

Nápověda Kaloka

Jak Kaloko nastavit, projít aplikaci s agentem, zkontrolovat, co postavil, a z výsledku vést changelog a dokumentaci. Obrázky pocházejí ze schválených běhů samotného Kaloka, takže ukazují obrazovky tak, jak vypadají dnes. I tahle nápověda je živá dokumentace publikovaná v Kaloku.

Tutoriály

- První běh za deset minut: instalace, přihlášení, průchod flow, sdílení a kontrola
- Navrhněte flow se svým agentem: živé HTML kroky, verze a reference designu

Kontrola a schválení

- Projít běh: obrazovky běhu na plachtě jednu po druhé
- Schválit nebo vrátit běh: schválit kroky, které sedí, a vrátit ty ostatní
- Kontrolovat jen to, co se změnilo: režim kontroly, jeho klávesy a převzatá schválení
- Co je nového: novinky s obrázky před a po
- Předat podepsaný záznam o přijetí: doklad, kdo co převzal, a jak ho ověřit

Průchod a kontroly

- Napsat scénář: flow, jeho kroky a co má každý krok ukázat
- Přidat kritéria a packy kontrol: automatické kontroly, otázky pro evaluátor, packy
- Projít víc prohlížečů a zařízení: WebKit, Firefox, telefony, tablety, tmavý režim
- Projít aplikace pro Android, iOS a desktop: emulátory, simulátory, Electron, macOS a Windows
- Natočit video průchodu: video a průběh každého kroku, jedno video celého běhu
- Spouštět Kaloko v CI a komentovat pull requesty: průchod každého preview nasazení
- Porovnat s referenčním během a ignorovat, co se mění pořad: rozdíly bez hodin a karuselů
- Kontrolovat stránky podle plánu (hostované běhy): prohlížeče Kaloka otevrou vaše stránky každý den nebo týden

Tým a přístup

- Pozvat lidi a hosty: členové, hosté v jednotlivých projektech, čtenáři z vaší domény
- Role a API tokeny: kdo smí co, tokeny pro agenty a CI
- Nastavit firemní přihlášení: přihlašovací domény, dvoufázové ověření, SAML, OIDC a SCIM
- Připojit Jiru, Linear, Slack a Teams: úkoly z kroků a kanály po projektech

Changelog a dokumentace

- Publikovat changelog s obrázky: připravit záznam, zapsat release, nasdílet ho
- Publikovat živou dokumentaci a hlídat zastaralé části: návody, jejichž obrázky drží krok s aplikací
- Exportovat dokumentaci a changelog: Markdown, Docusaurus, VitePress, MkDocs, HTML, PDF nebo JSON

Reference

- Příkazy CLI: všechny příkazy kaloko
- Soubor scénáře: všechny klíče scénáře

- Kritéria, kontroly a packy: všechny typy kontrol a co který pack přidá
- Konfigurační soubor: kaloko.config.yml
- Nástroje MCP: co agenti zvládnou přes MCP

Vysvětlení

- Jak Kaloko odpovídá vašim aplikacím: organizace, produkty, projekty, moduly, prostředí a verze
- Jak vznikají verdikty: kontroly, evaluátor a lidé
- Bezpečnost a vaše data: co opustí váš počítač a kdo to uvidí

Jak Kaloko odpovídá vašim aplikacím

Jeden repozitář je málokdy jedna aplikace. V monorepu bývá aplikací několik, jedna aplikace se táhne přes víc repozitářů, agentura má projekt pro každého klienta a každý tým pojmenovává prostředí jinak. Kaloko proto pracuje s několika pojmy, které jednou přiřadíte ke své situaci.

Pojmy

| Pojem | Co to je | Obvyklé přiřazení | ---|---|---| | Organizace | účet vaší firmy nebo agentury | jedna na firmu | | Produkt | to, co se vydává a verzuje jako celek, s vlastním changelogem a dokumentací | e?shop, administrace, mobilní aplikace, balíček | | Projekt | jednotka práce a přístupu: kdo ho vidí a kdo ho kontroluje | klient, tým, modul, celá aplikace | | Modul | část produktu s vlastní sekcí v dokumentaci a changelogu | košík, účet, fakturace | | Scénář a krok | flow a jeho obrazovky | patří k projektu; může uvést produkt a modul | | Prostředí | kde flow běží | local, preview, staging, produkce, production-cz | | Verze | vydaná verze produktu | z git tagů, package.json, vašeho nástroje na release nebo z kaloko release |

Produkty jsou volitelné

Bez produktů je produktem projekt a jeho changelog je changelog projektu. Tak začíná většina týmů. Až budete potřebovat víc, vypíšete produkty pod products: v kaloko.config.yml a scénáře uvedou svůj přes product: a module:. Nic se nemusí přesouvat.

Projekt může pokrývat několik produktů a na jednom produktu může pracovat víc projektů. Dokumentace a záznamy v changelogu odkazují na kroky, takže produkt a modul znají ze scénáře.

Verze patří produktům

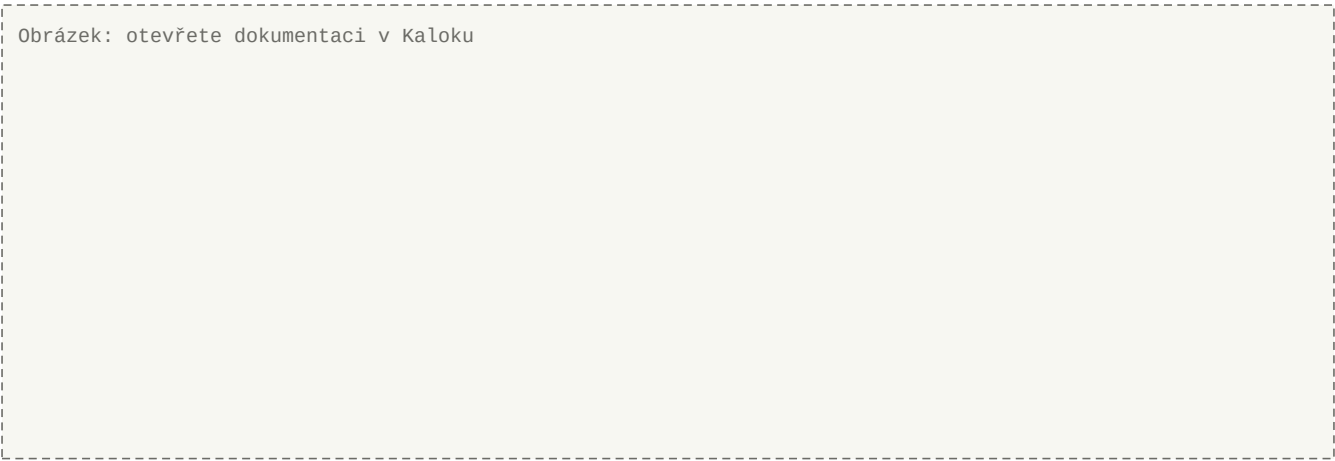
Prostředí jen říká, kde běh proběhl. Běh je svázaný s verzí produktu a dokumentace pro verzi 2.4 ukazuje přijaté běhy verze 2.4. Obrázky pocházejí z prostředí, které produkt určí jako referenční (reference_env), třeba staging u interního nástroje nebo production-cz u portálu jen pro Česko.

Čtyři obvyklá uspořádání

1. Jedna aplikace, jeden tým. Žádné produkty; projekt je aplikace a má jeden changelog. To je výchozí stav.
2. Monorepo s několika aplikacemi. Produkt pro každou aplikaci; projekty podle týmů nebo aplikací.
3. Agentura. Projekt pro každého klienta a produkt pro každou dodávku. Hosté od klienta čtou jen changelog a dokumentaci svého produktu.
4. Velká aplikace s moduly. Jeden produkt s moduly podle týmů. Každý tým se stará o dokumentaci svého modulu a release má jeden společný changelog se sekcí pro každý modul.

kaloko init navrhne produkt pro každou aplikaci ve složce apps/, která má vlastní package.json. kaloko products sync je pošle do Kaloka ještě před prvním vydáním a Produkty v menu aplikace ukážou každý produkt, který čtete.

Obrázek: otevřete dokumentaci v Kaloku



Stránka Produkty · cs · desktop

Každá část funguje sama

Můžete používat jen changelog, jen dokumentaci, jen dokumentaci jednoho modulu, jen obrázky pro web s dokumentací, který už máte (kaloko export), nebo jen e?mail "Co je nového". Nic nevyžaduje celou sadu.

Jak vznikají verdikty

Každé kritérium na plachtě skončí jako splněno, částečně, nesplněno, nebo čeká na člověka. Rozhodují o tom tři druhy soudců v pevném pořadí: kontroly, které počítají, AI evaluátor na to, co chce úsudek, a lidé, kteří mají poslední slovo.

Nejdřív kontroly, které počítají

Co jde změřit, se zkontroluje přímo na živé stránce při zachycení kroku: je tam text, pole má popisek, adresa po kroku sedí, barva odpovídá tokenu z designu, stránka se načte v časovém limitu. Ke každému výsledku zůstane důkaz, třeba prvek, který odpovídal, nebo naměřená hodnota, takže vidíte, proč kontrola prošla nebo neprošla. Tyto kontroly běží bez AI a nikdy nespotřebují žádný dotaz na evaluátor.

Evaluátor na úsudek

Kritéria jako "chybová hláška říká, co má člověk udělat" jdou sémantickému evaluátoru. Výchozí je JEV od TypeSafe. Čte zredukovanou osnovu stránky, ne screenshot, a vrátí skóre mezi 0 a 1. Nad prahem pro splnění (výchozí 0,85) je kritérium splněné, pod prahem pro nesplnění (0,15) nesplněné a mezi nimi čeká, až rozhodne člověk. Prahy nastavíte v kaloko.config.yml.

Kritéria označená vision: true a odpovědi v nejistém pásmu dostanou druhý pohled od evaluátoru, který vidí screenshot. Ten navíc jednou větou řekne, co rozhodlo; na plachtě ji najdete jako "Proč".

Evaluátor běží s vaším vlastním klíčem v každém plánu, nebo přes Kaloko od plánu Team v rámci měsíčního počtu dotazů. Odpovědi se ukládají, takže nezměněnou obrazovku vyhodnotíte znovu bez nového dotazu. Bez klíče i hostované evaluace zůstanou tato kritéria "nehodnoceno", dokud nerozhodne kontrolující. kaloko calibrate porovná evaluátor s verdikty vašich kontrolujících a navrhne prahey.

Rozhodují lidé

Kontrolující krok schválí, vrátí s poznámkou, nebo rozhodne kritérium, kterým si evaluátor nebyl jistý. Nakonec celý běh přijme nebo vrátí. Každé rozhodnutí nese jméno.

- Povinní schvalovatelé (od plánu Team): běh je přijatý, až ho určení lidé schválí.
- Schválení zůstává. Když se nasdílí nový běh, krok, jehož screenshoty a kritéria se od posledního přijatého běhu nezměnily a jehož kontroly pořád procházejí, si ponechá schválení, které mu dal člověk, i s tím, kdo a kdy. V režimu kontroly čeká jen zbytek. Převzaté schválení může kdokoli zrušit. Převzaté schválení nikdy běh nepřijme a nepočítá se za povinné schvalovatele.
- Agenti za lidi neschvalují. Rozhodnutí agenta nebo tokenu jsou označená jako jejich a za rozhodnutí člověka se nepočítají.

Obrázek: otevřete dokumentaci v Kaloku

Režim kontroly · cs · desktop

Proč v tomhle pořadí

Počítání je levné a přesné, proto jde první. Úsudek je drahý a občas se mýlí, proto řeší jen to, co spočítat nejde, a řekne, když si není jistý. Lidé vidí u každého verdiktu důkaz i důvod a za přijetí se zapíše jejich rozhodnutí.

Bezpečnost a vaše data

Kaloko běží tam, kde je váš kód a váš agent: CLI prochází, zachycuje a kontroluje na vašem počítači nebo ve vašem CI. Služba na kaloko.app ukládá to, co nasdílíte, zobrazuje plachtu a sbírá rozhodnutí. Tahle stránka vysvětluje, co kam putuje. Kde jsou data uložena a které firmy je zpracovávají, najdete na stránce o důvěře a bezpečnosti, kterou průběžně aktualizujeme.

Co opouští váš počítač

Dokud nesdílíte, nic. kaloko walk, evaluate a preview pracují lokálně a místní plachta se otevře z disku. kaloko share nahraje běh: screenshoty, zachycené HTML, kritéria s důkazy a videa tam, kde je váš plán uchovává. Soubory se posílají podle otisku obsahu, takže zachycení, které se nezměnilo, se znovu neposílá.

Co vidí evaluátor

U textových kritérií dostane evaluátor zredukovanou osnovu stránky se zamaskovanými e?maily a tokeny, surové HTML nikdy. Screenshot jde evaluátoru na obrázky jen u kritérií označených vision: true a u odpovědí, kterými si textový evaluátor nebyl jistý. S vlastním klíčem jde požadavek z vašeho počítače přímo k danému poskytovateli, bez něj přes Kaloko.

Tajné údaje zůstávají venku

Tajné údaje se do kaloko.config.yml nikdy nepíšou. Konfigurace jen pojmenuje proměnnou prostředí podle vašeho výběru (password_env: STAGING_PASSWORD) a hodnota se načte z prostředí, z .env projektu nebo z ~/.config/kaloko/.env. Hodnoty těchto proměnných se ze zachyceného HTML, textu stránky i výstupu konzole odstraní dřív, než se cokoli uloží. Přihlášení uložena přes kaloko auth save zůstávají na vašem počítači a nikdy se nenahrávají.

Produkce jen pro čtení

Prostředí označené jako produkce je v CLI vždy jen pro čtení: veřejné stránky jako návštěvník, žádné přihlašování do administrací, žádná vytvořená data. Aby tam scénář mohl běžet, musí uvést readonly: true.

Běhy vyprší

Každý sdílený běh má datum expirace, výchozí je 30 dní a nejvýš retence vašeho plánu, a potom se smaže. kaloko keep běh ponechá natrvalo; v plánech Business a Enterprise přijaté běhy nevypřší, dokud plán trvá. Od plánu Team si běh můžete kdykoli stáhnout jako ZIP přes kaloko export.

Jak se servírují soubory

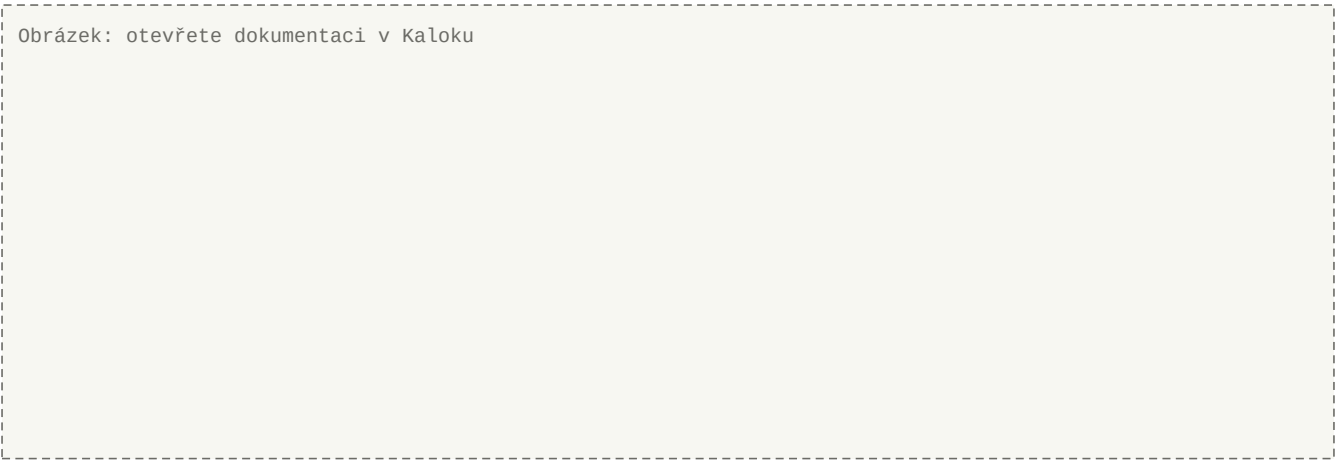
Nahrané soubory se servírují z oddělené domény přes krátkodobě podepsané odkazy. Zachycené HTML se ukazuje v sandboxu bez skriptů.

Kdo se dostane dovnitř

Lidé se přihlašují jednorázovým odkazem z e?mailu nebo přes Google, Microsoft či GitHub a jako druhý krok si můžou přidat passkey nebo autentizační aplikaci. Enterprise přidává firemní přihlášení přes SAML nebo OIDC a SCIM.

Co člověk smí, určuje role: viewer čte, komentuje, schvaluje a vrací; designer a vývojář / tester navíc nahrávají; admin navíc spravuje členy, tokeny a nastavení. Vieweři jsou v každém plánu zdarma. API tokeny mají vlastní rozsah a platnost a token, který CLI dostane přes kaloko login, platí nejvýš 30 dní.

Obrázek: otevřete dokumentaci v Kaloku



API tokeny v Nastavení · cs · desktop

Na co se ptá bezpečnostní prověrka

Stránka o důvěře a bezpečnosti odpovídá, kde jsou data, jak jsou šifrovaná, kdo je zpracovává a jak daleko je příprava na SOC 2. Kvůli nahlášení zranitelnosti nebo smlouvě o zpracování osobních údajů napište na kontakt, který tam najdete.

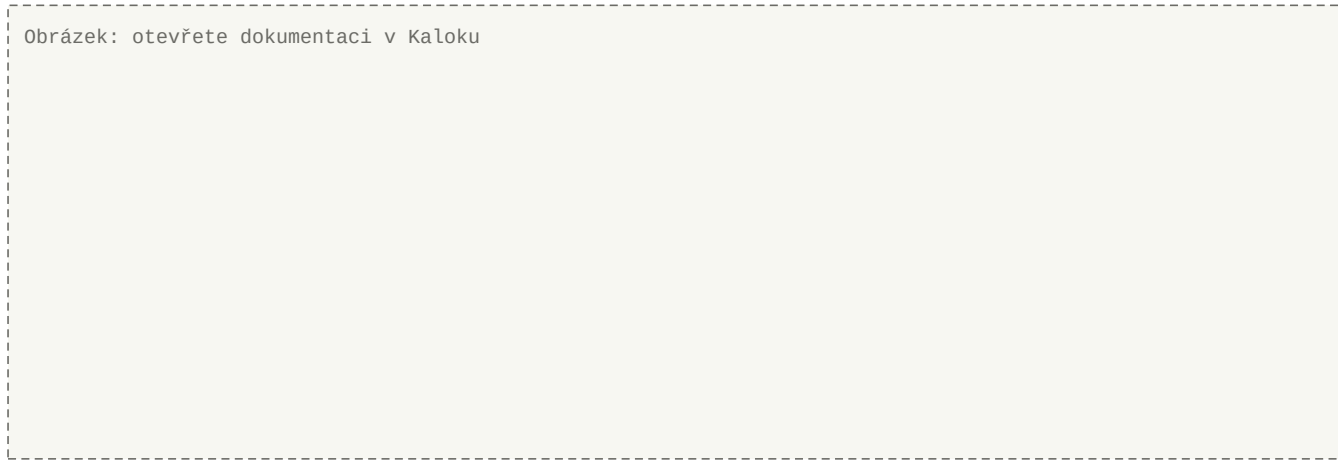
Předat podepsaný záznam o přijetí

Když se klient nebo auditor zeptá, kdo release převzal, dejte mu záznam o přijetí běhu. Je v něm, kdo co a kdy schválil, výsledky kritérií a otisk každého screenshotu, a Kaloko ho podepíše. Záznamy jsou součástí plánů Business a Enterprise.

Stáhněte ho

1. Otevřete přijatý běh a jeho panel Záznam o přijetí.

Obrázek: otevřete dokumentaci v Kaloku



Záznam o přijetí u přijatého běhu · cs · desktop

2. Pro lidi zvolte Stáhnout PDF, pro systém, který záznamy archivuje, JSON. PDF nese stejný JSON uvnitř. Běh, který přijatý není, záznam nemá. Nejdřív ho přijměte, nebo o to požádejte toho, kdo to má udělat.

Ověřte záznam, který jste dostali

Záznam ověří kdokoli i bez účtu. Vývojáři spustí `kaloko signoff --verify zaznam.pdf`; příkaz zkontroluje podpis proti veřejnému klíči Kaloka a řekne, jestli se v záznamu něco změnilo. Stačí upravit jediný znak a ověření neprojde.

Přidat kritéria a packy kontrol

Každý krok má svá kritéria: co musí obrazovka splňovat. Kaloko zná tři druhy a dobrý scénář použije ten nejjednodušší, který na otázku odpoví.

Dá se spočítat: deterministická kontrola

Co se dá spočítat nebo přechíst ze stránky, se ověří přímo ve stránce a k výsledku se přiloží důkaz:

```
criteria:
- id: AC1
text: Košík ukazuje dvě položky.
check: { type: deterministic, assert: count, selector: ".cart-item", min: 2, max: 2 }
- id: AC2
text: Po zaplacení se zobrazí poděkování.
check: { type: deterministic, assert: url_matches, pattern: "/thanks$" }
- id: AC3
text: Tlačítko Zaplatit má barvu značky.
check: { type: deterministic, assert: style_equals, selector: "button.pay", property:
background-color, token: color.brand.primary }
```

Další kontroly čtou text, atributy, HTTP status, přetečení na telefonu nebo hodnotu, kterou stránka vystavuje (js_equals). Celý seznam je v referenci kontroly a packy. Skript kroku nikdy nemusí nechávat značku, kterou by kritérium hledalo.

Otázka: sémantická kontrola

Co chce úsudek, dostane evaluátor jako otázku ke stránce:

```
- id: AC4
text: Hlavní výzva k akci je zřejmá.
check: { type: semantic, question: Ukazuje stránka jednu zřetelnou hlavní výzvu k akci?,
scope: main }
```

Evaluátor si přečte zredukovanou osnovu stránky a odpoví skóre. Jasně ano projde, jasně ne neprojde a všechno mezi tím rozhodne člověk. Když jde o vzhled (rozložení, obrázky, "vypadá to jako naše značka"), přidejte ke kontrole vision: true: na screenshot se pak podívá ještě model, který vidí obrázky. Používejte to jen u kroků, kde je to potřeba.

Rozhodne člověk: ruční kontrola

```
- id: AC5
text: Právníci schválili znění souhlasu.
check: { type: manual }
```

Ruční kritérium zůstane "nehodnoceno", dokud o něm kontrolující nerozhodne na plachtě.

Packy kontrol

Pack přidá ke kroku hotovou sadu kritérií, takže běžné kontroly nepíšete ručně:

```
nodes:
- id: home
packs: [seo, a11y, perf, console]
```

| Pack | Co kontroluje | |---|---| | seo | jeden H1, titulek a popis, canonical, hreflang, Open Graph, indexaci, html lang | | usability | účel, jednu hlavní akci, texty odkazů, pole bez popisku | | a11y | alternativní texty, názvy, popisky, nadpisy, kontrast, landmarky, průchod tabulátorem a viditelný focus | | perf | LCP, CLS, TTFB a váhu stránky naměřené při průchodu | | console | jestli chyby v konzoli souvisejí s tím, k čemu krok slouží | | errors | chybové stránky: skutečný status 4xx nebo 5xx, cestu zpět, žádný výpis zásobníku | | app | obrazovky aplikací: popisky ovládacích prvků, velikost dotykových ploch, useknutý text, pády | | design | implementaci proti přijatému

návrhu: pixely, strukturu, tokeny, kontrast |

Limit změníte vlastním kritériem se stejným id, jaké má pack (třeba AC401 pro LCP); vaše kritérium má přednost. Jak se z výsledků skládá verdikt, vysvětluje stránka jak vzniká verdikt.

Schválit nebo vrátit běh

Teprve vaše rozhodnutí udělá z "agent říká, že je hotovo" opravdu hotovou věc. Rozhodujete po krocích a pak za celý běh.

Rozhodněte o každém kroku

1. Otevřete krok a přečtěte si jeho kritéria. Zelené kritérium zkontrolovalo Kaloko; u kritéria, které potřebuje člověka, to stojí přímo u něj a rozhodnutí je na vás.

Obrázek: otevřete dokumentaci v Kaloku

Krok s kritérii a komentáři · cs · desktop

2. Když krok sedí, zvolte pod komentáři Schválit. Když ne, napište, co se má změnit, a zvolte Zamítnout: "celková cena poskočí, když se načte cena", ne "oprav to". Agent si poznámku přečte a podle ní pokračuje.

Rozhodněte o běhu

1. Když jsou schválené všechny kroky, zvolte Přijmout běh. Běh se stane obrázkem této verze: novinky i dokumentace od té chvíle ukazují jeho obrazovky.

2. Pokud se má něco nejdřív změnit, zvolte Vrátit běh. Kdo ho sdílel, dostane vaše poznámky; až nasdílí další běh, komentář v pull requestu ukáže na něj.

Obrázek: otevřete dokumentaci v Kaloku

Běh na plachtě · cs · desktop

Příště to bude kratší

V dalším běhu si kroky, jejichž obrazovky se nezměnily, vaše schválení ponechají. Díváte se jen na to, co je nové nebo jiné, a na plachtě je vidět, která schválení se přenesla a z kterého běhu.

Agenti mohou komentovat a navrhovat, ale běh přijímá jen člověk.

Projít víc prohlížečů a zařízení

Běh ve výchozím stavu zachytí každý krok v Chromiu, v každém jazyce a viewportu scénáře. Přidat můžete další prohlížeče, předvolby zařízení, tmavý režim a několik dalších nastavení. Každá kombinace je samostatný snímek, takže přidávejte jen to, o čem úkol je: chyba v Safari potřebuje WebKit u dotčených kroků, ne všechna nastavení u všech kroků.

Ve scénáři

```
viewports: [iphone-se, iphone-15, pixel-8, ipad-landscape, laptop]
browsers: [chromium, webkit, firefox]
color_schemes: [light, dark]
reduced_motion: [no-preference, reduce] # volitelné
forced_colors: [none, active] # volitelné, vysoký kontrast ve Windows
display_modes: [browser, standalone] # volitelné, stránka jako nainstalovaná webová aplikace
```

Výchozí hodnoty pro celý projekt patří pod walk: v kaloko.config.yml. Příkazová řádka má přednost před scénářem a scénář před konfigurací.

Pro jeden běh

```
npx kaloko start --scenario qa/flows/checkout.yml --env staging \
--browsers chromium,webkit --color-schemes light,dark --viewports
iphone-se,pixel-8,desktop-hd
npx kaloko walk # jeden průchod pro každý prohlížeč a zařízení
npx kaloko walk --browsers webkit # jen snímky běhu ve WebKitu
```

Předvolby zařízení

npx kaloko devices vypíše předvolby: telefony (iphone-se, iphone-15, pixel-8, galaxy-s24...), skládací telefony (fold-folded, fold-open), tablety (ipad-portrait, ipad-landscape...), počítače (laptop, desktop-hd, desktop-wide) a tv-1080p. Přípona -landscape otočí telefon, skládačku nebo tablet na šířku (iphone-15-landscape). Předvolba nastaví velikost, poměr pixelů, dotyk a user agent zařízení. Odhalí chyby rozložení, zbytek najde až skutečné zařízení.

Nainstalujte prohlížeče

WebKit (jádro Safari) a Firefox stáhnete jednou na každý počítač nebo job v CI:

```
npx kaloko browser install --browsers webkit,firefox
```

npx kaloko doctor vypíše prohlížeče, které vaše scénáře potřebují, a řekne, jak doinstalovat chybějící. V CI na Linuxu přidejte --with-deps kvůli systémovým knihovnám.

Co čekat

- Několik kontrol běží jen v Chromiu: názvy ze stromu přístupnosti, viditelný focus a posuny rozložení, ve WebKitu navíc průchod tabulátorem. V ostatních prohlížečích jsou "nehodnoceno" i s důvodem a krok kvůli nim není částečně splněný.
- Kontrast se kontroluje pro každé barevné schéma zvlášť, takže chyba kontrastu v tmavém režimu neprojde na tmavém snímku.
- Běh odmítne víc než 400 kombinací.

Na plachtě najdete přepínač prohlížeče, barevného schématu a dalších nastavení vedle přepínače jazyka. Detail kroku s předvolbou zařízení vykreslí jeho rámeček a volba Vedle: ukáže stejný krok v jiném prohlížeči nebo schématu vedle něj.

Nastavit firemní přihlášení

Pro adminy organizace. Kaloko se přizpůsobí přihlášení, které ve firmě už máte: osobní přihlášení v každém plánu, povinný druhý krok pro celou organizaci od plánu Business a vašeho poskytovatele identit se SCIM v plánu Enterprise.

Přihlášení v každém plánu

Lidé se přihlašují jednorázovým odkazem z e-mailu, nebo přes Google, Microsoft či GitHub. Každý si v nastavení účtu může přidat passkey nebo autentizační aplikaci jako druhý krok.

Obrázek: otevřete dokumentaci v Kaloku

Přihlašovací stránka · cs · desktop

V plánu Business může admin v Nastavení -> Zabezpečení vyžadovat druhý krok po celé organizaci a nastavit limity relací. Kdo druhý krok nemá, musí si ho přidat, než bude pokračovat.

Nejdřív ověřte doménu

Firemní přihlášení i SCIM se týkají jen adres z ověřených domén.

1. Otevřete Nastavení -> Lidé a přidejte doménu v části Domény pro přihlášení.
2. Přidejte do DNS záznam TXT, který Kaloko ukáže, nebo doménu prokažte přihlášením přes Google Workspace či Microsoft 365.
3. Stiskněte Ověřit. Dokud doména není ověřená, nikdo se přes ni nepřipojí.

Připojte poskytovatele identit (Enterprise)

1. Otevřete Nastavení -> Zabezpečení -> Firemní přihlášení.
2. Zvolte SAML (Okta, Entra ID, Google Workspace a další) a vložte metadata poskytovatele, nebo OIDC s issuerem, client ID a secretem. Metadata Kaloka i s certifikátem najdete na stejné stránce.
3. Stiskněte Vyzkoušet přihlášení. Ukáže, co poskytovatel poslal a jestli by to Kaloko přijalo; test nikoho nepřihlásí.
4. Přihlášení vynuťte. Lidé z ověřených domén se pak přihlašují jen přes vašeho poskytovatele. Admini s passkey nebo autentizační aplikací mají cestu dovnitř i v den, kdy poskytovatel nefunguje.

Další volby na stejné stránce:

- Šifrované assertion: vytvořte pár klíčů organizace; nový pár nechá starý fungovat, dokud ho neodeberete.
- Jednotné odhlášení: odhlášení u poskytovatele odhlásí i z Kaloka a naopak.
- Přihlášení z dlaždice aplikace u poskytovatele, vypnuté, dokud ho nezapnete, s výchozí stránkou, kam lidé přijdou.

- Druhý krok u vašeho poskytovatele může Kaloko uznat jako svůj, pro přihlášení přes vašeho poskytovatele. Hosté z jiných firem se přihlašují jako dřív a vidí jen projekty, do kterých byli pozváni.

Zřizujte lidi přes SCIM (Enterprise)

Základní adresu SCIM a token z Nastavení vložte do Okty nebo Entra ID. Přiřazení lidé přibudou s výchozí rolí své domény, skupiny se mapují na projekty a role a odebrání člověka ukončí jeho relace, zneplatní jeho tokeny a vyřadí ho ze všech projektů. Admina SCIM nikdy nevytvoří.

Omezte, odkud se lidé připojují (Enterprise)

V Nastavení -> Zabezpečení povolíte rozsahy IP adres pro aplikaci a zvláště pro API, CLI a MCP. Audit log může proudit na podepsaný webhook, do Splunku nebo na endpoint kompatibilní s Datadogem.

Kde jsou vaše data a kdo je zpracovává: Bezpečnost a data.

Porovnat s referenčním během a ignorovat, co se mění pořad

Před vydáním chcete znát jednu odpověď: co vypadá jinak než verze, kterou tým přijal, a jestli je to záměr. Dobrý běh scénáře se stane jeho referencí a každý další běh se s ní porovná.

Nastavte referenci

Na plachtě otevřete běh, se kterým jste spokojení, zvolte Přijmout běh a potom Nastavit jako referenci. Každý scénář má jednu referenci a ta se změní, jen když někdo nastaví novou. Z terminálu:

```
npx kaloko reference --scenario checkout --baseline --run <run-id>
npx kaloko reference --scenario checkout # ukáže současnou
```

Porovnejte běh

Na plachtě zvolte Porovnat a referenci (nebo předchozí běh, případně jakýkoli jiný běh scénáře). Každý krok dostane štítek s podílem změněných pixelů a označí se kritéria, kterým se změnil verdikt nebo důkaz.

Obrázek: otevřete dokumentaci v Kaloku

Běh porovnaný s referencí · cs · desktop

V detailu kroku se na změnu podíváte čtyřmi způsoby: Vedle sebe, Rozdíl, Posuvník (táhnete čáru přes oba screenshoty) a Prolnutí (jeden přechází do druhého). Klávesa G ukáže jeden krok ve všech jazycích a velikostech obrazovky najednou.

Stejně porovnání jako text, pro agenta nebo poznámky k vydání:

```
npx kaloko compare # aktuální běh proti referenci, jinak proti předchozímu běhu
npx kaloko compare --run <id> --baseline <id> --json
```

Ignorujte, co se mění pořad

Hodiny, reklamy, karusely a "před 3 minutami" se mění při každém běhu a skutečné rozdíly v nich zapadnou. Vynechte je z porovnání pixelů pomocí ignore: u kroku:

```
nodes:
- id: home
ignore:
- { selector: ".ticker", note: živé ceny }
- { rect: [0, 620, 1280, 180], viewport: desktop }
```

Hledat je ručně nemusíte:

```
npx kaloko walk --stability 3 # tři screenshoty každého snímku krátce po sobě
npx kaloko stability # co se pohnulo, ve srovnání s dřívějšími běhy scénáře
npx kaloko stability --apply # zapíše návrhy do ignore:
```

Než návrhy použijete, podívejte se na screenshoty: oblast, která se změnila, protože se opravdu změnila stránka, není šum.

Šum mohou označit i kontrolující. V detailu kroku na kaloko.app zvolí Označit proměnlivou oblast a nakreslí na screenshot obdélník. Ten se pak čárkovaně ukáže u každého běhu flow a rozdíl na plachtě ho hned vynechá. Do scénáře tyto oblasti dostanete příkazem:

```
npx kaloko stability --pull
```

Dotčené kroky pak projděte znovu: snímek si zapamatuje, které oblasti vynechal.

Připojit Jiru, Linear, Slack a Teams

Pro adminy organizací v plánu Team a vyšších. Po připojení lidé založí z vráceného kroku úkol přímo na plachtě a každý projekt dá vědět do svého kanálu, co se stalo.

Jira nebo Linear

1. Otevřete Nastavení -> Integrace a připojte Jira Cloud (přihlášením přes Atlassian, nebo vložení API tokenu) či Linear (přihlášením, nebo vložení API klíče).
2. Každý projekt v Kaloku přiřaďte k projektu a typu úkolu v Jiře, nebo k týmu v Linearu. Výchozí řádek pokryje projekty, které nevyjmenujete.
3. U Jiry přidejte do svého webu Jira webhook, který vám Kaloko ukáže, aby dokončený úkol hned uzavřel svůj komentář v Kaloku. Bez něj to Kaloko kontroluje každých 30 minut.

Lidé pak u kroku zvolí Nové issue. Úkol dostane screenshot, kritérium a poznámky kontrolujících i s odkazem zpět.

Kanály ve Slacku nebo Teams

1. V Nastavení -> Integrace přidejte k projektu kanál: pro Slack tlačítkem Add to Slack, pro Microsoft Teams adresou toku z Workflows.
 2. Zaškrtněte události, které kanál chce: nový běh, vrácení, přijetí, komentář, zmínka, zapsaný release.
 3. Nastavte tiché hodiny, když kanál nemá od Kaloka v noci nic slyšet; zprávy počkají, až skončí.
- Omezený projekt hlásí jen do kanálů, které ho výslovně jmenují. Kanál ve Slacku pro celou organizaci funguje dál.

Exportovat dokumentaci a changelog

Váš web s dokumentací už možná má vlastní build, nebo klient chce návod jako PDF. Kaloko export zapíše dokumentaci i changelog s obrázky jako obvyčejné soubory, takže nic nezůstane zamčené v Kaloku.

Vyberte formát

```
npx kaloko export --format docusaurus --out website/docs
```

| --format | Co dostanete | Plán | |---|---|---| | md | Markdown se složkou assets/ s obrázky, pro jakýkoli statický web | každý plán | | json | changelog jako strukturovaný JSON | každý plán | | docusaurus | složka pro Docusaurus s front matter, kategoriemi a postranním menu | od plánu Business | | vitepress | složka pro VitePress s konfigurací | od plánu Business | | mkdocs | složka pro MkDocs s mkdocs.yml | od plánu Business | | html | HTML pro offline čtení: stránka na každý návod, changelog a stránka pro tisk | od plánu Business | | pdf | jedno PDF vytištěné prohlížečem | od plánu Business |

Z odkazů na kroky se stanou obvyčejné obrázky v jazyce a velikosti obrazovky exportu.

Vyberte, co do exportu patří

- --what docs nebo --what changelog vyexportuje jen jedno z nich; výchozí jsou obě.
- --version 2.4.0 vyexportuje dokumentaci vydání místo té nejnovější.
- --lang cs vyexportuje jeden jazyk.
- --product shop vybere produkt, když jich konfigurace má víc.
- --out určí složku nebo soubor, kam se export zapíše.

Publikované, nebo lokální

Bez dalších voleb export vezme to, co jste publikovali přes kaloko docs publish. S --local vezme soubory ve vašem repozitáři, aniž by je publikoval. Hodí se to, když chcete změnu zkontrolovat dřív, než půjde ven.

Udržujte web s dokumentací v souladu

Existující web si nechá vlastní build. Buď do jeho složky vyexportujete a výsledek commitnete, nebo export spustíte v jeho CI před buildem:

```
npx kaloko export --format vitepress --what docs --out docs-site/guide
```

Token v CI potřebuje jen čtení; viz Role a API tokeny.

Stáhněte export ze stránky dokumentace

Čtenáři, kterým vyhovuje soubor, použijí Stáhnout na stránce dokumentace nebo changelogu. Nabídne stejné formáty; v plánech pod Business jsou i tam jen Markdown a JSON.

Každý export se otevře i bez Kaloka. Odkud stránky pocházejí a jak je držet aktuální, popisuje Publikovat živou dokumentaci a hlídat zastaralé části.

Kontrolovat stránky podle plánu (hostované běhy)

Marketingový web se mění několikrát týdně a nic z toho neprochází vašimi pull requesty. Hostovaný běh otevře stránky scénáře jen pro čtení v prohlížečích Kaloka každý den nebo týden, pořídí screenshoty celých stránek ve všech jazycích a velikostech obrazovky, zkontroluje, co se dá ze stránky přečíst, a nasdílí běh pod vaším jménem. CLI nemusí nikdo spouštět. Hostované běhy patří k plánu Business (2 000 načtených stránek měsíčně) a Enterprise (10 000).

1. Připravte scénář jen pro čtení

Kroky scénáře otevírají stránky. Začít můžete přímo od stránek:

```
npx kaloko draft https://example.com/ https://example.com/pricing
```

2. Naplánujte ho

```
npx kaloko schedule add --scenario qa/flows/public.yml --env production --every day --at 5
npx kaloko schedule add --scenario qa/flows/public.yml --env production --every week
--weekday mon
```

--at je hodina v UTC. --every manual vytvoří plán, který poběží, jen když ho spustíte.

Fungují i stránky za přihlášením. Přihlašovací údaje HTTP Basic z prostředí se pošlou jednou a uloží zašifrované. Pro stránky přihlášeného uživatele si člověk uloží relaci přes `npx kaloko auth save --env production --account member` a plán ji použije s `--account member`.

3. Spravujte ho

```
npx kaloko schedule list # příští a poslední běh, stránky využité tento měsíc
npx kaloko schedule run <id> # spustit hned
npx kaloko schedule pause <id>
npx kaloko schedule resume <id>
npx kaloko schedule remove <id>
```

Stejné plány najdete na `kaloko.app` v Nastavení -> Integrace.

Přečtete si výsledek

Běh se objeví mezi ostatními, porovnaný s referencí scénáře, se stejnou plachtou, komentáři i schvalováním. Kontroly selektorů, textů, titulků a meta tagů se vyhodnotí. Kritéria, která potřebují plný průchod, zůstanou "nehodnoceno" i s důvodem.

Co hostované běhy neumějí

Stránky jen otevrou a přečtou. Nic neklikají, nepíší, neodesílají ani nevytvářejí, takže kroky, které dělají víc než otevření stránky, vynechají a vyjmenují. Plný průchod potřebují i nativní aplikace, místní nebo neveřejné adresy, přihlašovací údaje posílané v hlavičkách a různé základní adresy pro jednotlivé jazyky.

Pro ně plán vyexportujte jako workflow pro GitHub Actions, které scénář projde s plným CLI:

```
npx kaloko schedule export github --scenario qa/flows/checkout.yml --env staging --every week
```

Workflow vypíše, které tajné proměnné potřebuje.

Jak se stránky počítají

Počítá se každá načtená stránka: každý krok v každém jazyce a velikosti obrazovky. Scénář s 10 stránkami ve 2 jazycích na desktopu a mobilu načte za jeden běh 40 stránek, takže denní běh spotřebuje zhruba 1 200 stránek měsíčně.

Pozvat lidi a hosty

Běh k něčemu je, až když ho uvidí ti, kdo rozhodují. Přiveďte kolegy do organizace, pozvěte klienta do jednoho projektu, nebo nechte číst každého z firemní domény. Vieweři jsou v každém plánu zdarma, takže nemusíte šetřit tím, kdo se smí dívat.

Pozvěte kolegu do organizace

1. Otevřete Nastavení -> Lidé a zvolte Pozvat do organizace.

Obrázek: otevřete dokumentaci v Kaloku

Členové v Nastavení · cs · desktop

2. Zadejte e-mail a vyberte roli. Viewer čte, komentuje a schvaluje; ostatní role popisuje stránka Role a API tokeny.

3. Člověk dostane e-mail s odkazem a přes něj se k organizaci připojí.

Seznam v Nastavení ukazuje každého člena s rolí a s tím, kdy byl naposledy vidět, a nepřijaté pozvánky s tím, kdo je poslal a kdy vyprší.

Pozvěte někoho do jednoho projektu

Klienti, externisté i kolegové z jiných týmů často potřebují jeden projekt, ne celou organizaci.

1. Otevřete projekt a zvolte Sdílet.

2. Napište e-mailovou adresu, klidně i mimo vaši organizaci, a vyberte roli.

3. Zvolte Pozvat. Pozvánka platí 14 dní.

Adresa mimo organizaci přijde jako host. Host vidí jen projekty, do kterých byl pozván: ne ostatní projekty, ne lidi z organizace, ne nastavení ani fakturaci. Dialog Sdílet také ukáže, kdo už přístup má a jestli je projekt otevřený všem členům, nebo omezený jen na uvedené lidi.

V plánu Business můžete hostovi nastavit datum, kdy přístup skončí; ten den projekt opustí a přijde mu e-mail. Hosté s rolí designer nebo vývojář / tester patří k plánu Enterprise; v ostatních plánech je host viewer.

Pust'te dovnitř firemní doménu

Místo zvaní kolegů po jednom přidejte firemní doménu v Nastavení -> Lidé -> Domény pro přihlášení. Kontrolu nad ní prokážete TXT záznamem, který Kaloko ukáže; dokud doména není ověřená, nikdo se přes ni nepřipojí. Pak se každý s e-mailem na ověřené doméně přihlásí s výchozí rolí domény (viewer, pokud nevolíte jinou). Domény veřejných schránek, třeba gmail.com, si přivlastnit nejde.

Čtenáři z vaší domény

V plánu Business přepínač Čtenáři z naší domény na stejném místě pustí každého, kdo se přihlásí z ověřené domény, ke changelogu a dokumentaci všech produktů, i když je projekt omezený. Vstoupí jako vieweři a místo nezabírají. Více na stránce Publikovat changelog s obrázky.

Kdo zabírá místo

Vieweři nikdy. Designer, vývojář / tester a admin zabírají jedno místo, jednou za celou organizaci, ať jsou to členové, nebo hosté. Pozvánka s takovou rolí drží místo, dokud ji člověk nepřijme nebo ji nezrušíte.

Publikovat changelog s obrázky

Vaše přijaté běhy už obsahují každou obrazovku vydání. Kaloko z CHANGELOG.md ve vašem repozitáři udělá stránku s obrázky před a po, kterou si přečtou klienti, podpora i marketing. Takhle vzniká i changelog samotného Kaloka a je veřejný: kaloko.app/p/sinfin/kaloko/changelog.

Jednou to nastavte

```
npx kaloko init --docs=changelog
```

Přidá CHANGELOG.md (Keep a Changelog) a blok changelogu do kaloko.config.yml. Repozitář s několika aplikacemi je vypíše pod products; npx kaloko products sync je dá do Kaloka ještě před prvním vydáním.

Dejte do záznamu obrázky

Obrázek odkazuje na krok místo na soubor:

- Ceník srovnává všechny plány v jedné tabulce.

```
![Předtím](kaloko:KALOKO-public/pricing?version=prev) ![Potom](kaloko:KALOKO-public/pricing)
```

KALOKO-public je id scénáře a pricing id kroku. Bez ?version= odkaz ukáže nejnovější přijatou obrazovku; ?version=prev tu z předchozího vydání.

Připravte koncept záznamu

```
npx kaloko changelog draft # vypíše záznam Unreleased
```

```
npx kaloko changelog draft --apply # zapíše ho do CHANGELOG.md
```

Koncept posbírání pull requesty a commity od posledního vydání, roztřídí je do Added, Changed, Fixed a Removed a přidá obrázky před a po u obrazovek, které přijaté běhy ukazují jinak. Řádky pak přepíšete pro lidi: co teď můžou udělat, jedna změna na řádek. --since <tag> začne od jiného vydání.

Zapište vydání

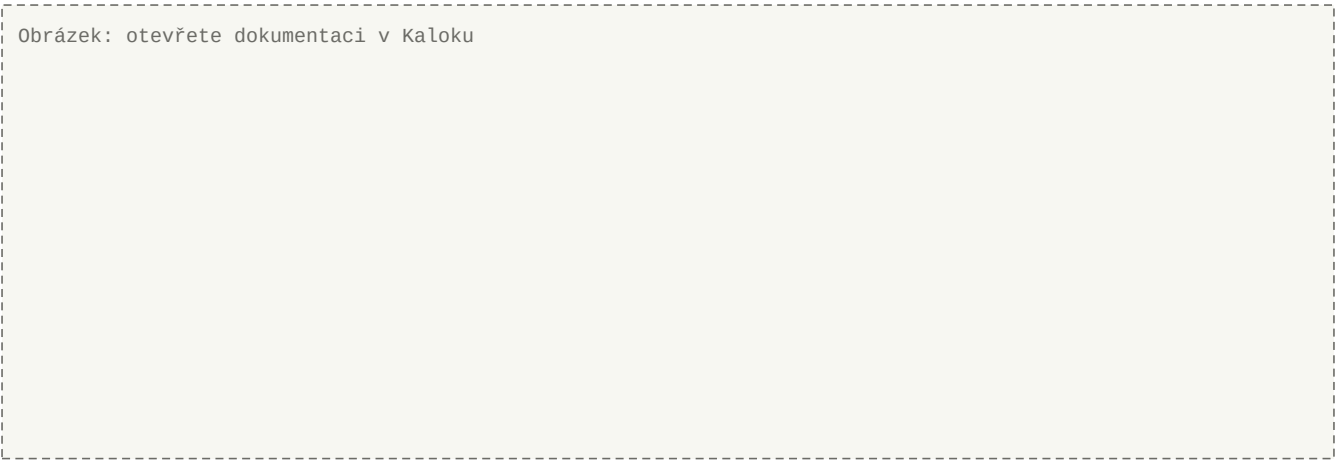
```
npx kaloko release 2.4.0
```

Z Unreleased se stane 2.4.0, vydání se zapíše i s přijatými běhy, které dávají obrázky, a příkaz vypíše adresu stránky. Bez čísla verze si ji Kaloko přečte z gitových tagů, package.json, release-please nebo changesets.

Když vydání dělá release-please, changesets nebo semantic-release, píše CHANGELOG.md dál ten nástroj a Kaloko ho jen čte. npx kaloko ci github --release přidá workflow, které zapíše každé zveřejněné vydání na GitHubu.

Kdo ho čte

Obrázek: otevřete dokumentaci v Kaloku



Stránka Produkty · cs · desktop

Changelog najdete v aplikaci pod Produkty. Čte ho každý, kdo vidí některý z projektů produktu, i hosté. V plánech Business a Enterprise může admin těch projektů přímo na stránce zvolit, kdo ho čte dál:

- Čtenáři z naší domény: každý, kdo se přihlásí z ověřené domény vaší organizace.
- Kdokoli s odkazem: veřejná stránka s odběrem Atom. Vyhledávače ji indexují, jen když to povolíte.

Čtenáři si na stránce changelogu můžou zapnout i e-mail "Co je nového" (od plánu Business): každé nové vydání jim přijde s obrázky v jejich jazyce.

Jak stránku používají čtenáři: Co je nového. Návody stejně aktuální udržíte podle stránky Publikovat živou dokumentaci.

Publikovat živou dokumentaci a hlídat zastaralé části

Návody ve vašem repozitáři můžou místo vložených screenshotů ukazovat obrazovky z přijatých běhů, takže obrázky nikdy nezestárnou. Když se obrazovka po napsání textu změní, Kaloko řekne, kterou část zkontrolovat. Stránky, které právě čtete, vznikají přesně takhle: `kaloko.app/p/sinfin/kaloko/docs`.

Připravte složku

```
npx kaloko init --docs=full
```

Vytvoří `docs/` se čtyřmi složkami podle Diátaxis, v každé šablonu:

| Složka | Čtenář chce | |---|---| | `tutorials/` | naučit se to od začátku do konce | | `how-to/` | udělat jeden konkrétní úkol | | `reference/` | dohledat fakt | | `explanation/` | pochopit proč |

`index.md` je úvodní stránka. Front matter je nepovinný: `title` (jinak první nadpis) a `order` (pořadí v postranním menu). Stránky propojujte relativními odkazy na `.md`.

Napište stránku

Obrázky jsou odkazy na kroky, stejně jako v changelogu:

1. V horní liště zvolte ****Přihlásit****.

`![Přihlašovací stránka](kaloko:KALOKO-public/login)`

Překlad leží vedle originálu: `pay.cs.md` vedle `pay.md`, jazyky vyjmenuje konfigurace (`docs.languages: [en, cs]`, jazyk originálů první). Obrázky se řídí jazykem textu.

Když má každá položka seznamu jednu akci a obrázek hned pod sebou, dá se stránka projít i jako průvodce: čtenář zvolí Projít krok za krokem, vidí vždy jeden krok a může si odškrtnout Mám vyzkoušeno.

Publikujte

```
npx kaloko docs publish # dokumentace pro latest
```

```
npx kaloko docs publish --version 2.4.0 # dokumentace vydání (Business)
```

Čtenáři dostanou výběr verze, přepínač jazyka a obrazovky a hledání a každý obrázek otevře svůj krok na plachtě. `kaloko release` publikuje složku s dokumentací pro svou verzi sám, pokud nepřidáte `--no-docs`. Kdo dokumentaci čte, se řídí stejnými pravidly jako changelog: viz Publikovat changelog s obrázky.

Hlídejte zastaralé části

Při publikování si Kaloko zapamatuje obrazovku za každým odkazem. Když pozdější přijatý běh ukáže jinou obrazovku, uvidí lidé, kteří dokumentaci píšou, u části poznámku "Obrázovka se změnila ve verzi 2.4 — zkontrolujte text." (Business).

```
npx kaloko docs check # proti publikované dokumentaci; potřebuje token
```

```
npx kaloko docs check --offline # jen lokální soubory
```

Chyby: rozbité odkazy, kroky, které scénáře nemají, odkazy bez přijatého obrázku a zastaralé části. Varování: chybějící nebo neaktuální překlady a položky changelogu, které lidé uvidí, ale nemají obrázek. Při chybě příkaz skončí kódem 1; s `--strict` i při varování, což se hodí do CI.

U každé zastaralé části otevřete krok, porovnejte ho s textem a přepište, co už nesedí; poznámka zmizí při dalším `kaloko docs publish`. Pokud text pořád sedí, `kaloko docs publish --reviewed` poznámky smaže beze změny textu. Použijte ho, až když jste je přečetli.

Jak dokumentaci dostat ven jako soubory pro jiný web: Exportovat dokumentaci a changelog.

Co je nového

Každé vydání produktu má v novinkách záznam psaný pro lidi, kteří produkt používají, s obrazovkami, které změnilo. Obrázky jsou ze schválených běhů, takže ukazují to, co někdo schválil.

Kde novinky najdete

1. V hlavní nabídce otevřete Produkty a u produktu zvolte Novinky, nebo otevřete odkaz, který vám někdo poslal. Každý produkt má své novinky a nejnovější vydání je nahoře.
2. Verze v horní části vás přenesou na své vydání. Pod každou verzí je, co se v ní změnilo, rozdělené na Nové, Změny, Opravy a Odebráno.

Porovnejte předtím a potom

1. U změněné obrazovky leží obrázek předtím a potom přes sebe. Posuvníkem posunete předěl.
2. Vedle sebe dá oba obrázky vedle sebe (na telefonu pod sebe).
3. Jazyk a Obrazovka přepnou všechny obrázky do jiného jazyka nebo velikosti obrazovky, pokud je běh zachytil.
4. Otevřít flow na plachtě ukáže celé flow, pokud máte k běhu přístup.

Bud'te v obraze

- Posílat e-mail o novinkách: ke každému novému vydání vám přijde jeho záznam s obrázky ve vašem jazyce. Odkaz na konci každého e-mailu je zase vypne.
- Stáhnout nabídne novinky jako JSON, Markdown, HTML pro offline čtení nebo PDF, třeba pro poznámky k vydání nebo zprávu pro klienta.
- Veřejné novinky mají i odběr pro čtečky.

Natočit video průchodu

Screenshot ukáže, kde krok skončil. Video ukáže, jak se tam dostal: točící se kolečko, které se nezastavilo, menu, které se otevřelo dvakrát. Nahrávejte, když screenshot neukáže, co má kontrolující posoudit: animace, načítání, přetahování, krok, který jednou projde a jednou ne, nebo ukázkou pro klienta.

Nahrajte průchod

```
npx kaloko walk --record # každý krok od předchozí obrazovky po jeho snímek
npx kaloko walk --record flow # celý průchod pro každý jazyk, kapitola na krok
npx kaloko walk --record failed # jen kroky, kterým selže skript nebo kontroly
npx kaloko walk --record --record-quality low --steps checkout
```

Pokud chcete nahrávat vždy, nastavte to ve scénáři nebo v kaloko.config.yml:

```
capture:
record: { mode: failed, quality: low, max_seconds: 45 }
nodes:
- id: checkout
record: true # tento krok vždy; record: false nikdy
```

Kvalita je low, medium (výchozí) nebo high. Video jsou mnohem větší než screenshoty, takže nenahrávejte každý průchod.

Průběh má každý webový krok

Každý prošlý webový krok si nechá průběh: kliknutí a psaní, síťové požadavky se statusem a časem a zprávy z konzole. Je malý a zapnutý ve výchozím stavu; vypne ho --no-trace.

```
npx kaloko video list # co se nahrálo a zaznamenalo, i s velikostí
npx kaloko video trace --step checkout # průběh kroku jako časová osa v terminálu
```

Na plachtě najdete v detailu kroku vedle snímku přepínač Video a na záložce Technické průběh. Klikněte na řádek a video na to místo skočí. Přehrání běhu pustí video celého flow po kapitolách.

Udělejte z běhu jedno video

```
npx kaloko video export --format mp4 # nebo webm, gif
npx kaloko video export --locale cs --steps cart,payment --lang cs --out checkout.mp4
```

Export seřadí videa podle flow a před každý krok dá úvodní kartu s jeho názvem a verdiktem. Takový soubor pošlete klientovi nebo přiložíte k poznámkám k vydání.

Potřebné nástroje

Videa z webu a z Electronu potřebují jen Chrome. S nainstalovaným ffmpeg dostanete i MP4 pro Safari, videa aplikací (Android, iOS Simulator, macOS) a export celého průchodu. npx kaloko video tools vypíše, co na počítači je. Průchody ve WebKitu a Firefoxu se nenahrávají a průběh se u nich nezaznamenává.

Sdílení

Nahrávat můžete v každém plánu. kaloko share nahraje videa spolu s během v plánech Business a Enterprise; ve Free a Team zůstanou v místním náhledu a sdílí se screenshoty a průběhy. Sdílená videa se uchovávají 30 dní (Enterprise si dobu nastaví v Nastavení -> Úložiště). Video větší než 25 MB zůstane jen u vás: snižte kvalitu nebo max_seconds.

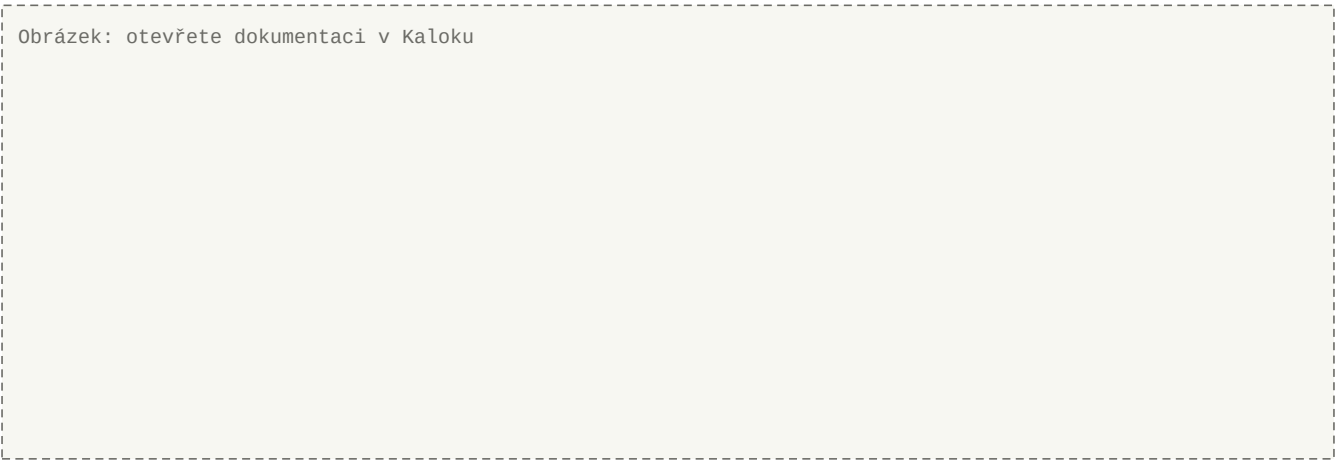
Projít běh

Běh je jeden průchod agenta flow: každá obrazovka, na kterou se dostal, ve všech jazycích a velikostech obrazovky, se vším, co zkontroloval. Projdete ho v prohlížeči, nic se neinstaluje.

Otevřete běh

1. Otevřete odkaz z pull requestu, z issue nebo z e-mailu. Běh se otevře na plachtě: všechny kroky flow jako mapa se šipkami, které ukazují cestu mezi nimi.

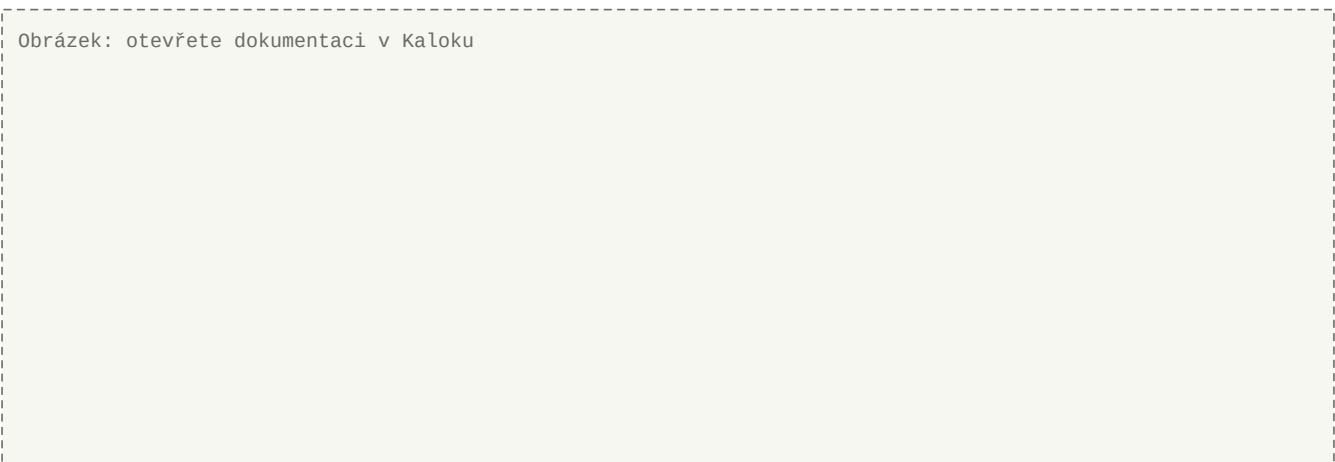
Obrázek: otevřete dokumentaci v Kaloku



Mapa běhu · cs · desktop

2. Pokud vám víc vyhovuje seznam, stiskněte I. Stejný běh se ukáže jako seznam kroků s výsledky a dá se v něm hledat i filtrovat podle stavu.

Obrázek: otevřete dokumentaci v Kaloku

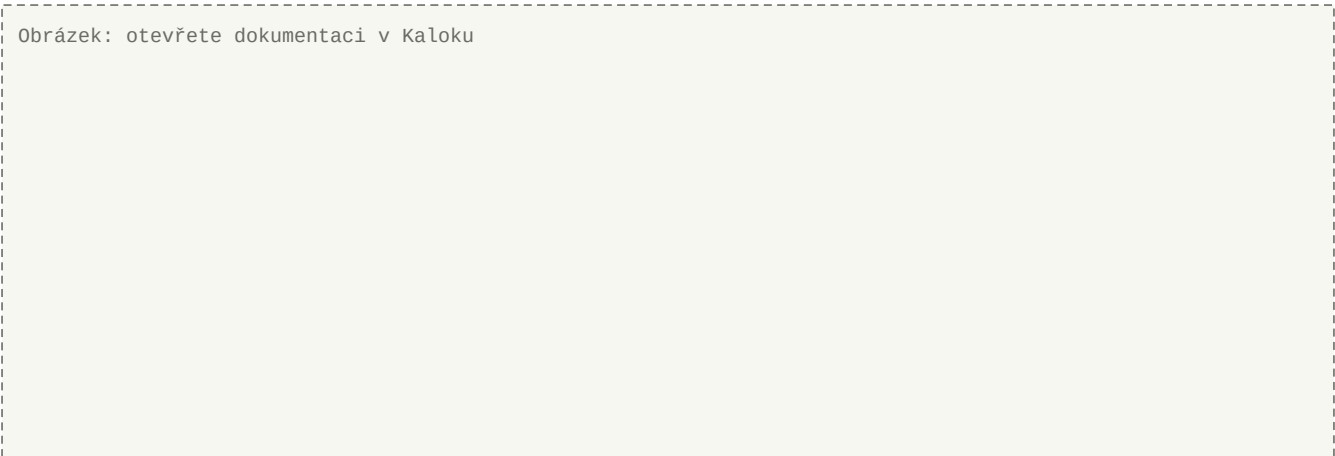


Běh jako seznam · cs · desktop

Podívejte se na krok

1. Klikněte na krok. V detailu uvidíte snímek obrazovky, kritéria s tím, co se zjistilo, a komentáře, které k němu lidé napsali. Šipkami <- a -> přejdete na předchozí a další krok.

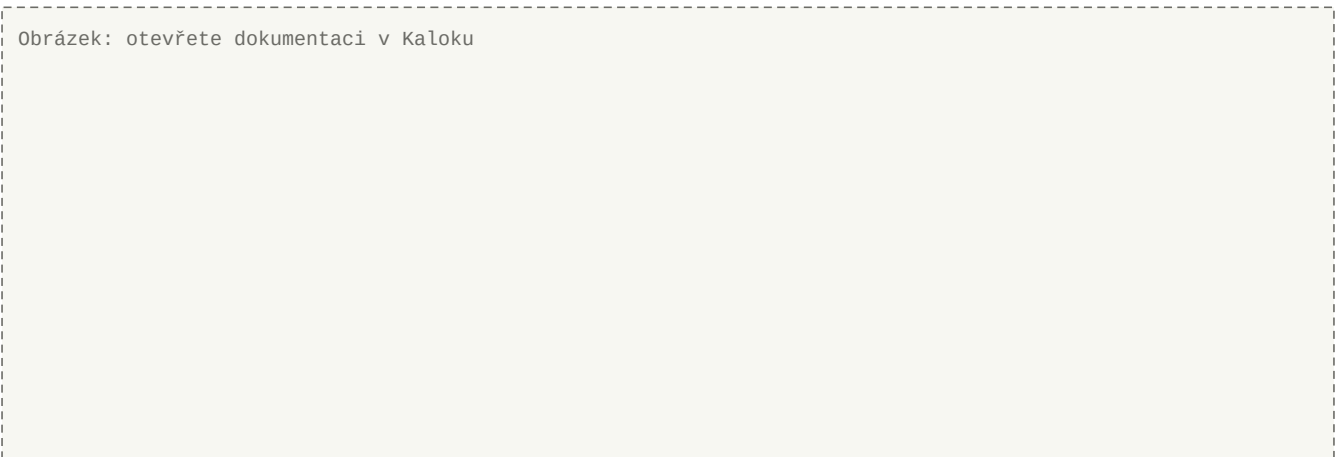
Obrázek: otevřete dokumentaci v Kaloku



Krok s komentářem · cs · desktop

2. Něco nesedí? Napište ke kroku komentář. Všechny otevřené komentáře běhu najdete v panelu komentářů, takže se mezi kroky nic neztratí.

Obrázek: otevřete dokumentaci v Kaloku

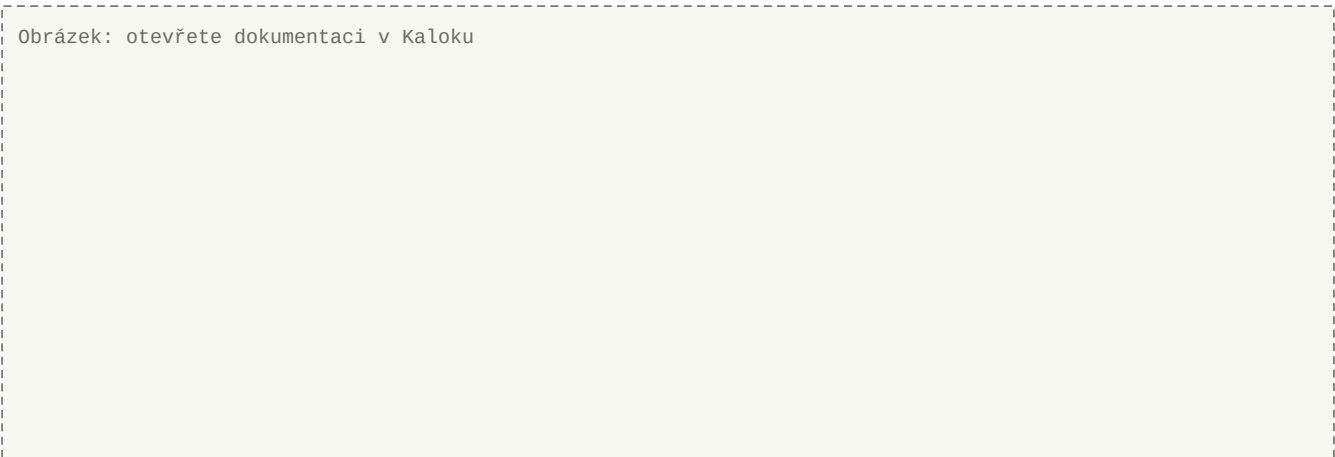


Panel komentářů · cs · desktop

Projděte ho jako uživatel

Klávesou R spustíte průchod během: jedna obrazovka po druhé, v pořadí, v jakém je uvidí člověk. Tak nejrychleji ukážete flow někomu, kdo plachtu ještě neviděl.

Obrázek: otevřete dokumentaci v Kaloku



Průchod během · cs · desktop

Až budete vědět, co si myslíte, schvalte nebo vraťte běh.

Kontrolovat jen to, co se změnilo

Režim kontroly ukazuje jeden krok po druhém a začíná těmi, které chtějí člověka: nesplněné a vrácené kroky, kritéria, kterými si AI nebyla jistá, kroky změněné od posledního přijatého běhu a kroky s otevřenými komentáři. Kroky, které se nezměnily, si ponechají schválení, které jim někdo dal dřív.

Otevřete režim kontroly

1. Otevřete odkaz na kontrolu z pull requestu nebo z e?mailu (končí #review), nebo otevřete běh a stiskněte Q.

Obrázek: otevřete dokumentaci v Kaloku

Režim kontroly · cs · desktop

2. Krok, o kterém rozhodujete, je vedle přijatého běhu, takže vidíte, co se pohnulo.

Rozhodujte z klávesnice

| Klávesa | Co udělá | |---|---| | J / K | další a předchozí krok | | A | schválí krok | | R | vrátí krok s poznámkou | | C | přidá komentář | | G | krok ve všech jazycích a velikostech obrazovky |

Když jsou rozhodnuté všechny kroky, celý běh přijměte, nebo vraťte.

Podívejte se zblízka

Ve velkém náhledu porovnáte oba běhy vedle sebe, jako rozdíl, posuvníkem (táhnete dělicí čáru přes oba screenshoty) nebo prolnutím (tento běh průsvitně přes ten druhý). Když chcete ukázat na problém, nakreslete při psaní komentáře do screenshotu rámeček, šipku nebo čáru, nebo vložte obrázek.

Převzatá schválení

U převzatého schválení je vidět, kdo ho dal a ve kterém běhu. Když si myslíte, že se má krok projít znovu, schválení zrušte a krok bude znovu čekat. Převzaté schválení za vás běh nikdy nepřijme.

Role a API tokeny

Každý člověk v organizaci má roli a každý token jedná za člověka, nikdy s většími právy, než ta role dovolí. Agenti a CI používají tokeny; místo v plánu nezabírají.

Role

| Role | Smí | Místo | |---|---|---| | Viewer | číst běhy a návrhy, komentovat, schvalovat a vracet | zdarma | | Designer | navíc pracovat s koncepty, verzemi, větvemi a sadami tokenů v Kaloko Design | jedno místo | | Vývojář / tester | navíc nahrávat a spravovat běhy a své tokeny | jedno místo | | Admin | navíc spravovat členy, domény, všechny tokeny a nastavení organizace | jedno místo |

Projekt může mít vlastního admina projektu, který do projektu zve lidi, mění jim role a projekt může omezit nebo otevřít. Projekty zakládají vývojáři / testéři a admini; kdo projekt založí, stane se jeho adminem. Roli člena změníte v Nastavení -> Lidé.

Vytvořte token v aplikaci

1. Otevřete Nastavení -> API tokeny.

Obrázek: otevřete dokumentaci v Kaloku

API tokeny v Nastavení · cs · desktop

2. Pojmenujte token podle toho, kde bude žít, třeba "CLI na notebooku" nebo "GitHub Actions".

3. Zaškrtněte, co smí. Číst může vždycky.

- komentáře a schvalování: komentáře a verdikty vaším jménem
- design: kaloko sync designových konceptů
- nahrávání běhů: kaloko share z CLI nebo z CI
- vše, co dovolí moje role

4. Vyberte, jak dlouho platí: 30 dní, 90 dní nebo rok.

5. Volitelně ho omezte na vybrané projekty; pak vidí jen ty a nové nezaloží.

6. Token jednou zkopírujte a uložte do .env projektu jako KALOKO_TOKEN.

Seznam ukazuje u každého tokenu vlastníka, roli, prefix, poslední použití a platnost. Zneplatnit token ukončí okamžitě.

Přihlaste se z terminálu

Na vlastním počítači to jde rychleji:

```
npx kaloko login
```

Otevře se prohlížeč, kde kód schválíte, vyberete organizaci a to, co token smí. Token platí nejvýš 30 dní a uloží

se do `~/config/kaloko/.env`. S `--print` ho jen vypíše, s `--no-open` ukáže odkaz bez otevření prohlížeče.

Tokeny pro CI

Vytvořte token s rozsahem nahrávání běhů, nejlépe omezený na projekt, a uložte ho v CI jako secret `KALOKO_TOKEN`. `npx kaloko ci github` a `ci gitlab` napíšou workflow, které ho čte; víc v referenci CLI.

Pravidla, která nastaví admin

V Nastavení -> Zabezpečení může admin povolit tokeny se všemi právy jen adminům, vypnout tokeny omezené na projekty a rozhodnout, jestli si tokeny smějí vytvářet hosté. Token hosta vidí jen jeho projekty. V plánu Enterprise drží tokeny pro automatizaci, která nepatří žádnému člověku, servisní účty.

Spouštět Kaloko v CI a komentovat pull requesty

V CI spouští Kaloko stejné příkazy jako na vašem počítači. Obvyklé nastavení projde na preview nasazení (Vercel, Netlify, Cloudflare Pages nebo jakákoli review app) to, co pull request změnil, a odkaz na plachtu napíše do komentáře v pull requestu.

1. Napište workflow

```
npx kaloko ci github # zapíše .github/workflows/kaloko.yml
npx kaloko ci gitlab # zapíše kaloko.gitlab-ci.yml; vložte ho do .gitlab-ci.yml přes include
npx kaloko ci github --print # jen ukáže, nic nezapíše
```

--scenario a.yml,b.yml ho omezí na vybrané scénáře. Příkaz zároveň přidá do kaloko.config.yml prostředí preview s base_url: \${KALOKO_PREVIEW_URL}. Scénář, který vyjmenovává environments:, musí obsahovat i preview, jinak ho kaloko start odmítne; příkaz vám řekne, kterých scénářů se to týká.

2. Přidejte token

V Nastavení -> API tokeny vytvořte token s rolí tester a přidejte ho do tajných proměnných CI v repozitáři jako KALOKO_TOKEN. Na GitLabu přidejte ještě KALOKO_GITLAB_TOKEN: project access token s rozsahem api, aby Kaloko mohlo napsat poznámku do merge requestu. Workflow commitněte.

Co workflow dělá

Po každém úspěšném preview nasazení pull requestu:

1. Nainstaluje projekt a prohlížeč (npx kaloko browser install --with-deps), a když o ně scénář žádá, i WebKit a Firefox.
2. npx kaloko ci preview-url --wait 600 --github-env najde adresu preview, pull request a jeho základní větev a předá je dál jako KALOKO_PREVIEW_URL, KALOKO_PR a KALOKO_BASE.
3. U každého scénáře kaloko affected --check zjistí, jestli změna zasahuje některý jeho krok. Pokud ne, scénář přeskočí, jinak spustí:

```
npx kaloko start --scenario qa/flows/checkout.yml --env preview
npx kaloko walk --ephemeral --affected
npx kaloko share --pr
```

share --pr přidá do pull requestu jeden komentář za každý scénář a při dalších pushích ho aktualizuje. Soubory se posílají podle obsahu, takže nezměněné screenshoty se znovu nenahrávají.

Dobré vědět

- walk --affected potřebuje historii gitu: na GitHubu checkout s fetch-depth: 0, na GitLabu GIT_DEPTH: "0".
- --trigger pull_request spouští workflow při každém pushi místo události nasazení a na preview počká.
- kaloko ci preview-url vypíše jen adresu, takže funguje v jakémkoli CI: export KALOKO_PREVIEW_URL="\$(npx kaloko ci preview-url)".
- Preview chráněné Vercel deployment protection potřebuje tajnou proměnnou VERCEL_AUTOMATION_BYPASS_SECRET a blok access.headers prostředí preview, který je ve vygenerované konfiguraci zakomentovaný.
- V pull requestu se komentář neobjeví? Obvykle chybělo preview nasazení nebo token. Log jobu řekne, co z toho.

Pokud vaše vydání publikuje release-please, changesets nebo semantic-release, přidá npx kaloko ci github --release druhé workflow, které každé vydání zapíše do Kaloka. Všechny volby najdete v referenci CLI.

Projít aplikace pro Android, iOS a desktop

Nativní aplikace Kaloko prochází stejně jako webové stránky: z každého kroku je screenshot s UI stromem, zkontrovaná kritéria a místo na plachtě. Scénář říká, o jakou platformu jde, a konfigurace, kterou aplikaci otevřít.

1. Pojmenujte aplikaci pro každé prostředí

```
# kaloko.config.yml
environments:
  staging:
    app:
      android: { package: com.acme.shop.staging, apk: build/app-staging.apk, reset: true }
      ios: { bundle: com.acme.shop, app: build/Shop.app, device: iPhone 17 }
      desktop:
        app:
          electron: { main: dist/main.js }
# desktop: { name: Notes } # aplikace pro macOS podle okna
# windows: { name: contoso, path: dist/Contoso.exe }
```

2. Nastavte platformu ve scénáři

```
platform: android # android | ios | electron | desktop | windows
devices: [pixel-8-api-34, pixel-7-api-33]
color_schemes: [light, dark]
nodes:
  - id: product
kind: screen
path: shop://product/42 # deep link
packs: [app]
```

Na kroky se aplikace dostane přes deep link nebo skriptem kroku (`app.tap(...)`, `app.type(...)`). V každém jazyce se aplikace spustí znovu a v daném jazyce, `reset: true` nejdřív smaže její data.

Co která platforma umí

| Platforma | Co ji ovládá | Poznámka | |---|---|---| | Android | adb: emulátor, telefon přes USB nebo přes Wi-Fi | neběžící emulátor se spustí; tmavý režim a omezený pohyb se nastaví pro každý snímek | | iOS Simulator | xcrun simctl | ťuknutí a UI strom potřebují idb; bez něj Kaloko přečte text z obrazovky | | iPhone a iPad | devicectl a nástroj na screenshots | jen zachycení, text se čte z obrazovky | | Electron | Playwright | okno je webová stránka: platí všechny webové kontroly | | Aplikace pro macOS | okno aplikace a strom přístupnosti | jen zachycení; mezi obrazovkami přechází váš nástroj, AppleScript nebo člověk | | Aplikace pro Windows | UI Automation přes PowerShell | ťuknutí i psaní; procházejte ji ve Windows |

Pack app kontroluje popisky ovládacích prvků, velikost dotykových ploch, useknutý text a pády.

Obrazovky z jiných nástrojů

Screenshot může pořídít Maestro, Appium, XCUITest, Detox nebo člověk a Kaloko ho rozloží i s UI stromem:

```
npx kaloko capture --step checkout --image shot.png --tree hierarchy.xml --device "Pixel 8" --density 2.625
```

--tree přečte XML z uiautomatoru nebo Appia, idb ui describe-all --json a JSON hierarchie z Maestra. Bez stromu přečte --ocr text z obrázku (na macOS). Celou složku screenshotů nebo report testů nahrajete přes `npx kaloko import`.

Nejdřív zkontrolujte počítač

`npx kaloko doctor`

Vypíše, které nástroje pro aplikace na počítači chybějí (adb, simctl, idb, ffmpeg), a jak je nainstalovat. Na macOS si kaloko doctor --fix vyžádá oprávnění k nahrávání obrazovky a zpřístupnění, která desktopové snímky potřebují.

Napsat scénář

Scénář je flow, které Kaloko projde: soubor YAML s obrazovkami, cestou mezi nimi a tím, co má každá obrazovka splňovat. Scénáře k úkolům bývají u úkolu (docs/tasks/TASK-123/qa/scenario.yml), dlouhodobá flow, třeba pokladna, v qa/flows/.

Nechte ho napsat agenta

S nainstalovaným skillem nemusíte psát YAML, stačí zadání:

Přečti si úkol v docs/tasks/TASK-123, napiš k němu akceptační plán a scénář pro Kaloko a zvaliduj ho.

Agent napíše dva soubory: ACCEPTANCE.md pro lidi (cíl, kroky, kritéria AC1, AC2...) a scénář pro běh. Když se zadání změní, upraví nejdřív plán a potom scénář.

Z čeho se soubor skládá

```
schema: qawalk.scenario.v1
id: TASK-123-checkout
title: Pokladna
kind: task # task | flow
product: shop # volitelné, s module:
locales: [cs, en]
viewports:
desktop: [1280, 800]
mobile: [390, 844]
environments: [local, staging]
nodes:
- id: cart
kind: screen
title: Košík
path: /cart
purpose: Návštěvník vidí, co kupuje, a může pokračovat k platbě.
execution:
strategy: playwright # playwright | agent | manual
script: steps/cart.mjs
criteria:
- id: AC1
text: Stránka má právě jeden hlavní nadpis.
check: { type: deterministic, assert: count, selector: h1, min: 1, max: 1 }
edges:
- { from: cart, to: payment, label: pokračovat }
```

- Id kroků (cart) neměňte: odkazují na ně odkazy, komentáře, schválení i obrázky v dokumentaci.
- purpose je jedna věta o tom, k čemu obrazovka slouží. Čte ji evaluátor.
- execution říká, kdo se na obrazovku dostane: skript v Playwrightu, agent, který ji zachytí přes kaloko capture, nebo člověk.
- Další druhy uzlů: decision pro větvení, email pro zprávu ze schránky, stack pro mnoho stránek jedné šablony.

Všechny klíče najdete v referenci souboru scénáře. Jak přidat kontroly, popisuje návod Přidat kritéria a packy kontrol.

Zkontrolujte ho

```
npx kaloko validate qa/flows/checkout.yml
npx kaloko validate qa/flows/checkout.yml --coverage
```

validate ověří schéma a odkazy mezi kroky. --coverage porovná kritéria s akceptačním plánem a vypíše, co plán

chce a scénář nekontroluje.

Začněte od veřejných stránek

U marketingového webu nebo dokumentace začněte konceptem:

```
npx kaloko draft https://example.com/ https://example.com/pricing
```

Kaloko stránky stáhne a napíše k nim produkční scénář jen pro čtení i s plánem. Jako účel použije titulek, H1 a popis stránky a přidá packy seo, a11y, perf a console. Přidá i krok s adresou, která neexistuje, takže se zkontroluje i stránka 404. Pak účel každé stránky upřesněte a doplňte, co musí opravdu splňovat.

Kritéria, kontroly a packy

Kritérium je jedna věc, kterou má krok splnit. Má id (AC1), text pro lidi a právě jednu kontrolu check. Jak je do scénáře přidat: Přidat kritéria a packy. Jak se z výsledků stane verdikt: Jak vznikají verdikty.

Druhy kontrol

| type | Kdy ho použít | Volby | |---|---|---| | deterministic | cokoli spočitatelného nebo přesného: přítomnost, počty, text, URL, styly, hodnoty ve stránce | assert a jeho volby (níže) | | semantic | úsudek: je text srozumitelný, je na stránce jedna hlavní akce | question (anglicky, ano = splněno), true a false (co se počítá za ano a za ne), scope (CSS selektor, který se pošle evaluátorovi), about: console (čte chyby v konzoli místo stránky), vision: true (ukáže i screenshot) | | manual | nehodí se ani jedno; rozhodne kontrolující na plachtě | note |

Čísla, data a počty patří do deterministických kontrol, nikdy do sémantické otázky. V jedné otázce se ptejte na jednu věc.

Deterministické kontroly (assert)

Společné volby: selector (CSS), pattern (regulární výraz) s flags (i, m, s), value, min, max, all (vyhovět musí každá shoda, ne jen jedna).

Obsah stránky

| assert | Splněno, když | |---|---| | exists | selector najde aspoň jeden prvek | | absent | selector nenajde nic | | count | počet shod je mezi min a max | | text_matches | viditelný text prvku odpovídá pattern | | text_absent | žádný viditelný text neodpovídá pattern | | text_equals | celý text se rovná value (mezery sloučené) | | attr_equals | atribut attr se rovná value | | attr_matches | atribut attr odpovídá pattern | | no_overflow | stránka se neposouvá do strany | | no_console_errors | v konzoli nejsou chyby |

Adresa, metadata a odkazy

| assert | Splněno, když | |---|---| | title_matches | titulek stránky odpovídá pattern | | meta_matches | content meta tagu ze selector odpovídá pattern | | link_matches | href odkazu ze selector odpovídá pattern | | url_matches | URL po kroku odpovídá pattern; part vybere href, path, query, hash, host nebo origin | | url_equals | URL se rovná value (cesta nebo celá URL); ignore_query vynechá ?... a #... | | http_status | stav dokumentu je mezi min a max (výchozí 200–299), nebo se rovná value | | link_home | stránka odkazuje zpět na úvodní stránku | | links_ok | každý odkaz na stejný původ odpoví pod 400 (max odkazů na stránku, pattern přeskočí cesty, follow mimo prostředí jen pro čtení) | | hreflang_pairs | aspoň min jazyků, každá verze odpoví 200 a odkazuje zpět |

Styly a hodnoty ve stránce

| assert | Splněno, když | |---|---| | style_equals | spočítaná CSS vlastnost property se rovná value nebo design tokenu token; op porovnává čísla ($\geq$), tolerance dovolí zaokrouhlení v px | | style_matches | spočítaná property odpovídá pattern | | js_equals | jeden JavaScriptový výraz expression, který jen čte, vrátí ve stránce value (nebo odpovídá pattern, nebo se porovná přes op) |

Přístupnost (používá je pack a11y)

| assert | Splněno, když | |---|---| | a11y_images_alt | obrázky mají alternativní text | | a11y_names | odkazy a tlačítka mají přístupný název | | a11y_labels | formulářová pole mají popisky | | a11y_headings | úrovně nadpisů se nepřeskakují | | a11y_contrast | text splňuje kontrast WCAG AA | | a11y_main | stránka má landmark main | | a11y_unique_ids | id prvků jsou jedinečné | | a11y_focus_order | pořadí tabulátoru odpovídá stránce a dojde ke každému ovládacímu prvku | | a11y_focus_visible | fokus je vidět s kontrastem aspoň 3 : 1 | | a11y_keyboard_trap | tabulátor nikde neuvízne | | a11y_skip_link | odkaz pro přeskočení nebo hlavní obsah do tří zastavení tabulátoru | | a11y_ax_names | ovládací prvky mají název ve stromu přístupnosti prohlížeče | | a11y_landmarks | landmarky jsou úplné a opakované mají popisek | | a11y_aria | role a atributy ARIA jsou platné |

Výkon, aplikace a design

| assert | Splněno, když | |---|---| | perf_lcp, perf_cls, perf_ttfb, perf_weight | vykreslení největšího obsahu, posun

rozvržení, čas do prvního bajtu nebo váha stránky nepřekročí max | | app_labels | každý ovládací prvek aplikace má přístupný popis | | app_touch_targets | dotykové plochy mají aspoň 48 dp (Android) nebo 44 pt (iOS) | | app_no_truncation | žádný text není useknutý trojtečkou | | app_no_crash | aplikace nespadá ani nepřestala odpovídat | | design_pixels | obrazovka se od reference designu liší nejvýš v podílu ratio pixelů (výchozí 0,03) | | design_structure | nadpisy, landmarky, akce a pole odpovídají referenci designu | | uses_tokens | barvy, písma, velikosti, mezery a zaoblení pocházejí ze sady tokenů |

Packy

packs: [seo, a11y] u kroku přidá pevnou sadu kritérií. Vlastní kritérium se stejným id nahradí kritérium z packu; tak se mění limit (AC401 s max: 4000).

| Pack | Id | Co přidá | |---|---|---| | seo | AC101–AC111 | jeden H1, titulek 10–70 a popis 50–160 znaků, canonical, hreflang pro každý jazyk, Open Graph, žádný noindex, meta viewport, html lang, žádné přetečení, žádné chyby v konzoli | | usability | AC201–AC204 | účel je jasný z první obrazovky, jedna hlavní akce, pole s popisky, texty odkazů, které říkají, co udělají | | a11y | AC301–AC314 | všechny kontroly přístupnosti výše; průchod navíc projde stránku tabulátorem, nic nekliká ani nepíše | | perf | AC401–AC404 | LCP do 2,5 s, CLS pod 0,1, TTFB do 800 ms, váha stránky do 3 MB (měřeno v prohlížeči průchodu) | | console | AC501 | evaluátor si přečte chyby v konzoli a řekne, jestli se týkají toho, co krok přijímá | | errors | AC601–AC606 | skutečný stav 4xx/5xx, odkaz na úvod, navigace nebo hledání, žádný výpis chyby ze serveru, srozumitelná zpráva s cestou dál, žádné chyby v konzoli | | app | AC701–AC705 | popisky, dotykové plochy, žádný useknutý text, žádný pád, jasný účel a další krok | | design | AC801–AC803 | implementace proti referenci designu: pixely, struktura, tokeny |

Pack a11y pokrývá část WCAG 2.2 AA. Audit se čtečkou obrazovky nenahradí.

Příkazy CLI

Všechny příkazy CLI kaloko a co dělají. Spouštějí se jako `npx kaloko <příkaz>` v projektu, který má balíček nainstalovaný. `npx kaloko help` vypíše stejný seznam se všemi volbami.

Volby, které fungují všude: `--run <id|složka>` (nebo `KALOKO_RUN`) vybere běh, `--config <soubor>` konfiguraci a `KALOKO_AGENT=<jméno>` označí, co zapsal agent.

Nastavení projektu

| Příkaz | Co dělá | |---|---| | `kaloko init` | Vytvoří `kaloko.config.yml`, nainstaluje skill pro Claude Code nebo Codex (`--agent`), pozná framework, dev server, jazyky a hlavní cesty a připraví první scénář; `--docs=off` | `changelog` | `full` založí changelog a složky dokumentace | | `kaloko login` | Přihlásí tento počítač přes prohlížeč a uloží token (nejvýš na 30 dní) do `~/.config/kaloko/.env`; s `--print` ho jen vypíše | | `kaloko doctor` | Zkontroluje Node, Chrome, konfiguraci, prostředí, porty prohlížeče, token, klíč evaluátoru a nástroje pro aplikace; `--fix` si na macOS řekne o oprávnění pro snímání | | `kaloko draft <url...>` | Připraví první plán a scénář jen pro čtení pro veřejné stránky, s packy seo, a11y, perf a console | | `kaloko validate <soubor...>` | Zkontroluje scénáře proti schématu a jejich odkazy; `--coverage` porovná kritéria s akceptačním plánem | | `kaloko scenarios` | Vypíše scénáře, které konfigurace najde | | `kaloko ci github` | `gitlab` | Napíše workflow, které u každého pull requestu projde dotčené kroky na preview nasazení a napíše k němu komentář | | `kaloko ci github --release` | Napíše workflow, které zapíše každý release vydaný vaším nástrojem na releasy | | `kaloko ci preview-url` | V CI najde adresu preview, pull request a jeho základ |

Průchod a snímání

| Příkaz | Co dělá | |---|---| | `kaloko start` | Založí běh scénáře `--scenario` v prostředí `--env` a nastaví ho jako aktuální; `--locales`, `--viewports`, `--browsers`, `--color-schemes`, `--devices` a další určí, co se nasnímá | | `kaloko walk` | Spustí kroky v Playwrightu, nasnímá je a vyhodnotí; `--affected`, `--record`, `--heal`, `--stability N`, `--concurrency N`, `--ephemeral` | | `kaloko capture --step <id>` | Nasnímá aktuální stránku pro krok; `--crop` přiloží prvek, `--image` převezme obrazovku z jiného nástroje | | `kaloko open --step <id>` | Otevře adresu kroku ve sdíleném prohlížeči | | `kaloko mail --step <id>` | Zachytí e-mail ze schránky prostředí, nebo uloženou zprávu přes `--file` | | `kaloko browser start` | `stop` | `status` | Jeden sdílený prohlížeč na prostředí, který ovládají agenti, skripty i lidé | | `kaloko browser install` | Stáhne prohlížečová jádra (`--browsers chromium,webkit,firefox`) | | `kaloko devices` | Vypíše předvolby zařízení, které lze použít jako viewporty | | `kaloko session reset` | Začne jako nový návštěvník: smaže cookies a úložiště ve sdíleném prohlížeči | | `kaloko auth save` | Člověk se jednou přihlásí (SSO, MFA) a průchod relaci použije znovu; `kaloko auth list` ukáže uložené relace | | `kaloko affected` | Řekne, kterých kroků se změněné soubory týkají a proč | | `kaloko import <cesta>` | Rozloží screenshots, reporty a trace z Playwrightu nebo průchod jiného agenta jako běh | | `kaloko status` | Ukáže, co aktuální běh nasnímá |

Výsledky a kontrola

| Příkaz | Co dělá | |---|---| | `kaloko evaluate` | Spustí deterministické kontroly a sémantický evaluátor pro všechny snímky; `--recheck` po změně scénáře | | `kaloko build` | Zapíše `run.json` a místní náhled | | `kaloko preview` | Sestaví náhled a vypíše cestu k němu | | `kaloko share` | Nahraje běh do `kaloko.app`; `--pr` napíše komentář do pull requestu nebo merge requestu | | `kaloko review` | Co na sdíleném běhu ještě musí rozhodnout člověk, a odkaz do režimu kontroly | | `kaloko feedback` | Otevřené připomínky recenzentů, vrácené kroky jako první | | `kaloko comment` | Komentuje, odpovídá, uzavírá a znovu otevírá připomínky z terminálu | | `kaloko compare` | Textový rozdíl dvou běhů: stavy, verdikty, skóre a důkazy | | `kaloko stability` | Najde oblasti, které se mění při každém běhu; `--apply` je přidá do scénáře, `--pull` převezme oblasti nakreslené na `kaloko.app` | | `kaloko calibrate` | Porovná sémantický evaluátor s hodnocením lidí a navrhne prahy | | `kaloko signoff <run-id>` | Stáhne podepsaný záznam o přijetí přijatého běhu; `--verify <soubor>` ověří záznam i bez tokenu | | `kaloko video export` | Udělá z videí běhu jedno video; `kaloko video list`, trace a tools ukážou videa, průběh kroků a nástroje | | `kaloko screens` | Hledá v nejnovějších screenshotech všech kroků napříč vašimi flow | | `kaloko inbox` | Vaše upozornění na `kaloko.app` |

Sdílené běhy

| Příkaz | Co dělá | |---|---| | kaloko projects | Projekty organizace | | kaloko runs | Sdílené běhy, u každého scénáře ten nejnovější; --all vypíše všechny | | kaloko trash <run-id...> | Přesune běhy do koše; kaloko restore <run-id> jeden vrátí | | kaloko prune | Ponechá nejnovější běhy každého scénáře a prostředí a ostatní vypíše ke smazání; --yes je přesune do koše | | kaloko keep <run-id> | Ponechaný běh nikdy nevyprší ani se neuklidí; --off ho uvolní | | kaloko archive <scenario-id> | Skryje vyřazený scénář ze seznamu; jeho běhy zůstanou | | kaloko public | Zpřístupní sdílený běh bez přihlášení | | kaloko export | Stáhne sdílený běh jako zip; --scenario nejnovější snímek každého kroku; --format dokumentaci a changelog |

Jira, Linear a hostované běhy

| Příkaz | Co dělá | |---|---| | kaloko issue create | Úkol v Jiře nebo Linearu z vráceného kroku nebo nesplněného kritéria; kaloko issue list je vypíše | | kaloko schedule add | Kaloko každý den nebo týden otevře stránky scénáře jen pro čtení a běh nasdílí | | kaloko schedule list\run\pause\resume\remove | Správa naplánovaných běhů | | kaloko schedule export github | Workflow pro GitHub Actions, které scénář pravidelně projde celým CLI |

Produkty, changelog a dokumentace

| Příkaz | Co dělá | |---|---| | kaloko products sync | Pošle produkty z konfigurace do Kaloka, aniž by zapsal release | | kaloko products list | Produkty, které v Kaloku čtete, a produkty z konfigurace, které tam ještě nejsou | | kaloko changelog draft | Napíše sekci Unreleased z pull requestů a změněných obrazovek; --apply ji zapíše do CHANGELOG.md | | kaloko release [<verze>] | Zapíše release se záznamem z changelogu, obrázky a dokumentací | | kaloko docs publish | Pošle složku s dokumentací do Kaloka jako dokumentaci jedné verze | | kaloko docs check | Rozbité odkazy, zastaralé sekce, chybějící překlady a položky changelogu bez obrázku; při chybách skončí kódem 1 | | kaloko export --format <f> | Dokumentace a changelog jako Markdown, Docusaurus, VitePress, MkDocs, HTML, PDF nebo JSON |

Design

| Příkaz | Co dělá | |---|---| | kaloko sync | Vymění revize návrhu se společným konceptem (--pull, --push) | | kaloko design lint | Zkontroluje složku s návrhem podle pravidel přenositelného HTML | | kaloko design import figma | Udělá z framů ve Figmě designové kroky a z proměnných tokeny | | kaloko history | Verze flow a revize kroku | | kaloko spec [krok] | Co postavit podle schváleného designového kroku: prvky, styly napojené na tokeny, stavy | | kaloko drift [krok] | Čím se implementace liší od schváleného návrhu; dokud se liší, končí kódem 1 | | kaloko reference | Ukáže nebo nastaví referenci designu a srovnávací běh | | kaloko tokens pull\push\check\validate | Design tokeny projektu (DTCG) | | kaloko branch list\create\merge | Větve konceptu návrhu |

kaloko help vypíše všechno výše se všemi volbami.

Konfigurační soubor

kaloko.config.yml leží v kořeni repozitáře a říká CLI, kde vaše flow běží. kaloko init ho vytvoří i s komentáři, kaloko doctor ho zkontroluje. Jeho formát je qawalk.config.v1 a povinné jsou schema, org a environments.

```
schema: qawalk.config.v1
org: acme
project: shop
scenarios: [qa/flows/*.yml]
environments:
  local:
    base_url: http://localhost:3000
    staging:
      base_url: https://staging.acme.test
    access:
      basic: { user_env: STAGING_USER, password_env: STAGING_PASSWORD }
      accounts:
        member: { user: qa@acme.test, password_env: MEMBER_PASSWORD }
      production:
        base_url: https://acme.com
        readonly: true
        blocked_paths: ["/account/**"]
    walk:
      browsers: [chromium]
    share:
      expires_days: 30
```

Klíče nejvyšší úrovně

| Klíč | Co obsahuje | |---|---| | schema | vždy qawalk.config.v1 | | org | zkratka (slug) organizace na kaloko.app | | project | projekt v Kaloku pro běhy; výchozí je název repozitáře | | scenarios | masky souborů se scénáři | | output | složka běhů (výchozí tmp/kaloko) | | environments | kde flow běží (níže) | | browser | channel (kanál Playwrightu, třeba chrome; prázdný = přibalené Chromium) a port sdíleného prohlížeče (výchozí 9333) | | walk | výchozí hodnoty pro každý běh: concurrency (jazyky najednou, 1–16), browsers, color_schemes, reduced_motion, forced_colors, display_modes | | capture | výchozí hodnoty pro každý scénář: record, trace | | evaluators | sémantický evaluátor jev a evaluátor obrázků vision (níže) | | share | url (výchozí https://kaloko.app), token_env (výchozí KALOKO_TOKEN), expires_days (výchozí 30) | | docs | nastavení changelogu a dokumentace (níže) | | products | co vydáváte a verzujete jako celek (níže) |

Prostředí

Každý klíč pod environments je název, který použijete v kaloko start --env.

| Klíč | Co obsahuje | |---|---| | kind | app (výchozí), nebo design (složka živých HTML kroků) | | base_url | adresa, nebo jedna pro každý jazyk se záložní default; \${VAR} se doplní z proměnných prostředí (review aplikace) | | readonly | true pro produkci: CLI jen čte, nikdy se nepřihlašuje do administrací a nevytváří data | | blocked_paths | masky cest, které běh jen pro čtení nikdy neotevře | | allowed_post_paths | masky cest, kam běh jen pro čtení smí poslat POST (košík) | | may_create_data | false zakáže vytváření dat (výchozí true) | | access | ochrana před celým prostředím: basic (user_env, password_env), headers (servisní tokeny), client_certificate a origins, které jsou dostanou | | accounts | lidé, kteří se přihlašují, pod stejnými klíči jako account ve scénáři: user nebo user_env, password_env, totp_env, storage_state (přihlášení uložené přes kaloko auth save) | | mail | schránka: provider (mailpit, script, none), url, user_env, password_env, address, script, options, auth | | app | testovaná aplikace pro každou platformu: android, ios, electron, desktop, windows | | serve, fixtures, tokens_mode | designová prostředí: složka kroků, JSON fixtures, režim tokenů | | html_view | experimentální: ke každému snímku statický HTML pohled |

Evaluátory

| Klíč | Co obsahuje | |---|---| | evaluators.jev.api_key_env | proměnná s vaším vlastním klíčem k evaluátoru (výchozí TYPESAFE_API_KEY) | | evaluators.jev.hosted | bez místního klíče vyhodnocuje z měsíčního limitu vašeho plánu (výchozí true); false použije jen váš klíč | | evaluators.jev.thresholds | skóre pro pass a fail | | evaluators.jev.model, base_url, max_state_chars, signals, concurrency | jemnější nastavení, potřeba jen výjimečně | | evaluators.vision | druhý evaluátor, který se dívá na screenshot (poskytovatel anthropic, klíč v ANTHROPIC_API_KEY); false ho vypne |

Dokumentace a produkty

| Klíč | Co obsahuje | |---|---| | docs.dir | zdroje dokumentace (výchozí docs) | | docs.changelog | soubor changelogu (výchozí CHANGELOG.md) | | docs.style | keep-a-changelog nebo release-notes | | docs.version_source | auto, git-tag, package.json, release-please, changesets, semantic-release, pyproject, cargo, gemspec, manual | | docs.languages | jazyky dokumentace; první je jazyk originálů | | docs.framework | auto, docusaurus, vitepress, mkdocs, next, plain | | docs.publish | kdo čte: members, domain, nebo public (poslední dvě potřebují admina) | | docs.images | viewport a locale (reader nebo jazyk) obrázků | | products.<slug> | name (text nebo pro každý jazyk), version (source, path, pattern), docs, project nebo projects, modules, reference_env, default_scenario, publish |

Bez products je produktem projekt.

Tajné údaje

Tajné údaje do konfigurace nikdy nepatří. Kde je nějaký potřeba, konfigurace jen pojmenuje proměnnou prostředí (password_env: STAGING_PASSWORD, { env: CF_ACCESS_CLIENT_SECRET }). Kaloko hodnotu hledá:

1. v prostředí procesu,
2. v .env vedle kaloko.config.yml,
3. v ~/.config/kaloko/.env.

Proměnná, která už je nastavená, má přednost před soubory. Hodnoty se odstraní ze zachyceného HTML, z textu stránky i z výstupu konzole. kaloko doctor řekne, které pojmenované proměnné chybí. Token pro sdílení je KALOKO_TOKEN, pokud share.token_env neříká jinak; viz Role a tokeny.

Nástroje MCP

MCP server Kaloka dovolí agentovi číst běhy, připomínky, návrhy, changelog i dokumentaci a odpovídat kontrolujícím. Je součástí plánu Team a vyšších. Zachytávání, vyhodnocení a nahrávání zůstávají v CLI.

Připojení

Server je na <https://kaloko.app/mcp> (MCP přes Streamable HTTP). Ověříte se API tokenem organizace v hlavičce Authorization; jak ho vytvořit: Role a tokeny.

```
claude mcp add --transport http kaloko https://kaloko.app/mcp --header "Authorization: Bearer $KALOKO_TOKEN"
```

Ostatní agenti (Codex, Cursor) dostanou stejnou adresu a hlavičku jako HTTP MCP server. Na čtení stačí token jen pro čtení; nástroje, které zapisují, potřebují token s příslušným rozsahem. Screenshoty a soubory přes MCP necestují: odpovědi nesou jejich adresy, které stáhnete stejným tokenem.

Běhy a kontrola

| Nástroj | Co dělá | |---|---| | get_started | kde organizace stojí, její ukázkový běh a další kroky | | add_sample_run | přidá ukázkový běh, nebo vrátí ten, který už existuje | | list_projects | projekty organizace s jejich zkratkami | | list_runs | sdílené běhy od nejnovějšího, s filtrem podle scénáře, projektu, verdiktu, autora, prostředí nebo větve | | get_run_summary | jeden běh: stav každého kroku a jazyka, kritéria, která chtějí pozornost, s důkazy a zdůvodněním, adresy screenshotů, odkaz na plachtu | | get_stack_items | stránky stack kroku s jejich stavem a nesplněnými kritérii | | get_step_assets | nejnovější snímek každého kroku scénáře na stálých adresách (Business) | | get_step_recording | video a záznam průběhu kroku | | get_criteria_verdicts | verdikty kontrolujících u ručních nebo přepsaných kritérií | | get_review_queue | o čem v běhu ještě musí rozhodnout člověk, a odkaz na kontrolu, který mu předáte | | decide_step | vlastní schválení nebo vrácení kroku agentem; označí se jako agentovo a nikdy se nepočítá jako lidské | | revoke_carried_approval | zruší schválení převzaté u kroku, na který se má někdo podívat znovu | | get_signoff_record | podepsaný záznam o přijetí běhu (Business) | | get_affected_steps | ke kterým krokům scénáře změněné soubory dosáhnou, a proč | | list_notifications | schránka majitele tokenu |

Připomínky a úkoly

| Nástroj | Co dělá | |---|---| | get_feedback | verdikt běhu, schválené kroky, otevřená vrácení a komentáře | | list_comments | otevřená vlákna v celé organizaci s odpověďmi, špendlíky a nakreslenými oblastmi | | add_comment | komentář ke kroku, běhu nebo celému flow, nebo odpověď | | resolve_comment | uzavře vlákno s poznámkou, co se změnilo | | reopen_comment, edit_comment, delete_comment | znovu otevře vlákno; upraví nebo smaže vlastní komentář | | list_ignore_regions | oblasti, které lidé nakreslili, aby se vynechaly z porovnání pixelů | | add_ignore_region | navrhne takovou oblast | | create_issue | úkol v Jiře nebo Linearu z kroku, se screenshotem, kritériem a poznámkami | | list_issues | úkoly vytvořené z kroků a jejich stav | | list_schedules, run_schedule | hostované běhy a jejich limit stránek; jeden spustí hned |

Design

| Nástroj | Co dělá | |---|---| | get_draft | kterou revizi ukazuje každý krok společného konceptu a kdo ji změnil | | get_step | předání kroku: živé soubory, tokeny, struktura stránky, otevřené komentáře | | get_step_spec | co se má u kroku postavit podle schváleného návrhu: prvky, styly převedené na tokeny, stavy, kontrast | | get_step_drift | čím se běh implementace liší od reference designu | | get_step_history | verze a revize kroku s autory | | list_step_changes | kroky, které se změnily mezi dvěma referencemi nebo od posledního přijatého běhu | | get_references, get_tokens | reference designu a baseline; sada tokenů projektu | | get_draft_feedback | komentáře ke konceptu | | get_version_notes, set_version_notes | přečte nebo přepíše, co se ve verzi změnilo | | publish_version, set_reference, set_step_status, add_draft_comment | publikuje koncept jako verzi, nastaví referenci, nastaví stav kroku, okomentuje koncept |

Obrazovky, changelog, dokumentace a produkty

| Nástroj | Co dělá | |---|---| | search_screens | knihovna obrazovek: nejnovější screenshot každého kroku, hledání podle slov, projektu, produktu, jazyka nebo stavu | | list_products | produkty, které čtete, s nejnovějším vydáním, dokumentací a čtenáři | | get_changelog | vizuální changelog produktu, volitelně i s obrázky | | draft_changelog_entry | co přijaté běhy od posledního vydání ukazují jinak, jako koncept záznamu s odkazy na kroky před a po | | resolve_step_image | jeden odkaz na krok (kaloko:<scénář>/<krok>) jako adresa obrázku | | get_docs | publikovaná dokumentace produktu: verze, jazyky, postranní menu, nebo jedna stránka | | docs_check | co v publikované dokumentaci chce pozornost: rozbité odkazy, zastaralé části, chybějící překlady |

Chyby se vracejí jako výsledek nástroje označený jako chyba (neznámý běh, chybějící argument). Neplatný token dostane HTTP 401; organizace v plánu nižším než Team se dozví, že MCP server její plán nemá.

Soubor scénáře

Scénář je soubor YAML (obvykle v qa/flows/), který popisuje jedno flow: jeho kroky, šipky mezi nimi a to, co má každý krok splnit. Jeho formát je qawalk.scenario.v1; kaloko validate <soubor> soubor podle něj zkontroluje. Jak scénář napsat krok za krokem: Napsat scénář.

```
schema: qawalk.scenario.v1
id: WEB-12-checkout
title: Checkout
locales: [en, cs]
viewports: { desktop: [1280, 800], mobile: iphone-15 }
environments: [local, staging]
nodes:
- id: cart
  kind: screen
  title: { en: Cart, cs: Košík }
  path: /cart
  purpose: The shopper sees what they are buying and can continue.
  execution: { strategy: playwright, script: steps/cart.mjs, instructions: Add one product and
  open the cart. }
  criteria:
  - { id: AC1, text: Shows the total, check: { type: deterministic, assert: exists, selector:
  .cart-total } }
  - id: payment
  kind: screen
  title: Payment
  path: /checkout/payment
  packs: [a11y]
  edges:
  - { from: cart, to: payment }
```

Klíče nejvyšší úrovně

Povinné jsou schema, id, title, nodes a edges.

| Klíč | Co obsahuje | |---|---| | schema | vždy qawalk.scenario.v1 | | id | id scénáře (malá písmena, číslice, pomlčky); úkol v něm (WEB-12-...) propojí issue | | title | název pro lidi | | kind | task (výchozí) nebo flow | | archived | true skryje vyřazený scénář ze seznamu; jeho běhy zůstanou dostupné | | source | cesta k akceptačnímu plánu (ACCEPTANCE.md), používá ji kaloko validate --coverage | | version | číslo, které zvýšíte, když se flow změní | | locales | jazyky, ve kterých se prochází, např. [en, cs] | | viewports | názvy s velikostí ([1280, 800]), předvolba zařízení (iphone-15, ipad-landscape) nebo seznam předvoleb | | browsers | chromium (výchozí), webkit, firefox | | color_schemes | light, dark | | reduced_motion | no-preference, reduce | | forced_colors | none, active (vysoký kontrast ve Windows) | | display_modes | browser, standalone (nainstalovaná webová aplikace) | | devices | scénáře aplikací: emulátory, simulátory nebo telefony, na kterých se prochází | | environments | prostředí z konfigurace, na kterých scénář smí běžet | | readonly | true u scénáře, který jen čte; na prostředí jen pro čtení je povinné | | platform | web (výchozí), android, ios, electron, desktop, windows | | tags | volné štítky | | nodes | kroky (níže) | | edges | šipky mezi kroky (níže) | | lanes | popisky řádků na plachtě, index = číslo řádku | | issue | úkol: number, repo (owner/name), key (Jira nebo Linear, ENG-42), url, title | | sessions | lidé ve flow, každý s title, volitelně mailbox, account, blocked_on | | project | projekt v Kaloku pro jeho běhy; přebije konfiguraci | | product, module | který produkt a modul z products: flow ukazuje, pro changelog a dokumentaci | | capture | kvalita snímků pro všechny kroky (níže) |

Klíče kroku (nodes)

Každý krok potřebuje id, kind a title.

| Klíč | Co obsahuje | |---|---| | id | id kroku, používá se v odkazech, v --steps a v odkazech na kroky (kaloko:<scénář>/<krok>) | | kind | screen, email, external (obrazovka třetí strany), decision, stack (mnoho

stránek jedné šablony) | | title | text, nebo jeden pro každý jazyk ({ en: Cart, cs: Košík }) | | path | adresa vůči base_url, nebo jedna pro každý jazyk; ve scénářích aplikací deep link; {random} vytvoří adresu, která nemůže existovat | | purpose | jedna věta o tom, k čemu krok slouží; čte ji evaluátor | | lane | řádek na plachtě | | readonly | krok jen čte | | execution | strategy (playwright, script, agent, manual), script (cesta ke skriptu kroku), instructions (povinné), requires (např. browser) | | mail | e-mailové kroky: subject a to | | criteria | co má krok splnit (níže) | | session | který člověk ze sessions krok dělá | | affects | masky souborů, které krok ovlivňuje, pro kaloko walk --affected, nebo always | | packs | hotové sady kritérií: seo, usability, a11y, perf, console, errors, app, design | | capture | kvalita snímků jen pro tento krok | | record | true nahraje video kroku, i když průchod nenahrává; false nikdy | | ignore | oblasti vynechané z porovnání pixelů: { selector } nebo { rect: [x, y, w, h], viewport }, volitelně omezené přes viewports, locales, s poznámkou note | | items | stack kroky: odkud se berou stránky (urls, file, sitemap s match/exclude, crawl, script) | | fields | stack kroky: až 12 sloupců indexu stacku (id, label, from) |

Kritéria

Každé kritérium má id (AC1, AC12b), text pro lidi, volitelně highlight (CSS selektor, který se na screenshotu orámuje) a právě jednu kontrolu check: deterministic, semantic nebo manual. Všechny typy kontrol a jejich volby najdete v Kritéria, kontroly a packy.

Šipky (edges)

| Klíč | Co obsahuje | |---|---| | from, to | id kroků | | label | text u šipky, třeba volba v rozhodnutí | | kind | main (výchozí), nebo alt pro alternativní cestu |

Snímky (capture)

| Klíč | Co obsahuje | |---|---| | format | webp (výchozí), nebo png (bezeztrátový, pro tisk) | | scale | 1 nebo 2 (pixely zařízení na snímku) | | quality | kvalita WebP od 0,3 do 1 | | mask | CSS selektory, které na obrázku zakryje plný obdélník | | annotate | očíslované značky u prvků kritérií s highlight | | full_page | celá stránka (výchozí), nebo s false jen první obrazovka | | record | true, step, flow, failed nebo off; viz kaloko walk --record | | trace | ke každému kroku záznam akcí, požadavků a konzole (výchozí true) |

capture kroku má přednost před scénářem a scénář před konfigurací.

Navrhnete flow se svým agentem

V tomto návodu navrhnete obrazovky jednoho flow jako živé HTML se svým agentem, necháte tým, ať je okomentuje, publikujete verzi a uděláte z ní návrh, se kterým se porovná každá implementace. Potřebujete projekt, ve kterém už proběhl kaloko init, a scénář pro dané flow, třeba `qa/flows/checkout.yml`.

Přidejte designové prostředí

Designové prostředí je složka, kterou CLI při průchodu servíruje. Přidejte ho do `kaloko.config.yml`:

```
environments:  
design: { kind: design, serve: design }
```

Scénář zůstává stejný, jaký používá vývoj a QA: stejné kroky, stejná kritéria. Návrh se tak podle akceptačních kritérií zkontroluje dřív, než ho někdo začne stavět.

Nechte agenta nakreslit kroky

Požádejte agenta, s jakýmkoli design skillem, který váš tým používá:

> Navrhni checkout flow do `design/` podle našeho brand skillu: jedna složka na každý krok z `qa/flows/checkout.yml`, > sdílené CSS v `design/_shared`, tokeny v `design/tokens.json`. Pak spusť `kaloko walk` a `kaloko sync`.

Z každého kroku vznikne `design/<krok>/index.html`: obyčejné HTML, CSS a JS, které se otevře i z disku. Odkazy mezi kroky jsou relativní, třeba `../payment/index.html`.

Projděte, zkontrolujte a synchronizujte

```
npx kaloko start --scenario qa/flows/checkout.yml --env design  
npx kaloko design lint  
npx kaloko walk  
npx kaloko sync
```

`design lint` zkontroluje složku podle pravidel přenosného HTML: relativní adresy, žádné CDN, žádné úložiště v prohlížeči, předvídatelný obsah. `walk` vykreslí jen kroky, které se změnily, a `sync` je uloží do společného konceptu a převezme kroky, které změnili ostatní.

Obrázek: otevřete dokumentaci v Kaloku

Koncept designového flow · cs · desktop

Komentujte společně

Tým otevře koncept na plachtě, proklikne si živé kroky a připíná komentáře k prvkům. Každý vidí, kdo se právě dívá. Agent si komentáře přečte přes `kaloko feedback`, upraví, o co žádají, a synchronizuje znovu.

Obrázek: otevřete dokumentaci v Kaloku

Živý designový krok · cs · desktop

Publikujte verzi

```
npx kaloko share --name "Checkout v2"
```

Tím publikujete verzi celého flow, třeba "v2 · 3 změněné, 17 beze změny". Nezměněné kroky se ukládají jen jednou. `share --notes-preview` vám před publikováním ukáže poznámky o tom, co se změnilo.

Obrázek: otevřete dokumentaci v Kaloku

Publikovaná verze · cs · desktop

Udělejte z ní referenci

Až verzi někdo přijme, nastavte ji jako referenci designu (od plánu Team):

```
npx kaloko reference --design --version 2
```

U kroků implementace ve scénáři zapněte packs: `[design]`. Od plánu Business se pak každý běh implementace porovná s referencí: pixely, struktura, tokeny a kontrast, a kaloko drift napíše, který token použít tam, kde se kód liší.

Když máte návrh ve Figmě

Pokud návrh už žije ve Figmě, kaloko `design import figma <adresa-souboru> --scenario qa/flows/checkout.yml` udělá z framů designové kroky a z proměnných tokeny. Složka framů vyexportovaných z Figmy funguje i bez tokenu.

První běh za deset minut

Na konci tohoto návodu budete mít jedno flow své aplikace na plachtě v kaloko.app, nasdílené, se všemi obrazovkami ve všech jazycích a velikostech obrazovky, které scénář uvádí. Potřebujete Node 20 nebo novější, Chrome a aplikaci spuštěnou lokálně.

Nainstalujte CLI

V kořeni projektu:

```
npm install --save-dev kaloko
npx kaloko init --agent claude --org <vaše-organizace>
```

Pro Codex použijte --agent codex, pro oba --agent all, a když budete CLI ovládat sami, volbu vynechte. init zapíše kaloko.config.yml a zkopíruje skill k vašemu agentovi. Zároveň si projekt přečte: framework, příkaz a adresu dev serveru, jazyky a hlavní cesty. Podle nich připraví první scénář qa/flows/main-pages.yml, takže první průchod funguje bez úprav.

Pokud organizaci ještě nemáte, nejdřív ji založte; stanete se jejím adminem.

Přihlaste tento počítač

```
npx kaloko login
npx kaloko doctor
```

login otevře prohlížeč, kde kód schválíte a vyberete, co smí token dělat. Uloží se do ~/.config/kaloko/.env. doctor zkontroluje Node, Chrome, konfiguraci, prostředí a token a u každého x napíše, co opravit.

Projděte flow

Spusťte dev server a pak:

```
npx kaloko start --scenario qa/flows/main-pages.yml --env local
npx kaloko walk
npx kaloko preview
```

start založí běh, walk otevře v prohlížeči každý krok, zachytí ho a průběžně spouští kontroly. Na konci vypíše verdikty: co neprošlo a na co se má podívat člověk. preview vypíše cestu k místní plachtě, která se otevře přímo z disku.

S nainstalovaným skillem nemusíte nic psát a stačí požádat agenta: "Projdi s Kalokem hlavní stránky na localu a ukaž mi, co neprochází." Spustí stejné příkazy.

Nasdílejte ho

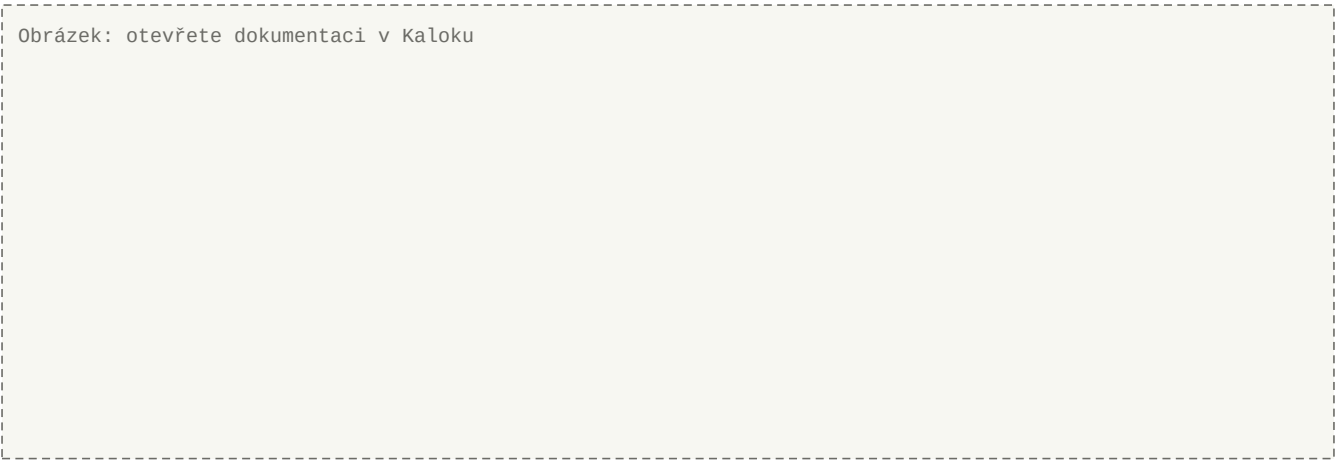
```
npx kaloko share
```

Běh se nahraje do kaloko.app a příkaz vypíše odkaz. Na větvi s pull requestem přidá share --pr odkaz i do komentáře v pull requestu.

Podívejte se a rozhodněte

Otevřete odkaz. Běh se ukáže jako mapa kroků; kliknutím na krok uvidíte snímek, kritéria a co se zjistilo.

Obrázek: otevřete dokumentaci v Kaloku



Mapa běhu · cs · desktop

Kroky, které sedí, schvalte, ostatní vraťte s poznámkou, a až budete spokojeni, běh přijměte. Váš agent si poznámky přečte přes kaloko feedback.

Kam dál

- Napište scénář pro flow, na kterém vám záleží nejvíc.
- Projít běh ukáže plachtu podrobněji.
- Spouštějte ho v CI u každého pull requestu.
- Přehled CLI uvádí všechny příkazy.

Co je nového: Kaloko

0.13.0

Vydáno 2026-10-06

Added

- `kaloko <command> --help` (or `-h`) and `kaloko help <command>` show the help of that one command, with the options every command takes and the exit codes: 0 done, 1 checks found problems or the command failed, 2 a usage error. `kaloko --version`, `-v` and `kaloko version` print the version.
- `kaloko start --json` prints the run id and its folder, `kaloko status --json` what each step has captured.
- `--org <slug>` on every command works with another organization than `org:` in the config.
- API errors name their reason in code next to error. When the plan does not include something (HTTP 402) the answer links the organization's Plan & billing in `billing_url`, and the CLI prints that link.
- Webhooks carry `Kaloko-Signature` and `Kaloko-Event` next to `QAwalk-Signature` and `QAwalk-Event`, with the same values, so receivers written before the rename keep working.
- The trust page names a contact for reporting abuse (phishing, malware) on `kaloko.app`, `files.kaloko.app` and design previews at `*.kaloko-usercontent.com`.
- The trust page explains how Kaloko hides personal data in captures (masks, [hidden] in the stored HTML and the outline evaluators read, pii: redact) and links the how-to. The subprocessor list now names Workers AI at Cloudflare, which Clef triage uses only when an admin turns the experiment on.
- Every guide is also plain Markdown at its own address with `.md` added (`/guides/ci.md`, `/cs/pruvodci/ci.md`), for agents and answer engines. `llms-full.txt` now carries the pricing table and billing questions, the comparisons, the situations and the trust facts.
- The comparison pages answer the questions people ask about them.

Changed

- An option a command does not take gets a "Did you mean `--env?`" instead of being skipped quietly. `prune`, `trash`, `restore`, `keep`, `archive`, `public`, `release` and `comment` stop before they change anything; other commands say it and go on.
- A usage mistake (unknown command or option, missing argument) exits with code 2 and points to `kaloko help <command>`, without the support line.
- `runs`, `prune` and `screens` take `--env` like the other commands; `--environment` still works. `archive` takes `--off` like `keep` and `public` (`--undo` still works), and `prune` takes `--dry-run`, which is what it does without `--yes`.
- A service account token on a plan without service accounts is answered with HTTP 402 (it was 403): the plan, not the token, has to change.
- Webhook requests say `User-Agent: Kaloko-Webhooks/1`.
- Pricing: the print-quality row is gone, a row shows that hiding personal data works on every plan, and another that the GitHub check on the pull request comes with Team. The site says plainly that your data stays readable on every plan and that exporting a run as a ZIP comes with Team.
- Terms of service, version 4: a trial started with an invitation code needs no card; if none is added, the organization returns to Free when the trial ends. The terms and the pricing page call the billing settings Plan & billing, as the app does.
- Kaloko's public changelog and help have one address per language, which search engines and link previews now understand. The Czech changelog page says that the release notes are written in English.
- `/design` and `/cs/navrh` lead straight to the guide on designing with your agent.

Fixed

- `kaloko prune --env preview --yes` pruned the runs of every environment: `--env` was not read. It now prunes only

that environment.

- --key=value keeps everything after the first =: --map="Sign in=login" arrives whole.
- Commands run without org: in the config say how to set it instead of failing with a 404 on /orgs/undefined.
- kaloko runs says when the list stops at the newest 300 and how to narrow it.
- kaloko design import --json prints only the JSON on standard output; progress goes to standard error.

0.12.2

Vydáno 2026-10-06

Fixed

- Czech changelog headings (Přidáno, Změněno, Opraveno, Odstraněno, Zastaralé, Bezpečnost) count as Added, Changed, Fixed and the other Keep a Changelog groups, so a Czech CHANGELOG.md is grouped like an English one.
- A form that needed a fresh second sign-in step (inviting someone, changing a setting) is sent again after the step instead of coming back empty. The fields wait in the browser tab, never on the service.

Changed

- Screens on the canvas cards stay sharp: thumbnails are 640 wide and scaled down step by step (one big jump made small text fall apart), and when you zoom in past what a thumbnail holds, the visible cards load the full screenshot. Runs shared before this get the zoom part too.
- Videos play on: once you play a step's video, the next step's video starts by itself in the step detail, and replay with video moves to the next step when a video ends (a decision waits for you, a still step shows its screenshot for three seconds). The next step's video loads while the current one plays.
- A step's video is at least 2.5 seconds long, held on its last frame, so a quick step no longer flashes past. A step where nothing moved keeps no video at all, and the canvas shows its screenshot.
- kaloko.com and www.kaloko.com lead to the same address on kaloko.app.

0.12.1

Vydáno 2026-10-05

Added

- A trial code from us gives an organization's first subscription a longer free trial without a card. Open the sign-up link with the code, or enter it under Plan & billing. Two weeks before the trial ends, admins are asked for a card; without one the organization goes back to Free on its own and keeps its runs.

Changed

- kaloko share refuses unmasked personal data only in a run that hides it (pii on the run, scenario or environment, or mask: auto); other runs are shared with the same list as a warning. The site's own contacts no longer count: e-mails on the environment's domain, e-mails and phone numbers in the page footer, and those of an Organization or ContactPoint in JSON-LD (pii: { site_contacts: false } checks them again). Sample addresses such as you@company.com, placeholder text and SVG drawings are not read as personal data.
- Phone numbers written 603 123 456 count only next to a word like Telefon or Mobil, and amounts, EANs and order numbers no longer read as phone numbers. The nine-digit birth number needs its label (RČ), as does one without the slash.
- pii: pseudonymize gives one value the same stand-in in every run of the scenario, so screenshots and approvals carry over, and two values never share one. KALOKO_PII_SALT adds a secret of your own.

Fixed

- Hidden form fields, value attributes and personal data in URLs (also encoded, jan%40firma.cz) are masked.
- The capture record no longer keeps real data: page title, headings, JSON-LD, microdata, console lines, blocked requests, the URL and check evidence all go through the mask, and a capture whose masked page

cannot be read stores nothing.

- Masking no longer changes what the page reacts to: console errors are read before it, data-* attributes are masked only in the stored HTML, and the keyboard walk runs on the real page with masked focus shots.
- Short masked values such as 2024 or Praha are no longer replaced across the whole run (width:100% stays).
- Recordings cover personal data in modal dialogs and popovers, open shadow DOM and same-origin frames, and on a new page from its first frame with everything masked so far. Screenshots of steps mask the same places.
- Values masked in one kaloko process are masked in the next process of the run too.
- Triage (experimental) fits long pages into Clef: two page tiles and smaller recording frames instead of four full tiles, and a request still too large is asked again with half the images. A Cloudflare token in the project that cannot use Workers AI hands triage to Kaloko instead of stopping it, and such a token is used only when `evaluators.triage` is set in `kaloko.config.yml`.
- A failure while masking leaves the page as it was.
- Long words without spaces no longer slow down detection.

0.12.0

Vydáno 2026-10-05

Added

- Triage, an experiment an admin switches on in Settings -> Security -> Experiments. kaloko evaluate shows each screen, and a few frames of its recording, to Clef on Cloudflare, which names the kind of problem it sees: overlapping elements, cut-off text, a raw translation key, a placeholder, an error, an endless spinner, a layout that jumps. The canvas lists them under "Possible problems"; one click turns a right one into an ordinary comment, and findings never change a verdict. `evaluators.triage` in `kaloko.config.yml` tunes or turns it off.
- MCP tools `get_findings` and `answer_finding`.
- Masks hide personal data in everything a step stores, not only in the screenshot. The captured HTML and the page outline the evaluators read `get [hidden]` of the same length (`[skryto]` in Czech runs), and traces, recordings, crops, focus shots and the HTML view are covered too. Checks still read the real page, so their verdicts do not change.
- pii: redact on a scenario finds e-mails, phone numbers, birth numbers, IBANs, account numbers, dates of birth, addresses and names in greetings without a selector, in Czech formats too; capture: `{ mask: auto }` does it for one step. `pii.allow` in `kaloko.config.yml` lists values that only look personal, such as your support line.
- pii: pseudonymize replaces personal data with made-up values instead of boxes. One value gets the same stand-in for the whole run, so you can still see a name travel from the form to the summary and the e-mail.
- Captured e-mails hide their recipients (To and Cc, now captured too) whenever a step masks anything; greetings and payer blocks follow the step's masks.
- kaloko share checks the run before it uploads: an unmasked e-mail, phone number, birth number or IBAN stops it, with the step and the element. `--allow-pii` uploads anyway.
- An environment with `pii: required` gets no run that would keep personal data; kaloko start `--pii redact` (or `pii: on` on the scenario or the environment) satisfies it.
- The canvas says "Personal data hidden" on such runs and outlines the masked places in the step detail, so a dark box is not taken for a bug in the app.

Changed

- The masks of a scenario and of its step now add up; before, a step's `capture.mask` replaced the scenario's.
- `kaloko.app` serves its own fonts and the canvas you open from disk carries them inside, so no page asks Google Fonts for anything and Google Fonts is off the subprocessor list on the trust page.

0.11.0

Vydáno 2026-10-04

Added

- The approval page of kaloko login says where the request came from (city, country and address), which terminal asked and when, and starts with a plain warning: approve only a sign-in you started yourself, just now. When the request came from another country than the one you are in, the page says so in red.
- SCIM tokens are valid for a year. Settings -> Security -> SCIM shows the date, and "Renew for a year" keeps the same token, so nothing changes in your identity provider. Admins get a reminder in the inbox 30 days before a token expires. Tokens you already have run for a year from this release.
- A company sign-in over SAML set up before "Require a signed response" existed turns it on with one click in Settings -> Security. Kaloko offers the button once it has seen your identity provider sign its responses, so people keep signing in as before.

Changed

- A passkey used as the second step must ask for your PIN, fingerprint or face. A security key that signs without asking no longer counts; your account page names the key and says how to bring it back (set a PIN on it, or add another passkey).
- A run whose title, environment, branch, commit message or another field is too long is refused at the start of the upload with a message that names the field and the limit, instead of an unexpected error halfway through.

Fixed

- A browser that launched but then stopped answering no longer holds kaloko walk, evaluate --recheck or a capture until someone kills it. A new browser must open its first window within 20 seconds or it is closed and started once more, and later windows have the same limit. kaloko browser start, kaloko doctor, kaloko auth save, the PDF export and connecting to the shared browser now have the time limit and the second try too (KALOKO_LAUNCH_TIMEOUT, seconds).
- A js_equals check on a busy machine no longer fails when the page answers a little late: the expression keeps its 2-second limit, and Kaloko waits up to 17 seconds for the page to reply.

Security

- The sign-in cookie can be set only by kaloko.app itself. You stay signed in while it changes over. Your session also gets a new id at every sign-in and right after the second step.
- App and website pages tell the browser which features they never use (camera, microphone, location, payments and the like), and a Kaloko tab opened from another site is cut off from that site's window.
- Billing events from Stripe and pull request events from GitHub are applied once, also when someone sends a captured copy again.

0.10.0

Vydáno 2026-10-04

Added

- A step can list its other states: states: [empty, error, loading]. Walks run the step script once per state (it gets state), a design draws cart/empty.html next to cart/index.html, and each state gets its own screenshots and verdicts. The canvas has a State switcher next to the language, and review mode and the grid show the state you pick. A criterion with states: [empty] is judged on the empty cart only.
- A design whose token set has several modes, such as two brands, is rendered in each of them. Switch modes on the canvas like colour schemes; tokens_mode in the environment stays the default. Light and dark alone still follow color_schemes. Variants of a stack step are captured in every state and mode too.
- kaloko capture --state, kaloko spec --state and a state option on the MCP step tools (get_step, get_step_spec, get_step_drift, get_step_assets, get_step_recording).

- A design step can come in variants of one template, such as an e-mail for each reason support can pick. Make it a stack step and give each variant a folder, `design/<step>/<variant>/index.html`. kaloko walk captures every variant in every viewport, kaloko sync sends them to the draft, and in a published version the step opens a list of its variants with their results. Each variant opens live. kaloko design lint reports a variant folder without its page, and both design exports include the variant folders.
- Web fonts on live design steps. An organization admin lists the addresses designs may take fonts, stylesheets and images from (Google Fonts, Adobe Fonts, your CDN) in Settings -> Security, and live steps load them instead of falling back to a system font. Scripts still load only from the design itself. List the addresses under origins of the design environment and kaloko design lint says which of them the organization does not allow yet.
- Prototypes remember things between steps: a cart filled on one step shows on the next, in Live mode and in the presentation. The state stays in the viewer's browser tab, every step opened alone still shows its own sample data, and "Start over" in the presentation clears it. Design agents use `window.kaloko.state`.

Changed

- Large runs are quicker on the canvas. On a run of 500 steps in seven languages on three devices, with the CPU slowed down to a phone's, a zoom step takes 33 ms instead of 133 ms, switching between desktop and mobile 650 ms instead of 910 ms, and the map is drawn once when it opens (it was drawn twice). Scrolling the step index no longer sorts every step on each frame.
- On `kaloko.app` a large run's canvas data and a scenario's history page come from a cache after the first visit (27 ms instead of 113 ms, 9 ms instead of 230 ms), and the changelog, the products page, the screen library and review mode ask the database far fewer times.

0.9.0

Vydáno 2026-10-04

Added

- After a release, people who read the product see a short "What's new" bar in the app that links its illustrated changelog. It comes once per release and person; open the changelog or close the bar and it stays gone.
- The canvas tells you when the connection drops or Kaloko cannot save. Comments, approvals and verdicts wait and go out once you are back online, or with "Try again"; nothing is posted twice and nothing is lost quietly.
- `kaloko doctor --fix` installs a missing browser, creates missing folders and fills in org: from your token. An expired token points you to kaloko login.
- Long stack walks show roughly how much time is left.

Changed

- On a phone, review mode puts Approve and Return under your thumb at the bottom of the screen and says what you just decided and which step comes next.
- The first-steps list links straight to where each step is done: your newest run for comments and approvals, with the command to copy for the steps done in the terminal.
- Every error page has a way to write to support, with the address already in the e-mail, and a link to the status page. The app and website footers link the status page too.
- CLI errors say what happened and what to do next (sign in again, wait and retry, check the network) and end with `support@sinfin.cz`. Losing the connection no longer prints a stack trace.

Fixed

- A browser video encoder that stops responding no longer holds kaloko walk `--record`: the recording ends with a note and the walk goes on.
- Dark mode is easier to read: form fields have visible edges, the highlighted plan column on the pricing page is readable, and shortcut keys on the canvas's review buttons stand out.

0.8.0

Vydáno 2026-10-04

Added

- Steps and scenarios say what kind of page they are: page: landing, work, list, settings, form or doc. The usability pack asks questions that fit: "one clear primary action" only on landing pages, and on lists, settings, forms and docs a question of their own. Without page, public pages count as landing pages and app screens as work pages, so a signed-in overview is no longer asked for a single call to action.
- A criterion can apply to some environments only: environments: [production] on hreflang pairs, a live demo or the production sitemap. On a local or staging walk it shows as not applicable to that environment and does not make the step partial.
- kaloko validate warns when a scenario walks several languages and a text check matches only one of them, for example pattern: "Plans and payment" without the Czech wording.
- Empty pages in the app say what to do next and give the command with a copy button: a product without a release or docs, its changelog and docs, an organization without runs, the runs and projects lists, the inbox and the screen library. People who only read a product see a plain sentence instead of a command.
- Kaloko's help covers the whole product, in English and Czech: two tutorials, how-to guides from writing a scenario to company sign-in, a reference of every CLI command, scenario key, check, pack, config key and MCP tool, and explanations of how verdicts are decided and how Kaloko maps to your applications.
- The website links Kaloko's own public changelog and help, so you can see both working before you set them up.

Obrázek: otevřete dokumentaci v Kaloku

The home page · cs · desktop

- Company sign-in over SAML can require a signed response: the identity provider must sign the whole response, not only the assertion. New connections start with it on. Connections set up earlier keep working as before, and Settings -> Security recommends turning it on.

Changed

- Error pages say why you got there and how to go on. A missing page offers the overview or another sign-in; a page that needs a role you do not have offers an e-mail to your organization's admins, already written; an expired link says to ask for a new one; an ended session signs you in and brings you back to the same address.
- Unsent text on the canvas survives. The verdict, return and issue notes keep their draft like comments do, and when your session expires while you write, the canvas says so, keeps the text and returns you to the same step after you sign in again.

Fixed

- A browser that starts but never answers no longer leaves kaloko walk, video or a check waiting: each launch has a time limit and one more try (KALOKO_LAUNCH_TIMEOUT in seconds, default 45).

- kaloko init proposes products from npm, Yarn and pnpm workspaces and from packages/ too, not only from apps/; shared libraries without a dev, start or serve script are left out.
- kaloko stability --pull names the button people use on kaloko.app: Mark a changing region.
- A table in published docs can hold a pipe inside a cell, written as `\|`.
- Hosted runs open only pages whose address leads to the public internet. An address that points into a private network is refused when you schedule the run and again before each run. Saved passwords and cookies are sent only to https addresses.
- Text and attribute checks in hosted runs finish quickly whatever the pattern. Patterns with backreferences or lookahead are reported as not evaluated there, with the reason; kaloko walk still checks them in full.
- Signing in, sign-up, invitation links and the second step have their own limits on repeated attempts. Exports and new API tokens have a generous hourly or daily limit per person.

0.7.1

Vydáno 2026-10-04

Fixed

- kaloko walk of an Android, iOS, desktop or Electron app ends with the verdicts, like a walk of a website; before, the criteria stayed unchecked until you ran kaloko evaluate.

0.7.0

Vydáno 2026-10-04

Added

- Products in the app. The main menu has Products: every product you read with its newest version, its docs, its modules and projects, who reads it and links to its changelog and docs. A project page names the products it belongs to. kaloko products sync puts the products of kaloko.config.yml on Kaloko before their first release, and kaloko products list shows which ones are there.

Obrázek: otevřete dokumentaci v Kaloku

The Products page · cs · desktop

- A trust page at kaloko.app/trust (in Czech at /cs/duvera): where your data lives, how it is encrypted, who can get in, which companies process it and where our SOC 2 preparation stands. Kaloko is not SOC 2 certified yet, and the page says so. The security contact is also in /.well-known/security.txt.

Obrázek: otevřete dokumentaci v Kaloku

The trust page · cs · desktop

- Seat warnings. When 80 % of the paid seats are taken, and again when all are, organization admins see a banner on the home page, in Settings and on the billing page, and get one e-mail per threshold in each billing period. GET /api/v1/orgs/<org>/usage reports the seats too.
- Yearly Enterprise subscriptions can go above their paid seats. Kaloko keeps the highest number of seats in each quarter; after the quarter, Kaloko billing reviews the true-up and invoices the extra seats for the rest of the subscription year or waives them. Seats added in the last quarter are not charged for that year: a week before the renewal the subscription rises to them and the renewal invoice includes them. The billing page shows the current quarter and every earlier true-up.
- Company sign-in over SAML accepts encrypted assertions. Create the organization's key pair under Settings -> Security -> Company sign-in; its certificate is in the SP metadata, and a new key pair keeps the old one working until you remove it.
- Single logout with your identity provider: signing out at the provider signs people out of Kaloko, and signing out of Kaloko can sign them out at the provider too.
- Sign-in from the app tile of your identity provider, off until an admin turns it on, with a default page people land on.

Fixed

- The keyboard checks of kaloko walk no longer report a trap inside a third-party widget such as an anti-bot check, and judge a link wrapped over two lines and a small radio inside its label as people see them. Tab lists with arrow keys count as one tab stop.
- The canvas works better from the keyboard: every control shows a clear focus ring in light and dark mode, a "Skip to content" link jumps past the top bar, and while a step detail or another overlay is open, Tab stays in it instead of wandering through the hidden canvas. Screen readers get named side panels and dialogs.
- The canvas no longer jumps while it loads.
- A link from a comment notification opens the step with its comments in view, on a phone too.
- On a phone, a live page in the step detail fills the room above the sheet instead of a short strip, an open sheet leaves a strip of the preview that you can tap to lower it, and "Ctrl+Enter sends" shows only with a keyboard.

0.6.0

Vydáno 2026-10-03

Added

- A visual changelog. kaloko changelog draft writes the Unreleased entry of CHANGELOG.md from the pull requests since the last release and adds before and after pictures of the screens your accepted runs show differently. kaloko release <version> records the release with its pictures, and kaloko.app shows it as a page you can share with clients, support or anyone outside development.

- Step references in Markdown: always shows the newest accepted screen, and ?version=2.4.0 the screen of that release.

Zatím bez schváleného obrázku

kaloko:checkout/payment

- Products and modules in kaloko.config.yml for repositories with several apps; scenarios say which product they show (product:, module:). Without them the project is the product.
- Readers from your domain: everyone who signs in from a verified domain reads the changelogs, also of restricted projects (Business and up).
- kaloko init asks whether to keep a visual changelog, kaloko doctor checks that every product's version can be read, and agents get get_changelog, draft_changelog_entry and resolve_step_image over MCP.
- Docs that keep up with the app. Guides in your docs/ folder show steps instead of screenshots; kaloko docs publish puts them on kaloko.app with a version picker, language and screen switches and search, and every picture opens its step on the canvas.

Obrázek: otevřete dokumentaci v Kaloku

A guide with its pictures · cs · desktop

- When a screen changes after its guide was written, the section says "the screen changed in 2.4 — check the text", and kaloko docs check lists it with broken references, missing translations and changelog items without a picture, so CI can stop on them.
- A guide plays as a walkthrough, one step at a time with its words as the caption, and readers tick off the steps they tried.
- Exports for your own docs site: kaloko export --format md, docusaurus, vitepress, mkdocs, html, pdf or json, with the pictures as files. The docs and changelog pages offer the same downloads.
- A "What's new" e-mail: readers ask for it on a product's changelog and get each new release with its pictures in their language (Business and up).
- kaloko ci github --release records every release that release-please, changesets or semantic-release

publishes.

- Issues in Jira and Linear from a returned step or a failing criterion, on the canvas, in review mode or with kaloko issue create. The issue gets the screenshot, the criterion, the notes and a link back, and the comment on the step is resolved when the issue is done. Branches and pull request titles that name an issue link the run to it.
- Slack and Teams channels per project, each with the events it wants and quiet hours. Restricted projects reach only channels chosen for them.
- A short list of first steps on the home page (share, comment, approve, invite, CI, release) until the team has done them.
- Hosted runs: Kaloko opens the pages of a read-only scenario every day or week and shares the run (kaloko schedule add, Business). kaloko schedule export github writes a scheduled workflow for the full walk.

0.5.2

Vydáno 2026-10-03

Added

- The first version of the visual changelog: kaloko changelog draft, kaloko release, step references and the changelog page on kaloko.app with a before and after slider. It is described in full under 0.6.0.

0.5.1

Vydáno 2026-10-03

Added

- Swipe and onion skin compare next to side by side and diff, in the step detail and in review mode.
- Regions to ignore can be drawn by hand on a capture. The canvas diff skips them, and kaloko stability --pull writes them into the scenario.
- A grid (key G) shows one step in every language and screen size at once, also on a phone and in review mode.
- Comments can carry drawn boxes, arrows and lines, and attached images. Agents get the marked region in pixels.
- An accepted run has a signed acceptance record: a PDF with thumbnails and a printable page. kaloko signoff --verify checks the signature.

0.5.0

Vydáno 2026-10-03

Added

- Checks of the page itself: the address a step ends on, computed styles of an element and values the page exposes, so scenarios no longer need data-qa markers.
- Runs in WebKit and Firefox besides Chromium, on device presets (phones, foldables, tablets, TV), in dark mode, with reduced motion, forced colours or as an installed web app. Every step is captured in each combination.
- App walks on several devices at once: Android emulators, iOS simulators, a real iPhone and Android WebViews, and Windows apps next to macOS ones.
- One picker on the canvas for language, browser and colour scheme that stays usable with many values.

Obrázek: otevřete dokumentaci v Kaloku

The canvas of a run · cs · desktop

- Workflows written by kaloko ci install the browser engines your scenarios walk.

Fixed

- Design drafts keep the browsers, schemes and device presets of their scope.
- An e-mail captured during a walk in several browsers counts once.
- Language and theme switchers work in the phone menu.
- Ignored regions, imported screenshots and cached evaluator answers carry over between capture variants.

0.4.11

Vydáno 2026-10-03

Added

- Recordings: kaloko walk --record films each step from the previous screen to its capture, or the whole flow with a chapter per step, and keeps a trace of actions, requests and console messages for every step.
- The canvas plays a step's recording and shows its trace.

Obrázek: otevřete dokumentaci v Kaloku

A recording on the canvas · cs · desktop

- kaloko video export turns the recordings into one walkthrough video with a title card per step.
- Shared runs carry their recordings on the Business and Enterprise plans, kept for 30 days unless the organization chooses otherwise.

Fixed

- A recording too large to upload stays in the local preview instead of failing the share.

0.4.10

Vydáno 2026-10-03

Added

- kaloko ci github|gitlab writes a workflow that walks what each pull request changes on its preview deployment and comments on the pull request; kaloko ci preview-url finds the preview address in CI.
- kaloko import lays out screenshots, recordings and traces that an agent or Playwright already produced as a run.
- kaloko walk --affected and kaloko affected walk only the steps that the changed files reach, and say why.
- kaloko walk --heal proposes a new selector when the page lost the old one, and applies it once the step passes.
- Several languages walk at the same time (--concurrency).
- kaloko init recognises the framework, the dev server, languages and main routes, and drafts a first scenario.
- Uploads skip screenshots the service already has and resume after a broken connection.

0.4.9

Vydáno 2026-10-03

Added

- A vision evaluator that looks at the screenshot and says why, keyboard and screen reader checks, and regions that change on every run can be ignored.
- Dev mode in the step detail of a design: box model, tokens, redlines, states and assets.

Obrázek: otevřete dokumentaci v Kaloku

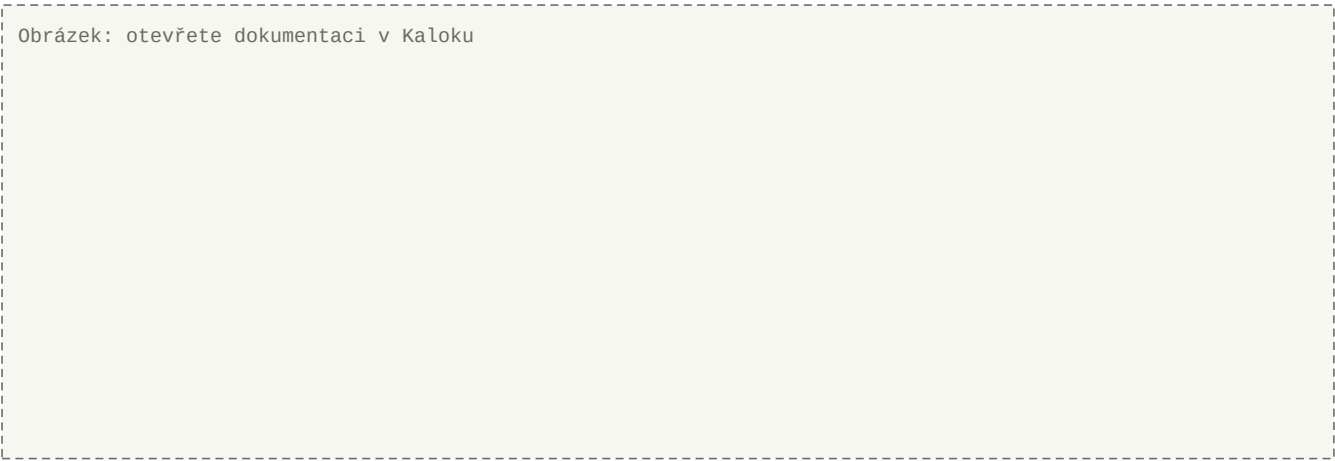
Dev mode in the step detail · cs · desktop

- kaloko spec tells an agent what to build for an approved design step, and kaloko drift what its implementation does differently.

Fixed

- The canvas on phones: a smaller stack badge, readable stack lists, the step header above its tabs and no sideways scrolling.

Obrázek: otevřete dokumentaci v Kaloku



A stack page on a phone · cs · mobile

- Page counts in Czech use the right plural.

0.4.8

Vydáno 2026-10-03

Added

- Review mode: the canvas walks a person through what still needs a decision, with keyboard shortcuts, and kaloko review lists the same for agents.
- Approvals carry over to steps whose screenshots did not change since the last accepted run.
- Every new organization gets a sample run with a guided first review.
- Two-step verification with passkeys or an authenticator app, and an organization rule that requires it.
- An audit log for admins, tokens limited to chosen projects, and company sign-in with SAML or OIDC, SCIM and an IP allowlist on Enterprise.

Fixed

- Plan badges on the pricing page no longer overflow on phones.

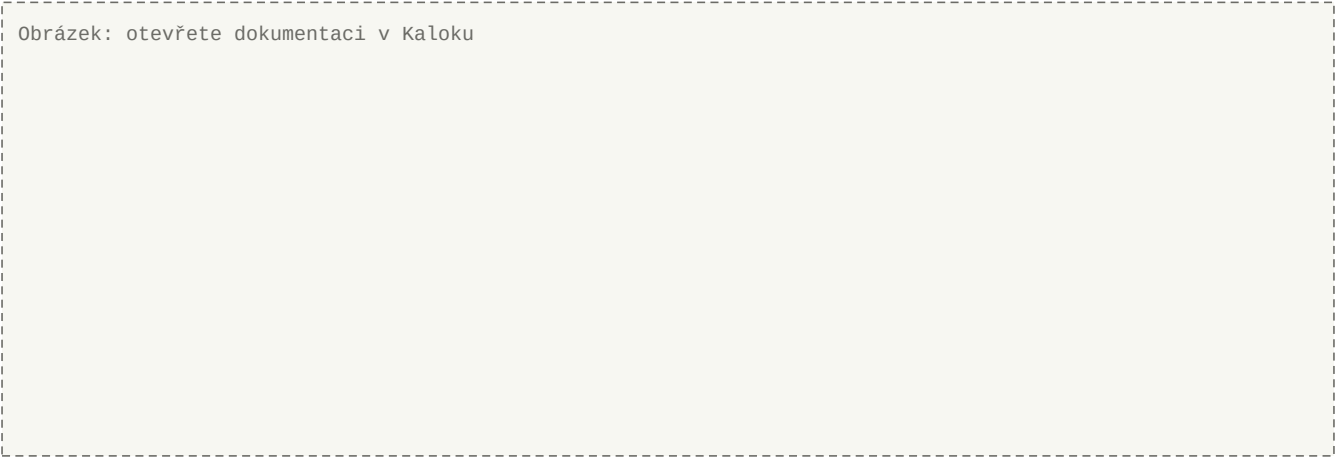
0.4.7

Vydáno 2026-10-03

Added

- Sign in with Google, Microsoft or GitHub where the service offers it.

Obrázek: otevřete dokumentaci v Kaloku



The sign-in page · cs · desktop

- kaloko login approves a token in the browser instead of copying it from Settings.
- Invitations: people outside the organization join single projects as guests, and the Share dialog of a project shows who has access.

0.4.5

Vydáno 2026-10-03

Added

- A Designer role and API tokens with chosen scopes (read, comment, design, upload).
- Focus mode shows the preview alone, and the step detail has a layout for phones.

0.4.4

Vydáno 2026-10-02

Fixed

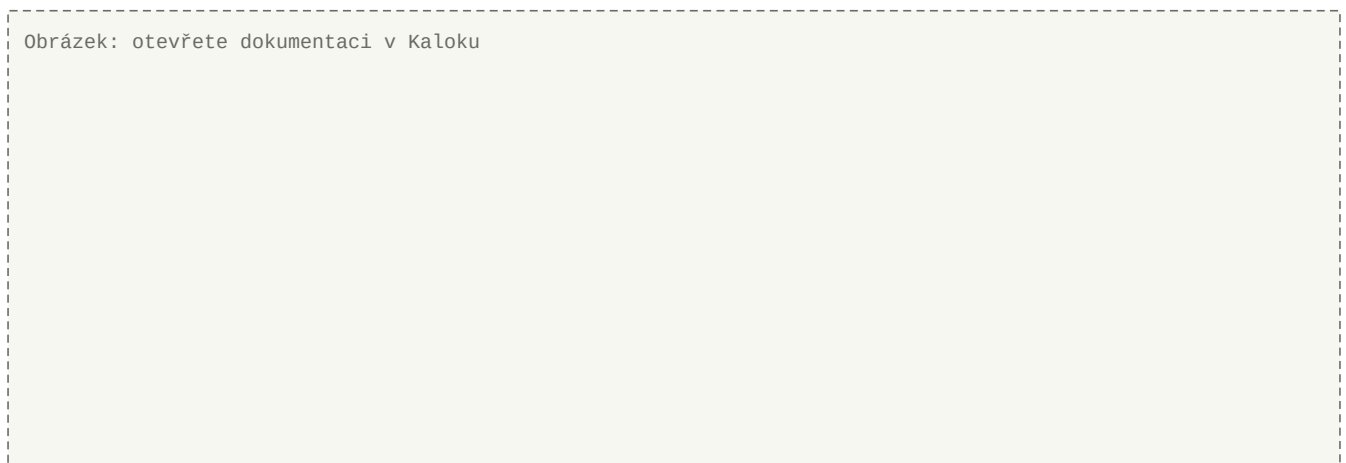
- kaloko evaluate --recheck says that it asks the semantic questions again too.
- A lane on the canvas is as tall as its tallest card.

0.4.3

Vydáno 2026-10-02

Fixed

- Signed-in pages that Kaloko's own walk found wanting: labelled fields, real settings tabs, cards on phones and the whole inbox.



Settings · cs · desktop

- The canvas has a main landmark, readable badges and a visible compare button, and no longer shifts while loading.
- Pages load their fonts sooner and keep their layout when the font arrives.

0.4.2

Vydáno 2026-10-02

Added

- The Kaloko wordmark with a yellow marker and a new icon.

Obrázek: otevřete dokumentaci v Kaloku

The home page · cs · desktop

- The Czech website lives under /cs/, so each language has its own address.
- Pages of a stack may carry their own purpose for the semantic evaluator.
- Live design revisions open on their own domain.

Fixed

- kaloko doctor prints the config file it actually read.

0.4.1

Vydáno 2026-10-02

Added

- links_ok and hreflang_pairs checks for pages and stacks.

Fixed

- Unknown addresses on kaloko.app answer 404 instead of the sign-in form.
- Every language version of the website answers at its own address.

0.4.0

Vydáno 2026-10-02

Changed

- QAwalk is now Kaloko: the CLI is kaloko (the qawalk command and qawalk.config.yml keep working) and the service lives at kaloko.app.

Obrázek: otevřete dokumentaci v Kaloku

The landing page · cs · desktop

Added

- Kaloko Design presents interactive HTML designs, lints pages that must work from disk, and exports a design in open formats.

0.3.0

Vydáno 2026-10-02

Changed

- The CLI is on npm as kaloko instead of @qawalk/cli. The qawalk command, qawalk.config.yml, the QAWALK_* variables and ~/.config/qawalk/.env keep working, and kaloko init replaces the old skill.
- The app and the website share kaloko.app. Pages on qawalk.com redirect there; the API, MCP, webhooks and file links on the old addresses keep answering.

Added

- A "Group by issue" switch on the lists of runs and scenarios. The lists remember it, and "Latest per scenario" too.

0.2.22

Vydáno 2026-09-30

Fixed

- Large runs open faster on the canvas, and kaloko evaluate needs less memory for big stacks.

0.2.21

Vydáno 2026-09-30

Fixed

- A kept run cannot be trashed, and releasing it gives it at least a week. Kept design versions are not pruned.
- The "no project" filter on runs works, failed step details load again, and an unusual cookie no longer breaks the lists.
- Lists, the home page and scenario history load faster.

0.2.20

Vydáno 2026-09-30

Added

- The project you pick is remembered across the lists and the scenario browser on the canvas.
- Scenarios and runs of one issue, or of one pull request when there is no issue, are listed together.

0.2.19

Vydáno 2026-09-30

Changed

- Shortcuts and help open from a keyboard button in the top bar, and the list now includes <- -> and Esc.

0.2.18

Vydáno 2026-09-29

Changed

- A shorter top bar on the canvas: Projects, Scenarios and Runs, the view switch, and the inbox at the right end.

0.2.17

Vydáno 2026-09-29

Changed

- Projects, scenarios and runs share one layout with the navigation on the left, so switching views does not jump. "All runs" is now "Runs".
- The run verdict has Accept and Return as the main buttons; note, baseline and public sit below them.

0.2.16

Vydáno 2026-09-29

Added

- A projects page: scenarios, runs and how the newest runs stand in each project.
- Console errors in the step detail are highlighted, wrapped and can be copied one by one or all at once.
- Times follow the reader's time zone, set in Settings -> Profile or taken from the browser.

Changed

- Screen size switches say the name and width (Desktop 1440, Mobile 360), and a click anywhere on a card opens the step.

0.2.15

Vydáno 2026-09-29

Changed

- The step detail has three tabs: Criteria, Page and Technical. Criteria that need attention come first; passing ones take one line.
- When only phones are shown, the map shows larger phone screenshots.

0.2.14

Vydáno 2026-09-29

Changed

- Evidence of built-in checks reads as a sentence in Czech or English ("TTFB 950 ms, the limit is 800 ms"); the raw text stays under Technical detail.

0.2.13

Vydáno 2026-09-29

Added

- Every criterion of the built-in packs says why it matters and how to fix a failure, in Czech and English. The walk summary prints the fix for the agent.

0.2.12

Vydáno 2026-09-29

Changed

- The run summary sorts problems by what to do: what failed and why, what a person should decide, and screens that look off.

0.2.11

Vydáno 2026-09-29

Added

- Quick signals on every screen: the evaluator also says whether the page does what it is for, is in the expected language or looks broken. Signals never change a verdict.
- kaloko walk evaluates while it walks and ends with a list of what failed or is unsure.

0.2.10

Vydáno 2026-09-29

Added

- Every criterion says how it was checked, in words: the check as a sentence, or the question the AI was asked with its score and the pass threshold.
- Hosted AI evaluation from the Team plan up, with a monthly allowance shown on the billing page and in kaloko doctor. Your own evaluator key still works.
- Design tokens in the current DTCG format, with aliases, modes and deprecated tokens; kaloko tokens validate.

Fixed

- Manual decisions show the criterion on its own line, with reason and verdict below.

0.2.9

Vydáno 2026-09-29

Added

- Scenarios come first in the navigation, and Runs show the newest run per scenario unless you filter.
- Keep a run so it never expires, prune old runs (dry run first) and archive scenarios you no longer walk, in the app or with kaloko runs, trash, restore, prune, keep and archive.

0.2.8

Vydáno 2026-09-29

Changed

- A claimed sign-in domain lets people join only once it is verified.
- Comments on public runs name nobody, and upload tokens cannot delete runs.
- Below 1500 px the navigation and view options fold into one menu.

Fixed

- The canvas shows the map sooner and loads step details after it; large runs upload faster.
- When the service asks the CLI to slow down, it waits and tries again.

0.2.7

Vydáno 2026-09-29

Added

- Comments on steps, whole runs and whole flows: threads, pins on the screenshot, resolve and reopen. Mentions are mailed at once, the rest goes to the digest, and agents use kaloko comment or MCP.
- Kaloko Design: design drafts as live HTML steps on the canvas, versions, an inspector with token names, comments pinned to elements, edits on the canvas, branches linked to git, and a design pack that checks an implementation against the approved design.

- An HTML view of captured pages (experimental) that can be forked into a design draft.

Changed

- A calmer run view on smaller screens: one top bar, view options in a popover, help behind ?.

Fixed

- A long step path no longer widens its card over the next one.
- macOS desktop captures ask for the Screen Recording and Accessibility permissions they need.

0.2.6

Vydáno 2026-09-29

Added

- Mobile and desktop apps: Android, iOS Simulator, Electron and macOS windows, with UI tree checks, touch target sizes, crash detection and an app pack. kaloko capture --image imports screens from other tools.
- A run index next to the map (key I): every step in a list with filters, search and CSV, and stacks as folders.

0.2.5

Vydáno 2026-09-29

Added

- The HTTP status of every capture, an errors pack for error pages, and {random} in a path for an address that cannot exist. kaloko draft adds a not-found step.
- The run card links the issue next to the pull request.

0.2.4

Vydáno 2026-09-29

Added

- Stacks: one step stands for many pages of one template, taken from a list, a sitemap, a crawl or a script. The canvas shows them as a stack with an index you can filter, search, sort and export as CSV.

Obrázek: otevřete dokumentaci v Kaloku

A stack opened as a folder · cs · desktop

0.2.3

Vydáno 2026-09-28

Added

- acceptDialogs() for step scripts, flags on text checks, kaloko evaluate --recheck after a scenario fix, and kaloko

walk --ephemeral for parallel walks.

Fixed

- kaloko mail reads only messages received since the run started.
- A walk captures fresh screens after --steps, basic auth survives a redirect, and only one walk at a time uses a shared browser.

0.2.2

Vydáno 2026-09-28

Added

- kaloko draft <url...> writes a first scenario and acceptance plan for public pages.
- Check packs a11y, perf and console.
- Live presence on the canvas: who is looking, their cursors, and following someone's view.
- Large flows stay readable: lanes with headers, long lanes wrapped into rows, tidy edges and a zoomed-out view.
- Scenario history with pass rate and criterion stability, and kaloko calibrate to tune evaluator thresholds.
- Print-quality captures for documentation, with masks over personal data and numbered marks.
- The newest capture of every step at a stable address, a list of changes, signed webhooks and kaloko export --scenario for documentation (Business and up).
- GitHub checks on pull requests that complete when the run is accepted or returned.
- A sign-in domain can be proved by signing in with Google Workspace or Microsoft 365, and Settings shows the DNS host with copyable record fields.

Fixed

- The overview follows reviewers' criterion verdicts.
- Public mailbox domains such as gmail.com cannot be claimed.
- A walk stops a session after three steps fail with the same error, instead of trying a wrong password again and again.
- An open menu or popover stays on the capture, signed links in the page lose their credentials, and text check evidence shows the actual match.
- kaloko share says when the plan grants a shorter lifetime than requested.

0.2.1

Vydáno 2026-09-25

Added

- Protected environments: HTTP Basic, service-token headers and client certificates, sent only to the environment's own addresses.
- Accounts per scenario with password and TOTP, and kaloko auth save for a single sign-on a person does once.
- \${VAR} in base_url for review apps, and secrets named in the config are removed from captures.
- Any mailbox through a script adapter, and kaloko mail --file for saved messages.

0.2.0

Vydáno 2026-09-25

Added

- The CLI is on npm (as @qawalk/cli until 0.3.0).

- Plans with a billing page, a 14-day trial and a public pricing page.
- Named sessions: several people in one flow, each in their own browser, shown in swimlanes on the canvas.
- Run verdicts, a baseline run, required approvers and reviewer verdicts for manual criteria.
- Compare a run with the baseline or any earlier run: pixel differences, changed criteria and evidence, and kaloko compare.
- Replay walks the run one step at a time with a branch choice at decisions, and the canvas has a minimap.
- An inbox with notifications, a daily or weekly digest, and Slack.
- kaloko share --pr comments on the pull request; kaloko export and the canvas download a run as a ZIP.
- An MCP server for agents.
- Check packs seo and usability, plan coverage with kaloko validate --coverage, and flaky criteria.
- A public demo run anyone can open without signing in.
- Projects per scenario, and a runs list filtered by project, scenario, verdict, author, environment and branch.
- Sign-in domains verified by a DNS record, token scopes and expiry, a list of your sessions, and a 7-day trash.
- kaloko doctor checks your setup and says what to fix.
- Guides on the website for use cases and agents.

0.1.0

Vydáno 2026-09-23

Added

- The first version. The CLI walks scenarios in Chrome, captures every step, runs the checks and asks an AI evaluator what a check cannot answer. A read-only policy keeps production safe.
- The canvas shows the run as a map of screens with criteria, page metadata (Open Graph, hreflang, structured data, headings) and links to a single step. A click on a criterion highlights it on the screenshot.
- The agent skill for Claude Code and Codex, installed with kaloko init --agent.
- The service: organizations by e-mail domain, sign-in by link, roles and API tokens. kaloko share uploads a run, reviewers approve or comment on each step, and kaloko feedback brings their notes back to the agent.
- The app and the website in Czech and English, with sign-up for new organizations.