати, протестувати й запустити застосунок, новій людині вистачить одного клонування й однієї команди. Якщо знання живуть частково в коді, а частково в чиїйсь пам'яті, перший тиждень іде на те, щоб розібратися, що де.

Як це виглядає на практиці

На одній платформі кодова база — це багатомовний монорепозиторій на Go, TypeScript, SQL, Python і shell, і кожна мова приносить власний спосіб збирати, тестувати й лінтити. Відповіддю стала єдина точка входу: шар go-task, що виріс до 578 іменованих цілей, 50 у кореневому файлі й 528 у файлах із просторами імен під ним, із режимами розробки native, Docker і HTTPS, якістю коду, керуванням базою даних і розгортанням. Будь-хто може виконати `task --list` і побачити весь інструментарій. Сторінка читає це число й як попередження: 578 цілей — більше, ніж будь-хто здатен тримати в голові, тож саме найменування і простори імен роблять усе це придатним до використання.

Інший проєкт почався 25 листопада 2025 року з репозиторію без єдиного файлу, і три рішення перших тижнів трималися й через дев'ять місяців і кілька тисяч комітів: шарова архітектура, доступ до даних через збережені функції і планка нуля попереджень, встановлена ще до того, як з'явилося багато коду. Міркування зі сторінки — серце цієї статті: стандарт, запроваджений на коміті 5 000, — це проєкт із прибирання, а той самий стандарт на коміті 50 — просто те, як працює репозиторій.

Внутрішні стіни кодової бази можуть бути самим дизайном. В описі архітектурного внеску засновниці компанії в межах роботи над NextMariner йдеться про морську платформу, побудовану у три шари: фронтенд на Next.js відповідає за відображення й нічого більше, навмисно тонкий Go API на Huma поверх Fiber маршрутизує, перевіряє й оркеструє, а бізнес-логіка живе у функціях PostgreSQL. Коли правило змінюється, воно змінюється в одному шарі, у SQL. Межі записали й забезпечили їх дотримання під час рев'ю, бо конвенція, за якою ніхто не стежить, перестає бути конвенцією.

Власна кодова база цього сайту проводить межу іншого роду. Сторінки проєктів пишуться в окремому репозиторії, експортуються сюди скриптом і комітяться, щоб сайт збирався самостійно. Усередині цієї кодової бази це згенеровані файли, і інструкції для роботи тут кажуть, що вручну їх не редагують.

Де виникають проблеми

Скопійований код розходиться

Двом застосункам потрібна та сама допоміжна функція, і найшвидше — скопіювати її. За рік існують дві версії з різними виправленнями, і помилка, закрита в одній, живе далі в іншій. Twelve-factor називає кілька застосунків зі спільним кодом порушенням методології і вказує вихід: винести спільну частину в бібліотеку, підключати її через менеджер залежностей, і нехай кожна кодова база вказує версію, яку використовує.

Секрет потрапляє в історію

Рядок підключення або ключ API потрапляє в коміт разом із конфігураційним файлом і відтоді лежить в історії, яку несе кожен клон, навіть після того, як пізніший коміт видалив рядок. Перевірку twelve-factor варто робити перед кожним комітом: чи можна сьогодні опублікувати цю кодову базу як відкритий код, не розкривши облікових даних? Усе, що цю перевірку провалює, належить до середовища кожного розгортання.

Згенерований файл правлять вручну

Файл, створений скриптом, виглядає як будь-який інший вихідний файл, тож хтось виправляє друкарську помилку просто в ньому. Наступний запуск скрипта записує старий текст назад, і виправлення тихо зникає. Позначайте згенеровані шляхи чітко, залиште генератор єдиним, хто їх пише, і спрямовуйте кожне виправлення у вхідні дані генератора.

Внутрішні стіни тоншають

Під тиском дедлайну запит до бази потрапляє просто в обробник HTTP-запиту, бо це найкоротший шлях, і шари, з якими починався проєкт, перестають описувати код. Кожен обхідний шлях малий. Разом вони перетворюють зміну у два рядки на таку, що зачіпає кожен шар. Записане правило допомагає, рецензування допомагає більше, а перевірка, яка зупиняє збирання, коли один шар лізе в інший, допомагає найбільше.

Стандарт покриває половину коду

Лінтер, налаштований для фронтенду, оминає shell-скрипти і SQL, і саме ці файли збирають ті проблеми, які лінтер мав зупиняти. Стандарт для всієї кодової бази мусить назвати кожну мову в ній, і це прихована ціна багатомовного репозиторію.

Форк стає новим продуктом

Команда копіює репозиторій, щоб почати споріднений продукт. Обидві копії зберігають той самий кореневий коміт, тож за правилом twelve-factor вони досі одна кодова база, яка тепер обслуговує два застосунки, і описане вище розходження починається в масштабі цілого репозиторію. Порівняння кореневих комітів показує, яка у вас ситуація:

```
git -C app-a rev-list --max-parents=0 HEAD
git -C app-b rev-list --max-parents=0 HEAD
```

Однаковий вивід означає спільний корінь і спільну кодову базу; різний — дві окремі історії.

Що почитати далі

- I. Codebase — правило «одна кодова база, багато розгортань» і перевірка спільним кореневим комітом, на якій воно тримається.
- II. Dependencies — як оголосити кожну залежність у маніфесті й ізолювати її, щоб кодова база точно казала, що підтягує.
- III. Config — перевірка відкритим кодом для всього, що має лишатися поза кодовою базою.
- gitsubmodules — як Git вбудовує один репозиторій в інший, коли кожен зберігає власну історію.
- git-rev-list — довідка про виведення списку комітів, зокрема кореневих комітів історії.

Словник

Обирайте за фактичними умовами

Опишіть робочий процес, наявні інтерфейси та користувачів, яким потрібна зміна. Ми можемо визначити перевірюваний результат і спосіб передачі, який дасть вашій команді змогу підтримувати його.

Дивитися, скільки це коштує Зв'язатися з нами →

Зробіть термінологію практичною

Запитайте, як «Що таке кодова база?» стосується вашої організації. Конкретного прикладу з поточної системи достатньо для початку розмови.

Переглянути наші послуги

welcome@engineer.company +45 50 10 61 84 Копенгаген, Данія

Візьміть цю сторінку із собою

Візитівка

Дбайливо зроблено з любов'ю — D.

CVR 45969703 · Tagensvej 105C, 3. th, 2200 København N

Авторське право © 2025 – дотепер · Engineer ApS · Усі права захищено.

Надруковано з https://engineer.company/uk/glossary/codebase/