



SEDM VĚCÍ, CO SI KAŽDÝ PLETE

Skill, plugin, hook, subagent. Všechno to vypadá stejně a dělá to něco jiného. Rozdíl je v jediné věci: kdy se to načte a kdo to spustí.

1. VŽDYCKY

Claude to má v hlavě od první vteřiny každé konverzace.

2. NA ZAVOLÁNÍ

Leží stranou. Claude ví, že to existuje, a celé si to otevře, až je to potřeba.

3. SÁM OD SEBE

Spustí se na událost, i když si na to nikdo nevzpomene.

SEDM VRSTEV, LIDSKY

01 CLAUDE.MD

VŽDYCKY

Textová poznámka pro Clauda. Přečte si ji na startu každé konverzace, ať řešíš cokoliv.

PŘÍKLAD „Piš česky. Nikdy nemaž soubory bez ptaní.“

02 SKILL

NA ZAVOLÁNÍ

Uložený postup. Celý se načte, až když je potřeba, takže může být klidně na tři stránky.

PŘÍKLAD Celý postup, jak vyrábíš fakturu. Zavoláš ho jedním slovem.

03 SLASH COMMAND

NA ZAVOLÁNÍ

Příkaz s lomítkem na začátku zprávy. Spustí vestavěnou funkci, nebo tvůj skill.

PŘÍKLAD /clear vyčistí konverzaci. /faktura spustí ten tvůj postup.

04 SUBAGENT

NA ZAVOLÁNÍ

Pomocník s vlastní hlavou. Pošleš ho na špinavou práci a vrátí ti jen výsledek.

PŘÍKLAD Projde sto souborů a napíše tři věty. Tobě zůstane čistý stůl.

05 HOOK

SÁM OD SEBE

Automatická pojistka. Nastane událost a hook se spustí, i když si Claude nevzpomene.

PŘÍKLAD Pokaždé, když Claude uloží soubor, proběhne kontrola. Bez ptaní.

06 MCP

NA ZAVOLÁNÍ

Kabel do cizí aplikace. Bez něj Claude o tvém Notionu ani mailu vůbec neví.

PŘÍKLAD Připojíš Gmail a Claude ti rovnou připraví odpověď.

07 PLUGIN

OBAL

Krabice na všechno výše. Běžně neumí nic navíc, zato to zabalí a pošle kolegovi.

PŘÍKLAD Nainstaluješ hotový plugin a máš nové příkazy během minuty.

A JEDNA POJISTKA

Pravidlo v textu je prosba. Claude ho většinou dodrží, ale zaručené to není.

ZAPAMATUJ SI Když něco musí platit vždycky, patří to do hooku, ne do poznámky.

CO SI LIDI PLETOU

CLAUDE.MD VS SKILL Poznámku čte pokaždé. Skill až když ho potřebuje.

SKILL VS PŘÍKAZ Skill je ten obsah. Lomítko je jen spínač.

SKILL VS SUBAGENT Skill je návod. Subagent je pracant s vlastní hlavou.

SKILL VS HOOK Skill si Claude přečte a rozhodne se. Hook při své události střílí vždy.

MCP VS SKILL MCP je kabel k aplikaci. Skill je návod, jak ji použít.

PLUGIN VS ZBYTEK Běžný plugin neumí nic navíc, jen to zabalí.

ZAČNI TÍMHELE

Nezakládej sedm věcí první den. Napiš si CLAUDE.md a dej do něj tři pravidla. Až Claude podruhé udělá stejnou chybu, přidej řádek. Až potřebíš kopíruješ stejný postup do chatu, udělej z něj skill. Zbytek počká, až tě něco naštve.

CLAUDE.MD



JEDEN SOUBOR, KTERÝ PŘESTANEŠ VYSVĚTLOVAT

Textový soubor ve složce tvého projektu. Claude si ho přečte sám na začátku každé konverzace, takže mu nemusíš pořád dokola opakovat to samé. **Píšeš ho normálními větami a za pět minut je hotový.**

BEZ NĚJ

Každou novou konverzaci začínáš vysvětlováním. Claude hádá, kde co je, a dělá pořádku samou chybu.

S NÍM

Startuje s tvými pravidly v hlavě. Ty píšeš rovnou zadání, ne úvod do projektu.

JAK HO ZALOŽIT

Všechno v aplikaci Claude, myší a jednou větou.

KROK 01

Otevři aplikaci Claude a klikni na záložku **Code**.

KROK 02

Vyber **Local**, klikni na **Select folder** a najdi složku svého projektu.

KROK 03

Do pole napiš tenhle příkaz. Claude projde soubory a napíše první verzi sám.

/init

KROK 04

Přečti si to. **Smaž, co nesedí. Přidej, co víš jenom ty.** Tohle je ta důležitá část.

Záložka Code se otevře jen na placeném účtu (Pro, Max, Team nebo Enterprise). A když už CLAUDE.md v projektu máš, **/init ho nepřepíše**, jenom navrhne, co doplnit.

CO V NĚM BUDE

Takhle vypadá hotový soubor. Klidně si ho opiš.

Soubor **CLAUDE.md** ve složce projektu

Web pro kavárnu U Kohouta

Jak se mnou mluví

- Odpovídej mi česky.
- Vysvětluj jednoduše, nejsem kodér.
- Než něco smažeš, vždycky se zeptej.

Pravidla webu

- Barvy jsou tmavě zelená a krémová. Jiné nepřidávej.
- Fotky neupravuj, jsou od fotografa.
- Telefon 777 123 456 je správně.

Na tohle sám nepříjdeš

- Ceny v menu schvaluje majitelka.
- Nikdy je neměň, jenom mi je navrhní.

JAK SE MNOU MLUV

Jazyk a styl. Ať ti neodpovídá slovníkem, kterému nerozumíš.

CO MÁ DODRŽET

Věci, které platí pokaždé. A čeho se nemá ani dotknout.

KONKRÉTNÍ ČÍSLA A JMÉNA

Telefon, adresa, barva. Když je má napsané, nemusí je hádat.

CO VÍŠ JENOM TY

Proč je něco tak, jak to je. Tohle z projektu nevyčte nikdo. Nejcenější řádky souboru.

NADPISY S KŘÍŽKEM

Dělají v textu pořádek. Claude se v něm orientuje stejně jako ty.

KAM SE ULOŽÍ

Pravidla pro tenhle jeden projekt

CLAUDE.md

Do složky projektu, vedle ostatních souborů. Tady začni, sem patří skoro všechno.

Tvoje pravidla do všech projektů

~/ .claude/CLAUDE.md

Platí ve všech projektech na tvém počítači. Hledat ho nemusíš, otevře ti ho /memory.

Když máš víc souborů najednou, Claude si přečte **všechny**. Nepřepisují se, poskládají se za sebe.

KDY DO NĚJ ZASE SÁHNOUT

Nepiš ho dopředu. Doplníš ho, až když tě něco naštve.

Claude udělal stejnou chybu podruhé. Napiš jeden řádek, ať ji nedělá potřetí.

Píšeš do chatu opravu, kterou jsi psal už minule. Přesně ta věta patří do souboru.

Musel bys to vysvětlovat někomu novému. Co potřebuje on, potřebuje i Claude.

TŘI CHYBY, CO DĚLÁ KAŽDÝ

„Napišu tam radši všechno hned.“

Nepiš. Čím delší soubor, tím hůř se ho Claude drží. **Drž se pod 200 řádky.**

„Napišu to obecně, ať to platí vždy.“

„Dělej to pořádně“ neznamená nic. „**Než něco smažeš, zeptej se**“ znamená.

„Je to napsané, takže to platí.“

Claude to bere jako prosbu, ne jako zákon. **Většinou poslechne. Vždycky ne.**

/memory

Vypíše tvoje soubory a otevře je k úpravě. **Hledat je po složkách nemusíš.**

/context

V seznamu **Memory files** uvidíš, co se fakt načetlo. Chybí tam? Claude to nevidí.

ZAČNI
TÍMHLE

Spust **/init** a nech Clauda napsat základ. Pak už nic nepiš. Až tě podruhé naštve tou samou věcí, přidej jeden řádek. Za měsíc máš dvacet řádků, které fakt platí, místo dvou set vymyšlených.



POSTUP, KTERÝ NAPIŠEŠ JEDNOU A PAK UŽ JEN ZAVOLÁŠ

Složka s jedním textovým souborem. Uvnitř je postup, který dneska lepíš do chatu pořád dokola. Napišeš ho jednou a pak stačí lomítko a jedno slovo. **Claude si ho otevře, až ho potřebuje, a ty už ho nediktuješ po krocích.**

KDY HO VYROBIT Nevymýšlej ho dopředu. Poznáš to sám, ve všech třech případech.

Potřetí lepíš do chatu ten samý postup. Podruhé to ještě přejdeš, potřetí už ne.

V CLAUDE.md ti vyrostl návod. Pravidlo tam patří. Postup na pět kroků už ne.

Máš checklist, na který si nikdy nevzpomeneš celý. Ať si ho pamatuje on.

JAK HO ZALOŽÍŠ Nic neinstaluješ. Složku i soubor udělá Claude sám.

KROK 01

V aplikaci Claude otevři záložku **Code** a svojí složku. Stejně místo jako u CLAUDE.md.

KROK 02

Popiš postup normální větou, přidej tuhle větu a krátké jméno.

udělej z toho skill

KROK 03

Claude soubor napíše a ukáže ti ho. **Podle režimu se před uložením zeptá, nebo ho uloží rovnou.**

KROK 04

Dopíš pravidla, která znáš jenom ty. Pak napiš lomítko a jméno a zkus ho naostro.

Když v projektu zakládáš skill úplně poprvé a Claude ho nenajde, napiš `/reload-skills`. Další skilly se už načtou samy.

JAK VYPADÁ HOTOVÝ Kávárna U Kohouta odpovídá na recenze. Takhle to má napsané.

Soubor **SKILL.md** ve složce **recenze**

description: Napiše odpověď na recenzi hosta. Použij, když přijde nová recenze na Googlu.

Odpověď na recenzi

- Najdi v recenzi konkrétní věc a odpověz na ni.
- Napiš tři až čtyři věty. Ne víc.

Pravidla

- Hostům vykáme. Podepisuje se Jana, majitelka.
- Žádné „mrzí nás to“. Buď konkrétní.

Tohle nech na mně

- Slevu ani voucher nikdy nenabízej.
- Recenze pod dvě hvězdy mi jen ukaž.

TŘI POMLČKY

Dva takové řádky svírají hlavičku souboru. Do ní patří jedna věta pro Clauda.

DESCRIPTION

Jediné, co Claude vidí předem (spolu se jménem). Napiš, co skill dělá a **kdy ho použít**.

POSTUP V KROCÍCH

Pořadí, ve kterém to děláš ty. Klidně deset kroků, tady se vejdou.

PRÁVIDLA

Věty, které jinak píšeš do chatu pokaždé znovu. Přesně ty patří sem.

HRANICE

Čeho se nemá dotknout. Nejdůležitější odstavec u všeho, co jde ven ke klientovi.

PROČ SMÍ BÝT DLOUHÝ Tohle je celý rozdíl mezi dvěma soubory.

CLAUDE.MD

Načte se celý na začátku každé konverzace, ať řešíš cokoliv. Proto se drží pod dvěma sty řádky.

SKILL.MD

Claude má v hlavě jen tu jednu větu nahoře. Zbytek se natáhne, až když skill fakt použiješ. **Proto unese pět set řádků.**

KAM SE ULOŽÍ Jméno složky je to, co píšeš za lomítko.

Postup jen pro tenhle projekt

`.claude/skills/recenze/SKILL.md`

Ve složce projektu. Tady začni. Když složku sdílíš s kolegou, má skill taky.

Postup do všech projektů

`~/ .claude/skills/recenze/SKILL.md`

Platí všude na tvém počítači. Sem patří to, co používáš pořád dokola.

Do složky můžeš vedle **SKILL.md** přihodit šablonu, ukázkou hotového výstupu nebo ceník. Claude je otevře, až když je potřebuje.

JAK SE SPUSTÍ Dvě cesty a jedna pojistka.

ZAVOLÁŠ HO TY

Napišeš lomítko a jméno. Načte se vždycky, ať se Claude rozhoduje jakkoliv.

VEZME SI HO SÁM

Napišeš „přišla nám recenze“ a Claude ho použije. Pozná to podle **description**.

KDYŽ TO NECHCEŠ

Do hlavičky přidej řádek **disable-model-invocation: true**. Pak startuje jen z tvój ruky.

Skill je pořád jenom text, který si Claude přečte. **Většinou ho dodrží do puntiku, zaručené to ale není.** U věcí, co jdou ven, si výsledek přečti.

TŘI CHYBY, CO DĚLÁ KAŽDÝ

„Do description napíšu nadpis.“

Samotné „Recenze“ nestačí. **Napiš do ní slova, která bys nahlas řekl kolegovi.**

„Nespustil se sám, je rozbitý.“

Většinou není. Nejdřív ověř, že ho vůbec vidíš. **Pak dopiš do description, kdy ho použít.**

„Nacpu tam všechno, co dělám.“

Jeden skill je jeden postup. **Drž se pod pěti sty řádky** a zbytek dej vedle jako druhý soubor.

`/skills`

Vypíše, co máš k dispozici. Většinu skillů tu jde i vypnout.

„Jaké skilly máš k dispozici?“

Otázka normální větou. Když ti tvůj nový skill nevymenuje, **nevidí ho.**

ZAČNI TÍMhle

Nevymýšlej seznam skillů. Otevři poslední konverzaci a najdi postup, který jsi do ní psal potřetí. Řekni Claudovi, ať z něj udělá skill. **Za deset minut máš první a hned víš, jestli to k něčemu je.**

/PŘÍKAZY



ZKRATKY, KTERÉ NEMUSÍŠ PSÁT VĚTOU

Napišeš do pole lomítka a aplikace ti sama nabídne seznam. Nejsou to zadání pro Clauda, jsou to páčky, které něco udělají hned. Čtyři z nich potřebuješ denně, na zbytek si vzpomeneš, až budeš muset.

ČTYŘI, CO POTŘEBUJEŠ DENNĚ

Kdyby sis měl zapamatovat jenom čtyři řádky z celého plakátu, tak tyhle.

/clear Založí novou konverzaci s prázdnou hlavou. Tvoje pravidla z CLAUDE.md zůstávají. Nový úkol si zaslouží čistý stůl.	/context Barevná mřížka ukáže, čím je hlava zaplněná. Když dojíždí, Claude začne být nepozorný.	/compact Zhustí konverzaci a uvolní místo. Zůstane shrnutí, ne celé povídání. Když nestíháš uklidit sám.	/rewind Vrátí úpravy i konverzaci zpět. Změny z příkazů v terminálu ne.
--	---	--	--

ZBYTEK PODLE SITUACE

Nic z toho se neučíš z paměti. Přečti a vrať se, až situace nastane.

Zakládáš nový projekt	/init /memory	Claude projde soubory a napíše první CLAUDE.md. Memory ti ho pak otevře k úpravě.
Chceš to promyslet dřív	/plan /model	V plan režimu navrhne postup a na soubory nesáhne. /model přepne model i pro další konverzace.
Něco se pokazilo	/rewind /doctor	Rewind vrátí úpravy a konverzaci. Doctor projde nastavení a řekne, co máš rozbité nebo zapomenuté.
Vracíš se k rozdělané práci	/resume /diff	Resume otevře dřívější konverzaci. Diff otevře přehled změn, které Claude v souborech udělal.
Napojuješ svoje aplikace	/mcp /permissions	MCP ukáže připojené aplikace a jejich stav. Permissions určí, co smí Claude dělat bez ptaní.
Nevíš, co všechno máš	/help /skills	Help vypíše všechny příkazy. Skills ukáže tvoje vlastní postupy a umí je i vypínat.
Potřebuješ výsledek ven	/copy /export	Copy hodí poslední odpověď do schránky. Export uloží celou konverzaci jako text.

JAK SE TO PÍŠE

Čtyři zprávy za sebou: uklidím, nastavím, spustím svůj skill, zkopíruju výsledek.

Pole, do kterého píšeš Claudovi	LOMÍTKO JE PRVNÍ ZNAK Musí být na začátku zprávy. Uprostřed věty je to obyčejný text a nic se nespustí.
/clear	APLIKACE NAŠEPTÁVÁ Napiš dvě tři písmena a nabídne ti zbytek. Zpaměti to umět nemusíš.
/model opus	TEXT ZA MEZEROU Co dopíšeš za příkaz, dostane jako zadání. Tady jde recenze rovnou do tvého skillu.
/recenze Ubytování super, snídaně slabá.	PAK ENTER Vestavěný příkaz proběhne hned. U tvého skillu se rozjede Claude a pracuje.
/copy	

TŘI DRUHY, CO VYPADAJÍ STEJNĚ

Všechny začínají lomítkem. Uvnitř je to pokaždé něco jiného.

VESTAVĚNÝ /clear /model /context Logika je přímo v aplikaci. /clear i /model dělají vždy to samé, Claude do toho nemluví.	HOTOVÝ SKILL /doctor /code-review Je to připravené zadání, které dostane Claude. Volnější, protože si to musíš promyslet.	TVŮJ VLASTNÍ /recenze /nabídka Složka se souborem SKILL.md, kterou sis napsal. Lomítko je jenom spínač , obsah je ten skill.
---	--	--

TŘI CHYBY, CO DĚLÁ KAŽDÝ

„Když se to pokazí, dám /clear.“ Ten čistí konverzaci, ne soubory. Na vrácení změn je /rewind.	„Nechám jednu konverzaci běžet celý den.“ Čím delší, tím hůř si Claude drží zadání. Nový úkol, nový /clear.	„Musím si ty příkazy zapamatovat.“ Nemusíš. Napiš samotné lomítko a aplikace vypíše všechno, co máš k dispozici.
--	---	--

/ Samotné lomítko na začátku zprávy. Vypíše seznam a našeptává, jak píšeš dál.	/status Ukáže verzi, model, účet a připojení. První otázka, když se Claude chová divně.
--	--

ZAČNI TÍMHLE

Nauč se jediný příkaz: **/clear**. Až dopíšeš jeden úkol, napiš ho a začni další načisto. Většina stížností na to, že Claude „přestal poslouchat“, je jenom přeplněná konverzace, kterou nikdo neuklidil.



SUBAGENT

POMOCNÍK, KTERÝ SI ŠPÍNU ODNESE JINAM

Claude si na vedlejší úkol zavolá pomocníka. Ten ho odpracuje ve vlastní hlavě a vrátí jenom výsledek. **Tobě v konverzaci nezbyde padesát souborů, které si stejně nikdy nepřečteš.**

PROČ HO CHCEŠ

Tři důvody, a každý z nich ti něco šetří.

ŠETŘÍ MÍSTO V HLAVĚ

Hledání a čtení proběhne stranou. Zpátky přijdou tři věty místo třiceti výpisů.

DÁ SE SVÁZAT

Určíš, co smí. Když mu necháš jenom čtení, **nemůže ti nic přepsat ani smazat.**

MŮŽE JET LEVNĚJI

Na mechanickou práci mu nastavíš slabší model. **Stejný výsledek, rychleji a za méně.**

TŘI, KTERÉ UŽ MÁŠ

Nic nezakládáš, tyhle Claude posílá sám, když se to hodí.

EXPLORE

Rychlý hledač. Projde soubory a vrátí, co našel. **Psát a mazat nesmí**, takže ho můžeš pustit bez obav.

PLAN

Průzkumník pro plánovací režim. Sbírá podklady, než ti Claude předloží postup. Taky jen čte.

GENERAL-PURPOSE

Univerzál na složitou práci, která zahrnuje i změny. Má skoro všechny nástroje.

JAK VYPADÁ TVŮJ VLASTNÍ

Kavárna U Kohouta si hlídá konkurenci v okolí. Tohle za ní obchází weby.

Soubor **pruzkum-konkurence.md** ve složce **.claude/agents**

```
---
name: pruzkum-konkurence
description: Projde weby kaváren v okolí
a vrátí ceny a otevírací dobu.
tools: Read, WebFetch, WebSearch
model: haiku
---
```

Jsi průzkumník. U každé kavárny zjistíš:

- cenu espressa a filtrované kávy
- otevírací dobu o víkendu
- čím se chlubí na hlavní stránce

Vrať tabulku, ne odstavce. Nekomentuj to. Když web nenajdeš, napiš to a jeď dál.

NAME

Jméno, kterým si ho zavoláš. Malá písmena a pomlčky, žádná diakritika.

DESCRIPTION

Podle ní Claude pozná, kdy ho poslat. Stejně pravidlo jako u skillu.

TOOLS

Co smí použít. Tady jen čtení a web, takže **nemá čím nic změnit.**

MODEL

Na obcházení webů stačí ten nejlevnější. Přemýšlení nech na hlavní konverzaci.

TEXT POD ČAROU

Jeho zadání. **Tvoji konverzaci nevidí**, takže mu sem napiš i to, co je ti jasné.

KAM SE ULOŽÍ

Jeden soubor, stejná logika jako u skillů.

Pomocník jen pro tenhle projekt

.claude/agents/jmeno.md

Ve složce projektu. Když složku sdílíš s kolegou, má pomocníka taky.

Pomocník do všech projektů

~/ .claude/agents/jmeno.md

Platí všude na tvém počítači. Sem patří ten, kterého chceš mít pořád po ruce.

Soubor nemusíš psát ručně. Řekni **Claudovi, koho potřebuješ, a napíše ho za tebe**. Když v projektu zakládáš pomocníka poprvé, aplikaci pak jednou zavří a otevří.

JAK HO POSLEŠ

Dvě cesty pro tebe, jedna pro Claudea.

ŘEKNEŠ TO VĚTOU

Pošli **pruzkum-konkurence** na tyhle tři weby Claude většinou poslechne. Nejjednodušší cesta.

ZAVINÁČ

@pruzkum-konkurence

Napišeš zavináč a vybereš ho ze seznamu. **Tímhle si ho vynutiš.**

VEZME SI HO SÁM

„Co mají za ceny vedle?“

Pozná to podle description. Proto ji piš, jako by to četl kolega.

KDY HO NEPOSÍLAT

„Musíme si to spolu ujasnit.“

Pomocník neumí ping-pong. Když čekáš doptávání, **zůstaň v hlavní konverzaci.**

„Je to na dvě minuty.“

Startuje s prázdnou hlavou a musí se rozkoukat. U drobností je to zdržení.

„Ať si to pamatuje na potom.“

Ve výchozím stavu si nepamatuje nic. Vrať shrnutí a skončí. **Co má zůstat, patří do CLAUDE.md.**

/tasks

Vypíše, co ti zrovna běží na pozadí.

Ctrl + B

Posle běžící úkol na pozadí. **Ty píšeš dál, pomocník pracuje.**

ZAČNI TÍMHLE

Nezakládej žádného. Až tě příště Claude zavalí výpisem, který jsi číst nechtěl, napiš mu: **„tohle příště pošli přes Explore“**. Vlastního pomocníka si napíšeš, až budeš třikrát chtít to samé hledání.



MCP

KABEL MEZI CLAUDEM A TVÝMI APLIKACEMI

Sám od sebe Claude o tvém Gmailu, kalendáři ani Notionu neví. Přes MCP se do nich připojí a umí z nich číst i do nich psát. **Zkratka** znamená Model Context Protocol, ale to si pamatovat nemusíš.

CO S TÍM UDĚLÁŠ Poznáš to podle toho, že něco pořád kopíruješ z jednoho okna do druhého.

PŘESTANEŠ KOPÍROVAT

„Vytáhni z Notionu zadání a napiš k němu nabídku.“
Claude si to načte sám. Ty nepřepínáš okna a nic neztratíš cestou.

NACHYSTÁ ODPOVĚDI

„Projdí nepřečtené mailý a připrav odpovědi.“
Drafty ti počkají v Gmailu. Před odesláním se tě standardně zeptá.

VIDÍ DO SVÝCH DAT

„Kolik schůzek mám příští týden?“
Kalendář, tabulky, databáze. Odpovídá z reality, ne z odhadu.

JAK HO PŘIPOJÍŠ Nejjednodušší cesta nevede přes Claude Code, ale přes web claude.ai.

KROK 01

Na webu **claude.ai** otevři **Customize** a v něm **Connectors**.

KROK 02

Vyber službu ze seznamu a přihlas se do ní. Potvrdíš, co všechno smí číst.

KROK 03

Otevři Claude Code přihlášený účtem claude.ai. Konektor tam **naskočí sám**.

KROK 04

Ověř si to. V seznamu musí svítit jako připojený.

/mcp

Gmail, Kalendář a Microsoft 365 jde připojit **jenom touhle cestou**, přímo z Claude Code je nepřipojíš. Ostatní služby se dají přidat i příkazem v terminálu, ale nemusíš to umět.

CO SE DÁ PŘIPOJIT Příklady, které jmenuje přímo dokumentace. Serverů jsou dneska stovky.

Mail a kalendář

Gmail, Google Kalendář, Microsoft 365. Čtení, hledání a příprava odpovědí.

Poznámky a úkoly

Notion, Slack, Figma. Zadání si přečte tam, kde ho reálně máš.

Firemní systémy

GitHub, Stripe, HubSpot, Sentry, databáze. Sem už většinou sáhne někdo z IT.

JAK TO VYPADÁ Zjednodušený průběh jedné žádosti, ne přesný vzhled aplikace.

Připojený **Gmail**, jedna zpráva a co se stalo

Ty Najdi mail od Jany o zářijovém termínu a připrav odpověď.

Claude gmail · hledání ve vláknech
našel 1 vlákno, 3. srpna

gmail · založení draftu
draft připravený, neodeslal jsem ho

VIDÍŠ, NA CO SÁHL

U každého kroku je vypsané, kterou službu použil a co v ní udělal.

SCHVALUJEŠ TY

U akce, která něco mění, se zastaví a počká. **Ticho neznamená souhlas.**

DRAFT ČEKÁ NA TEBE

Zůstane ležet v Gmailu. Odeslání je tvoje rozhodnutí.

NEŽ NĚCO PŘIPOJÍŠ Tahle sekce je důležitější než všechno ostatní na plakátu.

POUŠTÍŠ CIZÍHO DO SVÝCH DAT

Server uvidí to, co mu povolíš. **Připoj jen ty, které znáš**, a projdi si, co po tobě chtějí.

TEXT ZVENKU JE JEN TEXT

V cizím mailu nebo na webu může být schovaný pokyn pro Clauda. **Nikdy to není příkaz od tebe.**

ODESLÁNÍ SI HLÍDEJ

U všeho, co jde ven, si nech schvalování. Co smí proběhnout bez ptaní, fukneš v **/permissions**.

TŘI CHYBY, CO DĚLÁ KAŽDÝ

„Připojím všechno, ať to mám.“

Každý server něco stojí a něco vidí. **Připoj jeden a používej ho měsíc.**

„Když to má konektor, umí to všechno.“

Umí to, co server nabízí. Podívej se, jestli jen čte, nebo i píše a odesílá.

„Nefunguje to, je to rozbité.“

Nejdřív se podívej na stav. Většinou jen **vypršelo přihlášení** a stačí se přihlásit znovu.

/mcp

Seznam připojených služeb a jejich stav. **První místo, kam se podívat.**

/permissions

Určíš, co smí Claude udělat bez ptaní a na co se tě má vždycky zeptat.

ZAČNI TÍMhle

Připoj jednu jedinou službu, tu, ze které nejčastěji kopíruješ. Měsíc jenom čti, nic nepiš. **Když si po měsíci nevzpomeneš, že ji máš, zase ji odpoj.**



HOOK

PRAVIDLO, KTERÉ SE NEDÁ UKECAT

Pravidlo v CLAUDE.md si Claude přečte a většinou ho dodrží. Hook se spustí sám, bez ohledu na to, co si Claude zrovna myslí. **Nečeká, až si někdo přečte odrážku, a nedá se přemluvit.**

PROSBA VERSUS ZÁMEK

Tohle je celý důvod, proč hooky existují.

PRAVIDLO V CLAUDE.MD

„Nikdy neupravuj soubor s hesly.“ Claude to čte jako prosbu. Většinou poslechne. Vždycky ne.

HOOK

Sáhne na ten soubor a hook akci zamítne. **Do rozhodování mu nikdo nemluví, ani Claude, ani ty.**

KDY SE SPOUŠTÍ

Událostí je přes třicet. Tyhle jsou ty, kvůli kterým to lidi zakládají.

Když session začíná	SessionStart	Vloží kontext, který má být po ruce pokaždé, i po vyčištění konverzace.
Než odejde tvoje zpráva	UserPromptSubmit	Můžeš ji doplnit o kontext, nebo ji rovnou zastavit.
Než Claude něco spustí	PreToolUse	Tady se dá konkrétní akci zakázat dřív, než se stane. Tady je ta síla.
Když akce doběhla	PostToolUse	Formátování, testy, zápis do logu. Zakázat už se nedá, stalo se to.
Když chce Claude skončit	Stop	Můžeš ho poslat zpátky do práce, když něco nedodělal.

JAK VYPADÁ

Zapíše se do settings.json. Těhle hlídá mazání souborů.

Soubor `.claude/settings.json` ve složce projektu

```
"PreToolUse": [
  {
    "matcher": "Bash",
    "hooks": [
      {
        "type": "command",
        "if": "Bash( rm *)",
        "command": "${CLAUDE_PROJECT_DIR}/.claude/hooks/block-rm.sh"
      }
    ]
  }
]
```

PRETOOLUSE

Kdy se to spustí. Tady těsně předtím, než Claude pustí příkaz.

MATCHER

Hrubé síto. Reaguj jen na spouštění příkazů, ne na psaní souborů.

IF

Jemné síto. Z příkazů si všiměj jenom mazání. Zbytek projde.

COMMAND

Tvůj skript. Vypíše **deny** a akce se nestane, nebo **ask** a zeptá se tebe.

NEŽ SI HO PUSTÍŠ

Tahle sekce je důležitější než všechno ostatní na plakátu.

JE TO PROGRAM S TVÝMI PRÁVY

Nikdo ho neizoluje. Vidí tvoje soubory, proměnné i klíče. **Piš ho stejně opatrně jako cokoliv jiného.**

CIZÍ REPO, CIZÍ HOOK

Hooky z projektu se spustí, jakmile složce důvěřuješ. **Než cizí složce uvěříš, přečti si její .claude/settings.json.**

MODEL NENÍ ZÁMEK

Hook umí rozhodovat i promptem. Ten se dá zmást. Na tvrdou hranici patří skript, ne úvaha.

TŘI CHYBY, CO DĚLÁ KAŽDÝ

„Napišu to do CLAUDE.md, ono to platí.“

Neplatí. Když to musí platit vždycky, **je to hook, ne odrážka.**

„Skript vrátí jedničku a zastaví to.“

Nezastaví. Zastaví ho jen **návratový kód 2**. Kód 1 jen zapíše chybu a jede se dál.

„Přidám hook na všechno.“

Každý je zdržení navíc a další věc, co se může rozbít. Začni jedním.

/hooks

Vypíše všechny hooky i to, ze kterého souboru se berou. **Čti dřív než důvěřuj.**

deny

Odpověď skriptu, která akci zamítne. Vedle ní jsou **allow** a **ask**. **defer** platí jen v **claude** **-p**.

ZAČNI TÍMhle

Nezakládej hooky do zásoby. Vezmi jednu věc, kterou si nemůžeš dovolit ztratit, a zamkni ji. **Jeden hook, který drží, je víc než deset odrážek v CLAUDE.md, které Claude občas přečte a občas ne.**



PLUGIN

KRABICE NA VŠECHNO, CO SIS NASTAVIL

Skilly, pomocníky, hooky a napojení na aplikace zabalíš do jedné složky. Kolega si ji nainstaluje jedním příkazem a má přesně to, co ty. Běžný plugin neumí nic navíc, jenom to zabalí, očísluje a rozdává.

KDY STAČÍ SLOŽKA A KDY KRABICE

Doporučení z dokumentace: začni vlevo, přejdi doprava, až to bude chtít někdo další.

OBYČEJNÁ SLOŽKA .CLAUDE/

Tvoje věci v projektu. Rychlé zkoušení, osobní postupy. Skill se jmenuje **/nabídka**. Sdílíš to kopírováním.

PLUGIN

Verzované, instalovatelné, sdílené. Skill se jmenuje **/muj-plugin:nabídka**, takže se nemůže srazit s cizím.

CO SE DO NĚJ VEJDE

Všechno nepovinné. Klidně zabal jenom jeden skill.

skills/	Postupy. Každý ve vlastní složce se souborem SKILL.md, přesně jako mimo plugin.
agents/	Pomocníci s vlastní hlavou. Jeden soubor na jednoho.
hooks/hooks.json	Zámky na události. Tohle je ta část, kvůli které se cizím pluginům dříve na zoubek.
.mcp.json	Napojení na aplikace. Kolega je má hned, nemusí nic klikat.
bin/	Spustitelné programy, které plugin přidá do cesty, dokud je zapnutý.

JAK VYPADÁ UVNITŘ

Jedna složka, obvykle s manifestem, zbytek volitelný.

Složka muj-plugin kdekoli v disku	PLUGIN.JSON Jméno, popis, verze. Jméno se stane předponou všech skillů uvnitř.
<pre>muj-plugin/ ├── .claude-plugin/ │ └── plugin.json ├── skills/ │ └── nabídka/SKILL.md ├── agents/ │ └── pruzkum.md └── hooks/hooks.json</pre>	DO .CLAUDE-PLUGIN NIC VÍČ Nejčastější chyba. Uvnitř je jenom manifest. Všechny ostatní složky patří o patro výš.
	VERZE ROZHODUJE O UPDATU Když máš v manifestu verzi, ostatním se nová nestáhne, dokud číslo nezvedneš.
	ZKOUŠÍŠ BEZ INSTALACE Spustíš Claude Code s odkazem na složku a plugin se načte jen pro tu session.

ODKUD SE BEROU

Anthropic drží dvě hlavní tržiště. Zbytek internetu není kurátorovaný.

CLAUDE-PLUGINS-OFFICIAL

Hlavní tržiště od Anthropic, většinu pluginů v něm dělají partneři. Přidá se samo při prvním spuštění.

CLAUDE-COMMUNITY

Pluginy třetích stran, které autoři poslali Anthropic. Přidáš si ho ručně.

NEŽ NĚCO NAINSTALUJEŠ

Plugin není jenom text. Může s sebou nést kód, který běží.

INSTALUJEŠ CIZÍ KÓD

Hooky uvnitř se spustí samy a MCP servery uvidí na tvoje data. **Instaluj jen to, čemu věříš.**

PROJDI, CO PŘINESL

Po instalaci se podívej do **/hooks** a **/mcp**. Uvidíš přesně, co ti přibýlo.

TRŽIŠTĚ NENÍ ZÁRUKA

I plugin z tržiště Anthropic běží s tvými právy. Odpovědnost za to, co si pustíš, zůstává na tobě.

TŘI CHYBY, CO DĚLÁ KAŽDÝ

„Plugin bude umět něco navíc.“

Běžně neumí. Je to ta samá věc jako v **.claude/**, jen zabalená. **Krabice, ne motor.**

„Nainstaloval jsem a nic tam není.“

Podívej se na hlášku po instalaci. Občas chce ještě **/reload-plugins**.

„Udělám plugin hned na začátku.“

Zbytečná práce. Nej dřív ať to měsíc funguje tobě, teprve pak to zabaluj.

/plugin

Správce. Prohlížení tržišť, instalace, zapínání a vypínání, záložka s chybami.

/reload-plugins

Načte změny bez restartu. Používáš při vývoji vlastního.

ZAČNI TÍMHLE

Žádný nedělej. Otevři **/plugin**, projdi kurátorovanou nabídku a nainstaluj jeden, který ti ušetří práci. **Vlastní zabaluj až ve chvíli, kdy si o tvoje nastavení někdo řekne podruhé.**